

Pedestrian Detection and Tracking Using Three-dimensional LADAR Data

Luis E. Navarro-Serment, Christoph Mertz, and Martial Hebert

Abstract

The approach investigated in this work employs three-dimensional LADAR measurement to detect and track pedestrians over time. The sensor is employed on a moving vehicle. The algorithm quickly detects the objects which have the potential of being humans using a subset of these points, and then classifies each object using statistical pattern recognition techniques. The algorithm uses geometric and motion features to recognize human signatures. The perceptual capabilities described form the basis for safe and robust navigation in autonomous vehicles, necessary to safeguard pedestrians operating in the vicinity of a moving robotic vehicle.

Keywords

Human detection, 3D LADAR sensing and perception, field and service robotics

1. Introduction

The ability to avoid colliding with other objects is essential in autonomous vehicles, especially in cases where they operate in close proximity to people. The timely detection of a pedestrian makes the vehicle aware of a potential danger in its vicinity, and allows it to modify its course accordingly.

Robust and reliable detection and tracking has attracted a lot of attention in recent years, driven by applications such as pedestrian protection, vehicle platooning, and autonomous driving. This is a difficult problem, which becomes even harder when the sensors (e.g. optical sensors, radar, laser scanners) are mounted on the vehicle rather than being fixed, such as in traffic monitoring systems. Effective detection and tracking require accurate measurements of object position and motion, even when the sensor itself is moving. Range sensors are well suited to this problem because a first-order motion correction can be made by simply subtracting out self-motion from range measurements. Unfortunately, merely subtracting out ego-motion does not eliminate all of the effects of movement because the perceived object's shape changes as different aspects of the object come into view, and this change can easily be misinterpreted as motion. In addition, the perceived appearance of an object depends on its pose, and can also be affected by nearby objects. Finally, complex outdoor environments frequently involve cluttered backgrounds, susceptibility to ground returns in uneven terrains, and unpredictable interaction between traffic participants.

There is a large body of work done using laser line scanners as the primary sensor for pedestrian detection and tracking. In our group, we have developed detection and

tracking systems using SICKTM laser line scanners; these implementations work well in situations where the ground is relatively flat (Navarro-Serment et al. 2006). However, a three-dimensional (3D) LADAR (i.e. one which produces a set of 3D points or point cloud) captures a more complete representation of the environment and the objects within it. In previous work (Navarro-Serment et al. 2008), we presented an algorithm that detects pedestrians from 3D data. Its main improvement over the version with 2D data was that it constructs a ground elevation map, and uses it to eliminate ground returns. This allows pedestrian detection even when the surrounding ground is uneven. To classify the humans the algorithm uses motion, size, and noise features. Persons are classified well as long as they are moving. However, there are still too many false positives when classifying stationary humans.

In this paper, we describe a strategy to detect and classify humans using the full 3D point cloud of the object. This will improve the classification of both moving and static pedestrians. However, the improvement will be most significant for static humans. The algorithm quickly detects the objects that have the potential of being humans using a subset of the point cloud, and then classifies each object using statistical pattern recognition techniques. We present experimental results of detection performance using 3D

The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA.

Corresponding author:

Luis E. Navarro-Serment, The Robotics Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA.
Email: lenscmu@ri.cmu.edu



Fig. 1. The Demo III Experimental Unmanned Vehicle (named XUV) used in our tests. The vehicle shown is equipped with a 3D LADAR mounted on a pan-tilt platform.

LADAR, which were obtained from field tests performed on a Demo III Experimental Unmanned Vehicle (XUV) (Shoemaker and Bornstein 1998) (Figure 1).

2. Related Work

The most commonly used approaches to detection and tracking of moving objects for vehicular applications involve both passive and active range sensors (Hebert 2000). Passive sensors, such as video cameras, acquire data in a non-intrusive way. Sun et al. (2006) presented an extensive review of vision-based on-road vehicle detection systems.

Active range sensors, such as radar and LADAR, are capable of measuring distances directly without requiring powerful computing resources. In particular, recent models of laser scanners are capable of gathering high-resolution data at high scanning speeds, and are available in enclosures suitable for vehicular applications. The closest work related to our approach involves the use of laser line scanners. Fuerstenberg et al. (2002) describe the application of a multi-layered laser scanner for pedestrian classification. Vehicle odometry is used to estimate self-motion, to remove the effects of sensor motion. A Kalman filter estimates the object velocity. Tracked objects are classified as car, pedestrian, etc., based on their shape and behavior over time. This group also produced a second system (Streller et al. 2002), in which a Kalman filter estimates motion based on the change in position of an object's estimated center point. Object classification is used to fit a class-specific prior rectangular model to the points. Although not mentioned explicitly, this appears to be an approach to reducing shape-change motion artifacts. The success of this technique depends on the correctness of the classification. Each object class also has distinct fixed Kalman filter parameters. A multi-hypothesis approach is used to mitigate the effect of classification error. The emphasis of both efforts is on single-LADAR systems, and multi-scanner fusion is not considered.

A system that deals particularly well with crowds of people is described in Cui et al. (2007). The authors use a feature extraction method based on accumulated distribution of successive laser frames. The final tracker is based on the combination of independent Kalman filter and Rao-Blackwellized Monte Carlo data association filter.

Some researchers have applied classification techniques to the detection and tracking problem. The approach reported by Arras et al. (2007) applies AdaBoost to train a strong classifier from simple features of groups of neighboring points. This work focuses on 2D range measurements. Examples using 3D data include Morris et al. (2006), where 3D scans are automatically clustered into objects and modeled using a surface density function. A Bhattacharya similarity measure is optimized to register subsequent views of each object enabling good discrimination and tracking, and hence detection of moving objects. Howard et al. (2007) describe a pedestrian detection system which uses stereo vision to produce a 3D point cloud, and then classifies the cloud according to the point shape distribution considering the first two central moments of the 2D projections using a naive Bayes classifier. Motion is also used as a cue for human detection.

Thornton et al. (2008) report an algorithm capable of detecting both stationary and moving humans. Their approach uses multi-sensor modalities including 3D LADAR and long-wave infrared video (LWIR). Similarly, in Thornton and Patil (2008) the same research group presented a technique for detecting humans that combines the use of 3D LADAR and visible spectrum imagery. In both efforts the authors employed a 2D template to extract features from the shape of an object. Among other differences, as opposed to our work, they extract a shape template from the projection in only one plane, and compute a measure of how uniformly distributed the returns are across the template.

3. Algorithm Description

In this section, the algorithm for pedestrian detection and classification is described. In our approach, since operation in real time is a chief concern, we perform object detection and tracking in a 2D data subset first, and then use the object's position and size information to partition the set of 3D measurements into smaller groups, for further analysis. We describe these steps in the following sections.

3.1. Projection onto a 2D Plane

The LADAR data from a single cycle is corrected for vehicle motion using an inertial navigation system. Then, to reduce the computational cost of processing the entire point cloud, we isolate a 2D virtual slice, which contains only points located at a certain height above ground. As shown in Figure 2, a 3D scanner produces a point cloud, from which a "slice" is projected onto the 2D plane, resulting in a virtual scan line. This scan line is a vector of range measurements coming from consecutive bearings, which

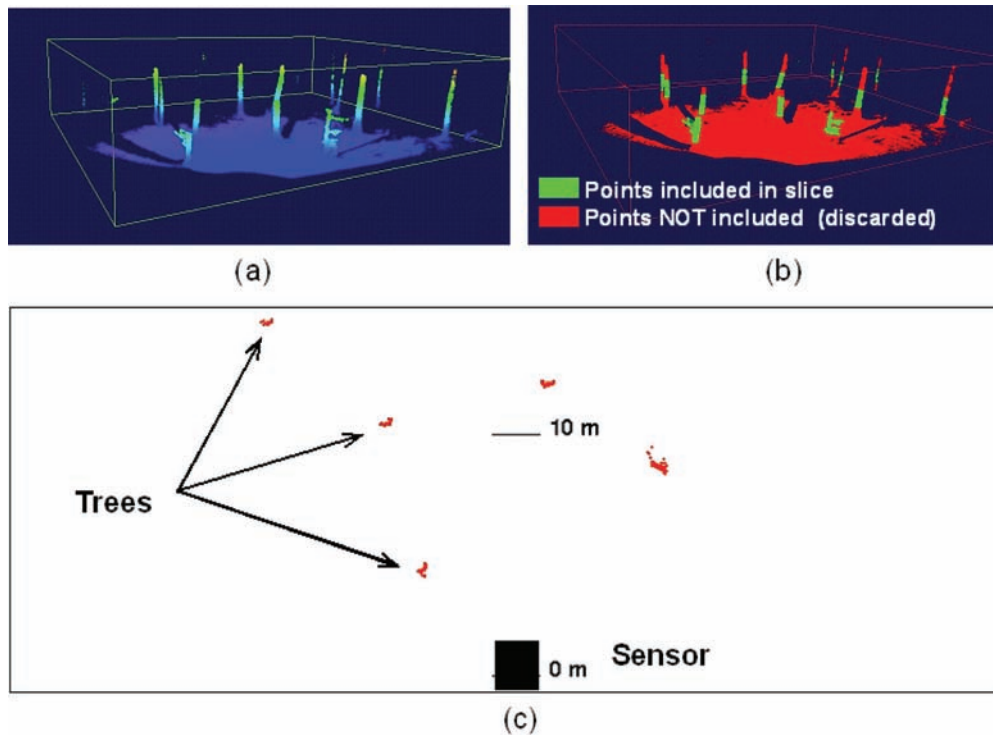


Fig. 2. Projection of the virtual scan line. (a) A point cloud collected from the environment illustrated in Figure 1 is shown. The points located within a certain height above ground are projected onto a 2D plane, and processed as if it were a single scan line (b). The resulting projection is shown in (c), top view.

resembles the kind of data reported directly by a line scanner such as the SICKTM laser scanner. This is done by collapsing into the plane all of the points residing within the slice, which is defined by its height above the ground.

The ground elevation is stored in a scrolling grid that contains accumulated LADAR points and is centered at the vehicle's current position. The mean and standard deviation of the heights of all scan points that are inside each cell are computed, and the elevation is then calculated by subtracting one standard deviation from the average height of all of the points in the cell. To increase robustness against sensor misalignment, a forgetting factor is applied to the height measurements, so more recent points have a heavier weight. The key properties of this simple algorithm are that mean and standard deviations can be calculated recursively, and that the elevation is never below the lowest point while still having about 80% of the points above ground.

The system adapts to different environments by varying the shape of the sensing plane, i.e. by adjusting the height of the slice from which points are projected onto a 2D plane. Spurious measurements produced by ground returns are avoided by searching for measurements at a constant height above the ground. Since our research was conducted mostly in an open outdoor environment, we did not encounter overhanging structures such as overpaths or ceilings. Dealing with these structures might be the topic of future work.

3.2. Object Segmentation and Tracking

Once a virtual scan line is obtained, it undergoes the processes of object segmentation, and object tracking. The first step is segmentation, whose purpose is to determine which of the points contained in the scan line are likely to belong to the same object. We use a distance threshold segmentation process, which clusters the points that fall within a fixed distance from each other. Each object is represented by the center of the bounding box enclosing the points belonging to the object. The object's size is represented by the length of the diagonal of the bounding box.

The objects detected in the current scan are then tested for association with objects from previous scans. If the bounding boxes of two objects from current and past scans overlap, they are considered to be the same object. For two objects to overlap, at least one actual measurement point must fall inside the object's bounding box, taking into account its predicted motion path. Lines or corners are fit to the points, and the corner and end points are the feature points that are fed into a Kalman filter, which estimates the position and velocity of the object¹. These estimates are stored and made available to subsequent steps in the detection process.

The tracking algorithm attempts to track objects through temporary loss. If an object disappears, e.g. because it becomes occluded by another object or goes out of the field of view, it is kept for several cycles and its

position is updated according to its last motion estimate. When the object emerges again and its current position is close to the estimated position, the object is associated with the old track.

3.3. Motion Features

After detecting and tracking objects using the virtual scan line we can compute a *motion score* (MS). The MS is a measure of how confident the algorithm is that the detected object is a human, based on four motion-related variables: the object's size, the distance it has traveled, and the variations in the object's size and velocity. The size test discriminates against large objects such as cars and walls. The distance traveled test discriminates against stationary objects such as barrels and posts. The variation tests discriminate against vegetation, since their appearance changes a lot due to their porous and flexible nature. The individual results of these tests are scored, and then used to calculate the MS.

3.3.1. Calculation of the MS. The four measures that go into the MS are the size, the distance traveled (d_t), and the variation in the size (σ_s) and the velocity (σ_v). The variation is the variance computed over the last 14 cycles.

The MS is the product of the scores of the size and distance traveled test and the square root of the scores of the variance tests:

$$MS = S_{d_t} S_{size} \sqrt{S_{\sigma_s} S_{\sigma_v}}. \quad (1)$$

We take the square root of the variance tests so that together they have the same weight as the other two tests.

For size and variance tests the score is calculated as follows:

$$S_{size, \sigma_s, \sigma_v}(x) = \begin{cases} 0 & x \geq a_e \\ \frac{a_e - x}{a_e - a_b} & a_e > x > a_b \\ 1 & x \leq a_e \end{cases} \quad (2)$$

and for the distance traveled test the score is

$$S_{d_t}(x) = \begin{cases} 0.75 \frac{a_b + x}{2a_b} & S_1 < 1, x < a_b \\ 0.75 & S_1 < 1, x \geq a_b \\ 0.75 + \frac{x}{4a_m} & S_1 = 1, a_m > x > a_b \\ 1 & \text{otherwise} \end{cases} \quad (3)$$

where

$$S_1 = S_{size} \sqrt{S_{\sigma_s} S_{\sigma_v}}.$$

The values used for the thresholds a_b , a_e , and a_m were determined experimentally, and are listed in Table 1.

The reasoning behind these calculations is as follows. The a_b values are the thresholds for the best accuracy, i.e. the weighted average of true positive and true negative rates, is maximal. Here a_e are the extreme values, i.e. the largest values observed for humans. For the size the numbers agree with straightforward intuition: 1 m is slightly larger than a normal cross section of a person, and 2 m is

the span of outstretched arms. The calculation for distance traveled test is different. In contrast to the other measures, the larger the distance traveled is, the more confident the algorithm can be that the object is a human. But also, a pedestrian can have a distance traveled of 0 m, so there is no upper or lower limit that can discriminate against a person; instead, it is possible to discriminate against fixed objects. The value a_m is the extreme value we observed for fixed objects, which can have the appearance of movement because of miss-association.

3.4. Geometric Features

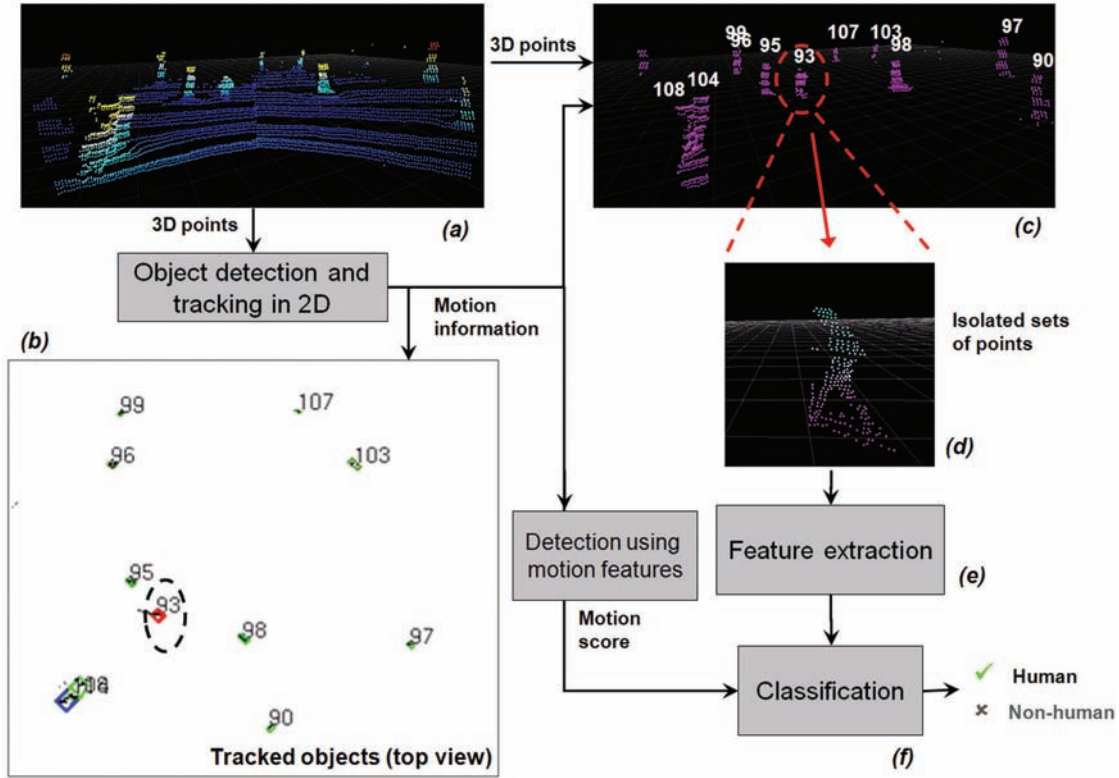
To discriminate better against static structures, we also compute a group of distinguishing geometric features from the set of points belonging to each object being tracked in 2D, and then feed these features to a classifier, which determines whether the object is a human or not. This concept is depicted in Figure 3.

As shown in Figure 3(a), the process starts when a point cloud is read from the sensor. We define $Z_j = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ as the set of N points contained in a frame collected at time t_j , whose elements are represented by Cartesian coordinates $\mathbf{x} = (x, y, z)$. The points corresponding to one frame are shown, and are colored according to their height above ground. To avoid the computational cost of processing the entire point cloud, we extract a 2D virtual slice, as described in Section 3.1 (Figure 3(b)). For each one of the M objects being tracked at time t_j , its position, velocity, and size estimates are used to compute its MS. In addition, the object's position and size information are used to isolate, from the original point cloud, only those points corresponding to potential humans, as shown in Figure 3(c). The ground elevation is used to remove the lowest points, to ensure that only the points belonging to the object are extracted. In this way, the 3D information corresponding to each object is recovered in the form of smaller sets of points. At this point, we have a collection of M sets $\{S_1, S_2, \dots, S_M\}$, where $S_{i \in \{1, 2, \dots, M\}} \subset Z_j$. A feature vector is computed from each of these sets (Figure 3(d) and (e)), and then fed to a classifier that determines for each object whether it is a human or not, Figure 3(f). This decision is made for each object, and is based on the most recent set of points collected from the sensor. The classifier also takes into account the information used to calculate the MS; this is described in a subsequent section.

A set of features is computed with the purpose of extracting the most informative signatures of a human from the 3D data. The legs are particularly distinctive of the human figure, so the algorithm computes statistical descriptions from points located around the legs. Similar descriptions are computed from the trunk area, representing the upper body. In addition, the moment of inertia tensor is used to capture the overall distribution of all points. Finally, to include the general shape of the human figure, we

Table 1. Thresholds for Motion Score Tests

	a_b	a_e	a_m
Size (m)	1.0	2.0	—
Size variation (m ²)	0.035	0.45	—
Velocity variation ((m/s) ²)	0.01	0.1	—
Distance traveled (m)	1.5	—	3.0

**Fig. 3.** Improved pedestrian detection. Geometric features present in subsets of the point cloud are used by a classifier to distinguish pedestrians from static objects.

compute the normalized 2D histograms on two planes aligned with the gravity vector.

3.4.1. Feature Extraction. Let $S_k = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ be the set of points belonging to the object k , whose elements are represented by Cartesian coordinates $\mathbf{x} = (x, y, z)$. A set of suitable features is computed from S_k , as depicted in Figure 3(d), which constitutes a profile of the object.

We begin by performing principal component analysis (PCA) using all of the elements of S_k , to identify the statistical patterns in the 3D data (see Figure 4). This involves the subtraction of the mean $\mathbf{m}$ from each of the three data dimensions. From this new data set with zero mean, we calculate the covariance matrix $\Sigma \in \mathbb{R}^{3 \times 3}$ (Equation (4)), and the normalized moment of inertia tensor $\mathbf{M} \in \mathbb{R}^{3 \times 3}$ (Equation (5)), treating all points as unit point masses:

$$\Sigma = \frac{1}{n-1} \sum_{k=1}^n (\mathbf{x}_k - \mathbf{m})(\mathbf{x}_k - \mathbf{m})^T, \quad (4)$$

$$\mathbf{M} = \begin{bmatrix} \sum_{k=1}^n (y_k^2 + z_k^2) & -\sum_{k=1}^n x_k y_k & -\sum_{k=1}^n x_k z_k \\ -\sum_{k=1}^n x_k y_k & \sum_{k=1}^n (x_k^2 + z_k^2) & -\sum_{k=1}^n y_k z_k \\ -\sum_{k=1}^n x_k z_k & -\sum_{k=1}^n y_k z_k & \sum_{k=1}^n (x_k^2 + y_k^2) \end{bmatrix}. \quad (5)$$

Since both Σ and $\mathbf{M}$ are symmetric, we only use six elements from each as features.

Resulting from the PCA are three pairs of eigenvectors and eigenvalues, sorted according to decreasing eigenvalue. Call these eigenvectors $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$, with their corresponding eigenvalues $\lambda_1 > \lambda_2 > \lambda_3$. The principal component $\mathbf{e}_1$ is expected to be vertically aligned with the person's body. Together with the second largest component $\mathbf{e}_2$, it forms the main plane (Figure 4, center top), and

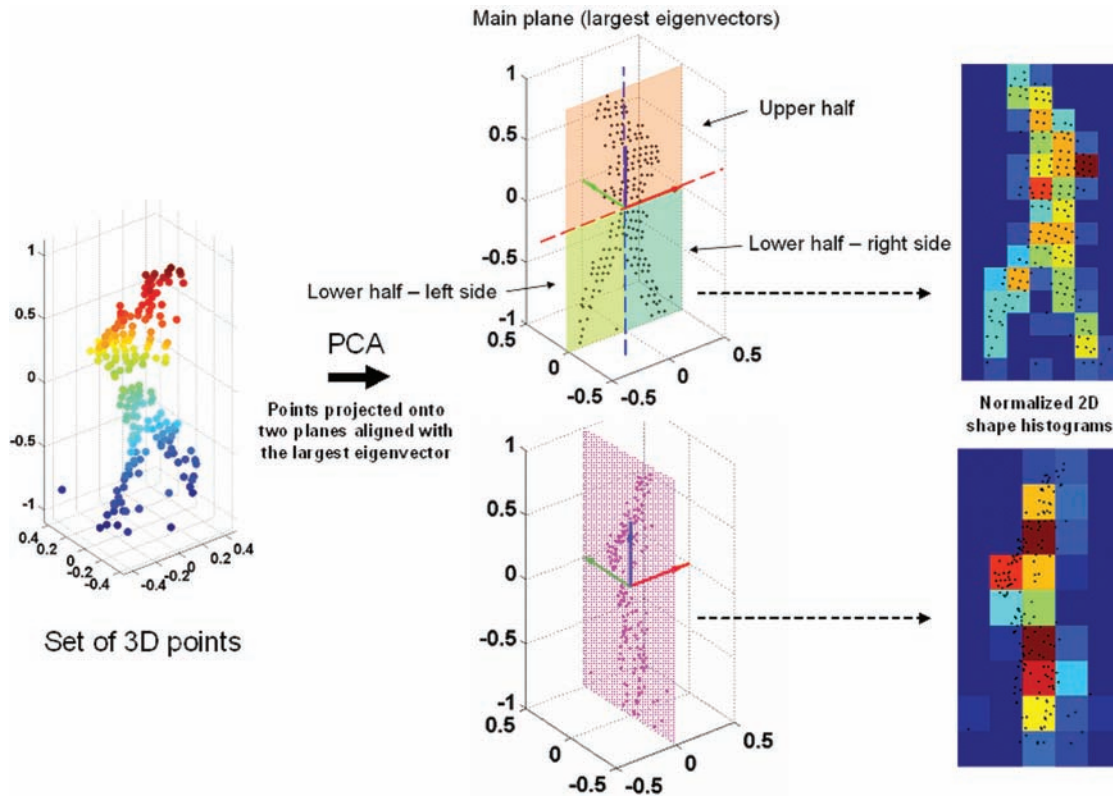


Fig. 4. A set of geometric features is computed from the set of points belonging to an object.

also forms the secondary plane with the smallest component, e_3 (Figure 4, center bottom). We then transform the original data into two representations using each pair of components e_1, e_2 and e_1, e_3 , from which we proceed to compute additional features (the third possible representation, i.e. using the two smallest components e_2, e_3 , is not used).

We focus on the points included in the main plane, to analyze the patterns that would correspond to the legs and trunk of a pedestrian, as shown in Figure 4, center top. These zones are the upper half, and the left and right lower halves. After separating the points into these zones, we calculate the covariance matrix (in two dimensions) over the transformed points laying inside each zone. This results in nine additional features (three unique values from each zone).

Finally, we compute a normalized 2D histogram for each of the two principal planes (Figure 4, right), to capture the shape of the object. We use 14×7 bins for the main plane (defined by e_1 and e_2), and 9×5 for the secondary plane (formed by e_1 and e_3). These histograms are centered at the origin of the basis for the transformed data. The bins are square in both cases, and their size is fixed (0.15 m for the main plane and 0.25 m for the secondary plane). The size and number of bins was experimentally chosen. Each bin is used as a feature, so there are 143 features representing the shape. A total of 164 geometric features are determined for each object.

Although it is possible to compute these features with at least two points on each of the zones (i.e. upper half, and

the left and right lower halves, main plane), the information encoded will not be enough to adequately capture the geometry of the object. Consequently, the geometric features are computed only when at least 15 points per target are available. This threshold was determined experimentally.

3.5. Human Detection

The classifier shown in Figure 3(f), which is composed of two independent support vector machines (SVM) (Burges 1998), determines for each object whether it is human or not (Figure 5). The first classifier is a SVM that receives the vector of 164 geometric features computed directly from S_k , and scores how closely the set matches a human shape. We call this the *geometric score* (GS). The GS is particularly effective for detecting static pedestrians. Similarly, the features used to determine the MS (i.e. object's size, the distance it has traveled, and the corresponding size and motion noises) contain valuable information about the motion of the target. Together with the GS, these features are fed to a second SVM, whose output represents the distance to the decision surface of the SVM. The *strength of detection* (SOD), the total measure of how strongly the algorithm rates the object as being a human, is calculated as the logistic function of the distance to the decision surface. This number is reported for each object. If the GS cannot be computed (e.g. only insufficient data from a distant target is available), then the MS is reported as the SOD for

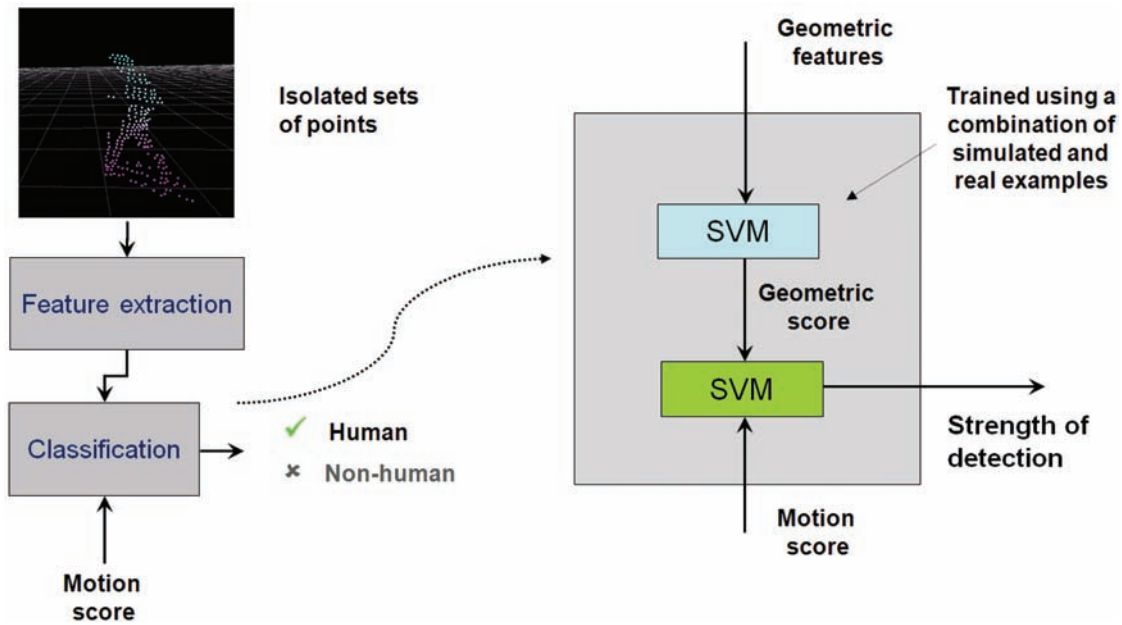


Fig. 5. In the classification step, a classifier, which is composed of two independent Support Vector Machines (SVM), determines the

that object. The objects evaluated with a SOD larger than a given threshold are classified as humans.

The use of the SOD is motivated by several goals: first, the SOD supports the fusion of detections from different systems by serving as a common currency for comparison. Second, different applications may need to increase or relax the rigorosity of the detection uncertainty to adapt to the mission requirements. Third, the comprehensive nature of the SOD helps to maintain a low count in the number of parameters to adjust when tuning the algorithm for other sensors.

3.5.1. Training. We trained the GS classifier using a combination of simulated and real examples. Because it is impossible to collect enough real data to evaluate perception algorithms in all possible situations, we have created a simulator capable of producing synthetic examples of sensor data. The simulator uses a ray tracing engine to generate a set of ray intersections between the sensor and the objects in the scene to simulate. This information is then used to produce synthetic LADAR measurements according to a set of parameters for a particular sensor, as shown in Figure 6. We trained the GS classifier using over 3,500 examples (27.4% humans, 72.6% non-humans). The human set included 62% of simulated examples. The set also included 6% of examples of pedestrians partially seen by the LADAR (i.e. partially inside the LADAR's field of view, or partially occluded). The second classifier was trained using only real examples, since the motion and size noises used to determine the MS are of a dynamic nature and consequently harder to simulate efficiently (over 46,000 examples: 6% humans, 94% non-humans).

The training sets are unbalanced in terms of their positive versus negative example ratio. In the real world the system typically deals with many more negative examples and also the variety of non-human objects is larger. Attempting to create a balanced training set will inevitably result in an incomplete representation of the non-human set. We chose instead to use as many negative examples as were practically available to us, from both simulated and real data.

We trained both SVMs using a five-fold cross validation procedure. We found that both radial basis function (RBF) and polynomial kernels resulted in similar levels of classification performance. After multiple tests, we determined that a RBF kernel was the best for the calculation of the GS, while a polynomial kernel was preferred for the second classifier. The algorithm is executed in real time, and its runtime scales linearly with the number of objects in the scene.

4. Experimental Results

4.1. Target Platform

This section presents the results of several experimental runs. These results were obtained from field tests performed on a Demo III XUV (Shoemaker and Bornstein (1998)). The data comes from 14 different runs, where the variations include static and moving vehicles, pavement and off-road driving, and pedestrians standing, walking, or jogging. The data was taken at 17 Hz, and the average duration of each run was about 1 minute. There were altogether 48 humans and 1,075 non-human objects, where those which came in and out of the field of view were counted twice. The ground truth was produced by

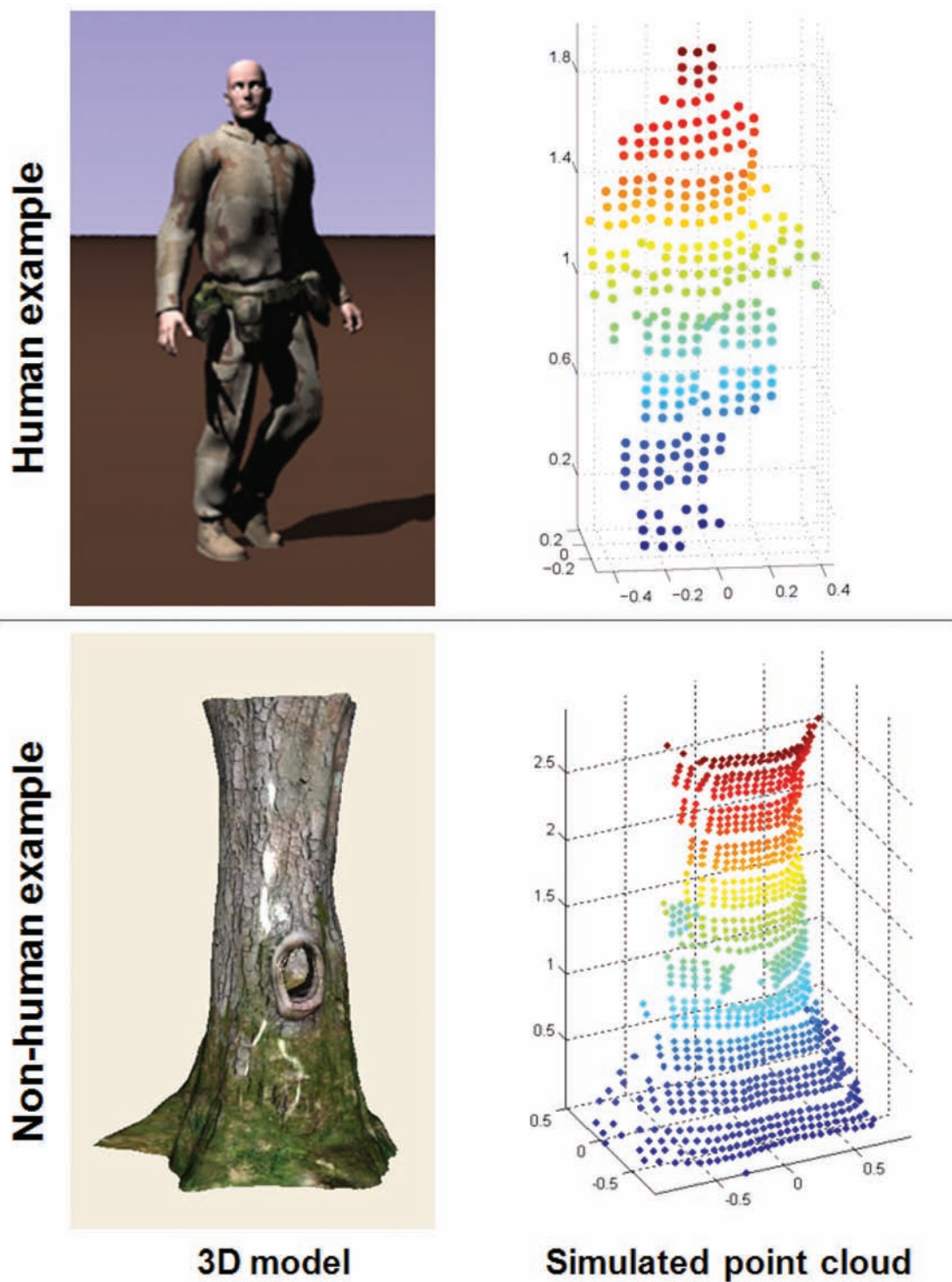


Fig. 6. Simulated object models (left), and their corresponding point clouds (right).

labelling the data by hand. These data sets are annotated with bounding boxes of ground truth objects. A detection is declared correct based on the overlap between the bounding boxes of the detected and ground truth objects. The detection performance is evaluated using receiver operating characteristic (ROC) and precision-recall curves illustrating both per-cycle and per-run performance².

In the upper part of Figure 7 the ROC curve and the precision-recall curves are shown. Each human in one cycle is a positive example and each non-human object in one cycle is a negative example. There are about 6,300 positive and 60,000 negative examples. These plots illustrate the current performance of our system. The dot-dashed traces indicate the MS score alone. The dashed traces indicate the

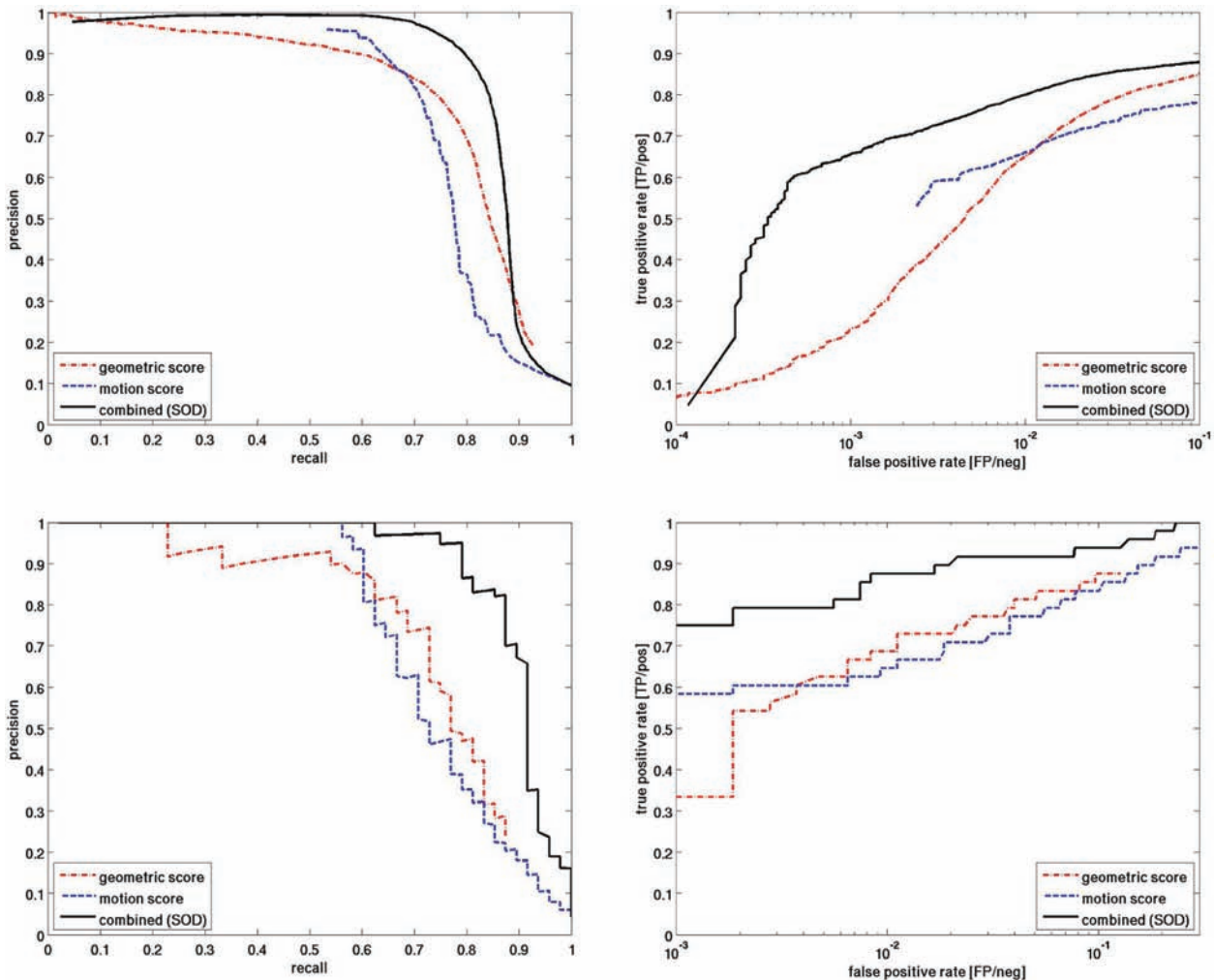


Fig. 7. The plots on the left show the precision-recall and those on the right the ROC curves. Shown are the curves produced by using the geometric score (dashed), the motion score (dot-dashed) and the combined (solid). For the upper curves each object in each cycle and in the lower curves each object in the full run is counted as one example.

geometric score, i.e. the classification using the geometric features computed directly from the object's set of 3D points, but without any motion clues. As seen in the plots, neither algorithm by itself outperforms the other throughout the entire operational range. For low false-positive rates the MS is better, while the GS is better at high false-positive rates. As we mentioned earlier, the MS works for static humans at the expense of high false-positive rates. The synergistic combination of both results has significantly better performance, as indicated by the solid traces.

An alternative representation of ROC and precision-recall is shown in the lower part of Figure 7, where each object is counted per run. The score of an object is the mean of the score over all cycles, with a minimum of 10 cycles. As mentioned above, there are 48 humans (= positive examples) and 1,075 other objects (negative examples). A noteworthy operating point is where there are basically no false positives (rate is 10^{-3}) and still the true-positive rate is 0.75.

The breakdown of detection performance by range is shown in Figure 8. As seen in the plots, the best performance is obtained in the 10-20 m range. This is expected, because humans in close proximity to the sensor are frequently only partially seen (i.e. illuminated) by it. Similarly, since the number of LADAR returns decreases significantly with range, pedestrians farther away from the sensor are represented with smaller numbers of points, which results in inferior detection performance. In the 10-20 m range, pedestrians are typically completely imaged by the sensor with adequate point density, resulting in better classification performance.

The performance of the GS in terms of distance is presented in Figure 9. As expected, the GS performs best in the range of 10-20 m, where humans can be fully seen by the sensor with enough points per target. Similarly, at short distances (0-10 m) the performance is slightly decreased, since many of the humans are only partially seen by the sensor. However, at larger distances (20-30 m), the algorithm attempts to identify statistical patterns in the data using fewer

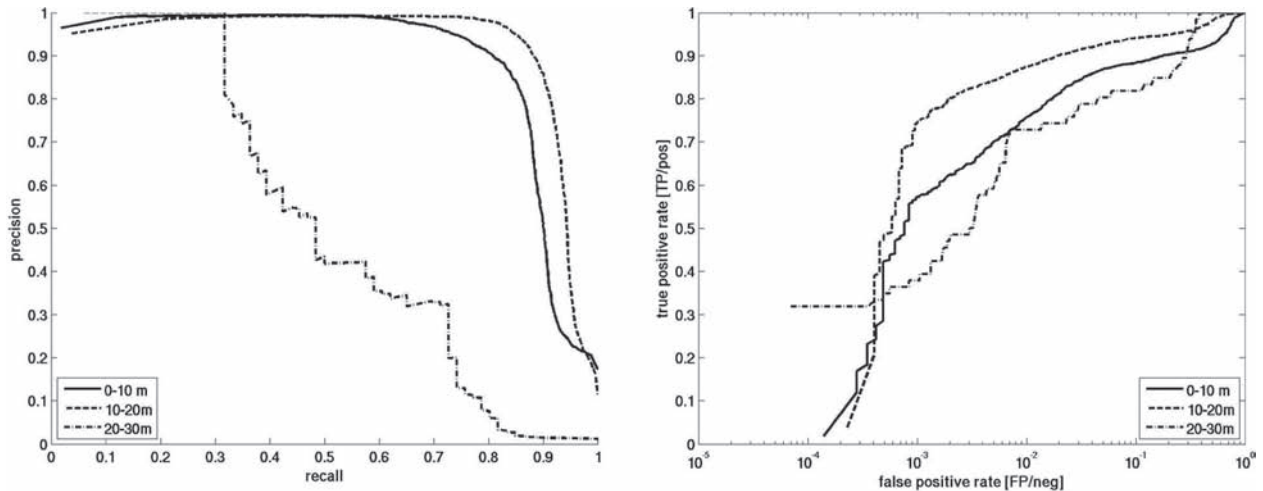


Fig. 8. Detection performance by range. Precision-recall (left) and ROC (right) curves are shown, describing detection performance in the 0-10 m (solid), 10-20 m (dashed), and 20-30 m (dot-dashed) ranges.

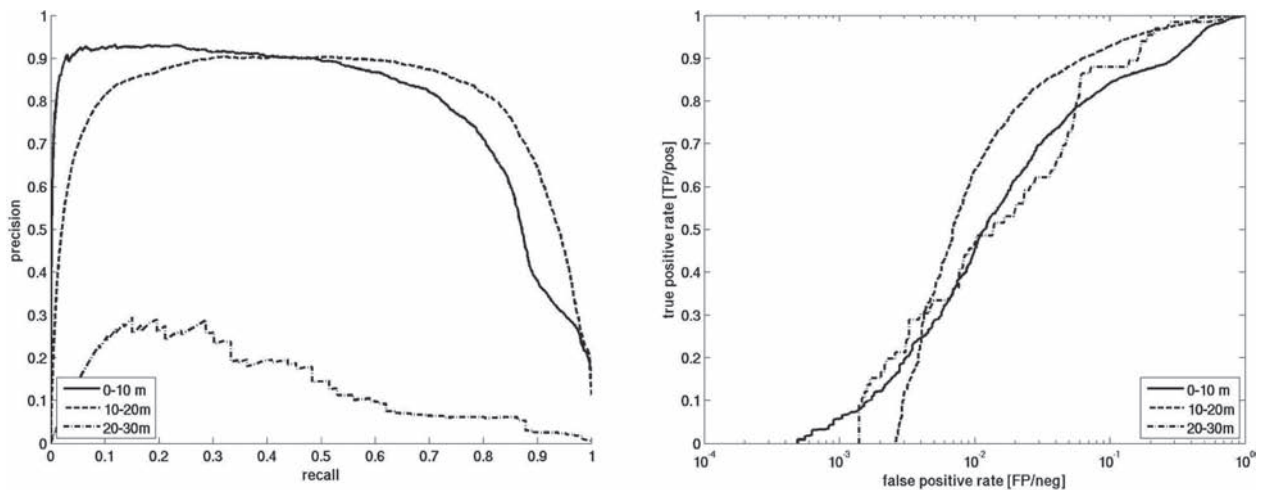


Fig. 9. Performance of geometric score (GS) by range. Precision-recall (left) and ROC (right) curves are shown, describing GS-based detection performance in the 0-10 m (solid), 10-20 m (dashed), and 20-30 m (dot-dashed) ranges.

measurements, which results in a degradation of performance. In addition, at longer distances there are many more negative examples because a larger area is covered by the sensor, which explains why the precision-recall curve for the long distance range is especially low, whereas in the ROC curve the performances at longer and shorter distances look fairly similar.

Figure 10 shows GSs for several examples of humans and objects, where 1 is the highest possible and 0 the lowest possible. The very best is a human at 10 m distance with a high score of 0.925 (Figure 10(a)). At very short distance the human is only partially visible (Figure 10(b)) and the score is 0.624. In the same picture a fence is partially seen at scores 0.315. Assuming we use 0.5 as the dividing value between humans and objects, these examples are correct detections. Figure 10(c) shows a human at 20 m with a score of 0.257. The reason for the low score is the low number of points. A false positive detection is shown in Figure 10(d), it is a tree that is rated at 0.777.

Finally, pedestrians in non-upright positions usually result in false-negative detections. This is expected, because of the particular attention paid to legs and torso when extracting geometric features. An exception to this is the case of kneeling humans, which we have been able to detect consistently, though they are usually borderline classified as humans. Solving these problems is the focus of current research.

4.2. Alternate Sensor

We also tested our algorithm with data that is different from the LADAR data collected with the Demo III XUV. A research group at Carnegie Mellon University made Velodyne LADAR data available to us. Although both sensors produce a point cloud representing the environment, these representations differ in the density of measurements, the organization of the data (e.g. the LADAR image data

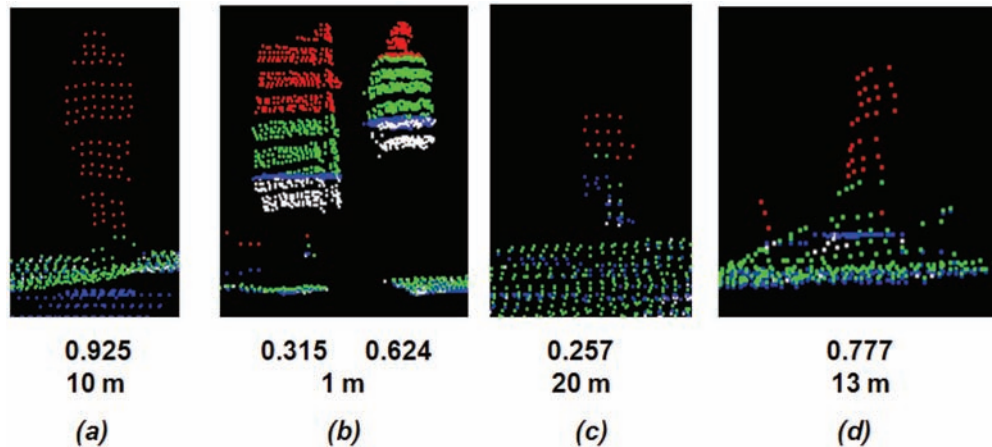


Fig. 10. Geometric score (GS) versus distance. Examples of point clouds from several objects collected from several distances, and their corresponding GS.

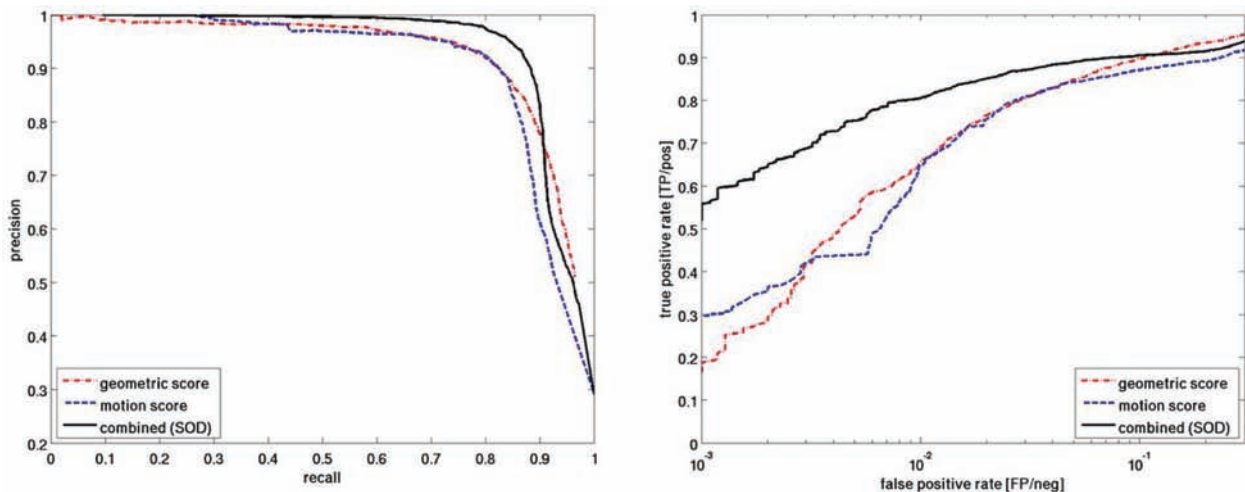


Fig. 12. Evaluation using Velodyne™ data. The plots on the left show the precision-recall and those on the right the ROC curves. Shown are the curves produced by using the geometric score (dashed), the motion score (dot-dashed) and the combined (SOD).

may contain gaps every few rows, as seen in Figure 3(a), measurement noise, sensitivity to target reflectivity, etc. A snapshot of the testing environment and the corresponding LADAR data is shown in Figure 11.

We converted the data into a format that is usable by our system, and tested our system's performance using this data. The results are shown in Figure 12.

Each object in each cycle counts as one example. There were 943 cycles. We had 4,468 positive examples and 10,883 negative examples. Variations of the experiments, as shown in Figure 11, include container boxes, moving vehicles, dirt road driving, and pedestrians standing, or walking. The ground truth was produced by labelling the data by hand.

It is worth mentioning that these results, which are comparable to the results obtained with the target platform, were obtained without adjusting any parameters. The SVM models, which were trained using a combination of

simulated and real Demo III XUV examples, were not updated for use with the Velodyne™. This performance provides evidence of the flexibility of our classification algorithm.

5. Conclusion

We described a pedestrian detection and tracking system using only 3D data. The approach uses a combination of geometric and motion features to recognize human signatures, and clearly improves the detection performance achieved in our previous work. The algorithm performs well with different sensors without requiring major modifications.

The set of features used to determine the human and motion scores was designed to detect humans in upright positions. To increase the robustness of detection of humans in other postures, in future research we will

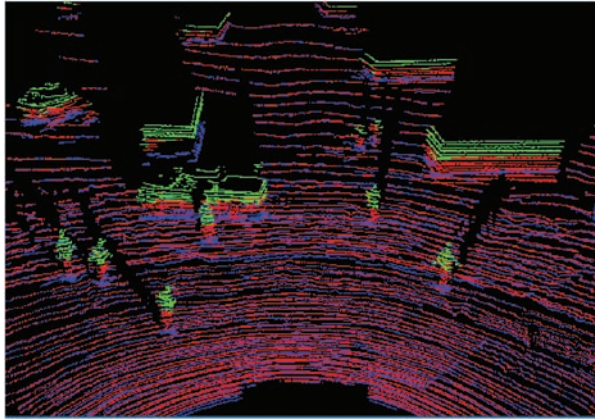


Fig. 11. A Velodyne™ LADAR is mounted on a flatbed truck, which traverses the test environment. A snapshot of the LADAR data is shown.

investigate ways of extracting signatures from the point cloud that are invariant to deformations of the human body.

The experiments described were carried out in moderately cluttered environments. However, more cluttered environments present more challenging situations, where detection performance will decrease. For example, if a person was standing close to an object (e.g. less than 1 m from another object) or next to another person, it may not be properly segmented, and the isolated set of points representing the person could be contaminated by points from the object, or even missed altogether, resulting in incorrect classifications. More clutter also means more partial occlusions. As discussed above, the system does work for moderately occluded persons, however at a decreased performance. One approach to deal with more cluttered scenarios is to focus on improving the object segmentation process by making it more capable of separating different objects, instead of trying to implement robustness against clutter directly into the classifier. This could be done by using more of the information about the environment contained in the point cloud to extract cues that make the segmentation-tracking process more aware of the context. Furthermore, when the system detects that an object is partially occluded, it could use more of the past information of the object to improve the classification.

Notes

1. A comprehensive discussion of these processes is outside the scope of this paper. More detailed descriptions are presented in MacLachlan and Mertz (2006) and Navarro-Serment et al. (2008).
2. There is an ongoing debate of whether per-cycle performance should be compared, as opposed to the per-run performance measures. For completeness we show both. There is also a per-fixed-time (typically 1 second) performance measure, but that is actually a measure between those two others. An interesting discussion of related issues within the computer vision community is presented in Dollar et al. (2009).

Acknowledgments

We thank General Dynamics Robotic Systems for their support. We also thank the UPI team for making the Velodyne™ data available to us. This work was conducted through collaborative participation in the Robotics Consortium sponsored by the U.S. Army Research Laboratory under the Collaborative Technology Alliance Program, Coop. Agreement DAAD19-01-209912.

References

- Arras, K. O., Mozos, O. M. and Burgard, W. (2007) Using boosted features for the detection of people in 2d range data. *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, Roma, Italy, 10–14 April 2007, pp. 3402–3407.
- Burges, C. J. C. (1998) A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, Vol. 2. Boston, MA, Kluwer Academic, 1998, pp. 121–167.
- Cui, J., Zha, H., Zhao, H. and Shibasaki, R. (2007) Laser-based detection and tracking of multiple people in crowds. *Computer Vision and Image Understanding*, **106**: 300–312.
- Dollar, P., Wojek, C., Schiele, B. and Perona, P. (2009) Pedestrian detection: a benchmark. *IEEE Conference on Computer Vision and Pattern Recognition CVPR*, 2009, pp. 304–311.
- Fuerstenberg, K., Dietmayer, K. and Willhoeft, V. (2002) Pedestrian recognition in urban traffic using a vehicle based multilayer laserscanner. *Intelligent Vehicle Symposium*, 2002, pp. 31–35.
- Hebert, M. (2000) Active and passive range sensing for robotics. *Proceedings 2000 IEEE International Conference on Robotics and Automation*, Vol. 1, pp. 102–110.
- Howard, A., Matthies, L. H., Huertas, A., Bajracharya, M. and Rankin, A. (2007) Detecting pedestrians with stereo vision: safe operation of autonomous ground vehicles in dynamic environments. *Proceedings of the 13th International Symposium of Robotics Research*, 26–29 November 2007.
- MacLachlan, R. and Mertz, C. (2006) Tracking of moving objects from a moving vehicle using a scanning laser rangefinder. *Proceedings of the IEEE Intelligent Transportation Systems Conference*, 17–20 September 2006.
- Morris, D., Colonna, B. and Haley, P. (2006) Ladar-based mover detection from moving vehicles. *Proceedings of the 25th Army Science Conference*, November 2006.
- Navarro-Serment, L. E., Mertz, C. and Hebert, M. (2006) Predictive mover detection and tracking in cluttered environments. *Proceedings of the 25th Army Science Conference*, 27–30 November 2006.
- Navarro-Serment, L. E., Mertz, C., Vandapel, N. and Hebert, M. (2008) LADAR-based pedestrian detection and tracking. *1st IEEE Workshop on Human Detection from Mobile Platforms*, Pasadena, CA, 20 May 2008.
- Shoemaker, C. M. and Bornstein, J. A. (1998) The Demo III UGV program: a testbed for autonomous navigation research. *Proceedings of the IEEE International Symposium on Intelligent Control*, Gaithersburg, MD, September 1998, pp. 644–651.
- Streller, D., Fuerstenberg, K. and Dietmayer, K. (2002) Vehicle and object models for robust tracking in traffic scenes using

- laser range images. *Proceedings of Intelligent Transportation Systems*, 2002, pp. 118–123.
- Sun, Z., Bebis, G. and Miller, R. (2006) On-road vehicle detection: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**(5): 694–711.
- Thornton, S., Hoffelder, M. and Morris, D. (2008) Multi-sensor detection and tracking of humans for safe operations with unmanned ground vehicles. *1st IEEE Workshop on Human Detection from Mobile Platforms*, Pasadena, CA, 20 May 2008.
- Thornton, S. and Patil, R. (2008) Robust detection of humans using multi-sensor features. *Proceedings of the 26th Army Science Conference*, 1–4 December 2008.