Probabilistic Terrain Analysis
For High-Speed Desert Driving

Sebastian Thrun, Mike Montemerlo, Andrei Aron
Stanford Artificial Intelligence Lab (SAIL)
353 Serra Mall, Stanford, CA 94305
{thrun,mmde,naron}@stanford.edu

Abstract— **The ability to perceive and analyze terrain is a key problem in mobile robot navigation. Terrain perception problems arise in planetary robotics, agriculture, mining, and, of course, self-driving cars. Here, we introduce the PTA (probabilistic terrain analysis) algorithm for terrain classification with a fast-moving robot platform. The PTA algorithm uses probabilistic techniques to integrate range measurements over time, and relies on efficient statistical tests for distinguishing drivable from non-drivable terrain. By using probabilistic techniques, PTA is able to accommodate severe errors in sensing, and identify obstacles with nearly 100% accuracy at speeds of up to 35mph. The PTA algorithm was an essential component in the DARPA Grand Challenge, where it enabled our robot Stanley to traverse the entire course in record time.**

I. INTRODUCTION

The DARPA Grand Challenge required an autonomous robot to traverse unrehearsed desert terrain at speeds in excess of 30mph, and without any human intervention. Stanley, our robotic vehicle described in this article, successfully traversed 131.6 miles of unpaved desert terrain in just below 7 hours [14].

The focus of this article is the central software component that enabled this vehicle to navigate and ultimately win this historical race. Robotic autonomy has been available in airborne and submersed systems for multiple decades; however, until recently no comparable auto-pilots existed for ground vehicles. The reason why ground navigation is harder than aerial or underwater navigation arises from the obstacles on the ground. In the desert, obstacles include rocks, vegetation, berms, ruts, cliffs, overhangs, man-made artifacts like abandoned vehicles, and so on. Perceiving and navigating in the presence of obstacles is a key prerequisite for autonomous ground navigation.

This article focuses on an algorithm for terrain perception that has been the core enabling factor of our autonomous robot shown in Fig. 1. This algorithm, called the *probabilistic terrain analysis (PTA)* algorithm, processes range data acquired by a single-axis laser scanner mounted horizontally on a moving robotic platform. It constructs a 2D environment map suitable for robotic driving. The map is generated online with less than 300ms latency, making it suitable for autonomous ground vehicle navigation.

The key difficulty addressed by the PTA algorithm arises from the noise in pose estimation that naturally occurs in a fast-moving ground rover. Even though our system uses state-of-the-art inertial guidance (multiple GPS and inertial sensors integrated using a nonlinear filter), the residual errors are still large enough to render the point cloud unusable for driving. PTA bases its terrain assessment on a probabilistic model of the uncertainty in the data acquisition process. In doing so, PTA can distinguish between actual obstacles and "phantom" obstacles resulting from the uncertainty in the inertial guidance system. The PTA algorithm also uses a discriminative machine learning algorithm for acquiring the parameters of this probabilistic model. This learning algorithm makes it possible to optimally tune the algorithm using data acquired within a few minutes of human driving.

Empirically, the PTA algorithm has been shown to be accurate. In fact, we claim it was an essential innovation in the design of the winning vehicle of the DARPA Grand Challenge. Stanley traveled at speeds of up to 38mph over extremely rugged, unpaved desert terrain. In this article, we provide empirical evidence that the core element of PTA, its probabilistic noise analysis and associated learning procedure, reduce the error rate significantly when compared to a non-probabilistic algorithm. In one dataset, it reduces false-positive error rate from 12.6% to 0.002% without significantly affecting the false-negative rate. Such numbers mattered greatly for the DARPA Grand Challenge: false-positives correspond to "phantom obstacles" that (as we show) easily mislead the robot into hazardous terrain.

II. RELATED WORK

There exists a huge body of related work on terrain perception for autonomous driving. Early work in the field includes that by Dickmanns and Pomerleau [10], [2]. This seminal research focused on vision-based onroad driving, and led to a huge body of follow-up research. A more recent overview of research in this area is given in [4].

The idea of using lasers for outdoor terrain perception is not new; in fact, lasers have played essential roles in a series of Government-funded off-road driving projects [13] and in both DARPA Grand Challenges [15]. Grid representation like the one used in this paper have become popular in a number of systems using lidar or stereo for depth perception [5], [9], [16]; in fact, the work in [16] attaches confidence factors to grid cells that reflect the information content in the sensor measurements.

Figure 1. At approximately 1:40pm on Oct 8, 2005, a mobile robot wins the DARPA Grand Challenge. The algorithm discussed in this paper played an essential role in the robot's ability to navigate the 2005 Grand Challenge course.

Figure 2. The vehicle uses a single line scanner to acquire surface data from the terrain to make driving decisions.

The topic of pose error in 3D point cloud acquisition has received considerable attention in the literature. For example, in [7], Levoy et al. describe a method for mapping historical objects using a robot arm and a range scanner. The "classical" method for aligning such point clouds is the *iterative closest point* algorithm (ICP) [1]; see [11] for a real-time implementation, and [8] for a variant known as *scan matching*. However, the fundamental assumption behind these methods is that the robot scans the same object more than once. Multiple scans are aligned by detecting the area of overlap, and using the mismatch for scan alignment.

In our driving domain, such a method is inapplicable. This is because most scans cover new territory, hence correspond to no previous scan. The general lack of overlap "breaks" the correspondence step in ICP or scan matching, rendering these method inapplicable for pose error compensation. In fact, we remark that the authors indeed implemented ICP and found the performance to be intolerably poor.

From a probabilistic viewpoint, ICP uses a probabilistic error model (a Gaussian) to recover an accurate world model from inaccurate data; the method here uses a probabilistic error model to define suitable tests over an inaccurate world model. This subtle difference is important, as the available data is insufficient to recover an accurate 3D terrain model.

III. POINT CLOUD ACQUISITION

The PTA algorithm operates on a 3-D point cloud acquired by the robot while in motion. Because the specifics of the data acquisition matter, we briefly describe the robot and the various sensors involved in the acquisition process. All methods described in this section are commonly used in the field of vehicle guidance [3], [12].

The robotic vehicle is shown in Fig. 1. To determine its location relative to an external coordinate frame, the robot is equipped with an *inertial guidance system*. The task of this

system is to determine the *pose* of the robot, by which we mean the georeferenced 3-D coordinates (denoted $\mathbf{x}_k$) and the 3-D orientation of the vehicle in Euler angles (pitch-roll-yaw, denoted $\mathbf{\Psi}_k$):

$$\mathbf{x}_k = \begin{pmatrix} x_k \\ y_k \\ z_k \end{pmatrix} \quad \mathbf{\Psi}_k = \begin{pmatrix} \phi_k \\ \theta_k \\ \psi_k \end{pmatrix} \quad (1)$$

Here k is the time index. As usual, the orientation vector $\mathbf{\Psi}_k$ induces a rotation matrix, which we will denote by R_k.

The inertial guidance system computes the pose estimate based on measurements from two differential GPS systems and a six-degree-of-freedom inertial measurement unit (IMU). The GPS systems measure the absolute coordinates and velocities of the vehicle and its pitch and yaw angle (but not roll). The IMU measures angular velocities and linear accelerations. As is common in the field of vehicle guidance, the data is integrated using a Kalman filter. Our implementation relies on an *unscented Kalman filter (UKF)* [6] chosen over classical methods for its improved accuracy. The spatial accuracy in $\mathbf{x}_k$ of the guidance system is dependent on the satellite reception, and varies from 20cm to 2m. The maximum error in the vehicle's orientation $\mathbf{\Psi}_k$ is on the order of one degree. Such values are within the norm for moderate-cost guidance systems such as the one deployed.

To acquire the 3-D point cloud, the robot is equipped with single-scan lasers (multiple lasers are used for redundancy but processed separately). Each laser is mounted horizontally on the robot's roof, slightly tilted downward to scan the ground ahead. Fig. 2 illustrates the scanning process. Each laser scan generates a vector of 180 range measurements spaced 0.5 degrees apart. If we denote the angle of the i-th element in this vector by α_k^i, and the measured range value by r_k^i, the corresponding measurement is projected into the external GPS-reference frame via the straightforward projective equation:

$$\begin{pmatrix} X_k^i \\ Y_k^i \\ Z_k^i \end{pmatrix} = r_k^i \, R_k \, R_{\mathrm{sens}} \begin{pmatrix} \cos \alpha_k^i \\ \sin \alpha_k^i \\ 0 \end{pmatrix}$$
$$+ \mathbf{x}_k + R_k \, \mathbf{x}_{\mathrm{sens}} \quad (2)$$

Here R_k is the rotation matrix that corresponds to the vehicle orientation $\mathbf{\Psi}_k$; $\mathbf{x}_{\mathrm{sens}}$ and R_{sens} are the displacement and
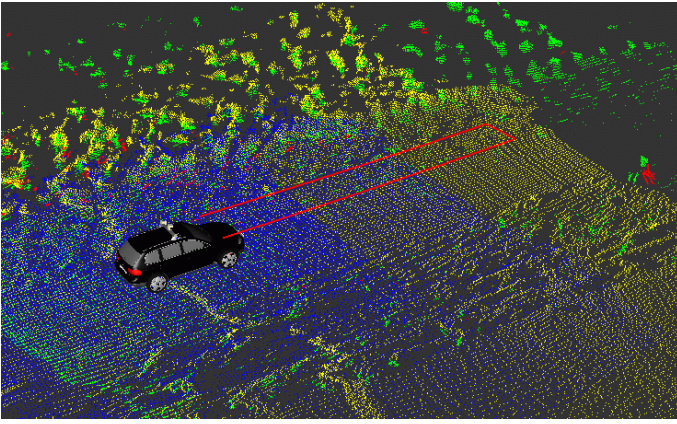
Figure 3. 3D point cloud acquired by the moving vehicle. The scan data is integrated into point clouds using an inertially guided system for determining the location and orientation of the sensor.
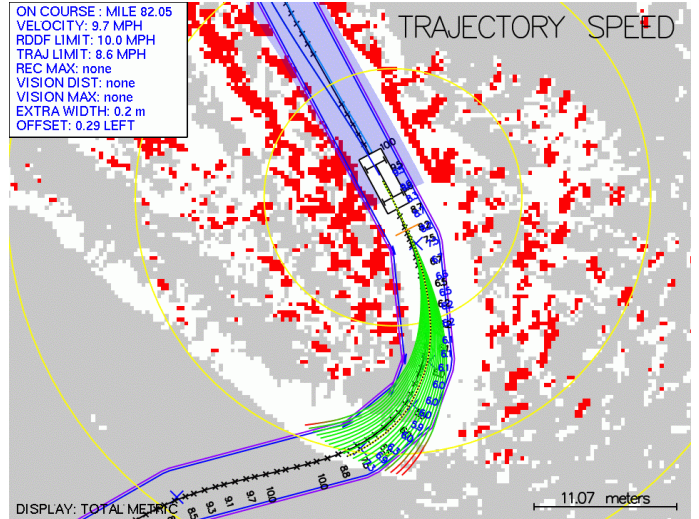

Figure 4. Snapshot of the path planner as it processes the drivability map. This snapshot is taken from the most difficult part of the 2005 DARPA Grand Challenge, a mountainous area called *Beer Bottle Pass*.

pointing angle of the scanner relative to the vehicle's local coordinate frame. Fig. 3 illustrates such a point cloud.

Obviously, the 3-D point cloud is only of intermediary interest as we seek to enable the robot to make the right driving decisions. To this end, we define a *terrain labeling function* that assigns to each 2-D location X_q, Y_q one of three values:

- **Obstacle** if we can find two points, $(X_k^i \quad Y_k^i \quad Z_k^i)^T$ and $(X_m^j \quad Y_m^j \quad Z_m^j)^T$ whose x-y distance to the query point X_q, Y_q is smaller or equal to ε, and for which $|Z_k^i - Z_m^j|$ exceeds a critical vertical distance δ. In our implementation, $\varepsilon = 30$cm and δ is between 15cm and 20cm (see section on parameter tuning below).
- **Drivable** if not an obstacle, but we can find at least one point $(X_k^i \quad Y_k^i \quad Z_k^i)^T$ within an ε-range of the query coordinates X_q, Y_q.
- **Unknown** if no point $(X_k^i \quad Y_k^i \quad Z_k^i)^T$ exists within an ε-range of the query coordinates X_q, Y_q.

The search for nearby points is conveniently organized in a 2-D grid, and the same grid is used as the final 2-D drivability map that is provided to the vehicle's navigation engine. Fig. 4 shows an example map. Here red=obstacle, white=drivable, and grey=unknown.

IV. PROBABILISTIC TERRAIN ANALYSIS (PTA)

A. Temporal Noise Model

Unfortunately, the algorithm just described yields results inappropriate for robot navigation. Fig. 5 shows such an instance, in which a small error in the vehicle's role/pitch estimation leads to a massive terrain classification error—forcing the vehicle off the road. Such situations occur even for roll/pitch errors below 0.5 degrees. The sensitivity to roll and pitch for the vehicle is the result of the fact that the scanner is pointed forward, detecting objects at up to 30 meters range. Unfortunately, such a range is necessary for safe vehicle operation at Grand Challenge driving speeds. In our reference dataset of labeled terrain, we found that 12.6% of known drivable area is classified as obstacle, for a height threshold parameter $\delta = 15$cm.

To accommodate such errors, one might be tempted to adjust the 3-D point cloud through an ICP-type algorithm. However, ICP is prone to fail for the reasons discussed in the related work section above.

In contract, the PTA algorithm does not attempt to reconstruct a 3-D model. Instead, it runs statistical tests over the data, to probe for obstacles. To do so, PTA models the error in the data acquisition process using a first order Markov model, and uses this model when determining whether a location X_q, Y_q is drivable.

The first order Markov model is given by a stochastic process with noise variables β_k and γ_k:

$$\begin{pmatrix} \mathbf{x}_k^* \\ \mathbf{\Psi}_k^* \end{pmatrix} = \begin{pmatrix} \mathbf{x}_k \\ \mathbf{\Psi}_k \end{pmatrix} + \beta_k + \gamma_k \tag{3}$$

Here the asterisks denote the *estimated* vehicle state, which is of course corrupted by noise. The variable β models the noise over time, and the variable γ the momentary noise in the pose estimates. Assuming Gaussian noise, we have

$$\beta_k \sim \mathcal{N}(\beta_{k-1}, B) \tag{4}$$
$$\gamma_k \sim \mathcal{N}(0, C) \tag{5}$$

where B and C are the time-invariant noise covariances. As a result of Eq. 4, the error increases over time[1]. This models the fact that the state estimate is generally noisy, but noise evolves slowly over time.
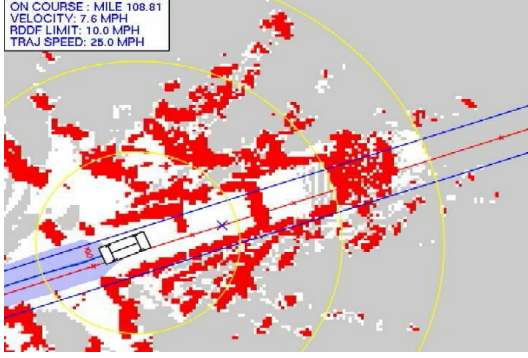
We further define B and C to be of the following form:

$$B = \text{diag}(\sigma_{xyz}^2, \sigma_{xyz}^2, \sigma_{xyz}^2, \sigma_{\phi\theta\psi}^2, \sigma_{\phi\theta\psi}^2, \sigma_{\phi\theta\psi}^2) \tag{6}$$
$$C = \text{diag}(\tau_{xyz}^2, \tau_{xyz}^2, \tau_{xyz}^2, \tau_{\phi\theta\psi}^2, \tau_{\phi\theta\psi}^2, \tau_{\phi\theta\psi}^2) \tag{7}$$

---

[1] We shall not be concerned with the observation that the absolute error may diverge under this model, as we only use it to determine the relative error when comparing to measurements acquired in short temporal succession.

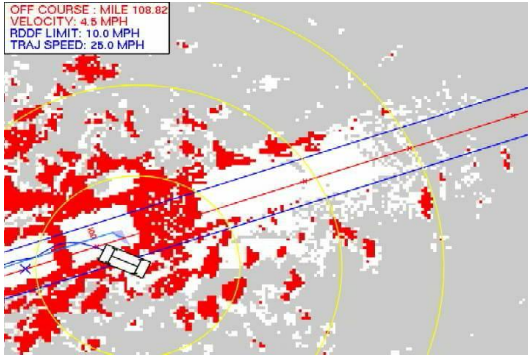**(a) Mapper failure**

(b) Fatal vehicle reaction

Figure 5. Small errors in pose estimation (smaller than 0.5 degrees) induce massive terrain classification errors, which frequently force the robot off the road.

for the noise covariances σ^2_{xyz}, $\sigma^2_{\phi\theta\psi}$, τ^2_{xyz}, and $\tau^2_{\phi\theta\psi}$. Those shall be defined later, when we focus on our attention on a learning method for parameter tuning.

B. The Probabilistic Obstacle Test

The first order Markov chain enables us to reformulate the obstacle test as a *statistical* test. While in its general form, this test is difficult to compute, we shall soon see that for the specific laser configuration there is a simple approximation that works well in practice.

Specifically, given two points $(X^i_k \quad Y^i_k \quad Z^i_k)^T$ and $(X^j_m \quad Y^j_m \quad Z^j_m)^T$ (without loss of generality assume $m > k$), the height difference is distributed according to

$$
Z^{i*}_k - Z^{j*}_m \tag{8}
$$
$$
\sim \quad \mathcal{N}(\underbrace{Z^i_k - Z^j_m}_{\text{mean}}, \underbrace{\nabla^{iT}_k[Z] \, (2C + |m-k| \, B) \, \nabla^j_m[Z]}_{\text{covariance}})
$$

Here we approximated the non-linear projection defined in Eq. 2 with a first order Taylor expansion. The term $\nabla^i_k[Z]$ is the Jacobian of the value of Z^i_k with respect to the state vector.

In general, these Jacobians are difficult to calculate. However, for a forward-pointed laser, a vehicle that is approximately level, a laser for which $\alpha^i_k \approx \alpha^j_m$, and if we ignore the laser offset $\mathbf{x}_{\mathrm{sens}}$, this Jacobian resolves to a distribution that

is relatively easily computed

$$
\nabla^i_k[Z] \quad \approx \quad \begin{pmatrix} 0 \\ 0 \\ 1 \\ r^i_k \cos\alpha^i_k \\ r^i_k \sin\alpha^i_k \\ 0 \end{pmatrix} \tag{9}
$$

This expression depends linearly on the error in the estimate of z_k, and also on the error in pitch and roll, which are both amplified through the actual measured range r^i_k.

Plugging this back into 8, and observing that our model assumes equal covariance for roll and pitch error, we obtain the relatively simple distribution

$$
Z^{i*}_k - Z^{j*}_m \sim \mathcal{N}(\underbrace{Z^i_k - Z^j_m,}_{\text{mean}} \tag{10}
$$
$$
\underbrace{|m-k|(\sigma^2_{\mathrm{xyz}} + r^i_k \sigma^2_{\phi\theta\psi}) + 2\tau^2_{\mathrm{xyz}} + r^i_k \tau^2_{\phi\theta\psi} + r^j_m \tau^2_{\phi\theta\psi})}_{\text{covariance}}
$$

This expression grows linearly with the time difference $|m-k|$. The key unknowns are the noise covariances σ^2_{xyz}, $\sigma^2_{\phi\theta\psi}$, τ^2_{xyz}, and $\tau^2_{\phi\theta\psi}$, which we generally do not know.

We will return to the problem of finding those noise covariance in the next section, when we address the issue of learning the Markov model. Assuming knowledge of those parameters, we can not use a statistical test for determining if the Z-value of a pair of ε-nearby points is indeed larger than the height threshold δ. The test uses the Z-value of the normal distribution just defined, to calculate

$$
p(|Z^{i*}_k - Z^{j*}_m| > \delta) \quad > \quad \pi \tag{11}
$$

for the error probability threshold π.

For $\pi = 0.05$, this resolves to

$$
\frac{|Z^{i*}_k - Z^{j*}_m| - \delta}{\sqrt{|m-k|(\sigma^2_{\mathrm{xyz}} + r^i_k \sigma^2_{\phi\theta\psi}) + 2\tau^2_{\mathrm{xyz}} + r^i_k \tau^2_{\phi\theta\psi} + r^j_m \tau^2_{\phi\theta\psi}}} > 1.64
$$

or

$$
(|Z^{i*}_k - Z^{j*}_m| - \delta)^2
$$
$$
> \quad (1.64)^2 |m-k|(\sigma^2_{\mathrm{xyz}} + r^i_k \sigma^2_{\phi\theta\psi}) + 2 \cdot (1.64)^2 \tau^2_{\mathrm{xyz}}
$$
$$
+ (1.64)^2 r^i_k \tau^2_{\phi\theta\psi} + (1.64)^2 r^j_m \tau^2_{\phi\theta\psi} \tag{12}
$$

According to our Markov model, this test is an obstacle acceptance test for the presence of an obstacle at the 95% confidence level.

We note that the quantity on the right-hand side of the test increases monotonically with the time difference $|m-k|$. This observation is important: the acceptance probability for an obstacle *decreases* with time because of the noise. That is, two nearby points in the 3-D point cloud are more likely to be witness of an obstacle if they were recorded in short temporal succession.

Put differently, the minimum height difference between two measurements required to mark those measurements as witnesses of an obstacle, is a function of the time difference

at which those measurements were acquired. The more time went by, the larger this minimum height difference.

C. Efficient Implementation of the PTA Test

A key provision in the PTA analysis pertains to the efficient search of possible pairs of points that are a witness of an obstacle. As in the non-probabilistic algorithm above, PTA caches all information over a 2-D grid of the environment. The resolution of the grid is $\varepsilon/2$. Thus, when analyzing a new range measurement, PTA only queries the corresponding 2-D grid cell and its immediate neighbors.

Even though many points in the 3-D point cloud might fall within a grid cell, it suffices to store two Z values per cell. One of these values defines the tightest upper value for future tests, and one defines the tightest lower value. Along with these two values, we have to store the times at which the corresponding measurements were recorded, so that the test can correctly fold in the increase of uncertainty over time.

The fact that those two measurements suffice is a direct result of the monotonic growth of the uncertainty over time. Suppose we observe a new measurement for a cell which was previously observed. The one or more of three cases will be true:

1) The new measurement might be a witness of an obstacle, according to the probabilistic test in PTA. In this case, PTA simply marks the cell as obstacle and no further testing takes place.

2) The new measurement does not trigger as a witness of an obstacle, but in future tests it establishes a tighter lower bound on the minimum Z-value than the previously stored measurement. In this case, we simply replace the previous measurement with this new one. The rationale behind this is simple: If the new measurement is more restrictive than the previous one, there will not be a situation where a test against this point would fail while a test against the older one would succeed. Hence, the old point can safely be discarded.

3) The third car is equivalent to the second, but with a refinement of the upper value.

Notice that a new measurement may refine simultaneously the lower and the upper bounds.

The fact that we only have to memorize two measurements per grid call renders PTA highly efficient in space and time—which is important for real-time robotic driving.

D. Learning the Parameters

The probabilistic Markov model possesses a number of unknown parameters. The final component of the PTA is concerned with fitting the various parameters:

- The height threshold δ.
- The statistical acceptance probability threshold π.
- The Markov chain error parameters, which are the noise covariances σ^2_{xyz}, $\sigma^2_{\phi\theta\psi}$, τ^2_{xyz}, and $\tau^2_{\phi\theta\psi}$.

Each of Stanley's five lasers possesses its own parameter set.

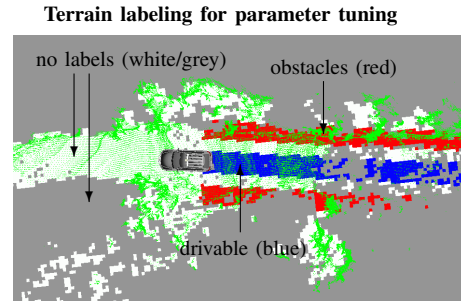**Terrain labeling for parameter tuning**

Figure 6. Terrain labeling for parameter tuning: The area traversed by the vehicle is labeled as "drivable" (blue) and two stripes at a fixed distance to the left and the right are labeled as "obstacles" (red). While these labels are only approximate, they are extremely easy to obtain and significantly improve the accuracy of the resulting map when used for parameter tuning.

Instead of determining such parameters by hand, which is difficult[2], PTA uses a discriminative learning algorithm for (locally) optimizing these parameters. More specifically, the parameters are tuned in a way that maximizes the discriminative accuracy of the resulting terrain analysis on labeled training data.

A tedious way of labeling data would be to label each grid cell manually as to whether it corresponds to an obstacle or not. Such a method would require extreme effort, since a person would have to manually inspect each area near the robot.
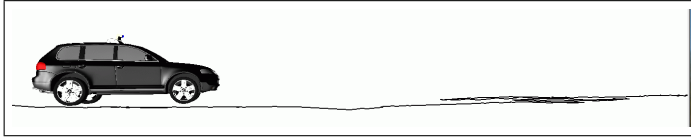
To get around this issue, we developed an approximate labeling technique, in which a human diver simply labels terrain by driving. Fig 6 illustrates the idea: Terrain that a person chooses to drive over is assumed to be obstacle-free, hence labeled as "0". This area corresponds to the blue stripe in Figure 6. A stripe left and right of this corridor is assumed to be all obstacles, as indicated by the red stripes in Figure 6. Clearly, not all of those cells are occupied; however, the learning technique is still pushed to label as many of those as possible as occupied.

The learning algorithm is now implemented through coordinate ascent. In the outer loop, the algorithm performs coordinate ascent relative to a data-driven scoring function. Given an initial guess, the coordinate ascent algorithm modifies each parameter one-after-another by a fixed amount. It then determines if the new value constitutes an improvement over the previous value when evaluated over a logged data set, and retains it accordingly. If for a given interval size no improvement can be found, the search interval is cut in half and the search is continued, until the search interval becomes smaller than a pre-set minimum search interval (at which point the tuning is terminated).

Empirically, we find that the learning algorithm converges quickly to a robust solution when trained with about 2 minutes of driving data. In developing the DARPA Grand Challenge entry, we often included in the training set data from extreme

---

[2]GPS manufacturers do not report temporal drift; and drift is often escalated by the fact that high-accuracy receivers switch between multiple internal filters turned for different signal strengths.

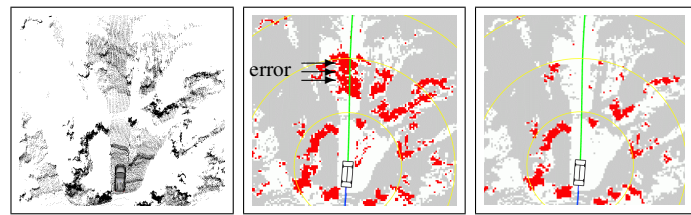**(b) 3-D point cloud**   **(c) non-probabilistic method**   **(d) PTA result**

Figure 7. Comparison of non-probabilistic algorithm and PTA: (a) shows a scan over time, (b) the 3-D point cloud, (c) the erroneous map and (d) the result of PTA.

(a) Robot and laser scan plotted over time

(b) 3-D point cloud   **(c) non-probabilistic method**   **(d) PTA result**

Figure 8. A second comparison. See text.

terrain, with large errors in the vehicle state estimation (and hence errors in the map). In this way, the data focused on the hardest instances available.

V. RESULTS

The probabilistic analysis paired with the discriminative algorithm for parameter tuning has a significant effect on the accuracy of the method. Using an independent data set acquired in the Sonoran Desert, we found that the false positive rate (the area labeled as drivable in Fig. 6) drops from 12.6% to 0.002% when measured over 50,000 grid cells when measured on an independent test set. At the same time the rate at which the area off the road is labeled as obstacle remains approximately constant (from 22.6% to 22.0%). This rate is *not* 100% simply because most of the terrain there is still flat and drivable. Our approach for data acquisition *mislabels* the flat terrain as non-drivable. Such mislabeling however, do not impede with the parameter tuning algorithm, and hence is preferable to labeling pixels manually (which would be extremely tedious).

Fig 7 shows an example: a snapshot of the vehicle from the side illustrates that part of the surface is scanned multiple times due to a change of pitch. As a result, the non-PTA method hallucinates a large occupied are in the center of the road, as shown in Panel c of Fig 7. PTA overcomes this error and generates a map that is good enough for driving. A second example is shown in Fig 8.

Based on post-screening of the sensor data logged during the race, we find that the PTA algorithm was essential for the robot's success. Figure 9 displays the situation along the most difficult part of he course, known as "Beer Bottle Pass." The two blue contours in the bottom image mark the GPS corridor provided by DARPA, which aligns poorly with the map data, indicating an approximate localization error of 2 meters. This analysis suggests that a robot that followed the GPS via points blindly would likely have failed to traverse this narrow mountain pass.

We also used the algorithm in the National Qualification Event, which was DARPA's selection event for the race
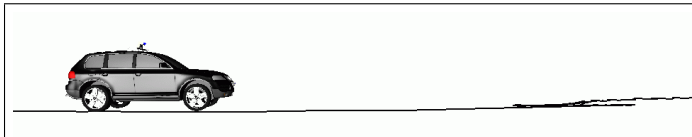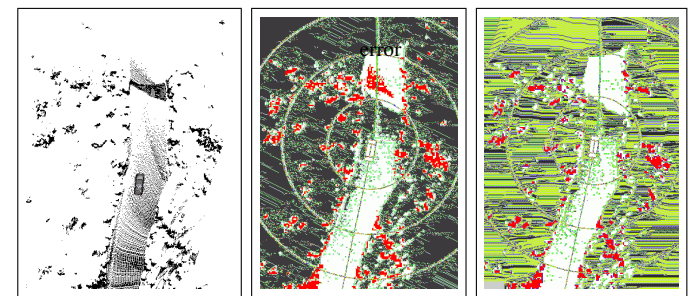
contestants. In this event, our robot emerged as the only robot that never collided with an obstacle or missed a gate marked by traffic cones. All other robots were unable to escape collisions. This performance was echoed in the actual race, where several other vehicles collided with obstacles along the way, whereas our robot emerged free of any scratches or dents. While collision-free motion requires more than just accurate terrain analysis, the terrain analysis is a clear prerequisite of collision-free motion.

The suitability of the PTA error model can also be seen in surface data. Figure 10 shows a scatter plot generated from measurements taken on a flat road, free of obstacles. The horizontal axis is the time difference between two measurements within a grid cell, and the vertical axis is $|Z_k^{i*} - Z_m^{j*}|$, which on flat terrain is the estimation error (ideally, the value should always be zero). As this plot indicates, the vertical estimation error is indeed a function of time: the more time elapses, to larger the relative Z-error. This dependence is captured by our Markov model, which also models an increase of uncertainty over time.

(a) Beer Bottle Pass   **(b) Map and GPS corridor**
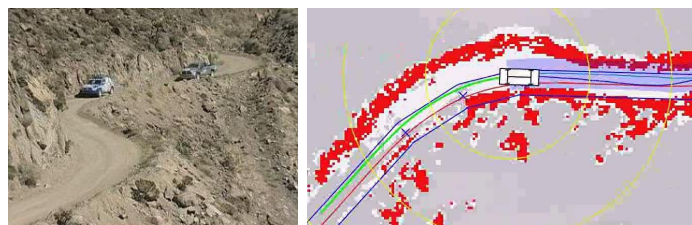

Figure 9. Snapshot of the map acquired by the robot on the "Beer Bottle Pass," the most difficult passage of the DARPA Grand Challenge. The two blue contours mark the GPS corridor provided by DARPA, which aligns poorly with the map data. This analysis suggests that a robot that followed the GPS via points blindly would likely have failed to traverse this narrow mountain pass.
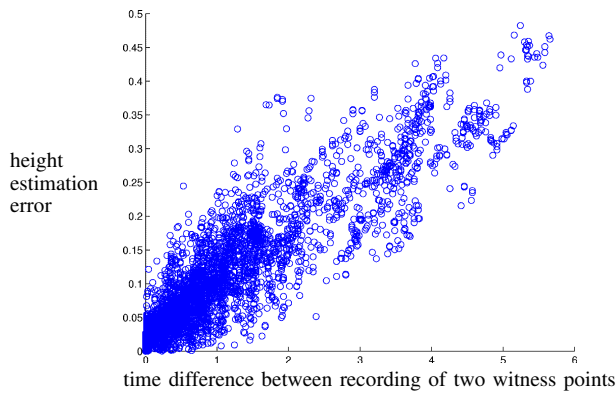
height
estimation
error

time difference between recording of two witness points

Figure 10. For terrain known to be flat, this diagram shows the estimated vertical height difference.

VI. CONCLUSION

We have presented a new algorithm, called PTA (short for *probabilistic terrain analysis*). This algorithm enables offroad vehicles, equipped with a single-scan laser, to analyze data from this laser so as to discriminate between drivable and non-drivable terrain.

Th heart of this algorithm is a statistical error model of the pose estimation error, and a statistical test for the presence of obstacles based on this error model. The PTA algorithm also features a learning component in which labeled training data is used to tune the parameters of the probabilistic error model. In empirical testing, the approach was found to provide excellent results. In fact, we believe PTA was essential in our successful bid for the DARPA Grand Challenge.

ACKNOWLEDGMENT

REFERENCES

[1] P. Besl and N. McKay. A method for registration of 3d shapes. *Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[2] E.D. Dickmanns, R. Behringer, D. Dickmanns, T. Hildebrandt, M. Maurer, J. Schiehlen, and F. Thomanek. The seeing passenger car VaMoRs-P. In *Proceedings of the International Symposium on Intelligent Vehicles*, Paris, France, 1994.

[3] M.S. Grewal, L.R. Weill, and A.P. Andrews. *Global Positioning Systems, Inertial Navigation, and Integration*. John Wiley and Sons, Inc., New York, NY, 2001.

[4] M. Hebert, C. Thorpe, and A. Stentz. *Intelligent Unmanned Ground Vehicles: Autonomous Navigation Research at Carnegie Mellon University*. Kluwer Academic Publishers, 1997.

[5] T.H. Hong, M. Abrams, T. Chang, and M. Shneier. An intelligent world model for autonomous off-road driving. Submitted to the Computer Vision and Image Understanding Journal, 2001.

[6] S. Julier and J. Uhlmann. A new extension of the Kalman filter to nonlinear systems. In *International Symposium on Aerospace/Defense Sensing, Simulate and Controls*, Orlando, FL, 1997.

[7] M. Levoy, S. Rusinkiewicz, M. Ginzton, J. Ginsberg, K. Pulli, D. Koller, S. Anderson, J. Shade, B. Curless, L. Pereira, J. Davis, and D. Fulk. The digital michelangelo project: 3D scanning of large statues. In *Computer Graphics (SIGGRAPH 2000 Conference Proceedings)*. ACM SIGGRAPH, 2000.

[8] F. Lu and E. Milios. Robot pose estimation in unknown environments by matching 2d range scans. *Journal of Intelligent and Robotic Systems*, 18:249–275, 1998.

[9] K. Murply, M. Abrams, S. Balakirsky, T. Chang, T. Hong, A. Lacaze, and S. Legowik. Intelligent control for off-road driving. In *Proceedings of the First International NAISO Controess on Autonomous Intelligent Systems*, Geelong, Australia, 2002. Deakin University.

[10] D. A. Pomerleau. Knowledge-based training of artificial neural networks for autonomous robot driving. In J. H. Connell and S. Mahadevan, editors, *Robot Learning*, pages 19–43. Kluwer Academic Publishers, 1993.

[11] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Proc. Third International Conference on 3D Digital Imaging and Modeling (3DIM)*, Quebec City, Canada, 2001. IEEEComputer Society.

[12] N.A. Shneydor. *Missile Guidance and Pursuit: Kinematics, Dynamics and Control*. Horwood Publishing, 1998.

[13] C. Shoemaker, J. Bornstein, S. Myers, and B. Brendle. Demo III: Department of defense testbed for unmanned ground mobility. In *Proceedings of SPIE, Vol. 3693, AeroSense Session on Unmanned Ground Vehicle Technology*, Orlando, Florida, 1999.

[14] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney. Stanley, the robot that won the DARPA Grand Challenge. *Journal of Field Robotics*, Forthcoming.

[15] C. Urmson, J. Anhalt, M. Clark, T. Galatali, J.P. Gonzalez, J. Gowdy, A. Gutierrez, S. Harbaugh, M. Johnson-Roberson, H. Kato, P.L. Koon, K. Peterson, B.K. Smith, S. Spiker, E. Tryzelaar, and W.L. Whittaker. High speed navigation of unrehearsed terrain: Red team technology for grand challenge 2004. Technical Report CMU-RI-TR-04-37, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 2004.

[16] J. Weber and L. Matthies. A obstacle representation for off-road autonomous driving. In *Proceedings of the Intelligent Vehicles Symposium*, Tokyo, 1996.