Learning in Reality: A case study of Stanley, the robot that Won the DARPA Challenge

Christian Glaser
Intelligent Autonomous Systems
Technische Universität
Darmstadt, DA 64283
c_glaser@rbg.informatik.tu-darmstadt.de

supervisor: Philipp Hennig
Department of Empirical Inference
Max Planck Institute for Intelligent Systems
Spemannstraße, 72076 Tübingen
phennig@tuebingen.mpg.de

Abstract

The Grand Challenge by the Defense Advances Research Projects Agency (DARPA) got a great impact on Machine Learning and Computer Vision research. Stanley was the first robot which firstly was able to drive autonomously a 175 mile course in desert terrain which was a great success in autonomous driving which may be leading to autonomous cares in urban environment in the future. Stanley faced a bunch of problems which needed to be solved to drive autonomously through the course. Some of those Problems are the terrain which has obstacles which need to be avoided and another is the driving speed which has to be maintained to win the challenge. Stanley uses therefore GPS, laser range scanners and a color camera to classify the road into drivable and non-drivable area. This classification is used to control speed and steering of the robot.

1 Introduction

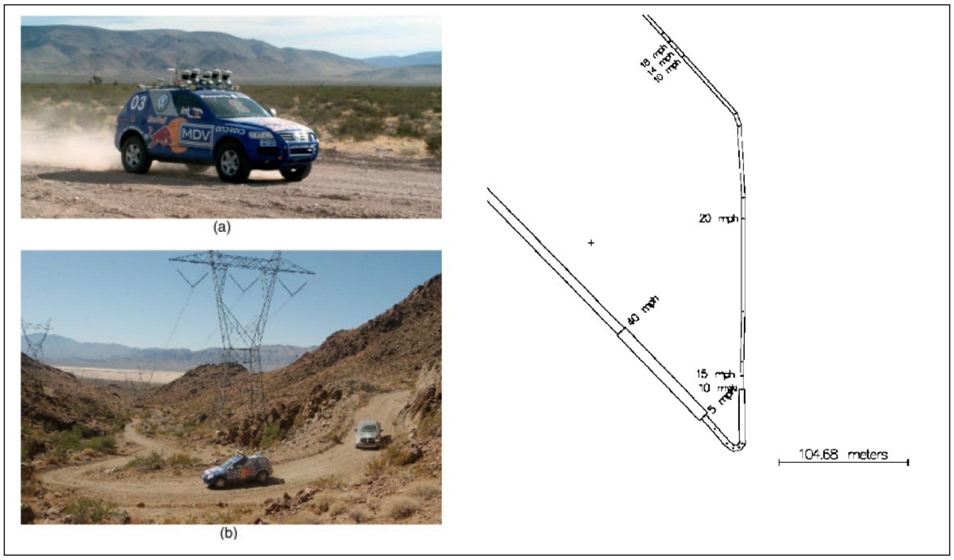

Figure 1: Stanley and a section of the GPS Waypoints.

The Grand Challenge was launched by the Defense Advanced Research Projects Agency (DARPA) and was a race of autonomous driving cars. The first DARPA Grand challenge was held in 2004 and no team was able to drive more than 5% of the course which was a 142-mile long in the Mojave desert. The winning team should win 1M $. One year later the DARPA Grand Challenge was held again with an raised price of 2M $ where the robot Stanley was able to drive the course in under 7 hours winning the competition.

The rules where simple: All teams use the same car, so the challenge is really one of the algorithms and not in building cars that can drive more easily in desert terrain. The course was given by DARPA in terms of 2935 GPS waypoints. They also include speed limits and road corridor widths. Meaning no global path planning was needed. The challenge was purely to stay on the road and avoid obstacles. Also if one vehicle is faster than another the slower car would be stopped by DARPA officials so it could be treated as normal obstacle, no dynamic processing was needed.

Thrun et. al. explained in their paper [5] in detail how Stanley is build and how he is internally working. I want to give an wrap up of their work by looking on it from an Learning and Vision point of view. This means that here the Problems which they had to face when driving in a desert terrain and how they solved them through vision and learning techniques. All the Details on how Stanley is build up internally and how he is doing its path planning and how exactly the learning results are used to change driving direction and speed are mostly skipped.

The rest of this paper continues with a short wrap up of Stanleys components which are used in the learning algorithms. Then the problems which the team around Stanley encountered in desert terrain. The main part is then the important algorithms that do pose estimation and terrain labeling (both using laser scanning and also using computer vision). Finally I conclude the results in the last chapter.

2 Stanley

All the tech Stanley needs to solve the Problems when driving in desert Terrain is mounted on the roof of the car. The important components here are:

- 5 laser range scanners with different tilt angles to view the terrain at different distances, Each of the laser scanners generates a vector of 181 measurements which are spaced 0.5° apart. These measurements are put together in a 3d point cloud (including measurements at

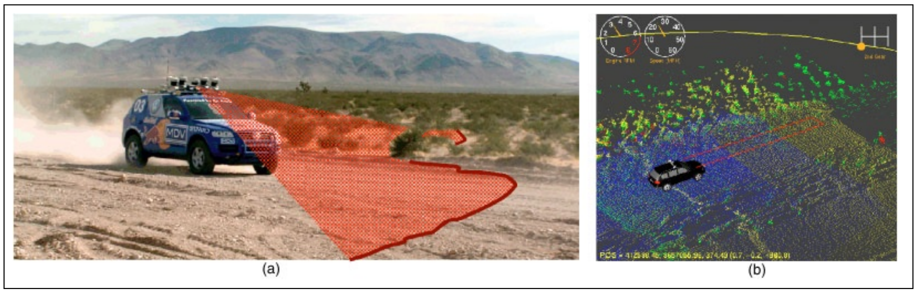

Figure 2: Stanleys Laser Range Scanner

different time steps) The Laser Scanners should be used to detect obstacles up to 25 meters in front of the car.

- gps receiver which is mounted on top of the roof rack where the signal is at least obstructed. GPS is used to get the position of the car and such determining the direction to the next waypoint. Also what is import for obstacle detection it is one factor in determining the vehicle pose.
- a trunk-mounted inertial measurement unit (IMU) which also is used together with the GPS-receiver to determine the vehicle pose.
- color camera which should be used to detect obstacles ranged further away than the laser scanners range can sense.

3 Challenges and their Solution through Vision and Learning

Autonomous driving in desert terrain yields several problems. Problems they encountered are the following:

- pose estimation:
  - the ground is slippery and vehicle movement often is not predictable with a simple model in which the vehicle only moves in driving direction, because of sliding and skippering of the wheels on desert underground
  - gps reception is not the best and there can be long times of gps outtakes in which the car must keep a valid estimation of its position and pose
- terrain labeling
  - there will be obstacles on the road and also conditions of the ground which are not drivable with a certain amount of speed, without damaging the robots equipment
  - small errors in pose estimation can lead to great errors on terrain labeling results with laser scanners. Robust algorithms needed to be found.
  - the achieve high driving speed which is needed to win the challenge laser scanning is not enough because there is also prediction needed to make, which go further than the range of the scanners.
  - using a camera to label terrain as drivable and non-drivable is not trivial

3.1 Pose Estimation

Stanley needs to have a consistent state vector all the time which includes 16 variables like in figure 3 shown. This is needed to keep Stanley on course and also for further calculations in terrain labeling. We will see, that small errors in pose will have a large impact on the results in terrain labeling, so this must be as good as possible.

Like already said the terrain can let the car slide and slip and so it can happen that the car could move in any direction. For this reason as long as gps is available the model used is the one of a moving mass that can be moved in any direction. GPS will give enough data so that pose estimation stays

No. of values	State variable
3	Position (longitude, latitude, and altitude)
3	Velocity
3	Orientation (Euler angles: roll, pitch, and yaw)
3	Accelerometer biases
3	Gyro biases

Figure 3: State Vetor for Pose Estimation

accurate with this model. But unfortunately this is not the case in gps outtakes. Therefore it doesnt represent driving of a car clear enough. So the model in GPS outtakes will switch to a more precise model in which the car can only move in the direction in which it is pointing and including wheel motion into the model. The state vector which gives the pose of the car is then robustly estimated through Kalman filtering.

In Kalman filtering [6] information on the previous states is used to predict the next state. Therefore the model in which a transition to one state to another is used to give a clue to how a state change can look like ans such eliminate outliers on gps and accelerometer data.

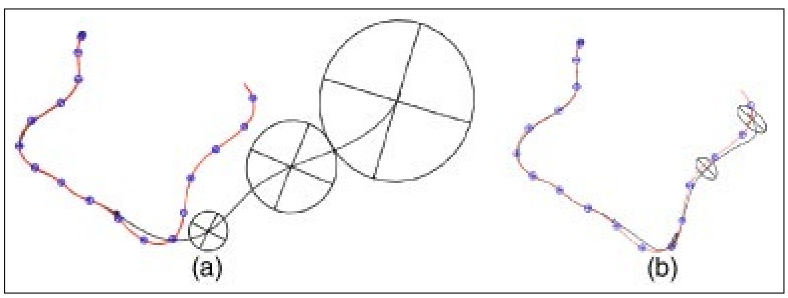

Figure 4: Uncertainty of the Pose estimation. With moving mass model a) and more restricted model including wheel motion b)

To make this even more accurate Stanley uses an unscented Kalman filter [3]. In a normal Kalman filter the next state is linearly predicted using previous ones. But the transition function may not be linearly and so depending on when the prediction is made this could be inaccurate. In an unscented Kalman filter the transition function is sampled to pick a set of sampling points. Then the mean of linear predictions on this sampling points is used to get the next state.

Figure 4 shows the uncertainty of Kalman filtering without a) and with the more restricted model b). The improvement can clearly be seen seen and Stanley could drive over one km and only accumulate an error of 1.7m.

3.2 Terrain Labeling

The key to avoiding obstacles in Stanleys system lies in perceiving and labeling the area in front as drivable or not drivable. To achieve this laser scanners and also an vision extension using a digital camera is used to increase the range in which Stanley can detect obstacles and thus drive faster. Like already states this includes many challenges which needs to be solved and good algorithms need to be found.

3.2.1 Laser Scanning

The first approach to determine if an the road in front of the vehicle is blocked by an obstacle is using the point cloud which results from laser scanning. The space in front of the vehicle is divided

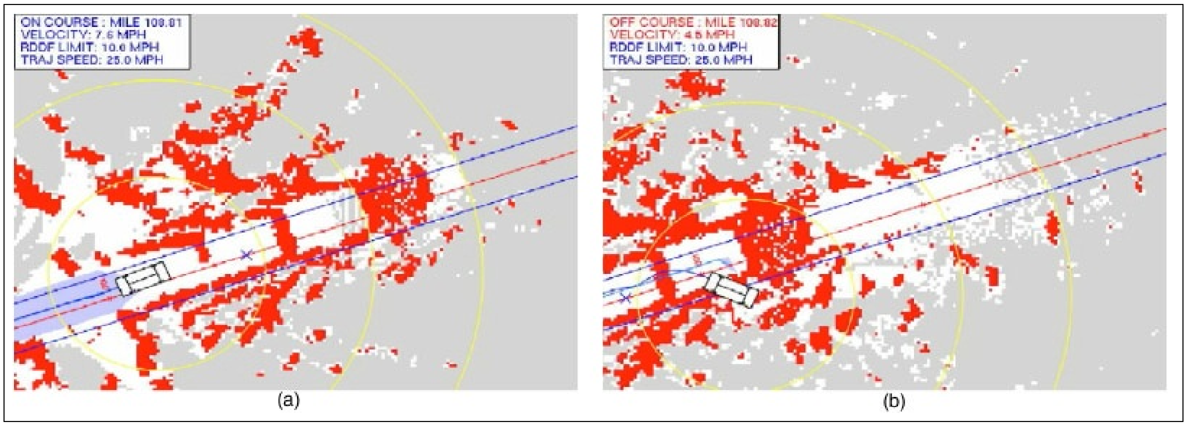

Figure 5: Small Pose Estimation errors causing misclassification through laser scanning forcing Stanley of the road.

into a 2d grid. for each grid point points in the point cloud are selected which are near. The grid cell is then declared as occupied, free or unknown by looking at the vertical difference of nearest measurements:

$$|Z_k^i - Z_m^j| > \delta (1)$$

i and j are the indices for the points and k and m are the time stamps at which the measurements were taken and δ is a critical value to decide for:

- occupied: if two or more measurements are near and formula (1) is true.
- free: if two or more measurements are near but none of them makes formula (1) true.
- unknown: if no two nearby measurements can be found.

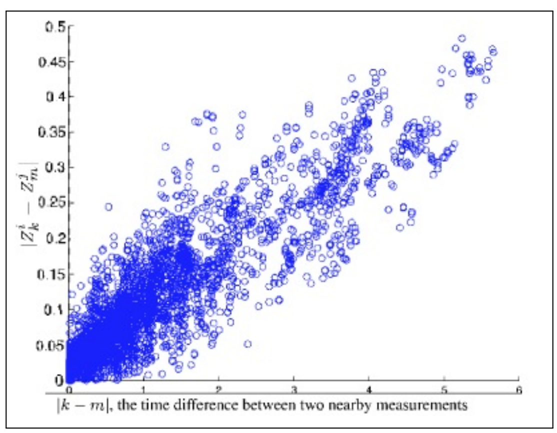

Figure 6: Correlation of time difference in measurements and Z distance of measurements.

Now we can see where the problems arise with small errors in pose estimation. Because we compare measurements from different time steps in which due to pose errors not completely correlate to each other. This will result in misclassification and terrain which isn't been blocked is classified as occupied forcing the car of the road.

In fact by looking at figure 6 we can see, that the distance between nearby measurements and the time difference are correlated. Stanley models this as a normal distribution which variance scales linearly with the time difference $|k - m|$. Obstacles can then be detected with an probabilistic test:

$$p(|Z_k^i - Z_m^j| < \delta) > \alpha (2)$$

where α is a confidence threshold (e.g. α = 0.05)

The parameters δ and α are tuned by supervised learning. Labeled training data therefore is gathered by human driving. The driver is instructed to only drive over obstacle free terrain. Thus this is labeled as obstacle free and left and right of the car in a distance (which is set by hand based on average road width) is all labeled as obstacles. Although this is not all obstacles it is sufficient enough approximation to tune the parameters.

3.2.2 Vision Extension

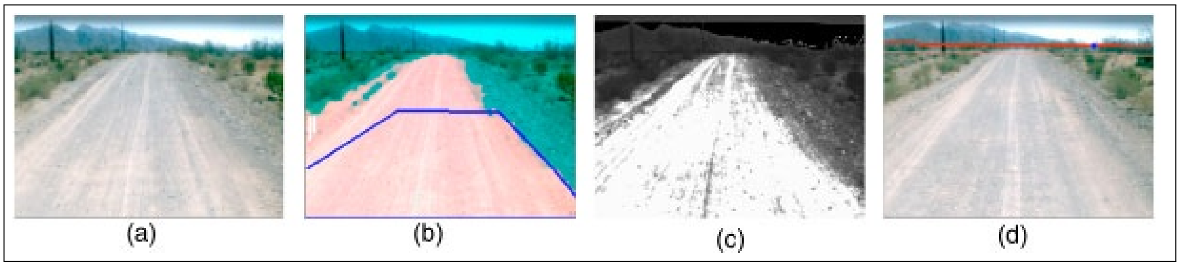

Figure 7: Vision extension: a) imput image b) projected laser scanning results c) classification results d) sky removal

The range at which the laser mapper could be used for terrain labeling doesn't give a sufficient enough range to win the challenge as it only allows for driving speed of up to 25 mph. To win Stanley needed speeds of 35mph. To increase the labeling range Stanley uses the also installed color camera. But using a camera only to label terrain in drivable and non-drivable is not easy due to changes in road appearance caused by different factors which are not easy to measure like material changes, lightning changes, dust, ... which is trying to be solved by many scientists [4][1][2]

Therefore Stanley can simplify the problem because he doesn't need to rely on camera image only. Fore a limited range Stanley already knows drivable terrain cause of the laser scanning. This can be used to get training data on the fly for the actual road condition. Therefore a quadliteral in front of the vehicle which is all labeled as drivable is projected onto the actual camera image like shown in figure 7 b). Pixels on this projected can be used as positive labeled training data which can be used to build a mixture of Gaussians which classify drivable pixel colors. Each of n Gaussian mixture components are formulized through a mean RBG color μ_i, covariance σ_i and a number of pixels m_i which were used to train this Gaussian.

For the learning algorithm it is important that it can smoothly adapt to new lightning conditions but also rapidly change if the material of the road changes resulting in completely different pixel colors. Therefore when a new image arrives k¡n new Gaussian mixture components are generated out of the projected image region. Compare the jth new Gaussian to the already saved Gaussian in memory with the mahaboli distance:

$$d(i,j) = (\mu_i - \mu_j)^T(\Sigma_i + \Sigma_j)^(-1)(\mu_i - \mu_j)$$

based on this distance find mixture component i which is nearest end decide on the following

- if the distance $d(i,j) <= \Phi$ use the new mixture component to refine the old one using the formula:

$$\mu_i := \frac{m_i \mu_i}{m_i + m_j} + \frac{m_j \mu_j}{m_i + m_j}$$

$$\Sigma_i := \frac{m_i \Sigma_i}{m_i + m_j} + \frac{m_j \Sigma_j}{m_i + m_j}$$

$$m_i := m_i + m_j$$

- if the distance $d(i,j) > \Phi$ then add this new mixture component and eventuelly delete another mixture component which pixel count m_i is smallest if already n mixture components are set.

Using this mixture components Stanley can classify pixels that are not in the region which is projected with the laser scanner results as drivable.

These results still don't guarantee to classify non drivable regions right. Since some changes in road properties are normal. If they road changes von stone to grass outside the laser scanner range the pixel intensities will clearly differ and so not be classified as drivable. So instead using this results to change driving direction it is instead used to control Stanleys speed. If no drivable corridor can be found in front at the actual driving direction the car is slowed down to a speed where laser scanning is sufficient enough. This means the vision extension can give an early warning that there might be an obstacle so better slow down but there doesn't need to be one.

4 Conclusion

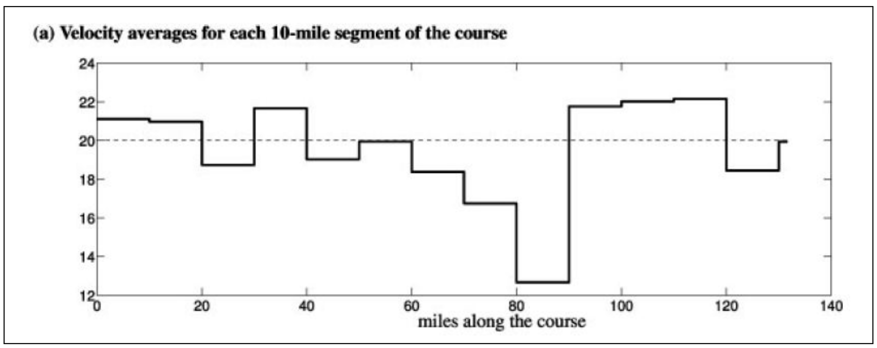

Figure 8: Average driving speeds

I introduced into the problems which arise at autonomous driving in desert terrain which where for example bad gps signal, bad road condition full of obstacles and the need for speed to win the challenge. and Computer Vision the Stanley team was able to solve the problems at an extend to finish to course far ahead of all other participating robots. Figure 8 shows the average driving speeds which Stanley could achieve throughout the course. Using Machine Learning. The two key algorithms to this success where a robust pose estimation and terrain labeling. In pose estimation using an unscented Kalman filter on switching models deciding in gps availability was the key. In terrain labeling firstly laser scanning together with an probabilistic approach give a robust classification on the terrain at close range. This classification is used to project it onto the camera image and use an learning algorithm to extend the range of obstacle detection.

References

[1] J.D. Crisman and C.E. Thorpe. Scarf: A color vision system that tracks roads and intersections. *Robotics and Automation, IEEE Transactions on*, 9(1):49–58, 1993.

[2] E.D. Dickmanns. Vision for ground vehicles: History and prospects. *International Journal of Vehicle Autonomous Systems*, 1(1):1–44, 2002.

[3] S.J. Julier and J.K. Uhlmann. A new extension of the kalman filter to nonlinear systems. In *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls*, volume 3, page 26. Spie Bellingham, WA, 1997.

[4] D.A. Pomerleau. Knowledge-based training of artificial neural networks for autonomous robot driving. *Robot learning*, pages 19–43, 1993.

[5] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, et al. Stanley: The robot that won the darpa grand challenge. *The 2005 DARPA Grand Challenge*, pages 1–43, 2007.

[6] G. Welch and G. Bishop. An introduction to the kalman filter. *University of North Carolina at Chapel Hill, Chapel Hill, NC*, 7(1), 1995.