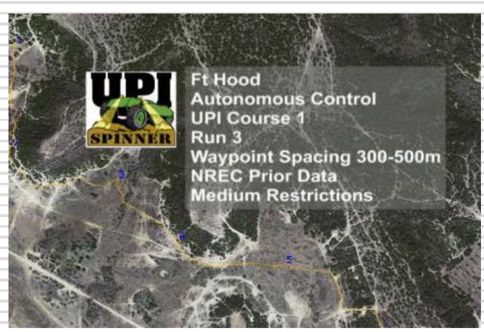
Self-Supervised Online Learning Approaches for Robot Navigation

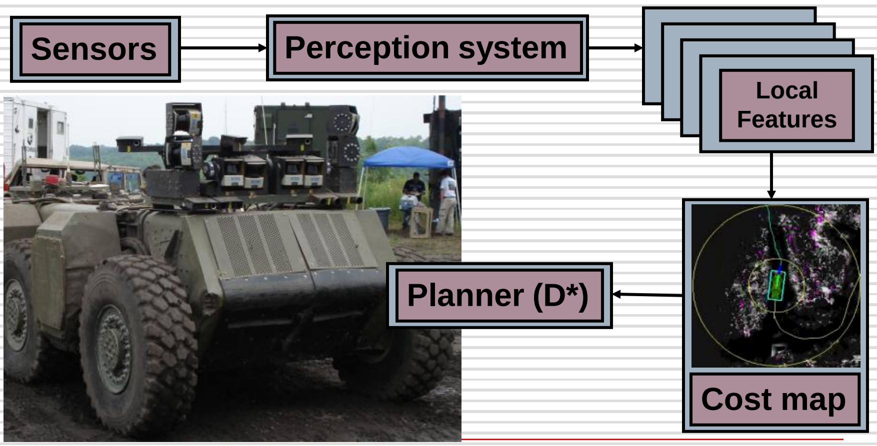
16-831, Fall 2008
October 16

Mobile Robot Navigation



2

Local Perception System



Motivation

- Sensing range
  - Onboard perception system loses effectiveness at longer ranges (past 12-15 meters in this case)
  - Results in inefficient and often dangerous exploration

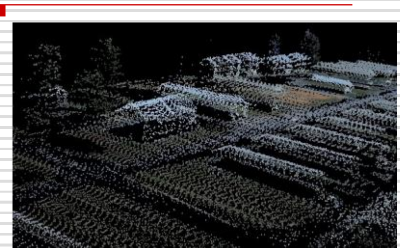
Motivation

- Sensing range
  - Onboard perception system loses effectiveness at longer ranges (past 12-15 meters in this case)
  - Results in inefficient and often dangerous exploration

5

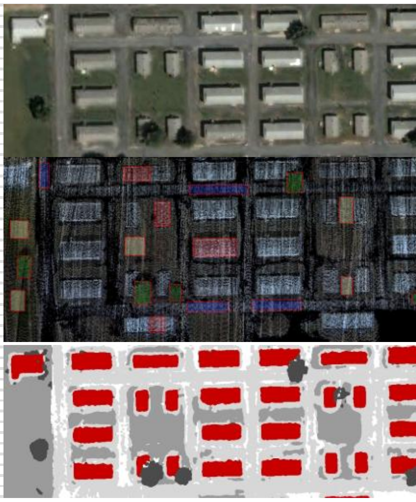
How to Improve

- Use overhead data (imagery, elevation, etc.)

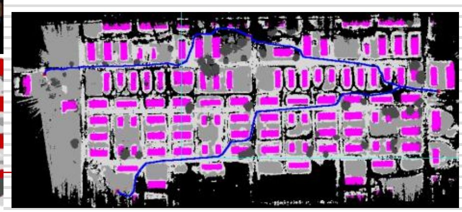


6

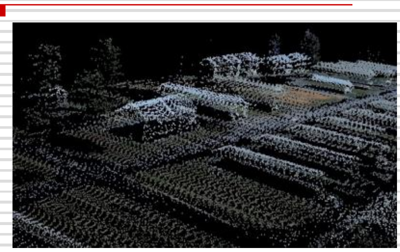
Hand-Train Overhead Interpreter



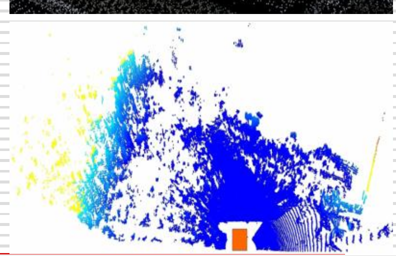
- Hand-train overhead classifier / cost predictor
- Apply to larger map
- Use resulting map for planning



How to Improve



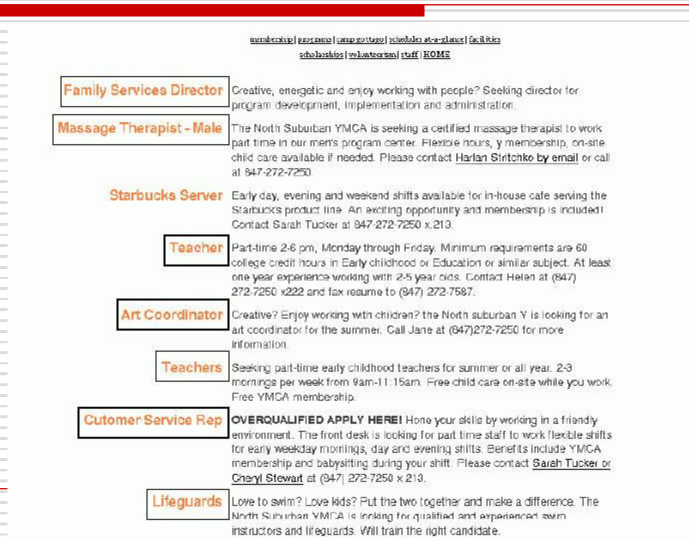
- Use overhead data (imagery, elevation, etc.)
  - Difficult to interpret consistently
    - Variations in terrain, lighting, weather, time of gathering
- Extend the range of the perception system
  - Not enough data to accurately generate perception system's features
    - Can't estimate ground plane, inaccurate density, etc.
  - Features that are computable are difficult to interpret consistently



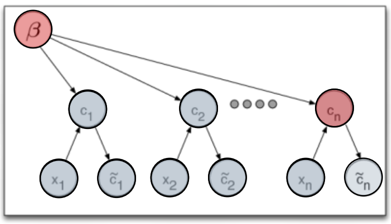
How to Improve

- Overhead data features
  - Color, texture, clustering, elevation, PCA, neighbor features, etc.
- Far-range sensor data features
  - Color, ladar point elevation spread / std, neighbor features, etc.
- How can we best use these potentially powerful, but difficult to generalize, features?

Scoped Learning



The Algorithm



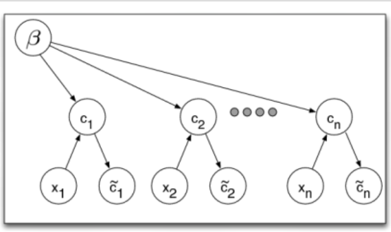
x_i - computed locale-specific features
c_i - true traversal costs
$\tilde{c}_i$ - estimates generated by perception system
β - captures relationship between x_i and c_i

$$E[c_{n+1}] = \beta^T x_{n+1}$$
$$p(c_{n+1}|\tilde{c}_{1\ldots n}, x_{1\ldots n+1})$$

- Model the relationship between these features and measured traversal cost in a Bayesian probabilistic framework
  - Online Bayesian Linear Regression
- Relate multiple data sources with different "scope" to each other

Blei, '02 11

The Algorithm



x_i - computed locale-specific features
c_i - true traversal costs
$\tilde{c}_i$ - estimates generated by perception system
β - captures relationship between x_i and c_i

$$E[c_{n+1}] = \beta^T x$$
$$p(c_{n+1}|\tilde{c}_{1\ldots n}, x_{1\ldots n+1})$$
$$p(\beta|\tilde{c}_{1\ldots n}, x_{1\ldots n}) \propto p(\tilde{c}_{1\ldots n}|\beta, x_{1\ldots n})p(\beta)$$

- Initialize the distribution to the prior distribution $p(\beta)$
- We want to compute $p(\beta|\tilde{c}_i, x_i, D)$
  - For every training example $(x_i, \tilde{c}_i)$, multiply the distribution by $p(\tilde{c}_i|\beta, x_i)$

$$P \leftarrow P + \frac{x_i x_i^T}{\sigma^2} \qquad J \leftarrow J + \frac{\tilde{c}_i x_i}{\sigma^2}$$

- By computing $p(\beta|\tilde{c}_i, x_i, D)$, we are performing self-supervised learning using a Bayesian linear regression model

12

The Algorithm

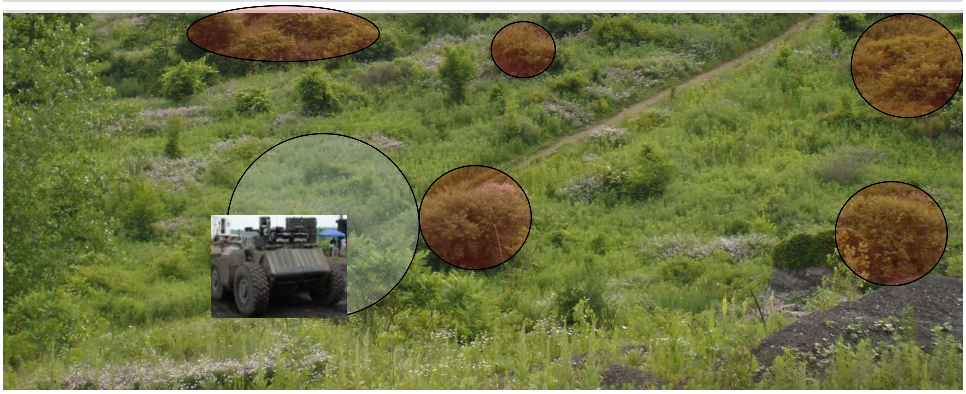
- Learn to interpret these locale-specific features by taking advantage of the globally interpretable features from the perception system

The Algorithm

- Learn to interpret these locale-specific features by taking advantage of the globally interpretable features from the perception system

The Algorithm

- Learn to interpret these locale-specific features by taking advantage of the globally interpretable features from the perception system

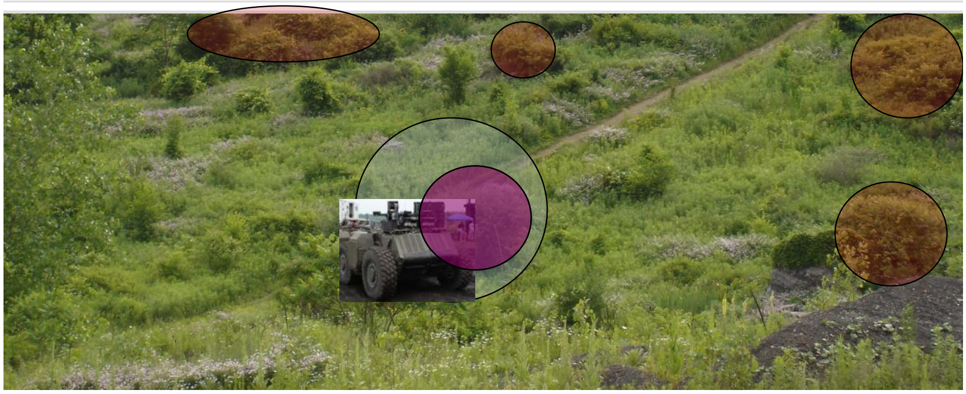


The Algorithm

- Learn to interpret these locale-specific features by taking advantage of the globally interpretable features from the perception system

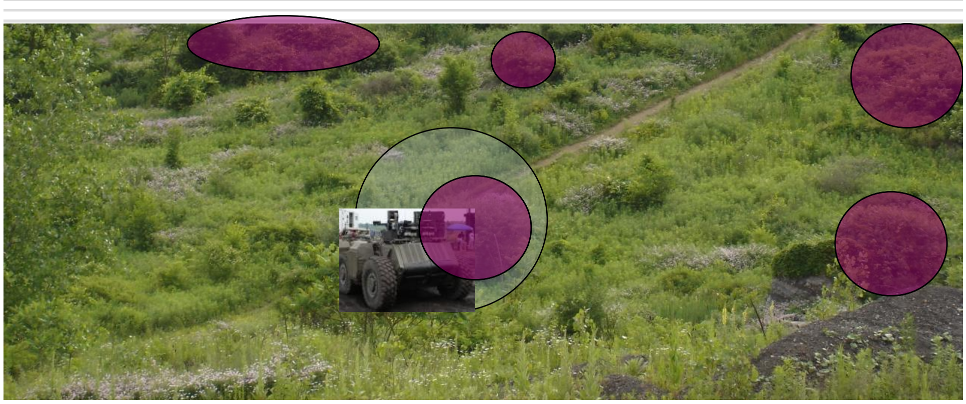
The Algorithm

- Learn to interpret these locale-specific features by taking advantage of the globally interpretable features from the perception system



The Algorithm

- Learn to interpret these locale-specific features by taking advantage of the globally interpretable features from the perception system



Results

- Overhead Online Learning
  - Online use
  - Offline use
- Far-Range Online Learning

19

Results

- Overhead Online Learning
  - Online use
  - Offline use
- Far-Range Online Learning

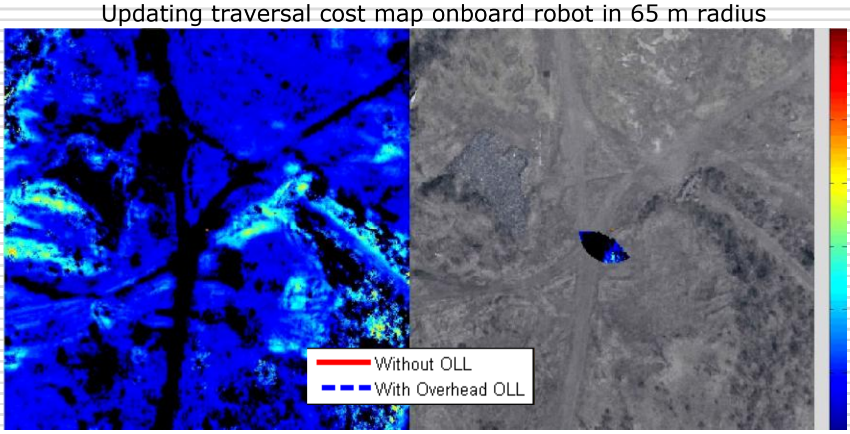
20

Results

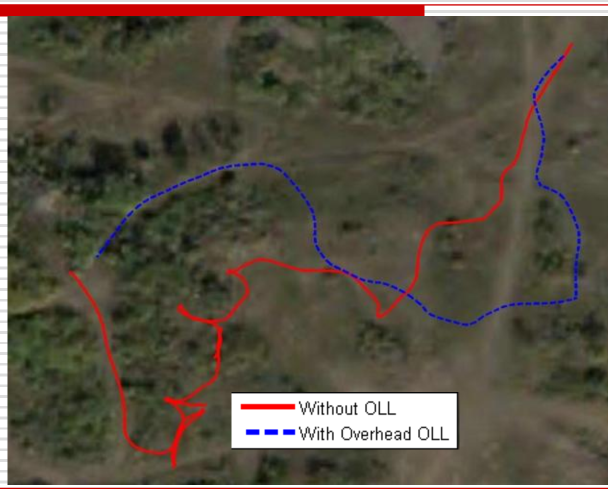
- Overhead Online Learning
  - Online use
  - Offline use
- Far-Range Online Learning

Overhead Online Learning

Using features from 40cm color imagery and elevation data
Updating traversal cost map onboard robot in 65 m radius



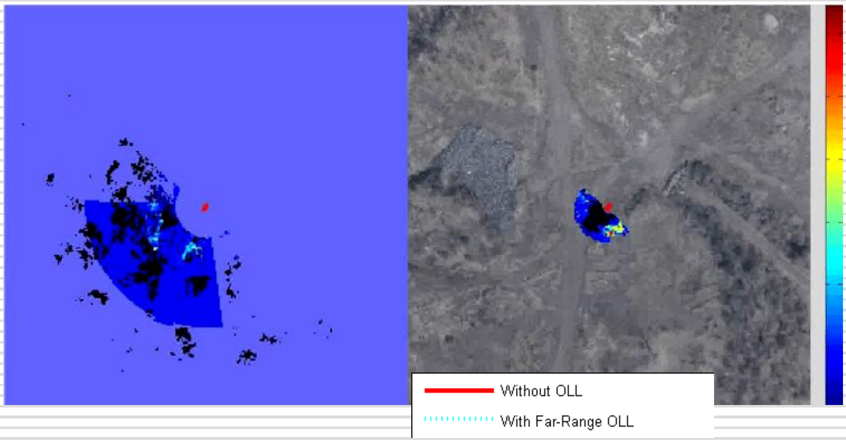
Overhead Online Learning



Results

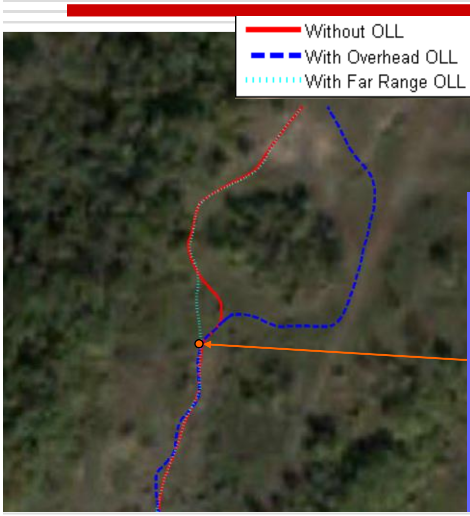
- Overhead Online Learning
  - Online use
  - Offline use
- Far-Range Online Learning

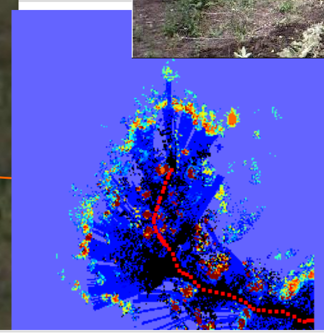
Far-Range Online Learning



Far-Range Online Learning

Results

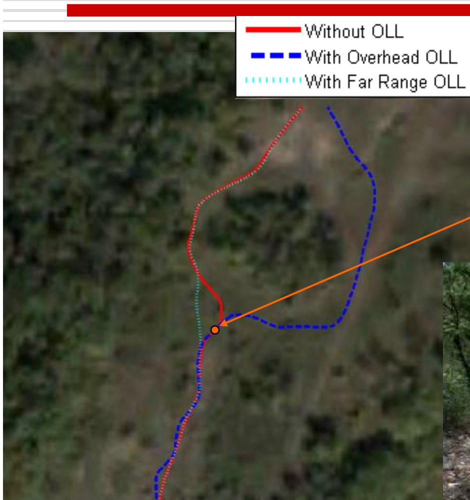




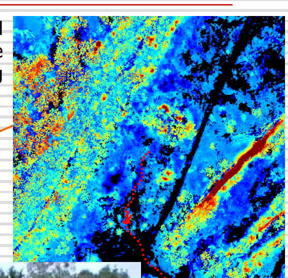
Far Range
Online Learning

27

Results



Overhead
Online
Learning



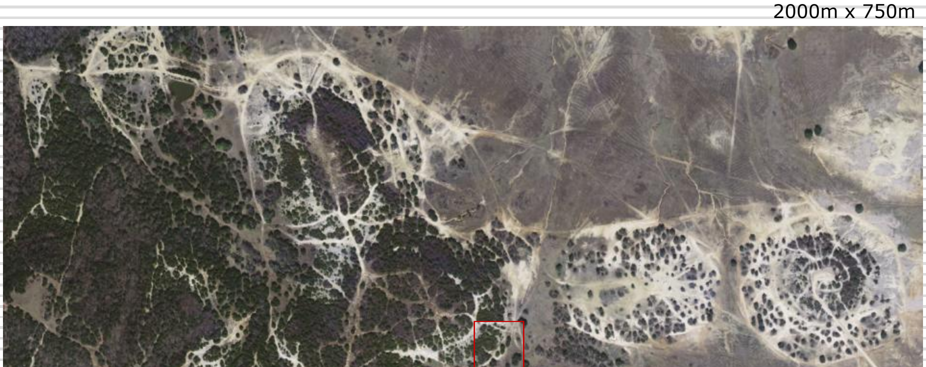
28

Results

- Overhead Online Learning
  - Online use
  - Offline use
- Far-Range Online Learning

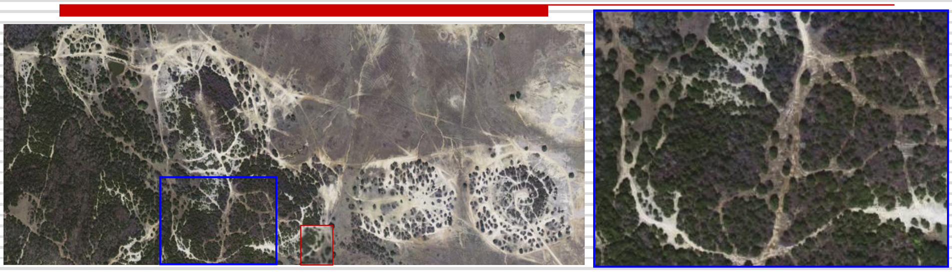
Overhead Online Learning (Offline)

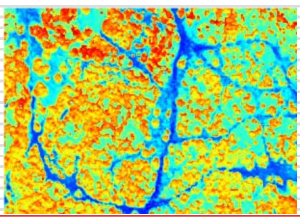
2000m x 750m



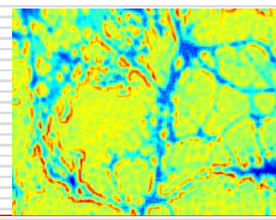
Training course

Overhead Online Learning (Offline)





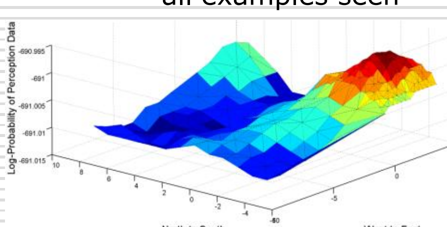
Using 35cm color imagery data

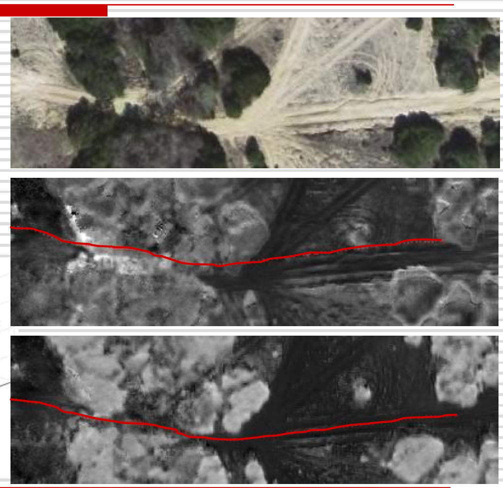


Using 1m black and white imagery data

Overhead Online Learning (Offline)

- Data alignment
  - Use $p(\tilde{c_1}, \ldots, \tilde{c_n})$ to detect most likely map alignment
  - Use alignment with the highest average log probability over all examples seen





Additional Benefits

- Reversible learning
- Confidence-rated predictions

Reversible Learning

- Multiple estimates of single quantity
  - Receive example $(x_i, \tilde{c}_i)$

$$P \leftarrow P + \frac{x_i x_i^T}{\sigma^2} \qquad J \leftarrow J + \frac{\tilde{c}_i x_i}{\sigma^2}$$

  - Receive lower variance estimate $\tilde{c}_i{}'$

$$J \leftarrow J - \frac{\tilde{c}_i x_i}{\sigma^2} + \frac{\tilde{c}_i{}' x_i}{\sigma^2}$$

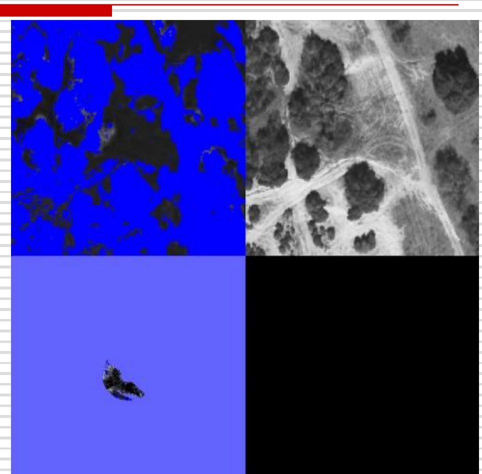
  - β always takes into account only best estimates available for all examples

Additional Benefits

- Reversible learning
- Confidence-rated predictions

Confidence-rated predictions

- Use variance estimate (HW3!) provided by algorithm for the probability of each estimate as measure of confidence
- “Surprise” at seeing set of features



Far-Range Online Learning with Velodine

Movie…

Questions?

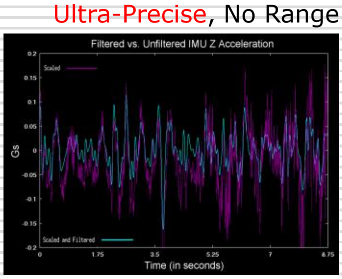
A Self-Supervised Terrain Roughness Estimator for Off-Road Autonomous Driving

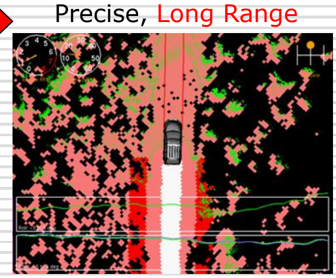
David Stavens and Sebastian Thrun
Stanford Artificial Intelligence Lab

"Combines" strengths of multiple sensors.

Ultra-Precise, No Range

Precise, Long Range





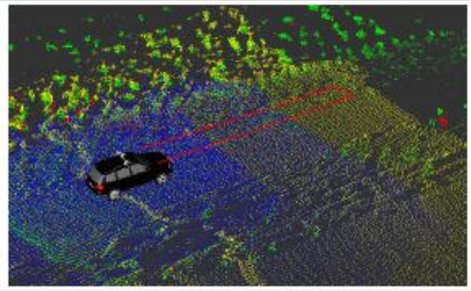
Velocity Planning for DGC 2005

- Mobile robotics traditionally focuses on steering.
- But *speed* is also important.
  - Beyond stopping distance and lateral maneuverability.
- Stanley adapted its speed to terrain conditions, minimizing shock:
  - Increases electrical and mechanical reliability.
  - Mitigates pose error for laser projection.
  - Increases traction for improved maneuvers.
  - Correlated with slowing on “hard” terrain.

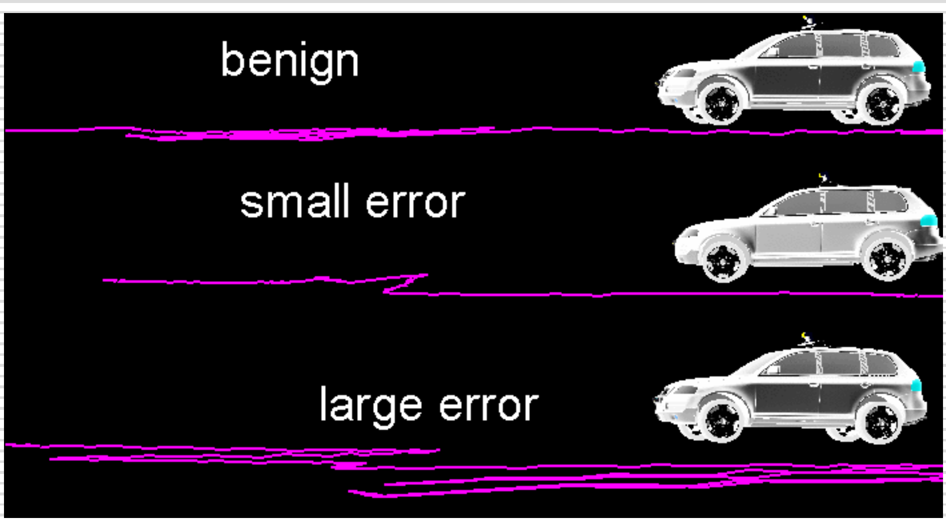
Reactive Approach (used during DGC)

- Simple three state algorithm:
  - Drive at speed limit until shock threshold exceeded.
  - Slow to bring the vehicle within the shock threshold.
    - Uses approx. linear relationship between shock and speed.
  - Accelerate back to the speed limit.

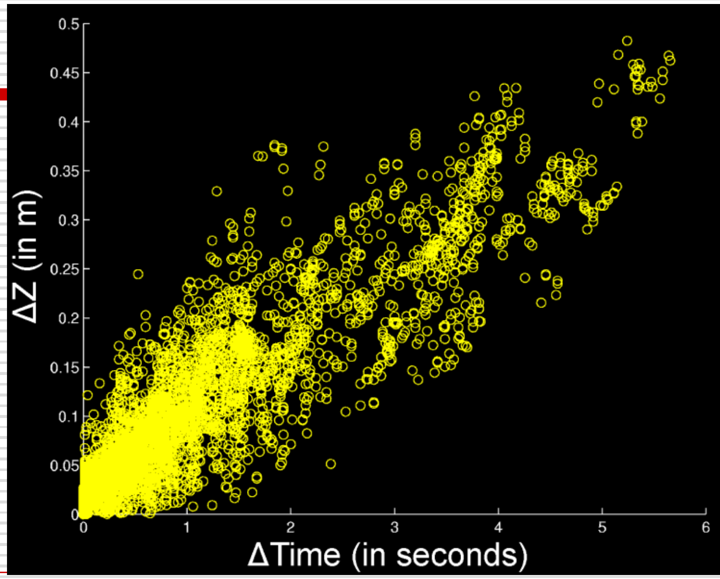
Acquiring a 3D Point Cloud



Movie…



Goal: know amount of error that is expected so that actual rough terrain or obstacles may be better identified.



More than Δt

- "Spread" of plot implies more factors than Δt.
- Also related to:
  - Amount/rate of pitching.
  - Distance between the two scans.

Comparing Two Laser Points

Uncertainty = Δ_{pair} =

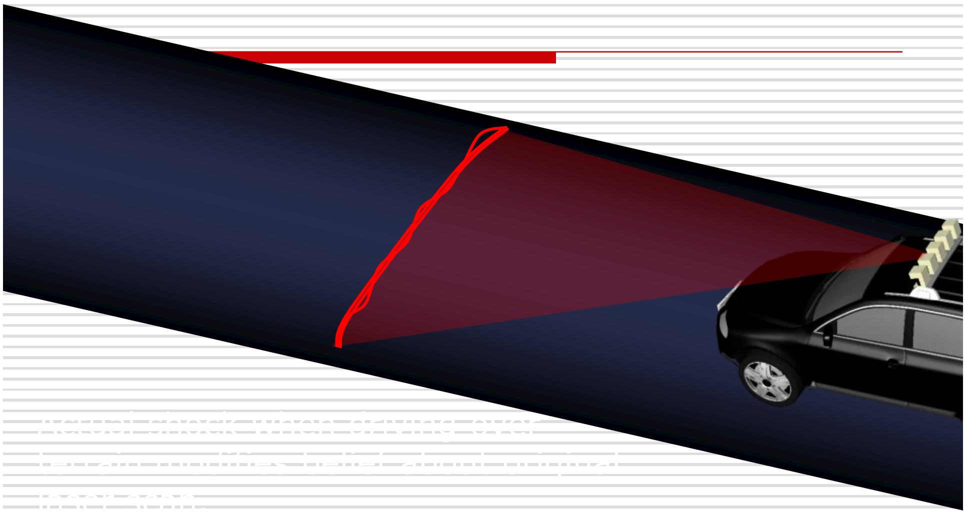
$$\alpha_1 |\ \Delta z\ |^{\alpha 2} -$$
$$\alpha_3 |\ \Delta t\ |^{\alpha 4} -$$
$$\alpha_5 |\ \text{xy distance}\ |^{\alpha 6} -$$
$$\alpha_7 |\ \text{dpitch}_1\ |^{\alpha 8} - \alpha_7 |\ \text{dpitch}_2\ |^{\alpha 8} -$$
$$\alpha_9 |\ \text{droll}_1\ |^{\alpha 10} - \alpha_9 |\ \text{droll}_2\ |^{\alpha 10}$$

- Seven Features: Δz, Δt, xy distance, dpitches, drolls
- 10 Parameters: α_1 α_2 ... α_{10} (generated with self-supervised learning)

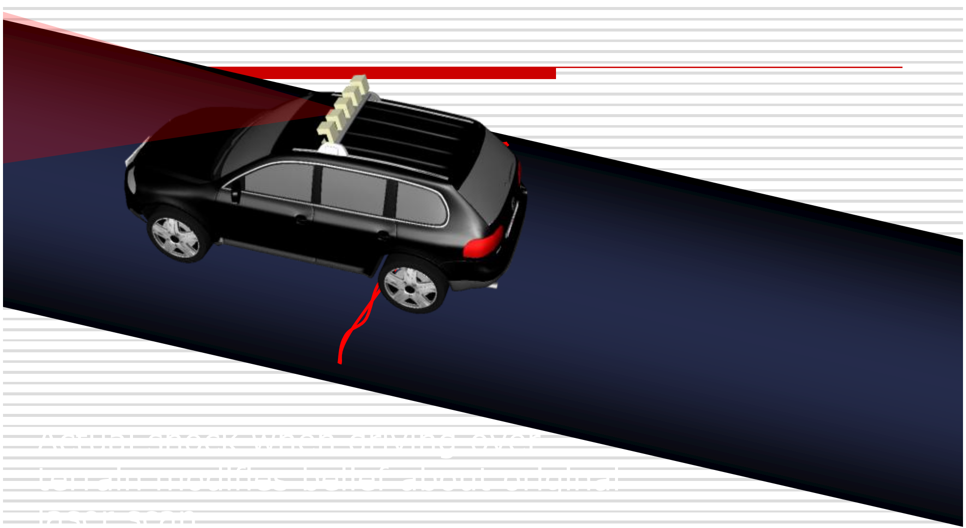
Estimate Roughness

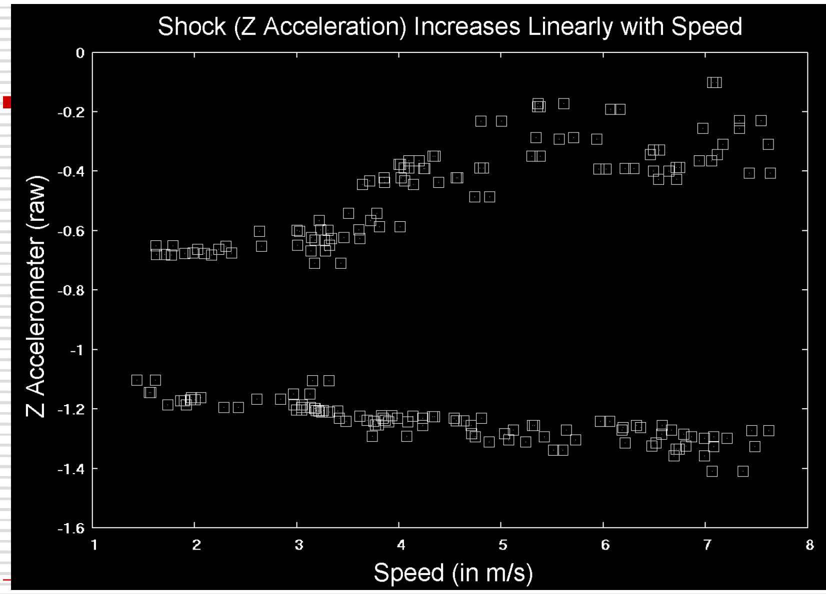
- Combine points in estimated future locations of wheels to estimate a roughness score, R, for terrain patch.
- But how do we assign target values to R?

Self-Supervised Learning



Self-Supervised Learning





Mapping from R to Shock

Learn a simple suspension model in parallel with the classifier:

$$R_{combined} = R_{left}^{\alpha 11} + R_{right}^{\alpha 11}$$

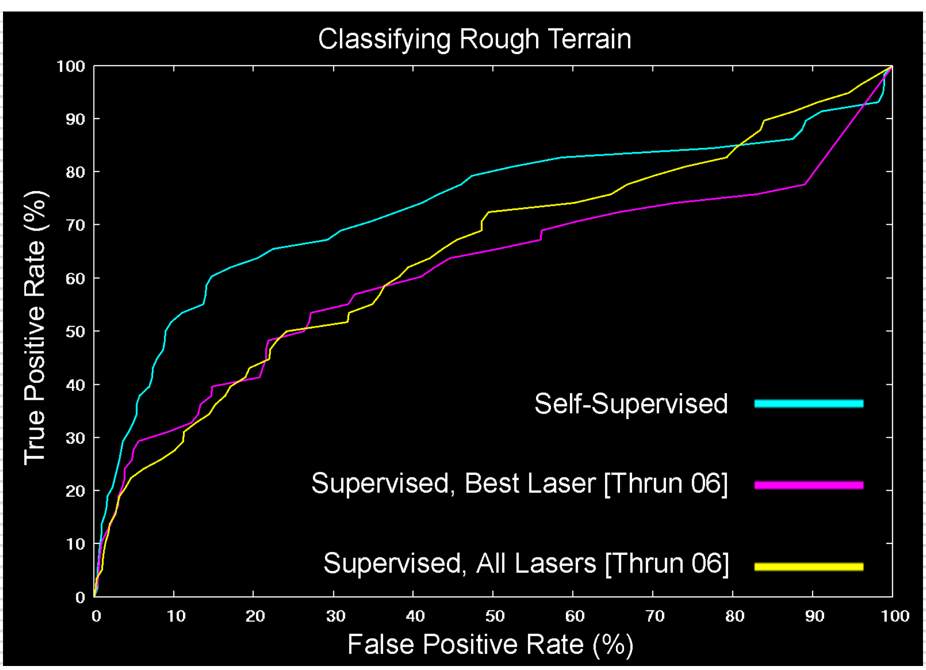
R_{left} and R_{right} is for the terrain under each wheel.

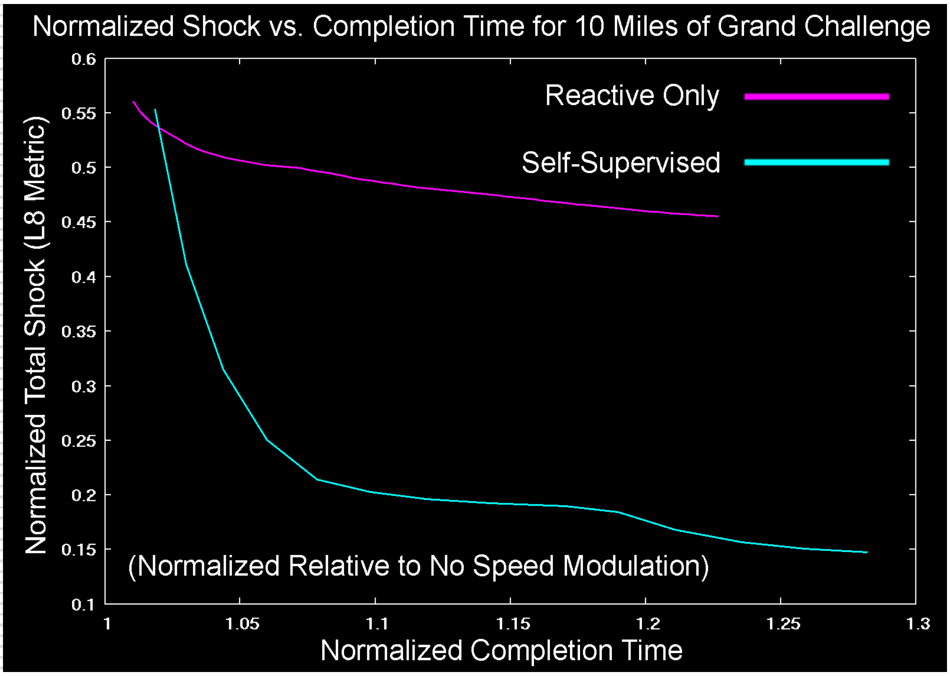
Learning Parameters

- Tp = true positive rate
- Fp = false positive rate

- Maximize Tp – λFp
  - Used λ = 5 to minimize false positives
- Optimized through coordinate ascent
  - Greedily optimize each parameter individually, decreasing learning rate each cycle by factor of 2.

53





Self-Supervised Monocular Road Detection in Desert Terrain

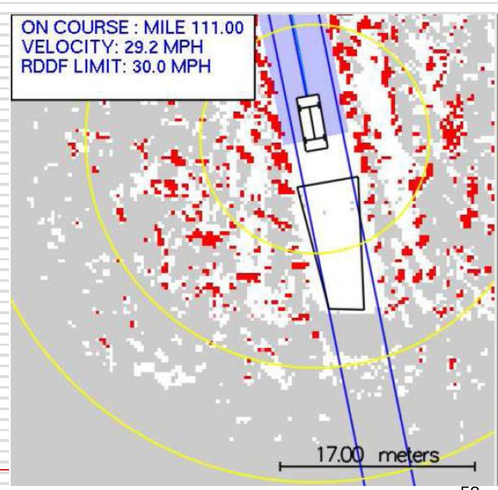
Hendrik Dahlkamp, Adrian Kaehler, David Stavens, Sebatian Thrun, and Gary Bradski

Stanford University, Intel Corporation

Goal: Detect drivable surface for aiding speed calculations

Extract "training" area using laser data

- Project onto camera image
- Assume that area contains only drivable surface
- Remove sky and shadows
- Range: ~22m



Learn visual model of nearby road

- Approximate using mixture of *k* Gaussians in RGB space
- Additional Gaussians describing training history

Score visual field by road model

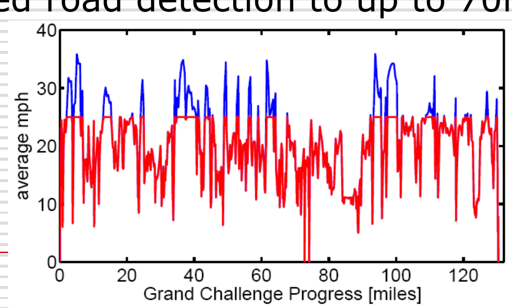
- Use distance from each pixel to nearest Gaussian to assign a "roadness" score.

Select identified patches

- Threshold image points further away than 3σ to get a binary drivability image.
- Run several filters to remove small non-drivable areas (rocks, leaves) while preserving bigger obstacles.

Usage

- Used as pre-warning system for capping speed (if can't see clear road for 40m).
- Ran at 12fps on single processor on 320 x 240 images.
- Extended road detection to up to 70m.



Pretty video…

63