Building a Local, Vehicle-centered Map
1) Robustly fit a ground plane to the disparity point cloud.
2) Fill a polar-coordinate histogram with disparity points:
   - points on the ground are ground points
   - points at any other range are obstacles
3) Determe traverseability of each histogram cell
4) Translate the polar map to a cartesian vehicle map

Overview: Trusting our eyes
- Separate local, tactical driving from global strategic driving.
- Short-range, tactical driving is done in a local, vehicle-centered map, so it cannot be harmed by positioning errors.
- Global, strategic planning is done in the global map, by a fast path planner similar to A-Star.
- Polar coordinate map is used for local map
- Cell size increases naturally with distance, azimuthal angle is precise.

Driving and Learning Strategies for Offroad Robots

The Vehicle Map
- The vehicle map combines stereo information, long-range vision information, and confidence into a final cost map that is used for local driving.
- Candidate waypoints are found in the vehicle map, and one is chosen based on the current global route. Driving commands towards this waypoint are issued.
- The vehicle map is composed into the global map on every frame.

Path Planning in the Global Map
- A-Star is optimal, but very slow due to large "open list".
- RayStar uses rays, not points, and keeps few waypoints - much faster (but not optimal).

Global Map Examples
- Robot's path is black line
- IR = Laser (non-traversable with high confidence)
- GPS = Global Positioning System
- Course of left was lawn, trees, and fences around tennis court
- Course of right was a dirt path through thick woods (top) followed by a wide concrete path through scrub and tall grass.

Spatial Label Propagation
- Expand number and variety of labeled points per frame through semi-supervised label propagation.
- Propagate labels back and forth in time
- All training points are put into a fixed-size ring buffer
- For each labeled point, query and extract all points at the same XYZ world location
- Label new points with label from querying point
- Train on all points
- For each labeled point, query and extract all points at the same XYZ world location
- Label new points with label from querying point
- Train on all points
- Label new points with label from querying point
- Train on all points

Training the Classifier On-the-Fly
- Classifier is a logistic regression on a fixed RBF layer
- Radial Basis Function centers are trained with unsupervised learning (K-means)
- Output of RBF layer is a vector of D. For input window X and RBFs $K_{x,k}$

$$D_{w} = \sum_{x \in X} [k(x, W) - \frac{1}{D_{w}^*}]$$

$$W = \sum_{x \in X} [k(x, W) - \frac{1}{D_{w}^*}]$$

Learning the Problem
- The standard approach for vision-based obstacle detection:
  - stereo-matching algorithm produces pixel cloud of disparity values
  - derive traversability map from point cloud using heuristics
- Problem: stereo range is limited to 10 to 12 meters and grows sparse and inaccurate with distance.
- Robots that navigate using stereo are limited:
  - driving into dead ends
  - missing distant paths
  - driving in a "fog"
- Humans can easily identify paths and obstacles without stereo vision, from monocular images (see examples below)

The Robot
- Platform and baseline software developed by CMU/NREC
- DARPA program: LAGR (Learning Applied to Ground Robots)
- Sensors:
  - 4 color cameras in 2 stereo pairs (passive vision only: no LADAR)
  - short-range (1m) infrared sensor
  - GPS receiver
  - Bumpers, switches
- Computing:
  - 4 identical computers with 2.0 Ghz Pentium M processors and 1 GB RAM

The Task
- Fully autonomous navigation through any terrain; learned adaptability