150 miles
across the Mojave Desert
10 hours
No driver
$1,000,000 prize

The rules – tough

- Totally autonomous. No radio communication.
  - DARPA controls remote emergency stop.
- Route announced 2 hours before race.
  - DARPA distributes CD-ROM at 0430 hrs
  - Start at 0630 hrs.
- Route is 2000 GPS waypoints.
  - Width of route varies from 10 feet to 200+ feet.
- Drivable in a 4x4 pickup truck.
- Every year until somebody finishes in 10 hours.

The terrain – hard

Team Overbot

The competitors - disappointing

- 65 entrants applied, 20 showed up.
- Only a few have a clue.
  - Caltech – student project with JPL advisors.
  - CMU – 50 students, $350K cash expenditure.
  - Overbot – not ready in time, skipped this year.
- MIT, Stanford – no entry
- No big company entries.
- Little real innovation.

The 2004 results - disastrous

What went wrong?

- **Red Team** (CMU) – At mile 7.4, plowed through sheet metal fence. On switchbacks in a mountainous section, vehicle went off course, got caught on a berm.
  - Analysis – botched tight turn control at speed.
- **SciAutonics II** – At mile 6, vehicle went into an embankment and became stuck.
  - Analysis – off course. Reason unknown.
- **Team DAD** – At mile 6, hung up on rock
  - Analysis – off course. Used GPS guidance only.

All others failed near the start

- **Golem Group** – throttle problem at mile 5
- **Caltech** – off course at mile 1.3
- **SciAutonics I** – off course at mile 1.5
- **TerraMax** – went into reverse at mile 1.5
- **Virginia Tech, Axion Racing, CajunBot, Ensco, Palos Verdes HS, Cimar, Blue Team** – failed within sight of the starting line.

Why nobody won

- Rush job – inadequate testing
- Poor sensor technology
  - Stereo vision doesn't work well on dirt.
  - Stereo from motion doesn't work yet.
  - Image understanding doesn't work at all.
  - Fixed line-scan LIDAR has too small a field of view.
  - Movable line-scan LIDAR is a mechanical nightmare.

Team Overbot

Team Overbot

- A Silicon Valley project
- All volunteers
- Many Stanford alumni
- Privately funded (about $400K)
- Startup space in Redwood City
- About 20 people over the last year.
- Four people now, starting on 2005.

Base vehicle

- Polaris Ranger 6x6.
- Most rugged platform available stock.
- Top speed 40MPH.
- Continuously-variable auto transmission.
- 2 wheel steer, 4 or 6 wheel drive.
- No computers.

Rear view

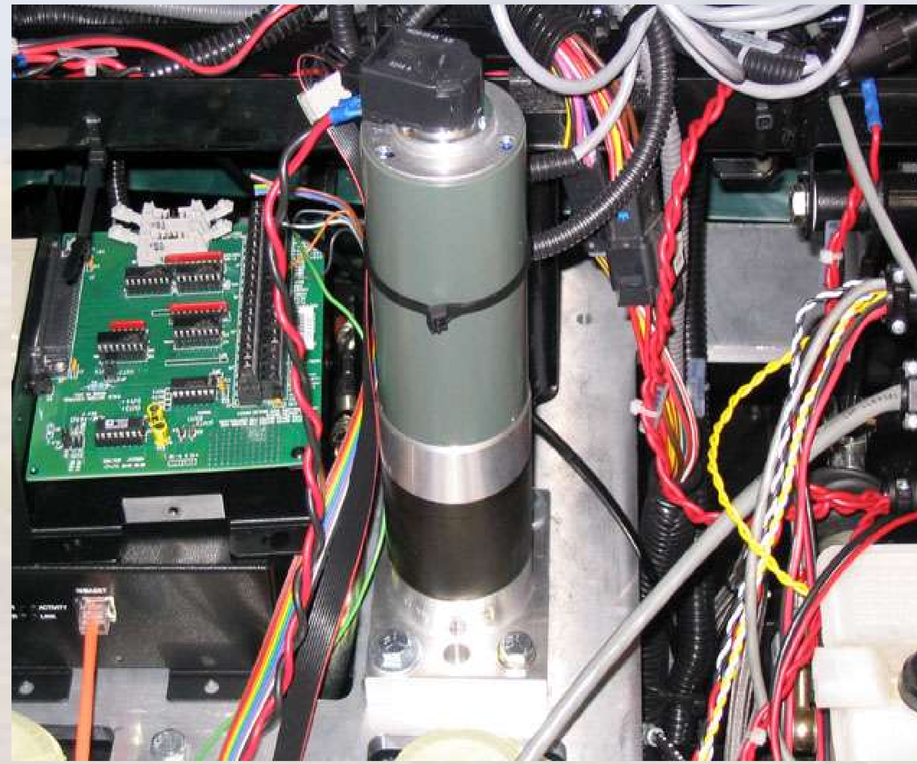
- Electronics in back bed of vehicle.
- Filtered air, but no air conditioning
- Engine generator

Computer box

- Nova 8660 single-board Pentium 4 industrial computers (2)
- Crossbow INS unit
- Industrial Ethernet hubs.
- VORAD radar interface.

Vehicle actuators

- Transmission
- Steering
- Throttle
- Brake



Distributed control

- Five Galil motor controllers.
- On Ethernet
- M680xx ColdFire CPU.
- Some programmability.
- Power amp.

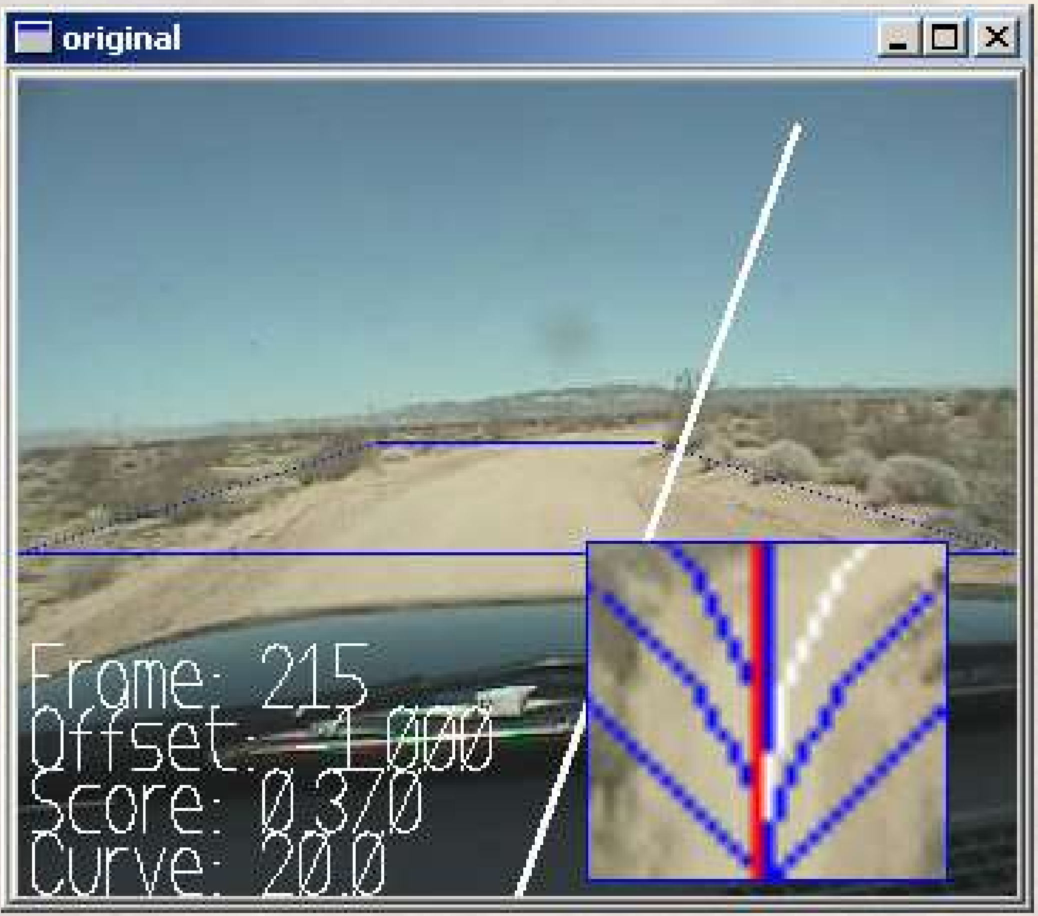
External sensors

- Scanning laser rangefinder (terrain)
- Phased array radar (anti-collision)
- Digital camera (road following)
- Inertial guidance (attitude, heading)
- GPS with corrections (location)
- Radar speedometer (odometry, slip)
- Sonars (backup, tight spots)
- Water detectors (fording)

Phased-array radar

Radar speedometer

Visual road follower



- Images from camera atop vehicle
- Images projected onto ground plane
- Looks for linear/curved uniformity
- Statistical, not neural net

Internal sensors

- Driveshaft speed
- Engine RPM
- Brake pressure
- Encoders on motors

Misc. systems

- Emergency stop
  - Engine to idle, brakes on
  - 100ms watchdog timer
  - Radio link
  - Emergency stop buttons
- 40 gallon fuel tank
  - NASCAR-qualified
- Aux generator
  - Polaris generator far too small
- LED sign
  - Only output device
- Flashing lights
- Horn

Sensing the terrain ahead

- Stereo vision
  - Dirt doesn't have enough edges for stereo lock.
  - Sizing potholes and ditches is tough.
- Submillimeter radar
  - Technology not here yet, although promising
- Image understanding
  - Very hard problem. Long history of failure.
- Laser rangefinders
  - Good range and data quality, but have limitations.

The key to success: better laser rangefinders

- Existing devices are big, heavy, slow, expensive, and have too many big moving parts.
- This is an electronics problem Silicon Valley can and should solve.
- Much R&D has been done with DoD funding, but few if any products have resulted.
- There's an opportunity here. The true 3D camera is within reach.

What we have now
Laser line scanner on tilt mount

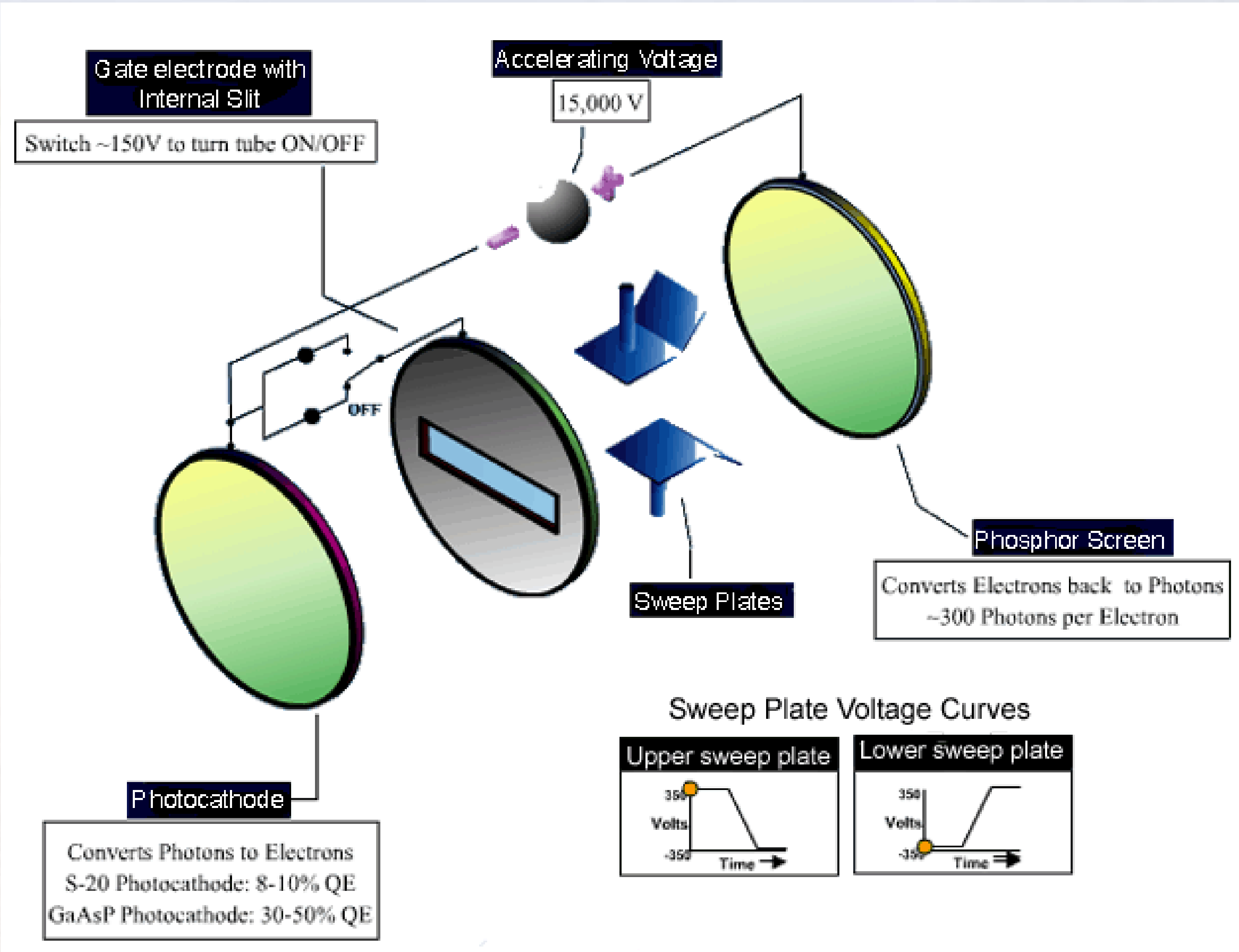
- Spinning prism scan
- Custom tilt head

What CMU has:
Laser line scanner on 3D gimbal

- REIGL spinning prism line scanner
- Big, heavy 3 axis stabilized gimbal.
- Expensive (>$100K).
- CMU unit damaged in crash, backup unit performed badly.
- Technical dead end

Team Overbot

Streak tube 2D flash LIDAR receiver



Arete Industries (Tucson, AZ)

Flash line scan

Works in sunlight.

Claimed cost $20K per tube.

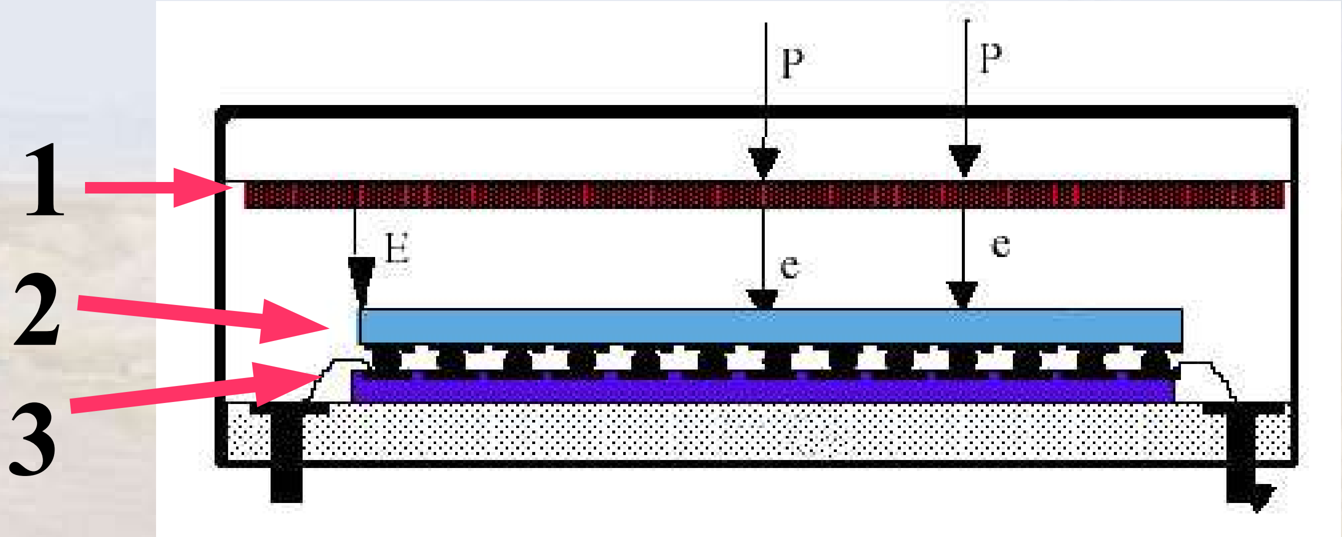
In prototype.

Team Overbot

ADLR 3D Flash LIDAR prototype

- Advanced Scientific Concepts, Inc. (Santa Barbara)
- 128x128 pixels
- True 3D depth images
- Sunlight tolerant
- > 50m range.
- Cost: $20K to $70K

ADLR detector principles

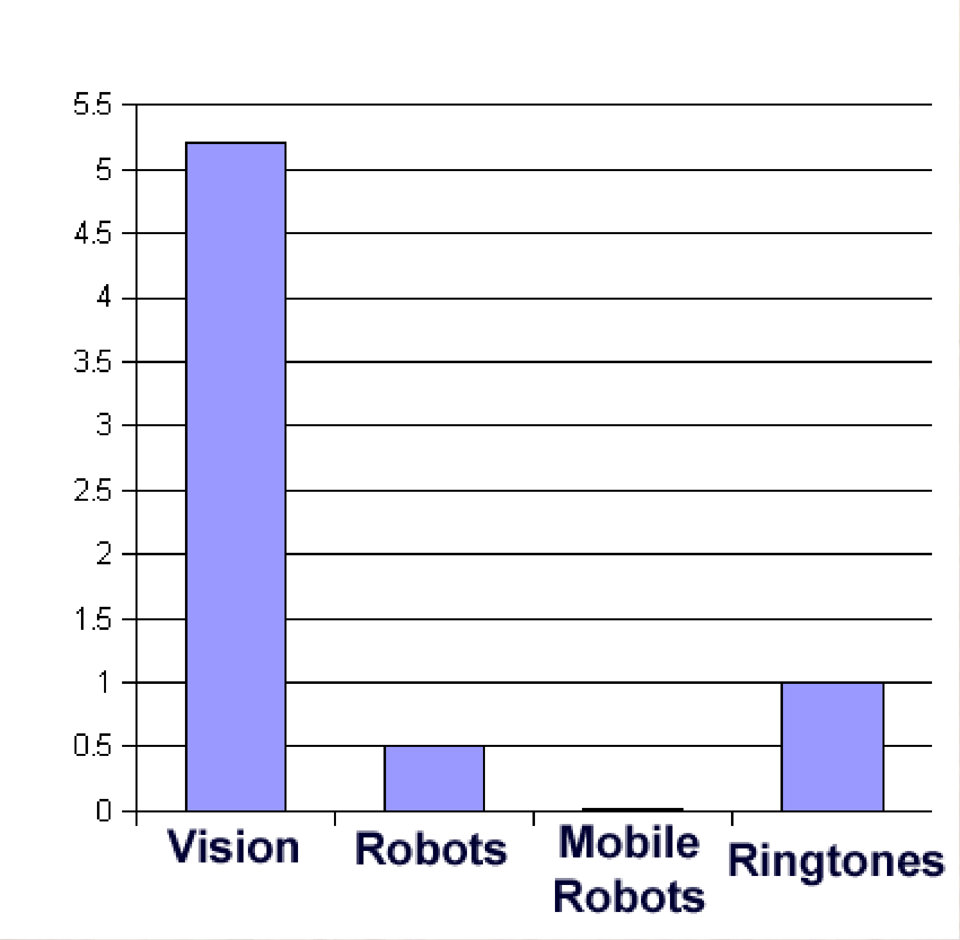


- **1.** Photons hit intensifier photocathode. Electrons emitted. 1KV+ electric field accelerates electrons.
- **2.** Accelerated electrons hit silicon detector array.
- **3.** Timing/readout chip bonded to detector array times incoming events for each pixel.

Other no-moving parts LIDAR R&D

- CSEM (Switzerland)
  - Low cost, short range, CW (not flash)
  - Not sunlight tolerant
- Raytheon missile seeker
  - > 1KM range, not eye safe
- MIT Lincoln Labs (?)
- General Dynamics Robotics
  - One mechanically scanned axis, 16 lasers

True 3D camera technology is almost ready for the real world



2003 sales, US$ BN

- It's time to commercialize this technology. But not primarily for robots.
- Computer vision is a viable industry. Robotics is tiny.
- General 3D camera applications as image quality improves.

Team Overbot

Automatic Driving

- With all this data from sensors, how do we drive?
- We have 1000+ waypoints and a really good GPS.
- Now what?

Approach #1 – Preplan in detail

- Get available high resolution imagery and elevation data
- Make plan
- Follow plan using GPS.
- Classic “Stanford approach” (Latoumbe)

Available imagery

- Off-the-shelf imagery
  - Resolution high in major US urban areas, low in Mojave Desert.
  - Elevation data very low res (20m)
- Overflight with LIDAR-equipped aircraft
  - 20cm resolution.
  - $0.50/acre, minimum $5000
  - Area of race is 20,000 square miles.
  - CMU did this. Didn't help

Why preplanning won't work

- Available imagery resolution too low.
- Need to see anything the vehicle can't roll over.
- DARPA may place obstacles on route.
- CMU tried and failed.

Approach #2 – reactive behaviors

- React to sensor data
  - "MIT approach" (Brooks)
  - No maps, no models.
  - Deals well with unstructured environments.

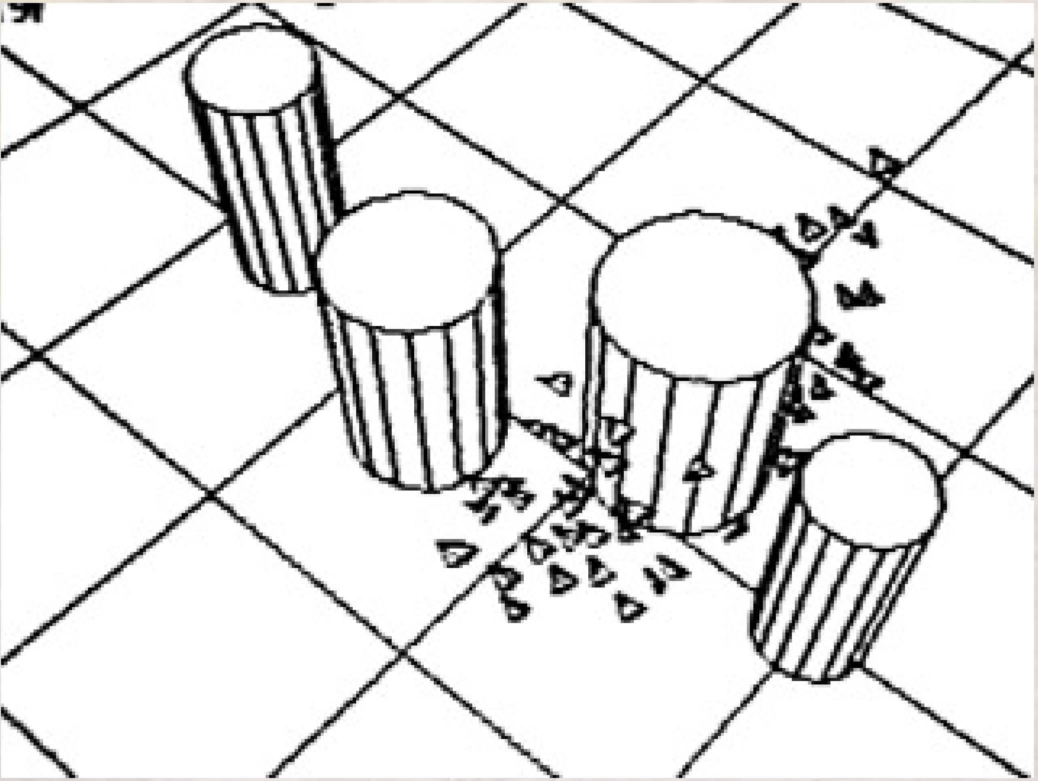
Limits of reactive behaviors

- Slow, bumbling movement.
- Maxes out at insect-level AI.
- Insects work best when viscosity dominates inertia.
- Work on purely reactive systems peaked a decade ago.

Approach #3 – game AI

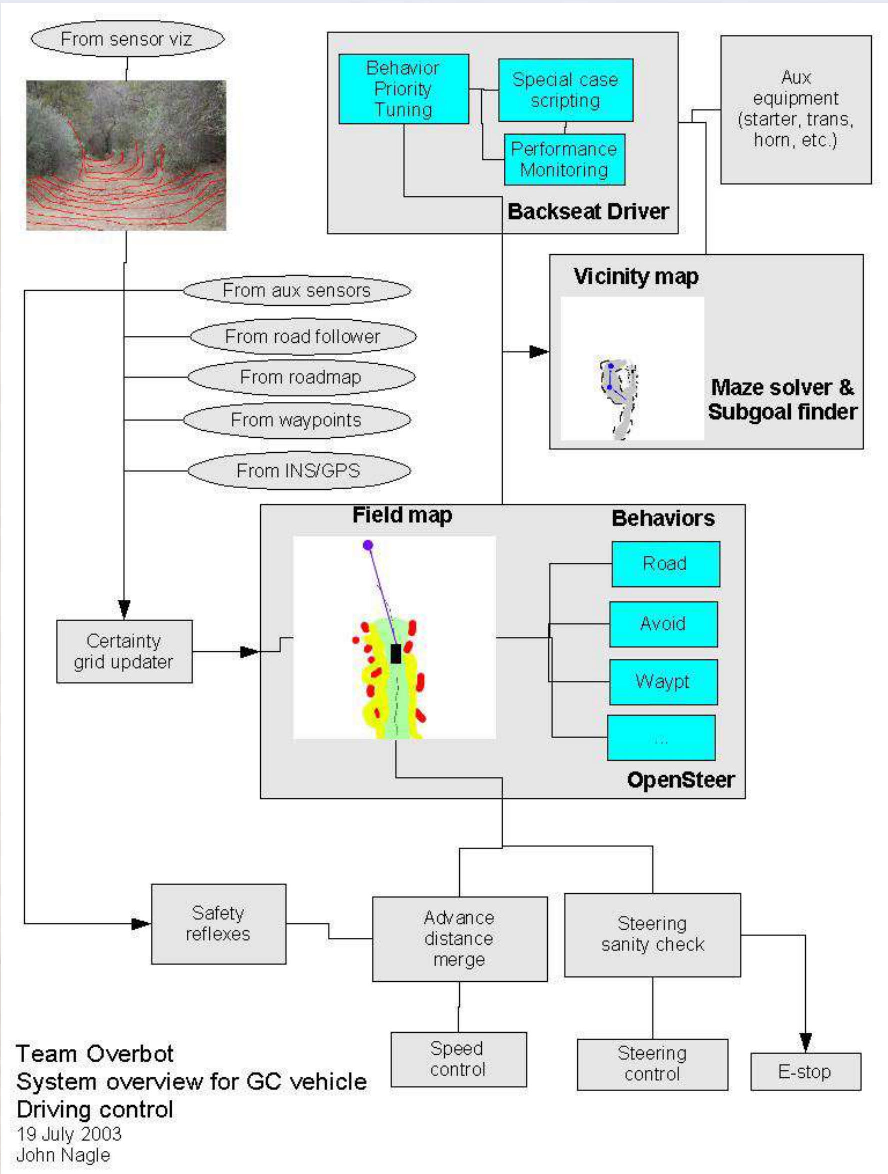
- Games have good vehicle control.
- Game AIs don't cheat as much as they used to.
- Game AI R&D efforts sizable – real applications.

Field-based control



- First seen in Reynolds' “boids” (1985)
- Attracted to goal (constant)
- Repelled by obstacle (1/x, exp)
- Improved versions widely used in games

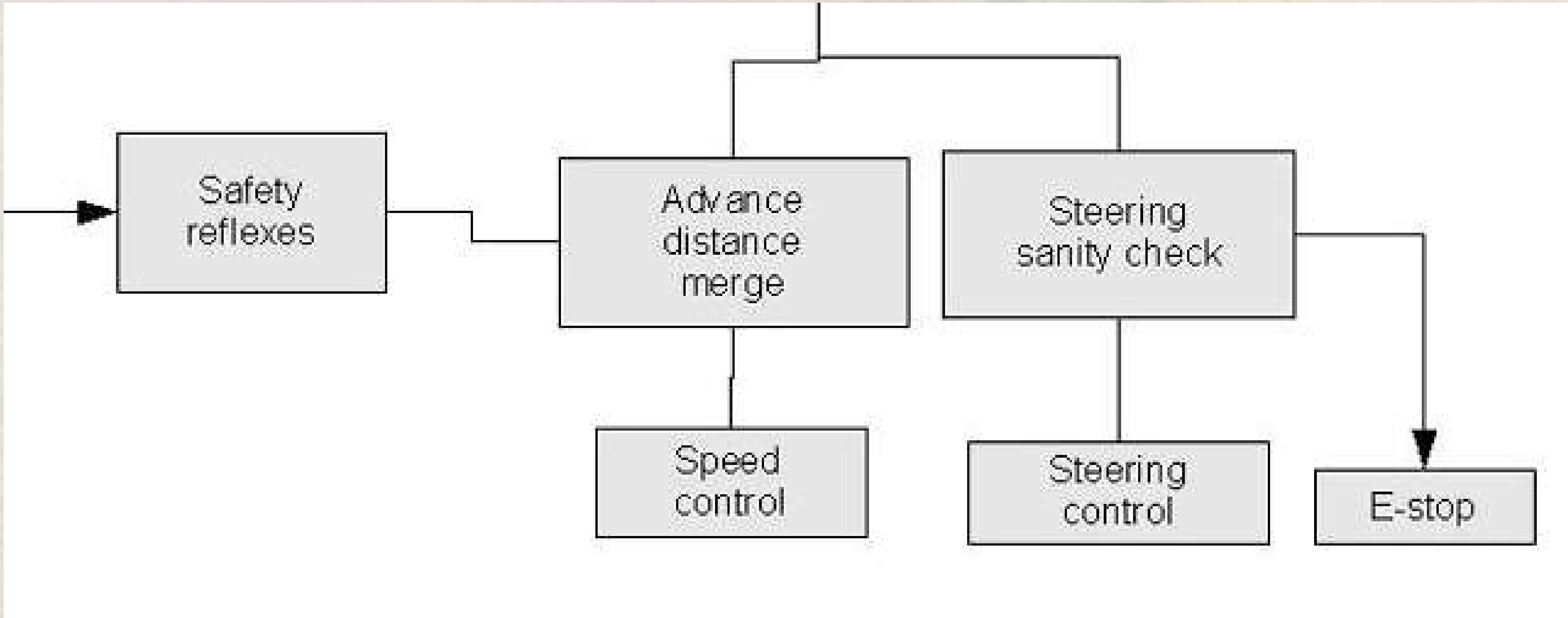
Our architecture



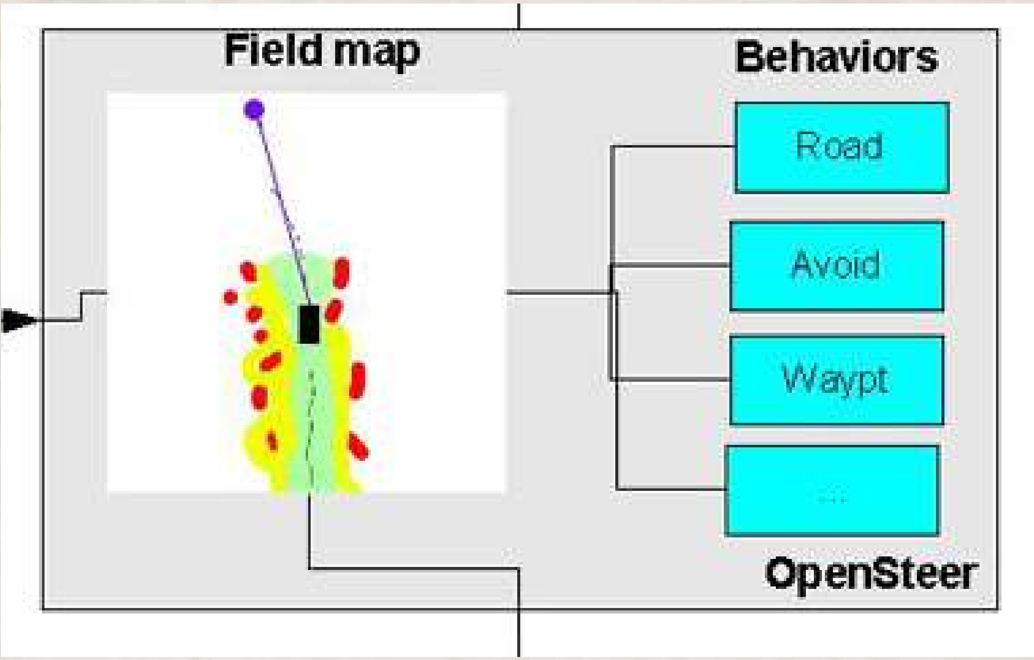
- Reactive survival behaviors at the bottom.
- Field based map in the middle
- "Back seat driver" adds subgoals when stuck.

Low-level – reactive behaviors

- Self-protection reactions
- Anti-collision, anti-tip, etc.
- Has veto power over higher levels.

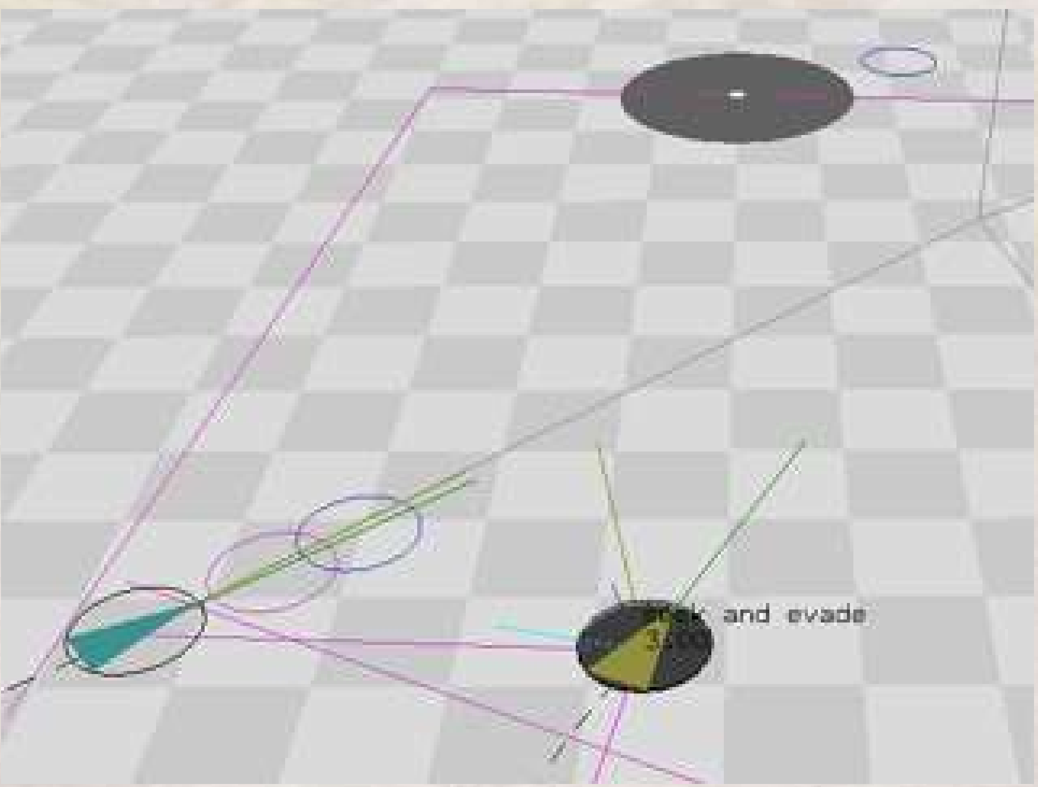


Mid-level – field map



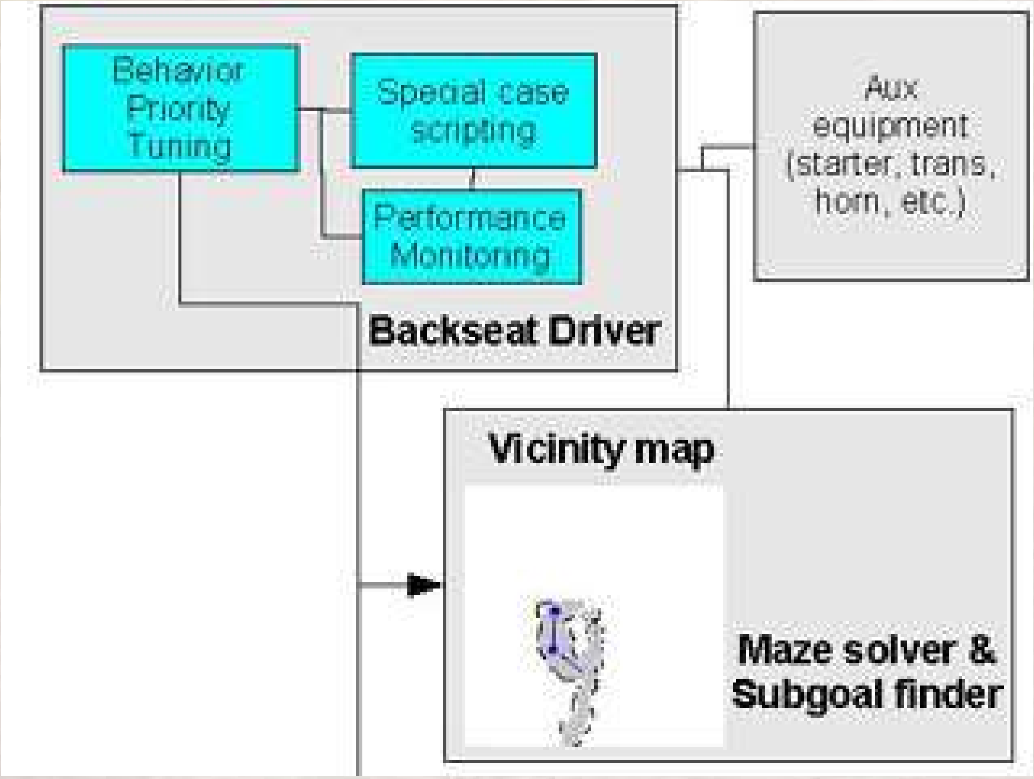
- Does most of the work.
- Map centered on vehicle.
- Sensors put data into map. GPS/INS data used to scroll map.
- Borrowed from game technology

OpenSteer as a starting point



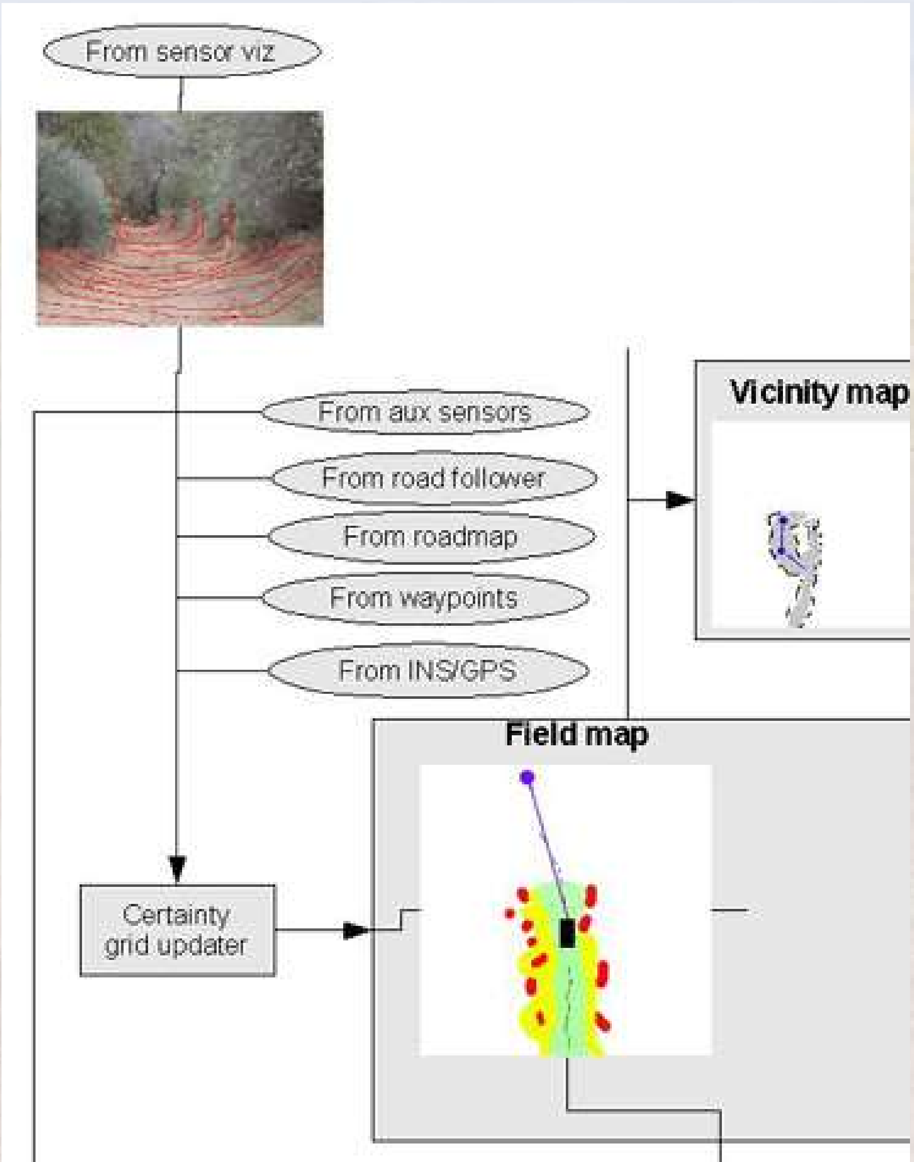
- Starting from OpenSteer (Reynolds, SourceForge)
- Changes to model
  - Obstacles as certainty grids
  - Vehicle path as snake
  - World map scrolls

High-level - “backseat driver”



- Limited authority, hence the name.
- Can change goal direction; can't directly control vehicle.
- Classic A* planner.
- Used only when field map stuck in local minima.

Map updating



- All sensors feed into map.
- Map is centered on vehicle.
- Map moves with vehicle.
- Map is a certainty grid (Moravec)

Rejected approaches

- “Learning” (a.k.a. hill-climbing)
  - Terrain too varied
  - Poor-performing systems hard to improve.
- More vision processing
  - We have a good road-follower. That's do-able.
  - General image understanding still doesn't work.
  - Stanford vision group eliminated last year.
- Logic-based AI
  - Hard to map to real world.

Underlying OS – QNX

- Hard real-time microkernel message passing OS
- Guaranteed response times for external events
- True microkernel
  - Very stable. Seldom changed.
  - Kernel: message passing, clocking, memory management
  - File systems, networking, etc. are user programs.
- POSIX compatible, but not UNIX-based.
- GCC toolchain

Software architecture

- Many small processes, rather than a few big ones.
- Intercommunication via message passing.
- Asynchronous organization.

Restart

- Main reliability feature is ability to restart everything.
- Watchdog process monitors all others.
- Hardware watchdog monitors watchdog process.
- Fault trips emergency stop hardware.
  - Brakes lock, engine drops to idle, everything restarts.
- Programmers are told that it's OK to abort, but not OK to fail to start.

Summary

- A good problem – there's no faking it.
- Hard, but solvable
- We need volunteers
  - **Two senior software leads.**
  - **Electronics techs.**
  - **Control engineer.**
  - No pay, some risk, share the $2,000,000 prize.
- Web site: **www.overbot.com**
- Questions?

Beyond Overbot - Applications

- 3D cameras
  - Effects
  - Inspection
  - Security
- Military
  - Limited use in combat environment – too confusing
  - Useful for logistics
- Automotive
  - Parallel parking
  - Rental car return
  - Off-site parking
  - Forklifts, tractors, trucks