CS 378 - Autonomous Vehicles in Traffic I

Week 1 - Course Introduction
Teaching Staff

- Piyush Khandelwal - piyushk@cs.utexas.edu
- Jack O’Quin - jack.oquin@gmail.com
- Todd Hester - todd@cs.utexas.edu
- Jesse Vera - advisor21@gmail.com
- Kevin Chang - kchang@cs.utexas.edu
- Todd Swinson - swinson@gmail.com
Today

- DARPA Urban Challenge
- Class overview
Autonomous Vehicles

- An **autonomous ground vehicle** is a vehicle that navigates and drives entirely on its own with no human driver and no remote control.

- Through the use of various sensors, the vehicle determines all the characteristics of its environment required to enable it to carry out the high-level task it has been assigned.
DARPA Urban Challenge
Grand Challenge History

**DGC I**
Barstow to Primm
March 13, 2004
142 miles, 10 hours
$1M; no winner
Best result 7 miles (CMU)

**DGC II**
Desert Classic
October 8, 2005
132 miles, 10 hours
195 contestants
5 finishers
$2M Stanford wins

**DGC III**
Urban Challenge
November 3, 2007
60 miles, 6 hours
89 contestants
$2.75M (1st, 2nd, 3rd)
Previous Challengers
Even the top teams had problems...
DARPA Urban Challenge overview

Program Objective

Safe autonomous driving in traffic

• Safe
  No collisions

• Capable
  Turns, stops, intersection, passing, merging, parking, following

• Robust
  Blocked roads, erratic drivers, sparse waypoints, GPS outage
Austin Robot Technology (ART)

● Group of local hobbyists
  ○ Built the car for the 2005 Grand Challenge
    ■ Made it to the semi-finals that year
  ○ They own the car.
    ■ So listen to them if they are around

● Team members:
  ○ Arturo Martin-de-Nicolas founder
  ○ Juan Martin-de-Nicolas mechanic / fix-it guy
  ○ Don McCauley electronics and computers
  ○ Jack O’Quin low-level/high-level software and testing
  ○ Jorge Martin-de-Nicolas - low-level software
  ○ Others: Jon Brogdon, Dave Tuttle etc.

● In 2007, the CS378 class (led by Prof. Peter Stone) joined the ART team, several of which attended the national event.
Marvin
Specific Challenges in Urban Driving

- Need to sense far ahead in order to safely navigate at 30 mph

- Need to detect static and dynamic obstacles around vehicle
  - Ignore (mostly) approaching vehicles in other lanes

- Need to obey traffic laws, re-plan at road blocks
2007 Results: First steps

- 89 teams were accepted in 2006

- Site Visit (Basic navigation and intersection management; no moving traffic)
  - Track A teams automatically got site visit
  - Track B teams (including us) had to submit video of the vehicle autonomously driving a loop and passing a stalled vehicle (class goal for cs378 in Spring 2007)

- 35 teams passed site visit
  - ART was one of those teams
  - Much of the ART code was created by the juniors/seniors in cs378
2007 Results: National Qualifying Event

• After Site Visit, most code above the driver level was re-written
  ○ 3 months

• 35 teams at NQE
  ○ Decommissioned Air Force base in Victorville, CA
  ○ 3 test areas
    ■ Merging into and across moving traffic
    ■ Long term navigation/parking/gauntlet
    ■ Site visit style test

• 20 teams were supposed to make the final
  ○ Only 11 teams ended up in final
    ■ We placed somewhere between 12th and 21st.
NQE Team
Videos

- On-board video compilation from our vehicle
  - [http://www.youtube.com/watch?v=sHbdr3LAEfg](http://www.youtube.com/watch?v=sHbdr3LAEfg)
Next Challenge

● There is not another planned DARPA competition.
  ○ The MAGIC (Multi Autonomous Ground-robotic International Challenge) Competition in 2010 was close to the scale of the DARPA Grand Challenges.

● The 2007 Urban Challenge was a big step forward, but . . .
  ○ No pedestrians
  ○ Final race much easier than NQE events
  ○ Teams still rely heavily on expensive computing/sensing capabilities
  ○ Vehicles that work 9/10 times are not good enough

● We now do research outside of the DARPA competitions through the FRI stream
Class Overview
About this course

● This is a *unique opportunity* for undergraduates at the freshman level to engage in research

● Work will be spread over 2 semesters
  ○ You can optionally also work over the summer

● This year we'll have training in the Spring and guide your research in the Fall.

● There are no prerequisites. However this research stream requires some programming skills. So you'll need to work on those over the Spring semester.
Why is this course hard?

● Research is hard because it requires
  ○ self-motivation
  ○ perseverance
  ○ a good deal of luck and ingenuity
  ○ not slacking off

● Programming
  ○ Research in this stream involves a lot of systems development.
  ○ Decently advanced programming skills are required.
  ○ As of now, we are limited to C++ or Python.

● Difficulty
  ○ That of a graduate level course
More about programming

- If you know Java but not C++
  - We'll go through 1 lecture on moving from Java to C++
  - A ton of C++ tutorials online - I'll point you to these

- If you don't know programming
  - We don't have the time to go through a lot of basic programming concepts in class
  - However, we have a number of people to help you with programming outside the lecture hours
  - You can also try enrolling in a programming course this semester itself.
    - Learning programming will require quite a bit of self-effort, but as long as you try you'll be fine.
Course Design

- Spring
  - This semester is about acquiring the skill-set required for research in this stream!
  - Lectures and assignments are designed to help improve your programming skills and knowledge about ROS
  - We won't explicitly discuss research topics in class until the last couple of weeks. If necessary, I'll discuss research material with you in office hours.

- Fall
  - We'll concentrate on research this semester
  - Run like a graduate level research course
  - We'll read a research paper and discuss some advanced research topics every week.
Spring Semester

● Things we'll discuss in class
  ○ writing C++ programs and debugging.
  ○ working with our codebase
  ○ the ROS framework

● 2 tracks
  ○ Training - Assignment track (7 programming + 3 reading)
  ○ Research - Project track (3 or 4 programming + 3 reading + project)
  ○ You can choose track halfway through semester
Fall Semester

- At the end of the spring semester, you'll have some idea whether you want to continue with the stream into the Fall.

- You'll start work on your project within the first couple of weeks.
- We'll have a short recap of material from the Spring - a couple of weeks (and maybe an assignment)
- Then we'll switch to research paper discussions + lectures on advanced topics
- You'll have to send in project updates or have weekly meetings
What I expect from every student

● Ask questions
  ○ You have a problem, let us know. *Follow it up!*

● Expand your knowledge
  ○ For instance, if we give you a list of commands to run: *look up what they do!*

● Search for solutions on the internet
  ○ If you have a problem, someone else has probably had it in the past as well.

● A serious effort to learn and improve programming skills

● Working on your own time
Why work hard on this course?

- Because autonomous vehicles are pretty cool

- Knowledge for future directions
  - Do I want to go to grad school?
  - Work in a research lab?

- Publish academic work
  - Conference and workshop publications
  - Senior honor’s thesis

- Work on a substantial project and an actual real world problem.

- Get early exposure to AI, Robotics and Software Engineering (in practice).
Syllabus

Let's take a look at the handout
Things to do before next week

- Make sure you have a CS account.
  - [https://udb.cs.utexas.edu/amut/acut/](https://udb.cs.utexas.edu/amut/acut/)

- By the end of this week, get yourself added to the mailing list: cs378-spr12-announce@utlists.utexas.edu.

- If you have any questions you can mail me directly (piyushk@cs.utexas.edu) or the entire teaching staff (cs378-spr12-submit@utlists.utexas.edu)

- Reading Assignment 1 (due Tuesday 10 PM) on class webpage: [http://z.cs.utexas.edu/users/piyushk/doku.php/courses/spring12](http://z.cs.utexas.edu/users/piyushk/doku.php/courses/spring12)
How to read a paper

● Pass 1
  ○ Read abstract, introduction and results/conclusion
  ○ Figure out what paper is about
  ○ Don't dive into the paper without getting the overall idea!

● Pass 2 (Required for understanding)
  ○ Read the paper in depth
  ○ Be critical and inquisitive
  ○ Jot down notes/questions on the margins - mark everything you did not understand
  ○ These points will help you discuss the paper
  ○ Not necessary to understand all technical details
How to read a paper

- Pass 3 (Required for implementation)
  - If necessary, try and understand every technical detail
  - Pull resources from the internet
  - Study the related works and citations

- I have placed a couple of useful links on reading a research paper on the class webpage.