ROS after CS1?

6 robots, 6 quadcopters, 60 students, and ROS
An experiment *underway*...

1-unit lab
(one is required)

5 sections of 12
4 with CS1 or CS2
1 of seniors

6 Creates + 6 Kinects +
6 netbooks + 6 drones
\approx \$6k

**Goal:** to increase students' computational sophistication

*not* to teach ROS or robotics
N/A?

all situations are different...
### The labs

**Weekly, 2-3 hrs.; 1 hr. for write-up; no other work**

<table>
<thead>
<tr>
<th>Week</th>
<th>Lab challenge</th>
<th>New hardware</th>
<th>CS Topics Emphasized</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Line-following</td>
<td>creates</td>
<td>command-line &amp; Python</td>
</tr>
<tr>
<td>2</td>
<td>Color segmentation</td>
<td>Kinects: RGB</td>
<td>event-handling</td>
</tr>
<tr>
<td>3</td>
<td>Finite-state control</td>
<td>-</td>
<td>finite-state machines</td>
</tr>
<tr>
<td>4</td>
<td>Visual servoing</td>
<td>-</td>
<td>understanding vs. implementing</td>
</tr>
<tr>
<td>5</td>
<td>Wall-angle estimation</td>
<td>Kinects: Depth</td>
<td>proportional control</td>
</tr>
<tr>
<td>6</td>
<td>Robot minion!</td>
<td>-</td>
<td>larger-scale integration</td>
</tr>
<tr>
<td>7</td>
<td>Wall-following</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Gestural control</td>
<td>quadcopters</td>
<td>GUI/visualization</td>
</tr>
<tr>
<td>10</td>
<td>Autonomous flight</td>
<td>-</td>
<td>defensive programming</td>
</tr>
<tr>
<td>11-14</td>
<td>(Open-ended projects)</td>
<td>all/any</td>
<td>self-defined problems</td>
</tr>
</tbody>
</table>
**The labs**

weekly, 2-3 hrs.; 1 hr. for write-up; *no other work*

<table>
<thead>
<tr>
<th>Week</th>
<th>Lab challenge</th>
<th>New hardware</th>
<th>CS Topics Emphasized</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Line-following</td>
<td>creates</td>
<td>command line &amp; Python</td>
</tr>
<tr>
<td>2</td>
<td>Color segmentation</td>
<td>Kinects: RGB</td>
<td>event-handling</td>
</tr>
<tr>
<td>3</td>
<td>Finite-state control</td>
<td></td>
<td>finite-state machines</td>
</tr>
<tr>
<td>4</td>
<td>Visual servoing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Wall-angle estimation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Wall-following</td>
<td></td>
<td>proportional control</td>
</tr>
<tr>
<td>7</td>
<td>Wall-following</td>
<td></td>
<td>larger-scale integration</td>
</tr>
<tr>
<td>9</td>
<td>Gestural control</td>
<td>quadcopters</td>
<td>GUI/visualization</td>
</tr>
<tr>
<td>10</td>
<td>Autonomous flight</td>
<td></td>
<td>defensive programming</td>
</tr>
<tr>
<td>11-14</td>
<td>(Open-ended projects)</td>
<td>all/any</td>
<td><em>self-defined</em> problems</td>
</tr>
</tbody>
</table>

*Singly, in groups, or as a full course – we hope the modules might be of use.*


Western State College

from visiting last summer...

15 hours later
Western State College

... to outreach this winter
Western State College

... to outreach this winter
Week 1

Line following
Create
command-line and Python

easy
medium
difficult

thanks: Matt Boutell
Week 1

Line following
Create
command-line and Python

made it! (the video is 4x)
Week 2

Color segmentation
Kinect: RGB
keyboard/mouse events

RGB and HSV

original image with overlay
segmented image
Week 2

challenging (!) warm-up activity

... to identify RGB and HSV components
Week 2

Color segmentation

Kinect: RGB

keyboard/mouse events

Creativity?
Recursion!
Week 3

Finite-state control
Kinect: RGB
finite-state machines

+ Kinect images...

+ your own custom state-machine thread

robot sensors...

keybd

F' key

forward

2 seconds

bump!

back
**Task:** to drive the robot to any point clicked in the Kinect's image

Extra! *then get back again...*

Why is this difficult? – and why is it even more difficult to *go left*?
Week 4

Finite-state control

Kinect: RGB

understanding vs. implementation
Week 4

Finite-state control
Kinect: RGB
understanding vs. implementation

The race!
Week 4

Finite-state control

Kinect: RGB

understanding vs. implementation

Race #1: to the tape dot and back

The race: a more polished view...
Week 5

- Wall-angle estimation
- Kinect: Depth
- Understanding vs. implementation

Kinect's range images
What if this pixel at (42,240) reads 1 meter "deep"?
What are that point's coordinates in meters?
Week 5

**Wall-angle estimation**

**Kinect: Depth**

**understanding vs. implementation**

---

**Top-down look at one slice of the Kinect's view (row=240)**

- **distance axis** – these are the Kinect's readings (in meters)
- these black dotted lines are the edges of the Kinect's 57° field of view. The 640-pixel width divides that FOV into equal-sized pieces

- **horizontal axis** – these values need to be computed

---

\[ \text{pixel}_x = 42 \]

\[ \text{pixel}_x = 0 \]

at (42,240) the Kinect reports this depth = 10 m

\[ \text{pixel}_x = 639 \]

57° total field of view
Week 5

Wall-angle estimation

Kinect: Depth

understanding vs. implementation

What is this angle, in depth?
Feedback on labs 3-5...

"Our biggest challenge was figuring out the math behind aligning the robot with the target point."
- Sarah and Steve

"We ran into a lot of problems in doing the actual math which should direct the roomba."
- Alexa, Edward, and Spencer

"We focused most of our attention on getting the math correct."
- Eric and Benson

"Math is amazingly hard.."
- Jessi and Haak
Feedback on labs 3-5...

understanding vs. implementation
Week 6

**Challenge:** lead your robot out the door, into another room, and back

things can always be worse...?
Weeks 6-7

lots of *floor* time
Weeks 6-7

lots of floor time

even when it does work
Week 6

Robot follower
Kinect: Depth
proportional control

success... (at 2x)
Week 7

Wall follower

Kinect: Depth

integration and debugging

moving-wall following (2x)
Week 7

Wall follower

Kinect: Depth

integration and debugging

don't the halls...
"Supermanning" the drone – *from the drone's point of view.*
Week 10

Autonomous flight
quadcopters
defensive programming

Escape!
Weeks 11-14

open-ended projects

self-defined problems

U Penn-like perching?

drone/Create cooperation

cat-and-mouse robots (4x)
Summer!
In conclusion...

Computational confidence-building right after CS1 command-line flaky hardware making state explicit difficult debugging compelling applications!

Self-directed portfolio of results (and failures)

Getting beyond DWIC...

10-20 page write-up with screenshots, videos, descriptions, and reflection
Week 2

challenging (!) warm-up activity

... to identify RGB and HSV components