Parking Lot Occupancy Monitor

Student: Taylor Ullrich
Project Advisor: Dr. Takashi Buma

Motivation

- Due to the limited number of parking spaces at Reamer, it is often difficult to find parking
- The ability to check a website to find when a parking spot is open in Reamer throughout the day

Available Technologies

- Q-Free
  -single space sensors mounted above the end of each parking space
- SmartParking
  -Individual parking space sensors that gather and transmits information for management, payment and compliance monitoring
- Traffiko
  -Intelligent Digital Transport Enforcement System, IDITES Camera

Performance Criteria and Requirements

Conceptual Requirements:

- Recognizes parking in a parking lot of four spaces in a row
- Update a user interface correctly and in time for the information to be relevant
- Is able to operate in different lighting and weather conditions
- accuracy

Specific Requirements:

- Field of View (number of parking spaces)

  -required to monitor the four parking spaces in Reamer Circle
- Refresh Rate

  -the system will update the user interface in less than 1 minute
- User Interface

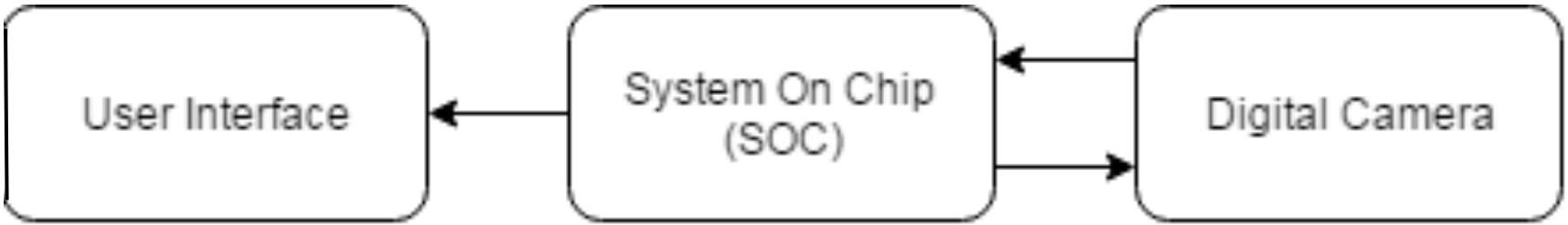
  -graphic representation of parking area

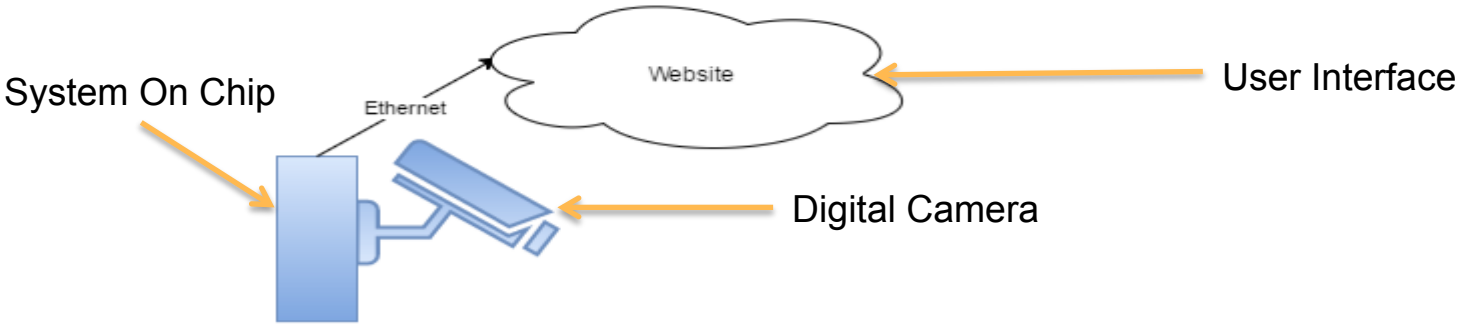
  -easy to build, familiar interface
- Accuracy

  -must be somewhat accurate, accurately decting parking at a

  80 % success rate

Block Diagram





Sensing Options

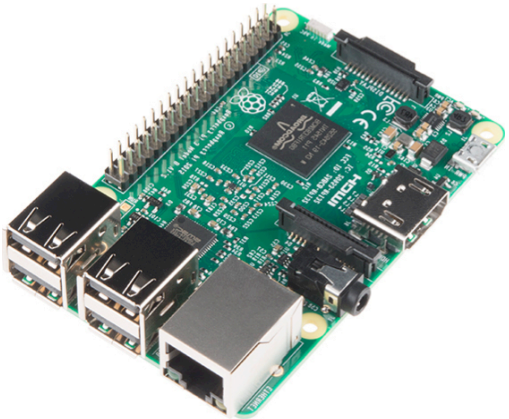
- Infrared Distance Sensors
- Ultrasound Distance Sensors
- RADAR
- Camera

	Sensor	Number Required	Cost ($,$$,$$$)	Difficulty (1-10)
	Infrared	1 or more per space	$	7
	Ultrasound	1 or more per space	$	7
	Radar	1 total	$$$	9
✅	Digital Camera	1 total	$	5

System On Chip

Platform	Cost	Familiarity	Memory	Internet Access	Image Processing Capability
Arduino	$25 (Uno)	Slightly	32 Kbytes	Ethernet Board Add on	Limited by Memory
✅ Raspberry Pi	$40 (Pi 3 B)	Very	1 GB	Built in Ethernet/Wifi	Fully Capable

The Hardware



Raspberry Pi 3 B:

- CSI camera port
- Ethernet and WiFi capable
- Supports Raspbian
- Python

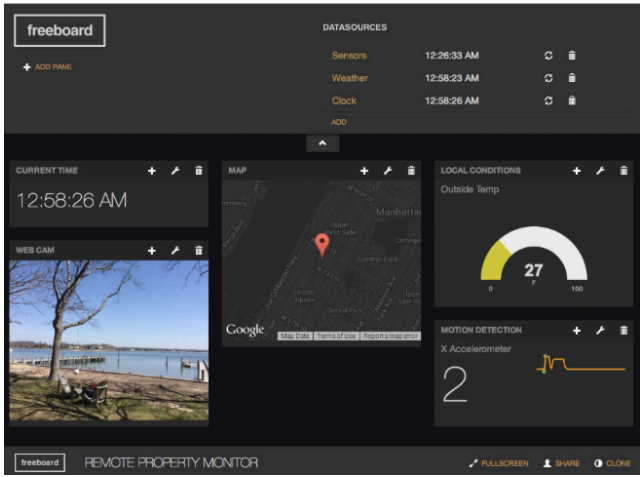
Raspberry Pi Camera Module V2:

- Max Resolution 3280 x 2464
- CSI Cable Connector
- Runs on Raspbian OS
- Field of view of 48.8 degrees

Data Transmission and User Interface

Dweet.io:

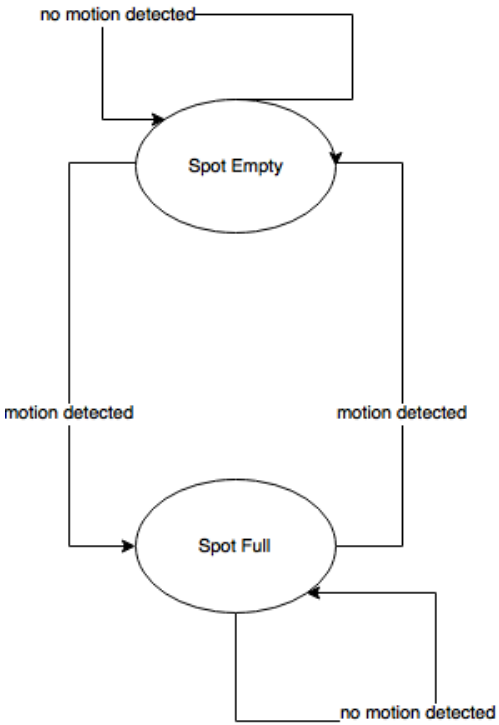
- “Ridiculous Simple Messaging for the Internet of Things”
- Free
- Used in ECE-481 Internet Of Things (IoT)

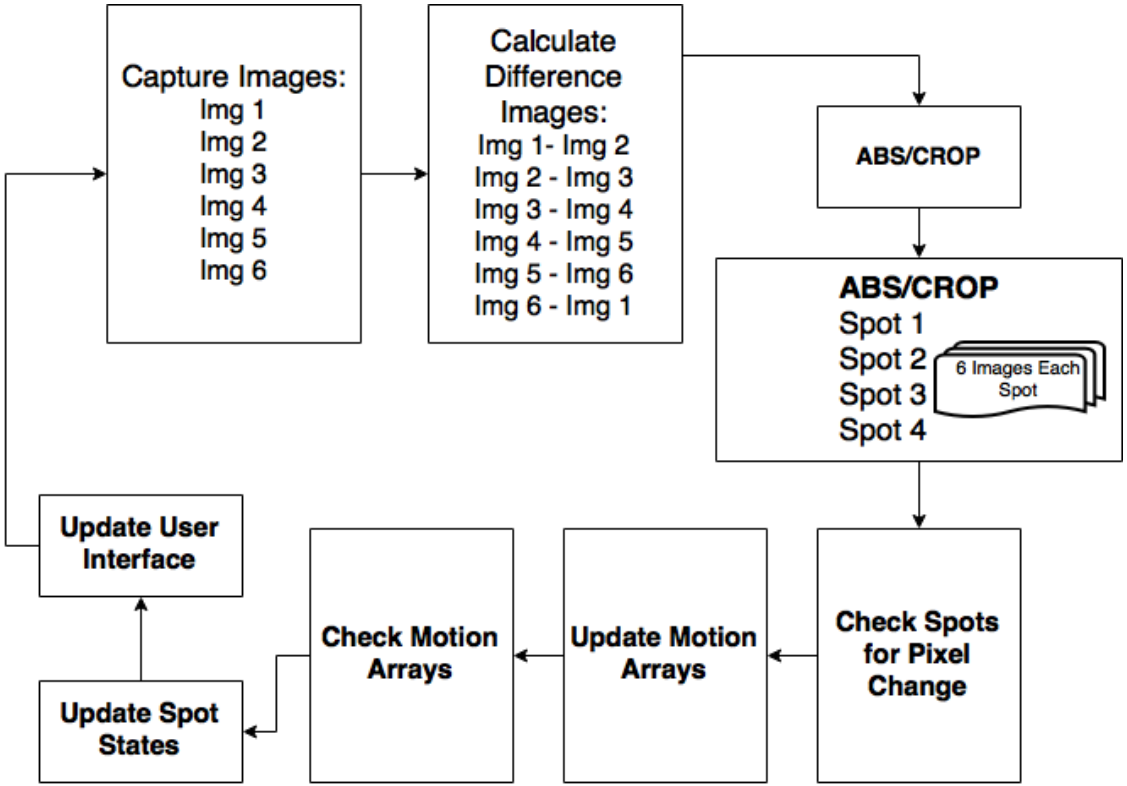


Freeboard.io:

- Used in ECE-481 with Dweet.io
- Free

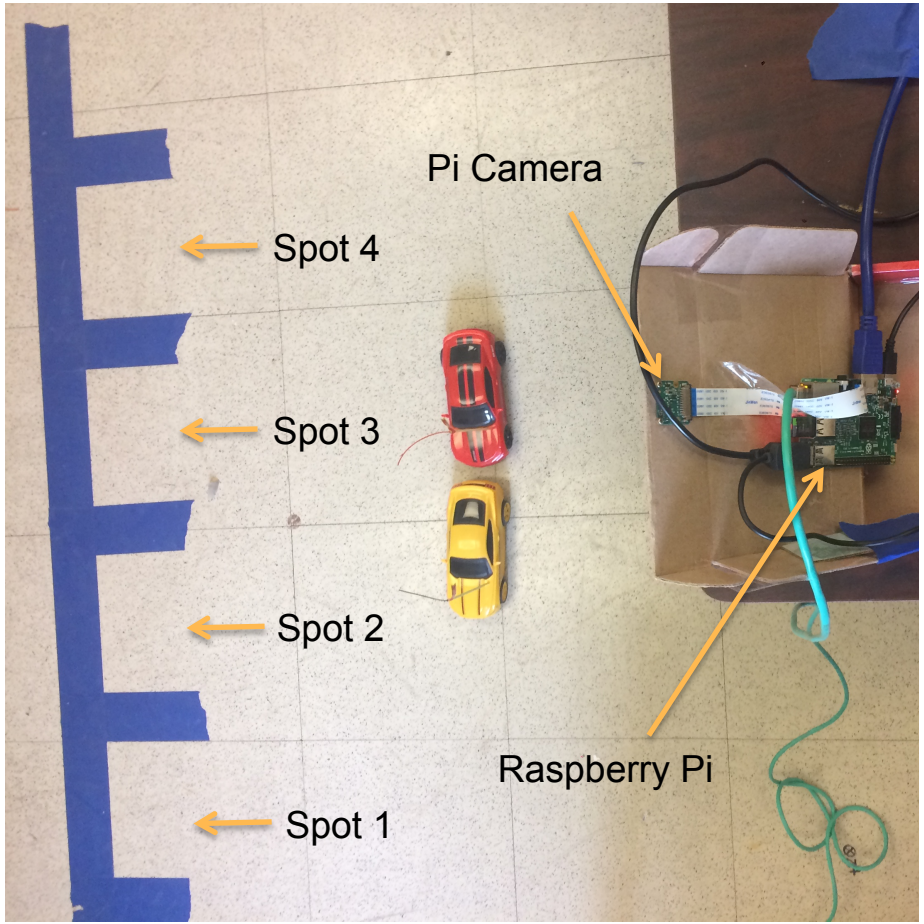
State Machine and System Algorithm Outline



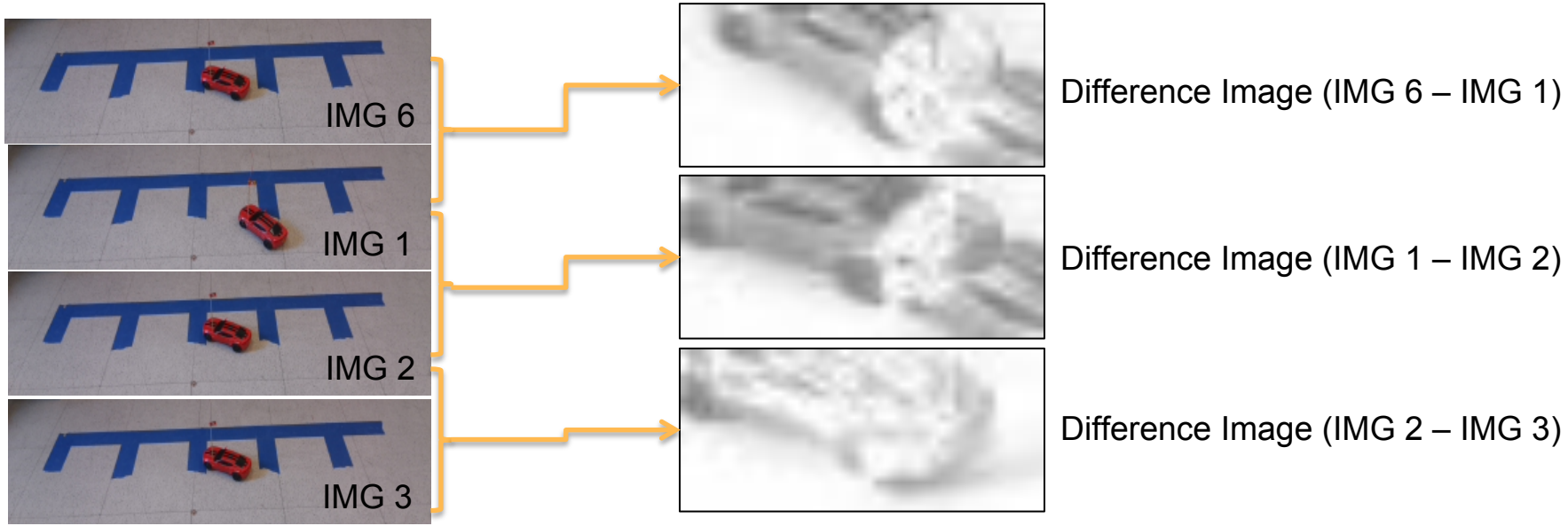


Test Bed

- In order to develop and test this system, the parking lot environment was simulated indoors on a smaller scale
- Remote controlled cars simulate real vehicles
- Raspberry Pi is setup on a desk focusing on the parking lot



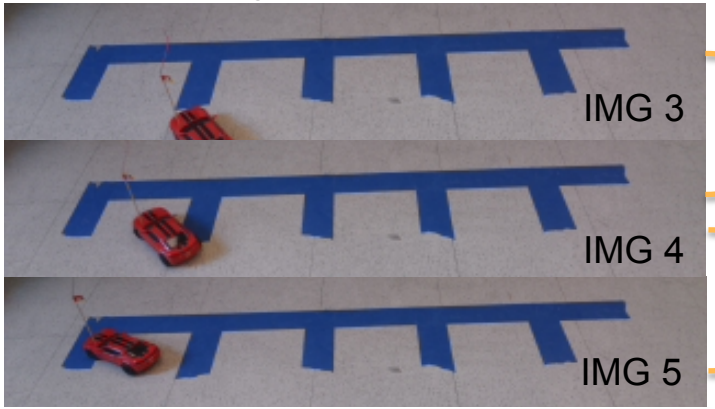
Indoor Environment Results



- Each difference image is created from the two adjacent images above
- Motion of a single car is detectable
- Difference Image (IMG 6 – IMG 1) shows motion from the first image to the last image taken in a loop

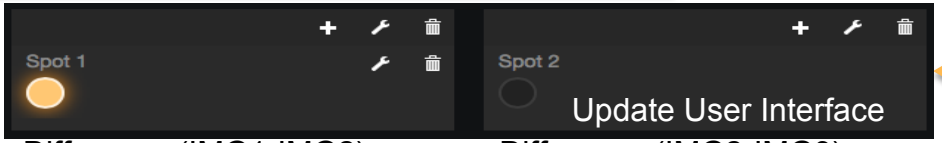
Indoor Environment Results Continued

First Set of Six Images:

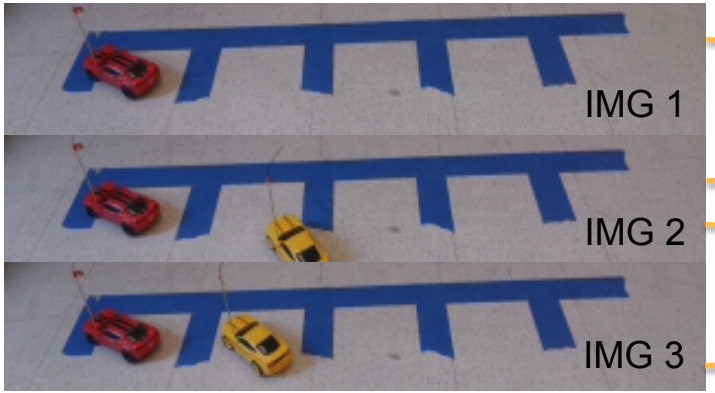


Difference (IMG3-IMG4)

Difference (IMG4-IMG5)

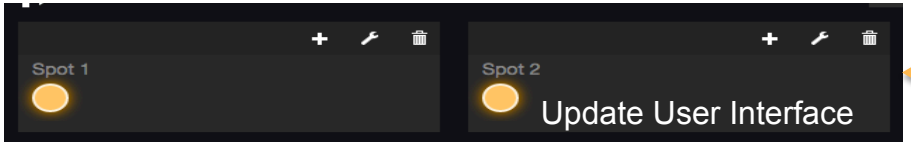


Second Set of Six Images:



Difference (IMG1-IMG2)

Difference (IMG2-IMG3)



Results Continued (OUTSIDE WORK)

Run Time Analysis and Data Usage

- In the current software, 1 second delays are implemented in between each picture
- The processing and capturing of 6 images only takes an average of 10.1075 – 6(^*1 second delays)= 4.1075 seconds
- The data required to store the image files captured, as well as the difference images, totals to about (this amount of memory)

Number of Loop Executions	Time per loop (seconds)
1	10.0092508793
2	9.99955177307
3	10.1097118855
4	10.3290479183
5	10.2096188068
6	10.163695097
7	10.1319139004
8	10.0188269615
9	9.97830796242
10	10.1258809566
AVG (delays)	10.1075
AVG (no delays)	4.1075

Conclusions

- System was successful at identify parking in the indoor environment, and could update the user interface accurately
- Indoor tests show that without proper lighting, detection is not possible
- Processing run time is small enough to not effect overall run time of the system
- Testing is affected by the wind, system would have to be sturdy

By the Requirements:

Field of View:

- System can see all the spaces in Reamer parking lot

Refresh Rate:

- Time required to send messages is approximately _____
- Processing time without delays is 4.1075 seconds

User Interface:

- User interface successfully updates four different parking spaces
- Is easily understood by a user, i.e. obviously labels each spot in the parking lot

Accuracy:

- The implemented system does not meet the 80 % accuracy requirement

Future Work

- System requires much more rigorous tests outdoors, including weather and lighting tests
- Implement software to recognize the parking spaces on its own, eliminating an initialization phase
- Take more image data and store some useful images from previous iterations of the loop

Questions?