Supply and Demand Parking Meter

Project #33

Adam Barbato - barbato2
Nick Johanson - njohans2
TA - Yuchen He

Table of Contents:

Objective

Parking in dense urban areas is a problem where no obvious solution exists. Many approaches attempt to make it easier to find parking spots, however we believe the problem is that when people spend copious amounts of time searching for the spot it doesn't exist. Thus, the real issue is that there isn't enough parking, or perhaps parked vehicles are inadequately distributed. If parking is equivalently priced across an entire city people are incentivized purely on the distance between their parking spot and their destination, so they park as close as possible. This results in particularly high parking densities near popular locations and the inevitable 30 minute quest for a parking spot.

We propose a parking meter which can be implemented with a backend capable of monitoring prices across a city and creating economic incentives for a more favorable parking distribution. This parking meter would monitor the usage of spots it's responsible for and the prices that were applied, then this information would be relayed to another computer which will alter the prices proportionally. Prices would be reduced in areas with low density and increased in places with high density. The algorithm will have a goal density in mind and attempt to reduce the maximum density below that goal.

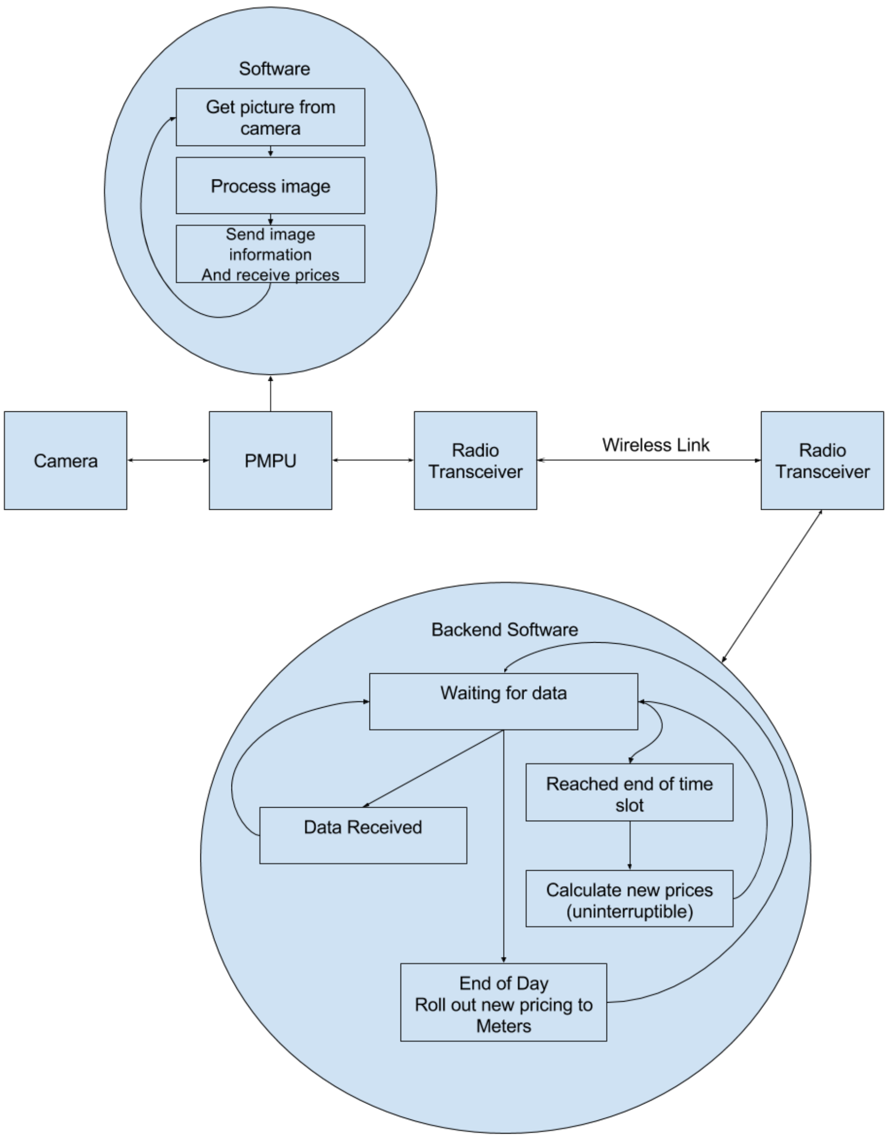
Background

Finding street parking is an annoying problem anybody who was driven into a large city has encountered. However, bigger problems with free, or underpriced, parking was laid out in the book *The High Cost of Free Parking*, in which UCLA Professor Donald Shoup explains that free parking encourages driving, which encourages that more free parking be built, which encourages more driving and so forth. This eventually leads to city maps that are largely empty parking lots. So not only could our project, if implemented, help people plan their trips and daily routines better by offering them choices of parking spots and prices but it could also help cities restore walkability.

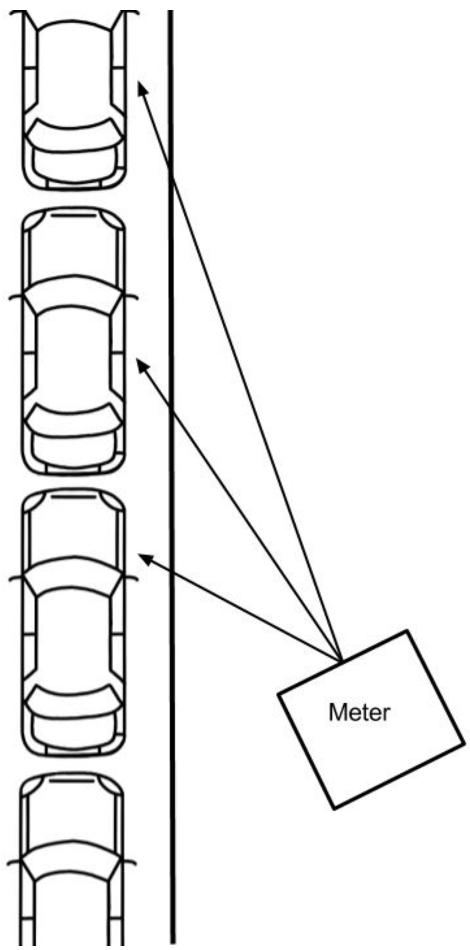
High Level Requirements

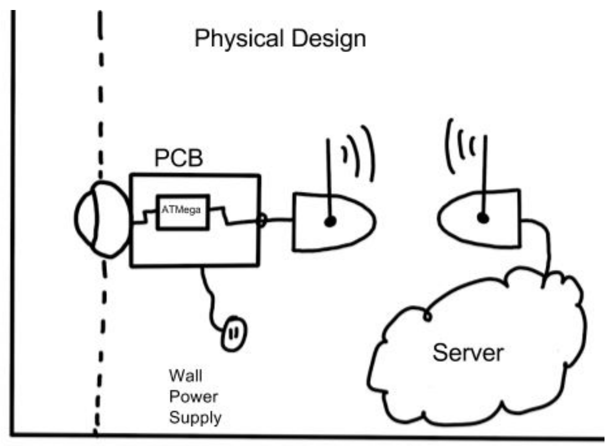
1. Correctly identify the number of cars within the FOV of the camera(s) at 5 minute intervals with at least 90% accuracy.
2. Reduce the maximum density in a simulation to below 85% or the difference between the maximum and minimum density by 15%.
3. Reliably transmit at least 90% of packets between the transceivers.

Block Diagram



Physical Design





Functional Overview

- *Camera*
  - The camera is the piece of the device that collects all the data for the system. It will be mounted on the parking meter at approximately a 60 degree angle from the street as to get the best angle for viewing multiple cars.
  - At least every five minutes it will receive a signal from the processing unit to take a QVGA (320 X 240 pixels) picture of the three parking spots within its view and send the picture back to the processing unit.
- *Parking Meter Processing Unit (PMPU)*
  - The PMPU hardware will consist of a custom PCB powered by an ATMega microprocessor. It will control and receive input from both the camera and the radio module.
  - The software will be a looping program that will run as fast as possible, but is expected to take around, but no longer than, five minutes.
    - It will begin by sending the camera the signal to take a picture and will receive the QVGA picture and store it in memory. It will then move into the processing stage where it will take the new picture, split the picture into three sections, one for where each parking spot would be, and then compares the new picture to a pre-stored one in memory with no cars parked to see if any cars are parked within its space. Once that information has been acquired, the program will timestamp it and initiate contact with the radio unit and send the information to the backend. It will then wait for confirmation from the backend and see if it has any price update information to send before beginning the loop again.
- *Radio Unit*
  - The radio unit actually consists of two physically separate radios, one connected to the PMPU and one connected to an internet connected computer that’s away from the street.
  - The radio unit is responsible for providing a wireless connection between the two parts it's connected and allows the PMPU to operate without an internet connection. This is most likely be implemented with an Xbee radio solution that uses proprietary radio software on a 900MHz channel and should be able to deliver adequate bandwidth.
- *Backend Software*
  - Each day will be divided into four, six hour time segments and an array of prices will be stored for each time slot. Then over the course of a day the average parking density will be gathered from parking meters. The backend algorithm idles until it receives data, which it will then record. Once the end of a time slot is reached, the pricing adjustment will be calculated for that time slot. Then at the end of the day the pricing adjustments will be rolled out to the parking meters.
  - For testing purposes we will use a goal density of 85% and set up a simulation where the maximum density exceeds that. This simulation will consist of a grid with 400 (20x20) tiles with 100 parking spots each. Then 30,000 (75% of

available spots) bot personalities will attempt to park in a set location with a price they are willing to pay and a maximum distance they are willing to park from the desired location. After the distribution is obtained, prices will be adjusted by the algorithm and the bots will attempt to park in the new price adjusted setting. Average parking densities will be sampled at 10, 20, 30, 40, 50, 100 and in intervals of 100 afterwards to 1000. Each personality (with unique maximum price and range) will have a set number of bots with that personality so we can load a personality and decrement a counter instead of storing all the bots in memory, bots will be drawn at random to be allowed to try and park.

Block Requirements

1. *Camera*
    - 1.1. Must be able to take and transmit a QVGA (320 X 240) picture, in clear, sunny conditions, within 30 seconds from when the PMPU instructs it
    - 1.2. Angle of lens must allow three cars to be captured in one picture
    - 1.3. Must run on standard 5V ~1-2A power supply
2. *PMPU*
    - 2.1. Must be able to interface with both the camera and the Radio Unit
    - 2.2. Must be able to run on standard low voltage power (<= 5V DC) ~500mA-2A
    - 2.3. Must be able to identify cars in a picture given a previous picture of a car-less area, if the pictures are taken during the day in clear weather conditions
    - 2.4. Must be able to reliably send and receive data to backend software using Radio Unit
    - 2.5. Must be computationally powerful enough to finish one complete cycle of the software within 5 minutes
    - 2.6. Must have memory space greater than 153600 (=320*240*2) bytes to store both pictures
3. *PMPU Software*
    - 3.1. Must be able to work on two 320x240 RGB
    - 3.2. Must be able to segment the picture into three pictures that will be approximately 160x240, 100x240, and 60x240 pixels
    - 3.3. Must be able to subtract the background of one picture from the other while attenuating noise by at least 5 dB
    - 3.4. FFT operation must be fairly efficient and have the standard O(N*Log N) runtime
4. *Radio Unit*
    - 4.1. Must be able to interface with both the PMPU and the backend software
    - 4.2. Must be able to transmit data both directions between the parking meter and a waiting computer within 100 ft of the parking meter
    - 4.3. Can use a proprietary transmission protocol in the 900MHz band
    - 4.4. Must transmit at least 1 kb/s
5. *Backend Software*

5.1. It should be able to receive information from the transceiver asynchronously, but ensure this doesn't interrupt or alter a calculation occurring at the same time.
5.2. It should change prices proportionally, with larger changes the farther some local density is away from the goal.
5.3. Simulation should reach goal within 100 iterations (about 3 months).
5.4. Simulation should show reduction in maximum parking density to below 85%.

Risk Analysis

The riskiest part to the completion of this project is absolutely the PMPU. Granted, the PMPU is really two parts: hardware and software, but even separate they would be the riskiest parts.

The hardware is a fairly simple PCB design with standard components soldered on, but any hardware that is self designed has chances to fail, either in the design stage, soldering stage or simply because of manufacturing defects in the board or components. This really becomes a problem when the time to reprint another board is on the order of weeks rather than days, which is why the process to begin building this should be started as early as possible.

Additionally, the software on the PMPU is critically important because it is the thing that collects the parking data. Without the parking data, all the backend software that makes the project work will cease to function. Furthermore, the software that collects said data is a Computer Vision algorithm and they are known to be very finicky. We'll have to adjust and tune it just right so it can work in situations where lighting changes (ie. dawn vs midday) or cases where people or animals may be moving in front of the camera when it decides to take a picture. All of these things make the software tricky to implement but critical to the project.

Ethics and Safety

Safety

In regards to safety, due to the mundane nature of the project and its goals the only real safety risk during construction would relate to soldering components to a PCB or gathering data while near moving vehicles. In regards to soldering, standard lab safety guidelines which relate to soldering and general lab safety apply here and will be followed and in regards to street safety, standard practices will also apply and be followed.

During theoretical application of our project in a real world environment there aren't any physical safety requirements. During operation interactions with humans are limited to payments for parking.

Ethics

A large ethical issue that has been brought to our attention is the ability to abuse our system to price gouge or discriminate against impoverished people. Any malicious act like this would go directly against the IEEE Code of Ethics' point 1 about keeping the public's welfare in mind and point 8 by discriminating against those who cannot pay. This is absolutely a possibility with our system, as it would be with any system that can control prices of any kind. However, the system was not designed to work that way and changing the system to accomplish either of these unethical goals would be against its main purpose: to make easy parking more available for everybody.