Detecting a Person Walking on a Driveway Using the Arduino with a BumbleBee Radar

The Samraksh Company, September 2015

In this experiment, an Arduino with the BumbleBee radar was positioned on a long (100 ft) suburban driveway near a house, facing towards the road. The object was to detect displacement motion along the driveway. The results of the experiment are that displacement was detected to a distance of 45 feet from the radar.

There were a variety of plants and trees lining the driveway, and a gentle breeze that was blowing them back and forth. Since the displacement analysis filters out oscillating behavior, these did not interfere with the experiment results.

Setup

The photos below shows the setup of the Arduino-BumbleBee. It is in the plastic tub on the step ladder, connected to a PC via USB which logs the BumbleBee data as analyzed and reported by the Arduino. The red surveyor flags are at 10 ft intervals to a distance of 60 ft.

Facing the street

Facing the house at a distance of about 65 ft.

Running the Experiment

1. The Arduino with BumbleBee was connected to the PC via USB. The Arduino Displacement Detector program was configured to log the sampled I and Q values (adjusted by the running mean) and the results of detection and confirmation analysis.
2. The PC was running the Arduino Displacement Detector client program.
3. Startup. The operator:
    a. Started logging the Arduino Displacement Detector log output.
    b. Started the video camera.
    c. Pressed the Arduino Displacement Detector sync button in a way that was visible to the camera.
4. First walk
    a. Operator moved to the side of the BumbleBee and waited a few seconds.
    b. Operator walked slowly to the 60 ft flag, raising arms at each 10 ft mark, and stopped.
    c. Operator turned around and waited a few seconds.
    d. Operator walked slowly to the Bumblebee, raising arms at each 10 ft mark, and waited a few seconds.
5. Second walk
    a. Operator turned and repeated steps b-d above.

6. Shutdown. The operator:
    a. Pressed the sync button in a way that was visible to the camera.
    b. Stopped the video camera.
    c. Stopped logging.

Analysis

The files related to the analysis are in the experiment zip file archive. This contains:

- A Log.xlsx file: the logged data imported into an Excel spreadsheet.
- A Truth.mp4 file: a ground-truth video showing what actually happened.
- A Validation.rtf file: the results of the detection/confirmation validation.
- DispConf.fig and DispConf.png files: graphs showing displacement and confirmation start and stop time.
- A Meta.xlsx file: metadata about the experiment, combining the ground truth and log events.
- A MatLab folder containing analysis programs.

Using the log file for input, the MatLab program ValidateArduinoDetects.m was used to compare the analysis of cuts, displacement detection and displacement confirmation against those values as computed by the program, using different algorithms than those used on the Arduino. This resulted in the .Validation.rtf file. (The program on the Arduino is constrained by speed and efficiency; the MatLab program, not having these constraints, uses more direct but less efficient algorithms.)

The MatLab program PlotArduinoDetectConfirm.m was run against the log file to generate the DispConf graph files showing the times (relative to the start of the log file) when displacements were detected and when confirmation occurred.

Using the log file, the ground-truth video and the displacement-confirmation graphs, the metadata spreadsheet was populated.

1. Calculate the time difference between the log and the camera. The clocks are not closely synchronized so this lets us calculate the offset between log and camera events. The following steps give the offset, which is further adjusted in recognition that the displacement noted in one second actually applies to the previous second.
    a. Each press of the sync button generates a special entry tagged #s in the log file. The time for the first occurrence of this was entered into the metadata sheet along with the beginning time of logging. This gives the offset of the sync event.
    b. Looking at the ground-truth video, the offset time of the first sync press was entered into the metadata sheet along with the wall-clock time when the video started.
2. The walk start/stop times and distances were entered into the sheet, using log time. The offset from step 1 was used to give the corresponding camera time.
3. Each displacement begin and each end was entered into the sheet, using camera time, with the offset used to give the corresponding log time.

4. Each displacement event (begin or end) was assigned a distance from the BumbleBee based on the position of the operator at the time of the event as shown in the video. The red surveyor flags along with the operator arm raises were used to estimate the distance.

The last step gives the range values mentioned in the introduction.