Design of an Object Detection Algorithm for Use on a Two-Track Robot

Project Members:
Christopher McGirr
Yulric Sequeira
Supervisor: Professor Joseph Vybihal

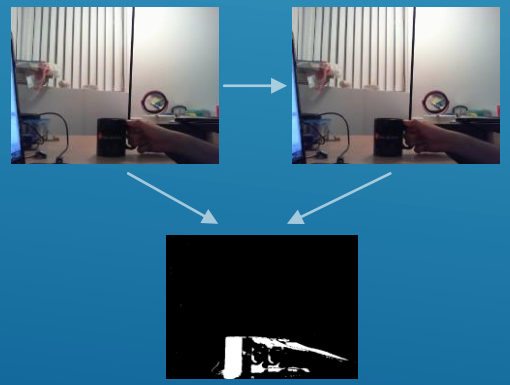
Project Goal

To implement an object detection algorithm using a USB Webcam on a robot in order for the robot to avoid moving objects, such as pedestrians walking by.

Theory

By comparing two consecutive frames we can determine the difference between the frames and so detect if motion has occurred within a certain region of the image. Assumptions are that the observer is stationary between frame capture and that the lighting has not changed.

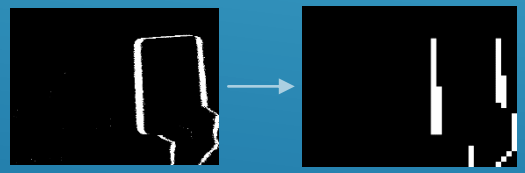
Difference Filter



The Image Difference Filter compares two pixels in each image and checks to see if the pixel intensity has changed. If the change is greater than a threshold. We set that pixel as an area of motion

Full Detection Algorithm

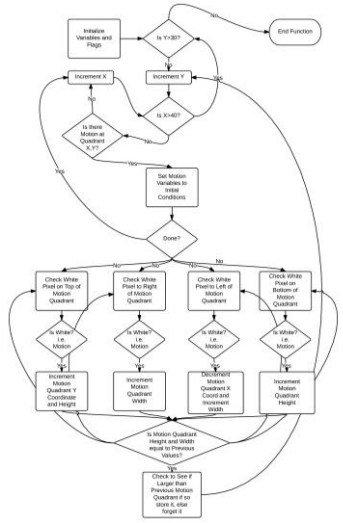
Optimized Difference Filter



The optimized difference filter searches the difference of an image in 16x16 quadrants. If more than half the pixels in the quadrant have different intensities, then that quadrant will be set high, i.e. white to indicate an area of motion. Thus a 640*480 image is reduced to a image of 40*30 quadrants. Reducing the number of points the motion algorithm will have to search. To reduce the amount of image processing the robot will have to accomplish only 25% of the image's pixels are searched. Reducing amount of pixel comparisons from 307200 to 78600. This also helps to reduce noise that is present in the difference of the image.

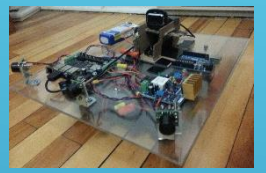
Motion Detection Algorithm

The motion detection algorithm searches the array of 40*30 quadrants outputted from the difference filter and uses a path finding algorithm to encapsulate areas of motion into regions. These regions are then combined into one large region to simplify tracking.



Example of Motion Regions being Combined into a large region for simpler processing

The Motion Region is then checked to see in which area it falls with respect to the robot. The left, right or center of the Robot's Field of View. This information then can be used by the robot to avoid a moving object in front of it.



Test Results

The algorithm was run on board the robot in a variety of configurations. In full image search mode we see the difference filter taking much longer than the motion detection. And so the amount of pixels searched was reduced in order to decrease the computation time.

Algorithm	Computation Time (seconds)
Optimized Difference Filter on Robot	0.96
Motion Detection on Robot	0.000001
Full Difference and Motion Detection on PC	0.02-0.04

Difference Image Search Percentage of 640x480	Amount of Pixels Searched	Computation Time (seconds)
100%	307200	0.96
50%	153600	0.62
25%	76800	0.45

Bluetooth Image Streaming

An attempt was made to offload image processing from the robot. However, Bluetooth did not yield desired speeds to allow for necessary results.

Image Size (WxH)	Raw Image Size in Bytes	Frames Per Second	Bluetooth Data Rate(KB/s)
640*480	307200	0.59	181.25
640*240	153600	1.18	181.25
640*120	76800	2.32	178.18