Low-power motion detection and building control using Computer Vision

Kwabena W. Agyeman
Department Electrical and Computer Engineering, ECE
Carnegie Institute of Technology, CIT
Carnegie Mellon University, Pittsburgh, USA
kwa@andrew.cmu.edu
Low Power Motion Detection Project Webpage

Current motion detectors that control lighting and building HVAC systems use passive infrared (PIR) sensors to detect the presence of a person in a room. However, PIR sensors cannot see small amounts of motion in a room and will turn the lights and HVAC system off on the occupants of a room who are sitting and working in the room. To remedy this problem I propose that a low-power computer vision system is used along with a PIR sensor to perform variable frame rate image differencing to detect the presence of people sitting and working in the room.

Low-power, CMUcam4, PIR sensor, inexpensive

I. Introduction

Making systems smarter and more energy efficient is one of the many problems faced by our world today. Dumb building lighting systems and HVAC systems consume tremendous amounts of energy while performing unnecessary work by illuminating and cooling or heating unoccupied rooms. Efforts have been made to incorporate motion detector systems into the lighting and HVAC control loop that turn the lights and HVAC system on when motion is detected, and that turn the lights and HVAC system off after motion has not been detected for a long period of time.

These motion detection systems rely on inexpensive passive infrared (PIR) sensors to detect motion within a room. PIR sensors work by looking for large changes in the passive infrared light emitted by objects in the room. Different objects at different temperatures in the room emit different amounts of infrared light. PIR sensors are used in motion detection systems to detect when the amount of passive infrared light in a room changes when an object enters or exits the PIR sensors field of view.

Thus, when a person, who is at a different temperature than the room, enters or exits the PIR sensors field of view it detects that the scene has changed. However, the PIR sensor cannot detect if a person entered or exited the room. The PIR sensor only detects if something in its field of view changed. Additionally, PIR sensors cannot detect small amounts of motion within their field of view because small amounts of motion do not drastically change the amount of passive infrared light in the room.

Because of the limitations of PIR sensors, PIR based motion detectors cannot sense the presence of, for example, an office worker sitting at his or her desk who is in the line of sight of the PIR sensor. The officer worker will have to get up and move around to re-trigger the sensor to keep the lights and HVAC system on. Additionally, when the office worker leaves his or her room, the PIR sensor will not be able to detect the absence of the officer worker's presence and will instead turn the lights off based on a countdown timer from the last time the PIR sensor was triggered. This means that the lights and HVAC system will continue to run for a long while in the absence of the office worker.

To solve these problems I propose that a low-power and inexpensive computer vision system is used along with the PIR sensor to perform variable frame rate image differencing to detect the presence of the office worker sitting in the room when the PIR sensor cannot. Additionally, the low power and inexpensive computer vision system could also possibly be used to count the number of occupants in a room to control the lighting and HVAC system more precisely.

II. Implementation

The proposed motion detection system will consist of a PIR sensor and the CMUcam4 low-power computer vision system. The CMUcam4 is an embedded system development platform that pairs a microcontroller with a cell phone camera.

The CMUcam4 will use the PIR sensor to detect motion in a room when the CMUcam4 cannot see because the lights are off. When the PIR sensor detects motion, the CMUcam4 will then turn the lights on and wake up from sleeping. The CMUcam4 will then adjust to the lighting in the room and take a picture of the room for reference. After which the CMUcam4 will proceed to go back to sleep to conserve power while leaving the lights on.

After sleeping for a while (the optimal sleep time must be determined empirically) the CMUcam4 will then wake up and take another picture and compare this new picture to the reference picture that was taken previously. The differences between the two pictures will be thresholded using a look up table and the resulting binary bitmap will be stored to memory. Additionally, the CMUcam4 will compute the average of the new image and the reference image and replace the reference image with the average image. Finally, the CMUcam4 will also store the histogram of the differences to memory to adjust the threshold look up table in response to different distributions of the magnitude of differences in an image.

The CMUcam4 will then go back to sleep, wake up again after some amount of time sleeping, and repeat the above process again. The CMUcam4 will detect motion in the room by looking at the binary bitmap of all the blobs in the thresholded pixel differenced image. All binary blobs larger than a certain size and having a certain pixel density will be treated as differences. The presence of large and dense binary blobs will tell the CMUcam4 to leave the lights on in the room.

The CMUcam4 will then go back to sleep for an amount of time inversely related to the amount of motion detected for the current sample period and the previous sample periods. When the CMUcam4 detects a lack of motion over time, it will sleep for shorter periods of time increasing its sample rate. When the CMUcam4 detects motion over time it will sleep for longer periods of time decreasing its sample rate.

Because the CMUcam4 averages images of the room each time it samples, objects that stop moving will slowly disappear into the background of the image. By having a variable sample rate, the CMUcam4 will be able to more quickly react to the lack of the presence of motion in a room to turn the lights off faster by increasing the sample rate causing non-moving objects to fade into the background. More importantly, the CMUcam4 will also able to stay in low power sleep mode for longer to avoid drawing excessive amounts of power when motion is detected.

The reason for using the CMUcam4 or any other low power vision system over a fully-fledged computer for this application is that the CMUcam4 is able to draw about 100 uA sleeping and less than 100 mA while operating. Any sensor that controls the lights for the room should not draw an amount of energy comparable to the load it controls.

III. Milestones and Deliverables

So far, I have completed the necessary interface library to the underlying hardware to drive the camera and control all of the lowest level functions of the system. Additionally, I have also made a rough draft of a simple histogram based motion detector that works about as well as a PIR sensor regularly. Because of time constraints and a lack of a partner, I will not likely be able to complete the bitmap based motion detector code. The reason for this is that the bitmap based motion detector system requires a lot of algorithm space exploration and testing. I was hoping to have a partner to help with this who would have been able to figure out a good algorithm while I was developing the low-level hardware library. However, because I do not have a partner, I do not now have time to try and figure out how to design a super effective bitmap based motion detection algorithm.

Thus, I will finish implementing the histogram based motion detection algorithm and get it running optimally, after which, I will test out the algorithm and record the average power consumption of the CMUcam4 running the algorithm code. I hope to have test results for the system by Thanks Giving so that I can integrate the test results into a final report, poster, and presentation for class.

All the code I have written for this project will be posted online and made available to the public. The low-level library I wrote includes functionality for frame differencing, frame averaging, thresholding, and histograming. The library can perform 256 unique and useful image-processing operations at 30 FPS.

IV. Releated

I am the inventor of the CMUcam4 open source programmable embedded color vision sensor. I have spent two years working on the project so far. I designed the CMUcam4 hardware and software. Additionally, I wrote all the documentation for the CMUcam4 and provided Arduino Interface libraries for it. I also run the CMUcam4 website and provide help support for CMUcam4 users.

For this project, I will have to rewrite the CMUcam4 firmware to make frame differencing possible. By default, the CMUcam4 performs none of the features needed for this project. The default CMUcam4 firmware performs color tracking and calculates basic color statistics among many other features, but not frame differencing due to the requirement of having to store a complete frame of the image in memory. I will be reusing some basic camera initialization code from the CMUcam4 firmware, but beyond that, I will have to rewrite the image processing pipeline code from scratch.

References

[1] C. Stauffer, W.E.L. Grimson, “Adaptive background mixture models for real-time tracking”, published.

[2] M. Piccardi, “Background subtraction techniques: a review”, unpublished.

[3] A. Verdant, A. Dupret, H. Mathias, P. Villar, L. Lacassagne, “Low Power Motion Detection with Low Spatial and Temporal Resolution for CMOS Image Sensor”, published.

[4] K. Agyeman, “CMUcam4: Open Source Programmable Embedded Color Vision Sensor”, unpublished.

Carnegie Mellon University, Carnegie Institute of Technology, Department of Electrical and Computer Engineering, CMUcam