Never-Fail Alarm Clock

Nicholas Eliades

Abstract

The purpose of the project was to create an alarm to be activated on your computer that opens your webcam and detects a waving hand motion to activate the snooze function. A secondary function was abandoned which was to detect if the user laid back down after turning off the alarm. The hand-waving function worked and was detected by finding the local mins and maxes on a graph containing Y vs X on the same axis(time). To use the webcam function, an OpenCV demo was slightly modified to capture the video input to be saved for use in the MATLAB environment. For the alarm, a MATLAB program was used called remindme, which was also modified to implement the detection function.

The Project

To start, the OpenCV program camshiftdemo was modified, despite a lack of knowledge of C++. It was very simple, it captures the webcam input, saves it as an .avi file, with a Divx encoder. This allows MATLAB to then read the webcam input and evaluate whether a wave occurred. Unsurprisingly, the file must be closed for MATLAB to deal with it though, meaning the webcam capture must occur completely before MATLAB can even deal with it. This alone means the program is far from the real-time application that was originally proposed. A minor problem with the camshiftdemo is that it has to be closed by pressing the escape key. Force closing it using the system command often causes problems with the .avi file. The only reliable way to close the demo is therefore to manually close it by pressing the escape key. This is unreasonable for use as a wake-up alarm. Although it could most likely be modified in the C++, a lack of knowledge in that language prevented a better way to do that.

Using the remindme program, which was a featured MATLAB program that uses a simple timer implementation to provide an alarm and snooze ability. However it uses MATLAB’s questdlg() function which uses the uiwait() function which blocks all other code
from running. Therefore, even if the camshiftdemo is opened before the questdlg, it cannot be closed. Even if the demo was opened and the closing function was put on a timer, the timer would not perform the function while the dialog box is up. Therefore the webcam must capture the video and the detect function must be run all before the dialog box is put up. There is therefore no way to turn off the alarm until the wave detection function is finished. A customized uicontrol dialog box was attempted to allow both run at once, but time did not permit and is quite complicated to implement in MATLAB. The solution was simple but unfortunate, a pause() was put in place to put a fixed time for the webcam capture and then the detect function was run before the dialog box was put up.

The method for detecting the hand motion was a simple background subtraction. To add some complexity to the equation, a running average was used instead of just simple frame subtraction.

\[ B_{i+1} = \alpha F_i + (1 - \alpha) B_i \]

After that, the black-white threshold was set to .15 and then the regions were separated. The largest mass and then center the center of mass was taken so that there was a single point to track for the moving mass. Because the center of mass could not be tracked all too accurately it was not used directly as the tracking method.
Therefore a XY vs. time graph was used which showed a periodicity for the hand wave motion. Unfortunately a sine wave could not be fitted to the period nature of the graph. Thus a local min and max was found and the period was tested to make sure it matched over at least three repetitions. A tolerance of about .1 was allowed because the hand wave motion itself is not regular.

In the end if a period wave hand motion is detected a simple true or false is returned, therefore telling the remindme program whether or not to snooze. Then the dialog box would be called if a snooze was not triggered.

The results were very good, a hand wave action was recognized every time, but gave false results occasionally. Also with the method used, a vertical hand wave motion could be detected, although not as accurately. Although, there were various problems with the alarm implementation, the hand wave detection turned out to be very effective. If the minor details were hammered out, this could work extremely well. Also if it was done entirely in C++ in conjunction with the OpenCV program, it could most likely be done in real time. Because it was done in MATLAB the calculation time took surprisingly long. In fact for every one minute of video, it took over three minutes of calculation. This was most likely due to MATLAB’s video reading function. Overall, the end implementation of the alarm system still was very effective, and the potential of adding the back-to-bed detection is still there. In general, the results were surprisingly slow but functional.