ECE 492 - Computer Engineering Design Project

Gesture Detection

Andrew Zhong | Patrick Kuczera | Shahzeb Asif | 2016

Overview

The purpose of this project is to detect the controlled movement of an LED flashlight. A camera is used to capture image frames while the flashlight is being moved. Each frame is processed, using a custom component on the DE2, to find the position of the illuminated point created by the flashlight.

The entire image is reduced to a single coordinate that represents the position of the flashlight. The incoming stream of position information is compared to stored gesture sequences. If a gesture is recognized, the name of the gesture is displayed on the DE2's LCD screen.

Recognized gestures are used as commands to control a music player on the DE2. The music player can be played or paused using the correct gestures.

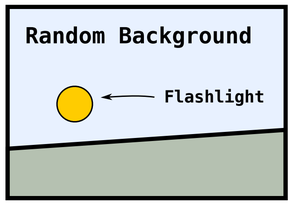
Video output is used to show the LED position tracking in real time.

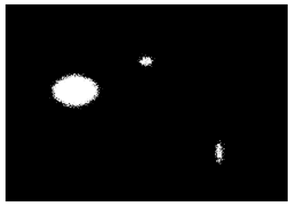
How It Works

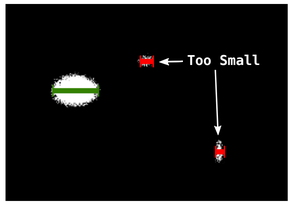
The entire program can be separated into three distinct phases. The first phase is the LED detection phase. The second phase is the gesture recognition and the final phase is the music control.

Phase 1: LED Detection

The first step is to connect a video source to the system. A Nikon S8200 camera is used to feed video into the DE2 video-in port. The video is decoded and analysed by a custom component.





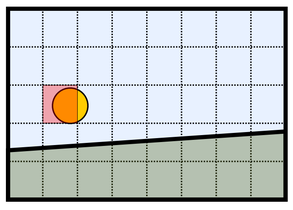


The custom component streams in an entire image frame one pixel at a time. The component first thresholds the image based on the value set by the switches on the DE2. Thresholding will give a binary image which is then an|alysed to find the longest sequence of white pixels. The longest sequence of pixels in an image is assumed to be the location of the LED.

Phase 2: Gesture Recognition

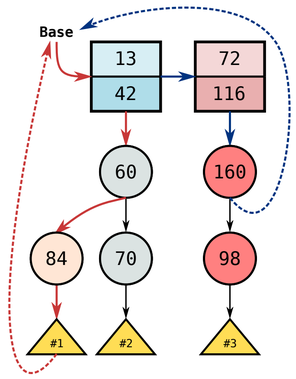
The second phase is gesture recognition. This is done by comparing the incoming sequence of position coordinates to stored sequences.

Gesture recognition is done in steps. The image space is first converted to a grid structure. As position coordinates are streamed in, they are converted to grid values.



These newly calculated grid values are compared against gestures stored in a Trie-like structure.

The data structure is a list that contains lots of trees. Each node in the list represents the first two grid numbers of a gesture sequence. The list nodes are compared to incoming data. If the incoming data is a match for one of the list nodes, the program will move down to the tree and start comparing the tree nodes.



The program will continue down the tree as long as the incoming nodes match. The red path, to the right, shows a successful gesture hit. The blue path shows a unsuccessful gesture.

Phase 3: Music Player

The final phase of our project is using recognized gestures to play music.

A .wav file sampled at 44.1 kHz is stored on an SD card formatted to FAT16. The 3-Wire SPI is used to read off of the SD card and the WM8731 Audio Codec, available on the DE2 board, is used to play our audio. Information read off the SD cared is stored into a buffer that feeds the audio components that actually play the music.

Gestures are used to control the music inside a uCOS task. Gestures are used to pause, play, or increase and decrease the volume.