Color Based Object Tracker

Jordan Kenny

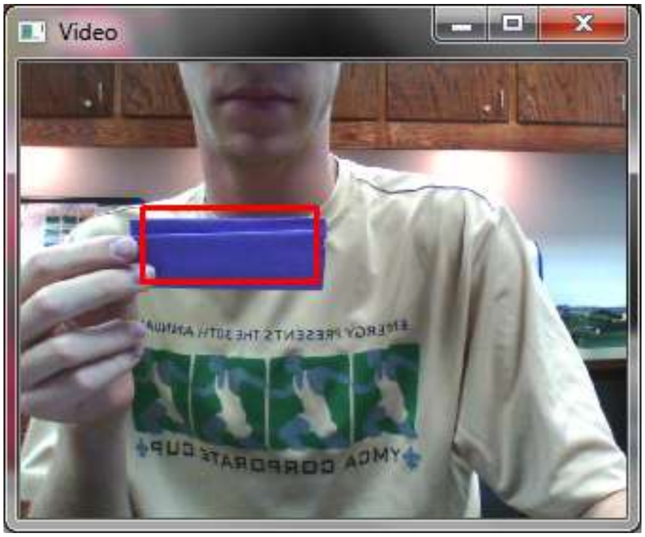
Objective:
The object of my project was to use the OpenCV language to design an object tracker based on the color features of the object.

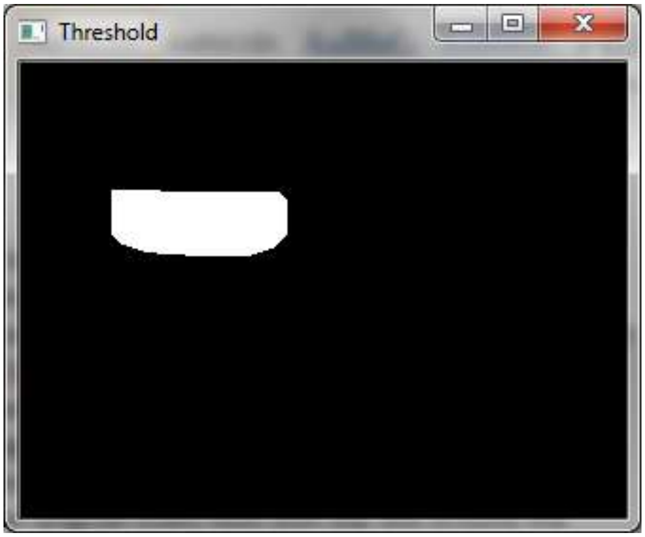
Goals:
My goals in completing this project were to learn and become familiar with the OpenCV language, be able to successfully track an object from my webcam based on the color characteristics of the object, and to implement my working code on an Android smartphone.

Method:
The method that I used to successfully track an object from my webcam was to first get the successive frames from the camera. The frame is originally stored in RGB color space but I wanted to use the hue characteristic of the object to track it so I needed to convert each frame from RGB color space to HSV color space. To determine what object I wanted to track I would just click on the object, so the next step was to interface the mouse to get the coordinates of the pixel that I clicked on. Then I used those coordinates to extract the hue characteristic from the HSV image at the pixel that was clicked. Now that I had the hue characteristic of the object that I wanted I needed to threshold the HSV image to keep only the pixels that are around that hue value. I used a range of about 10 less than to 10 more than the hue value. I also set the range of the saturation and intensity values from 50 to 255. I used these ranges because these characteristics can change slightly as the object moves around the frames. A binary threshold image was created which set all pixels in the HSV range to 1 and all other pixels were 0. Next I would find the contours in the binary image which would be the objects that I am trying to track. Lastly I would find a bounding rectangle around the contours and then draw this rectangle on the original video feed. The final form of my project was the video feed with a red box around the object that I clicked on and the box stays around the object as it moves around the screen.

Results:
I have achieved two out of the three goals that I set for myself. I was able to successfully learn and use the OpenCV programming language effectively. I was already familiar with the C++ programming language and OpenCV is very similar to C++ just with new functions and structures. I used a lot of these functions and structures in my final code, but I also experimented with some of the other functions and structures to find the most effective way to implement my algorithm. So my code was a lot of trial and error to see what functions and what structures would help me to implement my code most efficiently and effectively, which proved to be a big help in learning the language. I was also able to successfully track an object from my webcam based on its color. My code was very effective for the most part but there were some problems that I ran into that I explain below. I was able to make the code very efficient and so my results had a very high frame rate, which is very important in object

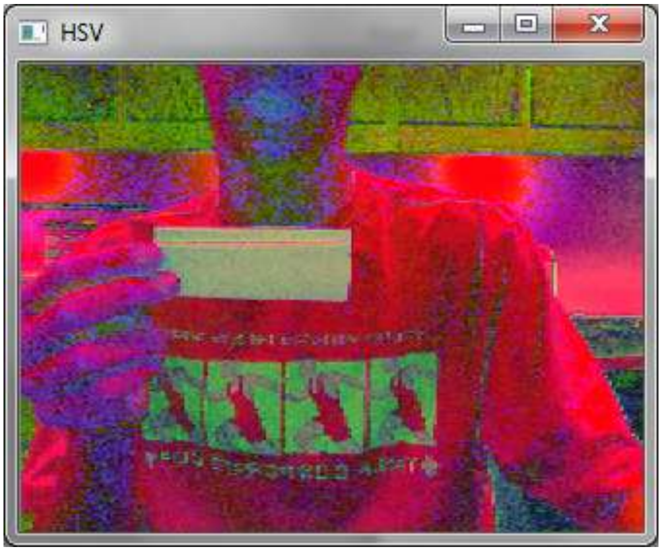
tracking. My code was also very effective in being able to continue tracking the object even if its shape changed or part of the object was hidden or not included in the frame. While I did achieve two of my goals, I was not able to implement my code on an Android smartphone. Mostly this was because of time constraints. It took me quite a while just to get the OpenCV language set up and working correctly on my computer and I ran into many problems when coding that took some time to fix. Not to mention this was my first time using the language so I spent a lot of time becoming familiar with the language. I plan to work on this over the summer and hopefully I will be able to implement it. The picture on the left shown below is a picture of the original video feed with the box around the object I am tracking. The picture on the right is the binary image obtained by the thresholding.

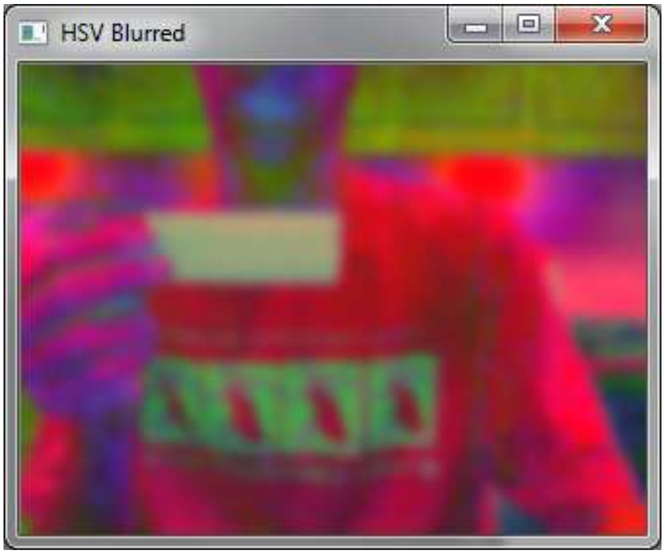




Problems:

As I was working on my project, I ran into many problems with my code. Some of them I was able to fix and others I was not able to do so. The first major problem that I ran into was that the video that was obtained from the webcam contained a lot of noise, which caused a lot of problems for my project. For instance when I clicked on an object, the hue characteristic extracted may not have been the actual characteristic of the object because I may have clicked on a noisy pixel. So the range for thresholding would be for some random hue value and not for the object I was trying to track. Although I wasn’t able to make the image completely noise free, I was able to reduce the noise and make my code work better was to use a Gaussian blur to smooth the image. Actually I found that I have to use the Gaussian blur twice on the image for it to be most effective. I did not smooth it any more than that because I began to lose important characteristics in the image if I did. Even though this helped to reduce noise and make the code more effective, it did not completely remove the noise. The picture on the left below shows the noise that occurs in the HSV image and the image on the right shows the HSV image after the Gaussian blurs. The next problem I faced was that because of the noise and maybe other colors in the image frame, my binary image contained a lot of noise and small objects, so it was finding many contours which I did not want. To solve this problem I opened and then closed the binary image to help get rid of those smaller contours that were not wanted. Other problem that occurred were mostly problems that are inherent in color based object tracking itself. Changes in many factors such as illumination, shading, reflection, etc. can cause the hue characteristic of the object in the image to change. If it became a value that was out of the range of the threshold then it would no longer be tracked. I found that as the object moved around the room, some of these factors changed and therefore the object would be lost. There was not much that I could do to correct this problem because as I said it is inherent in color based tracking. The last problem that occurred was that if there is another object in the frame, that had the same hue characteristic as the object I am tracking, I would also be tracking that object. Obviously when the image is thresheld that object will be in the hue range and therefore kept. I was not able to fix this problem but I could have used some sort of template matching as well to track only the object that I want. The different feature based tracking methods may not work that well separately but if you combine them, the tracker becomes more effective because there are more factors to identify the object. However the more complex the code becomes, the less efficient it becomes which is important to object tracking.





Conclusion:
Overall I am happy with the results of my project and was glad that I chose this project. I always enjoy learning a new programming language and being able to design a complex program with it. I am also pleased that I was able to make my project more complex that my original plan. At first I was just going to track an object of a set color, but I changed it to be able to select the object and color that I wanted to track. Lastly I was pleased with the results of my project. Even though there were some problems, they are not major problems at this point and my code works efficiently and effectively.