An Investigation into Computer Vision Applications

By Patrick Zwierzynski, ECE '19

Introduction

Computer Vision, at a high level, is about developing software programs "to make useful decisions about real physical objects and scenes based on sensed images", much like how a person can interpret their surrounding through sight (Shapiro, 1). It raises an interesting parallel, as most "humans solve many visual problems effortlessly, yet have very little analytical understanding of visual cognition as a process" (Shapiro, 2). Therefore, by taking such an intrinsically human experience and attempting to break it down into a systematic computer program, this becomes a rather difficult task. In fact, computer vision is still very much an active area of research and engineering across a variety of interdisciplinary applications. Some even see it as "an enterprise" which uses "statistical methods" and "models [that are] constructed with the aid of geometry, physics, and learning theory" (Forsyth, Ponce, xvii). While computer vision and artificial intelligence algorithms are still not quite able to match our natural human level of processing visual information, recent developments and research appears to show a "surprising progress" (Shapiro, 3). As more research, development, and production is made into computer vision systems, and as they become more integrated into our daily lives, it also becomes increasingly important to understand how and why these systems are being applied, as well as how they are limited.

Overview

The technical note will provide a brief overview of one technique used in image processing and computer vision using color analysis. After which, the technical note will conclude with a discussion the potential application and limitations of color analysis in relation to Team Manatee's senior design project.

Color Analysis

Many of the interesting computer vision algorithms being developed "require the ability to track moving objects in real time" (Kravtchenko, 1). In fact, "color has been widely used in machine based vision systems", offering "several significant advantages" over other image processing techniques, such as analyzing "geometric cues" or "gray scale intensity" (Kravtchenko, 3).

More specifically, color filtering can be a simple means of filtering the necessary information from an image to track or detect an object based on its color. To better explain this effect, I ran the 'range-detector' script from Adrian Rosebrock's 'imultils' library (link to GitHub in references). This script is especially useful when performing different kinds pre-processing on an image before inputting it into a computer vision model. To demonstrate the effect of color filtering, take the following image seen in Figure 1.

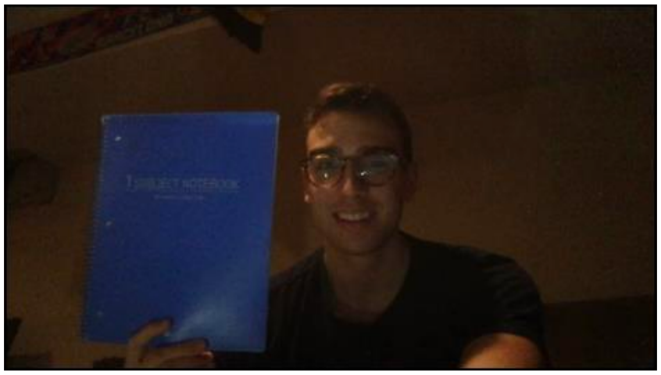

Figure 1. Colored Image Pre-Filter

In such a scenario, it makes sense to use a color filter as the notebook is the only blue object in the photo. As such, we can assign the color filter to only accept a certain range of blue values from an colored pixel reading, and display it in the final binary (black or white) image. An example of this can be seen in the image below:

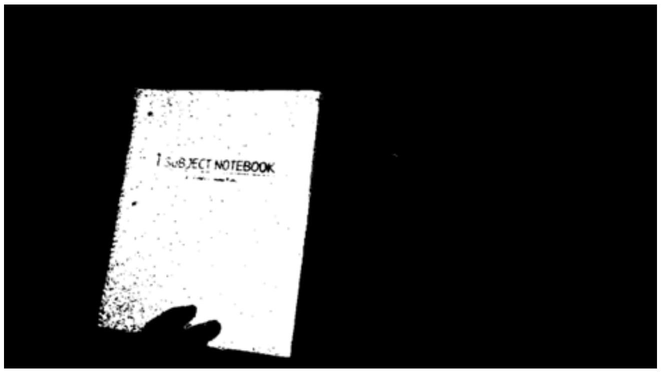

Figure 2. Black & White Image Post-Filter

The color thresholds here are chosen well enough that they are even able to isolate the blue notebook from the thin white font on the book in a low lit image.

Color Analysis Limitations

It is also worth noting that while the color thresholds do well to isolate objects in a particular RGB threshold, the filtered image is still sensitive to noise in the camera readings. This is best seen in the lower left corner of the binary image, where a significant number of the pixels on the notebook still appear black. Some further image processing techniques, including erosions, dilations, and closings can be applied to smoothen out the signal of the image (Morphological Transformations). In addition to this, the use of a color filter for object detection and tracking is limited to the environment in which the thresholds are set. For example, in a better lit environment, the notebook's shade of blue will appear much lighter when captured by a camera lens, and can be potentially filtered out with the currently set thresholds. In such cases, while a "wider" blue threshold can be set, it also makes the system more error prone to the shades of blue we wish to filter out.

Relevance to C.A.M.P.

Possible Application

The color analysis methods described earlier showed initial potential for being applied in Team Manatee's Computer Assisted Musical Performance project. More specifically, our team had initially planned on applying these methods in the drum implementation. This color analysis and tracking would have essentially been used to map, where in 2-D space the drumsticks would strike a part of the drum kit. By coloring the ends of the drumsticks to say, the color green, we can create a computer vision model using the above filtering technique to return the coordinates of the green points, thus allowing us to analyze which parts of the drum kit are being struck. Alternatively, an additional option exists which involves attaching infrared LEDs to the tips of the drumsticks and use a computer vision model to track those in a similar manner to color tracking.

Limitations

While the above methods could work in theory, it is important to note that the 'range-detector' script being used in conjunction with my webcam has a slow frame rate in processing. In a relatively low-cost system, making a computer vision model with a high enough frame rate that allows for it to keep up with a drummers fast swinging arms proved very difficult. It is because of this reason that we were unable to implement a computer vision model that satisfied our usability standards. Instead, our team pivoted towards solely using accelerometer sensor readings to detect when striking the drum had occurred. The part of the drum kit that would be struck is then based off of striking patterns as opposed to trying to find a spatial coordinate describing where the drum had hit.

Conclusion

Computer Vision research and development is still ongoing and necessary if we are to truly integrate its technologies in our everyday lives. While this paper only briefly discussed one specific image processing and computer vision technique in color analysis, we can still see how its application does not come without limitations. It is important to keep those limitations in mind as we move forward with adapting computer vision models, as these limitations can result in consequences that extend far beyond our senior design project. This is especially apparent in recent years with self-driving car development. For example, "in May [of 2016], a Tesla 'autopilot' enthusiast in Florida became the first known fatality in a self-driving car" (Kaplan). Although researchers seem to agree that this accident "was not a failure of computer vision", Tesla, in response to the accident, felt it was necessary to further "fine-tune its radar sensors to more accurately detect road hazards, and rely less on computer vision" (Lohr). Furthermore, some scientists also believe that "for cars to drive themselves safely, several years of continuous improvement – not an A.I. breakthrough – may well be enough"; This would involve "millions of miles of test driving in varied road and weather conditions", which would allow them to react properly and safely in all kinds of driving environments (Kaplan). While the application of these computer vision models is exciting and on the near-horizon, it is through acknowledging its current limitations, testing accordingly, and iterating on its design, that we are able to achieve the necessary standard to integrate computer vision technologies into our lives.

References

1. Shapiro, Linda G, & Stockman, George C. (2001). *Computer vision.* Upper Saddle River, NJ: Prentice Hall.

2. Kravtchenko, Vladimir. (1999). *Tracking color objects in real time.*

3. jrosebr1. "*jrosebr1/Imutils.*" GitHub, github.com/jrosebr1/imutils/blob/master/bin/range-detector.

4. Morphological Transformations (2018). OpenCV, https://docs.opencv.org/3.4/d9/d61tutorial_py_morphological_ops.html

5. Forsyth, David, & Ponce, Jean. (2003). Computer vision : A modern approach. Upper Saddle River, N.J. ; London: Prentice Hall.

6. Kaplan, J. (2016). *Roads That Work for Self-Driving Cars --- The new technology needs new infrastructure too*. Wall Street Journal, p. C.3.

7. Lohr, S. (2016). *Blind Spots Ahead*. The New York Times, p. D1.