

Abstract

Image processing is a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it. It is a type of signal processing in which input is an image and output may be image or characteristics/features associated with that image.

Image processing basically includes the following three steps:

1. Importing the image via image acquisition tools
2. Analyzing and manipulating the image
3. Output in which result can be altered image and information that is based on image analysis.

In this project we manipulate the image to detect certain colors and track them, as well as, defining the shape of the objects with the ability to track multiple colors and sorting them according to their sizes. These procedures are carried on via C# language with the various capabilities of the AForge library.

Content

(I)	The concept behind the project	1
(II)	Used IDE, Language, and libraries	2
(III)	The application's GUI	2
(IV)	How does the application work?	3
(V)	References	3

(I) The concept behind the project

Every object class has its own special features that helps in classifying the class – for example all circles are round. Object class detection uses these special features. For example, when looking for circles, objects that are at a particular distance from a point (i.e. the center) are sought. Similarly, when looking for squares, objects that are perpendicular at corners and have equal side lengths are needed. A similar approach is used for shape and color identification where shapes are processed with mathematical approaches, whereas colors are calibrated on RGB scale.

(II) Used IDE, Language, Libraries

In this project the IDE used is Visual Studio 2017 with the C# language due to the powerful capabilities of the .NET. The ease of using the Visual Studio IDE along with the object oriented featured in C# made it the best choice to implement the project. AForge Library used to manage the image processing on the images.

(III) The application's GUI

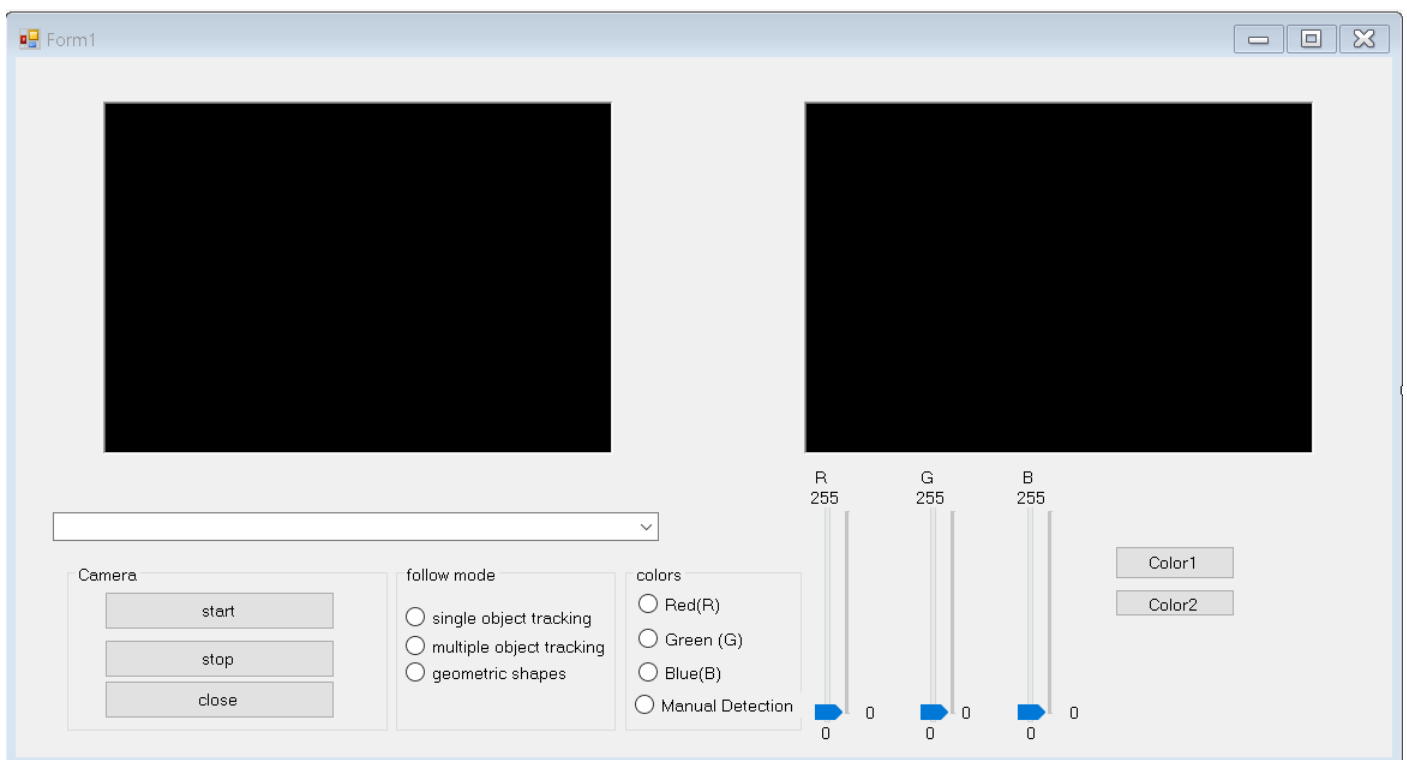


Figure 1 GUI

(IV) How does the application work?

- (1) The combo Box is to choose the video source.

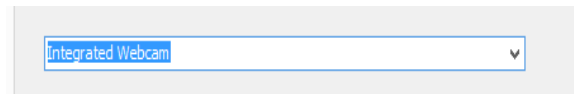


Figure 2 Combo Box

- (2) The left Picture Box shows the video signal from source.

- (3) The right Picture Box show the filtered image.

- (4) These three Buttons in figure 3 are made to start, and stop the transmission or to close the program.

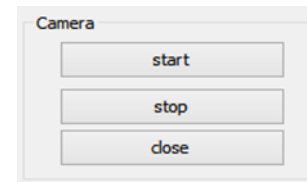


Figure 3 Buttons to control video source

- (5) Figure 4 shows three different radio buttons to choose the required follow mode as follows:

- Single object detection allows you to detect the largest objects only for two different colors.
- Multiple object detection sets borders around every shape with the colors specified and sorting them according to their size.
- Geometric shapes radio button detects geometric shapes and sets the borders to be exactly representing the shape.

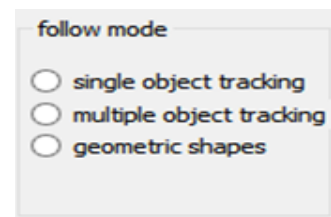


Figure 4 Follow Mode radio buttons

- (6) Figure 5 shows four radio buttons to set the RGB scale to a specific value as follows:

- Red radio button allows you to select the red objects to be show on the filtered image.
- Green radio button allows you to select the Green objects to be show on the filtered image.
- Blue radio button allows you to select the Blue objects to be show on the filtered image.
- Manual detection permits the user either to choose a color from the image only by clicking on the object or by adjusting the scale in figure 6.

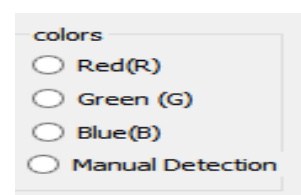


Figure 5 Colors

- (7) Three sets of track bars to adjust the RGB scale for the filtered image.(See figure 6)

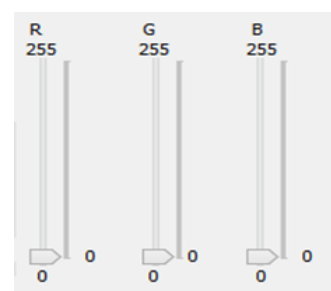


Figure 6 RGB Track Bars

(8) Two buttons to initials two different bitmaps with the required RGB Values.



*Figure 7 Data
Entering
Buttons*

(V) References

- (1) Deitel. Visual C#: How to program. Sixth Edition.
- (2) Castleman, K.R., Digital Image Processing. Second Edition.
- (3) <https://dotnet.microsoft.com/learn/dotnet/what-is-dotnet>
- (4) <http://www.aforgenet.com/framework>