



ISSN Print: 2394-7500
ISSN Online: 2394-5869
Impact Factor: 5.2
IJAR 2015; 1(9): 396-399
www.allresearchjournal.com
Received: 02-06-2015
Accepted: 05-07-2015

Nidhi
Dept. of Computer
Applications, NIT
Kurukshetra, Haryana, India.

Image Processing and Object Detection

Nidhi

Abstract

With the advancement of modern technologies areas related to robotics and computer vision, real time image processing has become a major technology under consideration. So I tried a novel approach for capturing images from the computer web cam in real time environment and process them as we are required. By using *open source computer vision library* (OpenCV for short), an image can be captured on the bases of its *hue, saturation* and *color value* (HSV) range. The basic library functions for image handling and processing are used. Basic library functions are used for loading an image, creating windows to hold image at run time, saving images, and to differentiate images based on their color values. I have also applied function to threshold the output image in order to decrease the distortion in it. While processing, the images are converted from their basic scheme *Red, Green, and Blue* (RGB) to a more suitable one that is HSV.

Keywords: OpenCV, HSV, RGB, threshold

1. Introduction

The research purpose of computer vision aims to simulate the manner of human eyes directly by using computer. Computer vision is such kind of research field which tries to percept and represent the 3D information for world objects. Its essence is to reconstruct the visual aspects of 3D object by analyzing the 2D information extracted accordingly ^[1]. Real life 3D objects are represented by 2D images.

The process of *object detection* analysis the input image and to determine the number, location, size, position of the objects. Object detection is the base for object tracking and object recognition, whose results directly affect the process and accuracy of object recognition. The common object detection method is: color-based approach, detecting objects based on their color values. The method is strong adaptability and robustness, however, the detection speed needs to be improved, because it requires test all possible windows by exhaustive search and has high computational complexity.

2. Introduction to OpenCV

OpenCV is an open source computer vision library. The library is written in C and C++ and runs under Linux, Windows and provides interfaces for Python, Ruby, Matlab and other languages. OpenCV library contains abundant advanced math functions, image processing functions, and computer vision functions that span many areas in vision.

A. Basic Class-

OpenCV 1.0 includes the following five modules ^[2]:

- 1) *CxCore*: Some basic functions (various data types and basic operations, etc.).
- 2) *CV*: Contains image processing and computer vision function (image processing, structure analysis, motion analysis, and object tracking, pattern recognition, and camera calibration).
- 3) *CvAux*: Some experimental functions (View Morphing, Three-dimensional Tracking, PCA, HMM).
- 4) *HighGUI*: Contains user interface GUI and image/video storage and recall
- 5) *CvCam*: Camera interface (After OpenCV 1.0 version, CvCam will be completely removed.).

Correspondence
Nidhi
Dept. of Computer
Applications, NIT
Kurukshetra, Haryana, India.

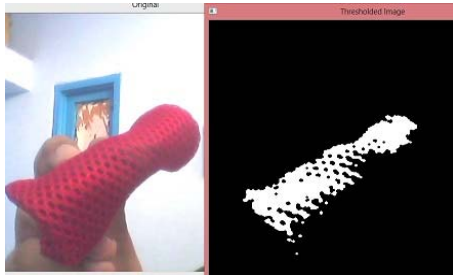


Fig 1: Example of Object Detection

3. Literature Survey

In past few years, the detection of Objects in real time and Image processing has become an active area of research and several new approaches have been proposed.

Several researchers have conducted many studies about Object detection-

1. S.V. Viraktamath, Mukund Katti, Aditya Khatawkar & Pavan Kulkarni has conducted a study of openCV and also have published an IEEE paper for Face Detection and Tracking using OpenCV. Their work is related with converting web cam captured 2D Images and convert them into 3D Images related to human faces by constructing 3D Geometry data outputs [1].
2. Ashish Pant, Arjun Arora, Sunnet Kumar and Prof. R.P. Arora form DIT Dehradun have researched about Image Processing and encrypting an Image in order to transfer safely over the networks.
3. They entitled their work as Sophisticated Image Encryption Using OpenCV [2].
4. Kevinhughes, an elite individual in Opencv area has written a number of blogs containing projects tutorials in this area and steps for installing various softwares [3].
5. Serge Belongie and Jitendra Malik, members of IEEE have done a vast study in the field of Shape Matching and Object Matching Based on their shapes, differentiating two object based on the difference in their shapes.
6. Orlando J. Tobias, and Rui Seara, *Member, IEEE*, have put their great efforts studding the ways and techniques for *Image Segmentation* and *histogram Thresholding*.

4. Description of Tools

In this section the tools and methodology to implement and evaluate face detection and tracking using OpenCV are detailed.

4.1. OpenCV

OpenCV (*Open Source Computer Vision Library*) is a library of programming functions mainly aimed at real time computer vision, developed by Intel and now supported by Willow Garage. It is free for use under the open source BSD license. The library is cross-platform. It focuses mainly on real-time image processing.



Fig 2: OpenCV logo [6]

4.2. Other Software

4.2.1 Mingw for including the GNU assembler, linkers and all windows API libraries (lib, dll). It is an Open source development environment for MS windows applications.

4.2.2 Cmake to manage Build process of the binary makefiles on windows Platform. It is an Open source cross platform software used for the applications that depends on multiple libraries.



Fig 3: Cmake logo

4.2.3 Codeblocks is an Open source IDE for C++ code development.

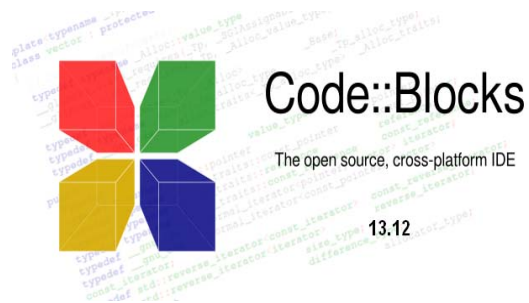


Fig 4: Codeblocks Logo

5. Environment Configuration

Before getting started with the coding part, we have to configure some software to work properly with each other.

Required Softwares-

1. OpenCV
2. MinGW
3. Cmake
4. IDE for C++ development

Prescribed steps are to be followed for configuration process.

Under windows 8.1 and Codeblocks 13.12, the steps that I followed are-

1. Downloading each software required.
2. Extracting OpenCV in "C:/" drive.
3. Adding Mingw to system path.
4. Installing Codeblocks.
5. Generating Binary makefiles using Cmake.
6. Add OpenCV to System paths.
7. Configuring Codeblocks with Mingw and OpenCV.

6. Representation of Images

OpenCV usually captures images in 8-bit, 3 channels, unsigned integer, BGR format.

1. Each image can be considered as combination of three matrices, each one corresponding to a color in RGB color scheme.
2. BGR – BLUE, GREEN, RED.
3. Each with integer value 0-255 (8 bit unsigned)

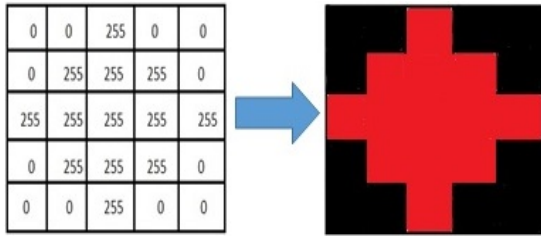


Fig 4: RED matrix example in RGB scheme.

Various Data types to store image in opencv-

1. CV_8U (8 bit unsigned integer)
2. CV_16U (16 bit unsigned integer)
3. CV_16S (16 bit signed integer)
4. CV_32S (32 bit signed integer)
5. CV_32F (32 bit floating point number)
6. CV_64F (64 bit float floating point number)

BUT, HSV color space is the most suitable color space for color based image segmentation. So, in our project, we will convert the color space from BGR to HSV image.

Hue values of basic colors under study-

1. Orange 00-22
2. Yellow 22-38
3. Green 38-75
4. Blue 75-130
5. Violet 130-160
6. Red 160-179

Saturation and Value depend on the lighting condition of the environment.

7. Thresholding the output Image [5]

Typical computer vision applications usually require an *image segmentation-preprocessing* algorithm as a first procedure. At the output of this stage, each object of the image, represented by a set of pixels, is isolated from the rest of the scene. The purpose of this step is that objects and background are separated into non-overlapping sets. Usually, this segmentation process is based on the image gray-level histogram. In that case, the aim is to find a critical value or Threshold. Through this threshold, applied to the whole image, pixels whose gray levels exceed this critical value are assigned to one set and the rest to the other.

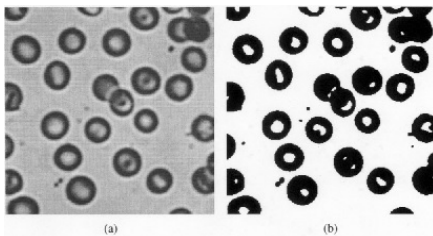


Fig 5: [5] a) Image before Thresholding b) Image after Thresholding

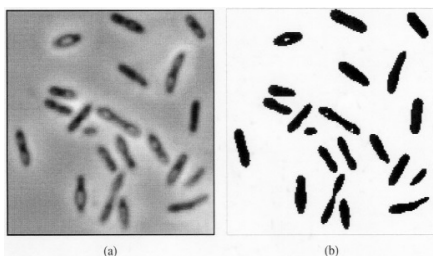


Fig 6: [5] a) Image before Thresholding b) Image after Thresholding

8. Results and Analysis

By choosing the range for Hue, Saturation and Value according to a particular color, we can easily detect any color object.

For example: values used in our project are
 Hue- 170-179 (for Red color)
 Saturation- 150-255
 Value- 60-255

With a RED object-



Fig 7: output of the program showing Object Detection.

Without a RED Object-

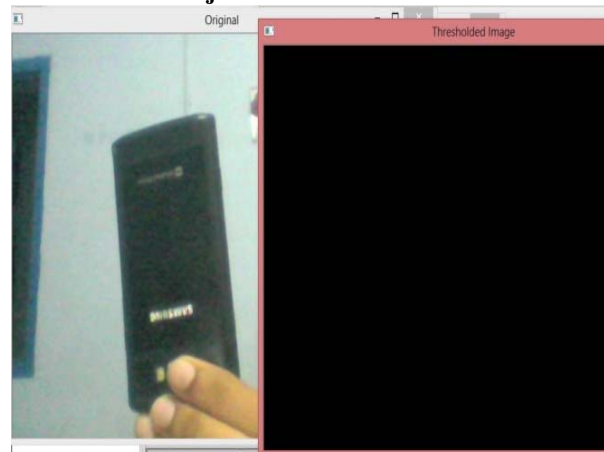


Fig 8: output of the program showing Object Detection.

So detection of any object by the computer web cam is done on the basis on the various color schemes values. [Figure 7] As the program was made just to detect the RED color object, it does so very well and give in output a thresholded image stream on the monitor of the computer.

The program can't detect any other color object, as shown in above [figure 8].

The main problem I was facing while working with this project was as we developed program just for RED object detection, whenever we want to detect any object with different color, we have to make changes, every time in the source code.

The best solution for this problem that I came up with is why not to make our program enough flexible that we can make changes according to the color of the object at *runtime*.

So, I included the *value bares* for HUE, SATURATION, and value range on the top of the program at run time.

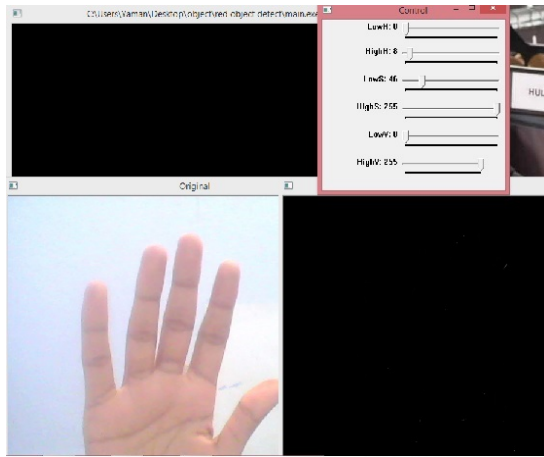


Fig 9: No output when we try to detect human hand

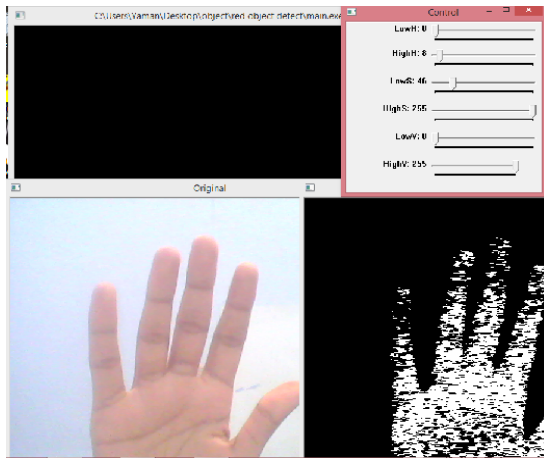


Fig 10: Detecting object with different colors with help of the value bars at runtime.

9. Future Work

Future work will include the Human Face Detection based on the generation of the 3D geometric data from the 2D image input.

Face recognition may also be implemented and we can try to generate a security alert system that will give alerts when an intruder is going to trespass any unauthenticated area based on his/her face recognition. But all authenticated people face imprints data has to be provided to the system in advance. A Virtual mouse controller may also be implemented by time, controlling the movement of mouse pointer on the computer screen virtually just by waving our hands.

Hand Gesture Recognition will also be an area open to work upon, making your computer system to work just by changing your hand positions and gestures.

10. Conclusion

Prototype system for color based object detection is successfully implemented and tested. The test results show that the detection method used in the paper can accurately detect and trace any object in real time. This paper shows the methods of Image processing and detecting an object in it based on its specific color, by using OpenCV real time implementation is possible. Thresholding of the generated image is necessary in order to segment the image pixels and let them free from each other.

Future work includes so many possibilities related to the

hand gestures recognition, virtual mouse controller, and also the face detection and recognition.

11. References

1. Viraktamath SV, Mukund Katti, Aditya Khatawkar, Pavan Kulkarni, "Face Detection and Tracking using OpenCV," The SIJ Transaction on Computer Networks & Communication Engineering (CNCE), 2013, 1(3).
2. Pant A, Arora A, Kumar S, Arora RP. "Sophisticated Image Encryption Using OpenCV," International Journal of Advances Research in Computer Science and Software Engineering, 2012, 2(1).
3. Kevin Hughes – One more robot learn to see (<http://kevinhughes.ca>)
4. Belongie S, Malik J, Puzicha J. "Shape Matching and Object Recognition using shape contexts," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002; 24(4):509-522,
5. Tobias OJ, Seara R. "Image Segmentation by Histogram Thresholding Using Fuzzy Sets," IEEE Transactions on Image Processing, 2002; 11(12):1457-1465.
6. <http://www.opencv.org>