e-Yantra Robotics Competition Plus

(eYRC+ Pilot)

Team ID: eYRC+#2447

Team leader name	Jayant Solanki
College	J.K. Institute of Applied Physics and Technology
e-mail	jayantjnp@gmail.com
Date	December 8, 2014

Scope of the Task (5)

Describe the task assigned.

Teams should write in their own words a description of the task assigned. What is the purpose of such a system? What other kinds of problems can you think of, where solutions to this task can be applied? You can draw some diagrams/figures to illustrate the problem assigned to you.

We implemented basic computer vision techniques using OpenCV, Numpy and Python to predict the ball number which will be shot using striker ball and cue stick on a pool table system.

Figures illustrating the problem statement as a whole

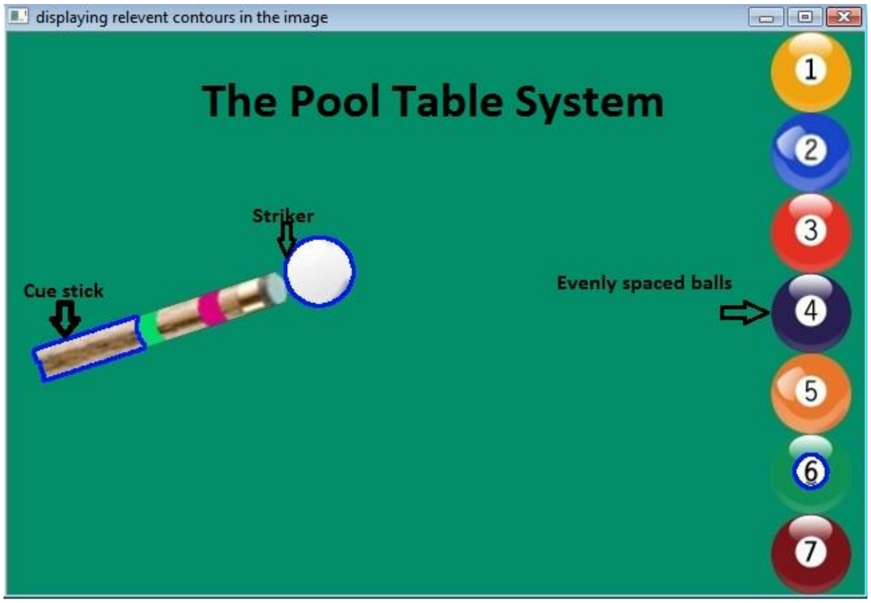
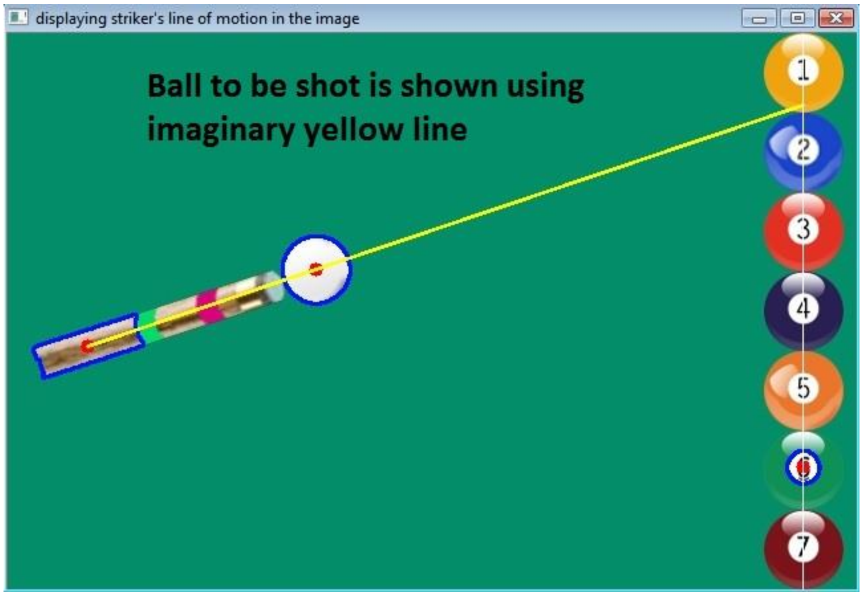


Figure 1

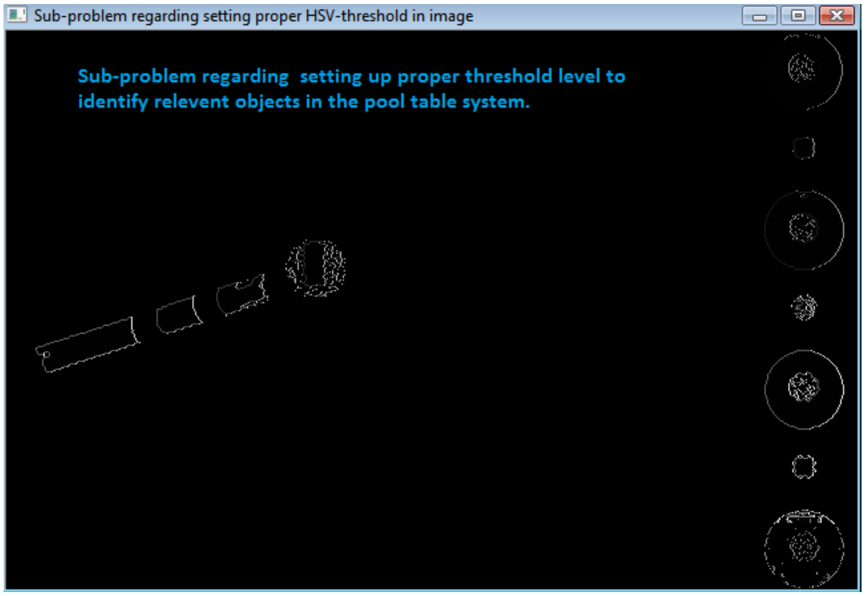


displaying striker's line of motion in the image

Ball to be shot is shown using imaginary yellow line

Figure 2

Figures illustrating the sub-problems.



Sub-problem regarding setting proper HSV-threshold in image

Sub-problem regarding setting up proper threshold level to identify relevent objects in the pool table system.

Figure 3

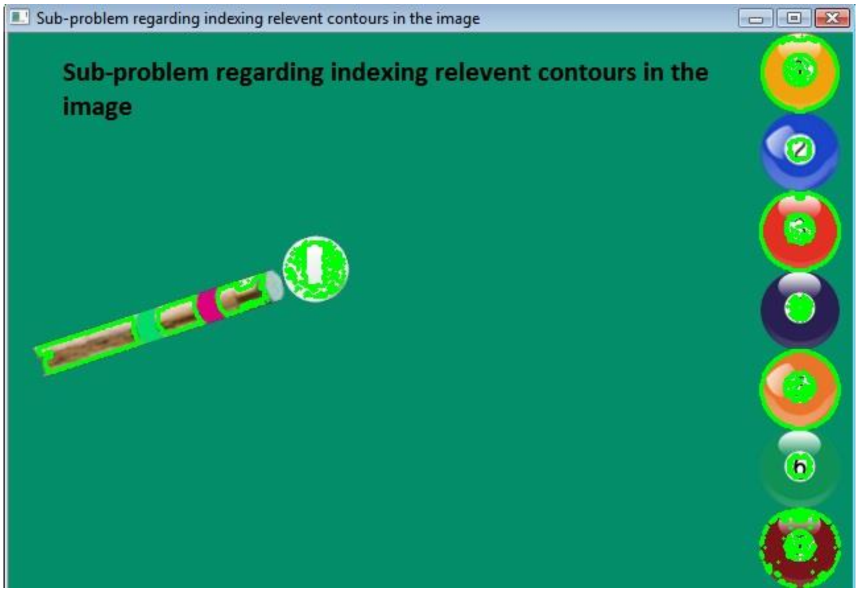

Sub-problem regarding indexing relevent contours in the image
Sub-problem regarding indexing relevent contours in the image

Figure 4

Algorithms we used in such system can be used to analyze Live video/images of a pool game to predict motions and collisions between balls for different angles of the cue with striker. Such algorithms can also be used to enhance viewing experience and taking better decisions during critical events in different sports like football, cricket, tennis and badminton where we are required to track moving objects.

Camera and Image Processing (8)

Write down the answers to the following questions.

1. What is the resolution (size) of the picture taken from your camera?
2. What is the resolution (size) of the test image assigned in the task?
3. What is the use of thresholding an image?
4. What is the use of color masks?

<

Answer format: Bulleted form

1. Answer to question 1

BMP VGA = 640*480 pixels

JPG VGA = 640*480 pixels

JPG 5MP = 2560*1920 pixels

JPG 8MP = 3200*2400 pixels

JPG 10MP = 3648*2736 pixels

JPG 12MP = 4096*3072 pixels

JPG 20MP = 5500*3640 pixels

2. Answer to question 2

640*420 Pixels

3. Answer to question 3

Thresholding provides segmentation of pixels on the basis of the different intensities or colors in the foreground and background regions of an image. It is the simplest way of image segmentation used for isolating interesting objects from the image background and transforming the image into more useful and easier to understand representation. Its main uses are in Traffic cameras for reading car plates, object tracking, MRI/Brain Mapping, astronomy.

4. Answer to question 4

It is a type of thresholding where we mask a particular color that is setting the desired color pixels' value to non-zero and rest others to zero. Masking is used to leave an individual object in the image in color. It is used in image analysis in the fields of bio medics, astronomy, weather forecasting, ocean topography for displaying false color images.

Software used (7)

Write down the answers to the following questions.

1. Write a function in python to open a color image and convert the image into grayscale. You are required to write a function *color_grayscale(filename,g)* which takes two arguments:
    a. filename: a color image (Test color image is in folder “Task1_Practice/test_images”. Pick first image to perform the experiment.)
    b. g: an integer

Output of program should be a grayscale image if g = 1 and a color image otherwise.

```python
def color_grayscale(filename,g):
    '''
    filename-- input color image stored as file(Test color image is in folder
    “Task1_Practice/test_images”. Pick first image to perform the
     experiment.)
    g -- int 0 or 1
    returns img-- grayscale of input image if g=1 else color image
    '''
    #add your code here

    if g==1 : #return  grayscale image if g==1
      img = cv2.cvtColor(filename,cv2.COLOR_BGR2GRAY)
      return(img)
    if g==0 : #return  color image if g==0
      return(img)
```

2. Write a function in python to return only the red portions of the image based on the appropriate HSV range.

```python
def red_threshold(img, hsv_low, hsv_high):
```

```
'''
img-- input color image stored as file (Test color image is in folder
“Task1_Practice/test_images”. Pick first image to perform the experiment.)
hsv_low-- int list for hsv low value eg. [50,200,300]
hsv_high--int list for hsv high value eg. [100,255,255]

returns img--with red part only
'''

MIN = [0,150,200] #lower hsv range for red portions
MAX = [5,255,255] #lower hsv range for red portions

hsv = cv2.cvtColor(img,cv2.COLOR_BGR2HSV) #color image to hsv image

## marking the red portions of the hsv image
hsv_low = np.array(MIN) #converting to OpenCV format
hsv_high = np.array(MAX)
mask = cv2.inRange(hsv, hsv_low, hsv_high) #identifying the red portion
img = cv2.bitwise_and(img, img, mask= mask)
## return the image showing red portions only
return(img)
```