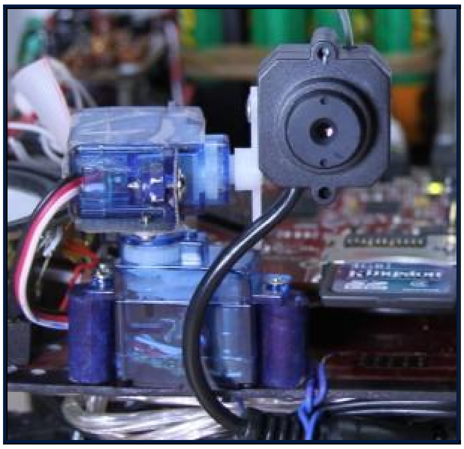
Fast Object Tracking – Robot Computer Vision

Tags: Beagleboard, Computer Vision, Image Processing

I wanted my robot to be able to track object and follow them. The first thing I wanted to do is give the robot the ability to follow an object with its head camera. The head camera is mounted on a pan-tilt servo system, and hence is capable of moving left and right, up and down (as seen in the picture below).



My second object tracking goal was to make the robot chase after an object, much like a dog would chase a ball thrown by his owner. This kind of tracking is quite harder – it would use the head camera tracking from the previous step, and combine it with rest of the robot sensors to follow the object.

How?

To achieve that I'm going to use several basic image processing \ computer vision algorithms. I'm going to use the OpenCV library. OpenCV, as it's name suggests, is an open-source computer-vision library originally developed by Intel. It is cross-platform (I have used it both on a PC and on the ARM based Beagleboard). OpenCV is fairly easy to use if you have basic knowledge in image-processing.

The first object I wanted to track was a plain colored orange ball.

Color based tracking

Filter only the orange color from the image. To do that, I converted the image to the HSV color-space, and then used the cvInRange filter twice to filter the orange colors.

In the first picture below you can see the video steam converted to the HSV color space, and in the second picture you can see the result of the red color filtering (all colors except red were filtered out).

Translator

Bens Translator

Tags

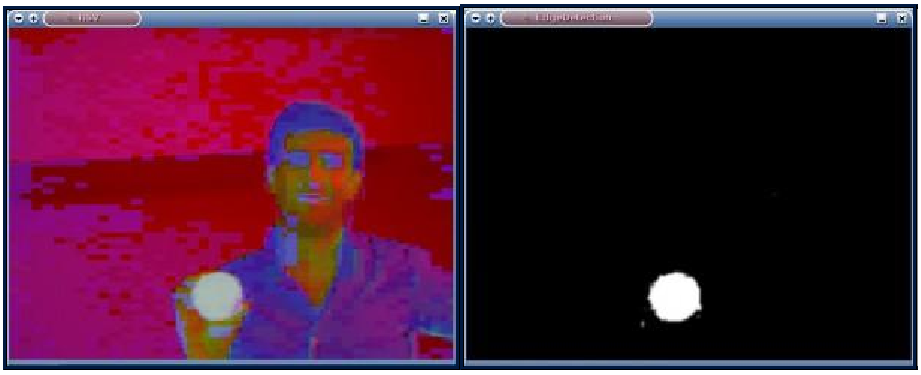
AVR Beagleboard beaglebone CNC Computer Vision DC Motor Debian Digital Signal Processing DIY Electronic Circuit Electronics Embedded Embedded Linux I2C ICSP Image Processing Kernel Linux Machine Microcontroller Motor Control PIC RaspberryPi Robot Robotics RS232 Sensor Servo Motor Software Speech Recognition Stepper Motor USB Wifi

Recent Posts

- Microcontroller In Circuit Serial Programming (ICSP) with Microchip PIC and Atmel AVR
- Fast Object Tracking – Robot Computer Vision
- Linux Robotic Platform – an Intelligent Robot
- Automatic Caller Identifier for Maemo (Nokia Linux OS)
- Building a Custom Debian Kernel for the BeagleBoard

Meta

- Register
- Log in
- Entries RSS
- Comments RSS
- WordPress.org

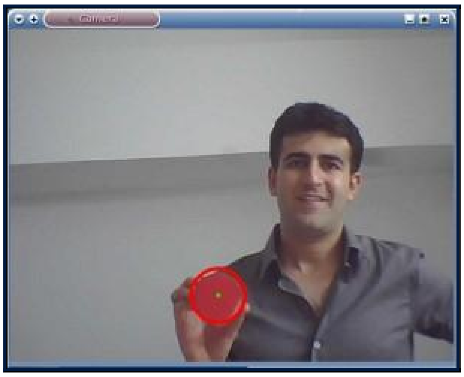


(python code)

```
</p>
<p>#declare necessary objects</p>
<p>hsv_frame = cvCreateImage(size, IPL_DEPTH_8U, 3)<br />
thresholded = cvCreateImage(size, IPL_DEPTH_8U, 1)<br />
thresholded2 = cvCreateImage(size, IPL_DEPTH_8U, 1)<br />
hsv_min = cvScalar(0, 50, 170, 0)<br />
hsv_max = cvScalar(10, 180, 256, 0)<br />
hsv_min2 = cvScalar(170, 50, 170, 0)<br />
hsv_max2 = cvScalar(256, 180, 256, 0)</p>
<p># convert to HSV for color matching<br />
# as hue wraps around, we need to match it in 2 parts and OR together<br />
cvCvtColor(frame, hsv_frame, CV_BGR2HSV)<br />
cvInRangeS(hsv_frame, hsv_min, hsv_max, thresholded)<br />
cvInRangeS(hsv_frame, hsv_min2, hsv_max2, thresholded2)<br />
cvOr(thresholded, thresholded2, thresholded)</p>
<p>
```

Shape based tracking

Then, I used Hough transform to detect the shape of a circle. Before applying the hough transform I smoothed the image because it seems to improve the results.

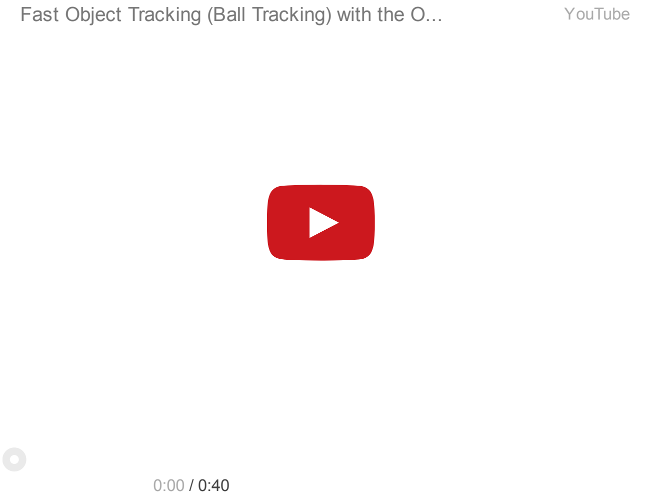


```
</p>
<p># pre-smoothing improves Hough detector<br />
cvSmooth(thresholded, thresholded, CV_GAUSSIAN, 9, 9)<br />
circles = cvHoughCircles(thresholded, storage, CV_HOUGH_GRADIENT, 2, thresholded.hei
<p>
```

The result

The result is pretty impressive. The code functions very well and detects the ball under most circumstances (i.e. the ball is far from the camera, close to the camera, slow movement, fast movement, etc).

You can see the result in this video:


Fast Object Tracking (Ball Tracking) with the O...

YouTube

0:00 / 0:40

Full python sources including servo movement

```
r />
****************************************************************************************<
 Name    : Fast object tracking using the OpenCV library                               *<
 Author  : Lior Chen &lt;chen.lior@gmail.com&gt;
 Notice  : Copyright (c) Jun 2010, Lior Chen, All Rights Reserved                      *<
         :                                                                             *<
 Site    : http://www.lirtex.com                                                       *<
 WebPage : http://www.lirtex.com/robotics/fast-object-tracking-robot-computer-vision   *<
         :                                                                             *<
 Version : 1.0                                                                         *<
 Notes   : By default this code will open the first connected camera.                  *<
         : In order to change to another camera, change                                *<
         : CvCapture* capture = cvCaptureFromCAM( 0 ); to 1,2,3, etc.                  *<
         : Also, the code is currently configured to tracking RED objects.             *<
         : This can be changed by changing the hsv_min and hsv_max vectors             *<
         :                                                                             *<
 License : This program is free software: you can redistribute it and/or modify        *<
         : it under the terms of the GNU General Public License as published by        *<
         : the Free Software Foundation, either version 3 of the License, or           *<
         : (at your option) any later version.                                         *<
         :                                                                             *<
         : This program is distributed in the hope that it will be useful,             *<
         : but WITHOUT ANY WARRANTY; without even the implied warranty of              *<
         : MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the               *<
         : GNU General Public License for more details.                                *<
         :                                                                             *<
         : You should have received a copy of the GNU General Public License           *<
         : along with this program.  If not, see &lt;http://www.gnu.org/licenses/&gt;
****************************************************************************************/
/usr/bin/python<br />
-*- coding: utf-8 -*-</p>
>from opencv.cv import *<br />
om opencv.highgui import *<br />
om threading import Thread<br />
mport serial</p>
>class RobotVision:<br />
  cvSize size<br />
  cvImage hsv_frame, thresholded, thresholded2<br />
  cvScalar hsv_min, hsv_max, hsv_min2, hsv_max2<br />
  cvCapture capture;</p>
>   def InitBallTracking():<br />
      globals size,  hsv_frame, thresholded, thresholded2, hsv_min, hsv_max, hsv_min2, hs
      print &quot;Initializing ball Tracking&quot;<br />
      size = cvSize(640, 480)<br />
      hsv_frame = cvCreateImage(size, IPL_DEPTH_8U, 3)<br />
      thresholded = cvCreateImage(size, IPL_DEPTH_8U, 1)<br />
      thresholded2 = cvCreateImage(size, IPL_DEPTH_8U, 1)</p>
>       hsv_min = cvScalar(0, 50, 170, 0)<br />
      hsv_max = cvScalar(10, 180, 256, 0)<br />
      hsv_min2 = cvScalar(170, 50, 170, 0)<br />
      hsv_max2 = cvScalar(256, 180, 256, 0)</p>
>       storage = cvCreateMemStorage(0)</p>
>       # start capturing form webcam<br />
      capture = cvCreateCameraCapture(-1)</p>
>       if not capture:<br />
          print &quot;Could not open webcam&quot;<br />
          sys.exit(1)</p>
>       #CV windows<br />
      cvNamedWindow( &quot;Camera&quot;, CV_WINDOW_AUTOSIZE );</p>
>   def TrackBall(i):<br />
      t = Thread(target=TrackBallThread, args=(i,))<br />
      t.start()</p>
>   def TrackBallThread(num_of_balls):<br />
      globals size,  hsv_frame, thresholded, thresholded2, hsv_min, hsv_max, hsv_min2, hs
      while 1:<br />
          # get a frame from the webcam<br />
```

```
     frame = cvQueryFrame(capture)</p>
>     if frame is not None:</p>
>       # convert to HSV for color matching<br />
       # as hue wraps around, we need to match it in 2 parts and OR together<br />
       cvCvtColor(frame, hsv_frame, CV_BGR2HSV)<br />
       cvInRangeS(hsv_frame, hsv_min, hsv_max, thresholded)<br />
       cvInRangeS(hsv_frame, hsv_min2, hsv_max2, thresholded2)<br />
       cvOr(thresholded, thresholded2, thresholded)</p>
>       # pre-smoothing improves Hough detector<br />
       cvSmooth(thresholded, thresholded, CV_GAUSSIAN, 9, 9)<br />
       circles = cvHoughCircles(thresholded, storage, CV_HOUGH_GRADIENT, 2, thresh
>       # find largest circle<br />
       maxRadius = 0<br />
       x = 0<br />
       y = 0<br />
       found = False<br />
       for i in range(circles.total):<br />
         circle = circles[i]<br />
         if circle[2] &gt; maxRadius:<br />
           found = True<br />
           maxRadius = circle[2]<br />
           x = circle[0]<br />
           y = circle[1]</p>
>       cvShowImage( &quot;Camera&quot;, frame );</p>
>       if found:<br />
         print &quot;ball detected at position:&quot;,x, &quot;,&quot;, y, &quot
>         if x &gt; 420:<br />
           # need to pan right<br />
           servoPos += 5<br />
           servoPos = min(140, servoPos)<br />
           servo(2, servoPos)<br />
         elif x &lt; 220:<br />
           servoPos -= 5<br />
           servoPos = max(40, servoPos)<br />
           servo(2, servoPos)<br />
         print &quot;servo position:&quot;, servoPos<br />
       else:<br />
         print &quot;no ball&quot;<br />
```

Sample Sources

This c++ code takes a video stream from an attached video camera, looks for an orange ball inside the stream, and prints the coordinates of the ball.
Three “debug” windows are shown to clarify the process: 1) the video capture. 2) the stream after the conversion to HSV, 3) the stream after conversion to HSV, color-filtering, and Hough transform.

```
<br />
/*********************************************************************************
*  Name    : Fast object tracking using the OpenCV library
*  Author  : Lior Chen &lt;chen.lior@gmail.com&gt;
*  Notice  : Copyright (c) Jun 2010, Lior Chen, All Rights Reserved
*          :
*  Site    : http://www.lirtex.com
*  WebPage : http://www.lirtex.com/robotics/fast-object-tracking-robot-computer-vi
*          :
*  Version : 1.0
*  Notes   : By default this code will open the first connected camera.
*          : In order to change to another camera, change
*          : CvCapture* capture = cvCaptureFromCAM( 0 ); to 1,2,3, etc.
*          : Also, the code is currently configured to tracking RED objects.
*          : This can be changed by changing the hsv_min and hsv_max vectors
*          :
*  License : This program is free software: you can redistribute it and/or modify
*          : it under the terms of the GNU General Public License as published by
*          : the Free Software Foundation, either version 3 of the License, or
*          : (at your option) any later version.
*          :
*          : This program is distributed in the hope that it will be useful,
*          : but WITHOUT ANY WARRANTY; without even the implied warranty of
*          : MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
*          : GNU General Public License for more details.
*          :
*          : You should have received a copy of the GNU General Public License
*          : along with this program.  If not, see &lt;http://www.gnu.org/licenses
*********************************************************************************
<p>#include &lt;opencv/cvaux.h&gt;<br />
#include &lt;opencv/highgui.h&gt;<br />
#include &lt;opencv/cxcore.h&gt;<br />
#include &lt;stdio.h&gt;</p>
<p>#include &lt;stdio.h&gt;<br />
#include &lt;stdlib.h&gt;<br />
#include &lt;string.h&gt;<br />
#include &lt;assert.h&gt;<br />
#include &lt;math.h&gt;<br />
#include &lt;float.h&gt;<br />
#include &lt;limits.h&gt;<br />
#include &lt;time.h&gt;<br />
#include &lt;ctype.h&gt;</p>
```

```
<p>int main(int argc, char* argv[])<br />
{</p>
<p>    // Default capture size - 640x480<br />
    CvSize size = cvSize(640,480);</p>
<p>    // Open capture device. 0 is /dev/video0, 1 is /dev/video1, etc.<br />
    CvCapture* capture = cvCaptureFromCAM( 0 );<br />
    if( !capture )<br />
    {<br />
            fprintf( stderr, &quot;ERROR: capture is NULL \n&quot; );<br />
            getchar();<br />
            return -1;<br />
    }</p>
<p>    // Create a window in which the captured images will be presented<br />
    cvNamedWindow( &quot;Camera&quot;, CV_WINDOW_AUTOSIZE );<br />
    cvNamedWindow( &quot;HSV&quot;, CV_WINDOW_AUTOSIZE );<br />
    cvNamedWindow( &quot;EdgeDetection&quot;, CV_WINDOW_AUTOSIZE );</p>
<p>    // Detect a red ball<br />
    CvScalar hsv_min = cvScalar(150, 84, 130, 0);<br />
    CvScalar hsv_max = cvScalar(358, 256, 255, 0);</p>
<p>    IplImage *  hsv_frame    = cvCreateImage(size, IPL_DEPTH_8U, 3);<br />
    IplImage*  thresholded   = cvCreateImage(size, IPL_DEPTH_8U, 1);</p>
<p>    while( 1 )<br />
    {<br />
        // Get one frame<br />
        IplImage* frame = cvQueryFrame( capture );<br />
        if( !frame )<br />
        {<br />
                fprintf( stderr, &quot;ERROR: frame is null...\n&quot; );<br />
                getchar();<br />
                break;<br />
        }</p>
<p>        // Covert color space to HSV as it is much easier to filter colors in t
        cvCvtColor(frame, hsv_frame, CV_BGR2HSV);<br />
        // Filter out colors which are out of range.<br />
        cvInRangeS(hsv_frame, hsv_min, hsv_max, thresholded);</p>
<p>        // Memory for hough circles<br />
        CvMemStorage* storage = cvCreateMemStorage(0);<br />
        // hough detector works better with some smoothing of the image<br />
        cvSmooth( thresholded, thresholded, CV_GAUSSIAN, 9, 9 );<br />
        CvSeq* circles = cvHoughCircles(thresholded, storage, CV_HOUGH_GRADIENT, 2
                                        thresholded-&gt;height/4, 100, 50, 10, 400
<p>        for (int i = 0; i &lt; circles-&gt;total; i++)<br />
        {<br />
            float* p = (float*)cvGetSeqElem( circles, i );<br />
            printf(&quot;Ball! x=%f y=%f r=%f\n\r&quot;,p[0],p[1],p[2] );<br />
            cvCircle( frame, cvPoint(cvRound(p[0]),cvRound(p[1])),<br />
                                    3, CV_RGB(0,255,0), -1, 8, 0 );<br />
            cvCircle( frame, cvPoint(cvRound(p[0]),cvRound(p[1])),<br />
                                    cvRound(p[2]), CV_RGB(255,0,0), 3, 8, 0 );<br
        }</p>
<p>        cvShowImage( &quot;Camera&quot;, frame ); // Original stream with detec
        cvShowImage( &quot;HSV&quot;, hsv_frame); // Original stream in the HSV co
        cvShowImage( &quot;After Color Filtering&quot;, thresholded ); // The stre
<p>        cvReleaseMemStorage(&amp;storage);</p>
<p>        // Do not release the frame!</p>
<p>        //If ESC key pressed, Key=0x10001B under OpenCV 0.9.7(linux version),<b
        //remove higher bits using AND operator<br />
        if( (cvWaitKey(10) &amp; 255) == 27 ) break;<br />
    }</p>
<p>     // Release the capture device housekeeping<br />
     cvReleaseCapture( &amp;capture );<br />
     cvDestroyWindow( &quot;mywindow&quot; );<br />
     return 0;<br />
   }<br />
```

Subscribe now via RSS and never miss a post!

Posted in Robotics by codex 51 Comments

51 Responses to "Fast Object Tracking – Robot Computer Vision"

triveni says:

July 21, 2011 at 19:07

Hey Lior,
Really found your work helpful:-)
I need some help of yours friend,
Actually I am not using the servo movement..instead want to use a mobile robot Lior..so i think there will b some changes in the code..So,can you please help us in giving me a brief idea of what the changes will be..
Or what m i supposed to do?

Thanking you in Advance Lior..

alexting says:

July 21, 2011 at 04:01

Hey, that was a great project done so far.
Btw, I am working on a similiar project with similiar concept.
Possible to share the whole project with me with the details on the hardware and software used?
Your help and advise shall be a great help to me.
Thank you very much!!

Dinesh Gosavi says:

July 13, 2011 at 12:44

Hey,
really nice job!
I referred the above code to track RED color object,
But now I have to track an object of BLACK color,
So how should I proceed ?
Should I convert the image to HSV first then filter out the color(I am not sure for the HSV values of BLCK color…??)
Or is there any other way….
your reply will be great help for me…
thanking you in advanced!!

Jaka says:

June 21, 2011 at 11:53

Nice work there! I am working on similar projects and I just have a question regarding the camera. I use normal USB web camera and Its quite slow for real-time apps. If the lightning is not good the fps are even lower.

I'd like to know how many fps do you get with your camera, what is the best choice, cam with Ethernet output or simple video output, which one do you suggest? I believe the fps speed of USB camera comparing to any other is noticeably lower, am I right? Thank you in advance…

Balakrishna says:

June 18, 2011 at 08:55

can you provide information about how to interface stpper motor,dc motor,sensor with beagleboard

Alireza says:

June 6, 2011 at 07:58

Thanks a lot for the code.

falkonin says:

June 2, 2011 at 11:30

Need help with saving this video ???

arijit says:

May 20, 2011 at 08:18

i am using devc++ ,OpenCV 2.1 ,windows 7 and after running the program it is showing

OpenCV Error: size of input argument donot match () in cvCvtcolor ,file C:\User\Vp\ocv\src\cv\cvcolor.cpp, line 2212

This program has requested the Runtime to terminate it in an unusual way.

please help is required to the power infinite .. :'(

ban says:

April 30, 2011 at 12:14

thanks alot for sharing your code,wish u mor success

debanjan says:

April 17, 2011 at 11:15

hi, how do we include the opencv library? i downloaded the library from the opencv website and then placed the opencv folder containing the cvaux.h highgui.h and other files in the same folder as the above code, but when i try to compile it it throws an error saying opencv/cvaux.h: No such file or directory. Please help..

Suchet says:

April 14, 2011 at 14:54

Thank you for this tutorial and source code.
This tutorial give me a lot of knowledge for my project.

endah says:

April 9, 2011 at 17:05

hi..
i'm doing my final project just the same like this.
i'm using visual studio 2008
but i have no idea how to access communication serial port
maybe you can give me solution.
thx

Nick says:

March 23, 2011 at 11:44

Thanks for sharing the program.but when i have compiled you C++ version of program in Ubuntu 10.10 using OpenCV2.1 i cant see anything in edge detection window.It is totally black…Due to with i get multiple circles on the screens…Please Help.

Peter says:

March 22, 2011 at 17:46

Can you do this in the RGB colour space also. I am doing shape (Ball) recognition for my part in the Team Project for a visually guided robot. Any help would be great.

Thanks

Pete

Peter says:

March 22, 2011 at 17:41

Can you do this but in the RGB colour space? I want to do this in C language for a Team Based Robot Project here at University… Any help would be great!

Cheers

Pete

ti7aaa says:

March 20, 2011 at 14:23

heyy my mind is going to blow up :D,,,,i wanted to ask

```
{CvCapture* capture = cvCaptureFromCAM( 0 );
if( !capture )
}
```

how a pointer is treated as an integer type or something like that,,,,,if u made int x ,,,,,int *y ,,,,,,then y=&x,,,,,, then i cant take y as y = ,,,,,,,and how a pointer of a type points to function ,,,,sure its something with me because i ran the code 😀 ,,,,,so i wish any text book or tutorial explains this thing,,,,,thanks,,,,,

Inprimus says:

March 18, 2011 at 19:21

Thank you so much for this short tutorial. I'm actually doing something similar to this project though I haven't interfaced with my servos yet. I want to simply (and quickly) detect motion then fire off an event which starts tracking from the next frame. Will absDiff-ing between two frames be enough to achieve this or do I have to dig deeper?
Thanks

Kai says:

March 17, 2011 at 02:47

Has finally solved the problem.

Kai says:

March 16, 2011 at 10:26

When I run the C++ source , I get an error. Anyone knows how to solve it??? Please help

Kai says:

March 11, 2011 at 10:16

Trying running the C++ in Visual studio 2008 but it fails.

ROYY says:

March 11, 2011 at 07:09

Hi lior, How did you do the video that you uploaded to youtube? did you use opencv or any other program?
if you used opencv How did you do??
I try to do it but the colors change in the recorded video, all the scene is kind of blue, the original colors get lost.

thanks a lot

Nikhil says:

March 4, 2011 at 21:29

I can see any thing in edge detection window.
I am using windows XP,Visual C++ 2008,OpenCV 2.1.
Please help. as soon possible

Rafi... says:

February 28, 2011 at 09:07

i implemented lane detection using hough transform for an image using C++

now i want to implement over an video…

if u have the code plz… send to my id…

thanx in advance…

Shawn says:

February 16, 2011 at 10:04

Hi…thank you for sharing your code. This is very helpful. I have few questions:

If my ball is another color, are hsv_min and hsv_max the only two values I need to change? And if so, do I directly get them from kcolorchooser app or are there any conversions involved?

Vik says:

February 16, 2011 at 09:46

Follow these instructions on how to run OpenCV in VS2008

http://uowteaminvincible.wordpress.com/2010/01/09/getting-started-with-opencv-in-microsoft-visual-studio-2008-in-windows-vista/

Ross says:

February 5, 2011 at 06:17

Thank you .Have you test this program in visual studio 2008? When i build it,there are several warnings for cvvidsurv.hpp & string.h,and couldn't run correctly.If anyone meets the same problem,please share the solution,thank you.

alin says:

January 20, 2011 at 00:53

Thats not even valid python code …

def InitBallTracking():
globals size

its def InitBallTracking(self):
self.size = "valid python"

and you dont declare the type of variable in python like cvSize size …
dont get me wrong but if you wanna post python code at least try to run it first …

bnc says:

January 18, 2011 at 15:43

I don't understands value in function hsv_min what is color
CvScalar hsv_min = cvScalar(150, 84, 130, 0);
CvScalar hsv_max = cvScalar(358, 256, 255, 0);

I understood is cvScalar(b,g,r,0);
but how to set hsv_min and hsv_max what is value ?
i want color matching yellow or orther?

thanks

swatz says:

January 10, 2011 at 03:27

Thanks for the code

Doli says:

January 3, 2011 at 14:31

Hai,

How if i want to track multiple objects (balls)?

Rizwan says:

December 2, 2010 at 19:47

Actually now i got it
This was because my image sizes were not same
i now made them them all same as input frame size
Thanks for Code 😃

Kai says:

March 16, 2011 at 10:55

Could you please tell me how you solve that error?

Rizwan says:

December 2, 2010 at 16:52

Hi !
i just tried ur code but it is crashes and show an exception error
I debugged it. The problem is in these statements
cvCvtColor(frame, hsv_frame, CV_BGR2HSV);
cvInRangeS(hsv_frame, hsv_min, hsv_max, thresholded);

every thing else is perfect
but when i include one or both of these statements it gives me error
cant figure out whats the problem

Mox says:

December 1, 2010 at 03:09

I've been searching for a simple solution to tracking a single ball for three weeks. You are the first person I found who is not only willing to share their source code, but also does so in a very organized and easy to follow way. This helps a ton. Well done and good luck with whatever else you're working on. Keep sharing, its what makes you more awesome than people who don't!!

Thanks a million.

Morpheus says:

November 27, 2010 at 18:23

Thanks A LOT for source code! I'm writing project in Gdańsk University of Technology and i've searched for this!

Amazing job!

andi says:

November 21, 2010 at 13:35

Hi,,,i have done your code, thanks for sharing…

but i want ask about kcolorchooser that you used to pick a color red,,,it (kcolorchooser) works in linux or can work in windows?

i want to change the color but i dont know how. cause in windows i just found RGB and HSL. And how i know the HSV in opencv (i use opencv) to change the color? thanks behzad says:

November 19, 2010 at 13:06

thanks alot for sharing

SAgirl:) says:

October 20, 2010 at 04:41

I really appreciate you sharing your work with us, Thank you

amit says:

October 17, 2010 at 20:27

i wanted to track the movement of multiple coloured objects and their movement……
can u help???

RoboGraphic says:

October 14, 2010 at 00:30

Thanks for sharing your work Lior, you have saved me some time writing this from scratch and it is great as a tutorial/skill builder. Good comments and nice logical flow to the code.

hSaif says:

October 6, 2010 at 10:35

Nice work.

I tried your source code, but there was a lot of noise in the binary image!!
is there any constraints on room lighting or camera hardware?

Regards.

joshu says:

September 26, 2010 at 18:37

I've just a question: how do you choose the values for the min/max hsv color matching? I mean, they are not RGB or HSV values, aren't they?

I'm talking about these :
CvScalar hsv_min = cvScalar(150, 84, 130, 0);
CvScalar hsv_max = cvScalar(358, 256, 255, 0);

Besides, I've tried the c++ code and they seem to be more useful for lips recognition (pink) than for tracking a red ball.

Thanks for sharing!

Lior Chen says:

September 28, 2010 at 15:46

They are HSV values. I chose them the easy way – I displayed the ball on the screen, and used another application – kcolorchooser, to show me the color value under the mouse pointer. After some experimentation I found the ranges that worked best. These values are very much dependent on the quality and type of the camera you are using and the lighting conditions.

NIKHIL says:

September 25, 2010 at 21:01

Great work…
Thanks for the program…

voelker says:

September 22, 2010 at 10:57

Your camera seems wireless, so do you perform the ball detection on a distant PC ? I'am doing almost the same on the beagleboard (opencv on the beagleboard), and the max frame rate i obtain for a 640×480 frame using the same detection technique is around 2-3FPS. On 320×250 frames, i can achieve up to 8FPS. This ball detection technique work great on homogeneous surface, but using it in real condition, with different light conditions will not give same results. The back projection technique (like in openCV examples) seems to give better results.

Lior Chen says:

September 28, 2010 at 15:42

Yes, I indeed to perform the ball detection over a distant PC. I tried doing it on the Beagleboard, but the results were not satisfying enough (low frame-rate as you said). In addition to the ball detection I intend to use OpenCV's face-detection to recognize people, and moreover there are speech-recognition and speech-synthesis software running on the Beagleboard, which seems more that enough already.
I will try the back projection technique, thanks for sharing!

bebatech says:

September 21, 2010 at 08:25

Great, thanks for sharing.

Can you share with us some details about the video camera?

tcmichals says:

September 21, 2010 at 21:59

It would be great to know the the type and cable… I was going to use a flat ribbon cable, but yours looks great!!!

Lior Chen says:

September 28, 2010 at 15:49

To what cable are you referring?

Lior Chen says:

September 28, 2010 at 15:36

Since I wanted most of the image processing to be performed in the PC and not in the embedded-linux board on the robot, I chose to use a wireless camera. It's very similar to the one found in this link: http://www.spycamera4u.com. It's light, compact, has a quite nice quality and a decent range (20m at least).

Andre (ddois) says:

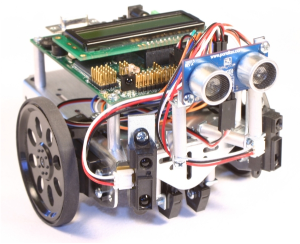
September 18, 2010 at 05:42

awesome, thanks to sharing =)

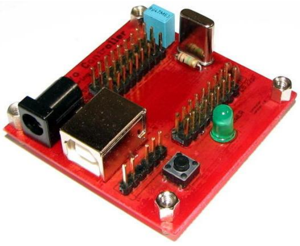
Related posts

Linux Robotic Platform – an Intelligent Robot

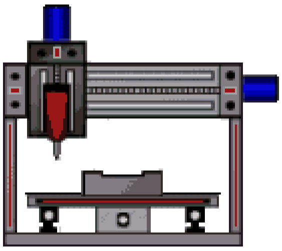
Introduction to Robotics



USB Servo Motor Controller



Homemade DIY CNC Machine using Step Motors, Dremel, and EMC2



Footer Left Content
To replace this, go to "Widgets" page and add your ow n w idgets to "Footer Left".

Suggested w idgets are: Categories, Recent Comments, Banners, Ads, Promotional Links etc.

Footer Middle Content
To replace this, go to "Widgets" page and add your ow n w idgets "Footer Middle".

Suggested w idgets are: Categories, Recent Comments, Banners, Ads, Promotional Links etc.

Footer Right Content
To replace this, go to "Widgets" page and add your ow n w idgets "Footer Right".

Suggested w idgets are: Categories, Recent Comments, Banners, Ads, Promotional Links etc.