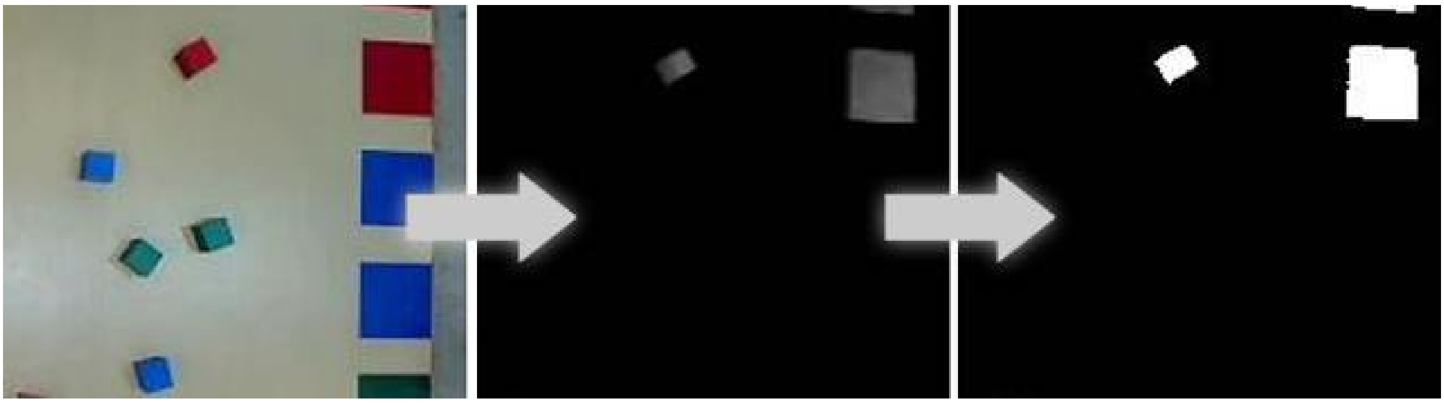


Thresholding

By Utkarsh | Published: January 3, 2010

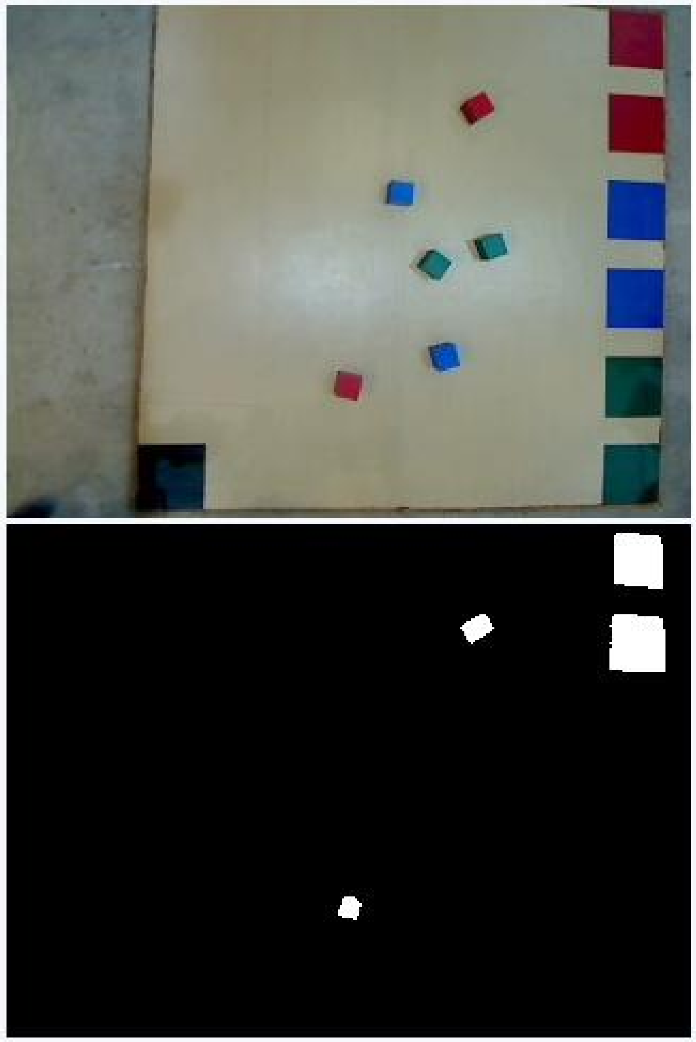
Introduction

Thresholding is one of the most basic techniques for what is called Image Segmentation. When you threshold an image, you get segments inside the image… each representing something. For example… complex segmentation algorithms might be able to segment out "house-like" structures in an image.

With thresholding, you can segment the image based on colour. For example, you can segment all red colour in an image.

Our Goal

Here's what we'll be making in this tutorial: an application that will threshold out the red regions in an image… something like this:



You might ask, exactly why go through all the trouble of thresholding an image? The reason for this is: The thresholded image is easier for the computer to analyse. It has got clear, stark boundaries, so the computer can easily find the boundary of each region (each of which represents a red patch).

Remember, thresholding isn't the best algorithm (as you'll see later)… but it gives reasonably good results for many images (as you can see in the image above).

The project

Start off by creating a C++ Win32 console application. Choose any name you want and click OK. Then, accept the default options and click Finish.

As always, we'll start off by adding OpenCV's headers:

```
#include <cv.h>
#include <highgui.h>
```

Then, goto the Project > Properties > Configuration Properties > Linker > Input, and put the following piece of text in the Additional Dependencies: "cv.lib cvaux.lib cxcore.lib highgui.lib", without the quotes of course. Now we're ready to use OpenCV.

We'll start off by loading an image. Write this in the main function:

```
int main()
{
```

```
IplImage* img = cvLoadImage("C:\\shaastra.jpg");
```

NOTE: Make sure you change the image’s path to something that actually exists.

Now, we’ll create 3 grayscale images. These 3 images will hold the red, green and blue channels of the image img. Add these lines to the main function:

```
IplImage* channelRed = cvCreateImage(cvGetSize(img), 8, 1);
IplImage* channelGreen = cvCreateImage(cvGetSize(img), 8, 1);
IplImage* channelBlue = cvCreateImage(cvGetSize(img), 8, 1);
```

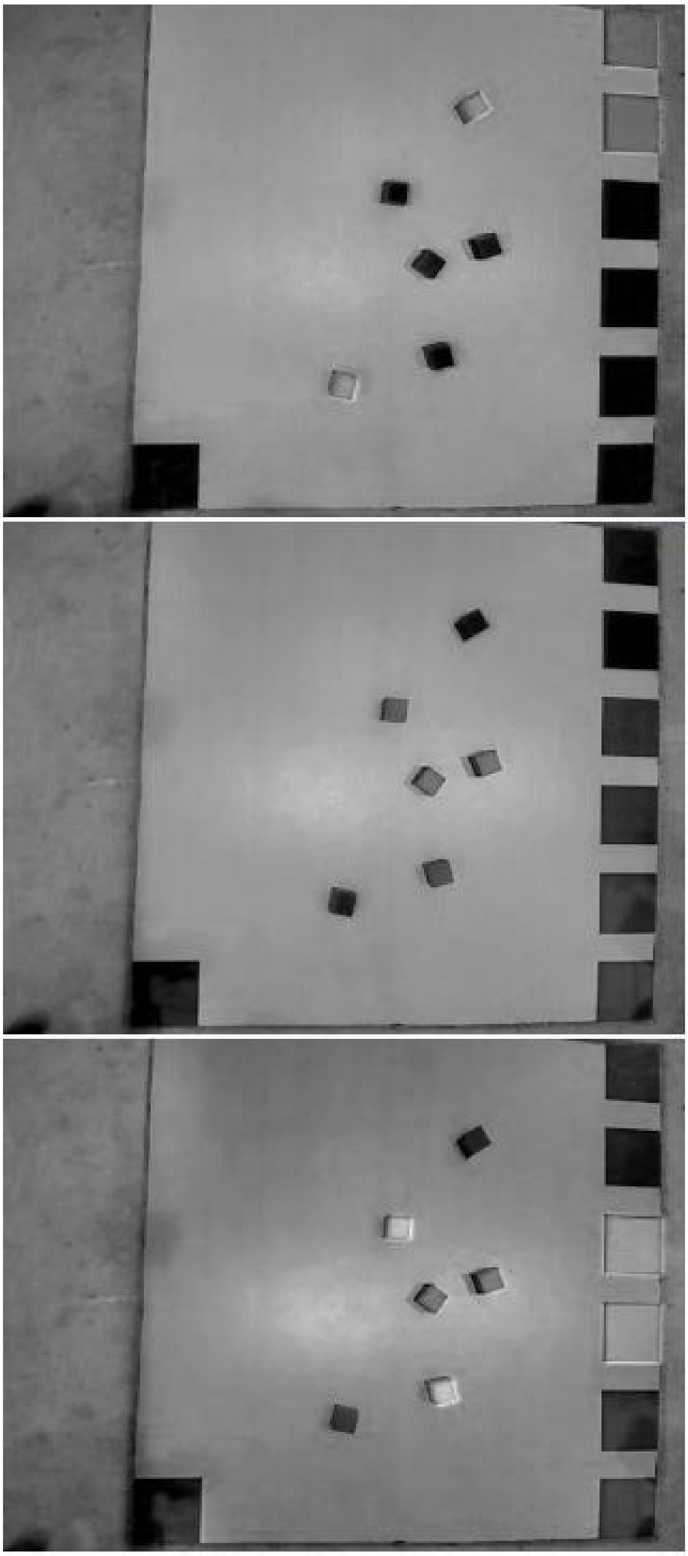
The cvCreateImage function allocates memory for an image. Currently, all these images are the same blank… each pixel is black.

Now, we’ll copy each channel into these images… one at a time:

```
cvSplit(img, channelBlue, channelGreen, channelRed, NULL);
```

The cvSplit command split the 3 channel image img into three different channels. The order is Blue, Green and Red because thats how its stored in memory.

Up till now, we have the three channels with us:



Red, green and blue channels of the image

We're interested in finding out the red regions… so we'll focus on the red channel (the top most in the above image).

You'll notice that not just the red regions are bright in the red channel… even the white regions are bright (because white=red+green+blue… so white is seen in all three channels).

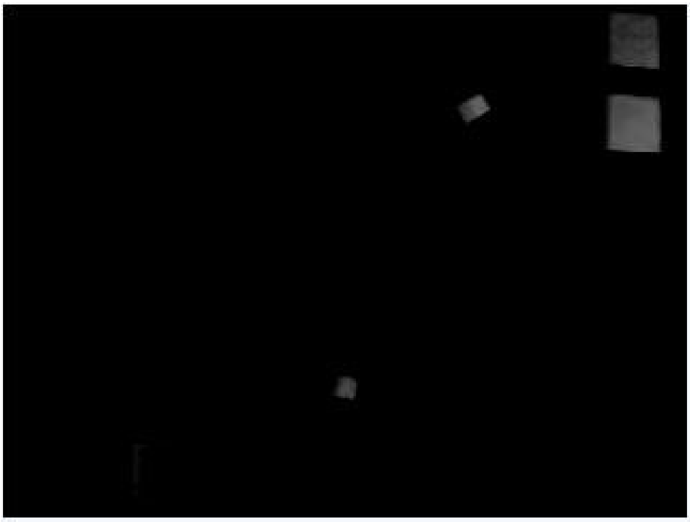
To extract actual red areas, we remove areas common with the green and blue channels… something like this: Just red = red channel – (green channel + blue channel). And we implement this in code using the following commands (add these to the main function):

```
cvAdd(channelBlue, channelGreen, channelGreen);
```

```
    cvSub(channelRed, channelGreen, channelRed);
```

Basically, what we've done is add the blue and green channels, and store the result in the green channel. Then, subtract this green channel from the red channel. And store the result in red channel.

Here is what channelRed has after these operations have been performed:

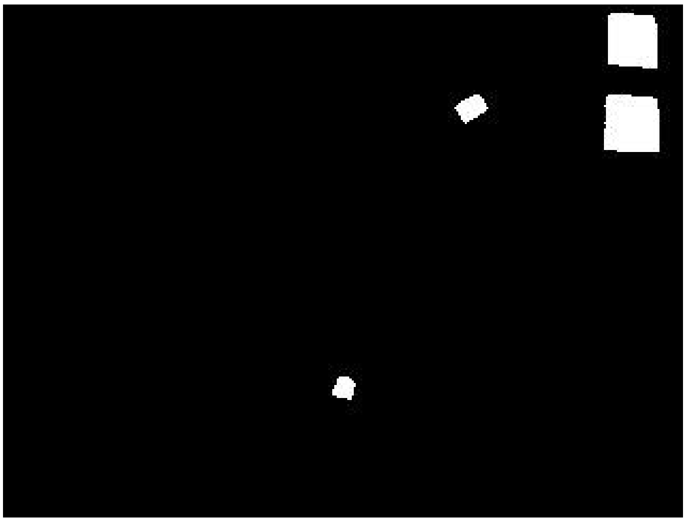


Notice that this in this image, only the actual red regions are bright… the rest of the image is dark. So uptill now we've been successful in isolating the red regions in the image.

Now, we do the thresholding. Add these lines to the main function:

```
    cvThreshold(channelRed, channelRed, 20, 255, CV_THRESH_BINARY);
```

After this line, channelRed stores this:



See how the red region turn bright? This is segmentation. You create segments… White=red in the original image… black=non red.

Now, an analysis of what actually happens in the cvThreshold function. The function goes through each pixel of the image. If the value of the pixel is greater than 20 (the third parameter)… it changes it to a 255 (the fourth parameter). Thats the reason all the reds turn bright.

The first parameter is the source image. The second is the destination image.

And see the last parameter? (CV_THRESH_BINARY) This parameter decides the behaviour of this function. CV_THRESH_BINARY change the value to 255 when the value is greater than 20. An other possible value is: CV_THRESH_BINARY_INV… which is the reverse: if the value is greater than 20, it is set it 0… else it is set to 255.

Now that we've thresholded the image, all we need to do is display it. Add these lines:

```
    cvNamedWindow("original");
    cvNamedWindow("red");
    cvShowImage("original", img);
    cvShowImage("red", channelRed);
    cvWaitKey(0);
    return 0;
}
```

Execute the program, and you should see the thresholded image.

Thresholding isn't that great

But it lets you get through a lot of situations quite easily (and efficiently). If you tried thresholding the same image to get the green patches, you'd get something like this:

You do get some greenish areas, but none of them isn't prominent enough.. and would probably be discarded as noise.

In such situations, you could try converting the image into HSV colour space. And then, segmenting the H channel. You'd have a narrow range of hues for green… to do that, you'd have to use the cvInRangeS function… which is described below.

Advanced Thresholding

The cvThreshold function is good for simple purposes… places where you just need to check if a pixel's value is greater or less than a particular value.

More powerful functions are the cvCmp, cvCmpS, inRange and inRangeS function. The functions ending in an

S let you compare the image against a particular value (like 20). The others let you compare the image against another image (so you could have different comparison values for different pixels).

The cvCmp function lets you specify the type of comparison (greater than, greater or equal, etc). Its syntaxes are:

```
cvCmp(src1, src2, dst, cmp_op);
cvCmpS(src, value, dst, cmp_op);
```

cmp_op can take these values:

- CV_CMP_EQ – equal to
- CV_CMP_GT – greater than
- CV_CMP_GE – greater or equal
- CV_CMP_LT – less than
- CV_CMP_LE – less or equal
- CV_CMP_NE – not equal to

The inRange function lets you specify a range of values (a minimum and a maximum) which is converted to a 255. Any value outside the range is set to 0. Its syntaxes are:

```
cvInRange(src1, lowersrc, uppersrc, dst);
cvInRangeS(src, scalarLower, scalarHigher, dst);
```

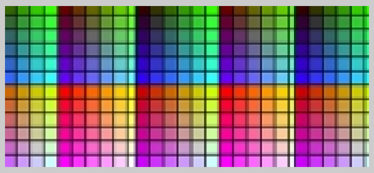
Pretty much self explanatory. This function can be used to threshold out an HSV image… where the H channel holds the colour image. I've written a tutorial that uses these function, you might be interested in reading it: Tracking coloured objects.

Wrap up

Thats it for now. I hope you learned something from this. And if you have any suggestions or criticisms, leave comments!

Issues? Suggestions? Visit the Github issue tracker for AI Shack

Back to top



A super fast

Color spaces

thresholding technique

22 Comments

Said RAHMANI
Posted August 9, 2010 at 2:56 am | *Permalink*

Hello,
I have to say that this tutorial helped me a lot to understand OpenCV. And it is really simple, unlike those tutorials out there that are a bit complicated (for me, since i am a bigginer)

Again thanks.

Said

Reply

Utkarsh
Posted August 9, 2010 at 9:19 am | *Permalink*

Glad you found this useful!

Reply

ferros
Posted September 30, 2010 at 10:13 pm | *Permalink*

Hi and good day
I wish to ask how i threshold out other color regions besides the 3 RGB color? For example, pink/orange color?

Reply

Utkarsh
Posted October 2, 2010 at 10:11 am | *Permalink*

For that, you could use RGB and compare against three values (something like 0<=R<128 && 128<=G<255 && B<=25). Or, you could use the HSV color space. One single channel is used to select the "color type". Have a look at this: http://www.aishack.in/2010/07/tracking-colored-objects-in-opencv/

Reply

ferros

Posted October 2, 2010 at 8:58 pm | *Permalink*

Yes i manage to do it, thanks Utkarsh, your explanation with coding really help me a lot.

Reply

Nkosi
Posted October 2, 2010 at 5:12 pm | *Permalink*

Hi
This tutorial helped me a lot, now I'm stuck with 1 thing. Is it possible to get the image positions of all red objects after applying a threshold?

Reply

Utkarsh
Posted October 2, 2010 at 5:20 pm | *Permalink*

Try using contours http://aishack.in/2010/01/an-introduction-to-contours/ . Or if you have just one object, try http://aishack.in/2010/07/tracking-colored-objects-in-opencv/

Reply

Manav Verma
Posted October 15, 2010 at 7:13 pm | *Permalink*

Hi Utkarsh,
Currently doing my project on finger tip detection. Initially I was getting any clue. However, I am hopeful after going through your tutorials :) God Bless.

Reply

Utkarsh
Posted October 16, 2010 at 11:02 am | *Permalink*

Great!

Reply

Keshan
Posted October 19, 2010 at 4:50 pm | *Permalink*

great stuff !! i learned a lot.. keep on writing.

Reply

Utkarsh
Posted October 19, 2010 at 4:55 pm | *Permalink*

😀

Reply

Uni
Posted October 25, 2010 at 12:53 am | *Permalink*

Nice one.. 🙂 Thanks so much!

Reply

Nadeeshani
Posted January 12, 2011 at 12:43 am | *Permalink*

Hi. I have a coin image which is in a red background. I want to segment the coin image from the background. I tried your code. Now it displays white for the red background and coin is filled with black color. How can I get only the coin? and coin should not be displayed in filled black color. I want to see the symbols on coin. Can you please help me? Thanks.

Reply

milad
Posted January 24, 2011 at 2:09 am | *Permalink*

Hi,I need your help to find orange ball. I wrote below code but it is not useful. please guide me.

```
private Image DetectOrange (Image Orginal)
       {
           Image Orange =
               Orginal.Convert().InRange(new Hsv(5, 70, 110), new Hsv(16, 255, 255));

           MemStorage stor = new MemStorage();
           Contour contours = Orange.FindContours(CHAIN_APPROX_METHOD.CV_CHAIN_APPROX_SIMPL

           Bgr bgrRed = new Bgr(Color.Red);

           while (contours != null)
           {
               Console.WriteLine("Contour # " + contours.BoundingRectangle.Width + " " + cc

               if (contours.BoundingRectangle.Width > 12 && contours.BoundingRectangle.Heig
               {
                   Rectangle rect = contours.BoundingRectangle;
                   rect.X = rect.X -1;//- 10;
                   rect.Y = rect.Y - 1;//- 10;
                   rect.Height = (rect.Height + 1);//20);
                   rect.Width = (rect.Width + 1);//20);

                   Orginal.Draw(rect, bgrRed, 1);
               }

               contours = contours.HNext;
           }

           return Orginal;
       }
```

Reply

Utkarsh
Posted February 11, 2011 at 11:42 am | *Permalink*

Tried using 2-8 as the Hue range?

Reply

sajad
Posted February 13, 2011 at 1:53 am | *Permalink*

hello
very good

Reply

Utkarsh
Posted February 13, 2011 at 2:04 am | *Permalink*

Thanks!

Reply

Saurabh
Posted March 3, 2011 at 12:43 pm | *Permalink*

Hi, my project is based on object tracking using opencv.My basic problem is occuring in thresholding.I am using cvThreshold function for that. Can you tell me the value of thresholding for red,green,blue.

Reply

Utkarsh

Posted March 3, 2011 at 12:55 pm | *Permalink*

Check this tutorial – Tracking colored objects in OpenCV I think this should point you in the right direction.

Reply

Abeer

Posted April 28, 2011 at 4:40 am | *Permalink*

Hello,
I am grateful to you for this tutorial,

But I have some questions since I am beginner with the opencv:
Is this code is also good with the color blue? how can I do that?
I would also ask about the object extracting (from the image) which its color is red for example, how can I do this to get a Black and White image (or binary image)?

Thanks a lot..

Reply

Utkarsh

Posted April 29, 2011 at 8:12 pm | *Permalink*

Search for tracking a colored object. I've written a tutorial on that!

Reply

lost

Posted July 27, 2011 at 2:58 pm | *Permalink*

thanx buddy really simple and interesting !!! 😃

Reply

Post a Comment

Your email is *never* published nor shared. Required fields are marked *

Name *

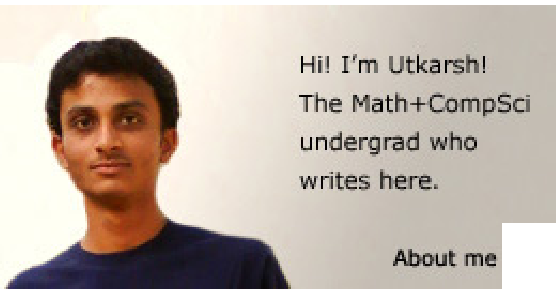
Email *

Website

Comment

You may use these HTML tags and attributes `<a href="" title=""> <abbr title=""> <acronym title=""> <b> <blockquote cite=""> <cite> <code> <del datetime=""> <em> <i> <q cite=""> <strike> <strong>`

Post Comment



Hi! I'm Utkarsh! The Math+CompSci undergrad who writes here.

About me

Popular Posts

- Scanning QR Codes

- The Canny Edge Detector
- Implementing Canny Edges from scratch
- Image Moments
- Predator: Tracking + Learning

More on AI SHack

Popular Beginners Utkarsh's favs

- Scanning QR Codes
- The Canny Edge Detector
- Implementing Canny Edges from scratch
- Image Moments
- Predator: Tracking + Learning

About me

My name is Utkarsh Sinha, and I'm an undergraduate student, pursuing B.E. Computer Science + M.Sc. Mathematics. Here, I help you understand ideas in Artificial Intelligence, using a not so techy and mathematical language. And in the process, learn more about Artificial Intelligence myself.

Read more at the about page