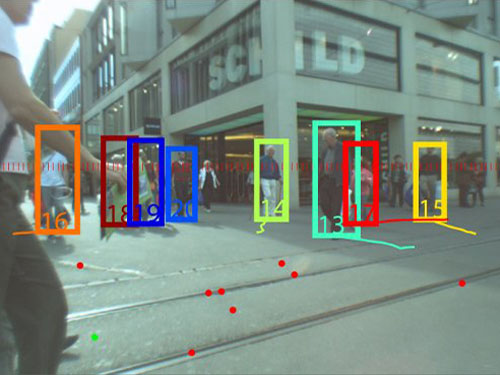
Tracking Motion with Raspberry Pi



8/23/2013

**Akansh Agrawal**

Table of Contents

# 1.Abstract

# 2.Introduction

### 2.1 Objectives

# 3.Background

### 3.1 Raspberry Pi

### 3.2 Image Processing

### 3.3 Object Detection

### 3.4 OpenCV

# 4.Programs and Algorithms

### 4.1 Loading an image

### 4.2 converting an rgb image to grayscale

### 4.3 videowriter and videocapture

### 4.4 convolution

### 4.5 thresholding

### 4.6 convolution

### 4.7 contours

# 5. email and its configuration

### 5.1 configuration

### 5.2 sending simple email

### 5.3 attachment

# 6. final program and results

# 7. conclusion and future work

# 8. references

Abstract

This project demonstrates a computational approach to object detection and tracking motion with Raspberry Pi. Tracking motion and object detection in its own, is a vast field and many researches are going on to improve it further. Object detection is a part of Image Processing, which again is a broader field and one should know the basics of image processing to understand how an object is detected in an image .Here we have tried to detect objects using an Open Source Vision Library (OpenCV) which comes with bundle of packages and methods, which makes it easier to compute. One part of OpenCV is Image Processing, which we have used in our project.

Raspberry Pi is a small computer and is becoming popular because it is really cheap and small. Although as it’s processor speed is less, it is slow as compared to ordinary PC, it can be supported by external memory and can be used in ubiquitous computing and embedded systems. One can say, it is a computer of future. As processor speed are becoming more fast, the day is not far where rather than using PC you can see devices like Raspberry Pi in destitute areas too.

Today, many applications that support image processing like Photoshop, Picassa are there which comes with ease so why we need to study image processing…?

With applications like Photoshop, you are restricted to specific use of Image Processing; but as you learn image processing and vision library, you can see a lot of things that you can do by your own. As digital world is not easy, with images, comes a lot of real world bugs that needs to be removed such as distortion, noise, jitter etc which really degrade the quality of an image. With image processing we have to remove these bugs so that we can work with images digitally. With OpenCV, you don’t really need to worry a lot, as there are predefined functions that can be used to remove real world errors.

Our project is based on object detection only. We have used the vision library with the Raspberry Pi. We have tried to show the strength of this new device and how we can use it further to explore its use in new areas and ubiquitously .Tracking motion on the other hand is still in its initial stage and can be very useful for security, automation and further its combination with machine learning can be used in various key areas.

introduction

Tracking Motion can be defined as

“The way to track any changes in the physical world.”

In the digital world, things are not that easy. The changes can be due to jitter, noise, distortion etc. Thus need a proper way to remove these so that we can process images in the digital world efficiently too. In digital world, tracking motion can be done with webcam etc. Object detection on the other hand is a way to detect any object in an image or video. Its not easy and still every object cannot be detected with the present techniques and algorithms. One should need to know what are objects in an image and how we process an image to detect objects in them.

# Objectives

Our Objective in this project is concerned with “Whenever there is motion, the camera will detect it, make a bounding box around the object which have motion and send a video of it via email to the user”.

Its uses where it can be deployed:

* In automated video surveillance-Suppose you own a bank and want to know who is coming to the bank and at what time he/she is leaving. So you can use this program to detect the person and send you an email whenever anyone is coming or leaving.
* In street lights-If any vehicle crosses the road when there is red signal, its car will be detected and a video of it will be send to you via email. So now you can detect which vehicle breaks the law.
* Around your safety room – If any person goes or leaves the room you will get to know.
* In public places-such as cafes, offices, libraries etc.

background

As part of our Project, we have focused on the following key areas:

* Raspberry Pi: Its background and how to setup, hardware/software needed for our project.
* Image Processing: How the image is stored, retrieved and modified. Various techniques and algorithms that support image processing. The packages need to be downloaded in our Raspberry Pi for streaming video.
* Object Detection: In this we focused on what are objects in an image and how to detect them, different algorithms related to it and how to track an object in a video or image.
* OpenCV Library: It is Open Source Computer Vision Library. As it is open source it comes free and anybody can download it. It comes with inbuilt packages and methods which can be used for image processing, object detection and tracking motion.

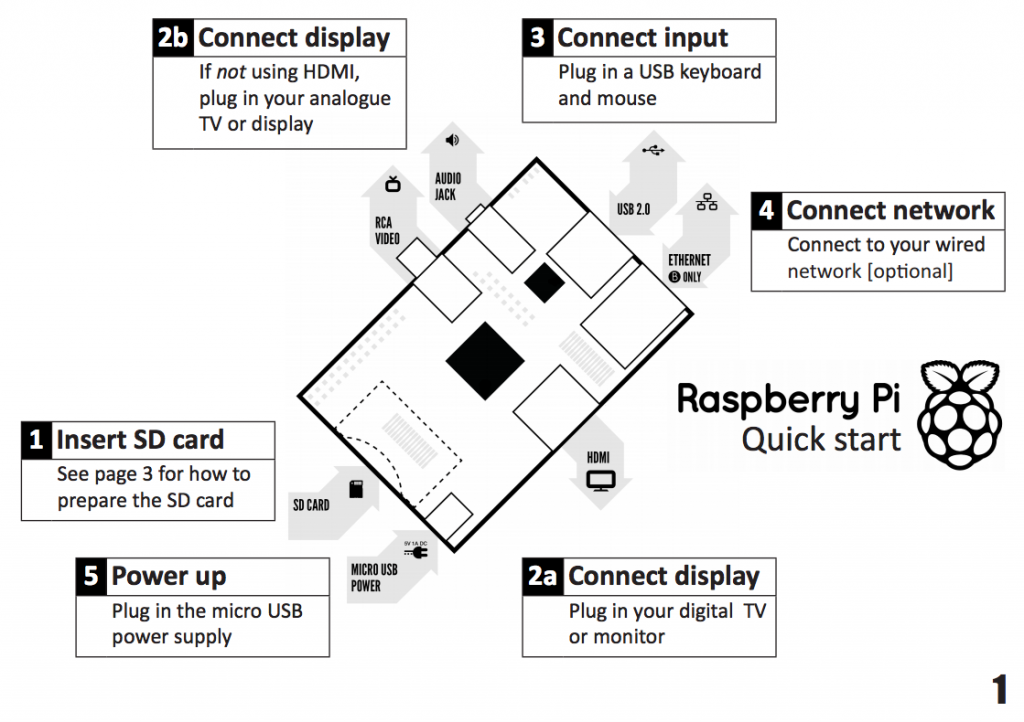
## Raspberry Pi

As I had already mentioned about this device, let see how we can use this device.

First you need to setup the SD card which will work as hard disk of our device and have your Operating System. So here is the link which will help you to setup it and also how to boot Raspberry Pi for the first time:

Refer References [10] for link:

The image below shows the specification of Raspberry Pi, the blocks in the image shows the hardware you will need for setting up Raspberry Pi. Some of these are optional such as Ethernet cable, but it will be helpful if you have.



## Image Processing

Image Processing is a vast field which deals with different morphologies with an image. With image processing you can change the contrast, brightness etc of an image, apply different morphological operations, thresholding, contours, convolution, feature detection, histograms, template matching etc. In our project we have used some of the operations of image processing such as Convolution, Contours, Thresholding , Loading and Converting an image, Capturing and writing a Video file and Object Detection.

We will go through each one of them in the next segment.

You will also need to download packages to stream video i.e. motion and to show your live webcam either on video player or browser. I find video player appropriate for my project, you can go with anyone. The following links will help you to download and setup these packages:

Refer to References [3] and [4].

## Object Detection and motion tracking

In this part we are concerned with what are objects in an image, what are features and how they can help us to detect an object. As an object moves, we need to track that object which in turn results to tracking motion. Both techniques are used in our project for our objectives. Tracking motion is often combined with different technologies such as machine learning to meet the requirement of using them ubiquitously.

## OpenCV

OpenCV is open source computer vision library which can be used for various aspect of fields related to vision. As Image Processing and Object Detection are part of vision, we have used this library to meet the requirement of our project. As it is open source it is totally free. This library comes with bundle of packages and methods which can be used directly, so you can just use them without worrying about all the math and background processing that it computes. We will describe some of the packages and methods of this library in our program.

For using this library, one have to download it and other packages to support OpenCV from the link given below:

Refer References [9].

Note:

* While running make command, you should run it through terminal only, don’t go to GUI.
* Download the latest version of OpenCV only.

# Programs and Algorithms

In this section, we will be going to cover some of the Techniques and related programs and algorithms that we have used in our project. They are:

* Loading an Image and Displaying it
* Converting an Image to Grayscale
* Convolution
* VideoWriter and VideoCapture ( CvCapture)
* Thresholding
* Contours

Now we will go and explain them one by one.

## Loading an Image

To load an image, you can use the OpenCV packages and functions as:

Here you first need to include packages in your program which contain these methods.

For imshow and imread you need to include package “highgui” as

#include <opencv2/highgui/highgui.hpp>

This package deals with all the reading and writing of images and videos.

Also you need to include “core” package which defines the basic building blocks of OpenCV library. (waitKey(), Mat etc)

#include <opencv2/core/core.hpp>

When you will run this program you will give an image as input (as a location of it) and you will get the following output.



## Converting an image to grayscale

By default, OpenCV loads an image in RGB format. If you want your image in grayscale then you need to convert it manually. The procedure for converting is as:

Why we convert an RGB image to grayscale – As RGB has three channels (red, green, blue), it becomes really difficult and complex to process the image. To avoid complexity, we convert the image to grayscale which contains only the intensity value.

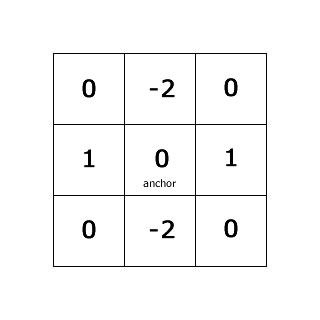
When you will run this program, you will see two windows (one for input and other for output) as:



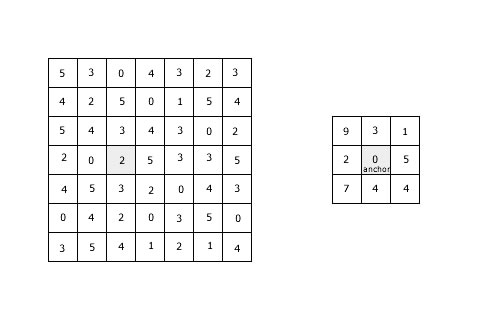
# Convolution

Convolution is a technique for general signal processing. We even apply the convolution method in mathematics where we convolute two functions which give the area overlap between the two.

Convolution is used in many uses such as calculate derivatives, detecting edges, apply blurs etc. To convolute in images, we use kernel, which is a small matrix. This matrix also has an anchor point:



So to convolute, we just pass this kernel over the entire image pixel by pixel and calculate the sum of the overlap cells of a matrix and put that value in anchor point.



How to apply Convolution using OpenCV :

OpenCV comes with a function filter2D which can be used to apply the convolution in an image.

You can use this function as:

filter2D(input image,output image,ddepth,kernel,anchor,delta,BORDER\_DEFAULT);

The parameters used in this function are as follows:

Input image-the input image.

Output image - the output image.

ddepth - it tells the output image depth, if it is -1,then the output image will have same depth as input image. It can have the following values:

CV\_8U – you want to represent your image as 8-bit uchar image-the image pixels will have values in unsigned character and values vary from 0 to 255.

Similarly, CV\_16U, CV\_32F, CV\_64F.

Kernel-convolution kernel you are applying

Anchor – the point in the kernel describes the present location of kernel in an image

Delta - if you want to add optinal delta value, it will be added to the output image.

Bordertype -again optional, values can be BORDER\_TRANSPARENT, BORDER\_CONSTANT, BORDER\_ISOLATED, BORDER\_DEFAULT etc. it is a pixel extrapolation method.

When you will run this program you will see the effect as this:



# Videowriter and VideoCapture(CvCapture)

Videowriter and videocapture are classes that are used for reading and writing a video file. If you want to read a video file using OpenCV, then you have to go through following steps:

# Threshold

Thresholding is one of the most basic technique that we use in image processing, and is also a part of image segmentation. When you threshold an image, you get the segments inside an image.

We can apply thresholding to segment the image based on the color. Suppose you want just the white color in your gray-scale image, so you can use this function. You can use the function threshold in OpenCV and there are 5 different operations to calculate the pixels w.r.t. threshold.

The threshold function is described as follows:

threshold(src,dst,threshold value,max\_BINARY\_value,threshold type);

src – source gray image

dst – output image

Threshold value – value which will be chosen to thresh the pixels

Max\_Binary\_value- maximum value to be set in the channel of an image

Threshold type – operation used for thresholding

After running the program you will get the result as:



# Contours

Contour as described in civil engineering as “It is a map which joins the point of equal heights”. So as through contours we can get a shape of any irregular object in an image, we have use contours in our program to make an outline of an object to track it.

OpenCV comes with handy of functions to support contours. Two functions that we are concerned with – to find the contours and to draw the contours. We have also used the convex hull which makes a bounding area around an object.

The procedure to find contour and draw them is as follows:

*The two functions that we have used are defined as:*

*findContours(source image,contour image,hierarchy,mode,method,offset);*

*hierarchy – it describes the hierarchy of contours to draw*

*mode – there are differnet modes defined:*

*CV\_RETR\_EXTERNAL – retrieves the external structure of the contour without the internal structure*

*CV\_RETR\_LIST – retrieves all the contours without hierarchy*

*CV\_RETR\_TREE – retrieves all the contours and make a hierarchy of them*

*Method – it is a contour approximation method defined as:*

 **CV\_CHAIN\_APPROX\_NONE** stores absolutely all the contour points. That is, any 2 subsequent points (x1,y1) and (x2,y2) of the contour will be either horizontal, vertical or diagonal neighbors, that is, max(abs(x1-x2),abs(y2-y1))==1.

 **CV\_CHAIN\_APPROX\_SIMPLE** compresses horizontal, vertical, and diagonal segments and leaves only their end points. For example, an up-right rectangular contour is encoded with 4 points.

 **CV\_CHAIN\_APPROX\_TC89\_L1,CV\_CHAIN\_APPROX\_TC89\_KCOS** applies one of the flavors of the Teh-Chin chain approximation algorithm.

*The other function defined as :*

*drawContours(source image,output image,contourIdx,color,thickness,linetype,hierarchy,maxlevel,offset);*

*contourIdx – index of the contour to be drawn.*

*Color – color of the contour*

*Thickness – thickness of the line to be drawn*

*lineType – There are different lineTypes such as LineType-8 linetype-4 and linetype-AA*

*hierarchy – optional information about hierarchy*

*maxlevel – maximum level for drawn contours.Differnet values can be used as*

*0 – only specific contour will be drawn*

*1 – draw contour and all the nested one*

*2 – draw contour,nested one and nested –nested one*

*and so on….*

*offset – it is a shift parameter that shifts the contour with the specified coordinates.*

*If you run this program you will see the result as:*

* *

Email

Email or the electronic mail is the technology through which one can send a message to any other or groups of people around any part of the world through the internet. Just as we send a mail to any one in analog world, through electronically we can send the message through email. We use different protocols in the application layer to send an email. There are three agents that we use to transfer email such as

MUA – This is message user agent

MTA – This is message transfer agent

MAA – message access agent

So as to transfer the email you need to download the packages which will act as MTA such as sendmail or exim4.Both are used to transfer the email.

I have downloaded the exim4 with mutt for my program to send a video of motion to the specified user.

In this segment we will see

* How to download sendmail or exim4 and do their configuration
* Send a simple email
* Sending a MIME mail and attachment

## Email Configuration

To download sendmail or exim4 in Raspberry Pi, you need to have an internet facility, then you just download the packages by running the command:

sudo apt-get install exim4

sudo apt-get install sendmail

You can also download mutt which act as MUA and adds additional features like attachment without you to worry about any clutters:

sudo apt-get install mutt

So now you have downloaded these packages you can use them to send a mail to the specific user. But for more advanced features such as attaching a file, sending mail to groups you need to do the configuration. I found these links useful:

Refer References [6] [7] and [8].

So you can just go through them and configure on your device.

## Sending a Simple Mail

Now as you have downloaded the packages for sending mail, we can now just send a mail to anyone as by executing the command as:

mail –s “Subject” email address

You can also echo your message as:

echo “message” |mail –s “subject” email address

## sending a mail with attachment

For attachment, I have used mutt, which without any configuration just let you send an email to the specified user.

The command for sending an attached mail using mutt as:

echo “message” |mutt -a “attachment file” email address

If you want to run this command or send an email through your program, then you first need to call the function system() as:

System(echo “message” |mutt -a “attachment file” email address);

This function will call internally the command from your program and you can just send a email from your program.

Note: Just copy your program to the root folder as only the root can call the system(); function directly, otherwise you will need to use sudo to call this function; but inside program, you cannot use it.

Final source code and results

So, finally we have reached to the end. Our final program will do the actual object detection and tracking motion. It involves all the techniques and algorithms that we have seen till now. The different phases of my program are as follows:

First we need to include all the header files and declare namespaces. Then take input from your webcam by just explained above and also declare variable for output video file and open it to write inside that file by using open function.

Then put a loop to get the frame by frame from your webcam. The input will be a RGB frame which you have to change to grayscale to avoid the complexity of working with three channels.

Then just apply convolution to avoid the noise, jitter etc. I have used Gaussian Blur to remove the noise, but you can use any blur or use filter2D to remove noise.

Then copy the first frame to another Mat object so that you can compare the two consecutive frames and see the difference between them. Its just the way to remove the background and get the foreground of the image.

So, you find the differences between two consecutive frames and square it to remove the negative values and then sum up to take it as a threshold value, which will tell you if there is any motion. The video that you will get after taking differences and squaring it will be:



As you can see this is not clearly distinguishing the differences between the black and white pixels, so we just apply the threshold to make those white pixels more dense. After threshold you will get the output as:



Now as you can get the white pixels, you can just draw the contours of them, so we apply findContours(); method and if you draw these contours you will get the final result as:



So here is the source code for it:

[Source Code](final%20program%20for%20tracking%20motion.zip)

And finally when you run the source code, you will get the final output as:



Conclusion and Future Work

This segment summarizes our project and our results, as well as offers new direction for future work based on our ideas, implementations and discoveries. I have tried my best to make the program as efficient as possible. As we have use OpenCV for our project, it helps us a lot to make our program easier to understand as well as implement it.

Our project is based on detecting motion with the new hardware Raspberry Pi, which is due to its cheap prize can open- new fields and can be use in ubiquitous computing. As we have discussed some of the examples where this can be used, its power lies within low budget and optimized result.

So we now come to conclusion that whether we met our objectives:

I have made the program which can be used in the examples discussed in our objectives. The program is not very efficient but can be further improved by using machine learning algorithms and train the Raspberry Pi to a level that it can detect any object approximately.

As Object Detection itself is in its introductory phase, there is no such efficient algorithm exist till now that can give a100% correctness. And therefore, many researches are going on to improve this field further so that object detection becomes more efficient. We have tried to meet the same goal and make our program efficient to give good results.

## Future Work

As I have just concerned to “detect the motion and made the bounding box and send it via email”. By exploring aspects of my work in more details, one can take this to the next level and make this program more efficient. You can use machine learning algorithms and train your data to detect objects.

* According to my study and discovery, one can use histograms to make the algorithm more efficient, normalizing it and then using machine learning algorithm to train the data.
* One can also use Kinect and detect motion in 3D.
* With Raspberry Pi, you can explore more to solve the day- to-day life problems. As it is a small, cheap and handy tool you can just use it in the places where till now, it is not possible to place a computer.

Example – To place a Raspberry Pi inside a Coffee machine and instruct it to serve coffee whenever one puts the glass in it or with the help of Bluetooth, one can interact with the machine which type of coffee it wants via email.

These are some of the ideas and research of mine. One can vision by itself and explore these new technologies to solve real world problems.

References

[1] Learning OpenCV by Gary Bradski and Adrian Kaehler ISBN 978-0-596-51613-0

[2] OpenCV Tutorial book comes with OpenCV

[3] <http://techspect.co.uk/index.php?id=how-to-raspberry-pi-webcam-server-stream>

[4] <http://www.youtube.com/watch?v=kqwHbq4QNlM>

[5] <http://www.raspberrypi.org/downloads>

[6] <http://www.debian.org/releases/stable/i386/ch08s05.html.en>

[7] <https://library.linode.com/email/exim/send-only-mta-debian-5-lenny>

[8] <http://www.simplehelp.net/2008/12/01/how-to-send-email-from-the-linux-command-line/>

[9] <http://docs.opencv.org/2.4/doc/tutorials/introduction/linux_install/linux_install.html>

[10] <http://www.youtube.com/watch?v=ZLWNr7sy-ZU>