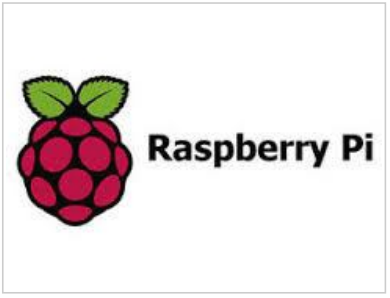
LOW LATENCY (~0.4 S) VIDEO STREAMING FROM RASPBERRY PI USING MJPEG-STREAMER AND OPENCV

BY PKOUT JUNE 9, 2014 ELECTRONICS



There are multiple ways to stream video from Raspberry Pi (RPi) to another computer via the wired or wifi ethernet. After trying multiple methods, stumbled upon one that leads to minimal latency and works really well over a wifi connection. The solution is the combination of mjpeg-streamer on the RPi and OpenCV client program on the other computer. In this tutorial, I am working with linux Ubuntu as the client computer. Let's get to it!

First, we will install the RPi software, then receiving computer software.

RPi Software Installation

We will not use the standard mjpeg-streamer package because that one doesn't have a built-in support for the RPi camera. Instead, we'll use this guy's great fork of mjpeg-streamer: https://github.com/jacksonliam/mjpg-streamer. It allows streaming video frames directly from your RPi camera, which is very efficient as well as convenient. So, let's install it:

Log into your RPi and go to /usr/src/ and create a directory mjpeg-streamer there:

```
cd /usr/src
sudo mkdir mjpg-streamer
sudo chown `whoami`:users mjpg-streamer
cd mjpg-streamer
```

- Now clone the mjpeg-streamer from the github repository there:

  ```
  git clone https://github.com/jacksonliam/mjpg-streamer.git .
  ```

- In order to compile the code, we'll need to install some library dependencies:

  ```
  apt-get install libv4l-dev libjpeg8-dev imagemagick build-essential cmake subversion
  ```

- Next, we'll need to compile the mjpeg-streamer. Enter:

  ```
  cd mjpg-streamer-experimental
  make
  ```

- Now we should be set to start streaming the video. There are many options you can set. For details, visit the GitHub page linked above and look at the readme page. Here, we will do a simple example of streaming of 640×480 resolution video at 20 frames per second. If you lower the resolution, the latency will get smaller.

  ```
  export LD_LIBRARY_PATH=.
  ./mjpg_streamer -o "output_http.so -w ./www" -i "input_raspicam.so -x 640 -y 480 -fps 20 -ex night"
  ```

  The export LD_LIBRARY_PATH variable sets the current directory as a path where programs should look for libraries. Our program uses output_http.so and input_raspicam.so libararies found in the current directory, which is why we added that directory to LD_LIBRARY_PATH.

- That's it! Mjpg-streamer will now stream the video to the port :8080 on this RPi. We will access the streamer from the other computer over the network and display it.

Client Computer Software Installation

Note down the RPi's IP address as we will need it to access the stream on its port 8080. Before we do so, we'll need to install Python and OpenCV, if they don't exist on your system yet.

- Install Python and OpenCV:

  ```
  sudo apt-get install python python-opencv python-numpy
  ```

- Now, you have two options how to view the stream from the RPi. Either you open a browser and enter your RPi's IP address followed by 8080 (e.g. http://192.168.0.193:8080). This will load a web page generated by the RPi. You can then click on the Stream tab and see the video stream right there. Alternatively, and those who want to further process the video will prefer this, you can create a file rpi-stream.py and paste the script below into it to get a video stream from the RPi and display it using OpenCV.

  ```
  touch rpi-stream.py
  ```

  Paste this into the file and save it.

  ```
  import cv2
  import urllib
  import numpy as np

  stream=urllib.urlopen('http://192.168.0.193:8080/?action=stream')
  bytes=''
  while True:
      bytes+=stream.read(1024)
      a = bytes.find('\xff\xd8')
      b = bytes.find('\xff\xd9')
      if a!=-1 and b!=-1:
          jpg = bytes[a:b+2]
          bytes= bytes[b+2:]
          i = cv2.imdecode(np.fromstring(jpg, dtype=np.uint8),cv2.CV_LOAD_IMAGE_COLOR)
          cv2.imshow('i',i)
          if cv2.waitKey(1) == 27:
              exit(0)
  ```

  (source: http://stackoverflow.com/questions/21702477/how-to-parse-mjpeg-http-stream-from-ip-camera)

  Don't forget to change the IP address in the script with your RPI's IP address.

- Finally with the mjpg-streamer running on the RPi, execute the above Python script on the computer where you want to see the video:

  ```
  python rpi-stream.py
  ```

  You should now see a window open and start streaming the video with a latency 0.5 seconds or less.

- BIG THANK YOU to jacksonliam user on GitHub for doing this work! You've done an amazing job!

Related

Parrot ARDrone 2.0 Video Streaming through OpenCV in Linux
October 19, 2012
In "Linux"

Install Node.js 6 on Raspberry Pi
September 13, 2016
In "electronics"

Raspberry Pi DXF Drawing
June 23, 2014
In "electronics"

TAGGED mjpg-streamer, raspberry pi, streaming. BOOKMARK THE PERMALINK.

« Setting up Django with MongoDB

Raspberry Pi DXF Drawing »

@pierre muratory Can you explain more about this? I'm having the same problem also in processing, when I'm handling cascade classifiers the video streaming starts to lag no idea why. but when no cascades being handled the video streaming go smoothly.
Would appreciate the help :)

• Reply • Share ›

pierre muratory ➚ Beabars Abaza • 21 days ago

Hello, it means the time to process your cascade classifier that is in the same while loop than the socket reading, takes more time than receiving a full image. so, in the socket more than one image is accumulated and since you manage image one after the other (and not the latest image), there is an increasing latency that appears when you display the image being processed. To resolve this latency, one of this solution is to flush the socket once you terminated the cascade classifier ('del stream') and wait for a new image coming in the socket and then handle this latest one. Hope this help. Rgds

• Reply • Share ›

Beabars Abaza ➚ pierre muratory • 21 days ago

@pierre muratory first thanks for your reply, could you provide a source can get me on track please?
I'm a bit newbie about the video streaming side..
My base code [taken from a tutorial] for video streaming on client pc is shown below.
I understood what you said and it makes sense but can't figure out how to apply that..

• Reply • Share ›

pierre muratory ➚ Beabars Abaza • 20 days ago

you should try something like this :

• Reply • Share ›

pierre muratory ➚ pierre muratory • 20 days ago

you should try something like this :

• Reply • Share ›

Izzat • a year ago

Hello @pkout Thanks for sharing such great comprehensive tutorial. I face some difficulty in executing rpi-stream.py codes. an error occurs after executing code, says File "C:/Users/Syed/PycharmProjects/rpi-stream/rpi-stream.py", line 14, in <module>
i = cv2.imdecode(np.fromstring(jpg, dtype=np.uint8), cv2.CV_LOAD_IMAGE_COLOR)
AttributeError: 'module' object has no attribute 'CV_LOAD_IMAGE_COLOR'

can you help me how can i fix it?

• Reply • Share ›

Marco Pistolesi • 2 years ago

Hi and THANK YOU SO MUCH for input_raspicam.so !
works great and it's amazing fast (I'm not using the pyton part but just need it on browser or java) if compared to other systems like input_file with raspistill !!

• Reply • Share ›

Izhar Shaikh • 2 years ago

Thank you very much! :) Works great!

• Reply • Share ›

CalBoy2015 • 2 years ago

What is the C++ equivalent of the rpi-stream.py code. The rpi-stream.py works perfectly. However, I
I tried following but I can't get it work. I google around but failed to find any solution for this. Any idea what I have done wrong?

```
#include "opencv2/opencv.hpp"
#include <iostream>
using namespace cv;
using namespace std;

int main(int, char**) {
VideoCapture cap;
// Note: None of the following that I tried work. What's wrong?
cap.open("http://192.168.0.124:8080/?action=stream");
//cap.open("192.168.0.124/?action=stream?dummy=param.mjpg");
//cap.open("http://192.168.0.124:8080/video?x.mjpg");
//cap.open("http://192.168.0.124:8080");
//cap.open("http://192.168.0.124:8080?stream=mpeg"); // a mjpeg , ipcam stream

if(!cap.isOpened()) { // check if we succeeded
```

see more

• Reply • Share ›

Izhar Shaikh ➚ CalBoy2015 • 2 years ago

The part --> cap.open("http://192.168.0.124:8080/?action=stream") <-- is wrong.

OpenCV expects the frame to be in the form of an array. The URL "http://192.168.0.124:8080/?action=stream" doesn't give any array, it gives the data over HTTP to be read from your web browser. This is the reason why urllib package has been imported by the author in python file, which can access the streaming over HTTP in python and numpy package can then convert the streaming into array frames.

For this to be implemented in C++, you need to use any equivalent library to urllib such as libcurl [http://curl.haxx.se/libcurl/] and then convert the data into a Mat (i.e. array) so the openCV can access that.

Hope that helps!

• Reply • Share ›

Vandan Revanur • 2 years ago

Wonderful article Works good. How do I measure the exact latency of the stream?

Wonderful article.Works good. How do I measure the exact latency of the stream?

Reply • Share ›

Tony Garcia • 2 years ago

Hi, great work, a question. On the client side is there any way to see the stream of multiple cameras? Thanks

Reply • Share ›

Mario Holzinger • 2 years ago

the script works out of the box on my RPi 1Bs :-)
The only big issue I faced is a huge latency when i like to display it on my LCD connected over the DSI on the pi.

Using a HDMI Display (TV) all works perfect, using my Tontec 480x320 display the latency of the video increases to 3-5sec :-(

Reply • Share ›

Rebecca • 3 years ago

is there anyway to make the rpi software side executable from a python script?

Reply • Share ›

pkout Mod ➚ Rebecca • 3 years ago

There surely is, but it's non-trivial to do and beyond the scope of this article, unless you simply want to wrap the executable in a Python script. If so, see this http://stackoverflow.com/qu...

Reply • Share ›

Rebecca ➚ pkout • 3 years ago

I'm just trying to get the camera stream to start when the raspberry pi boots up I've tried a couple of tutorials but none of them seem to work for me

Reply • Share ›

Mike Thomas • 3 years ago

This works on the iPhone in a UIWebView ! Perfect for my project

Reply • Share ›

Akın Evren Özsu • 3 years ago

How can I add date and and time? I used cv.Puttext but it is giving me an error:

cv.PutText(i,dati, (0,25),font, (0,255,0))
TypeError: CvArr argument 'img' must be IplImage, CvMat or CvMatND. Use fromarra
y() to convert numpy arrays to CvMat or cvMatND

Reply • Share ›

Tut • 3 years ago

Add the option -q 8 before -fps 20 when invoking mjpg-streamer. This reduces the bandwidth requirement drastically on cost of a slightly worse picture quality. For me this resulted in a latency of even much better than 0.4s. Instead of using the python script to view the stream try also open a browser pointing to your raspberry:8080 and try the JavaScript viewer - that worked so well for me that I don't have to use python script anymore.

Reply • Share ›

Vandan Revanur ➚ Tut • 2 years ago

Hey Tut, How did you measure latency of the stream?

Reply • Share ›

pkout Mod ➚ Tut • 3 years ago

Thanks for sharing your tips, Tut! That's great! I used the Python script because I wanted to further do OpenCV image processing on the received video and Python is a good way to accomplish that. There's also a browser interface one can use with mjpeg_streamer if one simply wants to see the video on the screen and do nothing else with it.

• Reply • Share ›

Jonathan Miller • 3 years ago

Thanks for a great post, latency is much better than other tutorials I have tried using VLC.

I am having an issue with the opencv python script you included though (works fine when using a web browser to get the stream). When I execute the python script I get the following error:

Traceback (most recent call last):

File "rpi_stream.py", line 14, in <module>

i = cv2.imdecode(np.fromstring(jpg, dtype=np.uint8),cv2.CV_LOAD_IMAGE_COLOR)

AttributeError: 'module' object has no attribute 'CV_LOAD_IMAGE_COLOR'

I am a bit of a n00b when it comes to python and opencv so any help you could provide would be appreciated.

• Reply • Share ›

pkout Mod → Jonathan Miller • 3 years ago

Hi Jonathan. What version of OpenCV do you have in your system? You can find out in python by running: import cv2; print cv2.__version__ . I run 2.4.8, but this should work on earlier versions as well. When I issue: print cv2.CV_LOAD_IMAGE_COLOR, I get 1 printed out. Otherwise, your OpenCV installation might be corrupted.

• Reply • Share ›

Jonathan Miller → pkout • 3 years ago

Managed to get this working using opencv2.4.9 :)

Guess some of the syntax must have changed in opencv3.0 beta.

• Reply • Share ›

Jonathan Miller → pkout • 3 years ago

Hi, I currently have opencv-3.0.0-beta installed but I am having a lot of problems with it so I am going to install 2.4.8 or 2.4.9 as I see that is what most people are using. Will let you know if that resolves the problem.

• Reply • Share ›

Jai ... • 3 years ago

Works nicely for me, very nicely. Very little latency and having it straight to python on the client is handy. Thanks for this!

• Reply • Share ›

pkout Mod → Jai ... • 3 years ago

I am glad it worked for you. Thanks for letting me and others know :).

• Reply • Share ›

Ingusan • 3 years ago

Hi there,

Thanks for the tutorial, it's comprehensive step-by-step guide. :)
Yeah, the latency is quite low.
However, I still can see latency (~1.5 seconds) when I stream it via VLC.
(I assumed it is caused by VLC's end)
I'm just curious, did you experience that as well when you stream the video over VLC?

• Reply • Share ›

pkout Mod • 3 years ago

I don't know what hardware you're using, Bruno, but I do get under half a second latency. I posted this post not as a click bait, but as a way to remind myself how to replicate this process when I forget down the road. Otherwise I wouldn't be doing it :).

pkout Mod • 3 years ago

Hi Adam,

This message means that the folder you're in when issuing the 'make' command doesn't contain the Makefile file. Looking at your command as you posted it, you have a space character in the name of the directory you're cding into "cd mjpeg-streamer -experimental". If you copy/pasted it from your terminal, then you failed to change directory and, therefore, the 'make' command never found the Makefile. Make sure the directory you're in contains the Makefile.

Adam Chatfield • 3 years ago

Hi there, I am having a bit of trouble with this ...

Next, we'll need to compile the mjpeg-streamer. Enter:

cd mjpg-streamer-experimental
make

I am getting this....

pi@raspberrypi /usr/src/mjpg-streamer $ cd mjpg-streamer -experimental

pi@raspberrypi /usr/src/mjpg-streamer/mjpg-streamer $ make

make: *** No targets specified and no makefile found. Stop.

Any help would be appreciated! Thanks
(bit of a beginner at this....)

SEARCH

CATEGORIES

- electronics
- Linux
- Other
- Programming
- Security
- Windows

TAGS

ardrone 2.0 arduino bind boost C++ cs-cart cscart cygwin django drawing dxf encrypt fowarding git gpg gpg4win kubuntu linux masm matlab mingw mjpg-streamer mongo mongoengine mysql network manager opencv parrot port forwarding push python raspberry pi rc receiver receiver remote repository security ssh-gent streaming sublime text subversion svn ubuntu video windows wordpress