Stream video from ffmpeg and capture with OpenCV

I have a video stream coming in on rtp to ffmpeg and I want to pipe this to my OpenCV tools for live streaming processing. The rtp linkage is working because I am able to send the incoming data to a file and play it (or play if via ffplay). My OpenCV implementation is functional as well because I am able to capture video from a file and also a webcam.

The problem is the streaming to OpenCV. I have heard that this may be done using a named pipe. First I could stream the ffmpeg output to the pipe and then have OpenCV open this pipe and begin processing.

What I've tried:

I make a named-pipe in my cygwin bash by:

```
$ mkfifo stream_pipe
```

Next I use my ffmpeg command to pull the stream from rtp and send it to the pipe:

```
$ ffmpeg -f avi -i rtp://xxx.xxx.xxx.xxx:1234 -f avi -y out.avi > stream_pipe
```

I am not sure if this is the right way to go about sending the stream to the named pipe but it seems to be accepting the command and work because of the output from ffmpeg gives me bitrates, fps, and such.

Next I use the named pipe in my OpenCV capture function:

```
$ ./cvcap.exe stream_pipe
```

where the code for cvcap.cpp boils down to this:

```
cv::VideoCapture *pIns = new cv::VideoCapture(argv[1]);
```

The program seems to hang when reaching this one line, so, I am wondering if this is the right way of going about this. I have never used named pipes before and I am not sure if this is the correct usage. In addition, I don't know if I need to handle the named pipe differently in OpenCV--change code around to accept this kind of input. Like I said, my code already accepts files and camera inputs, I am just hung up on a stream coming in. I have only heard that named pipes can be used for OpenCV--I haven't seen any actual code or commands!

Any help or insights are appreciated!

UPDATE :

I believe named pipes may not be working in the way I intended. As seen on this cygwin forum post:

> The problem is that Cygwin's implementation of fifos is very buggy. I wouldn't recommend using fifos for anything but the simplest of applications.

I may need to find another way to do this. I have tried to pipe the ffmpeg output into a normal file and then have OpenCV read it at the same time. This works to some extent, but I imagine in can be dangerous to read and write from a file concurrently--who knows what would happen!

opencv ffmpeg cygwin named-pipes

edited Oct 7 '11 at 14:48

asked Sep 27 '11 at 19:21

chembrad
549 3 11 26

I've been trying to do something similar. Did you find some article that explains about the pipe/etc? – Utkarsh Sinha Sep 27 '11 at 19:31

2 I found this small post ffmpeg-users.933282.n4.nabble.com/h264-problem-td2574908.html in the ffmpeg-users forum hinting at what he accomplished, which gave me hope. I also found someone trying this with VLC forum.videolan.org/viewtopic.php?f=32&t=69351 and something similar with x264 forums.creativecow.net/readpost/291/84 – chembrad Sep 27 '11 at 19:53

1 Answer

hope it's not too late to answer, but I have tried the same thing some time ago, and here is how I did it.

The video-decoding backend for OpenCV is actually ffmpeg, so all its facitilites are available in OpenCV as well. Not all the interface is exposed, and that adds some difficulties, but you can send the rtp stream address to OpenCV.

```
cap.open("rtp://xxx.xxx.xxx.xxx:1234");
```

Important: OpenCV is not able to access password-protected rtp streams. To do that, you would need to provide the username and the password, there is no API exposed for it.

edited Dec 10 '14 at 10:00

answered Nov 7 '11 at 10:14

sammy
13.5k 2 31 62

Could you please mention the command used to stream the file over to rtp? Did you use only ffmpeg for that? Or did you have to install something else? – user2565010 Nov 10 '14 at 15:00