

ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG  
— \* —

ĐỒ ÁN 3  
**STREAM VIDEO TỪ  
CAMERA IP**  
BỘ MÔN KỸ THUẬT MÁY TÍNH

Sinh viên thực hiện: Vương Chí Sơn  
MSSV: 20156407  
Lớp CN CNTT 3 - k60  
Giảng viên hướng dẫn: TS Nguyễn Hồng Quang

HÀ NỘI 5-2018



# Mục lục

1, Tìm hiểu chung .....	4
1.1, Cấu hình, dùng camera dưới góc độ người dùng cuối .....	4
1.2, Tìm hiểu sơ lược về chuẩn ONVIF.....	4
1.3, Tìm hiểu sơ lược về giao thức RTSP .....	5
2, Dùng TCP, UDP tạo gói tin RTSP .....	7
3, Dùng thư viện OpenCV .....	13
3.1, Tìm hiểu sơ lược về OpenCV .....	13
3.2, Cài đặt OpenCV .....	13
3.2, Viết chương trình.....	14
4, Hướng phát triển .....	19
Tài liệu tham khảo .....	21

# 1, Tìm hiểu chung

Đồ án tìm hiểu cách truyền dữ liệu từ 1 chiếc camera IP trực tiếp về server, phục vụ cho quá trình nhận diện bằng AI. Camera dùng trong đồ án do yoosee sản xuất, tài liệu mô tả của hãng: [http://support.hkvstar.com/file/Yoosee\\_Instruction\\_Manual.pdf](http://support.hkvstar.com/file/Yoosee_Instruction_Manual.pdf).



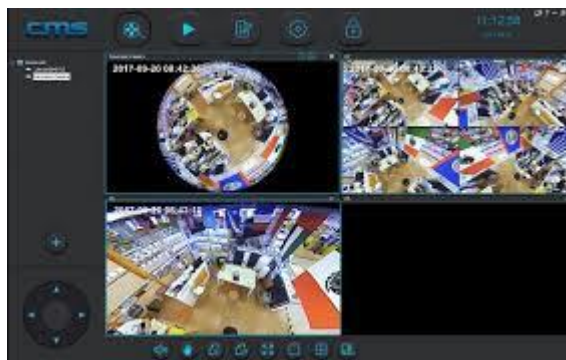
Theo tài liệu mô tả, camera dùng chuẩn ONVIF 2.4 Profile S, dùng giao thức RTSP để stream. Ta không thể lập trình trực tiếp trên camera, mà phải lập trình trên 1 máy tính khác. Lúc này, máy tính đóng vai trò là client, gửi các yêu cầu stream đến server camera, camera sẽ gửi luồng stream lại cho máy tính bằng giao thức RTSP. Nhận được gói tin RTSP, chương trình trên máy tính bóc tách, lấy phần dữ liệu, sau đó chuyển cho phần AI. Để kết thúc quá trình truyền, client gửi bản tin kết thúc đến camera, cả 2 đóng kết nối.

Trong khuôn khổ đồ án, em tìm hiểu 3 cách để stream dữ liệu về. Cách thứ nhất, tạo các bản tin RTSP, đóng gói vào UDP, trao đổi trực tiếp với camera. Cách thứ hai, dùng thư viện OpenCV. Cách thứ ba, dùng phần mềm Red5.

## 1.1, Cấu hình, dùng camera dưới góc độ người dùng cuối

Đầu tiên, cắm nguồn cho camera, và để ở nơi có sóng wifi mạnh, ổn định. Trên điện thoại, tải app Yoosee. Mở phần mềm này lên, chờ phần mềm dò được camera, sau đó nhập mật khẩu wifi để giúp camera dùng cho quá trình hoạt động, đặt tài khoản và mật khẩu.... Nếu không thiết lập được, giữ nút reset dưới đáy camera, và thực hiện lại.

Sau khi cấu hình, có thể dùng app Yoosee để theo dõi camera luôn. Trên máy tính, dùng ứng dụng CMS Client.



## 1.2, Tìm hiểu sơ lược về chuẩn ONVIF

Với sự bùng nổ mạng Internet, camera IP là một thiết bị tỏ ra hiệu quả, đơn giản, ít tốn kém khi chạy trên nền Internet. Người sử dụng có thể dùng camera để giám sát bất kỳ đâu, bất kỳ lúc nào, miễn là có mạng internet ổn định. Đã có rất nhiều hãng sản xuất phần cứng tham gia sản xuất camera IP. Tuy nhiên các loại camera IP của các hãng khác nhau lại dùng cách truyền dữ liệu khác nhau. Gây khó khăn trong việc thiết lập một hệ thống camera quan sát.

Để giải quyết vấn đề này, 27/08/2013 chuẩn ONVIF đã được công bố. Với nghiên cứu này, sẽ tích hợp hệ thống video và kiểm soát truy cập dựa vào IP từ một loạt video khác nhau và kiểm soát luôn vấn đề truy cập. Khả năng tương thích giữa các thiết bị camera có chuẩn

ONVIF trên camera quan sát IP sẽ giúp người dùng đơn giản hóa việc cài đặt và quản lý trên các đầu ghi hình IP camera khác nhau. Khi sử dụng chuẩn ONVIF có thể cung cấp thông tin về các điểm truy cập trong hệ thống lắp đặt camera quan sát, ngoài ra nó còn kiểm soát truy cập và báo động. Người dùng có thể truy cập các thiết bị an ninh cũng như kiểm soát video và âm thanh nhờ vào chuẩn ONVIF trên camera quan sát IP này.

ONVIF (Open Network Video Interface Forum) là một tiêu chuẩn để các sản phẩm IP trong việc giám sát hình ảnh và các vùng giám sát an ninh khác có thể giao tiếp với nhau. ONVIF một tổ chức thành lập năm 2008 bởi các thương hiệu lớn như Axis, Bosch, Sony. Là một tổ chức phi lợi nhuận, các thành viên thuộc diễn đàn ONVIF lập ra chuẩn này để các nhà sản xuất, nhà phát triển phần mềm, nhà tư vấn, tích hợp hệ thống, người dùng và các nhóm có quan tâm hướng đến 1 chuẩn giao thức IP nhất định.

Nhiệm vụ của ONVIF là giúp cho việc tích hợp giữa các thiết bị giám sát trở nên dễ dàng bằng cách sử dụng một tiêu chuẩn mở toàn cầu: tiêu chuẩn kết nối thiết bị giám sát trên nền lắp đặt camera quan sát IP giá rẻ. Tiêu chuẩn hóa giao thức giữa các thiết bị thu phát hình ảnh qua mạng. Làm cho tương thích giữa các sản phẩm thu phát hình ảnh qua mạng mà không phụ thuộc vào nhà sản xuất. Được sử dụng cho tất cả các công ty và tổ chức có quy mô đa quốc gia. Như vậy mục tiêu chính của ONVIF là định hướng các thiết bị an ninh hoạt động trên nền giao thức IP có thể giao tiếp với nhau thông qua các tiêu chuẩn có tính chất quốc tế. Hiểu một cách đơn giản, các hãng camera IP khác nhau (Camera IP Avtech, Camera IP Vantech, camera IP Questek...) hoạt động theo tiêu chuẩn ONVIF có thể kết nối với nhau trên cùng phần mềm quản lý giám sát hình ảnh tập trung.

### 1.3, Tìm hiểu sơ lược về giao thức RTSP

Real Time Streaming Protocol (RTSP) – Giao thức truyền tin thời gian thực là một giao thức điều khiển truyền thông mạng ở tầng ứng dụng được thiết kế để sử dụng trong các hệ thống giải trí và truyền thông để điều khiển máy chủ chứa các dữ liệu truyền tin đa phương tiện (streaming media). Giao thức này được sử dụng để thiết lập và điều khiển các phiên truyền thông giữa các trạm cuối. Các máy khách của các máy chủ truyền thông ban ra các lệnh kiểu VCR, chẳng hạn như chơi, dừng và tạm dừng, để điều khiển thời gian thực của các phương tiện truyền tin trực tuyến từ máy chủ tới máy khách (Video On Demand) hoặc từ máy khách đến máy chủ (Voice Recording) .

Việc truyền tải dữ liệu trực tuyến không phải là một nhiệm vụ của RTSP. Hầu hết các máy chủ RTSP sử dụng giao thức truyền tải thời gian thực (RTP) kết hợp với giao thức điều khiển thời gian thực (RTCP) để phân phối luồng truyền thông. Tuy nhiên, một số nhà cung cấp thực hiện các giao thức vận tải độc quyền. Phần mềm máy chủ RTSP của RealNetworks, thí dụ, cũng sử dụng Real Data Transport (RDT) độc quyền của RealNetworks.

RTSP hoạt động theo mô hình Client-Server. Camera đóng vai trò là Server, luôn chờ các request yêu cầu Streaming, khi có thì gửi lại các response trả lời chứa hình ảnh, âm thanh. Chương trình muốn lấy dữ liệu đóng vai trò là client. Các request của RTSP khá giống HTTP, gồm các phương thức (OPTIONS, DESCRIBE, SETUP, PLAY, PAUSE, RECORD,...), các mã trả về (200 OK,...), các trường header....

RSTP có các Control Sequence (Cseq), mỗi request sẽ được đánh số, response tương ứng có Cseq giống request. RTSP là giao thức statefull. Một phiên Streaming sẽ cần nhiều request-response thiết lập, điều khiển. RTSP dùng TCP để điều khiển, dùng UDP để truyền dữ liệu. Nó dùng cổng 554 cho cả TCP và UDP.

Ví dụ một vài RTSP Request:

### OPTIONS

OPTIONS request trả về các loại request mà server hỗ trợ.

```
C->S: OPTIONS rtsp://example.com/media.mp4 RTSP/1.0
      CSeq: 1
      Require: implicit-play
      Proxy-Require: gzipped-messages

S->C: RTSP/1.0 200 OK
      CSeq: 1
      Public: DESCRIBE, SETUP, TEARDOWN, PLAY, PAUSE
```

### DESCRIBE

```
C->S: DESCRIBE rtsp://example.com/media.mp4 RTSP/1.0
      CSeq: 2

S->C: RTSP/1.0 200 OK
      CSeq: 2
      Content-Base: rtsp://example.com/media.mp4
      Content-Type: application/sdp
      Content-Length: 460

      m=video 0 RTP/AVP 96
      a=control:streamid=0
      a=range:npt=0-7.741000
      a=length:npt=7.741000
      a=rtpmap:96 MP4V-ES/5544
      a=mimetype:string;"video/MP4V-ES"
      a=AvgBitRate:integer;304018
      a=StreamName:string;"hinted video track"
      m=audio 0 RTP/AVP 97
      a=control:streamid=1
      a=range:npt=0-7.712000
      a=length:npt=7.712000
      a=rtpmap:97 mpeg4-generic/32000/2
      a=mimetype:string;"audio/mpeg4-generic"
      a=AvgBitRate:integer;65790
      a=StreamName:string;"hinted audio track"
```

## SETUP

```
C->S: SETUP rtsp://example.com/media.mp4/streamid=0 RTSP/1.0
      CSeq: 3
      Transport: RTP/AVP;unicast;client_port=8000-8001

S->C: RTSP/1.0 200 OK
      CSeq: 3
      Transport: RTP/AVP;unicast;client_port=8000-
8001;server_port=9000-9001;ssrc=1234ABCD
      Session: 12345678
```

## PLAY

```
C->S: PLAY rtsp://example.com/media.mp4 RTSP/1.0
      CSeq: 4
      Range: npt=5-20
      Session: 12345678

S->C: RTSP/1.0 200 OK
      CSeq: 4
      Session: 12345678
      RTP-Info:
url=rtsp://example.com/media.mp4/streamid=0;seq=9810092;rtptime=34500
12
```

## PAUSE

```
C->S: PAUSE rtsp://example.com/media.mp4 RTSP/1.0
      CSeq: 5
      Session: 12345678

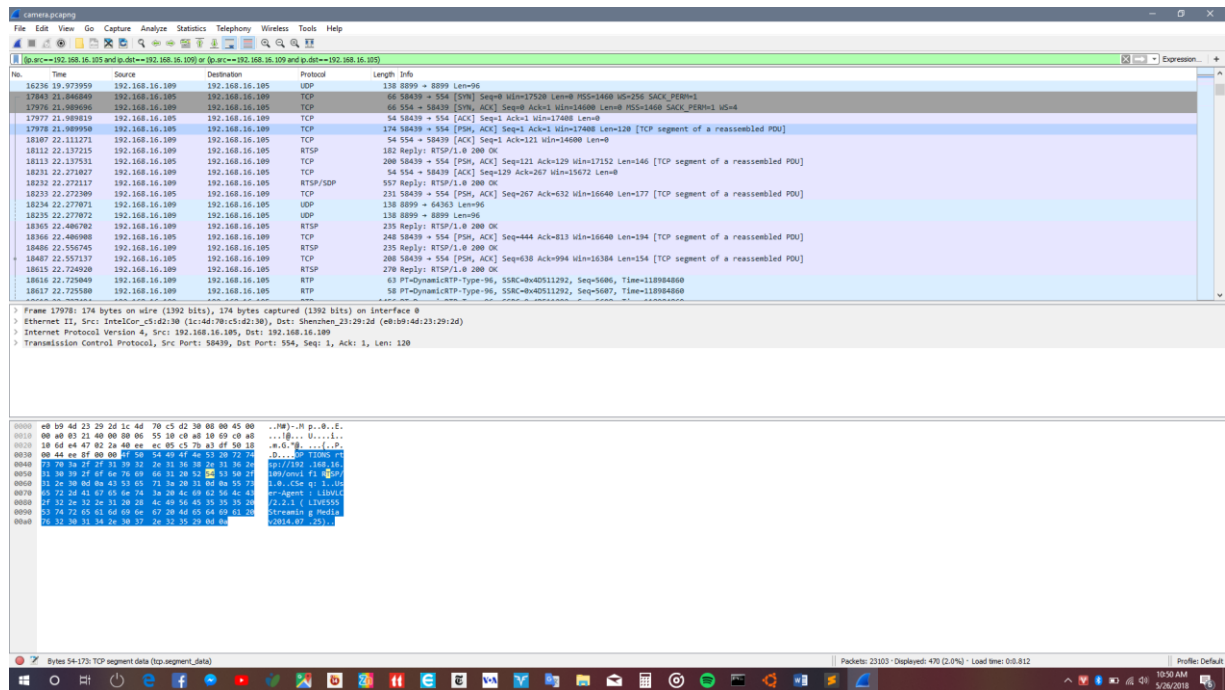
S->C: RTSP/1.0 200 OK
      CSeq: 5
      Session: 12345678
```

## 2, Dùng TCP, UDP tạo gói tin RTSP

Ý tưởng khá đơn giản, dùng cú pháp request thiết lập của RTSP, đóng gói vào bản tin TCP. Gửi cho camera. Sau đó, mở kết nối UDP, nhận gói tin RTSP chứa dữ liệu từ camera. Bóc tách, lấy được dữ liệu hình ảnh.

Để cho đơn giản, em dùng phần mềm CMS Client (đã cài ở phần 1), mở kết nối đến

camera. Sau đó dùng WireShark, bắt và phân tích các gói tin trao đổi giữa CMS và camera IP. Tiếp theo, viết chương trình “nhái” lại các gói tin này và trao đổi với camera.

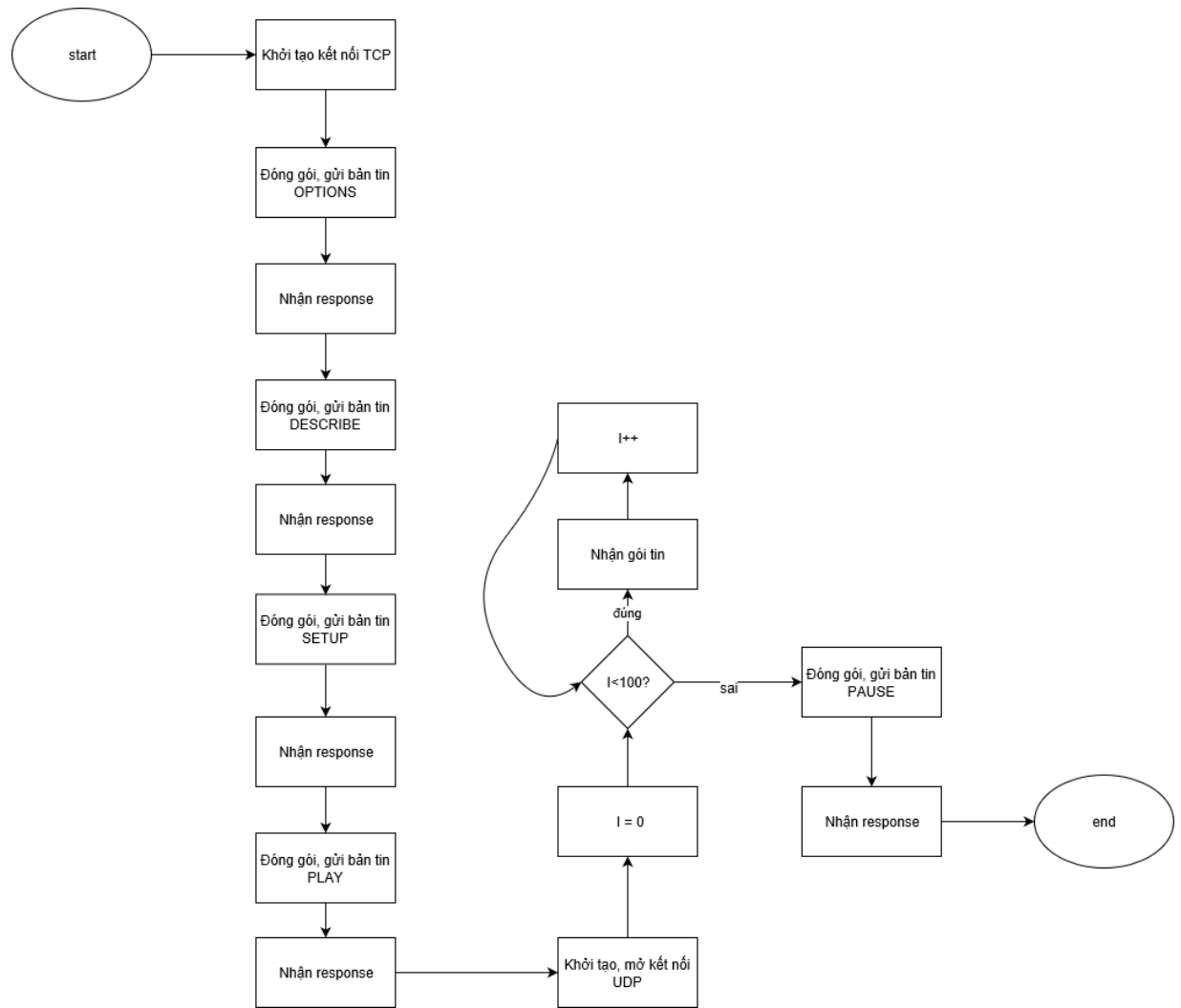


Hình trên là gói tin OPTIONS của CMS Client gửi đến Camera. Khi lập trình, chỉ cần copy nội dung gói tin, đưa vào xâu buffer, dùng TCP gửi buffer đó đến camera bằng thư viện sys/socket. Bằng cách này, chỉ cần áp dụng đúng thứ tự, nghe đúng cổng, thì chương trình hoàn toàn có thể trao đổi dữ liệu bình thường với Camera. Cách này đơn giản, thực hiện dễ dàng, nhanh chóng. Tuy nhiên, cách này khá cứng nhắc, khi thay đổi camera, lại phải làm lại.

Để minh họa, sau khi kết nối thành công đến camera, chương trình nhận 100 gói tin RTP, rồi tạm dừng bằng lệnh PAUSE.



Sơ đồ khối:



Mã nguồn:

```

#include <stdio.h>
#include <stdlib.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <string.h>
#include <unistd.h>

```

```

int main()
{

    const char* ip = "192.168.16.112";

    sockaddr_in addr;
    addr.sin_family = PF_INET;
    addr.sin_port = (in_port_t) htons(554);
    addr.sin_addr.s_addr = inet_addr(ip);
    int s = socket(PF_INET, SOCK_STREAM, 0);
    if (connect(s, (sockaddr*)&addr, sizeof(addr)) == -1)
    {
        printf("Connect error.\n");
        return 1;
    }
    else
        printf("Connected\n");

    //Gửi gói tin OPTIONS
    char msg[512];
    sprintf(msg, "OPTIONS rtsp://%s/onvif1 RTSP/1.0\r\nCSeq:
1\r\nUser-Agent: VCS\r\r\n\r\n", ip);

    if (send(s, msg, strlen(msg), 0) == -1)
    {
        printf("Error.\n");
        return 1;
    }
    char buf[256];

```

```

int res = recv(s, buf, sizeof(buf), 0);
if (res == 0 || res == -1)
{
    printf("Error.\n");
    return 1;
}

//DESCRIBE
sprintf(msg, "DESCRIBE rtsp://%s/onvif1 RTSP/1.0\r\nCSeq:
2\r\nUser-Agent: VCS\r\nAccept: application/sdp\r\n\r\n", ip);
send(s, msg, strlen(msg), 0);

recv(s, buf, sizeof(buf), 0);
printf("%s.\n", buf);

//SETUP
sprintf(msg, "SETUP rtsp://%s/onvif1/track1 RTSP/1.0\r\nCSeq:
3\r\nUser-Agent: VCS\r\nTransport: RTP/AVP;unicast;client_port=30004-
30005\r\n\r\n", ip);
send(s, msg, strlen(msg), 0);
recv(s, buf, sizeof(buf), 0);
printf("%s.\n", buf);

//PLAY
sprintf(msg, "PLAY rtsp://%s/onvif1 RTSP/1.0\r\nCSeq: 5\r\nUser-
Agent: VCS\r\nSession: 2bd7c725Range: npt=0.000-\r\n\r\n", ip);
send(s, msg, strlen(msg), 0);
recv(s, buf, sizeof(buf), 0);
printf("%s.\n", buf);

```

```

sockaddr_in recvAddr;
recvAddr.sin_family = PF_INET;
recvAddr.sin_port = htons(30004);
recvAddr.sin_addr.s_addr = htonl(INADDR_ANY);

int recvSocket = socket(PF_INET, SOCK_DGRAM, IPPROTO_UDP);

bind(recvSocket, (sockaddr*)&recvAddr, sizeof(recvAddr));

socklen_t len = sizeof(recvAddr);

//Nhận 100 gói tin RTP chứa dữ liệu
char rtpBuf[2048];
for(int i = 0; i < 100; i++)
{
    int res = recvfrom(recvSocket, rtpBuf, sizeof(rtpBuf), 0,
(sockaddr*)&recvAddr, &len);
    printf("%d byte\n%s\n", res, rtpBuf);
}

//PAUSE
sprintf(msg, "PAUSE rtsp://%s/onvif1 RTSP/1.0\r\nCSeq: 6\r\nUser-
Agent: VCS\r\n\r\n", ip);
send(s, msg, strlen(msg), 0);
recv(s, buf, sizeof(buf), 0);

return 0;
}

```

## 3, Dùng thư viện OpenCV

### 3.1, Tìm hiểu sơ lược về OpenCV

OpenCV là một thư viện mã nguồn mở hàng đầu cho thị giác máy tính (computer vision), xử lý ảnh và máy học, và các tính năng tăng tốc GPU trong hoạt động thời gian thực.

OpenCV được phát hành theo giấy phép BSD, do đó nó hoàn toàn miễn phí cho cả học thuật và thương mại. Nó có các interface C++, C, Python, Java và hỗ trợ Windows, Linux, Mac OS, iOS và Android. OpenCV được thiết kế để tính toán hiệu quả và với sự tập trung nhiều vào các ứng dụng thời gian thực. Được viết bằng tối ưu hóa C/C++, thư viện có thể tận dụng lợi thế của xử lý đa lõi. Được sử dụng trên khắp thế giới, OpenCV có cộng đồng hơn 47 nghìn người dùng và số lượng download vượt quá 6 triệu lần. Phạm vi sử dụng từ nghệ thuật tương tác, cho đến lĩnh vực khai thác mỏ, bản đồ trên web hoặc công nghệ robot.

OpenCV có class VideoCapture, class này giúp bắt luồng video theo từng frame một, hỗ trợ video lưu trên máy, lẫn cả video streaming.

### 3.2, Cài đặt OpenCV

Dưới đây, em xin trình bày cách cài OpenCV trên Ubuntu 16.10. Trên một vài phiên bản Ubuntu mới hơn, do Ubuntu thay đổi chế độ đồ họa, nên một số gói có thể bị lỗi.

Bước 1: Updating Ubuntu

```
$ sudo apt-get update
```

```
$ sudo apt-get upgrade
```

Bước 2: Cài đặt các gói

```
$ sudo apt-get install build-essential cmake git libgtk2.0-dev pkg-config libavcodec-dev libavformat-dev libswscale-dev
```

```
$ sudo apt-get install python3.5-dev python3-numpy libtbb2 libtbb-dev
```

```
$ sudo apt-get install libjpeg-dev libpng-dev libtiff5-dev libjasper-dev libdc1394-22-dev libeigen3-dev libtheora-dev libvorbis-dev libxvidcore-dev libx264-dev sphinx-common libtbb-dev yasm libfaac-dev libopencore-amrnb-dev libopencore-amrwb-dev libopenexr-dev libgstreamer-plugins-base1.0-dev libavutil-dev libavfilter-dev libavresample-dev
```

Bước 3: Clone OpenCV từ github

```
$ sudo -s
```

```
$ cd /opt
```

```
/opt$ git clone https://github.com/Itseez/opencv.git
/opt$ git clone https://github.com/Itseez/opencv_contrib.git
```

Bước 4: Build và cài đặt OpenCV

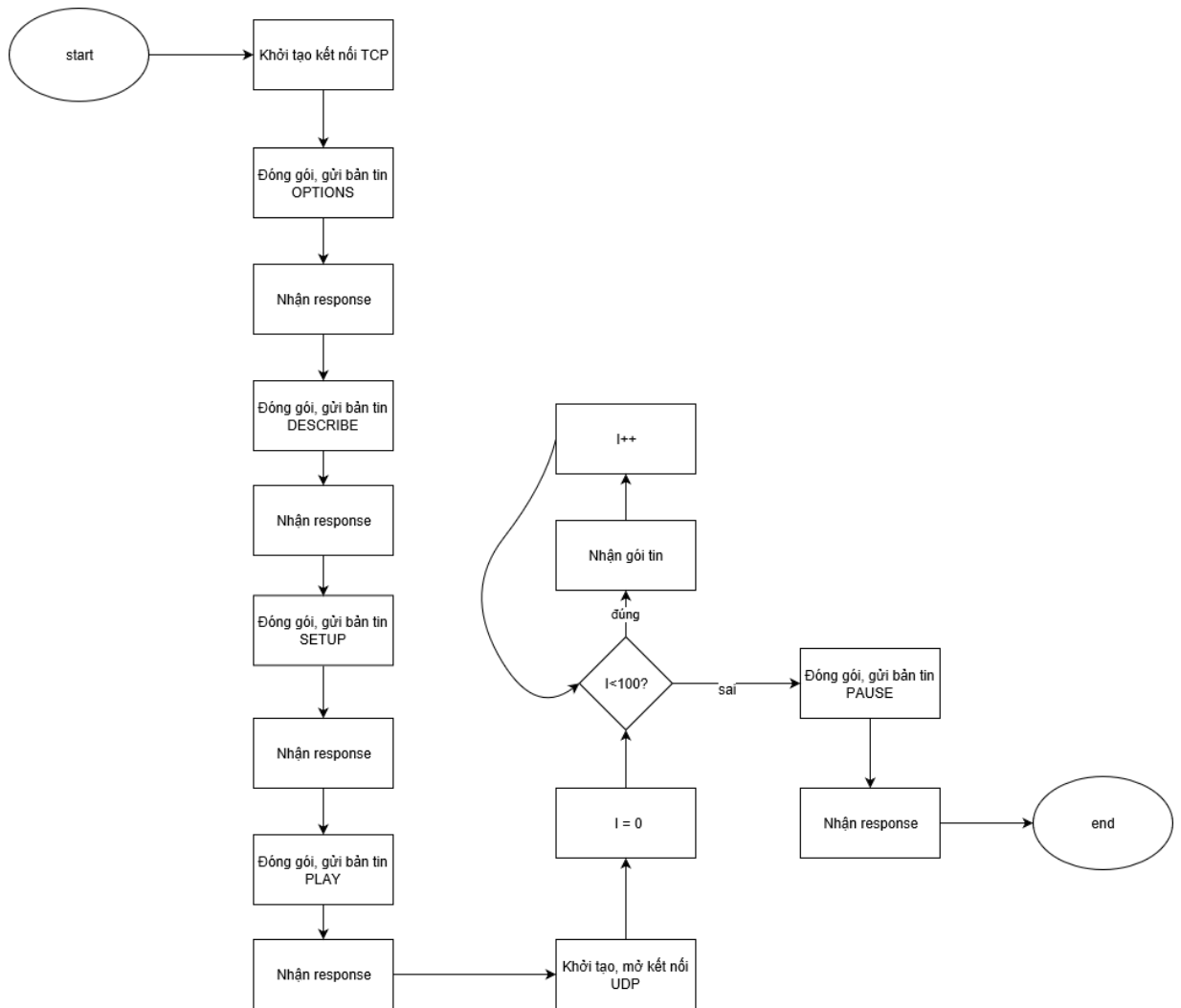
```
/opt$ cd opencv
/opt/opencv$ mkdir release
/opt/opencv$ cd release
/opt/opencv/release$ cmake -D BUILD_TIFF=ON -D WITH_CUDA=OFF -D
ENABLE_AVX=OFF -D WITH_OPENGL=OFF -D WITH_OPENCL=OFF -D WITH_IPP=OFF
-D WITH_TBB=ON -D BUILD_TBB=ON -D WITH_EIGEN=OFF -D WITH_V4L=OFF -D
WITH_VTK=OFF -D BUILD_TESTS=OFF -D BUILD_PERF_TESTS=OFF -D
CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local -D
OPENCV_EXTRA_MODULES_PATH=/opt/opencv_contrib/modules /opt/opencv/
/opt/opencv/release$ make -j4
/opt/opencv/release$ make install
/opt/opencv/release$ ldconfig
/opt/opencv/release$ exit
/opt/opencv/release$ cd ~
```

### 3.2, Viết chương trình

Ý tưởng:

- Đưa ra 3 lựa chọn định dạng đầu ra: video, image, pipe. Lựa chọn đầu tiên, lưu lại thành 1 đoạn video trong máy. Lựa chọn thứ 2, lưu thành 1 ảnh, ảnh này liên tục thay đổi theo các frame video truyền về. Lựa chọn thứ 3, lưu thành FIFO file, tạo thành pipeline trao đổi thông tin giữa 2 tiến trình.
- Dùng VideoCapture, mở kết nối RTSP, kiểm tra kết nối, tìm định dạng mã hóa, kích thước frame. Liên tục nhận luồng stream từ camera về, cắt thành từng frame một, lưu lại.

Sơ đồ khối:



Mã nguồn:

```

#include <iostream>
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include <opencv2/videoio.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv/highgui.h>
#include <opencv2/opencv.hpp>
  
```

```

using namespace cv;
using namespace std;

int main(int argc, char** argv)
{
    char ch;
    bool pipe = false, video = false, image = false;
    char *pipeName, *videoName, *imageName;

    //Tìm các option của người dùng p: FIFO file, v: video, i:
    image
    while((ch = getopt(argc, argv, "p:v:i:")) != EOF)
    {
        switch (ch)
        {
            case 'p':

                pipe = true;
                pipeName = optarg;
                break;
            case 'v':
                video = true;
                videoName = optarg;
                break;
            case 'i':
                image = true;
                imageName = optarg;
                break;
            default:

```



```

        fprintf(stderr, "Unknow option: '%s'\n", optarg);
        return 1;
    }
}

argc -= optind;
argv += optind;

if (argc != 1)
{
    fprintf(stderr, "capture: missing operand\nTry 'capture --help' for more information\n");
    return 1;
}

if (strcmp(argv[0], "--help") == 0)
{
    printf("capture [-p PIPENAME] [-v VIDEONAME] [-i IMAGENAME] source\n");
    return 0;
}

VideoCapture capture(argv[1]);

if (!capture.isOpened())
{
    fprintf(stderr, "Can't open\n");
    return 1;
}

//cout<< "Open" << endl;

```

```

//Video option
VideoWriter videoFile;
if (video)
{
    Size S = Size((int) capture.get(CAP_PROP_FRAME_WIDTH),
                  (int) capture.get(CAP_PROP_FRAME_HEIGHT));

    int ex = static_cast<int>(capture.get(CAP_PROP_FOURCC));
    //int codec = CV_FOURCC('M','P','E','G');
    videoFile.open(videoName, ex, capture.get(CAP_PROP_FPS), S,
true);
}

//Pipe option
VideoWriter pipeFile;
if (pipe)
{
    if (access(pipeName, F_OK) == -1)
    {
        //Create fifo file
        if (mkfifo(pipeName, 0666) == -1)
        {
            fprintf(stderr, "Can't create fifo file\n");
            return 1;
        }
    }

    Size S = Size((int) capture.get(CAP_PROP_FRAME_WIDTH),

```

```

        (int) capture.get(CAP_PROP_FRAME_HEIGHT));
    int ex = static_cast<int>(capture.get(CAP_PROP_FOURCC));
    pipeFile.open(pipeName, ex, capture.get(CAP_PROP_FPS), S,
true);
    }

    Mat frame;

    while(capture.read(frame))
    {
        if (video)
        {
            videoFile << frame;
        }
        if (image)
        {
            imwrite(imageName, frame);
        }
        if (pipe)
        {
            pipeFile << frame;
        }
    }

    return 0;
}

```

## 4, Hướng phát triển

Tiếp tục cải tiến, nâng cấp thêm cho phiên bản dùng OpenCV.

Phát triển thêm phiên bản dùng Red5. Red5 là phần mềm stream mã nguồn mở nổi tiếng. Nó có thể stream cùng lúc đến nhiều thiết bị với hiệu năng cao. Do tài liệu về red5 không nhiều, việc cài đặt không thuận lợi, kiến thức có hạn nên em chưa thể triển khai được. Trong tương lai, em sẽ cố gắng cài đặt, tùy chỉnh red5 thành 1 server stream.

Chuẩn ONVIF có hỗ trợ tìm thiết bị trong cùng mạng LAN, không cần phải nhập địa chỉ IP thủ công nữa. Chức năng này là rất cần thiết, bởi không phải lúc nào cũng dễ dàng biết địa chỉ IP của camera. Hơn nữa khi gặp sự cố, địa chỉ IP thay đổi, phần mềm có thể tự động dò tìm lại camera thì tiết kiệm thời gian hơn rất nhiều so với người quản trị phải dò lại IP một cách thủ công.

Bổ sung giao diện đồ họa để người quản trị dễ dàng truy cập, điều khiển.

## Tài liệu tham khảo

[1] Chuẩn ONVIF là gì? Tại sao phải sử dụng chuẩn ONVIF trong camera IP?

<http://www.cameravinhphat.com/tin-tuc/chuan-onvif-la-gi-tai-sao-phai-su-dung-chuan-onvif-trong-camera-ip-337.html>

[2] Real Time Streaming Protocol

[https://wikipedia.org/wiki/Real\\_Time\\_Streaming\\_Protocol](https://wikipedia.org/wiki/Real_Time_Streaming_Protocol)

[3] Hồ Sỹ Hùng, OpenCV và các ứng dụng của nó hiện nay

<https://techmaster.vn/posts/33943/opencv-va-cac-ung-dung-cua-no-hien-nay>

[4] How to Install OpenCV in Ubuntu 16.04 LTS for C / C++

<http://www.codebind.com/cpp-tutorial/install-opencv-ubuntu-cpp/>