opencv中通过VideoCaptrue类对视频进行读取操作以及调用摄像头，下面是该类的API。

1.VideoCapture类的构造函数：

C++: VideoCapture::VideoCapture()

C++: VideoCapture::VideoCapture(const string& filename)

C++: VideoCapture::VideoCapture(int device)

功能：创建一个VideoCapture类的实例，如果传入对应的参数，可以直接打开视频文件或者要调用的摄像头。

参数:

filename – 打开的视频文件名。

device – 打开的视频捕获设备id ，如果只有一个摄像头可以填0，表示打开默认的摄像头。

2.VideoCapture::open

功能：打开一个视频文件或者打开一个捕获视频的设备（也就是摄像头）

C++: bool VideoCapture::open(const string& filename)

C++: bool VideoCapture::open(int device)

参数:

filename – 打开的视频文件名。

device – 打开的视频捕获设备id ，如果只有一个摄像头可以填0，表示打开默认的摄像头。

通过对VideoCapture类的构造函数和open函数分析，可以发现opencv读入视频的方法一般有如下两种。比如读取当前目录下名为"dog.avi"的视频文件，那么这两种写法分别如下。

(1)先实例化再初始化：

VideoCapture capture;

capture.open("dog.avi");

(2)在实例化的同时进行初始化：

VideoCapture("dog.avi");

3.VideoCapture::isOpened

C++: bool VideoCapture::isOpened()

功能：判断视频读取或者摄像头调用是否成功，成功则返回true。

4.VideoCapture::release

C++: void VideoCapture::release()

功能：关闭视频文件或者摄像头。

5.VideoCapture::grab

C++: bool VideoCapture::grab()

功能：从视频文件或捕获设备中抓取下一个帧，假如调用成功返回true。（细节请参考opencv文档说明）

6.VideoCapture::retrieve

C++: bool VideoCapture::retrieve(Mat& image, int channel=0)

功能：解码并且返回刚刚抓取的视频帧，假如没有视频帧被捕获（相机没有连接或者视频文件中没有更多的帧）将返回false。

7.VideoCapture::read

C++: VideoCapture& VideoCapture::operator>>(Mat& image)

C++: bool VideoCapture::read(Mat& image)

功能：该函数结合VideoCapture::grab()和VideoCapture::retrieve()其中之一被调用，用于捕获、解码和返回下一个视频帧这是一个最方便的函数对于读取视频文件或者捕获数据从解码和返回刚刚捕获的帧，假如没有视频帧被捕获（相机没有连接或者视频文件中没有更多的帧）将返回false。

从上面的API中我们会发现获取视频帧可以有多种方法 ：

// 方法一

capture.read(frame);

// 方法二

capture.grab();

// 方法三

capture.retrieve(frame);

// 方法四

capture >> frame;

8.VideoCapture::get

C++: double VideoCapture::get(int propId)

功能：一个视频有很多属性，比如：帧率、总帧数、尺寸、格式等，VideoCapture的get方法可以获取这些属性。

参数：属性的ID。

属性的ID可以是下面的之一：

CV\_CAP\_PROP\_POS\_MSEC Current position of the video file in milliseconds or video capture timestamp.

CV\_CAP\_PROP\_POS\_FRAMES 0-based index of the frame to be decoded/captured next.

CV\_CAP\_PROP\_POS\_AVI\_RATIO Relative position of the video file: 0 - start of the film, 1 - end of the film.

CV\_CAP\_PROP\_FRAME\_WIDTH Width of the frames in the video stream.

CV\_CAP\_PROP\_FRAME\_HEIGHT Height of the frames in the video stream.

CV\_CAP\_PROP\_FPS Frame rate.

CV\_CAP\_PROP\_FOURCC 4-character code of codec.

CV\_CAP\_PROP\_FRAME\_COUNT Number of frames in the video file.

CV\_CAP\_PROP\_FORMAT Format of the Mat objects returned by retrieve() .

CV\_CAP\_PROP\_MODE Backend-specific value indicating the current capture mode.

CV\_CAP\_PROP\_BRIGHTNESS Brightness of the image (only for cameras).

CV\_CAP\_PROP\_CONTRAST Contrast of the image (only for cameras).

CV\_CAP\_PROP\_SATURATION Saturation of the image (only for cameras).

CV\_CAP\_PROP\_HUE Hue of the image (only for cameras).

CV\_CAP\_PROP\_GAIN Gain of the image (only for cameras).

CV\_CAP\_PROP\_EXPOSURE Exposure (only for cameras).

CV\_CAP\_PROP\_CONVERT\_RGB Boolean flags indicating whether images should be converted to RGB.

CV\_CAP\_PROP\_WHITE\_BALANCE Currently not supported

CV\_CAP\_PROP\_RECTIFICATION Rectification flag for stereo cameras (note: only supported by DC1394 v 2.x backend currently)

Note: 如果查询的视频属性是VideoCapture类不支持的，将会返回0。

9.VideoCapture::set

C++: bool VideoCapture::set(int propertyId, double value)

功能：设置VideoCapture类的属性，设置成功返回ture,失败返回false。

参数：第一个是属性ID，第二个是该属性要设置的值。

属性ID如下：

CV\_CAP\_PROP\_POS\_MSEC Current position of the video file in milliseconds.

CV\_CAP\_PROP\_POS\_FRAMES 0-based index of the frame to be decoded/captured next.

CV\_CAP\_PROP\_POS\_AVI\_RATIO Relative position of the video file: 0 - start of the film, 1 - end of the film.

CV\_CAP\_PROP\_FRAME\_WIDTH Width of the frames in the video stream.

CV\_CAP\_PROP\_FRAME\_HEIGHT Height of the frames in the video stream.

CV\_CAP\_PROP\_FPS Frame rate.

CV\_CAP\_PROP\_FOURCC 4-character code of codec.

CV\_CAP\_PROP\_FRAME\_COUNT Number of frames in the video file.

CV\_CAP\_PROP\_FORMAT Format of the Mat objects returned by retrieve() .

CV\_CAP\_PROP\_MODE Backend-specific value indicating the current capture mode.

CV\_CAP\_PROP\_BRIGHTNESS Brightness of the image (only for cameras).

CV\_CAP\_PROP\_CONTRAST Contrast of the image (only for cameras).

CV\_CAP\_PROP\_SATURATION Saturation of the image (only for cameras).

CV\_CAP\_PROP\_HUE Hue of the image (only for cameras).

CV\_CAP\_PROP\_GAIN Gain of the image (only for cameras).

CV\_CAP\_PROP\_EXPOSURE Exposure (only for cameras).

CV\_CAP\_PROP\_CONVERT\_RGB Boolean flags indicating whether images should be converted to RGB.

CV\_CAP\_PROP\_WHITE\_BALANCE Currently unsupported

CV\_CAP\_PROP\_RECTIFICATION Rectification flag for stereo cameras (note: only supported by DC1394 v 2.x backend currently)

至此，视频捕获类VideoCapture的API介绍完了，下面是一个应用示例：

#include <iostream>

#include <opencv2/core/core.hpp>

#include <opencv2/highgui/highgui.hpp>

int main(int argc,char\* argv[])

{

cv::VideoCapture capture(argv[1]);

if(!capture.isOpened())

{

std::cout<<"video not open."<<std::endl;

return 1;

}

//获取当前视频帧率

double rate = capture.get(CV\_CAP\_PROP\_FPS);

//当前视频帧

cv::Mat frame;

//每一帧之间的延时

//与视频的帧率相对应

int delay = 1000/rate;

bool stop(false);

while(!stop)

{

if(!capture.read(frame))

{

std::cout<<"no video frame"<<std::endl;

break;

}

//此处为添加对视频的每一帧的操作方法

int frame\_num = capture.get(CV\_CAP\_PROP\_POS\_FRAMES);

std::cout<<"Frame Num : "<<frame\_num<<std::endl;

if(frame\_num==20)

{

capture.set(CV\_CAP\_PROP\_POS\_FRAMES,10);

}

cv::imshow("video",frame);

//引入延时

//也可通过按键停止

if(cv::waitKey(delay)>0)

stop = true;

}

//关闭视频，手动调用析构函数（非必须）

capture.release();

return 0;

}