Web camera 相關處理狀況

Logitech c300規格: <https://www.cnet.com/products/logitech-c300-black/>

影片解析度格式參考: <https://www.reneelab.net/2k-4k-video-resolution.html#B>

3GP格式：176\*144

FLV格式：320\*240、640\*480

WMV格式：320\*240、640\*480

RMVB格式：640\*480、800\*600

MOV格式：320\*240、640\*480、768\*576

MP4格式：320\*240、480\*360、640\*480、854\*480、1280\*720、1920\*1080、3840\*2160

AVI格式：320\*240、640\*480、768\*576、1280\*720、1440\*1080、1920\*1080、3840\*2160

**4:3寬高比的解析度**— 640\*480、800\*600、960\*720、1024\*768、1280\*960、1440\*1080、1600\*1200、1920\*1440、2048\*1536等。

| **4:3常用解析度** | **對應的裝置** |
| --- | --- |
| 320\*240 | One Seg、舊Android手機、QVGA (1/4 VGA) |
| 640\*480 | VGA.MAC 13寸 |
| 720\*483 | NTSC |
| 768\*576 | PAL |
| 800\*600 | SVGA (Super-VGA) |
| 832\*624 | MAC 16英寸 |
| 1024\*768 | XGA (MAC 19英寸) |
| 1280\*960 | Quad-VGA |
| 1440\*1080 | Hi-Vision |
| 1600\*1200 | UXGA |
| 2048\*1536 | QXGA |

**16:9寬高比的解析度** — 1024\*576、1152\*648、1280\*720、1366\*768、1600\*900、1920\*1080、2560\*1440、3840\*2160等。

| **16:9常用解析度** | **對應的裝置** |
| --- | --- |
| 854\*480 | FWVGA |
| 1024\*576 | WSVGA |
| 1136\*640 | iPhone 5、5S、5C |
| 1280\*720 | 高清電視、Android手機 |
| 1920\*1080 | 全高清電視、Android手機、全高清Hi Vision |
| 2048\*1152 | QWXGA |
| 2560\*1440 | WQHD |
| 3200\*1800 | WQHD+ |
| 3840\*2160 | 4K電視 |

**16:10寬高比的解析度** — 1280\*800、1440\*900、1680\*1050、1920\*1200、2560\*1600等。

| **16:10常用解析度** | **對應的裝置** |
| --- | --- |
| 1280\*800 | WXGA |
| 1440\*900 | WXGA+ |
| 1680\*1050 | WSXGA+ |
| 1920\*1200 | WUXGA、Android平板電腦 |
| 2560\*1600 | WQXGA |
| 2880\*1800 | WQXGA |
| 3840\*2400 | QUXGA |

I420: 未压缩YUV颜色编码

PIMI: MPEG-1编码

XVID: MPEG-4编码

cv2.cv.CV\_FOURCC('P', 'I', 'M', '1') = MPEG-1 codec

cv2.cv.CV\_FOURCC('M', 'J', 'P', 'G') = motion-jpeg codec

cv2.cv.CV\_FOURCC('M', 'P', '4', '2') = MPEG-4.2 codec

cv2.cv.CV\_FOURCC('D', 'I', 'V', '3') = MPEG-4.3 codec

cv2.cv.CV\_FOURCC('D', 'I', 'V', 'X') = MPEG-4 codec

cv2.cv.CV\_FOURCC('U', '2', '6', '3') = H263 codec

cv2.cv.CV\_FOURCC('I', '2', '6', '3') = H263I codec

cv2.cv.CV\_FOURCC('F', 'L', 'V', '1') = FLV1 codec

# I420-avi=>cv2.cv.CV\_FOURCC('X','2','6','4');

# MP4=>cv2.cv.CV\_FOURCC('M', 'J', 'P', 'G')

<https://t.codebug.vip/questions-145967.htm>

* 你可以使用 cv2.VideoWriter() .這是函式頭
  1. cv2.VideoWriter([filename, fourcc, fps, frameSize[, isColor]])

例如，將FPS設置為30

* 1. cv2.VideoWriter('output.avi', self.codec, 30, (frame\_width, frame\_height))

這是一个用於从網路摄像頭流式傳輸和記錄幀的小部件。

* 1. from threading import Thread
  2. import cv2
  3. class WebcamRecordWidget(object):
  4. def \_\_init\_\_(self, src=0):
  5. # Create a VideoCapture object
  6. self.capture = cv2.VideoCapture(src)
  7. # Default resolutions of the frame are obtained (system dependent)
  8. self.frame\_width = int(self.capture.get(3))
  9. self.frame\_height = int(self.capture.get(4))
  10. # Set up codec and output video settings
  11. self.codec = cv2.VideoWriter\_fourcc('M','J','P','G')
  12. self.output\_video = cv2.VideoWriter('output.avi', self.codec, 30, (self.frame\_width, self.frame\_height))
  13. # Start the thread to read frames from the video stream
  14. self.thread = Thread(target=self.update, args=())
  15. self.thread.daemon = True
  16. self.thread.start()
  17. def update(self):
  18. # Read the next frame from the stream in a different thread
  19. while True:
  20. if self.capture.isOpened():
  21. (self.status, self.frame) = self.capture.read()
  22. def show\_frame(self):
  23. # Display frames in main program
  24. if self.status:
  25. cv2.imshow('frame', self.frame)
  26. # Press Q on keyboard to stop recording
  27. key = cv2.waitKey(1)
  28. if key == ord('q'):
  29. self.capture.release()
  30. self.output\_video.release()
  31. cv2.destroyAllWindows()
  32. exit(1)
  33. def save\_frame(self):
  34. # Save obtained frame into video output file
  35. self.output\_video.write(self.frame)
  36. if \_\_name\_\_ == '\_\_main\_\_':
  37. webcam\_record\_widget = WebcamRecordWidget()
  38. while True:
  39. try:
  40. webcam\_record\_widget.show\_frame()
  41. webcam\_record\_widget.save\_frame()
  42. except AttributeError:
  43. pass
* 11月前

2 #

OpenCV自動選擇第一个可用的捕获後端（參见此處）.可能是它没有自動使用V4L2.

同時設置两个 -D WITH\_V4L=ON   和 -D WITH\_LIBV4L=ON   如果从源頭建設.

可能，OpenCV選擇的畫素格式不支援您想要的分辨率.在Linux上你可以使用 v4l2-ctl --list-formats-ext   和 v4l2-ctl --all   看到設置.

為了設置要使用的畫素格式，請設置 CAP\_PROP\_FOURCC   捕获的屬性：

* 1. capture = cv2.VideoCapture(cam\_id, cv2.CAP\_V4L2)
  2. capture.set(cv2.CAP\_PROP\_FOURCC, cv2.VideoWriter\_fourcc('M', 'J', 'P', 'G'))

1. 首先，webcam本身就有解析度的問題，預設是640\*480 fps 5

但是這台webcam本身支援1280\*1024的情況下，能達到fps為30的視訊流及圖片輸出，在手動調整至1280\*1024的解析度下，只能達到15fps

這裡本身就是一個問題。

1. 影片錄製的時候有所謂的編碼與fps值問題，若沒有對應到相關的解析度與編碼，影片就算有輸出檔案也會造成無法開啟
2. 影片錄製的時候，音樂本身沒有錄進畫面裡，音樂本身也是
3. 針對某些對應按鍵所處理對應的行為，無法靈敏的偵測，這也是問題的所在之一
4. 在結合audio與video中 用opencv 與pyaudio 後與ffmpge做混和結合
5. 創建您自己的DirectShow視頻捕獲過濾器（有很多示例- [這](http://alax.info/blog/1223)是一個很好的例子），並處理其中的RTSP流。這樣，您可以自己實現穩定性

[http://netcamstudio.com](http://netcamstudio.com/)嘗試Moonware通用源過濾器。唯一的缺點是它只能創建同時發送視頻和音頻的視頻“複合”設備，而Skype只能看到視頻（我認為大多數應用程序都這樣做）。如果我發現拆分該流的簡單方法，則會在此處發布

我試圖將來自IP攝像機的RTSP流用作Windows上各種應用程序（例如Skype，Zoom，Microsoft Teams等）的視頻輸入源。

到目前為止，我發現的唯一解決方案是使用“ [**webcam 7**](http://www.webcamxp.com/home.aspx) ”，該應用程序可獲取RTSP流並創建一個虛擬的webcam驅動程序，該驅動程序在系統中註冊為webcam，然後任何應用程序都可以使用。不幸的是，此應用程序經常變得不穩定，並且可能會隨機崩潰。  
是否有其他/更好的方法來實現這一目標？

<https://stackoverflow.com/questions/26590174/convert-rtsp-stream-to-virtual-web-camera>