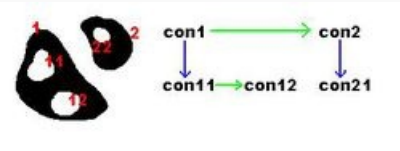
通过opencv videocapture捕获的视频流为 CV_8UC3

16位数据转8位数据

```
depth.convertTo(depth8, CV_8U, 255. / dmax, 0);
```

int cvFindContours(CvArr* image, CvMemStorage* storage, CvSeq** first_contour,
int header_size=sizeof(CvContour), int mode=CV_RETR_LIST,
int method=CV_CHAIN_APPROX_SIMPLE, CvPoint offset=cvPoint(0,0));
image
8比特单通道的源二值图像。非零像素作为1处理，0像素保存不变。从一个灰度图像得到二值图像的函数有：cvThreshold，cvAdaptiveThreshold和cvCanny。
storage
返回轮廓的容器。
first_contour
输出参数，用于存储指向第一个外接轮廓。
header_size
header序列的尺寸.如果选择method = CV_CHAIN_CODE, 则header_size >= sizeof(CvChain)；其他，则header_size >= sizeof(CvContour)。
mode
CV_RETR_EXTERNAL：只检索最外面的轮廓；
CV_RETR_LIST：检索所有的轮廓，并将其放入list中；
CV_RETR_CCOMP：检索所有的轮廓，并将他们组织为两层：顶层是各部分的外部边界，第二层是空洞的边界；
CV_RETR_TREE：检索所有的轮廓，并重构嵌套轮廓的整个层次。



蓝色表示v_next，绿色表示h_next
method
边缘近似方法（除了CV_RETR_RUNS使用内置的近似，其他模式均使用此设定的近似算法）。可取值如下：
CV_CHAIN_CODE：以Freeman链码的方式输出轮廓，所有其他方法输出多边形（顶点的序列）。
CV_CHAIN_APPROX_NONE：将所有的连码点，转换成点。
CV_CHAIN_APPROX_SIMPLE：压缩水平的、垂直的和斜的部分，也就是，函数只保留他们的终点部分。
CV_CHAIN_APPROX_TC89_L1，CV_CHAIN_APPROX_TC89_KCOS：使用the flavors of Teh-Chin chain近似算法的一种。
CV_LINK_RUNS：通过连接水平段的1，使用完全不同的边缘提取算法。使用CV_RETR_LIST检索模式能使用此方法。
offset
偏移量，用于移动所有轮廓点。当轮廓是从图像的ROI提取的，并且需要在整个图像中分析时，这个参数将很有用。
讨论部分cvDrawContours中的案例显示了任何使用轮廓检测连通区域。轮廓可以用于形状分析和目标识别——可以参考文件夹OpenCV sample中的squares.c

来源：<http://baike.baidu.com/link?url=8c1ijvxQ8i-iLvLeyLIzLr2pxMDMgk2hQ8hPgE0BU0o548tpUd0XvDiCo2MxTv03xGRQfcOAHJ6kFD4bx4J-xK>