Opencv3 编程入门笔记

By Xian2207, 13689903575, wszhangxian@126.com

1 第一章：邂逅 OpenCV，33 页： Canny 算子值的大小设置

语法

```
Canny(Detected_Edge, Detected_Edge, lowThreshold, highThreshold, Kernel_size)
```

例子

```
Canny(edge, edge, 3, 100, 3)
```

分析

edge 自己设定，低域值不能是负数，高域值不能太大，二者的意义在于高值时，图像几乎只有轮廓，但可能有断点，不连续；低值是为了链接这些断点，使其连续；Kernel_size 必须是 3，不可更改。知识点：原 RGB 图像转灰度图，然后高斯模糊，二值化处理，再使用 canny 算子逐像素计算，采双阈值法，即像素值大于 T2 时，判断是边缘点，小于 T1 时全部提出，置换为背景颜色；介于之间时，判断该像素周围 8 个像素点是否有大于 T2 的，如果有，说明该像素点是边缘点。

2 第二章：启程前的认知准备，45 页，头文件配置

分析：参考教材下载、安装、配置、调试，关键是#include <opencv2/opencv.hpp>包含了所有与 opencv 有关的头文件，如 core.hpp, improc.hpp, highgui.hpp, etc., 所以在 VS 上写头文件的时候，只需一个#include “opencv2/opencv.hpp”来调用 opencv 程序。至少 2.4.9 版本如此，之后的各种高级版本也如此。

3 第二章 56 页，main 函数

问题：int main(), int main(int argc, char* argv, char** env)

分析：VS 查看各类源码定义时，经常看到 main(int argc, char* argv)类函数，事实上我们偏爱 int main()这个。后两个来自 UNIX 和 LINUX 的习惯。argc - 记录使用 main 函数命令行参数的个数；argv - 是指向字符串数组的指针；env - 是指向字符串数组每个元素 value 的指针。总之，VS 环境下，新手一律写成 int main()没问题，编译通过且 debug 也能正常运行。而 int main(int argc, char* argv) 或者 int main(int argc, char** argv)是为了向 UNIX/LINUX 人展示代码的可读性，或者展示你确实有严格的书写习惯。

4 调用函数，记得带括号

错误代码：

```
VideoCapture capture;
capture.open(“filename”, filename.format);
capture.release;
```

正确代码：

```
VideoCapture capture;
capture.open(“filename”, filename.format);
capture.release();
```

分析：release 是释放或关闭某项函数调用的库函数，写完后，一定要加()。

5 第二章 60 页，printf 和 fprintf 区别

例子：

A: printf

printf(“\n\t %d %d\n”, a, b);

B: fprintf

FILE* stream;

fprintf(stream, “\n\t %d %d\n”, a, b);

分析：printf 用以非文件流的打印输出；fprintf 用以文件流的打印输出。比如打开文件，写入数据，而 printf 不具备这个特点。

6 第二章 68 页，输出图像到文件，imwrite()函数

例子：

bool imwrite(const string& filename, InputArray img, const vector<int>& params=vector<int>())

分析：第一个是引用的文件名，如”1.jpg”;第二个是 Mat 类型的图像数据，如 Mat frame; 第三个向量常量，引用 params，由于有=vector<int>(), 所以是特定格式保存的参数编码。一般不管，用到的话，查书就行。

7 第二章 71 页，imwrite 函数代码小心

(1) 失误例子：

Imwrite(“由 imwrite 生成的图片.jpg”, image);

正确代码：

Imwrite(“DotaLogo.jpg”, image)

建议：写代码的时候，要么纯中文，要么纯英文，否则会以为.jpg 是描述，其实它是文件名的格式后缀，例如”filename.jpg”。

8 Rect 和 addWeighted 用法

Mat imageROI;

ImageROI = srcImage(Rect(1300, 700, logoImage.cols, logoImage.rows));

addWeighted(imageROI, 0.7, logo, 0.3, 0., imageROI);

分析：首先，Rect(int x, int y, imageName.cols, imageName.rows)中 x，y 指 srcImage 的坐标位置，x 的方向是从左往右，y 的方向是从上到下。logoImage.cols, logoImage.rows 分别指融入图像的宽度和高度，这里代码是选择原大小融入，但也可以用 logoImage.cols/2; logoImage.rows/10 改变原来的大小。但注意：如果 x，y 的坐标超出了原图尺寸，Rect 的操作就会有误，代码编译成功，但 debug 运行不会成功的。所以有时候要通过“试”,找出最佳位置;其次,注意 addWeighted(srcImage, weighted_value1, src2Image, weighted_value2, gama_value, outImage)。其中 weighted_value1+weighted_value2 = 1，game_value 一般默认 0，outImage 还是 srcImage1，也就是融合图。

8 第二章 75 页，滑动条创建

例子：

...

g_srgImage1 = imread(“1.jpg”);

g_srgImage2 = imread(“2.jpg”);

...

createTrackbar(TrackbarName, WINDOW_NAME, &g_nAlphaValueSlider, g_nMaxAlphaVlaue,

on_Trackbar)
...
分析：运行最易犯错的地方是，上面两个图片大小不一致，导致编译通过，但 debug 无法完整运行。所以，解决办法是：要么复制同一张照片，要么网上去找大小尺寸相同的两个照片。

9 第二章 77 页，RNG g_rng(12345)，Rect 类，Mat 类

分析：

RNG 指 Random Number Generator，随机种子生成器，里面可以填写负数，正数，小数等等，意思告诉编译器，每次随机产生的都是固定的一组种子；如果每次循环都要随机，用 RNG rng((int)time(0))。如果还不清楚，建议看看 MATLAB rng 用法，就可以想起当初自己怎么随机加噪做历史拟合；

Rect 类用法：假设 rect 是类名，rect(x, y, width, height), rect.tl(), rect.area(), rect.br(), rect.width(), rect.height(), rect.contains(Point(x,y));意思是矩形的 top left, 面积，bottom right, 宽度，高度，矩形是否包含点（x,y）。总之，Rect 是关于矩形坐标，面积，维数参数的类函数；

Mat 类用法：Mat::Mat(int rows, int cols, int type)，创建矩阵用的，类似于 MATLAB，但要注意 int type，比如 Mat::matrix1(3, 3, CV_8UC3)，第三个参数是 type。

10 第二章 78 页 *(cv::Mat*) param

例子:

```
void on_MouseHandle(int event, int x, int y, int flags, void* param)
Mat& image = *(cv::Mat*) param;
```

分析：第一个&是引用 image，对 image 修改，原图 image 也会改变；第二个 Mat*很少见，因为 param 是特殊格式图片，所以 Mat*是告诉我们，无论 param 是什么格式图片，都可以在 Mat 类中进行操作。由于 param 是指针指向的类型，要想赋值给 image, 必须*(cv::Mat*) param。意思是把*param 指针指向的图片传递给 image, Mat 引用 image，节约内从空间，提高代码效率。

11 回调函数的用途

解答：允许用户在同一幅图上进行反复操作，如鼠标画图，滑动条来回展示图片效果等。

12 第六章 171 页，static void 函数和变量申明

```
#include<iostream>
#include”FUNC.h”
static void Func1(int argc, char* argv);
void Func2(int argc, char* argv);
```

分析：Func1 不论其他文件有无申明，如 external static void Func1(...)，Func1 不可用；而 Func2 只要有外部申明，不单在这个文件中可用，在另外一个文件中也可以用。

13 第六章 172 页,createTrackbar(“内核值”, “【1】<2>均值滤波”, &g_nBoxFilterVlaue.....)

解答：注意一下红色的“内核值”，如果换成“第一个标识-内核值”稍微长一点的，Trackbar 可能无法显示，因为第一个 stringName 不能太长。

14 第六章 171 页，static void on_BoxFilter(int, void*)可否写成...on_BoxFilter(int, char*)?

解答：不可以，凡是回调函数(TrackbarCallback)，只能用 void on_func(int, void*)固定形式，

最多在函数头前面加个 static 就 OK 了。

15 代码如下：

```
.....
imshow("结果帧", resultFrame);
//waitKey(3000);
if (waitKey(1000.0 / FPS) == 27) { //FPS 是浮点型，所用 1000.0
    cout << "ESC 退出" << endl;
    break;
}
backgroundFrame = currentFrame.clone();
int64 timeDuration = getTickCount() - timeStart; //计时结束
int64 framePerSecs = getTickFrequency() / timeDuration;  //当前的帧率
cout << "当前帧率：  " << framePerSecs << endl;
```

解答：如果不 comment 掉 waitKey(3000)，当前帧率将永远显示是 0，且图像显示会停留 3000 毫秒，不论循环多少次。因为 waitKey 打乱了计算机的正常计时，所以计算机只能用 0 来显示。做动态识别，最好不要用 waitKey，一是加速运行和追踪；二是避免计时器的紊乱。

16 FPS 代码配套不可或缺：

```
double FPS = capture.get(CV_CAP_PROP_FPS);
capture >> foregroundFrame;
if (waitKey(1000.0 / FPS) == 27) { //FPS 是浮点型，所用 1000.0
    cout << "ESC 退出" << endl;
    break;
}
imshow("原始帧", foregroundFrame);
```

解答：FPS 和下面的 waitKey(1000.0/FPS)是配套的，如果缺其中一个，imshow 显示的就是全灰度图，下一帧循环仍然是全灰度图。如果本身照片是彩色的，显示必须是彩色的，所以注意写代码的时候要配套。

17 指针高效遍历二维数组像素点

效率低代码：

```
for (int  i = 0; i<gray_diff1.rows; i++) {
    for (int  j = 0; j<gray_diff1.cols; j++){
        if (abs(gray_diff1.at<unsigned  char>(i,j)) >= 30) {
            gray_diff1.at<unsigned  char>(i, j) = 255;
        }
        else  {
            gray_diff1.at<unsigned  char>(i,j) = 0;
        }
            if (abs(gray_diff2.at<unsigned  char>(i, j)) >= 30) {
                gray_diff2.at<unsigned  char>(i, j) = 255;
        }
        else  {
```

```
                gray_diff2.at<unsigned  char>(i,j) = 0;
            }
        }
}
```

效率高代码：

```
for (int i = 0; i < grayDiff1.rows; ++i) {
    for (int j = 0; j < grayDiff1.cols; ++j) {
        uchar* value1;
        value1 = grayDiff1.ptr<uchar>(i, j);
        int a = *value1;
        debug_addShow(jniEnv0, "a \n");
        a = (a > 30) ? 255 : 0;

        uchar* value2;
        value2 = grayDiff2.ptr<uchar>(i, j);
        int b = *value2;
        b = (b > 30) ? 255 : 0;
        debug_addShow(jniEnv0, "b \n");
    }
}
```

分析：区别在于指针遍历像素点，比 opencv.at 函数、迭代器（类似 vector<int>::iterator iterator）更高效。

18 安卓平台无法使用三帧差法的问题

分析：安卓系统计算能力有限，即使像素遍历用指针，仍然无法摆脱 for 循环的大量计算。网上显示只能用背景差法，帧差法来及检测运动目标。

19 visual studio 平台运行视频流代码，程序编译通过，运行停不下来

代码：

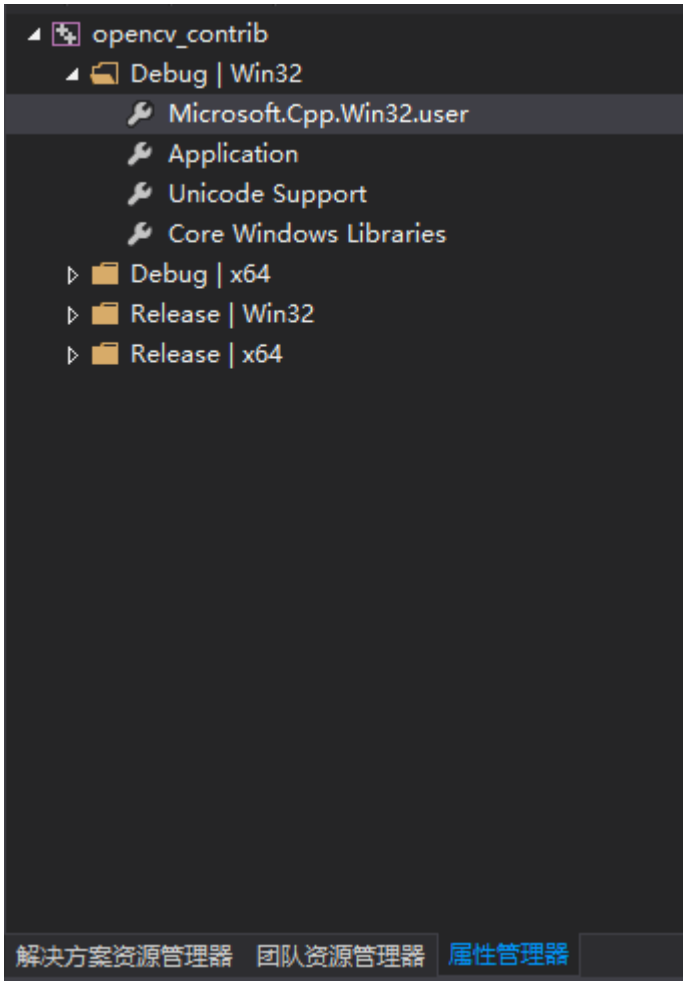
```
for (int i = 0; i < frameCount; i++) {
        capture >> image2;

        if (image2.empty() || waitKey(pauseTime) == 27) {
            break;
        }

        cvtColor(image2, image2Gray, CV_BGR2GRAY);
       calcOpticalFlowPyrLK(image1Gray, image2Gray, pts1, pts2, status, err, Size(15, 15),
3);

        for (int j = 0; j < pts2.size(); j++) {
            circle(image2, pts2[j], 1, Scalar(0, 0, 255), 2);
            line(image2, ptsCopy[j], pts2[j], Scalar(255, 0, 0), 2);
        }
```

```
imshow("现在角点特征光流", image2);
swap(pts1, pts2);
image1Gray = image2Gray.clone();
```

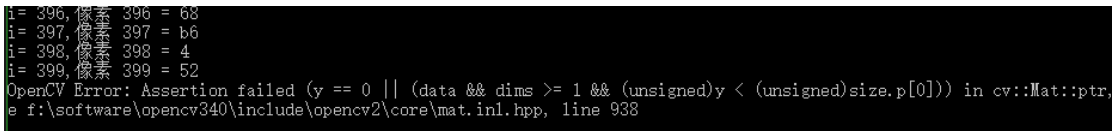
分析：画红线代码位置很关键，如果放到黄色代码后面，程序编译可以通过，但运行不会停止！不管用户强制 break，还是 for 循环判断条件已经满足。为避免引起代码今后测试，建议将判断条件写到上面。

20 卸载 OPENCV 后，安装新的 VS 版本，出现"link error: 无法打开文件.....opencvml.dll"等库

答：原因在于 VS 的附加依赖项仍然加载 opencv 的 dll，没有删除干净。如何删除，打开 VS -> 视图 -> 其他窗口 -> 属性管理器 -> 分别右键 MicrosoftCpp.win32.user 和 MicrosoftCpp.win64.user，把里面的附加依赖库，包含目录，库目录都清理一下。



21 opencv Assertion Failed Error (y==0 || dim or data > 1 && data < data length)



答：图像宽高访问时越界。

22 指针遍历像素，uchar 转 unsigned char*

正确代码如下：

```
unsigned char *mat2uchar(){
    Mat img = imread("2.jpg");
    assert(img.data);

    int channels = img.channels();
    int col = 10;  //故意取 10 列；
    int row = 10;  //故意取 10 行；

    if (img.isContinuous())
    printf("is Continuous!\n");

    int dataLength = channels*row*col;

    unsigned char *p;
    p = (unsigned char*)malloc(dataLength);

    uchar *data = img.data;                          //注意这步很关键

    for (int i = 0; i < dataLength; i++) {
        p[i] = data[i];
        printf("i= %d,p[%d] = %uc\n", i, i, p[i]);  //注意打印 unsigned char 用%uc
    }
    //free(p);                                       //此处调试时，释放 p 指针是可以的
    return p;
}

void main(){
    float *fp32;
    unsigned char *imgresized;

    imgresized = mat2uchar();

    fp32 = (float*)malloc(sizeof(*fp32)*3 * 10 * 10); //注意紫色部分，sizeof(*fp32) = 4;
                                                      因为涉及 unsigned char* 转 float*
    for (int i = 0; i < 300; i++) {
        fp32[i] = imgresized[i];
        printf("imgresized[%d] = %f\n", i, fp32[i]);
    }

    free(imgresized);
    getchar();
}
```

参考：(1)https://stackoverflow.com/questions/13947697/mat-to-unsigned-char;

(2)https://blog.csdn.net/jameshater/article/details/50119631;

Android Studio Opencv 学习心得

1 安卓 Studio 中 Bitmap 转 Mat

答：代码如下

```
# Android Studio 头文件需要 import：import org.opencv.android.Utils;
# 代码处写：
Bitmap bmp = UiModule.previewView0.getBitmap(160, 90);
Mat mat = new Mat(bmp.getWidth(),bmp.getHeight(),CvType.CV_8UC4);
Bitmap tmp = bmp.copy(Bitmap.Config.ARGB_8888, true); //ARGB 格式的 Mat
Utils.bitmapToMat(tmp, mat);
```

分析：ARGB_8888 是 opencv 的 RGBA 格式，带有透明度；RGB_565 是 opencv 的 RGB 格式，没有透明度信息。

2 安卓 studio 中使用 Mat(Range(rowStart, rowEnd), Range(colStart, colEnd))是错误的怎么办

分析：使用 Mat.submat(rowStart, rowEnd, colStart, colEnd)即可解决图片裁剪问题。

3 安卓 studio 如何 BGRA 转 GRAY?

答：Mat 一定要初始化，在安卓正确初始化方式是 Mat xxx = new Mat(); 如果 Mat xxx = new Mat(cols,rows, CvType.8UC1)是错的；注意下面这样正确代码

```
Mat gray_xx = new Mat();
Imgproc.cvtColor(src, gray_xx, Imgproc.COLOR_BGRA2GRAY);
```

4 onPreviewFrame()如何被调用

答：只要没有函数,i.e. getOneFrame(true)，去 call 醒 onPreviewFrame()函数，它永远不会被调用，最好用 ctrl+P 组合键看看 getOneFrame 函数，也许 true/false 都能调用，很多 Java 工程师喜欢用 true/false 来 Delay 调用时间，true 是手动 Delay,false 是系统默认。

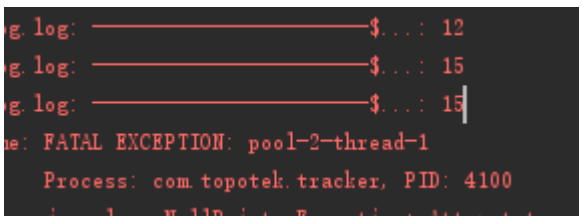
5 Asseration Failed (0<roi && 0< roi.x + roi.width....)越界错误

分析：图像 ROI 区域超过了图像尺寸大小，仔细检查一下涉及图片尺寸，画框，缩放还原等代码。

6 裁剪 ROI 检测后，框位置与原图不匹配

答：首先考虑加，其次考虑比例乘。

7 Pool 2-thread-1 报错



答：全局变量或静态变量只定义却没有初始化，程序运行时赋值导致线程占用（既要初始化，同时还要赋值）错误，如：

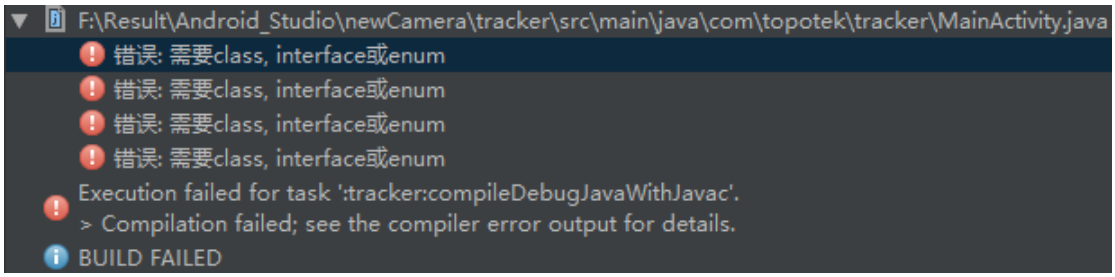
```
private static Rect box;
void(Rect rect){
    ....;
    box.x = rect.x;
    box.y = rect.y;
}
```

box 作为静态变量，没有初始化，直接赋值导致错误。也许是 opencv bug，Rect 定义的对象必须初始化才能赋值，Mat, int(float, double, char, etc), 布尔值等不需要。

8 ROI 取出来后，目标所在位置在原图的偏移量

答：Rect(x,y,width,height)其中 x 指的是横向坐标，宽的方向；y 指的是纵向坐标，列的方向。

9 出现下列错误

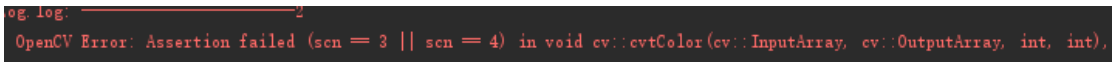


分析：某些参数（全局变量或静态变量）写在了函数体外，比如 a = system.clock()，a 没有指明类型。一般情况下，安卓会在代码下画红线提示。

10 安卓使用 opencv 的三个方式

答：一，使用 opencv for android 的包，它用 JNI 封装好的，直接在安卓环境下 import org.opencv.包名；二，使用 opencvSdk，它提供 C++的头文件和某些编译库文件，这样直接用 C++代码去调用 opencv 的函数，方法比第一种快；三，根据 opencv 的源码，自己编译成安卓的 Sdk 库，完了直接在安卓环境下调用 opencv 的函数。有点是能使用 contrib，opencv3.x 以后的新库，官方所不提供的，缺点是编译困难且不稳定。

11 出现 scn == 3 || scn == 4 错误



答：两幅图(Mat 类型)在相互传递时，灰度图和彩色图不能互传导致报错。此外注意，错误已经提示位置了，在 cvtColor(cv::InputArray, cv::OutputArray, int, int)处。

12 使用安卓做 Background Subtraction 的时候，Byte[] Array to Mat 如何实现？

答：错误代码

(1)引用的包：

```
import android.renderscript.RenderScript;
import android.renderscript.ScriptIntrinsicYuvToRGB;
import android.renderscript.Type;
```

```
import android.renderscript.Allocation;
import android.renderscript.Element;
```

(2)onCreate()函数里初始化

```
rs = RenderScript.create(this);
yuvToRgbIntrinsic = ScriptIntrinsicYuvToRGB.create(rs, Element.U8_4(rs));
```

(3)onPreviewFrame()函数里写

```
if (yuvType == null)
{
yuvType = new Type.Builder(rs, Element.U8(rs)).setX(frameData.length);
in = Allocation.createTyped(rs, yuvType.create(), Allocation.USAGE_SCRIPT);

rgbaType = new Type.Builder(rs,
Element.RGBA_8888(rs)).setX(frameWidth).setY(frameHeight);
out = Allocation.createTyped(rs, rgbaType.create(), Allocation.USAGE_SCRIPT);
}

in.copyFrom(frameData);
yuvToRgbIntrinsic.setInput(in);
yuvToRgbIntrinsic.forEach(out);
Bitmap bmpOut = Bitmap.createBitmap(MatWidth, MatHeight, Bitmap.Config.ARGB_8888);
out.copyTo(bmpOut);
Mat src = new Mat();
Utils.bitmapToMat(bmpOut, src);
```

即便用错误代码，Background Subtraction 做动态检测时仍然是对的，只不过不是适时检测，坑爹啊！害的我的逐行代码反向查错，直到代码源头才发现。

正确代码

```
Bitmap bmp = Bitmap.createBitmap(frameWidth, frameHeight, Bitmap.Config.ARGB_8888);
bmp.copyPixelsFromBuffer(ByteBuffer.wrap(frameData));
Mat src = new Mat();
Utils.bitmapToMat(bmp,src);
```

13 下列代码在安卓 C++环境下正确

C++代码

```
cvtColor(roi, gray, COLOR_BGRA2GRAY);
Rect searchWindow;
searchWindow.width = trackedBox.width * 3;
searchWindow.height = trackedBox.height * 3;
searchWindow.x = trackedBox.x + trackedBox.width * 0.5 - searchWindow.width * 0.5;
searchWindow.y = trackedBox.y + trackedBox.height * 0.5 - searchWindow.height * 0.5;
searchWindow &= Rect(0, 0, roi.cols, roi.rows);
```

```
Mat similarity;
matchTemplate(gray(searchWindow), model, similarity, CV_TM_CCOEFF_NORMED);
double error;
Point point;
minMaxLoc(similarity, 0, &error, 0, &point);
trackedBox.x = point.x + searchWindow.x;
trackedBox.y = point.y + searchWindow.y;
model = gray(trackedBox);
```

安卓代码移植也是对的：

```
Rect searchWindow = new Rect();
searchWindow.width = trackedBox.width*3 ;//3
searchWindow.height = trackedBox.height*3;//3
searchWindow.x = (int)Math.round(trackedBox.x + trackedBox.width * 0.5 - searchWindow.width
* 0.5);
searchWindow.y = (int)Math.round(trackedBox.y + trackedBox.height * 0.5 -
searchWindow.height * 0.5);

int x1 = Math.max(searchWindow.x,0);
int y1 = Math.max(searchWindow.y,0);
int SW = Math.min(searchWindow.x+searchWindow.width,roi_width) - x1;
int SH = Math.min(searchWindow.y+searchWindow.height,roi_height) - y1;
searchWindow.x = x1;
searchWindow.y = y1;

if( SW <= 0 || SH <= 0 )
searchWindow = new Rect(0,0,roi_width,roi_height);
Imgproc.matchTemplate(gray.submat(searchWindow), model, result,
```

但安卓却出现了报错

分析：1 template matching 不能像上面这样移植，必须按照安卓版的 opencv 去写；正确代码如下

```
Mat gray = new Mat(roi_height,roi_width,CvType.CV_8UC1);
Imgproc.cvtColor(roi,gray,Imgproc.COLOR_BGRA2GRAY);

Mat result = new Mat();
Imgproc.matchTemplate(gray, model, result, Imgproc.TM_CCOEFF_NORMED);

Point point;
MinMaxLocResult mmr = Core.minMaxLoc(result);
point = mmr.maxLoc;
// trackedBox.x = (int)point.x;
// trackedBox.y = (int)point.y;
// trackedBox.width = (int)point.x+ model.width(); 注意：point.x+model.width()的概念
```

```
// trackedBox.height= (int)point.x+ model.height();

trackedBox = new Rect(point,new Point(point.x+model.width(), point.y+model.height()));
model = gray.submat(trackedBox);
```

分析：//注释掉的语句和 trackedBox = new Rect(point,new Point(point.x+model.width(), point.y+model.height())); 不能互换，否则报错！因为 point.x+model.width 是代表点所在位置位于 point.x+model.width()大小的地方，比如 point.x=3,model.width()是 10，则 Point(point.x+point.width(),16)是指点 x 的坐标在 13，y 坐标在 16。而// trackedBox.width = (int)point.x+ model.width()则是指 trackedBox 的宽是 13，点和宽是不同的概念！

14 出现下列 pool-2-thread-4

```
g.log: ————————————H 36
e: FATAL EXCEPTION: pool-2-thread-4
   Process: com.topotek.tracker, PID: 4854
```

分析：原因可能出在图片尺寸或框的尺寸超过了原图的大小。

15 出现 assertion failed (ssize.area() in void ...)

```
OpenCV Error: Assertion failed (ssize.area() > 0) in void cv::resize(cv::InputArray, cv::OutputArray, cv::Size, double, double, int)
```

```
file /home/reports/ci/slave_desktop/50-SDK/opencv/modules/imgproc/src/imgwarp.cpp, line 1723
```

进而引发

```
OpenCV Error: Assertion failed ((img.depth() == CV_8U || img.depth() == CV_32F) && img.type() == templ.type()) in void cv::matchTemplate(
```

分析：检查错误从第一个开始，后续的不看，所以 matchTemplate 这个错是结果，不是根源。由此知是 ssize.area().....resize....引发的错。诱因是传入的图片或数据是空的，定位到代码：

```
BoundingBox boundingBox = mTldModule.processFrame(newFrameData, colEnd-colStart,
rowEnd-rowStart);
```

找到 newFrameData 是空的原因即可。

16 出现下述错误 Out of Memory Error

```
trol E/==MaApplication==: 'OutputStream' is null,serious issue and sendSocketCommand:updateapp_add=com.topotek.tracker
ATAL EXCEPTION: pool-2-thread-5
rocess: com.topotek.tracker, PID: 8106
ava.lang.OutOfMemoryError: Failed to allocate a 25165788 byte allocation with 4169824 free bytes and 9MB until OOM
   at java.util.ArrayList.add(ArrayList.java:124)
   at tld.BoundingBox.points(BoundingBox.java:75)
   at tld.Tld.track(Tld.java:274)
   at tld.Tld.processFrame(Tld.java:178)
   at com.topotek.trackingmodule.TldModule.processFrame(TldModule.java:71)
   at com.topotek.tracker.MainActivity.onPreviewFrame(MainActivity.java:106)
   at com.topotek.topotekmodule.ProjectModule$3.run(ProjectModule.java:117)
   at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1113)
   at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:588)
   at java.lang.Thread.run(Thread.java:818)
```

代码定位到 BoundingBox.java：75

```
final int stepx = (int) Math.ceil((width - 2 * POINTS_MARGIN_H) / POINTS_MAX_COUNT);
final int stepy = (int) Math.ceil((height - 2 * POINTS_MARGIN_V) / POINTS_MAX_COUNT);
```

分析：Math.ceil(double x)是函数的原型，也就是(width-2*POINT_MARGIN_H)/POINTS_MAX_COUNT 必须是 double 型结果，所以只需要将 POINT_MAX_COUNT 改为 double POINT_MAX_COUNT 即可。

17 出现 Attempt to read from field 'int org.opencv.core.Rect.x' on a null object reference

```
Process: com.topotek.tracker, PID: 3030
java.lang.NullPointerException: Attempt to read from field 'int org.opencv.core.Rect.x' on a null object reference
    at com.topotek.tracker.MainActivity.onPreviewFrame(MainActivity.java:108)
    at com.topotek.topotekmodule.ProjectModule$3.run(ProjectModule.java:117)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1113)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:588)
```

分析：定位到代码行

```
BoundingBox boundingBox = mTldModule.processFrame(frameData, frameWidth, frameHeight);
pointX = boundingBox.x + boundingBox.width / 2;
pointY = boundingBox.y + boundingBox.height /2;
```

其一，错误提示是....on a null object reference，意思指 boundingBox 是 null，故 boundingBox.x，boundingBox.width 不存在，只需设置判断条件，跳过 boundingBox 为空的情况即可；其二，BoundingBox 类是 Rect 型，故 boundingBox.x，boundingBox.width 是 int 型。而 pointX 是 int 型，所以要转计算结果的类型，如果不转换，安卓会自动为你转换。

18 出现!!! Failed Binder Transaction !!!

分析：程序多线程操作发生冲突。如 getOneFrame(true)和另一个已经被传值，正在做处理的冲突。

19 出现 H1.type() == CV_32F error

```
error: (-215) H1.type() == H2.type() && H1.type() == CV_32F in function double cv::compareHist
```

分析：先定位到代码行

```
Utils.bitmapToMat(bitmap1,mat1);
Utils.bitmapToMat(bitmap2,mat2);
mat1.convertTo(mat1, CvType.CV_32F);
mat2.convertTo(mat2, CvType.CV_32F);
double similarity = Imgproc.compareHist(mat1,mat2,Imgproc.CV_COMP_CORREL); //报错
```

所以错误源头在 mat1 和 mat2 上。报错提示 H1.type()==CV_32F，说明 mat1,mat2 并非 CV_32F 类型，百度知多通道的 mat 是不能转换位 CV_32F，只能转换为 CV_32FC3，上述 bitmap1 和 bitmap2 恰好是三通道图，故转换失败。解决方案是，转换成灰度图(单通道)，然后再转 CV_32F。BTW，上面得到的是两幅灰度图的对比，建议灰度图后，再转换成直方图，最后比较两个直方图更好。

20 出现 prevPyr[level].size() = nextPyr[level].size()问题

```
Error: Assertion failed (prevPyr[level * lvlStep1].size() == nextPyr[level * lvlStep2].size()) in virtual void cv::{anonymous}::SparsePyrLKOpticalFlowImpl::calc
```

答：代码如有 frame(box).copyTo(roi)，可能是 roi 全局变量，此前定义已有大小，现在由于 box 宽高变化，roi 大小变化导致错误，建议重新定义新的 mat 接收 frame 的裁剪。但后续函数处理，不接受图片大小变化，即一次成型，后续不改，运行此处还是会报错，可能需要查看官方文档。

21 图片出现条纹，一幅图出现多个带条纹的图片

答：图片格式问题，如 YUV420sp2BGR，可能需要改成 YUV420p2BGR，注意图片格式。

22 python-opencv 和 c++ opencv 读取同样一幅 JPG 图片，像素不同

答：读取同样的 JPG 格式图片，python 和 c++都调用 opencv 的 imread 函数，但读取到的像素值不同，如下列代码，原因在于 JPG 是有损压损格式，不同编译器读取图片解码方式不同，造成损失。解决方案是将图片保存为 bmp 格式。

```
for (int i = 0; i < 1; i++) {
      for (int j = 0; j < 15; j++) {
            printf("0 通道，dst[%d,%d] = %d;
                    1 通道，dst[%d,%d] = %d;
                    2 通道，dst[%d,%d] = %d\n", i, j,
                            src.at<Vec3b>(i, j)[0], i, j,    //CV_8UC3 格式存储
                            src.at<Vec3b>(i, j)[1], i, j,
                            src.at<Vec3b>(i, j)[2]);

                            //src.at<Vec3f>(i, j)[0], i, j,  //CV_32F 或 CV_32FC3 存储
                            //src.at<Vec3f>(i, j)[1], i, j,  //%d 打印时换成%f
                            //src.at<Vec3f>(i, j)[2]);
      }
}
```

23 基于 22，python 输出用 double 双精度，c++输出的是单精度 float

答：正解，特别是前者在 linux 环境，后者在 VS 环境，注意数值对比精度统一。

24 移植代码没错误或没警告提示，但 VS 编译时提示 opencv/dnn/shape_utils.hpp 有红色波浪线

答：头文件没有按标准的头文件进行引用，同时附加了其他头文件导致引用的 shape_utils.hpp， 如 Range clamp(std::max(r.begin, 0), r.end? std::(r.end): r.end+1)有问题。解决办法是去掉附加的头文件。

26 resize(src, dst2,Size(src.cols/2, src.rows/2), 0, 0, INTER_CUBIC)问题

```
//src 为 640x360
Tracker -> update(dst1,box);  //dst1 为 320x240，用时 200 毫秒
Tracker -> update(dst2,box);  //dst2 为 320x240，用时 1000 毫秒
```

同样图片，为什么缩放和裁剪造成的耗时不同？

答：缩放不同于裁剪，缩放不会损失像素，而裁剪会损失像素，即便维数相同，但像素处理数目不同，造成耗时不同。同样的，还有 pyrDown 和 pyrUp 函数。

27 灰度图是否有三通道

答：灰度图是特殊的三通道图，即 RGB 图，只不过每个通道进行特殊混合，像素值改变。读取彩图和灰度图区别：

```
Mat mat = imread("valid.jpg");
cvtColor(mat, mat, CV_BGR2GRAY);
uchar* data = mat.data;

int* p = (int*)malloc(sizeof(uchar*) * 10);
```

```
for (int i = 0; i < 10; i++) {
    p[i] = data[i];
    printf("mat[%d][%d]: %d\n", i, 0, p[i]);  //这个读出来的只是 0 通道
}
cout << "---------------" << endl;
for (int i = 0; i < 10; i++) {
    printf("mat[%d][%d]: %d\n", i, 0, mat.at<Vec3b>(i, 0)[0]); //注意这里的通道是 0
} //通道写成 1,2，打印出的像素会不同
```

28 TLD 窗口如何权值化处理，从而得到正确位置

```
float bbOverlap(const Rect &box1, const Rect &box2) {
    if (box1.x > box2.x + box2.width) { return 0.0; }
    if (box1.y > box2.y + box2.height) { return 0.0; }
    if (box1.x + box1.width < box2.x) { return 0.0; }
    if (box1.y + box1.height < box2.y) { return 0.0; }

    float colInt = min(box1.x + box1.width, box2.x + box2.width) - max(box1.x, box2.x);
    float rowInt = min(box1.y + box1.height, box2.y + box2.height) - max(box1.y, box2.y);

    float intersection = colInt * rowInt;
    float area1 = box1.width*box1.height;
    float area2 = box2.width*box2.height;
    return intersection / (area1 + area2 - intersection);
}

int main() {

    Mat img = imread("4.jpg");
    cout << "img size: " << img.cols << ", " << img.rows << endl;

    Rect box1(img.cols / 2 - 40, img.rows / 2 - 40, 80, 80);
    rectangle(img, box1, Scalar(0, 255, 0), 3);

    Rect box2(img.cols / 2 - 10, img.rows / 2 - 10, 80, 80);
    rectangle(img, box2, Scalar(255, 0, 0), 3);
    imshow("before", img);

    cout << "box1 size: " << box1.x << ", " << box1.y << ", " << box1.width << ", " <<
box1.height << endl;
    cout << "box2 size: " << box2.x << ", " << box2.y << ", " << box2.width << ", " <<
box2.height << endl;

    double result = bbOverlap(box1, box2);
```

```
    if (result > 0.7)
        cout << "Yes. Overlap > 0.7" << endl;
    else
        cout << "No." << endl;

    int cx = 0, cy = 0, cw = 0, ch = 0;

    int close_detections = 1;
    cx += box2.x;
    cy += box2.y;
    cw += box2.width;
    ch += box2.height;

    printf("cx, cy, cw, ch: %d, %d, %d, %d\n", cx, cy, cw, ch);

    Rect bbnext(0, 0, 0, 0);

    if (close_detections > 0) {

        //对与跟踪器预测到的box距离很近的box和跟踪器本身预测到的box进行坐标与大小平均
        //权值数1越大，则box1的值越大，相对的，bbnext越接近box1的位置
        bbnext.x = cvRound((float)(1 * box1.x + cx) / (float)(1 + close_detections));   //
weighted average trackers trajectory with the close detections
        bbnext.y = cvRound((float)(1 * box1.y + cy) / (float)(1 + close_detections));
        bbnext.width = cvRound((float)(1 * box1.width + cw) / (float)(1 +
close_detections));
        bbnext.height = cvRound((float)(1 * box1.height + ch) / (float)(1 +
close_detections));
        printf("processFrame: Tracker bb: %d %d %d %d\n", box1.x, box1.y, box1.width,
box1.height);
        printf("processFrame: Average bb: %d %d %d %d\n", bbnext.x, bbnext.y, bbnext.width,
bbnext.height);
    }

    rectangle(img, box1, Scalar(0, 255, 0), 3);
    rectangle(img, box2, Scalar(255, 0, 0), 3);
    rectangle(img, bbnext, Scalar(0, 0, 255), 3);
    imshow("after", img);
    waitKey(0);

    int aa = 0;
    return 0;
}
```

29 用 OpenCV 自带的 CascadeClassifier 做人脸检测

```
CascadeClassifier faceCascade;
int main()  {

    faceCascade.load("haarcascade_frontalface_default.xml");
    //Mat img = imread("img_face4test.JPG");      // 载入图片

    VideoCapture capture;
    capture.open("face.MP4");
    if (!capture.isOpened()) {
        printf("Failed in open video");
        return 1;
    }



    Mat frame,gray;
    vector<Rect> faces;
    int count = 0;

    while (capture.read(frame)) {
        count += 1;
        cout << "-------" << count << endl;

        cvtColor(frame, gray, COLOR_BGR2GRAY);

        faceCascade.detectMultiScale(gray, faces, 1.1, 3, 0);

        if (faces.size()>0) {
            for (int i = 0; i<faces.size(); i++) {
                rectangle(frame, Point(faces[i].x, faces[i].y), Point(faces[i].x +
faces[i].width, faces[i].y + faces[i].height), Scalar(0, 255, 0), 1, 8);    // 框出人脸
            }
        }

        imshow("Faces", frame);

        if(waitKey(25) == 27)
            break;

        faces.clear();

    }
    return 0;
}
```

分析：效果一般，图片尺寸越大，计算越慢，detectMultiScale(gray, faces, 1.1)中的1.1必须是

大于1的数字，如1.01,1.05,1.2,1.6都行，否则报错。

30 findContours 详解

原型：findContours(InputOutputArray image, OutputArrayOfArrays contours, OutputArray hierarchy, int mode, int method, Point offset=Point());

Image:单通道图像，可以是灰度图，一般最好是二值图，效果更好；

Contours: 轮廓，vector<vector<point>>型，contours.size()计算轮廓数量，而 contours[i].size()计算第 i 个轮廓上点的数量；

Hierarchy: vector<Vec4i> 型，即向量模板类定义的只有 4 个整型元素的向量；hierarchy[i][0] ~hierarchy[i][3]，分别表示第 i 个轮廓的后一个轮廓、前一个轮廓、父轮廓、内嵌轮廓的索引编号。如果当前轮廓没有对应的后一个轮廓、前一个轮廓、父轮廓或内嵌轮廓的话，则 hierarchy[i][0] ~hierarchy[i][3]的相应位被设置为默认值-1；

Mode：取值为整型，定义轮廓的检索模式，如只显示外部轮廓，内外都显示等；

Method：定义计算轮廓的近似方法；

Offset：指偏移量，所有的轮廓信息相对于原始图像对应点的偏移量，相当于在每一个检测出的轮廓点上加上该偏移量，并且 Point 还可以是负值！