# Debug和release切换时要重新配置连接器输入

# opencv有java版jar，此外java可以调用C++的dll

# CvRect和CvSize都是结构体名，不能用类的方法初始化

# IplImage结构来源于Intel Image Processing Library

# 截取图片

* 1. 对于Mat

Mat image= imread("lena.jpg");

Rect rect(100, 200, 200, 200);

Mat image\_roi = image(rect);

或

Mat image\_roi2;

image(rect).copyTo(image\_roi2);

* 1. 对于IplImage

CvSize size= cvSize(185, 220);//区域小

IplImage \*cropedImg=cvCreateImage(size, srcImg->depth, srcImg->nChannels);//创建目标图像

cvSetImageROI(srcImg, cvRect(55, 95, size.width, size.height));//设置源图像ROI

cvCopy(srcImg, cropedImg); //复制图像

cvResetImageROI(srcImg);//源图像用完后清空ROI

# 构造Mat

* 1. 拷贝构造函数则 只拷贝信息头和矩阵指针 ，而不拷贝矩阵

Mat A, C; // 只创建信息头部分

A = imread(“a.jpg”, CV\_LOAD\_IMAGE\_COLOR); // 这里为矩阵开辟内存

Mat B(A); // 使用拷贝构造函数

C = A; // 赋值运算符

通过任何一个对象所做的改变也会影响其它对象。

* 1. 你可以创建只引用部分数据的信息头。比如想要创建一个感兴趣区域（ ROI ），你只需要创建包含边界信息的信息头：

Mat D (A, Rect(10, 10, 100, 100) ); // using a rectangle

Mat E = A(Range:all(), Range(1,3)); // using row and column boundaries

* 1. 但某些时候你仍会想拷贝矩阵本身(不只是信息头和矩阵指针)，这时可以使用函数 clone() 或者 copyTo()

Mat F = A.clone();

Mat G;

A.copyTo(G);

1. Mat自动分配和释放内存空间，IplImage需手动为其分配和释放内存空间

# cvShowImage()不会立即显示出来，waitkey()时会显示

cvWaitKey不止是Wait Key 而已,其实还涉及到消息响应, 有这个函数cv内部的WndProc函数才能起作用, 才会更新窗口。

# imshow是OpenCV2.0版本之后从本来的C接口的cvshowimage转换到C++的替代物。

# error: 无法解析的外部符号 "class cv...

原因：缺少库文件，加上

#pragma comment( lib, "opencv\_highgui231d.lib")

#pragma comment( lib, "opencv\_core231d.lib")

或在链接器，附加依赖性中添加。

也有可能是opencv配置错误。

# 使用CMake时出错：By not providing "FindOpenCV.cmake" in CMAKE\_MODULE\_PATH

解决：在OpenCV\_DIR 手动添加"D:/opencv244/build"，这个目录下有OpenCVConfig.cmake

# 配置完opencv，运行跳出弹窗“计算机丢失MSVCP120D.dll”

虽然把缺失的文件放到了system32中，但这个问题仍然存在。放到opencv bin目录下问题解决。

# 数据的存储

Mat\_<uchar> 对应的是CV\_8U，

Mat\_<char> 对应的是CV\_8S，

Mat\_<int> 对应的是CV\_32S，

Mat\_<float> 对应的是CV\_32F，

Mat\_<double>对应的是CV\_64F，

# 去除一个看似没有使用的opencv版本后，一定要对所有项目重新编译

# 连续不停读取摄像头并显示，程序一动就崩溃，就算加了Sleep()也没用。改用SetTimer()

# int cvRound (double value)

对一个double型的数进行四舍五入，并返回一个整型数

# void cvGEMM(const CvArr\* src1, const CvArr\* src2, double alpha, const CvArr\* src3, double beta, CvArr\* dst, int tABC=0)

矩阵乘法。Mat类型可以直接相乘

# cvCvtColor(image,gray,CV\_BGR2GRAY)

颜色空间转换函数

# cvDestroyAllWindows()

销毁所有HighGUI窗口

# 一些基本函数

* 1. 数据类型

CV\_8UC1： 8位无符号单通道矩阵

CV\_32FC2: 32位浮点数双通道矩阵

（数据数有三种：U，F，S有符号整数）

* 1. 图象类型

IPL\_DEPTH\_8U:8位无符号整数图象

IPL\_DEPTH\_32F:32位浮点数图象

* 1. 矩阵定义与释放矩阵

CvMat\*M=cvCreateMat(4,4,CV\_32FC1)；//4行4列的数据类型为CV\_32FC1的矩阵；

cvReleaseMat(&M);

函数CvMat cvMat(int rows, int cols, int type, void\* data=NULL) 无需释放

* 1. 复制矩阵

CvMat\*M1=cvCreateMat(4,4,CV\_32FC1)；

CvMat\*M2；

M2=cvCloneMat(M1);

* 1. 初始化矩阵与单位矩阵

double a[]={1,2,3,4,5,6,7,8,9,10,11,12};

CvMat Ma=cvMat(3,4,CV\_64FC1,a);

* 1. 赋值和获取值

void cvmSet(CvMat\* mat, int row, int col, double value)

double cvmGet(const CvMat\* mat, int row, int col)

# 点线面

画点:cvCircle()

画线:cvLine()

写字:cvPutText()

# 转置

Mat M; M.t();

# 矩阵

Mat eye = Mat::eye(4,4,CV\_8U); // identity matrix

Mat ones = Mat::ones(4,4,CV\_8U);

Mat zeros = Mat::zeros(4,4,CV\_8U);