Opencv 笔记1 在Windows平台下的开发环境配置与测试

1. java开发方式
   1. 环境准备

已经安装好JAVA + Eclipse的开发环境

下载opencv2.4.11 for windows

* 1. 设置环境

安装opencv-2.4.11.exe

实际上这就是一个解压过程，将里面的文件解压到指定的目录即可，对于Java开发，我们关注的是Java目录下面的jar包，此文件用于Java的开发，但是实际的实现是依赖对应目录下面的opencv\_java2411.dll文件。

设置Eclipse开发环境

创建一个Java项目，然后在build path中设置jar包依赖（这一步比较简单）

同时因为其具体实现并不是java而是opencv\_java2411.dll，所以我们在jar中要设置native library location，指向对应平台下面的opencv\_java2411.dll

1.3代码测试

**import** org.opencv.core.CvType;

**import** org.opencv.core.Mat;

**public** **class** TestMain {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

System.*loadLibrary*("opencv\_java2411");

Mat m = Mat.*eye*(3, 3, CvType.***CV\_8UC3***);

System.***out***.println(m.dump());

}

}

1. VS2013开发环境设置&&读取图像并且显示

因为opencv本身就是C++开发的，所以最好就是利用C++进行开发。这里我们使用的是VS2013。(opencv已经解压)

下面是配置的过程:

因为现在很多的计算机都是64位系统，但是我们打开vs的时候，默认依旧是win32开发平台，所以在配置环境的时候，可以将win32&&x64一起配置好，然后在编译运行的时候，可以选择平台。

1. 添加opencv的环境变量 （根据自己解压的目录和版本进行适当的修改）

将C:\opencv2\_4\_11\build\x86\vc12\bin; C:\opencv2\_4\_11\build\x64\vc12\bin添加到环境变量PATH中

新建opencv, value为：C:\opencv2\_4\_11\build\

注意：这个环境变量需要重新启动才可以生效，我也觉得很诡异

1. 新建VS项目，添加include && lib &&连接器/输入/附加依赖项

在新建项目中切换到属性管理器，在debug/win32中的Microsoft.Cpp.Win32.user双击，在VC++目录中添加

”包含目录” ：

C:\opencv2\_4\_11\build\include;

C:\opencv2\_4\_11\build\include\opencv\;

C:\opencv2\_4\_11\build\include\opencv2

“包含库”：

C:\opencv2\_4\_11\build\x86\vc12\lib

对应x64平台添加C:\opencv2\_4\_11\build\x64\vc12\lib

连接器\输入\附加依赖项:添加在lib目录下面对的所有的库名字（见备注一）

1. 测试代码

#include <opencv2/opencv.hpp>

#include <stdio.h>

using namespace cv;

using namespace std;

int main(){

const char\* imagename = "C:\\20150329\_101805.jpg";

Mat m = imread(imagename, 1);

if (true == m.empty()){

fprintf(stderr, "can not read image %s\n", imagename);

return -1;

}

imshow("yang", m);

waitKey(0);

return 0;

}

#include <opencv2/opencv.hpp>

#include <stdio.h>

using namespace cv;

using namespace std;

int main(){

IplImage \* test = cvLoadImage("C:\\20150329\_101805.jpg");

cvNamedWindow("test\_demo", 1);

cvShowImage("test\_demo", test);

cvvWaitKey(0);

cvDestroyWindow("test\_demo");

cvReleaseImage(&test);

return 0;

}

杨腾飞

Guangzhou

2015.08.07

附录一：附加依赖项的名字

opencv\_calib3d2411d.lib

opencv\_contrib2411d.lib

opencv\_core2411d.lib

opencv\_features2d2411d.lib

opencv\_flann2411d.lib

opencv\_gpu2411d.lib

opencv\_highgui2411d.lib

opencv\_imgproc2411d.lib

opencv\_legacy2411d.lib

opencv\_ml2411d.lib

opencv\_nonfree2411d.lib

opencv\_objdetect2411d.lib

opencv\_ocl2411d.lib

opencv\_photo2411d.lib

opencv\_stitching2411d.lib

opencv\_superres2411d.lib

opencv\_ts2411d.lib

opencv\_video2411d.lib

opencv\_videostab2411d.lib