一 GUI指令

1.窗口管理

cvNamedWindow("win1",CV\_WINDOW\_AUTOSIZE) //创建一个新窗口

cvMoveWindow("win1",100,100) //定位一个新窗口，offset from the UL corner of the screen

IplImage\* img=0;

img=cvLoadImage(fileName,CV\_LOAD\_IMAGE\_COLOR); //载入图像

cvShowImage("win1",img); //显示图像

该函数可以在上面建立的窗口（win1）中显示彩色或灰度的字节型/浮点型图像。字节型图像像素值范围为[0-255]；浮点型图像像素值范围为[0-1]。彩色图像的三色元素按BGR（蓝-绿-红）顺序存储

cvDestroyWindow("win1"); //关闭窗口

cvResizeWindow("win1",100,100); //改变窗口大小 new width/heigh in pixels

2.输入处理

（1）鼠标处理程序

void mouseHandler(int event,int x,int y,int flags,void\*param)

{

switch(event)

{

case CV\_EVENT\_LBUTTONDOWN:

if(flags&CV\_EVENT\_FLAG\_CTRLKEY)

break;

case CV\_EVENT\_LBUTTONUP:

break;

}

}

注册该事件处理程序：

mouseParam=5;

cvSetMouseCallback(“windowName”,mouseHandler,&mouseParam); //第三个参数可以设置为NULL.

(2)键盘事件：

对键盘输入没有专门的事件处理程序。

Int key; key = cvWaitKey(10); //wait 10ms for input

中止程序等待键盘输入： int key;key = cvWaitKey(0); //wait indefinitely for input

键盘输入循环处理程序：

While(1)

{

Key = cvWaitKey(10);

If(key ==’q’)

Break;

Switch(key)

{

Case ‘h’:

....

Break;

Case ‘i’:

...

Break;

}

}

1. 处理滑动条事件：

定义一个滑动条处理程序：

Void trackbarHandler(int pos)

{

Printf(“trackbar position:%d\n”,pos);

}

注册该事件处理程序

Int trackbarVal = 25;

Int maxVal = 100;

cvCreateTrackbar(“barName”,”windowName”,&trackbarVal,maxVal,trackbarHandler);

获取当前滑动条位置：

Int pos = cvGetTrackbarPos(“barName”,”windowName”);

设置滑动条位置：

cvSetTrackbarPos(“barName”,”windowName”,nPos);

二 OpenCV的基本数据结构

1. 图像数据结构

IPL图像：

IplImage:  
 int nChannels; //颜色通道数目（1,2,3,4）

Int depth; //像素的位深 IPL\_DEPTH\_8U,IPL\_DEPTH\_8S,IPL\_DEPTH\_32F

Int width; //图像宽度 (像素为单位)

Int height; //图像高度

Char\* imageData; //图像数据指针 ;注意彩色图像按RGB顺序存储数据

Int dataOrder; //0--将像素点不同通道的值交错排在一起，形成单一像素平面；

//1--把所有像素同通道值排在一起，形成若干个通道平面，再把平面排列起来。

//cvCreateImage 智能创建像素交错排列式的图像

Int origin; //0--像素远点为左上角；1--像素远点为左下角。

Int widthStep; //相邻行的同列点之间的字节数 width\*nChannels\*depth/8

Int imageSize;//图像的大小（字节为单位）=height\*widthStep

Struct \_IplROI \*roi; //图像的感兴趣区域（ROI）。ROI为非空时对图像的，处理仅限于ROI区域。

Char \*imageDataOrigin; //图像数据未对齐时的数据原点指针//需要正确重新分配图像内存

Int align; //图像数据的行对齐 4 or 8 byte alignment //OpenCV中用widthStep代替

Char colorModel[4]; //颜色模型 OpenCV中忽略此项。

1. 矩阵与相量
2. 矩阵：

CvMat: //2D矩阵

Int type; //元素类型（uchar,short,int,float,double）与标志

Int step; //整行长度字节

Int rows,cols; //行，列数

Int height,width; //矩阵高度，宽度，与rows、cols对应

Union data;

uchar\* ptr; //data pointer for an unsigned char matrix

short\* s; // data pointer for a short matrix

int\* i; // data pointer for an integer matrix

float\* fl; // data pointer for a float matrix

double\* db; // data pointer for a double matrix

CvMatND //N维矩阵

Int type; //元素类型

Int dims; //矩阵维数

Union data;

uchar\* ptr; //data pointer for an unsigned char matrix

short\* s; // data pointer for a short matrix

int\* i; // data pointer for an integer matrix

float\* fl; // data pointer for a float matrix

double\* db; // data pointer for a double matrix

Struct dim[]; //各维信息

Size; //元素数目

Step; //元素间距（字节为单位）

CvSparseMat //N-维稀疏矩阵