1、cvFastArctan

计算反正切函数，返回范围0~360度。

2、输出文字

CString str;

CvFont font;

cvInitFont(&font,CV\_FONT\_HERSHEY\_COMPLEX\_SMALL,0.5,0.5,0,1,8);

cvPutText(img,str,cvPoint(0.5\*tp.x+0.5\*bp.x,0.5\*tp.y+0.5\*bp.y),&font,color2);

3、获取轮廓

CvMemStorage \*storage=cvCreateMemStorage(0);

CvSeq\* first\_contour=NULL;

int Nc=cvFindContours(img\_edg,storage,&first\_contour,sizeof(CvContour),CV\_RETR\_LIST);//轮廓个数

4、标注轮廓

for (CvSeq\*c=first\_contour;c!=NULL;c=c->h\_next)

{

double area=cvContourArea(c);

if (area/100<minarea||area/100>maxarea1)continue;

//得到矩形边界框

CvRect brect=cvBoundingRect(c);

//得到Ì最小轮廓长方形

CvBox2D box=cvMinAreaRect2(c);

//以下省略

……

CvPoint2D32f pt[4];

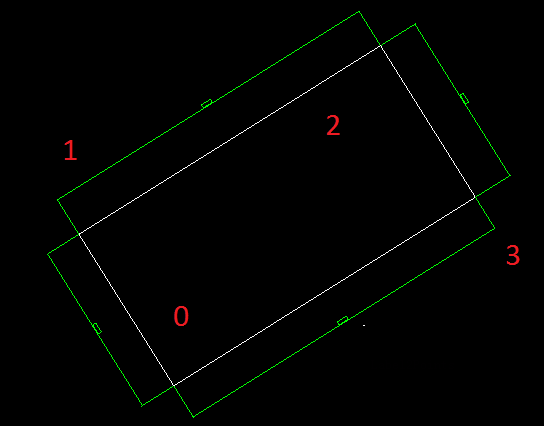
cvBoxPoints(box,pt);

//以下省略

……

}

用cvMinAreaRect2()获得的矩形，可用cvBoxPoints(box,pt)提取四个顶点，得到的顺序如下：



5、标注函数：

标注两点的顺序按逆时针顺序

void Label2Points(IplImage \*img,CvPoint \*p1,CvPoint \*p2,double rate,CvScalar color1,CvScalar color2,int ut)

{

double l=10;

double a=0;

double d= sqrt(double((p1->x-p2->x)\*(p1->x-p2->x)+(p1->y-p2->y)\*(p1->y-p2->y)));

CString str;

CvFont font;

cvInitFont(&font,CV\_FONT\_HERSHEY\_COMPLEX\_SMALL,0.5,0.5,0,1,8);

if(ut==1)//单位是mm

{

d=d/rate;

str.Format("%2.3lfmm",d);

}

else

{

str.Format("%2.3lfpx",d);

}

if (abs(p1->y-p2->y)<1)

a=0.0;

else if (abs(p1->x-p2->x)<1)

{

a=3.1415/2;

}

else

a=fabs(atan((double(p1->y-p2->y)/double(p1->x-p2->x))));

CvPoint tp,bp,ep1,ep2;

float angle=cvFastArctan(p2->x-p1->x,p2->y-p1->y);

if (angle>=0 && angle<90)

{

tp.x=p1->x+static\_cast<int>(l\*sin(a) + 0.5f);

tp.y=p1->y-static\_cast<int>(l\*cos(a) + 0.5f);

bp.x=p2->x+static\_cast<int>(l\*sin(a) + 0.5f);

bp.y=p2->y-static\_cast<int>(l\*cos(a) + 0.5f);

ep1.x=p1->x+static\_cast<int>(2\*l\*sin(a) + 0.5f);

ep1.y=p1->y-static\_cast<int>(2\*l\*cos(a) + 0.5f);

ep2.x=p2->x+static\_cast<int>(2\*l\*sin(a) + 0.5f);

ep2.y=p2->y-static\_cast<int>(2\*l\*cos(a) + 0.5f);

cvPutText(img,str,cvPoint(0.5\*tp.x+0.5\*bp.x,0.5\*tp.y+0.5\*bp.y),&font,color2);

}

if (angle>=90&&angle<180)

{

tp.x=p1->x-static\_cast<int>(l\*sin(a) + 0.5f);

tp.y=p1->y-static\_cast<int>(l\*cos(a) + 0.5f);

bp.x=p2->x-static\_cast<int>(l\*sin(a) + 0.5f);

bp.y=p2->y-static\_cast<int>(l\*cos(a) + 0.5f);

ep1.x=p1->x-static\_cast<int>(2\*l\*sin(a) + 0.5f);

ep1.y=p1->y-static\_cast<int>(2\*l\*cos(a) + 0.5f);

ep2.x=p2->x-static\_cast<int>(2\*l\*sin(a) + 0.5f);

ep2.y=p2->y-static\_cast<int>(2\*l\*cos(a) + 0.5f);

cvPutText(img,str,cvPoint(tp.x,tp.y),&font,color2);

}

if (angle>=180&&angle<270)

{

tp.x=p1->x-static\_cast<int>(l\*sin(a) + 0.5f);

tp.y=p1->y+static\_cast<int>(l\*cos(a) + 0.5f);

bp.x=p2->x-static\_cast<int>(l\*sin(a) + 0.5f);

bp.y=p2->y+static\_cast<int>(l\*cos(a) + 0.5f);

ep1.x=p1->x-static\_cast<int>(2\*l\*sin(a) + 0.5f);

ep1.y=p1->y+static\_cast<int>(2\*l\*cos(a) + 0.5f);

ep2.x=p2->x-static\_cast<int>(2\*l\*sin(a) + 0.5f);

ep2.y=p2->y+static\_cast<int>(2\*l\*cos(a) + 0.5f);

cvPutText(img,str,cvPoint(0.5\*tp.x+0.5\*bp.x,0.5\*tp.y+0.5\*bp.y),&font,color2);

}

if (angle>=270&&angle<360)

{

tp.x=p1->x+static\_cast<int>(l\*sin(a) + 0.5f);

tp.y=p1->y+static\_cast<int>(l\*cos(a) + 0.5f);

bp.x=p2->x+static\_cast<int>(l\*sin(a) + 0.5f);

bp.y=p2->y+static\_cast<int>(l\*cos(a) + 0.5f);

ep1.x=p1->x+static\_cast<int>(2\*l\*sin(a) + 0.5f);

ep1.y=p1->y+static\_cast<int>(2\*l\*cos(a) + 0.5f);

ep2.x=p2->x+static\_cast<int>(2\*l\*sin(a) + 0.5f);

ep2.y=p2->y+static\_cast<int>(2\*l\*cos(a) + 0.5f);

cvPutText(img,str,cvPoint(0.5\*tp.x+0.5\*bp.x,0.5\*tp.y+0.5\*bp.y),&font,color2);

}

cvLine(img,\*p1,ep1,color1);

cvLine(img,\*p2,ep2,color1);

cvLine(img,tp,bp,color1);

}

6、把图像显示到图片框中

CvvImage m\_cvvimg;

m\_cvvimg.CopyOf(frame,1);

m\_cvvimg.DrawToHDC(hDC,rect);

7、程序结束必须释放摄像头！

void CAutoDetectorDlg::OnClose()

{

cvReleaseCapture(&capture);

CDialogEx::OnClose();

}

8、改变鼠标的形状，在OnSetCursor消息处理函数中，判断鼠标位置，改变鼠标形状，用到的函数**SetCursor((HCURSOR)LoadCursor(NULL,IDC\_CROSS))。**

BOOL CAutoDetectorDlg::OnSetCursor(CWnd\* pWnd, UINT nHitTest, UINT message)

{

CPoint cp;

CRect pic\_pos;

GetDlgItem(IDC\_STATIC)->GetWindowRect(&pic\_pos);

GetCursorPos(&cp);

ScreenToClient(&cp);

ScreenToClient(&pic\_pos);

if (cp.x>pic\_pos.left && cp.x<pic\_pos.left+pic\_pos.Width() &&

cp.y>pic\_pos.top &&cp.y<pic\_pos.top+pic\_pos.Height())

{

**SetCursor((HCURSOR)LoadCursor(NULL,IDC\_CROSS));**

m\_inpic=true;

return true;

}

m\_inpic=false;

m\_flag=false;

return CDialogEx::OnSetCursor(pWnd, nHitTest, message);

}

9、保存图片

CFileDialog dlg=CFileDialog(FALSE);

dlg.m\_ofn.lpstrTitle="保存";

dlg.m\_ofn.lpstrDefExt="bmp";

dlg.m\_ofn.lpstrFilter="位图(\*.bmp)\0\*.bmp";

if(IDOK== dlg.DoModal())

{

CString savefilename=dlg.GetPathName();

cvSaveImage(savefilename,framecopy);

};

10、保存文本

CFileDialog dlg=CFileDialog(FALSE);

dlg.m\_ofn.lpstrTitle="保存文本";

dlg.m\_ofn.lpstrDefExt="txt";

dlg.m\_ofn.lpstrFilter="文本文件(\*.txt)\0\*.txt";

if(IDOK== dlg.DoModal())

{

CString savefilename=dlg.GetPathName();

CFile file(savefilename,CFile::modeCreate|CFile::modeWrite);

CString buf;

GetDlgItemText(IDC\_OUTPUT,buf);

file.Write(buf,buf.GetLength());

file.Close();

};

11、右键菜单

定义全局对象

CMenu menu;

CMenu \*pPopup;

初始化：

menu.LoadMenu(IDR\_MENU1);

pPopup=menu.GetSubMenu(0);

pPopup->EnableMenuItem(ID\_CAN,MF\_DISABLED);

pPopup->EnableMenuItem(ID\_MEA,MF\_DISABLED);

右键单击响应函数：

void CAutoDetectorDlg::OnRButtonDown(UINT nFlags, CPoint point)

{

if (detecting&&m\_inpic)

{

ClientToScreen(&point);

pPopup->TrackPopupMenu(TPM\_LEFTALIGN | TPM\_RIGHTBUTTON,point.x,point.y,this);

}

CDialogEx::OnRButtonDown(nFlags, point);

}

12、控件变量与控件值的数据交换

m\_unit=1;

UpdateData(FALSE);//根据m\_unit的值更新对应的控件值

而UpdateData(TRUE)是把控件的值赋值给控件变量

13、参考以下代码，体会CPtrArray的使用

//正y长¡è方¤?形?

**CLabelPoint \*lp=new CLabelPoint;**

lp->p00.x=brect.x+ROIrect.x;

lp->p00.y=brect.y+brect.height+ROIrect.y;

lp->p01.x=brect.x+ROIrect.x;

lp->p01.y=brect.y+ROIrect.y;

lp->p02.x=brect.x+brect.width+ROIrect.x;

lp->p02.y=brect.y+ROIrect.y;

lp->p03.x=brect.x+brect.width+ROIrect.x;

lp->p03.y=brect.y+brect.height+ROIrect.y;

//斜¡À长¡è方¤?形?

lp->p10.x=pt[0].x+ROIrect.x;

lp->p10.y=pt[0].y+ROIrect.y;

lp->p11.x=pt[1].x+ROIrect.x;

lp->p11.y=pt[1].y+ROIrect.y;

lp->p12.x=pt[2].x+ROIrect.x;

lp->p12.y=pt[2].y+ROIrect.y;

lp->p13.x=pt[3].x+ROIrect.x;

lp->p13.y=pt[3].y+ROIrect.y;

**PointArray.Add(lp);**

**以上代码属于一个循环的一部分，注意这里每次循环都创建一个CLabelPoint\*指针对象，对指针对象赋值，然后加入到PointArray集合中。而不是整个循环只在循环开始之前，只创建一个CLabelPoint\* 指针对象，然后对指针对象赋值，然后加入到PointArray集合中去。这个错误高试了很长时间才被发现，以后编程过程中要格外注意！！！**

**对CPtrArray的释放，采取以下方法：**

void DeletePtrArray(CPtrArray &PA)

{

int count=PA.GetSize();

for (int iPos =0;iPos<count ;iPos++)

{

delete PA.GetAt(iPos);

}

PA.RemoveAll();

}

无论CPtrArray集合中储存何种指针，旨可以用此函数释放。

14、关于内存泄漏

局部指针变量在用完以后要及时释放，不然会造成内存泄漏，直到崩溃！！！在本程序调试过程中，发现，在很短时间内，从任务管理器观察到其内存使用从3M到800M以上，而且还在不断增加！

例如在以下代码中：

void CAutoDetectorDlg::OnMouseMove(UINT nFlags, CPoint point)

{

// TODO: 在此添加消息处理程序代码和/或调用默认值

if (!detecting)

{

return;

}

DistplaytoHdc(framecopy);

**temimg=cvCloneImage(framecopy);**

if (m\_leftbuttondown)

{

m\_moving=true;

//如果m\_leftbuttondown已为真，且m\_flag=true 说明正在绘图矩形选择区域

if (m\_flag)

{

if (abs(m\_pold.x-point.x)>10||abs(m\_pold.y-point.y)>10)//屏蔽掉一些微小的鼠标移动

{

if (!prepareROI&&!m\_measurewhloe)

{

m\_drawRectangle=true;

m\_drawline=false;

//绘制中间过程

CvPoint tempoint=cvPoint(point.x,point.y);

DashRectangle(temimg,m\_pold,tempoint,m\_colorroi);

DistplaytoHdc(temimg);

// cvReleaseImage(&temimg);

}

}

}

}

else if (m\_flag)

{

//如果m\_leftbuttondown已为假，即鼠标已经释放，且m\_flag=true 说明正在绘图直线

//如果鼠标在左键被按下并被拖拽的范围超过一定阀值认为是在绘制矩形

if (abs(m\_pold.x-point.x)>10||abs(m\_pold.y-point.y)>10)

{

m\_drawRectangle=false;

m\_drawline=true;

CvPoint tempoint=cvPoint(point.x,point.y);

cvLine(temimg,m\_pold,tempoint,m\_colorline);

DistplaytoHdc(temimg);

//以下实现动态描绘：

//两种情况：1、绘制直线。2、绘制虚线矩形

//if之前把之前的图像作一个拷贝1

//作一个拷贝2=copy(拷贝1)

//在拷贝2上绘直线，显示拷贝2

}

}

**cvReleaseImage(&temimg);**//这里一定要释放，不然内存严重泄漏！

CDialogEx::OnMouseMove(nFlags, point);

}

代码中的temimg对象是一个临时对象，它在上述代码中被**cvCloneImage(framecopy);初始化，在函数结束时释放cvReleaseImage(&temimg);如果不释放，在每次执行上述代码时，内存不断泄漏！**

在不确定图像指针是否有值的情况下,可以这样尝试释放：

if (temimg)

cvReleaseImage(&temimg);

在对cvCloneimage和cvCopy图像1到图像2中的时候，务必要对图像2执行上述尝试释放操作。

15、启动界面的实现方式：

新建一个对话框添加一个图片控件，生成类，重绘函数中：

void CBeginDlg::OnPaint()

{

CPaintDC dc(this); // device context for painting

CDC \*pDC=GetDC();

CBitmap bitmap;

bitmap.LoadBitmap(IDB\_BEGIN);

CDC dcCompatible;

dcCompatible.CreateCompatibleDC(pDC);

dcCompatible.SelectObject(&bitmap);

CRect rect;

GetDlgItem(IDC\_PIC)->GetWindowRect(&rect);

pDC->BitBlt(0,0,rect.Width(),rect.Height(),&dcCompatible,0,0,SRCCOPY);

}

初始化函数中SetTimer：

BOOL CBeginDlg::OnInitDialog()

{

CDialogEx::OnInitDialog();

SetTimer(1,2000,NULL);

return TRUE; // return TRUE unless you set the focus to a control

}

OnTimer函数：

void CBeginDlg::OnTimer(UINT\_PTR nIDEvent)

{

if (nIDEvent==1)

{

OnOK();

}

CDialogEx::OnTimer(nIDEvent);

}

重写DestroyWindow()

BOOL CBeginDlg::DestroyWindow()

{

KillTimer(1);

return CDialogEx::DestroyWindow();

}

在主窗口的Create函数或初始化函数后

CBeginDlg dlg;

dlg.DoModal();

16、MoveWindow函数是坐标是相对于父窗口的。

17、cvReleaseImage用来释放cvCreateImage所创建的图像，如果是cvLoadImage得到的图像，不能用cvReleaseImage，而是直接img=NULL这样的形式释放内存。