

DoubleLi
517712484

博客园 :: 首页 :: 博文 :: 闪存 :: 新随笔 :: 联系 :: 订阅  :: 管理 ::  2860 随笔 :: 2 文章 :: 265 评论 :: 0 引用

< 2017年8月 >						
日	一	二	三	四	五	六
30	31	<u>1</u>	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

公告

昵称: DoubleLi
园龄: 7年6个月
粉丝: 1053
关注: 29
[+加关注](#)

搜索

- 常用链接
- [我的随笔](#)
 - [我的评论](#)
 - [我的参与](#)
 - [最新评论](#)
 - [我的标签](#)

- 随笔分类(3976)
- ASP.NET(30)
 - ASP.NET MVC(11)
 - Boost(85)
 - c#(8)
 - C++/C(668)
 - c++11(4)
 - cmake(15)
 - com/ATL/Activex(75)
 - Css(16)
 - CxImage(12)
 - DirectX(16)
 - Extjs(13)
 - ffmpeg、ffplay(53)
 - FREESWITCH
 - gcc/g++/gdb(29)
 - h264/h265/mpeg(17)
 - hisi(11)
 - hls(2)
 - html5(3)
 - http(10)
 - ICE(4)
 - java(13)
 - javascript(66)
 - jpeglib(3)
 - Jquery(9)
 - json(6)
 - JUCE(2)
 - libcurl(26)
 - LINPHONE
 - Linux(418)
 - linux驱动(27)
 - live555(18)
 - makefile(26)
 - Matlab(3)
 - mfc control(7)
 - mingw(6)
 - mysql(2)
 - nginx(77)
 - Nhibernate / Hibernate(3)
 - onvif(17)
 - OpenCV(23)
 - OpenGL(1)
 - Oracle(29)
 - P2P(2)
 - PJSIP
 - ProtoBuf(5)
 - Qt(10)
 - RPC(3)
 - RTSP/RTP/RTMP/HLS(53)
 - SDL(8)
 - shell(8)
 - SilverLight(1)
 - sql server(3)
 - sqlite(5)
 - SVN/git(42)
 - tcp/ip(14)
 - uboot/kernel/rootfs(42)
 - UML(2)
 - VC/MFC(588)

opencv2 使用鼠标绘制矩形并截取和保存矩形区域图像

前言

好长时间没写博文了，今天偷偷懒写篇关于opencv2中鼠标响应操作的文章。

鼠标操作属于用户接口设计，以前一直使用Qt来做，但是如果只需要简单的鼠标，键盘操作，直接调用opencv库的函数也未尝不可，鼠标操作之前已经接触很多了，在MFC，QT，OpenGL，等等中，理论主要就是两点，一是监控鼠标操作，鼠标点击，移动，松开，然后通过mouse_event识别判断出那一种鼠标的操作，根据不同的操作然后进行处理，二是在主函数中加入鼠标的回调函数，将鼠标操作与程序的窗口绑定。

第一节 函数介绍

暂时只接触了两个关于opencv2鼠标响应操作的函数，下面分别介绍一下：

1.1 回调函数

opencv2.4.5中，提供的鼠标回调函数是 setMouseCallback，函数声明如下：

```
01. | CV_EXPORTS void setMouseCallback(const string& winname, MouseCallback onMouse, void* userdata = 0);
```

函数参数介绍

const string& winname，windows视窗名称，对名为winname的视窗进行鼠标监控。

MouseCallback onMouse，鼠标响应处理函数，监听鼠标的点击，移动，松开，判断鼠标的操作类型，并进行响应的函数处理。

void* userdata = 0 鼠标响应处理函数的ID，与鼠标相应处理函数相匹配就行，暂时只用到默认为0的情况。

函数使用实例：

```
01. | namedWindow("img");
02. | setMouseCallback("img",on_mouse,0);
```

1.2 鼠标响应处理函数

opencv2.4.5中，鼠标相应处理函数一般默认形参和返回参数，函数形式如下：

```
01. | void on_mouse(int event,int x,int y,int flags,void *ustc)
```

函数参数介绍：

int event，鼠标操作时间的整数代号，在opencv2.4.5中，event鼠标事件总共有10中，从0-9依次代表如下：

Event:

01.	#define CV_EVENT_MOUSEMOVE 0	滑动
02.	#define CV_EVENT_LBUTTONDOWN 1	左键点击
03.	#define CV_EVENT_RBUTTONDOWN 2	右键点击
04.	#define CV_EVENT_MBUTTONDOWN 3	中间点击
05.	#define CV_EVENT_LBUTTONUP 4	左键释放
06.	#define CV_EVENT_RBUTTONUP 5	右键释放
07.	#define CV_EVENT_MBUTTONUP 6	中间释放
08.	#define CV_EVENT_LBUTTONDBLCLK 7	左键双击
09.	#define CV_EVENT_RBUTTONDBLCLK 8	右键双击
10.	#define CV_EVENT_MBUTTONDBLCLK 9	中间释放

int x,int y，代表鼠标位于窗口的（x，y）坐标位置，窗口左上角默认为原点，向右为x轴，向下为y轴，

int flags，代表鼠标的拖拽事件，以及键盘鼠标联合事件，总共有32种事件，依次如下：

flags:

01.	#define CV_EVENT_FLAG_LBUTTON 1	左键拖拽
02.	#define CV_EVENT_FLAG_RBUTTON 2	右键拖拽
03.	#define CV_EVENT_FLAG_MBUTTON 4	中间拖拽
04.	#define CV_EVENT_FLAG_CTRLKEY 8	(8~15)按Ctrl不放事件
05.	#define CV_EVENT_FLAG_SHIFTKEY 16	(16~31)按Shift不放事件
06.	#define CV_EVENT_FLAG_ALTKEY 32	(32~39)按Alt不放事件（后面8-39还有待研究）

void *ustc，函数参数的编号（暂时用不到）

第二节 鼠标操作实例

2.1 示例程序代码

程序如下，已经附上说明：

```
01. | #include <opencv2/core/core.hpp>
02. | #include <opencv2/highgui/highgui.hpp>
03. | #include <stdio.h>
```

web(17)
webrtc(11)
webserver/CGI(2)
wifi(51)
Windows 编程(30)
Windows控件开发/自绘(17)
wireshark(2)
WPF(1)
WTL(1)
编解码(12)
操作系统(9)
单片机(6)
多线程编程(33)
分布式系统(3)
服务器架构设计(6)
工程拓扑、视频设备(1)
工具、插件 开发与使用(156)
管理&投资(3)
计算机视觉(10)
计算机网络(4)
开源库(63)
其它(9)
嵌入式(207)
数字电路(4)
图像/ISP(2)
图形界面编程(114)
图形学(19)
网络编程(123)
无线网络(38)
异常诊断与调试(108)
音频(13)
音视频、流媒体(251)

随笔档案(2865)

2017年8月 (5)
2017年7月 (78)
2017年6月 (52)
2017年5月 (110)
2017年4月 (9)
2017年3月 (61)
2017年2月 (48)
2017年1月 (9)
2016年12月 (38)
2016年11月 (18)
2016年9月 (21)
2016年8月 (10)
2016年7月 (39)
2016年6月 (39)
2016年5月 (26)
2016年4月 (96)
2016年3月 (58)
2016年2月 (4)
2016年1月 (8)
2015年12月 (24)
2015年11月 (23)
2015年10月 (22)
2015年9月 (7)
2015年8月 (32)
2015年7月 (45)
2015年6月 (89)
2015年5月 (74)
2015年4月 (14)
2015年3月 (41)
2015年2月 (3)
2015年1月 (45)
2014年12月 (78)
2014年11月 (20)
2014年10月 (19)
2014年9月 (85)
2014年8月 (52)
2014年7月 (11)
2014年6月 (94)
2014年5月 (105)
2014年4月 (112)
2014年3月 (39)
2014年2月 (40)
2014年1月 (41)
2013年12月 (114)
2013年11月 (141)
2013年10月 (18)
2013年9月 (55)
2013年8月 (69)
2013年7月 (9)
2013年6月 (11)
2013年5月 (55)
2013年4月 (34)
2013年3月 (42)
2013年2月 (12)
2013年1月 (44)
2012年12月 (90)
2012年11月 (60)
2012年10月 (25)
2012年9月 (15)
2012年8月 (46)
2012年7月 (63)
2012年6月 (22)
2012年5月 (12)
2012年4月 (15)

```
04.
05. using namespace cv;
06.
07. cv::Mat org,dst,img,tmp;
08. void on_mouse(int event,int x,int y,int flags,void *ustc)//event鼠标事件代号, x,y鼠标坐标, flags拖拽和键盘操作的代号
09. {
10.     static Point pre_pt = (-1,-1);//初始坐标
11.     static Point cur_pt = (-1,-1);//实时坐标
12.     char temp[16];
13.     if (event == CV_EVENT_LBUTTONDOWN)//左键按下, 读取初始坐标, 并在图像上该点处划圆
14.     {
15.         org.copyTo(img);//将原始图片复制到img中
16.         sprintf(temp,"%d,%d",x,y);
17.         pre_pt = Point(x,y);
18.         putText(img,temp,pre_pt,FONT_HERSHEY_SIMPLEX,0.5,Scalar(0,0,0,255),1,8);//在窗口上显示坐标
19.         circle(img,pre_pt,2,Scalar(255,0,0,0),CV_FILLED,CV_AA,0);//划圆
20.         imshow("img",img);
21.     }
22.     else if (event == CV_EVENT_MOUSEMOVE && !(flags & CV_EVENT_FLAG_LBUTTON))//左键没有按下的情况下鼠标移动的处理函数
23.     {
24.         img.copyTo(tmp);//将img复制到临时图像tmp上, 用于显示实时坐标
25.         sprintf(temp,"%d,%d",x,y);
26.         cur_pt = Point(x,y);
27.         putText(tmp,temp,cur_pt,FONT_HERSHEY_SIMPLEX,0.5,Scalar(0,0,0,255));//只是实时显示鼠标移动的坐标
28.         imshow("img",tmp);
29.     }
30.     else if (event == CV_EVENT_MOUSEMOVE && (flags & CV_EVENT_FLAG_LBUTTON))//左键按下时, 鼠标移动, 则在图像上划矩形
31.     {
32.         img.copyTo(tmp);
33.         sprintf(temp,"%d,%d",x,y);
34.         cur_pt = Point(x,y);
35.         putText(tmp,temp,cur_pt,FONT_HERSHEY_SIMPLEX,0.5,Scalar(0,0,0,255));
36.         rectangle(tmp,pre_pt,cur_pt,Scalar(0,255,0,0),1,8,0);//在临时图像上实时显示鼠标拖动时形成的矩形
37.         imshow("img",tmp);
38.     }
39.     else if (event == CV_EVENT_LBUTTONUP)//左键松开, 将在图像上划矩形
40.     {
41.         org.copyTo(img);
42.         sprintf(temp,"%d,%d",x,y);
43.         cur_pt = Point(x,y);
44.         putText(img,temp,cur_pt,FONT_HERSHEY_SIMPLEX,0.5,Scalar(0,0,0,255));
45.         circle(img,pre_pt,2,Scalar(255,0,0,0),CV_FILLED,CV_AA,0);
46.         rectangle(img,pre_pt,cur_pt,Scalar(0,255,0,0),1,8,0);//根据初始点和结束点, 将矩形画到img上
47.         imshow("img",img);
48.         img.copyTo(tmp);
49.         //截取矩形包围的图像, 并保存到dst中
50.         int width = abs(pre_pt.x - cur_pt.x);
51.         int height = abs(pre_pt.y - cur_pt.y);
52.         if (width == 0 || height == 0)
53.         {
54.             printf("width == 0 || height == 0");
55.             return;
56.         }
57.         dst = org(Rect(min(cur_pt.x,pre_pt.x),min(cur_pt.y,pre_pt.y),width,height));
58.         namedWindow("dst");
59.         imshow("dst",dst);
60.         waitKey(0);
61.     }
62. }
63. void main()
64. {
65.     org = imread("1.jpg");
66.     org.copyTo(img);
67.     org.copyTo(tmp);
68.     namedWindow("img");//定义一个img窗口
69.     setMouseCallback("img",on_mouse,0);//调用回调函数
70.     imshow("img",img);
71.     cv::waitKey(0);
72. }
```

2.2 程序运行结果分析

程序源图形：

2012年3月 (20)
2012年1月 (10)
2011年12月 (12)
2011年11月 (6)
2011年10月 (14)
2011年9月 (5)
2011年8月 (1)
2011年5月 (1)

文章分类(2)
SilverLight(1)
sql server(1)

参考博客
chenyujing1234
com
Dean Chen的专栏
boost 等
linux驱动
morewindows
Nginx模块开发与原理剖析
Nginx模块开发与原理剖析
opencv教程
Sloan
大坡3D软件开发
个人开发历程知识库
关注DirectX
关注DirectX
回忆未来-向东
http://justwinit.cn/index.php
音视频FFmpeg等
音视频FFmpeg等

最新评论
1. Re:Windows上WinRAR.exe命令
行参数说明(转载) .
@echo offC:CD "C:\Program
Files (x86)\WinRAR"net stop
MSSQLserverwinrar.exe a -ag"-
[YYYY-MM-DD-HH-MM.....
--神奇小饼干
2. Re:Onvif开发之服务端发现篇
请问下这样会存在程序占用内存越
来越大, 怎么能解决?
--kerwinwolk
3. Re:opencv2 使用鼠标绘制矩形
并截取和保存矩形区域图像
QT中怎么实现? 我配置了
opencv2.4.9跟QTcreator, 然后
用这种方法总出错.
--骑着毛驴去旅行
4. Re:在Linux中使用VS Code编
译调试C++项目
博主你的文章很赞啊,但是现在
VSCode更新了,tasks.json的格式
变了不一样了,能不能更新一下
--蓝域小兵
5. Re:H264(NAL简介与I帧判断)
我想转载,可是我不会,刚刚复制
了一篇,注明是转载的
--Oliver William

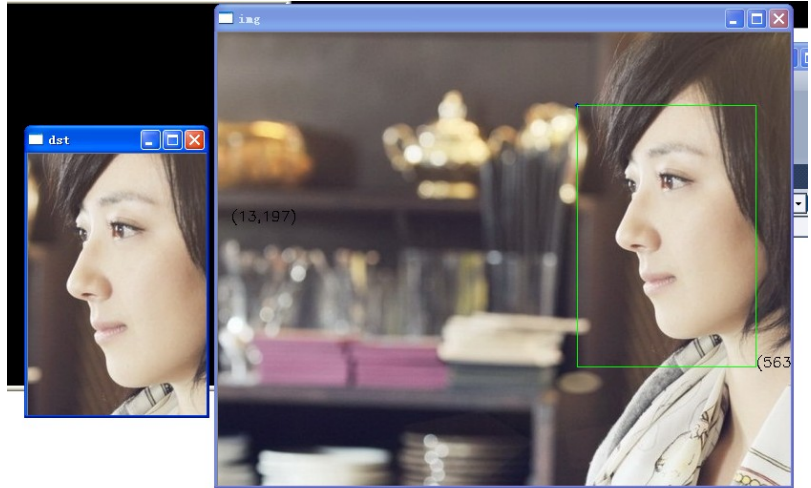
阅读排行榜
1. Nginx之location 匹配规则详解
(131584)
2. SVN安装配置与使用(80869)
3. C++的Json解析库: jsoncpp
和boost .(63556)
4. HTTP 错误 404 - 文件或目录未
找到的最终解决方法(43653)
5. C++中的头文件和源文件
(40332)

评论排行榜
1. 非IE内核浏览器支持activex插
件(22)
2. Javascript中定义类(15)
3. Nginx之location 匹配规则详解
(14)
4. C++中的头文件和源文件(8)
5. SVN安装配置与使用(5)

推荐排行榜
1. C++中的头文件和源文件(18)
2. Javascript中定义类(12)
3. Nginx之location 匹配规则详解
(12)
4. JavaScript中typeof知多少?
(8)
5. SVN安装配置与使用(6)



程序运行结果：



第三节，参考资料

- 既然参考别人的博文，当然要把源资料介绍给大家，让大家参考一下：
1. 在OpenCV中利用鼠标绘制矩形和截取图像的矩形区域
<http://blog.csdn.net/quarryman/article/details/6435527>
 2. OpenCV响应鼠标函数cvSetMouseCallback () 和其副程式onMouse () 的使用 (OpenCV2.4.5)
<http://blog.csdn.net/glb562000520/article/details/8938582>

分类: OpenCV

好文要顶 关注我 收藏该文

DoubleLi
关注 - 29
粉丝 - 1053

+加关注

« 上一篇： 略谈如何从工作线程中弹出对话框
» 下一篇： 计算机视觉领域的一些牛人博客，超有实力的研究机构等的网站链接

posted on 2013-11-22 15:02 DoubleLi 阅读(13589) 评论(5) 编辑 收藏

评论

#1楼 2014-08-06 16:01 夕下随风
这种材料好难找呀，费了我这么大的精力，多谢作者的分享，对我这种初入门的小子来说真是太有用了。
支持(0) 反对(0)

#2楼 2014-09-02 11:41 alvin小宇

请问如果我想多次截取这个图片上的部分，并分别把它们保存起来，该如何实现？

支持(1) 反对(0)

#3楼 2016-08-16 17:59 kekefudao

@ alvin小宇
请问你这个问题解决了么？

支持(0) 反对(0)

#4楼 2017-06-20 08:54 1234gmh

裁剪完感兴趣区域后，在此基础上做一些图像处理，怎么建界面啊？有知道的吗

支持(0) 反对(0)

#5楼 2017-07-24 14:18 骑着毛驴去旅行

QT中怎么实现？我配置了opencv2.4.9跟QTcreator，然后用这种方法总出错。

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

【推荐】50万行VC++源码：大型组态工控、电力仿真CAD与GIS源码库

成为薪资最高的前端工程师

前端开发工程师认证项目

| 零基础入门+进阶

Google

GitHub

联合打造

仅限300席

- 最新IT新闻：
- 探访微软总部：这3栋建筑代表着它的过去、现在和未来
 - 《我的世界》面向Win10和Android开放“Better Together”更新
 - 支付宝微信苹果开启新一轮补贴大战：最高4888元
 - 外媒：马斯克坦承躁郁症事实 称有不同人生体验
 - Facebook收购AI初创企业Ozlo：或融入Messenger推出数字助理
- » 更多新闻...

 JIGUANG | 极光

app 开发 用 极光

► 推送 IM 短信 统计 分享 ◀

- 最新知识库文章：
- 为什么你该开始学习编程了？
 - 小printf的故事：什么是真正的程序员？
 - 程序员的工作、学习与绩效
 - 软件开发为什么很难
 - 唱吧DevOps的落地，微服务CI/CD的范本技术解读
- » 更多知识库文章...