**OpenCV鼠标事件响应**

Posted on 2013年05月30日 by U3d / [Unity3D脚本/插件](http://www.unitymanual.com/category/script)/被围观 128 次

程序定义了一个drawbox结构，结构中包含矩形的两个对角点，两个图像指针（程序可以指定在哪个图像上画矩形）和一个控制是否画图的bool类型变量。在回调函数中，程序监听了三种类型的鼠标事件：左键按下、鼠标移动和左键抬起。这三种类型的鼠标事件定义了一个完整的鼠标画图过程。左键按下时开始画图，拖动显示所画的图形，左键抬起表示画图结束。在结束之前，为了实时显示所画的图形，将所画的图放在box结构中的temp变量中，待画图结束时在box结构的image中存放这一次画图的最终结果。

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |
| --- | --- |
| 01 | #include<cv.h> |
| 02 | #include<highgui.h> |
| 03 | #include<iostream> |
| 04 | **using** **namespace** std; |
| 05 | *// callback function* |
| 06 | **void** cvMouseCallback(**int** mouseEvent,**int** x,**int** y,**int** flags,**void**\* param); |
| 07 | **struct** drawbox |
| 08 | { |
| 09 | CvPoint point1; |
| 10 | CvPoint point2; |
| 11 | IplImage\* image; |
| 12 | IplImage\* temp; |
| 13 | **bool** isDraw; |
| 14 | }; Unity3D教程手册 |
| 15 | **int** main(**int** argc,**char**\* argv[]) |
| 16 | { |
| 17 | *// declare and initialize a struct drawbox variable* |
| 18 | drawbox box; |
| 19 | box.point1 = cvPoint(0,0); |
| 20 | box.point2 = cvPoint(0,0); |
| 21 | box.image = cvCreateImage(cvSize(640,480),IPL\_DEPTH\_8U,3); |
| 22 | cvZero(box.image); |
| 23 | box.temp = cvCloneImage(box.image); |
| 24 | box.isDraw = **false**; |
| 25 | *// register a mouse callback function* |
| 26 | cvNamedWindow("exam",CV\_WINDOW\_AUTOSIZE); |
| 27 | cvSetMouseCallback("exam",cvMouseCallback,&box); |
| 28 | **while**(1) |
| 29 | { |
| 30 | cvCopy(box.image,box.temp); |
| 31 | **if**(box.isDraw) |
| 32 | { |
| 33 | cvRectangle(box.temp,box.point1,box.point2,CV\_RGB(255,255,255)); |
| 34 | } |
| 35 | cvShowImage("exam",box.temp); |
| 36 | **if**(cvWaitKey(20) == 27) **break**; |
| 37 | } |
| 38 | cvReleaseImage(&box.image); |
| 39 | cvReleaseImage(&box.temp); |
| 40 | cvDestroyWindow("exam"); |
| 41 | **return** 0; |
| 42 | } |
| 43 | **void** cvMouseCallback(**int** mouseEvent,**int** x,**int** y,**int** flags,**void**\* param) |
| 44 | { |
| 45 | drawbox\* box = (drawbox\*)param; |
| 46 | **switch**(mouseEvent) |
| 47 | { |
| 48 | **case** CV\_EVENT\_LBUTTONDOWN: |
| 49 | box->point1 = cvPoint(x,y); |
| 50 | box->point2 = cvPoint(x,y); |
| 51 | box->isDraw = **true**; |
| 52 | **break**; |
| 53 | **case** CV\_EVENT\_MOUSEMOVE: |
| 54 | box->point2 = cvPoint(x,y); |
| 55 | **break**; |
| 56 | **case** CV\_EVENT\_LBUTTONUP: |
| 57 | box->point2 = cvPoint(x,y); |
| 58 | cvRectangle(box->image,box->point1,box->point2,CV\_RGB(0,255,0)); |
| 59 | box->isDraw = **false**; |
| 60 | **break**; |
| 61 | } |
| 62 | **return**; |
| 63 | } |