OpenCV Tutorial

Part V
Image Control and Taskbar Manipulation

08 December 2005

Gavin S Page
gsp8334@cs.rit.edu

Tasks

The HighGUI aspect of the OpenCV library provides control over image file manipulation, visualization, and interaction. This will go over some of the basic functionality not specifically covered in a previous tutorial.

HighGUI

It’s obvious that any image processing library is going to need a method to visualize resultant images HighGUI is the simple viewing library in place for use in OpenCV. It utilizes a container to display *IplImages. It additionally contains the functions for saving and loading images along with interacting with an external camera and converting images.*

Image Viewing

The *cvNamedWindow* is the container apparatus used to display *IplImages*. It is created using the function *cvNamedWindow(const char* name, int flags)* where the flag is a binary parameter for automatic adjustment of window size. The *IplImage* is placed inside using *cvShowimage(const char* name, const CvArr* image).* When a named window is no longer needed it should be destroyed using *cvDestroyWindow(const char* name)* or *cvDestroyAllWindows(void).* Additionally, if the image is no longer needed it should be

Trackbars can also be attached to the named window to manipulate parameters via callbacks.

Using Trackbars

Create Callback Function

The important part of the callback is the function which is called on a change from the trackbar. This function contains the steps to be performed to alter the state of the necessary variable.

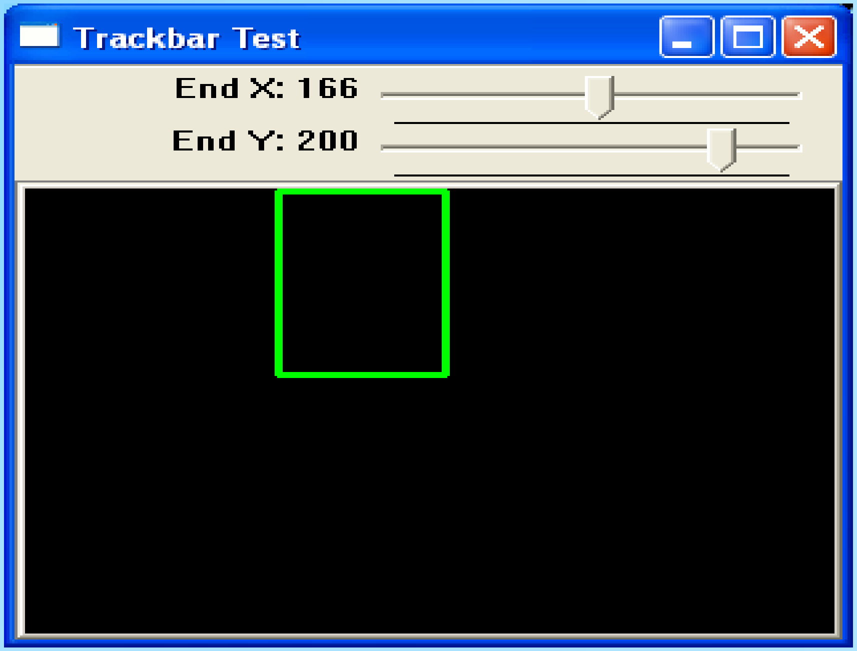
```
//Callback function for the trackbars
void CHANGE_DIMS(int a)
{
                    //release previous iteration’s image
                    cvReleaseImage(&hw);
                    //get a copy of the original image
                    hw = cvCloneImage(org);
                    //draw the rectangle
                    cvRectangle( hw, cvPoint(sx, sy), cvPoint(ex, ey), CV_RGB(0,
                                        255, 0), 2, 8, 0);
                    //send the image to the container
                    cvShowImage("Trackbar Test", hw);
}
```

A program requires calls registered the callback function and creating the trackbar(s).

```
CHANGE_DIMS(1);
//The next two lines pass a reference to the callback function.
//the reference is adjusted via the trackbar and the new value is returned
ex = cvCreateTrackbar("End X",
                    "Trackbar Test", &ex, 320, CHANGE_DIMS);
ey = cvCreateTrackbar("End Y",
                    "Trackbar Test", &ey, 240, CHANGE_DIMS);
```

Result

The program produces a window with 2 taskbars which manipulate the end x and y values of a rectangle drawn to a blank image



End X: 166

End Y: 200

Final

Trackbars provide a useful and simple means to adjust variables in a program. This is especially useful when testing different parameter values.