How to select a bounding box ( ROI ) in OpenCV (C++/Python) ?

https://secure.gravatar.com/avatar/708995cd13fb5756f6bc418aecb23b78?s=32&d=mm&r=g[Satya Mallick](https://www.learnopencv.com/author/spmallick/)

**MARCH 13, 2017**[**35 COMMENTS**](https://www.learnopencv.com/how-to-select-a-bounding-box-roi-in-opencv-cpp-python/#disqus_thread)

[](https://www.learnopencv.com/wp-content/uploads/2017/03/selectROI-OpenCV.gif)

In this tutorial, we will learn how to select a bounding box or a rectangular region of interest (ROI) in an image in OpenCV.  In the past, we had to write our own bounding box selector by handling mouse events. However, now we have the option of using a function **selectROI** that is natively part of OpenCV.

I am always amazed by the weird choices made in the OpenCV library. You would think that selectROI would be part of **highgui** that has functions for displaying images, drawing on images etc. However, selectROI is part of the tracking API! As you will notice later in the post, the choices made while writing selectROI are a bit odd. But, before we criticize we gotta be thankful that someone produced something useful even though it is not perfect.

Let’s dive in and see the usage of selectROI

How to select a region of interest in OpenCV

As selectROI is part of the tracking API, you need to have **OpenCV 3.0 ( or above )** installed with **opencv\_contrib**.

Let’s start with a sample code. It allows you to select a rectangle in an image, crop the rectangular region and finally display the cropped image.

We will modify the highlighted line to try different options.

**C++**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | #include <opencv2/opencv.hpp>  // selectROI is part of tracking API  #include <opencv2/tracking.hpp>    using namespace std;  using namespace cv;      int main (int argc, char \*\*arv)  {      // Read image      Mat im = imread("image.jpg");        // Select ROI      Rect2d r = selectROI(im);        // Crop image      Mat imCrop = im(r);        // Display Cropped Image      imshow("Image", imCrop);      waitKey(0);        return 0;  } |

The same code can be written in Python as

**Python**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | import cv2  import numpy as np    if \_\_name\_\_ == '\_\_main\_\_' :        # Read image      im = cv2.imread("image.jpg")        # Select ROI      r = cv2.selectROI(im)        # Crop image      imCrop = im[int(r[1]):int(r[1]+r[3]), int(r[0]):int(r[0]+r[2])]        # Display cropped image      cv2.imshow("Image", imCrop)      cv2.waitKey(0) |

Drag rectangle from top left to bottom right

If you are like me, you would prefer to drag a rectangle from the top left corner to the bottom right corner instead of the dragging it from the center. Well, we can easily fix that by replacing the highlighted line with the following line.

**C++**

|  |  |
| --- | --- |
| 1  2 | bool fromCenter = false;  Rect2d r = selectROI(im, fromCenter); |

**Python**

|  |  |
| --- | --- |
| 1  2 | fromCenter = False  r = cv2.selectROI(im, fromCenter) |

How to specify the window name?

Won’t it be nice, if you could use an existing window instead of ROI selector’s window. Well here you go

**C++**

|  |  |
| --- | --- |
| 1  2 | bool fromCenter = false;  Rect2d r = selectROI("Image", im, fromCenter); |

**Python**

|  |  |
| --- | --- |
| 1  2 | fromCenter = False  r = cv2.selectROI("Image", im, fromCenter) |

Don’t want to show crosshair?

Now, suppose you do not like the crosshair and would like to see the rectangle without it. You can modify the code to not show the crosshair.

**C++**

|  |  |
| --- | --- |
| 1  2  3 | bool showCrosshair = false;  bool fromCenter = false;  Rect2d r = selectROI("Image", im, fromCenter, showCrosshair); |

**Python**

|  |  |
| --- | --- |
| 1  2  3 | showCrosshair = False  fromCenter = False  r = cv2.selectROI("Image", im, fromCenter, showCrosshair) |

How to select multiple regions of interest (roi) of an image?

The function selectROI also allows you to select multiple regions of interest, but there appear to be two bugs.

**Bug Alert 1**: As per the instructions, you can drag a rectangle, and then press ENTER and drag another rectangle. However, there appears to be a bug in the implementation in OpenCV 3.2. You have to hit ENTER twice after the **first** rectangle. For all subsequent rectangles, you should hit ENTER once.

**C++**

|  |  |
| --- | --- |
| 1  2  3  4  5 | // Specify a vector of rectangles (ROI)  vector<Rect2d> rects;  bool fromCenter = false;  // The selected rectangles are in  selectROI("Image", im, rects, fromCenter); |

**Bug Alert 2**: I could not get the python version to work and there is no documentation. The following code runs, but the variable **rects** is not populated. The function also does not return anything. If you find a fix, please let me know in the comments below.

**Python**

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | # Note this code does not work.  # Specify a vector of rectangles (ROI)  rects = []  fromCenter = false  # Select multiple rectangles  selectROI("Image", im, rects, fromCenter) |