

# LEARN OPENCV BY EXAMPLES

OpenCV simplified for beginners by the use of examples. Learn OpenCV with basic implementation of different algorithms.

|                      |                               |                                   |                          |  |
|----------------------|-------------------------------|-----------------------------------|--------------------------|--|
| <a href="#">Home</a> | <a href="#">For Beginners</a> | <a href="#">Table of Contents</a> | <a href="#">Keywords</a> |  |
|----------------------|-------------------------------|-----------------------------------|--------------------------|--|

## Bitwise AND, OR, XOR and NOT

void **bitwise\_and**(InputArray src1, InputArray src2, OutputArray dst, InputArray mask=noArray())

Calculates the per-element bit-wise conjunction of two arrays or an array and a scalar.

Parameters:

- **src1** – first input array or a scalar.
- **src2** – second input array or a scalar.
- **src** – single input array.
- **value** – scalar value.
- **dst** – output array that has the same size and type as the input arrays.
- **mask** – optional operation mask, 8-bit single channel array, that specifies elements of the output array to be changed.

The function calculates the per-element bit-wise logical conjunction for:

- Two arrays when **src1** and **src2** have the same size:

$$\text{dst}(I) = \text{src1}(I) \wedge \text{src2}(I) \quad \text{if } \text{mask}(I) \neq 0$$

### SEARCH CONTENTS OF THIS BLOG

### POPULAR POSTS

- 1 Find Contour
- 2 Basic drawing examples
- 3 Line Detection by Hough Line Transform
- 4 Face Detection using Haar-Cascade Classifier
- 5 Perspective Transform
- 6 Sobel Edge Detection

- An array and a scalar when `src2` is constructed from `Scalar` or has the same number of elements as `src1.channels()`:

$$\text{dst}(I) = \text{src1}(I) \wedge \text{src2} \quad \text{if } \text{mask}(I) \neq 0$$

- A scalar and an array when `src1` is constructed from `Scalar` or has the same number of elements as `src2.channels()`:

$$\text{dst}(I) = \text{src1} \wedge \text{src2}(I) \quad \text{if } \text{mask}(I) \neq 0$$

void **bitwise\_not**(InputArray src, OutputArray dst, InputArray mask=noArray())

Inverts every bit of an array.

void **bitwise\_or**(InputArray src1, InputArray src2, OutputArray dst, InputArray mask=noArray())

Calculates the per-element bit-wise disjunction of two arrays or an array and a scalar.

void **bitwise\_xor**(InputArray src1, InputArray src2, OutputArray dst, InputArray mask=noArray())

Calculates the per-element bit-wise “exclusive or” operation on two arrays or an array and a scalar.

### Example:

-----

```

1  #include "opencv2/highgui/highgui.hpp"
2  #include "opencv2/imgproc/imgproc.hpp"
3
4  using namespace cv;
5  using namespace std;
6
7  int main( )
8  {
9      Mat drawing1 = Mat::zeros( Size(400,200), CV_8UC1 );
10     Mat drawing2 = Mat::zeros( Size(400,200), CV_8UC1 );
11
12     drawing1(Range(0,drawing1.rows),Range(0,drawing1.cols/2))=255;
13     drawing2(Range(100,150),Range(150,350))=255; imshow("drawing2",
14
15     Mat res;
16     bitwise_and(drawing1,drawing2,res);      imshow("AND",res);
17     bitwise_or(drawing1,drawing2,res);      imshow("OR",res);
18     bitwise_xor(drawing1,drawing2,res);      imshow("XOR",res);
19     bitwise_not(drawing1,res);               imshow("NOT",res);
20
21
22     waitKey(0);

```

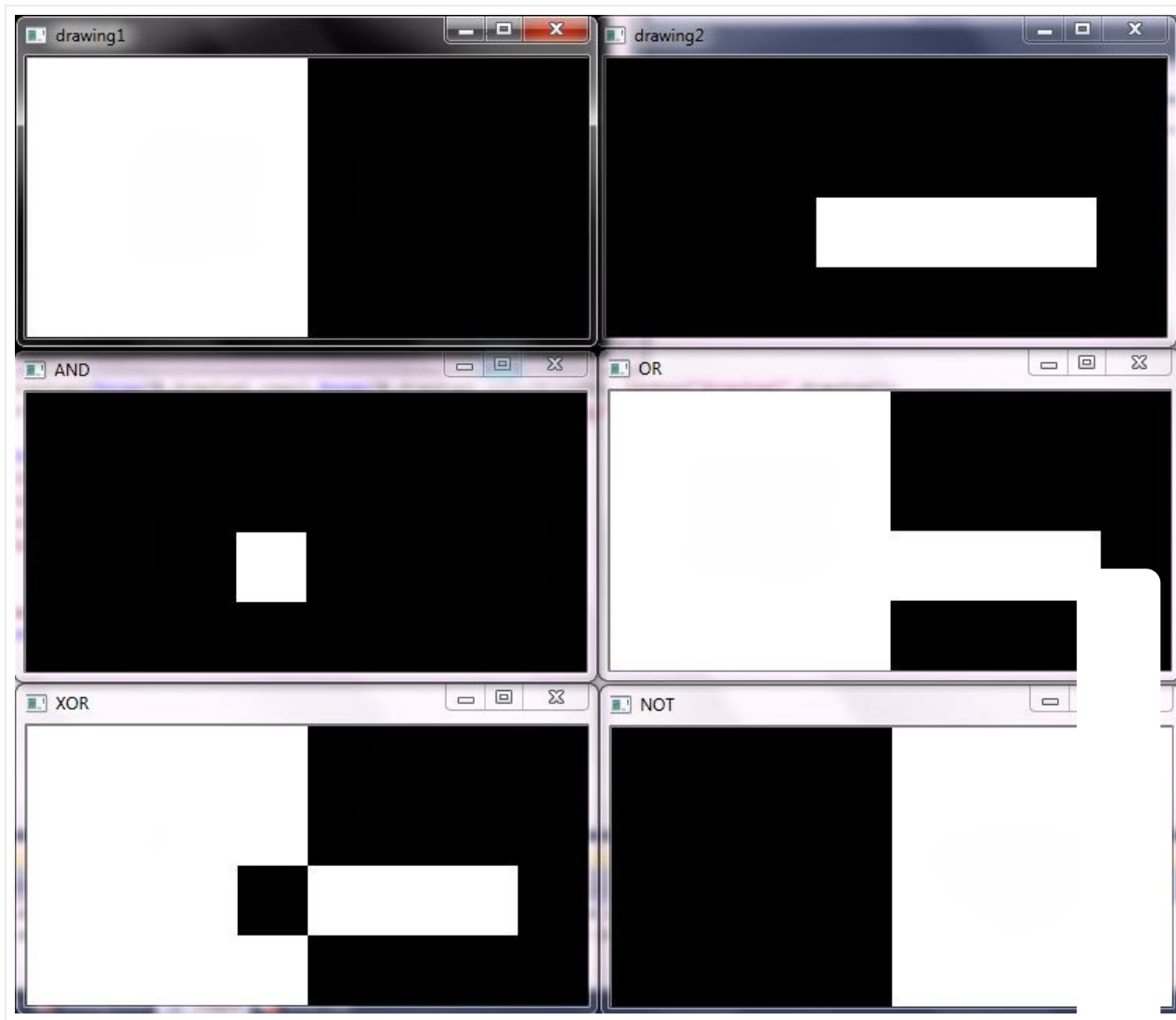
- 7 Kalman Filter Implementation (Tracking mouse position)
- 8 Histogram Calculation
- 9 OpenCV example to convert RGB to gray / other color spaces
- 10 Hough Circle Detection

## CATEGORIES

- [Accessory](#)
- [Applications](#)
- [Basics](#)
- [Edge Detection](#)
- [Feature Extraction](#)
- [Filter](#)
- [Miscellaneous](#)
- [Morphological Operation](#)

```
23 | return(0);  
24 | }
```

-----  
**Result:**



 Recommend this on Google

Labels: [Basics](#)