Processing cv::RotatedRect width and height

Asked 3 years, 7 months ago Active 3 years, 6 months ago Viewed 4k times

4

I need to define a rotated rectangle from its 4 corners. The rotated rectangle is defined by a center point, a size couple (width, height), and an angle.

How is it decided which size is the height, and which one is the width?

The width is not the length of the most horizontal edge, is it? E.g. if the angle is bigger than 90°, does it swap?

opencv

edited Jan 30 '17 at 5:47
Billal Begueradj
10.2k 16 65 93

asked Jan 27 '17 at 16:14
Grumot
51 1 6

2 Answers

Active | Oldest | Votes

3

`height` should be the largest side, `width` is the other one, and `angle` is the rotation angle (in degrees) in a clockwise direction.

Otherwise, you can get an equivalent rectangle with `height` and `width` swapped, rotated by 90 degrees.

You can use `minAreaRect` to find a `RotatedRect` :

```
vector<Point> pts = {pt1, pt2, pt3, pt4}

RotatedRect box = minAreaRect(pts);

// Be sure that largest side is the height
if (box.size.width > box.size.height)
{
    swap(box.size.width, box.size.height);
    box.angle += 90.f;
}
```

edited Jan 27 '17 at 16:32

answered Jan 27 '17 at 16:26
Miki
35.2k 10 84 169

Do you know what would be the Python equivalent of `box.size.width` or `box.size.height` ? I couldn't find a Python documentation for variable type of RotatedRect. – csg Mar 2 at 18:03

0

Ok, with Miki's help, and with some tests, I got it clearer...

It seems that the rotated rectangle is an upright rectangle (width and height are clearly defined, then)... that is rotated!

In image coords, y is directed to the bottom, the angle is given clockwise. In usual math coords (y to the top), the angle is given counter-clockwise. Then, it fits with c++ `<math.h>` included `atan2(y,x)` function for example (except that it returns radians).

Then, to summarize, if we consider one given edge of the rectangle (two corners), its length can be considered as the width if we retrieve the angle with `atan2` on its y difference and x difference. Something like:

```
Point pt1, pt2, pt3, pt4;
RotatedRect rect;

rect.center = (pt1 + pt2 + pt3 + pt4)/4;

// assuming the points are already sorted
rect.size.width = distance(pt1, pt2); // sqrt(...)
rect.size.height = distance(pt2, pt3);

rect.angle = atan2(pt2.y-pt1.y, pt2.x-pt1.x);
```

and this can be improved with `width` being the mean value of `dist(pt1,pt2)` and `dist(pt3,pt4)` for example. The same for `height` .

`angle` can also be calculated as being the mean value of `atan` for (pt1, pt2) and `atan` for (pt3, pt4).

answered Jan 27 '17 at 20:47

Grumot
51 1 6