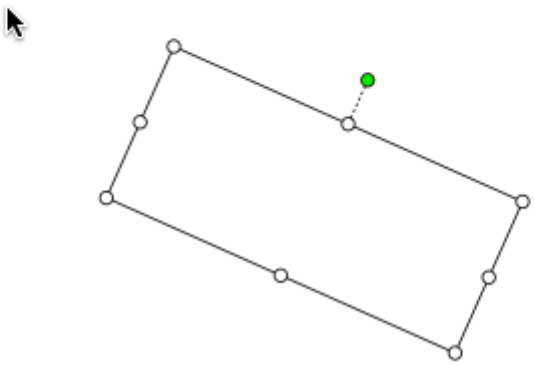
corner handle

Jump to bottom

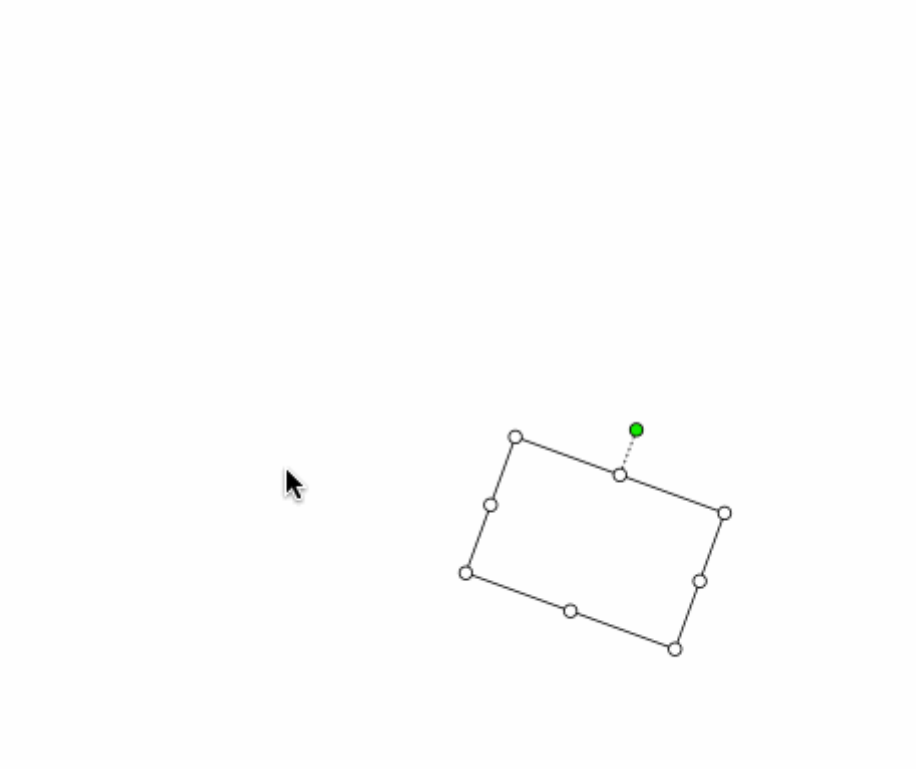
Wing Kai edited this page on 28 May 2018 · 2 revisions

拖动图形顶角手柄时的计算步骤

自由伸缩



锁定宽高比伸缩



前提

在 `svg` 中，绘制一个矩形元素 `rect` ，
需要传入矩形左上角的坐标(`x` 和 `y`)和 它的宽高 (`width` 和 `height`)四个参数，
所以绘制拉伸边线手柄后的图形，关键在于计算出新图形的 `xy` 以及 `width` 或 `height`

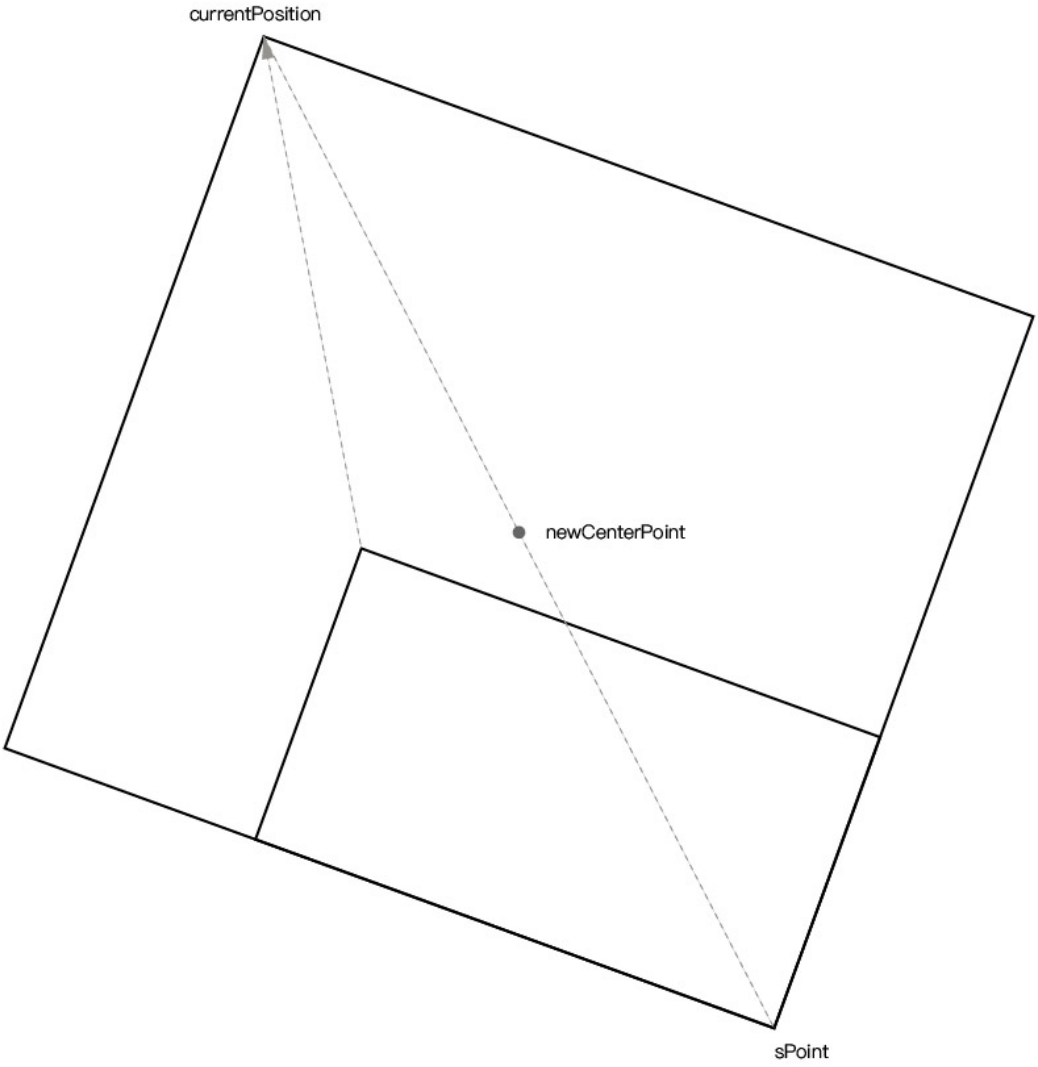
注意：

- 该例子的计算过程不适用于镜像伸缩的情况。
- 所有的坐标命名，都对应代码内变量的名字。

Step 1

以拖动图形的 **左上方** 手柄为例:

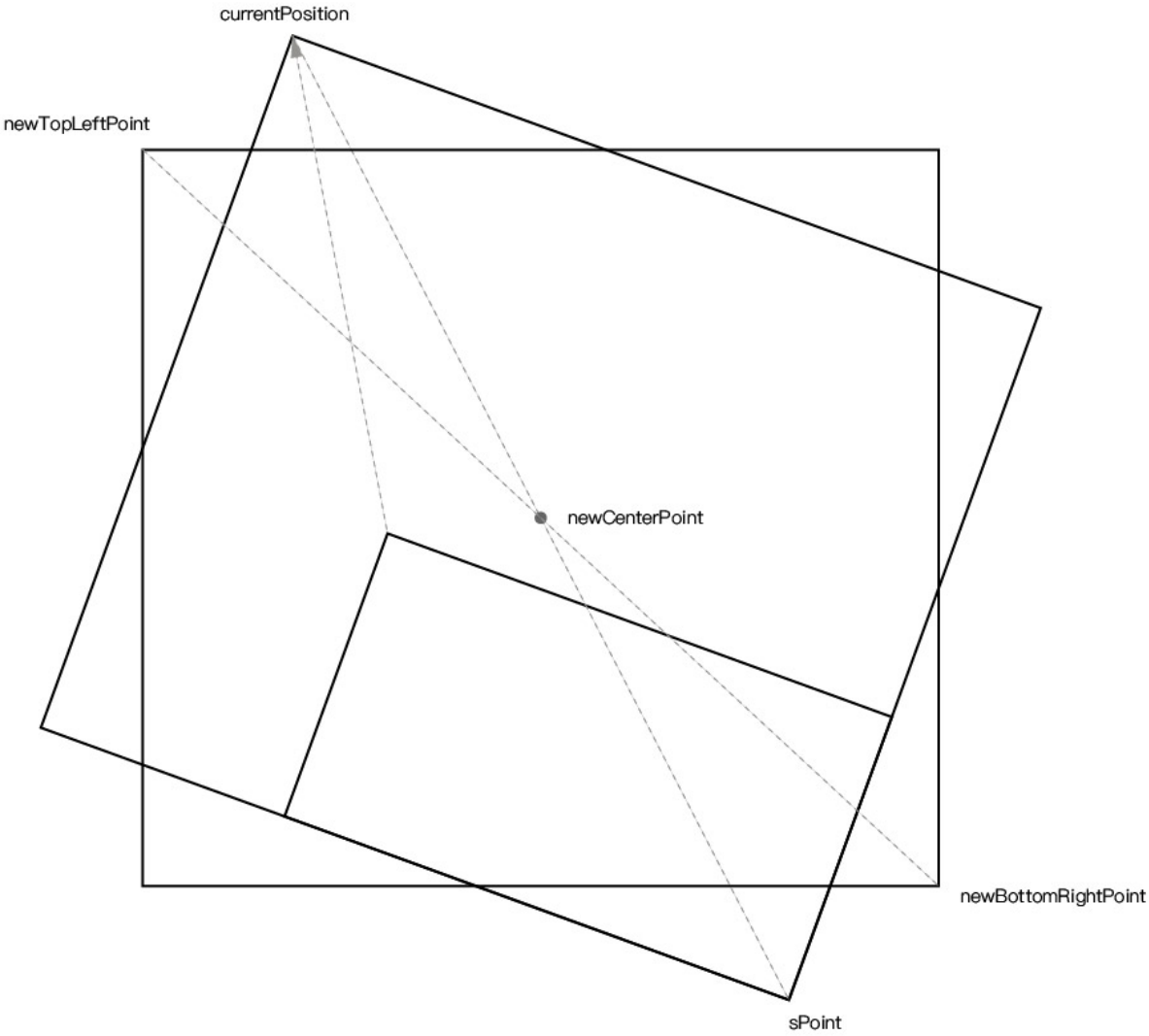
当鼠标摁下瞬间，通过原图形的中心点坐标计算出对称手柄（即右下角）的坐标 `sPoint`
`currentPosition` 为鼠标实时位置
通过以上两点，可以得到拉伸后图形的中心点坐标 `newCenterPoint`



Step 2

以 `newCenterPoint` 为旋转中心， 点 `currentPosition` 通过图形已有的旋转角度，
反向旋转后可以得到图形旋转前，左上角的点 `newTopLeftPoint` ，
右下角的点 `newBottomRightPoint`

有了以上两个对称点，且伸缩规则是 **不锁定** 原图形长宽比的话
即可绘制出拉伸后的图形。



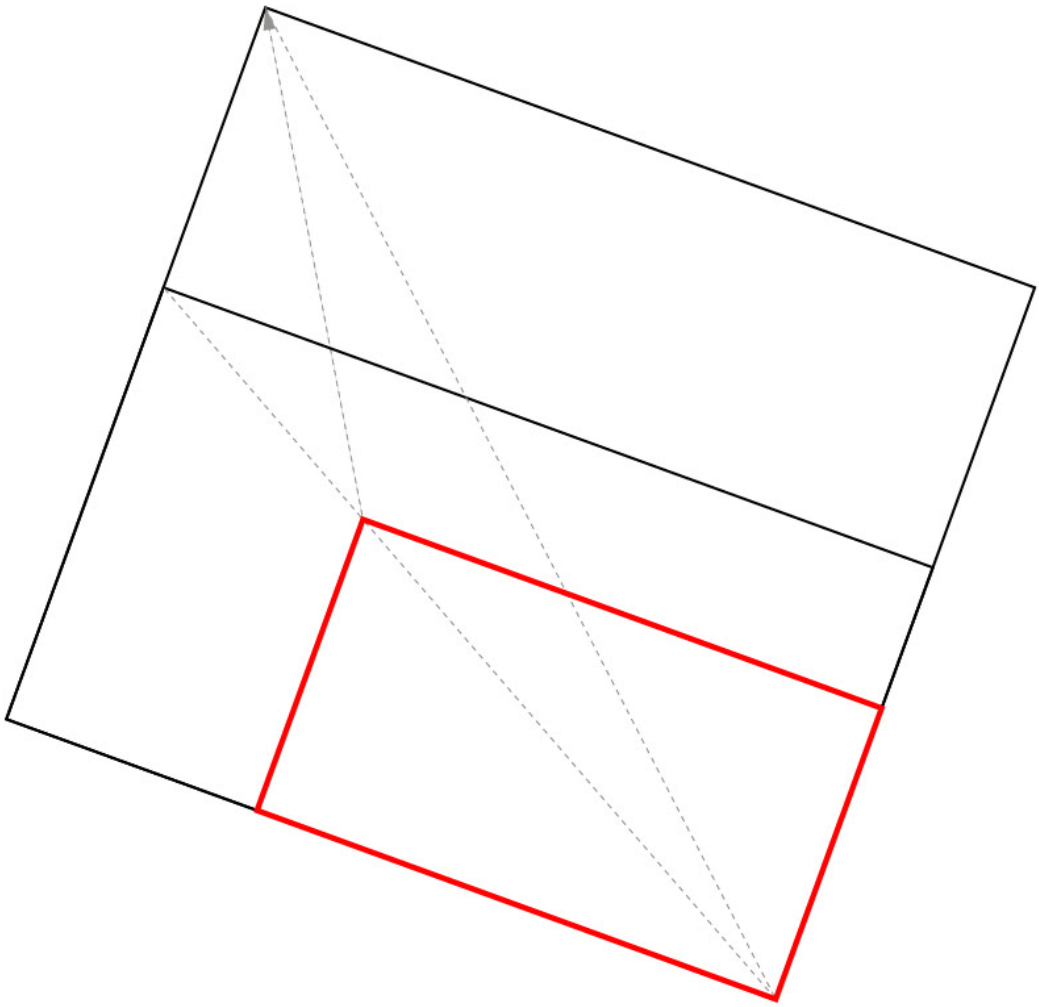
Step 3

但是当伸缩规则是 **锁定** 原图形长宽比，计算将会变得更复杂。
我们需要重新计算，拉伸后的图形的左上角坐标。

这里先确定好几个形状的命名

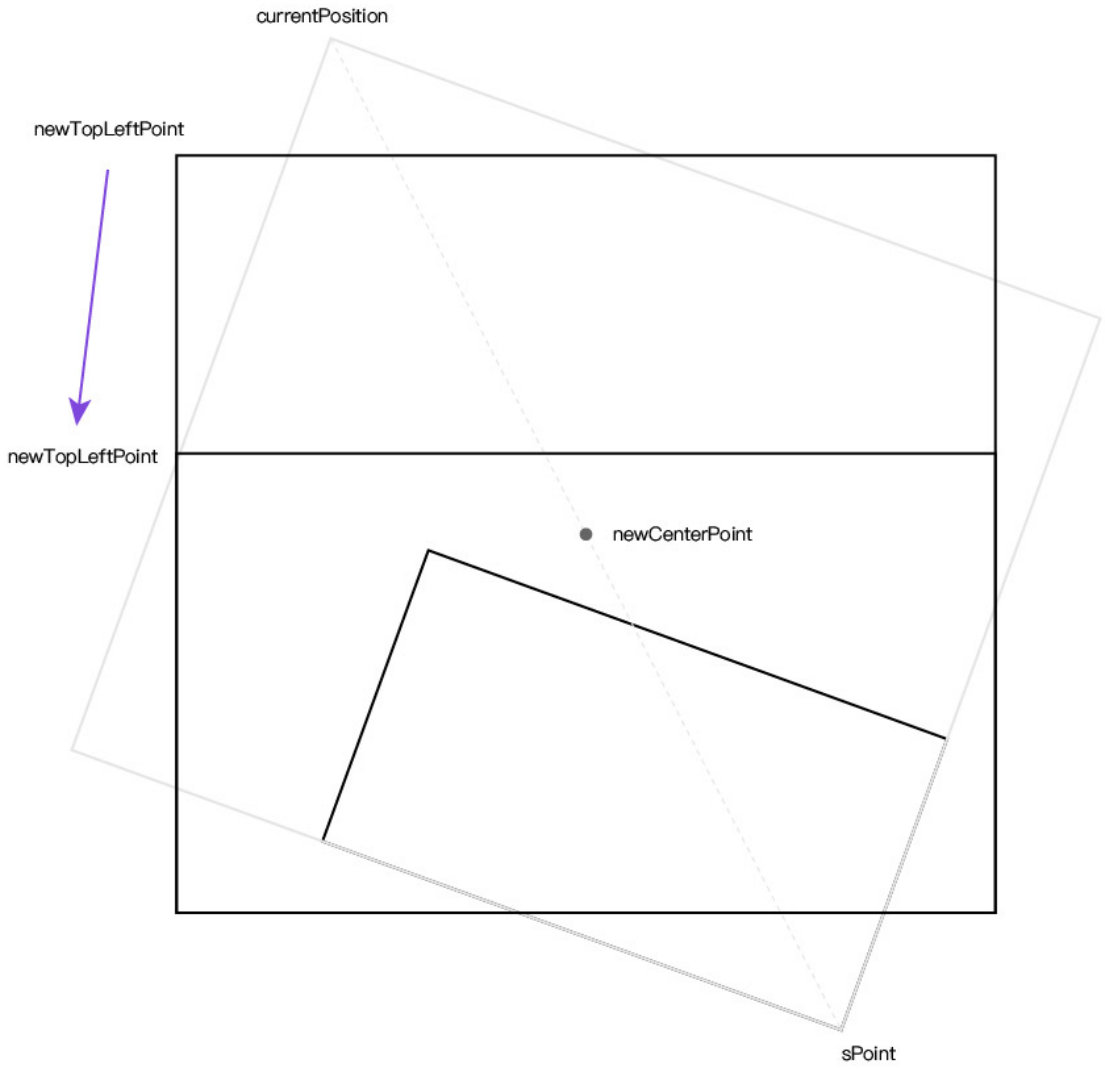
- 原图形： 红色部分
- 新图形： 蓝色部分
- 修正图形：绿色部分，即加上宽高比锁定规则， 新图形 修正后的图形

回到第一步，我们需要再计算多一个变量
当鼠标摁下瞬间，算出原图形的宽高比比值 `proportion`

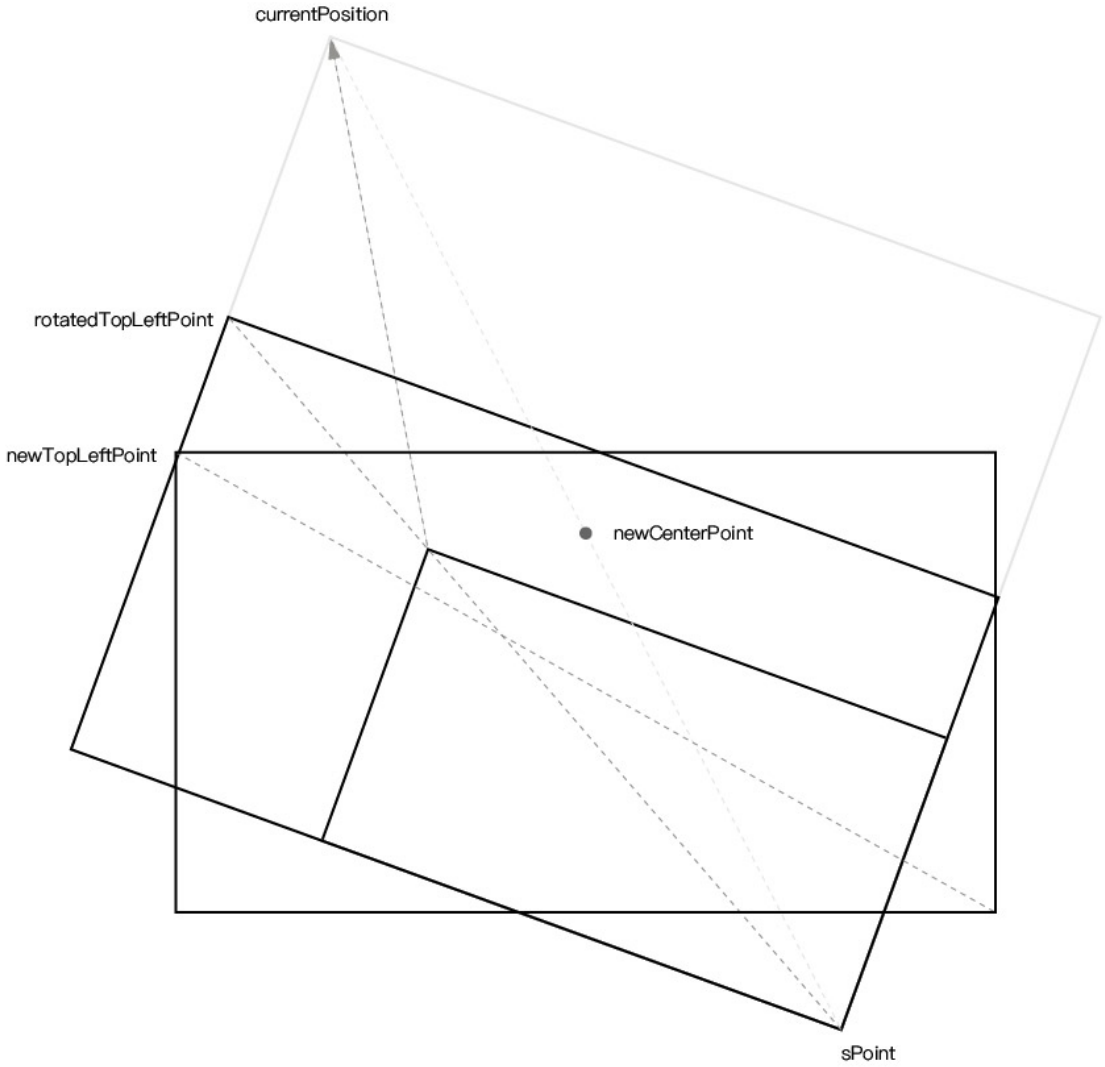


Step 4

这里我们要对 `newTopLeftPoint` 和 `新图形` 的宽或高进行修正：
当 `新图形` 的宽高比比值大于 `proportion`，则修正它的 `x` 坐标与 `宽度`
否则修正 `y` 坐标与 `高度`

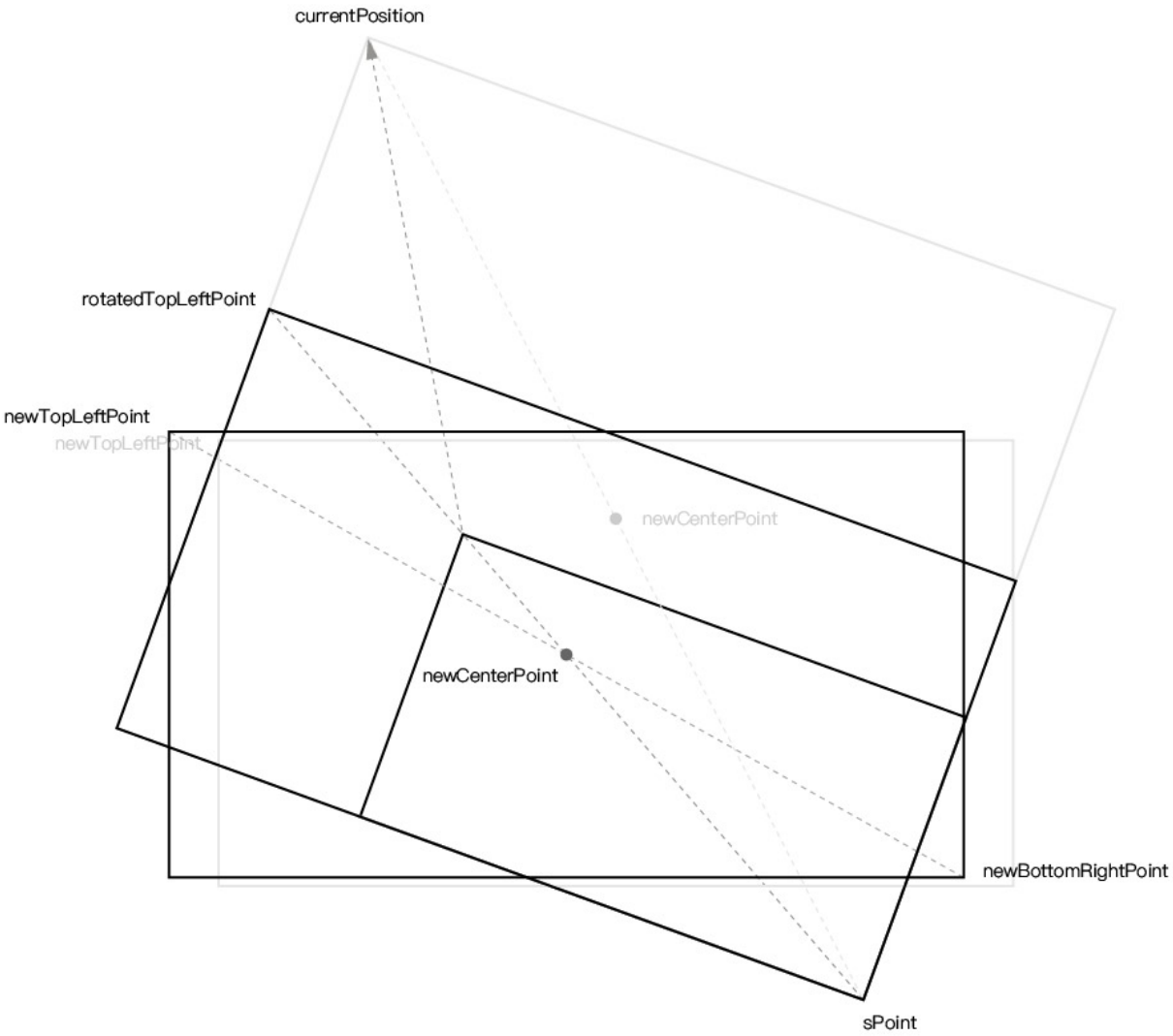


接着将 **修正后** 的 新图形 ，以 `newCenter` 为圆心，
加上 原图形 的旋转角度，算出 修正图形 的左上角的坐标 `rotatedTopLeftPoint`



Step 5

有了 `rotatedTopLeftPoint` 和 `sPoint`
最后就可以计算出 修正图形 的中心点 `newCenter` (原来的 `newCenter` 被覆盖)
进而算出它的旋转前的左上角 `newTopLeftPoint` 和右下角 `newBottomRightPoint`
最后算出修正图形的 xy宽高



Pages 3

Find a Page...

Home

corner handle

拖动图形顶角手柄时的计算步骤

自由伸缩

锁定宽高比伸缩

前提

Step 1

Step 2

Step 3

Step 4

Step 5

▸ side handle

Clone this wiki locally

```
https://github.com/shenhudong/snapping-demo.wiki.git
```