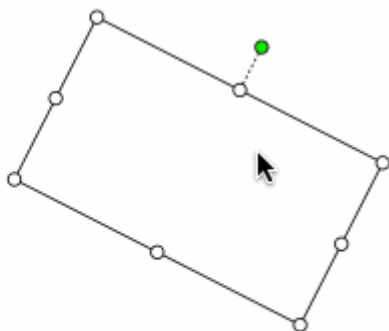


# side handle

[Jump to bottom](#)

wing-kai edited this page on 28 May 2018 · 2 revisions

## 拖动图形边线手柄时的计算步骤



### 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

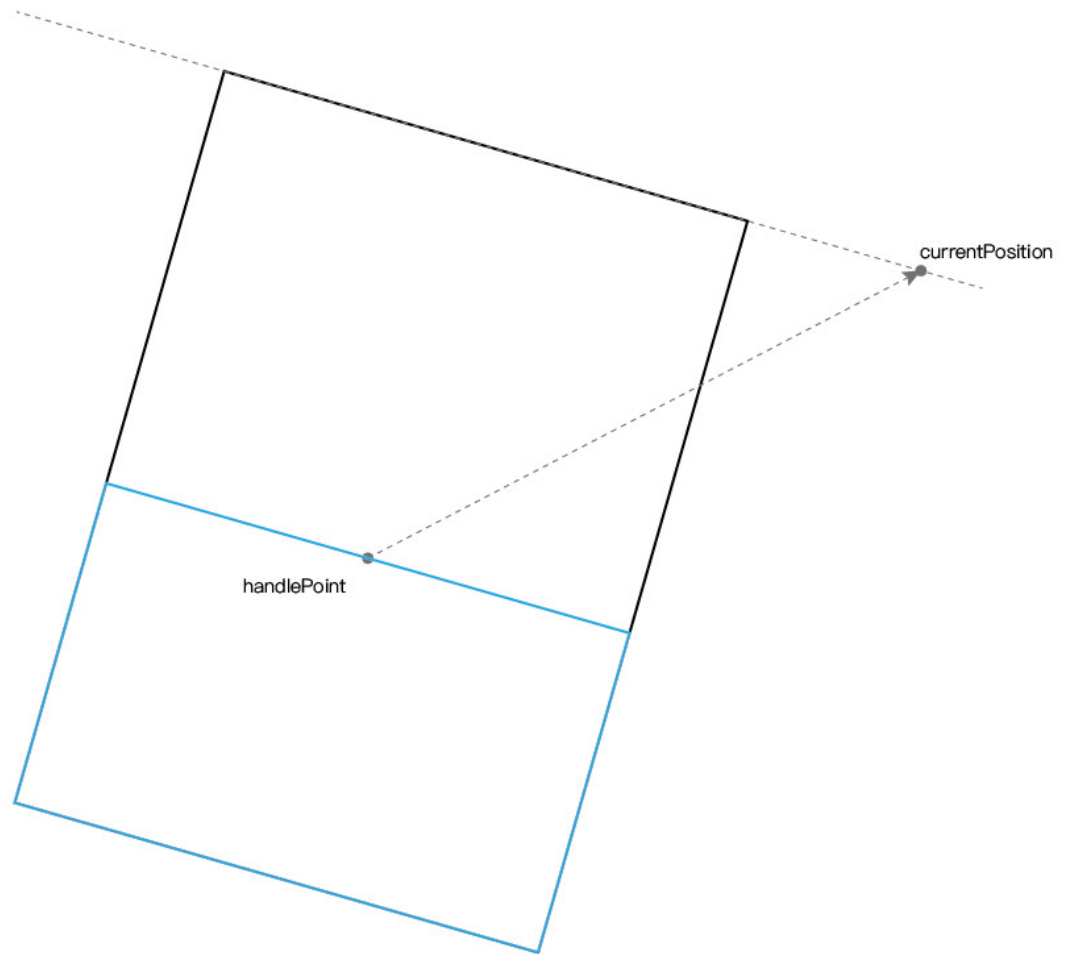
以拖动图形的 **中上方** 手柄为例:

handlePoint 手柄坐标 (即鼠标点下瞬间的指针坐标)

currentPosition 鼠标摁下后移动到的实时坐标

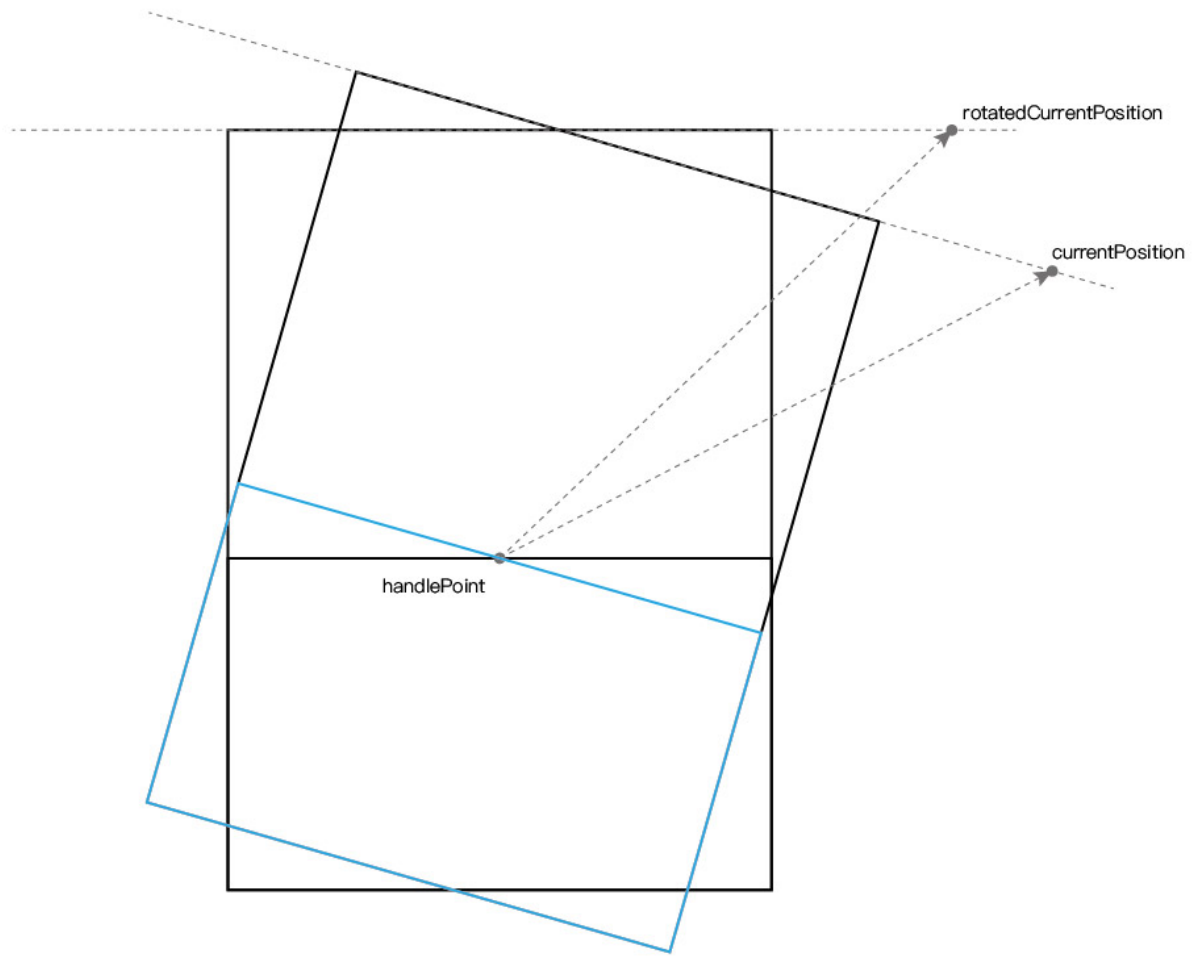
蓝色矩形为原图形的大小

黑色矩形为鼠标指针移动到 currentPosition 时应有的大小



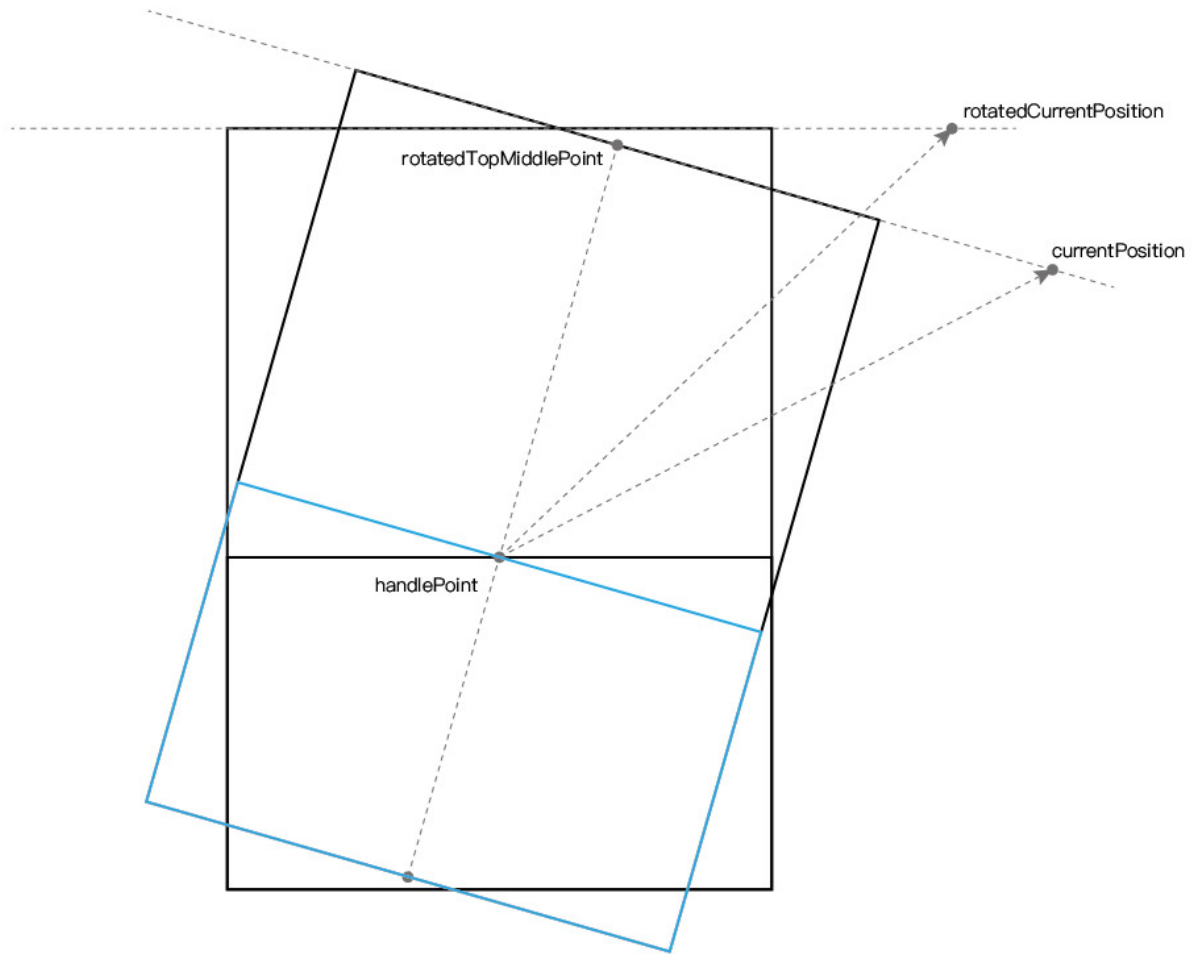
## Step 2

以 handlePoint 为旋转中心, 根据图形的旋转角度,  
可以求得 currentPosition 在反向旋转后的坐标,  
将这个坐标称为 rotatedCurrentPosition



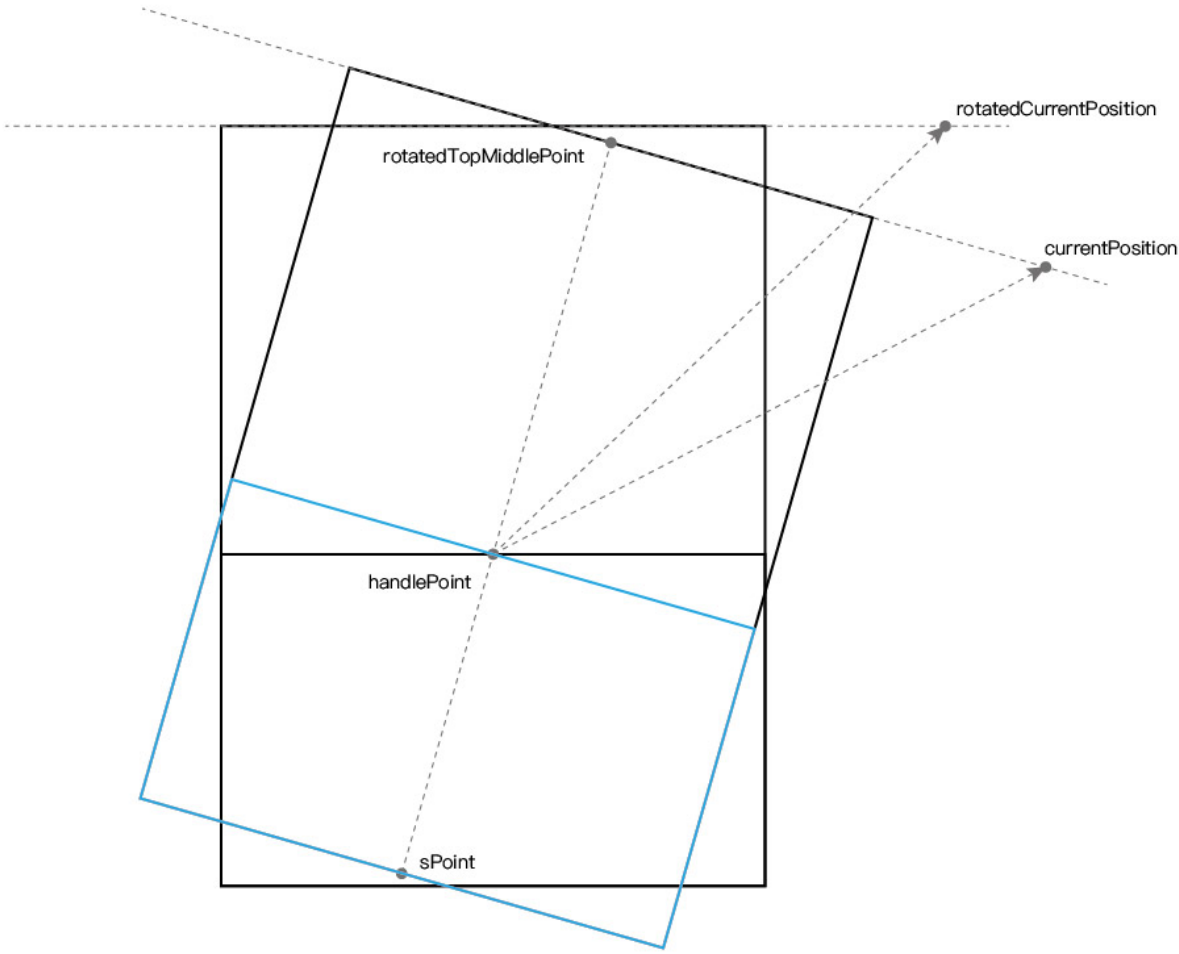
## Step 3

用 `rotatedCurrentPosition` 的Y轴坐标与 `handlePoint` 的X轴坐标组合，  
再根据图形的旋转角度，得到图形拉伸后，正在拖动的手柄的坐标  
将这个坐标命名为 `rotatedTopMiddlePoint` （因为我们拖动的是**中上方**的手柄）



## Step 4

根据 `handlePoint`，计算出它的对称手柄坐标（对应该例子的中下方坐标），将这个坐标称为 `sPoint`，通过它和 `rotatedTopMiddlePoint` 可以获得拉伸后的图形的新高度 `newHeight` 和 它的中心坐标 `newCenter` 最后根据这两点，推理出拉伸后的图形的左上角坐标，从而绘制出拉伸后的图形。



▼ Pages 3

Find a Page...

▶ Home

▶ corner handle

▼ side handle

拖动图形边线手柄时的计算步骤

前提

Step 1

Step 2

Step 3

Step 4

Clone this wiki locally

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

