

## 6. Image Rotation

### 6. Image Rotation

#### 6.1. Image rotation

#### 6.2. Actual effect display

### 6.1. Image rotation

`cv2.getRotationMatrix2D(center, angle, scale)`

Parameter explanation:

center: the center point of rotation

angle: the angle of rotation. Positive numbers are counterclockwise; negative numbers are clockwise.

scale: transformation scale (scaling). 1 means no change, less than 1 means shrinking, and greater than 1 means enlarging.

### 6.2. Actual effect display

Code path:

/home/pi/project\_demo/06.Open\_source\_cv\_fundamentals\_course/B.Geometric\_Transformations  
/06\_Image\_Rotation.ipynb

```
import cv2
import numpy as np
img = cv2.imread('yahboom.jpg',1)
#cv2.imshow('src',img)
imgInfo = img.shape
height = imgInfo[0]
width = imgInfo[1]
# 2*3
# cv2.getRotationMatrix2D(center, angle, scale)
# center: 旋转中心点 Rotation center point
# angle: 旋转角度正数表示逆时针, 负数表示顺时针 Rotation angle, positive number means
counterclockwise, negative number means clockwise
# scale: 变换尺度 Change scale
matRotate = cv2.getRotationMatrix2D((height*0.5, width*0.5), 45, 0.5)# mat
rotate 1 center 2 angle 3 scale
#100*100 25
dst = cv2.warpAffine(img, matRotate, (height,width))
#cv2.imshow('dst',dst)
#cv2.waitKey(0)
```

```
#bgr8转jpeg格式 bgr8 to jpeg format
import enum
import cv2

def bgr8_to_jpeg(value, quality=75):
    return bytes(cv2.imencode('.jpg', value)[1])
```

```
import ipywidgets.widgets as widgets

image_widget1 = widgets.Image(format='jpg', )
image_widget2 = widgets.Image(format='jpg', )
# create a horizontal box container to place the image widget next to eachother
image_container = widgets.HBox([image_widget1, image_widget2])

# display the container in this cell's output
display(image_container)
#display(image_widget2)

img1 = cv2.imread('yahboom.jpg',1)

image_widget1.value = bgr8_to_jpeg(img1)
image_widget2.value = bgr8_to_jpeg(dst)
```

