3.2几何变换

- 图形变换原理：

  将（x,y）变换成（μ,ν）,其中：

$$\mu = a_1 x + b_1 y + c_1; \qquad \nu = a_2 x + b_2 y + c_2$$

  矩阵形式为

$$\begin{bmatrix} \mu \\ \nu \end{bmatrix} = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

  Let $M = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{bmatrix}$, 则可用M表示图像变换

2.1放大/缩小、平移、旋转

- cv2.resize()

```
import cv2
import numpy as np
img = cv2.imread('data/ppang.jpg')

res = cv2.resize(img, None, fx=1/2, fy=1/2)
#使用缩放因子，因此有None；Default：interpolation=cv2.INTER_LINEAR

cv2.imshow('resize', res)
cv2.imshow('src img', img)
v2.waitKey(0)
cv2.destroyAllWindows()
```

- 平移：$M = \begin{bmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \end{bmatrix}$，cv2.wrapAffine()

```
import cv2
import numpy as np

# 移动了100,50 个像素。
img = cv2.imread('data/messi5.jpg', 0)

rows, cols = img.shape
M = np.float32([[1, 0, 100], [0, 1, 50]])
dst = cv2.warpAffine(img, M, (cols, rows))#由于row表示y，col表示x，因此（cols, rows)

cv2.imshow('img1', img)
cv2.imshow('img', dst)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

- 旋转：cv2.getRotationMatrix2D(旋转中心，旋转角度，缩放因子)， cv2.wrapAffine()

```
import cv2
import numpy as np

img = cv2.imread('data/messi5.jpg', 0)
rows, cols = img.shape

# 第一个参数为旋转中心 第二个为旋转角度,第三个为旋转后的缩放因子
M = cv2.getRotationMatrix2D((cols / 2, rows / 2), 45, 2)
dst = cv2.warpAffine(img, M, (2 * cols, 2 * rows))

cv2.imshow('img', dst)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

2.2 Affine变换

- 2.1本质为Affine变换特例
- 原图中平行的直线，在变换后的图像中，仍平行。否则使用2.3透视变换
- 用两点即可,使用cv2.wrapAffineTransfrom()得到M矩阵

```
import cv2
import numpy as np
img = cv2.imread('data/drawing.png')
rows, cols, ch = img.shape

pts1 = np.float32([[50, 50], [200, 50], [50, 200]])
pts2 = np.float32([[10, 100], [200, 50], [100, 250]])
M = cv2.getAffineTransform(pts1, pts2)
dst = cv2.warpAffine(img, M, (cols, rows))
cv2.imshow('img', dst)

cv2.waitKey(0)  # & 0xFF
cv2.destroyAllWindows()
```

2.3透视变换

- 三点，使用cv2.getPerspectiveTransform()获得M矩阵，cv2.wrapPerspective()

```
import cv2
import numpy as np

img = cv2.imread('data/screen_shot.png')
rows, cols, ch = img.shape

pts1 = np.float32([[68, 44], [223, 104], [45,458], [239,438]])
pts2 = np.float32([[0, 0], [cols, 0], [0, rows], [cols, rows]])
M = cv2.getPerspectiveTransform(pts1, pts2)
res = cv2.warpPerspective(img, M, (cols, rows))

cv2.imshow('result', res)
cv2.waitKey(0)
cv2.destroyAllWindows()
```