1.2.1 Affine Map

Affine Transformation(Affine Map) is a linear transformation from two-dimensional coordinates (x, y) to two-dimensional coordinates (u, v).
Its mathematical expression is as follows,

$$\begin{cases} u = a_1 x + b_1 y + c_1 \\ v = a_2 x + b_2 y + c_2 \end{cases}$$

The corresponding homogeneous coordinate matrix mathematical expression is as follow,

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

The affine transformation maintains the "straightness" of the two-dimensional graphics and "parallelism".

Straightness: The straight line will keep a straight line after affine transformation.

Parallelism: After the affine transformation, the parallel line will keep parallel line, and the position order of all points on the straight line will not be changed, and the relative positional relationship between the two straight lines will not be changed.

Non-collinear three pairs of corresponding points can determine an affine transformation.

Affine transformation = picture rotation + picture lift.
Affine transformation also requires an M matrix. Due to the complexity of the affine transformation, it is difficult to find this matrix. OpenCV can automatically solve the matrix M according to the correspondence between the three points before and after the transformation.

M = cv2.getAffineTransform (pos1, pos2), pos1 and pos2 are the corresponding positional relationship before and after the transformation.
The output is the affine matrix M.

Then, we can use the function **cv2.warpAffine ()**.

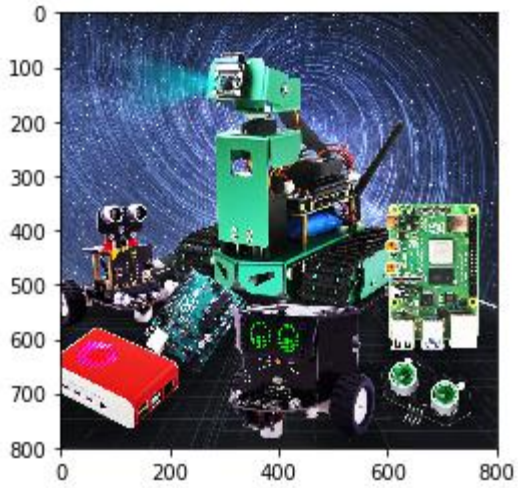
Two methods of image transformation **cv2.warpAffine** and **cv2.warpPerspective**

Code path:
/home/pi/Yahboom_Project/1.OpenCV_course/02Geometric_transformation/05_Affine_Map.ipynb

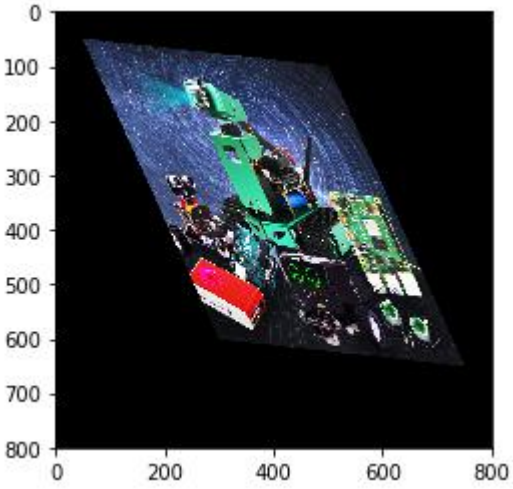
```
import cv2
import numpy as np
import matplotlib.pyplot as plt


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


img_bgr2rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
plt.imshow(img_bgr2rgb)
plt.show()
# cv2.waitKey(0)
```

```
imgInfo = img.shape
height = imgInfo[0]
width = imgInfo[1]
#src 3->dst 3 (Upper left corner, Lower left corner, Upper right corner)
matSrc = np.float32([[0,0], [0,height-1], [width-1,0]])
matDst = np.float32([[50,50], [300,height-200], [width-300,100]])
# Combine
matAffine = cv2.getAffineTransform(matSrc,matDst)# mat 1 src 2 dst
dst = cv2.warpAffine(img,matAffine,(width,height))
img_bgr2rgb = cv2.cvtColor(dst, cv2.COLOR_BGR2RGB)
plt.imshow(img_bgr2rgb)
```

After running the above program, we can see the picture shown below.



[Orginial picture]



[Affine Transformation picture]