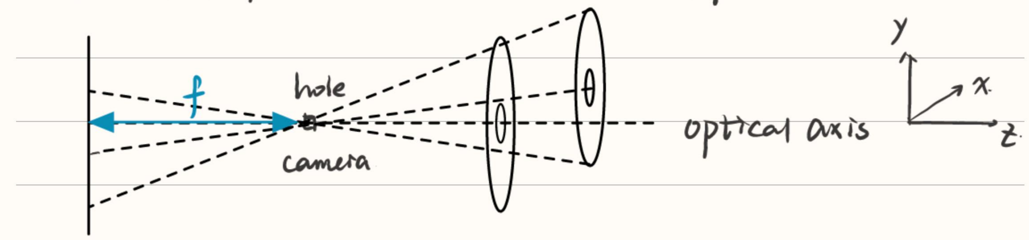
Question 1 (Written Assignment)

a). disk 不一定在光轴上 ⟶ 投影形状是什么?



设物体上的点为 (x_0, y_0, z_0)

对应 image plane 上的点为 (x_i, y_i, f)

$$\frac{z_0}{f} = \frac{x_0}{x_i} = \frac{y_0}{y_i} \Rightarrow x_i = \frac{f}{z_0} x_0 \quad y_i = \frac{f}{z_0} y_0$$

$\therefore$ image 上: $\left(\frac{f}{z_0} x_0, \frac{f}{z_0} y_0, f\right)$

disk $\parallel$ image plane $\Rightarrow$ 一旦 z_0 确定, disk 所在的平面固定.

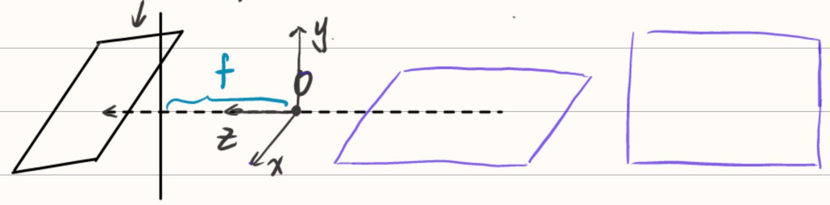
x_i, y_i 与 x_0, y_0 成线性关系, 且 $\Delta x, \Delta y$ 系数相同

$\Rightarrow$ disk 上任意等距两点映射到 image plane 的放大倍数相同

$\Rightarrow$ The shape of image of the disk is still circle.

b) Find vanish point of lines.

Plane: $Ax + By + Cz + D = 0$ 平面过 $\left(-\frac{D}{A}, -\frac{D}{B}, -\frac{D}{C}\right)$



① $A = C = D = 0$ $B = 1$ $\Rightarrow$ $y = 0$. 即为 x-z 构成的平面.

已知物点为 (x_i, y_i, z_i) $\Rightarrow$ 像上点为 $\left(f\frac{x_i}{z_i}, f\frac{y_i}{z_i}, f\right)$

vanishing point: $\left(f\frac{x_i}{z_i}, 0\right)$.

three different line directions in $y=0$, namely $\vec{l} = (x_i, 0, z_i)$

eg. $\vec{l_1} = (1, 0, 1) \Rightarrow$ vanishing point $(1, 0)$

$\vec{l_2} = (10, 0, 2) \Rightarrow (5f, 0)$

$\vec{l_3} = (3, 0, 4) \Rightarrow (0.75f, 0)$

② $B=C=D=0$ & $A=1 \Rightarrow x=0$ 即 y-z 平面 $\therefore x_i=0$

vanishing point $= \left(0, f\frac{y_i}{z_i}\right)$

eg. $\vec{t_1} = (0,2,3) \Rightarrow (0, \frac{2}{3}f)$

$\vec{t_2} = (0,4,8) \Rightarrow (0, 0.5f)$

$\vec{t_3} = (0,1,1) \Rightarrow (0, f)$

C. 定义 general 关系：平面参数 & 某平面上线的消失点

$Ax_i + By_i + Cz_i + D = 0$ 直线方向向量为 $\vec{t} = (x_i, y_i, z_i)$

vanishing point: $\left(f\frac{x_i}{z_i}, f\frac{y_i}{z_i}\right)$ 满足

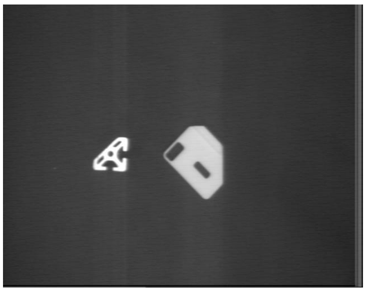
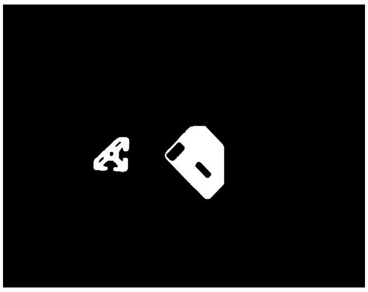
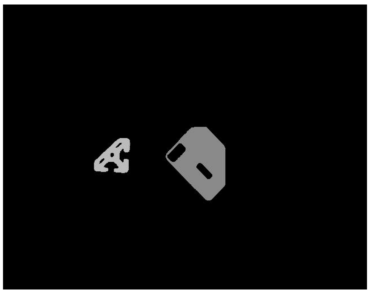
$Ax_i + By_i + Cz_i = 0$

Question 2 (Programming Assignment)

Design choice: use 128 as binary image threshold. The details of algorithm is written in my code. When merge equivalent label, I write a complement function called “get_root” to find same connected component with minimal label. Moreover, to give every connected component different grey colors, i use some linear transformation to make it visiable.

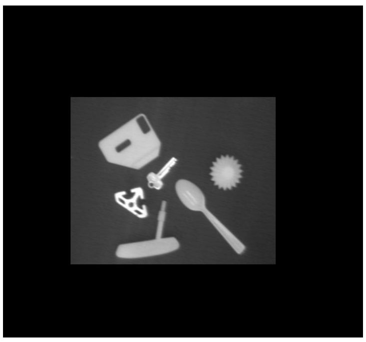
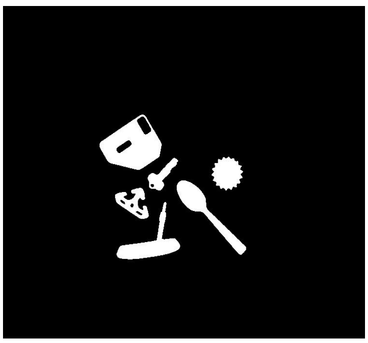
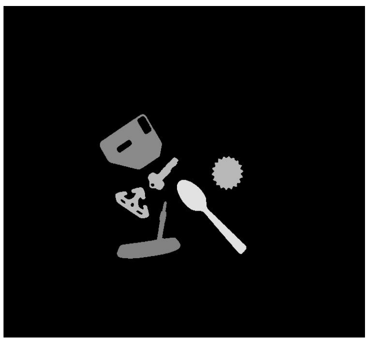
Run : python p1_object_attributes.py picture_name threshold

Output: Two_objects

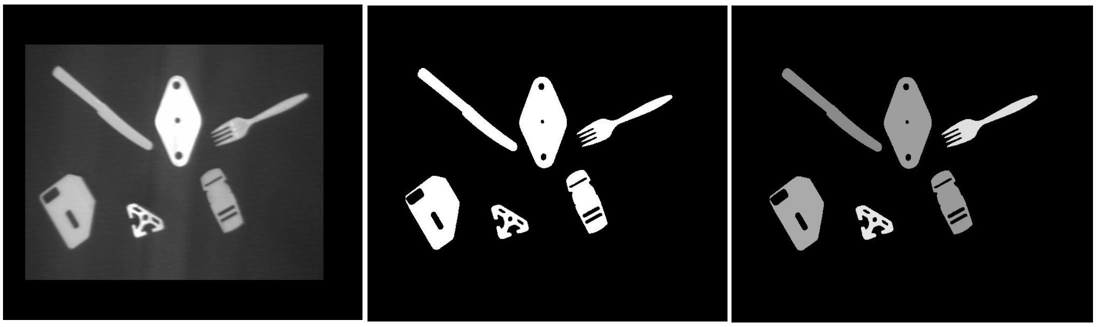

```
[{'position': {'x': 195.3160469667319, 'y': 222.3820939334638}, 'orientation': 0.6875462936637102, 'roundedness': 0.4799636466920244},
{'position': {'x': 349.33298470388286, 'y': 215.45365407242778}, 'orientation': 1.8816361301275777, 'roundedness': 0.5336319534756507}]
```


Many_objects_1

```
[{'position': {'x': 265.97616566814276, 'y': 364.13401927585306}, 'orientation': 0.0804272746023682, 'roundedness': 0.5217196889211242}
 {'position': {'x': 268.30828220858893, 'y': 256.85327198364007}, 'orientation': 2.602755480091518, 'roundedness': 0.48607322060127456}
 {'position': {'x': 303.571394686907, 'y': 177.27300759013283}, 'orientation': 0.4052019927265502, 'roundedness': 0.2702711841586362},
'position': {'x': 326.0154385964912, 'y': 308.29473684210524}, 'orientation': 0.7788385087053653, 'roundedness': 0.1331947199392465}, {
position': {'x': 417.71620665251237, 'y': 240.29181410710072}, 'orientation': 2.3655688092631006, 'roundedness': 0.024421609826596875},
{'position': {'x': 461.6430812129662, 'y': 312.7504356918787}, 'orientation': 1.2635628997702675, 'roundedness': 0.990266442734058}]
```

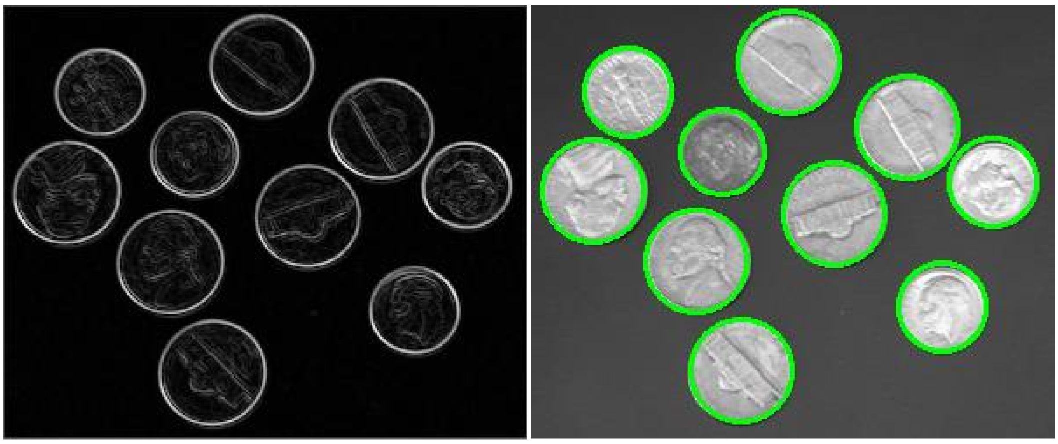

Many_objects_2




```
[{'position': {'x': 130.16157675232074, 'y': 187.1522938248352}, 'orientation': 1.6932113097868666, 'roundedness': 0.5078766943974482},
 {'position': {'x': 188.3515625, 'y': 356.90033143939394}, 'orientation': 2.4984505704173063, 'roundedness': 0.007633528961639356}, {'p
osition': {'x': 265.9671412924425, 'y': 168.6462212486309}, 'orientation': 2.6486233245483937, 'roundedness': 0.48091224785678205}, {'p
osition': {'x': 331.9617982504706, 'y': 337.21769460746316}, 'orientation': 1.6106730812607706, 'roundedness': 0.3072674402498843}, {'p
osition': {'x': 413.655668568 5934, 'y': 203.95137682957082}, 'orientation': 2.02368323627758, 'roundedness': 0.17394416151886896}, {'po
sition': {'x': 475.3399815894446, 'y': 338.9671678428966}, 'orientation': 0.40324741948779197, 'roundedness': 0.02085545128597042}]
```


Question 3 (Programming Assignment)

Design choice: There are two threshold in this algorithm: edge threshold and hough threshold. First use Sobel operator to calculate gradient to get edge image and every pixel magnitude is its normalized gradient. Then use hough transform and vote to find all circles. Finally, find unique circle by constricting the distance of different circle centers. To choose threshold, I have some tricks: use fewer large edge_thresh to make edge clear and use larger hough_thresh to exclude suboptimal circles as much as posible. By above trick and some times trial, I use egde_thresh=120 and hough_thresh=183. By the way, the redius values are from 20 to 29.



```
[(29, 32, 147), (29, 69, 215), (29, 105, 35), (29, 118, 173), (29, 145, 94), (29, 207, 120), (25, 49, 55),
 (25, 100, 264), (25, 171, 234), (24, 83, 109)]
```

CV-HW1

Student_name: 宋源祎

Student_ID：522030910158