AI3604 Computer Vision: Homework 1

Xiangyuan Xue (521030910387)

Written Assignment

1. (a) Since the circular disk is parallel to the image plane, we can assume that it is centered at (x_c, y_c, z_o) with radius r. Its edge can be described by the equation

$$(x_o - x_c)^2 + (y_o - y_c)^2 = r^2, \quad z = z_o$$

For the image plane at $z = f$, we have relationship that

$$\frac{x_i}{f} = \frac{x_o}{z_o}, \quad \frac{y_i}{f} = \frac{y_o}{z_o}$$

Substitute it into the equation, we have

$$\left(\frac{z_o}{f}x_i - x_c\right)^2 + \left(\frac{z_o}{f}y_i - y_c\right)^2 = r^2$$

Hence, the edge on the image plane can be described as

$$\left(x_i - \frac{x_c f}{z_o}\right)^2 + \left(y_i - \frac{y_c f}{z_o}\right)^2 = \left(\frac{rf}{z_o}\right)^2, \quad z = f$$

Therefore, the image is also a circular disk centered at $\left(\frac{x_c f}{z_o}, \frac{y_c f}{z_o}, f\right)$ with radius $\frac{rf}{z_o}$.

(b) For $A = C = D = 0$ and $B = 1$, the plane should be $y = 0$. We choose $(1, 0, 1)$, $(2, 0, 1)$ and $(3, 0, 1)$ as three line directions. According to the formula

$$(x_{vp}, y_{vp}) = \left(\frac{l_x}{l_z}f, \frac{l_y}{l_z}f\right)$$

Their vanishing points are $(f, 0)$, $(2f, 0)$ and $(3f, 0)$, which lie on the line $y = 0$.

For $B = C = D = 0$ and $A = 1$, the plane should be $x = 0$. We choose $(0, 1, 1)$, $(0, 2, 1)$ and $(0, 3, 1)$ as three line directions. Similarly, their vanishing points should be $(0, f)$, $(0, 2f)$ and $(0, 3f)$, which lie on the line $x = 0$.

(c) For the plane $Ax + By + Cz + D = 0$, any line direction (l_x, l_y, l_z) on it should be orthogonal to its normal vector (A, B, C), which yields

$$Al_x + Bl_y + Cl_z = 0$$

The vanishing point is $(x_{vp}, y_{vp}) = \left(\frac{l_x}{l_z}f, \frac{l_y}{l_z}f\right)$. Notice it holds that

$$Ax_{vp} + By_{vp} + Cf = \frac{f}{l_z}(Al_x + Bl_y + Cl_z) = 0$$

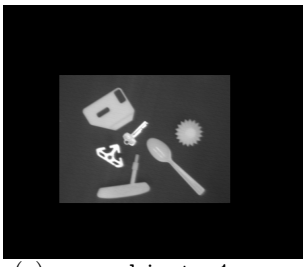
Therefore, all the vanishing points lie on the vanishing line $Ax + By + Cf = 0$.

Programming Assignment

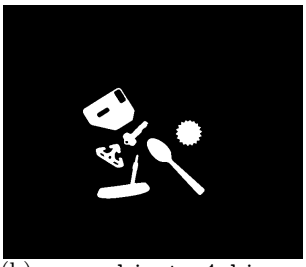
1. (a) Using 128 as the threshold brings satisfying results as shown in 1(b) and 1(f).

   (b) We apply find-union set to maintain the equivalence relation between labels. All the labels are corrected in the second pass. The results are shown in 1(c) and 1(g).

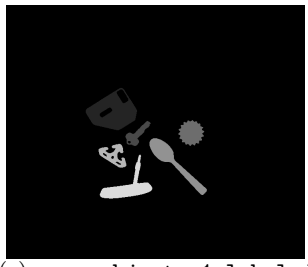
   (c) Apply the formula on each object separately. The output is shown in 1(d) and 1(h).



(a) `many_objects_1_gray`



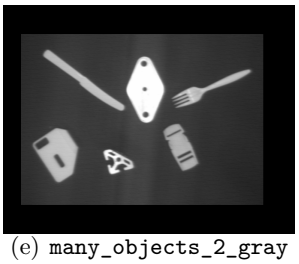
(b) `many_objects_1_binary`



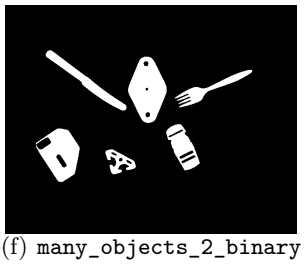
(c) `many_objects_1_labeled`

```
(opencv) xxyqwq@xxyqwq /e/Work/Homework/AI3604/Homework 1/Code python p1_object_attributes.py many_objects_1 128
[{'orientation': 0.08042727460236959,
  'position': {'x': 265.97616566814276, 'y': 364.13401927585306},
  'roundness': 0.5217196889211291},
 {'orientation': 0.7788385087054034,
  'position': {'x': 326.0154385964912, 'y': 308.29473684210524},
  'roundness': 0.1331947199392688},
 {'orientation': 1.2635628997731174,
  'position': {'x': 461.6430812129662, 'y': 312.7504356918787},
  'roundness': 0.9902664427338298},
 {'orientation': -0.7760238443266956,
  'position': {'x': 417.71620665251237, 'y': 240.29181410710072},
  'roundness': 0.02442160982694543},
 {'orientation': -0.5388371734983284,
  'position': {'x': 268.30828220858893, 'y': 256.85327198364007},
  'roundness': 0.4860732206012447},
 {'orientation': 0.40520199272654855,
  'position': {'x': 303.571394686907, 'y': 177.27300759013283},
  'roundness': 0.27027118415863505}]
```

(d) `many_objects_1_attribute`



(e) `many_objects_2_gray`



(f) `many_objects_2_binary`

(g) `many_objects_2_labeled`

```
(opencv) xxyqwq@xxyqwq /e/Work/Homework/AI3604/Homework 1/Code python p1_object_attributes.py many_objects_2 128
[{'orientation': -0.6431420831724862,
  'position': {'x': 188.3515625, 'y': 356.90033143939394},
  'roundness': 0.007633528961638819},
 {'orientation': 1.6106730812607657,
  'position': {'x': 331.9617982504706, 'y': 337.21769460746316},
  'roundness': 0.3072674402498929},
 {'orientation': 0.4032474194877935,
  'position': {'x': 475.3399815894446, 'y': 338.9671678428966},
  'roundness': 0.020855451285964458},
 {'orientation': 2.0236832362775745,
  'position': {'x': 413.6556685685934, 'y': 203.95137682957082},
  'roundness': 0.17394416151886066},
 {'orientation': 1.6932113097868653,
  'position': {'x': 130.16157675232074, 'y': 187.1522938248352},
  'roundness': 0.5078766943974417},
 {'orientation': -0.4929693290413842,
  'position': {'x': 265.9671412924425, 'y': 168.6462212486309},
  'roundness': 0.4809122478567922 6}]
```

(h) `many_objects_2_attribute`

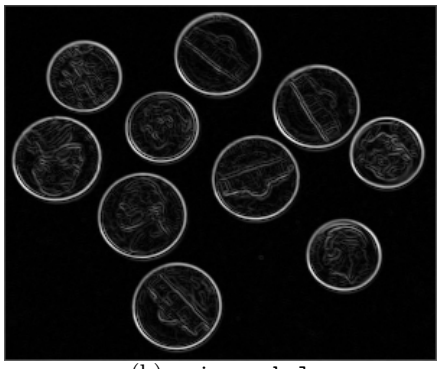
Figure 1: object attributes

2. (a) We implement a two-dimensional convolution as an auxiliary function. We convolve the image with the Sobel operator. The result is shown in 2(b).

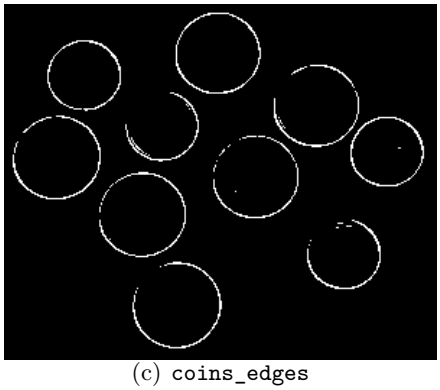
   (b) Since the intensity is normalized to $[0, 1]$, we set the threshold to 0.5 and the result is shown in 2(c). The Hough transform is vectorized to speed up the computation.

(c) We set the threshold to 75 and search the 5×5 neighborhood to find local maxima. All the circles are correctly detected as shown in 2(d). Details are shown in 2(e).

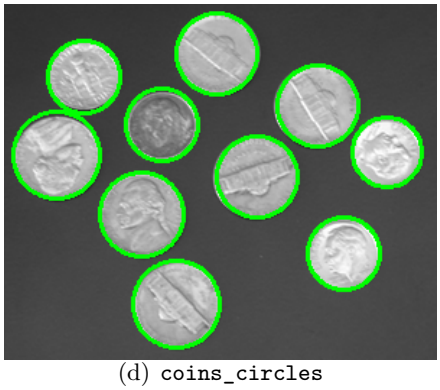
(a) `coins_gray`



(b) `coins_sobel`



(c) `coins_edges`



(d) `coins_circles`

```
(opencv) xxyqwq@xxyqwq /e/Work/Homework/AI3604/Homework 1/Code python p2_hough_circles.py coins 0.5 75
[(24, 102, 265),
 (25, 50, 55),
 (25, 83, 109),
 (25, 172, 235),
 (28, 34, 147),
 (28, 70, 217),
 (28, 119, 175),
 (29, 145, 95),
 (30, 104, 36),
 (30, 207, 119)]
```

(e) `coins_location`

Figure 2: hough circles

Appendix

To reproduce the results, run the following commands in terminal.

```
$ python p1_object_attributes.py many_objects_1 128
$ python p1_object_attributes.py many_objects_2 128
$ python p2_hough_circles.py coins 0.5 75
```