Project 3: Perspective Projections

I set u0 and v0 to be the extra offset in projection image. The projection will be in the center of output image, if u0 = 0 and v0 = 0. The output image will use the same size as the input image for the sake of contrast and aesthetics.

1. Official document.

A screenshot of a cell phone

Description automatically generated

cv2. warpPerspective(src, M, dsize)

The src is assigned to the input image. In order to keep the output image the same size as the input image, we set dsize = (cols, rows) which is the size of the input image. So, the most important thing is to determine the transformation matrix M.

2. Parameters in parts I.

A close up of a map

Description automatically generated

While our input image and output projection are shown with image coordinates, warpPerspective will transform the image with a geometry center in camera coordinates’ center. If we want to implement all the transformation based on the image’s center, we should do the following coordinate transformation.

image’s image coordinates -> image’s camera coordinates ->

projection’s camera coordinates -> projection’s image coordinates

If we use w to represent width and h to represent height. To implement the transformation between image coordinates and camera coordinates, we have:

3. Parameters in parts II.

In part II, I am asked to apply two images with hard coded. Besides c, we should decide the other parameters.

1) u0 & v0. To make the output image in the center, I set u0 = 0 and v0 = 0.

2) a & b.

Normal vector of plane: (a, b, c)

Direction vector of x axis (Normal vector of yoz): (1, 0, 0)

Direction vector of y axis (Normal vector of xoz): (0, 1, 0)

Vector a = (p1, q1, r1) ・Vector b = (p2, q2, r2) = p1p2 + q1q2 + r1r2

i) part I. For vertically flipping, plane should always perpendicular to the xoz plane.

(a, b, c) ・ (0, 1, 0) = 0

b = 0

If I want the left side is closer than the right side,

(a, 0, c)・(1, 0, 0) > 0

a < 0

ii) part II. For horizontally flipping, plane should always perpendicular to the yoz plane.

(a, b, c) ・ (1, 0, 0) = 0

a = 0

If I want the bottom side is closer than the top side.

(0, b, c)・(0, 1, 0) < 0

b > 0

3) f. Since I keep the output image the same size as the input image, f should be equal to c.

4. Other issues

1) Constraints

Since we discuss the project based on pinhole camera model, I set several simple constraints on the parameters.

i) f > 0. Distance f should be a positive number, otherwise we could not get a projection through the pinhole.

ii) c > 0. Intercept c should be a positive number, otherwise the image is on the same side of the projection which is also meaningless.

iii) Other parameters can be any number.

These constraints are set based on real world. With parameters beyond the range, we can still get valuable but meaningless result.

2) Virtual image

When I tested various parameters, I found that some images that should not appear show on the other side. After deeply thinking about it, I found that another part of the image was projected from the projection side.

A close up of a map

Description automatically generated

The proof is easy. I mark 8 points (A1, A2, A3, A4, B1, B2, B3, B4) in input image and (A1‘, A2’, A3‘, A4’, B1‘, B2’, B3‘, B4’) in output image. The position of 8 points is compared back and forth. As it can be seen from the schematic diagram, A1’A2’ is obtained from the image side from A1A2. A3’A4’ is obtained from the image side from A3A4. B1’B2’ is obtained from the image side from B1B2. B3’B4’ is obtained from the image side from B3B4. Extra part is obtained from the projection side which cause A3, A4, B3, B4 and A3’, A4’, B3’, B4’ are upside down.

A picture containing indoor, book, shelf, sitting

Description automatically generatedA picture containing umbrella

Description automatically generated