Computer Vision: Assignment #2

Professor Ling Zhang

2252709
Xuanhe Yang

December, 2024

Problem 1

For the line equation $x - 3y + 4 = 0$, the homogeneous coordinates of the point at infinity can be determined as follows.
For a general line equation $ax + by + c = 0$, the homogeneous coordinates of the corresponding point at infinity are $(a, b, 0)$.
In this case, the line equation is $x - 3y + 4 = 0$, so:

$$a = 1, \quad b = -3$$

Thus, the homogeneous coordinates of the point at infinity for the line $x - 3y + 4 = 0$ are:

$$(1, -3, 0)$$

Problem 2

Let me help solve this step by step to find the Jacobian matrix dp_d/dp_n. 1) First, let's clarify what we're computing. The Jacobian matrix will be 2×2:

$$\frac{dp_d}{dp_n} = \begin{bmatrix} \frac{\partial x_d}{\partial x} & \frac{\partial x_d}{\partial y} \\ \frac{\partial y_d}{\partial x} & \frac{\partial y_d}{\partial y} \end{bmatrix}$$

2) First equation for x_d:

$$x_d = x(1 + k_1 r^2 + k_2 r^4) + 2\rho_1 xy + \rho_2(r^2 + 2x^2) + xk_3 r^6$$

3) Second equation for y_d:

$$y_d = y(1 + k_1 r^2 + k_2 r^4) + 2\rho_2 xy + \rho_1(r^2 + 2y^2) + yk_3 r^6$$

4) Let's compute each partial derivative: a) For $\partial x_d/\partial x$:

$$\begin{aligned} \frac{\partial x_d}{\partial x} &= 1 + k_1 r^2 + k_2 r^4 + k_3 r^6 + k_1(2x^2) + k_2(4x^2 r^2) + k_3(6x^2 r^4) + 2\rho_1 y + \rho_2(2x + 4x) \\ &= 1 + k_1(r^2 + 2x^2) + k_2(r^4 + 4x^2 r^2) + k_3(r^6 + 6x^2 r^4) + 2\rho_1 y + 6\rho_2 x \end{aligned} \tag{1}$$

b) For $\partial x_d/\partial y$:

$$\begin{aligned} \frac{\partial x_d}{\partial y} &= x(2k_1 y + 4k_2 r^2 y + 6k_3 r^4 y) + 2\rho_1 x + 2\rho_2 y \\ &= 2xy(k_1 + 2k_2 r^2 + 3k_3 r^4) + 2\rho_1 x + 2\rho_2 y \end{aligned} \tag{2}$$

c) For $\partial y_d/\partial x$:

$$\begin{aligned} \frac{\partial y_d}{\partial x} &= y(2k_1 x + 4k_2 r^2 x + 6k_3 r^4 x) + 2\rho_2 y + 2\rho_1 x \\ &= 2xy(k_1 + 2k_2 r^2 + 3k_3 r^4) + 2\rho_2 y + 2\rho_1 x \end{aligned} \tag{3}$$

d) For $\partial y_d/\partial y$:

$$\begin{aligned} \frac{\partial y_d}{\partial y} &= 1 + k_1 r^2 + k_2 r^4 + k_3 r^6 + k_1(2y^2) + k_2(4y^2 r^2) + k_3(6y^2 r^4) + 2\rho_2 x + \rho_1(2y + 4y) \\ &= 1 + k_1(r^2 + 2y^2) + k_2(r^4 + 4y^2 r^2) + k_3(r^6 + 6y^2 r^4) + 2\rho_2 x + 6\rho_1 y \end{aligned} \tag{4}$$

Problem 2 continued on next page...

5) Therefore, the complete Jacobian matrix is:

$$\frac{dp_d}{dp_n} = \begin{bmatrix} 1+k_1(r^2+2x^2)+k_2(r^4+4x^2r^2)+k_3(r^6+6x^2r^4)+2\rho_1 y+6\rho_2 x & 2xy(k_1+2k_2r^2+3k_3r^4)+2\rho_1 x+2\rho_2 y \\ 2xy(k_1+2k_2r^2+3k_3r^4)+2\rho_2 y+2\rho_1 x & 1+k_1(r^2+2y^2)+k_2(r^4+4y^2r^2)+k_3(r^6+6y^2r^4)+2\rho_2 x+6\rho_1 y \end{bmatrix}$$

where $r^2 = x^2 + y^2$. This Jacobian matrix represents the local linear approximation of how changes in the input coordinates (p_n) affect the distorted coordinates (p_d).

Problem 3

Derivation of the Jacobian Matrix for Rotation Matrix with respect to Axis-Angle Representation

Problem Statement

Given:

- $\mathbf{d} = \theta\mathbf{n}$, where $\mathbf{n} = [n_1, n_2, n_3]^T$ is a 3D unit vector
- $\mathbf{R} = \cos\theta\mathbf{I} + (1 - \cos\theta)\mathbf{n}\mathbf{n}^T + \sin\theta\hat{\mathbf{n}}$
- Let $\alpha = \sin\theta$, $\beta = \cos\theta$, $\gamma = 1 - \cos\theta$

Derivation Steps

Step 1: Expand Rotation Matrix

$\hat{\mathbf{n}}$ is the skew-symmetric matrix:

$$\hat{\mathbf{n}} = \begin{bmatrix} 0 & -n_3 & n_2 \\ n_3 & 0 & -n_1 \\ -n_2 & n_1 & 0 \end{bmatrix}$$

$\mathbf{n}\mathbf{n}^T$ is:

$$\mathbf{n}\mathbf{n}^T = \begin{bmatrix} n_1^2 & n_1n_2 & n_1n_3 \\ n_1n_2 & n_2^2 & n_2n_3 \\ n_1n_3 & n_2n_3 & n_3^2 \end{bmatrix}$$

Therefore, $\mathbf{R}$ expands to:

$$\mathbf{R} = \begin{bmatrix} \beta + \gamma n_1^2 & \gamma n_1n_2 - \alpha n_3 & \gamma n_1n_3 + \alpha n_2 \\ \gamma n_1n_2 + \alpha n_3 & \beta + \gamma n_2^2 & \gamma n_2n_3 - \alpha n_1 \\ \gamma n_1n_3 - \alpha n_2 & \gamma n_2n_3 + \alpha n_1 & \beta + \gamma n_3^2 \end{bmatrix}$$

Step 2: Basic Derivative Relations

Derivatives with respect to θ

$$\frac{\partial\alpha}{\partial\theta} = \beta$$
$$\frac{\partial\beta}{\partial\theta} = -\alpha$$
$$\frac{\partial\gamma}{\partial\theta} = \alpha$$

Problem 3 continued on next page...

Derivatives with respect to d

Given $\theta = \|\mathbf{d}\| = \sqrt{d_1^2 + d_2^2 + d_3^2}$:

$$\frac{\partial \theta}{\partial d_i} = \frac{d_i}{\theta} = n_i$$

For unit vector components:

$$\frac{\partial n_i}{\partial d_j} = \frac{\partial}{\partial d_j}\left(\frac{d_i}{\theta}\right) = \frac{\delta_{ij}}{\theta} - \frac{d_i d_j}{\theta^3} = \frac{\delta_{ij} - n_i n_j}{\theta}$$

Chain Rule Applications

$$\begin{aligned}
\frac{\partial \alpha}{\partial d_i} &= \frac{\partial \alpha}{\partial \theta}\frac{\partial \theta}{\partial d_i} = \beta n_i \\
\frac{\partial \beta}{\partial d_i} &= \frac{\partial \beta}{\partial \theta}\frac{\partial \theta}{\partial d_i} = -\alpha n_i \\
\frac{\partial \gamma}{\partial d_i} &= \frac{\partial \gamma}{\partial \theta}\frac{\partial \theta}{\partial d_i} = \alpha n_i
\end{aligned}$$

Step 3: Complete Derivation for All Elements

For $r_{11} = \beta + \gamma n_1^2$:

$$\begin{aligned}
\frac{\partial r_{11}}{\partial d_1} &= -\alpha n_1 + n_1^2 \alpha n_1 + 2\gamma n_1 \frac{1 - n_1^2}{\theta} \\
&= \frac{2\gamma n_1(1 - n_1^2)}{\theta} + \alpha n_1(n_1^2 - 1) \\
\frac{\partial r_{11}}{\partial d_2} &= -\alpha n_2 + n_1^2 \alpha n_2 + 2\gamma n_1 \frac{-n_1 n_2}{\theta} \\
&= -\frac{2\gamma n_1^2 n_2}{\theta} + \alpha n_2(n_1^2 - 1) \\
\frac{\partial r_{11}}{\partial d_3} &= -\alpha n_3 + n_1^2 \alpha n_3 + 2\gamma n_1 \frac{-n_1 n_3}{\theta} \\
&= -\frac{2\gamma n_1^2 n_3}{\theta} + \alpha n_3(n_1^2 - 1)
\end{aligned}$$

For $r_{12} = \gamma n_1 n_2 - \alpha n_3$:

$$\begin{aligned}
\frac{\partial r_{12}}{\partial d_1} &= n_2 \alpha n_1 + \gamma \frac{n_2(1 - n_1^2) - n_1^2 n_2}{\theta} - n_3 \beta n_1 \\
&= n_1(n_1 n_2 - n_3) + \frac{\gamma n_2(1 - 2n_1^2)}{\theta} \\
\frac{\partial r_{12}}{\partial d_2} &= n_1 \alpha n_2 + \gamma \frac{n_1(1 - n_2^2) - n_1 n_2^2}{\theta} - n_3 \beta n_2 \\
&= n_2(n_1 n_2 - n_3) + \frac{\gamma n_1(1 - 2n_2^2)}{\theta} \\
\frac{\partial r_{12}}{\partial d_3} &= n_1 n_2 \alpha n_3 - 2\gamma \frac{n_1 n_2 n_3}{\theta} - (\beta n_3 n_3 + \alpha) \\
&= n_3(n_1 n_2 - n_3) + \frac{n_3^2 - 1 - 2n_1 n_2 n_3}{\theta}
\end{aligned}$$

Problem 3 continued on next page...

For $r_{13} = \gamma n_1 n_3 + \alpha n_2$:

$$\begin{aligned}
\frac{\partial r_{13}}{\partial d_1} &= n_3 \alpha n_1 + \gamma \frac{n_3(1-n_1^2) - n_1^2 n_3}{\theta} + n_2 \beta n_1 \\
&= n_1(n_1 n_3 + n_2) + \frac{\gamma n_3(1-2n_1^2)}{\theta} \\
\frac{\partial r_{13}}{\partial d_2} &= n_1 n_3 \alpha n_2 - 2\gamma \frac{n_1 n_2 n_3}{\theta} + (\beta + \alpha n_2^2) \\
&= n_2(n_1 n_3 + n_2) + \frac{1 - n_2^2 - 2n_1 n_2 n_3}{\theta} \\
\frac{\partial r_{13}}{\partial d_3} &= n_1 \alpha n_3 + \gamma \frac{n_1(1-n_3^2) - n_1 n_3^2}{\theta} + n_2 \beta n_3 \\
&= n_3(n_1 n_3 + n_2) + \frac{\gamma n_1(1-2n_3^2)}{\theta}
\end{aligned}$$

For $r_{21} = \gamma n_1 n_2 + \alpha n_3$:

$$\begin{aligned}
\frac{\partial r_{21}}{\partial d_1} &= n_2 \alpha n_1 + \gamma \frac{n_2(1-n_1^2) - n_1^2 n_2}{\theta} + n_3 \beta n_1 \\
&= n_1(n_1 n_2 + n_3) + \frac{\gamma n_2(1-2n_1^2)}{\theta} \\
\frac{\partial r_{21}}{\partial d_2} &= n_1 \alpha n_2 + \gamma \frac{n_1(1-n_2^2) - n_1 n_2^2}{\theta} + n_3 \beta n_2 \\
&= n_2(n_1 n_2 + n_3) + \frac{\gamma n_1(1-2n_2^2)}{\theta} \\
\frac{\partial r_{21}}{\partial d_3} &= n_1 n_2 \alpha n_3 - 2\gamma \frac{n_1 n_2 n_3}{\theta} + (\beta + \alpha n_3^2) \\
&= n_3(n_1 n_2 + n_3) + \frac{1 - n_3^2 - 2n_1 n_2 n_3}{\theta}
\end{aligned}$$

For $r_{22} = \beta + \gamma n_2^2$:

$$\begin{aligned}
\frac{\partial r_{22}}{\partial d_1} &= -\alpha n_1 + n_2^2 \alpha n_1 + 2\gamma n_2 \frac{-n_1 n_2}{\theta} \\
&= -\frac{2\gamma n_1 n_2^2}{\theta} + \alpha n_1(n_2^2 - 1) \\
\frac{\partial r_{22}}{\partial d_2} &= -\alpha n_2 + n_2^2 \alpha n_2 + 2\gamma n_2 \frac{1 - n_2^2}{\theta} \\
&= \frac{2\gamma n_2(1-n_2^2)}{\theta} + \alpha n_2(n_2^2 - 1) \\
\frac{\partial r_{22}}{\partial d_3} &= -\alpha n_3 + n_2^2 \alpha n_3 + 2\gamma n_2 \frac{-n_2 n_3}{\theta} \\
&= -\frac{2\gamma n_2^2 n_3}{\theta} + \alpha n_3(n_2^2 - 1)
\end{aligned}$$

For $r_{23} = \gamma n_2 n_3 - \alpha n_1$:

$$\begin{aligned}
\frac{\partial r_{23}}{\partial d_1} &= n_2 n_3 \alpha n_1 - 2\gamma \frac{n_1 n_2 n_3}{\theta} - (\beta + \alpha n_1^2) \\
&= n_1(n_2 n_3 - n_1) + \frac{-1 + n_1^2 - 2n_1 n_2 n_3}{\theta} \\
\frac{\partial r_{23}}{\partial d_2} &= n_3 \alpha n_2 + \gamma \frac{n_3(1 - n_2^2) - n_2^2 n_3}{\theta} - n_1 \beta n_2 \\
&= n_2(n_2 n_3 - n_1) + \frac{\gamma n_3(1 - 2n_2^2)}{\theta} \\
\frac{\partial r_{23}}{\partial d_3} &= n_2 \alpha n_3 + \gamma \frac{n_2(1 - n_3^2) - n_2 n_3^2}{\theta} - n_1 \beta n_3 \\
&= n_3(n_2 n_3 - n_1) + \frac{\gamma n_2(1 - 2n_3^2)}{\theta}
\end{aligned}$$

For $r_{31} = \gamma n_1 n_3 - \alpha n_2$:

$$\begin{aligned}
\frac{\partial r_{31}}{\partial d_1} &= n_3 \alpha n_1 + \gamma \frac{n_3(1 - n_1^2) - n_1^2 n_3}{\theta} - n_2 \beta n_1 \\
&= n_1(n_1 n_3 - n_2) + \frac{\gamma n_3(1 - 2n_1^2)}{\theta} \\
\frac{\partial r_{31}}{\partial d_2} &= n_1 n_3 \alpha n_2 - 2\gamma \frac{n_1 n_2 n_3}{\theta} - (\beta + \alpha n_2^2) \\
&= n_2(n_1 n_3 - n_2) + \frac{-1 + n_2^2 - 2n_1 n_2 n_3}{\theta} \\
\frac{\partial r_{31}}{\partial d_3} &= n_1 \alpha n_3 + \gamma \frac{n_1(1 - n_3^2) - n_1 n_3^2}{\theta} - n_2 \beta n_3 \\
&= n_3(n_1 n_3 - n_2) + \frac{\gamma n_1(1 - 2n_3^2)}{\theta}
\end{aligned}$$

For $r_{32} = \gamma n_2 n_3 + \alpha n_1$:

$$\begin{aligned}
\frac{\partial r_{32}}{\partial d_1} &= n_2 n_3 \alpha n_1 - 2\gamma \frac{n_1 n_2 n_3}{\theta} + (\beta + \alpha n_1^2) \\
&= n_1(n_2 n_3 + n_1) + \frac{1 - n_1^2 - 2n_1 n_2 n_3}{\theta} \\
\frac{\partial r_{32}}{\partial d_2} &= n_3 \alpha n_2 + \gamma \frac{n_3(1 - n_2^2) - n_2^2 n_3}{\theta} + n_1 \beta n_2 \\
&= n_2(n_2 n_3 + n_1) + \frac{\gamma n_3(1 - 2n_2^2)}{\theta} \\
\frac{\partial r_{32}}{\partial d_3} &= n_2 \alpha n_3 + \gamma \frac{n_2(1 - n_3^2) - n_2 n_3^2}{\theta} + n_1 \beta n_3 \\
&= n_3(n_2 n_3 + n_1) + \frac{\gamma n_2(1 - 2n_3^2)}{\theta}
\end{aligned}$$

For $r_{33} = \beta + \gamma n_3^2$:

$$
\begin{aligned}
\frac{\partial r_{33}}{\partial d_1} &= -\alpha n_1 + n_3^2 \alpha n_1 + 2\gamma n_3 \frac{-n_1 n_3}{\theta} \\
&= -\frac{2\gamma n_1 n_3^2}{\theta} + \alpha n_1(n_3^2 - 1) \\
\frac{\partial r_{33}}{\partial d_2} &= -\alpha n_2 + n_3^2 \alpha n_2 + 2\gamma n_3 \frac{-n_2 n_3}{\theta} \\
&= -\frac{2\gamma n_2 n_3^2}{\theta} + \alpha n_2(n_3^2 - 1) \\
\frac{\partial r_{33}}{\partial d_3} &= -\alpha n_3 + n_3^2 \alpha n_3 + 2\gamma n_3 \frac{1 - n_3^2}{\theta} \\
&= \frac{2\gamma n_3(1 - n_3^2)}{\theta} + \alpha n_3(n_3^2 - 1)
\end{aligned}
$$

Step 4: Complete Jacobian Matrix

Therefore, the complete 9×3 Jacobian matrix $\frac{\partial \mathbf{r}}{\partial \mathbf{d}^T}$ is:

$$
\begin{bmatrix}
\frac{2\gamma n_1(1-n_1^2)}{\theta} + \alpha n_1(n_1^2-1) & -\frac{2\gamma n_1^2 n_2}{\theta} + \alpha n_2(n_1^2-1) & -\frac{2\gamma n_1^2 n_3}{\theta} + \alpha n_3(n_1^2-1) \\
n_1(n_1n_2 - n_3) + \frac{\gamma n_2(1-2n_1^2)}{\theta} & n_2(n_1n_2 - n_3) + \frac{\gamma n_1(1-2n_2^2)}{\theta} & n_3(n_1n_2 - n_3) + \frac{n_3^2 - 1 - 2n_1n_2n_3}{\theta} \\
n_1(n_1n_3 + n_2) + \frac{\gamma n_3(1-2n_1^2)}{\theta} & n_2(n_1n_3 + n_2) + \frac{1 - n_2^2 - 2n_1n_2n_3}{\theta} & n_3(n_1n_3 + n_2) + \frac{\gamma n_1(1-2n_3^2)}{\theta} \\
n_1(n_1n_2 + n_3) + \frac{\gamma n_2(1-2n_1^2)}{\theta} & n_2(n_1n_2 + n_3) + \frac{\gamma n_1(1-2n_2^2)}{\theta} & n_3(n_1n_2 + n_3) + \frac{1 - n_3^2 - 2n_1n_2n_3}{\theta} \\
-\frac{2\gamma n_1 n_2^2}{\theta} + \alpha n_1(n_2^2-1) & \frac{2\gamma n_2(1-n_2^2)}{\theta} + \alpha n_2(n_2^2-1) & -\frac{2\gamma n_2^2 n_3}{\theta} + \alpha n_3(n_2^2-1) \\
n_1(n_2n_3 - n_1) + \frac{-1 + n_1^2 - 2n_1n_2n_3}{\theta} & n_2(n_2n_3 - n_1) + \frac{\gamma n_3(1-2n_2^2)}{\theta} & n_3(n_2n_3 - n_1) + \frac{\gamma n_2(1-2n_3^2)}{\theta} \\
n_1(n_1n_3 - n_2) + \frac{\gamma n_3(1-2n_1^2)}{\theta} & n_2(n_1n_3 - n_2) + \frac{-1 + n_2^2 - 2n_1n_2n_3}{\theta} & n_3(n_1n_3 - n_2) + \frac{\gamma n_1(1-2n_3^2)}{\theta} \\
n_1(n_2n_3 + n_1) + \frac{1 - n_1^2 - 2n_1n_2n_3}{\theta} & n_2(n_2n_3 + n_1) + \frac{\gamma n_3(1-2n_2^2)}{\theta} & n_3(n_2n_3 + n_1) + \frac{\gamma n_2(1-2n_3^2)}{\theta} \\
-\frac{2\gamma n_1 n_3^2}{\theta} + \alpha n_1(n_3^2-1) & -\frac{2\gamma n_2 n_3^2}{\theta} + \alpha n_2(n_3^2-1) & \frac{2\gamma n_3(1-n_3^2)}{\theta} + \alpha n_3(n_3^2-1)
\end{bmatrix}
$$

Problem 4

Code Implementation

```
import cv2
import os
import numpy as np

# Input and output directories
input_dir = "C:\\Users\\33426\\Desktop\\cv\\picture"
output_dir = "C:\\Users\\33426\\Desktop\\cv\\result"

# Create the output directory if it doesn't exist
if not os.path.exists(output_dir):
os.makedirs(output_dir)
```

Problem 4 continued on next page...

```
# Chessboard parameters
pattern_size = (9, 6)

# Initialize variables
obj_points = []  # Store 3D points in the world coordinate system
img_points = []  # Store 2D points in the image plane

# Prepare 3D points of the chessboard
objp = np.zeros((pattern_size[0] * pattern_size[1], 3), np.float32)
objp[:, :2] = np.mgrid[0:pattern_size[0], 0:pattern_size[1]].T.reshape(-1, 2)

# First pass: Camera calibration
for img_name in os.listdir(input_dir):
input_image_path = os.path.join(input_dir, img_name)

# Read the input image
frame = cv2.imread(input_image_path)
if frame is None:
print(f"Unable to read the input image: {input_image_path}. Please check the path.")
continue

# Detect chessboard corners
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
ret, corners = cv2.findChessboardCorners(gray, pattern_size, None)

if ret:
# Refine corner positions
corners2 = cv2.cornerSubPix(gray, corners, (11, 11), (-1, -1),
(cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 100, 0.001))

# Add 3D and 2D points
obj_points.append(objp)
img_points.append(corners2)
else:
print(f"Chessboard corners not found in image: {input_image_path}")

# Camera calibration
if len(obj_points) > 0:
ret, camera_matrix, dist_coeffs, rvecs, tvecs = cv2.calibrateCamera(obj_points, img_points,
    gray.shape[::-1], None, None)

# Save camera parameters to txt file
param_file_path = os.path.join(output_dir, "camera_parameters.txt")
with open(param_file_path, 'w') as f:
f.write("Camera Matrix:\n")
f.write(str(camera_matrix))
f.write("\n\nDistortion Coefficients:\n")
f.write(str(dist_coeffs))

print("Camera parameters saved to:", param_file_path)

# Second pass: Bird's eye view transformation
for img_name in os.listdir(input_dir):
input_image_path = os.path.join(input_dir, img_name)

# Read the input image
frame = cv2.imread(input_image_path)
if frame is None:
continue

# Undistort the image
undistorted_image = cv2.undistort(frame, camera_matrix, dist_coeffs)

# Convert to grayscale and detect chessboard corners
```

```
gray = cv2.cvtColor(undistorted_image, cv2.COLOR_BGR2GRAY)
ret, corners = cv2.findChessboardCorners(gray, pattern_size, None)

if ret:
# Refine corner positions
corners2 = cv2.cornerSubPix(gray, corners, (11, 11), (-1, -1),
(cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 100, 0.001))

# Select points for perspective transformation
src_pts = np.float32([corners2[0], corners2[pattern_size[0] - 1],
corners2[-1], corners2[-pattern_size[0]]])

# Define destination points
h, w = undistorted_image.shape[:2]
dst_pts = np.float32([[702.0, 514.0],  # 左上
[1222.0, 514.0],  # 右上
[1222.0, 914.0],  # 右下
[702.0, 914.0]])  # 左下

# Compute and apply perspective transformation
M = cv2.getPerspectiveTransform(src_pts, dst_pts)
birdseye_image = cv2.warpPerspective(undistorted_image, M, (w, h))

# Save the result
output_image_path = os.path.join(output_dir, f"birdseye_{img_name}")
cv2.imwrite(output_image_path, birdseye_image)

print(f"Bird's eye view saved as: {output_image_path}")
else:
print(f"Chessboard corners not found in image: {input_image_path}")
else:
print("No valid images found for camera calibration")
```

Implementation Details

Key Parameters

- Chessboard pattern size: (9×6)
- Output scaling factor: 1.2
- Border padding: 10 pixels

Perspective Transform

The perspective transform is computed using four corner points:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{5}$$

where H is the homography matrix calculated from the corner correspondences.

Border Removal Process

1. Convert image to grayscale

2. Apply threshold: $T(x, y) = \begin{cases} 255 & \text{if } I(x, y) > 1 \\ 0 & \text{otherwise} \end{cases}$
3. Find largest contour
4. Crop image with padding: $(x - p, y - p, w + 2p, h + 2p)$

Results

The algorithm produces bird's eye view images with:

- Corrected perspective
- Removed distortion
- Minimal black borders
- Consistent scale