CoordinateTransforms

October 2024

Let's tackle the coordinate transformations by considering the drone's roll, pitch, yaw angles, and the camera's distortion parameters. We'll start by deriving the mathematical transformations and then implement them in code.

Mathematical Derivation

Transformation 1: Image Coordinates to World Coordinates

Objective: Given the drone's position and orientation, the image coordinates of the landing pad, and the camera parameters (including distortion), compute the world coordinates of the landing pad.

Inputs:

- **Drone Position**: $\mathbf{t}_{wd}$ (world coordinates)
- **Drone Orientation**: Roll (ϕ), Pitch (θ), Yaw (ψ)
- **Image Coordinates**: (u, v) in pixels
- **Camera Parameters**:
  - Intrinsic Matrix K
  - Distortion Coefficients $(k_1, k_2, p_1, p_2, k_3)$
- **Camera Mounting Parameters**:
  - Rotation R_{dc} (from camera to drone frame)
  - Translation $\mathbf{t}_{dc}$ (from camera to drone frame)
- **Assumed Ground Plane**: $z = z_0$ in world coordinates

Steps:

1. Undistort Image Coordinates

Normalize pixel coordinates to normalized image coordinates:

$$x_d = \frac{(u - c_x)}{f_x}$$

$$y_d = \frac{(v - c_y)}{f_y}$$

Correct for lens distortion using the distortion coefficients. This typically requires solving for the undistorted coordinates (x_u, y_u) from the distorted ones (x_d, y_d). This can be done iteratively or using OpenCV's `undistortPoints` function.

2. Compute the Direction of the Ray in Camera Coordinates

$$\mathbf{r}_c = \begin{bmatrix} x_u \\ y_u \\ 1 \end{bmatrix}$$

3. Transform the Ray to World Coordinates

Drone Rotation Matrix R_{wd}:

$$R_{wd} = R_z(\psi) \cdot R_y(\theta) \cdot R_x(\phi)$$

Camera Rotation Matrix R_{wc}:

$$R_{wc} = R_{wd} \cdot R_{dc}$$

Camera Position in World Coordinates $\mathbf{t}_{wc}$:

$$\mathbf{t}_{wc} = \mathbf{t}_{wd} + R_{wd} \cdot \mathbf{t}_{dc}$$

4. Compute the Intersection of the Ray with the Ground Plane

The ray in world coordinates is:

$$\mathbf{X}(s) = \mathbf{t}_{wc} + s \cdot (R_{wc} \cdot \mathbf{r}_c)$$

The equation of the ground plane $z = z_0$.
To find s, solve:

$$(\mathbf{t}_{wc,z} + s \cdot (R_{wc} \cdot \mathbf{r}_c)_z) = z_0$$

Solve for s:

$$s = \frac{z_0 - \mathbf{t}_{wc,z}}{(R_{wc} \cdot \mathbf{r}_c)_z}$$

Compute the world coordinates of the landing pad:

$$\mathbf{X} = \mathbf{t}_{wc} + s \cdot (R_{wc} \cdot \mathbf{r}_c)$$

—

Transformation 2: World Coordinates to Image Coordinates

Objective: Given the drone's position and orientation, and the landing pad's world coordinates, compute the image coordinates and sizes of the landing pad in the image.

Inputs:

- **Drone Position**: $\mathbf{t}_{wd}$
- **Drone Orientation**: ϕ, θ, ψ

- **Landing Pad Position**: $\mathbf{X}_{pad}$
- **Camera Parameters**:
  - Intrinsic Matrix K
  - Distortion Coefficients $(k_1, k_2, p_1, p_2, k_3)$
- **Camera Mounting Parameters**:
  - Rotation R_{dc}
  - Translation $\mathbf{t}_{dc}$

Steps:

1. Compute Camera Position and Orientation in World Coordinates

Same as in Transformation 1.

2. Transform Landing Pad Coordinates to Camera Frame

Compute the vector from the camera to the landing pad in world coordinates:

$$\mathbf{X}_{cw} = \mathbf{X}_{pad} - \mathbf{t}_{wc}$$

Transform to camera frame:

$$\mathbf{X}_c = R_{wc}^{\top} \cdot \mathbf{X}_{cw}$$

3. Project onto Image Plane

Normalized image coordinates:

$$x_u = \frac{X_{c,x}}{X_{c,z}}$$
$$y_u = \frac{X_{c,y}}{X_{c,z}}$$

4. Apply Lens Distortion

Compute $r^2 = x_u^2 + y_u^2$.

Apply distortion model:

$$x_d = x_u(1+k_1r^2+k_2r^4+k_3r^6)+2p_1x_uy_u+p_2(r^2+2x_u^2)$$
$$y_d = y_u(1+k_1r^2+k_2r^4+k_3r^6)+p_1(r^2+2y_u^2)+2p_2x_uy_u$$

5. Convert to Pixel Coordinates

$$u = f_x x_d + c_x$$
$$v = f_y y_d + c_y$$

6. Compute the Size of the Landing Pad in the Image

Assume the landing pad is a square with known size s_{pad} (e.g., 1m).

Compute the corners of the landing pad in the camera frame by considering small displacements δ_x and δ_y around $\mathbf{X}_{pad}$.

Transform these corners to image coordinates using steps 2-5.

Compute the bounding box size in pixels.