

Coordinate Extraction from Camera Geometry

Using Known Camera Height, Tilt, and Field of View

Introduction

This document presents a basic geometric approach for estimating the real-world coordinates of a feature on the ground plane, using a pinhole-like camera model and known camera parameters. We assume:

- The camera is at a height H above a flat ground plane.
- The camera is tilted downward by an angle θ (in degrees).
- The horizontal and vertical fields of view (FOV) are ϕ_h and ϕ_v (in degrees), respectively.
- The image has resolution $W \times H_{\text{img}}$ pixels (width \times height).
- A detected feature is located at pixel coordinates (u, v) .

Our goal is to compute the horizontal position X_{world} (left-right) and distance Y_{world} (forward) relative to the camera's origin on the ground.

1. Image-Centered Coordinates

By default, many computer vision libraries treat the top-left corner of an image as $(0, 0)$. For this geometric method, it is helpful to recast the image so its center is $(0, 0)$:

$$u_c = u - \frac{W}{2}, \quad v_c = v - \frac{H}{2}.$$

Therefore, u_c is positive if the feature is to the right of the center; v_c is positive if the feature is below the center.

2. Angular Offsets

Translate the normalized image coordinates to angular offsets from the optical axis:

$$\alpha = \left(\frac{u_c}{\frac{W}{2}} \right) \times \left(\frac{\phi_h}{2} \right),$$
$$\beta = \left(\frac{v_c}{\frac{H}{2}} \right) \times \left(\frac{\phi_v}{2} \right).$$

Here, α is the left-right angle, and β is the up-down angle, measured from the camera center. Because the camera is tilted by θ degrees down from the horizontal, the *total* pitch angle γ to the point on the ground is:

$$\gamma = \theta + \beta.$$

Note that α , β , and γ should be converted to radians for trigonometric functions if your implementation requires it.

3. Ground-Plane Distance

We assume a flat floor and use simple trigonometry to locate the feature in a 2D plane where the camera is the origin. Let H be the camera height above ground:

$$R = \frac{H}{\tan(\gamma)},$$

where R is the *forward distance* (parallel to the ground) from the camera's base (the point on the ground directly below the camera) to the feature's vertical projection. We can then compute the horizontal offset X using the side-view angle α :

$$X = R \tan(\alpha).$$

Hence, the final coordinates $(X_{\text{world}}, Y_{\text{world}})$ relative to the camera on the ground plane are:

$$\begin{cases} X_{\text{world}} = R \tan(\alpha), \\ Y_{\text{world}} = R. \end{cases}$$

In practice, X_{world} is positive if the feature is to the right; negative if it is to the left.

4. Example with Lab Parameters

For a lab setup using:

- $H = 85$ mm (camera height),
- $\theta = 25^\circ$ (downward tilt),
- $\phi_h = 102^\circ$ (horizontal FOV),
- $\phi_v = 67^\circ$ (vertical FOV),

and an image resolution of 640×480 :

1. The image center is at $(u_0, v_0) = (320, 240)$.
2. If a detected feature is at $(u, v) = (320, 240)$, then $u_c = 0$ and $v_c = 0$.
3. Hence, $\alpha = 0$, $\beta = 0$, and $\gamma = \theta = 25^\circ$.
4. Then,

$$R = \frac{85 \text{ mm}}{\tan(25^\circ)} \approx 182 \text{ mm}.$$

5. $X_{\text{world}} = R \tan(0) = 0$; so the object is directly in front of the camera, about 182 mm away.

5. Caveats and Refinements

- **Lens Distortion:** Real cameras, especially with wide FOV, may exhibit notable lens distortion. Using a proper camera calibration (e.g., via OpenCV's `calibrateCamera`) is recommended for more accurate distance estimation.
- **Exact Pinhole Model:** These equations treat the camera as an ideal pinhole system. In reality, focal lengths, sensor size, and principal point offsets can introduce small errors.
- **Steep Angles and Noise:** At large tilt angles or near the image edges, small pixel errors in v can greatly affect γ , resulting in larger distance errors.
- **Non-Flat Ground:** This derivation assumes a flat ground plane; irregular or sloped surfaces add complexity to the geometry.

Conclusion

These equations and assumptions provide a straightforward path for converting a 2D image coordinate of a feature to a 3D ground-plane coordinate relative to the camera. Understanding these relationships enables tasks such as robotic navigation, object avoidance, and line tracking based on camera feedback. For greater accuracy, combine these equations with detailed camera calibration and more sophisticated models where necessary.