Chapter Outline

Multimodal Perception Coordinate Transformations

Chapter 8 Multimodal Perception and Intelligent Decision-Making

8.1 Image Processing Fundamentals

8.2 Recognition, Tracking, Optical Flow

8.3 Localization, Mapping, and Planning

8.4 Common Perception Sensors

8.5 Coordinate Transformations

8.6 Fusion and Control Decision-Making

8.7 Tools and Framework Interfaces

8.8 Experiments and Real-World Transfer

8.9 Exercises and Classroom Discussion

4 Units in This Section

8.5.1 Pixel and Image Coordinates
Textbook: Sec. 8.1.5

8.5.2 From Image to Camera Coordinates
Textbook: Sec. 8.1.5

8.5.3 From Camera to Body and Ground Frames
Textbook: Sec. 8.1.5

8.5.4 Geographic Coordinates and a Complete Worked Example
Textbook: Sec. 8.1.5

8.5.1 Pixel and Image Coordinates: Transformation Chain

Five Coordinate Representations of One Target

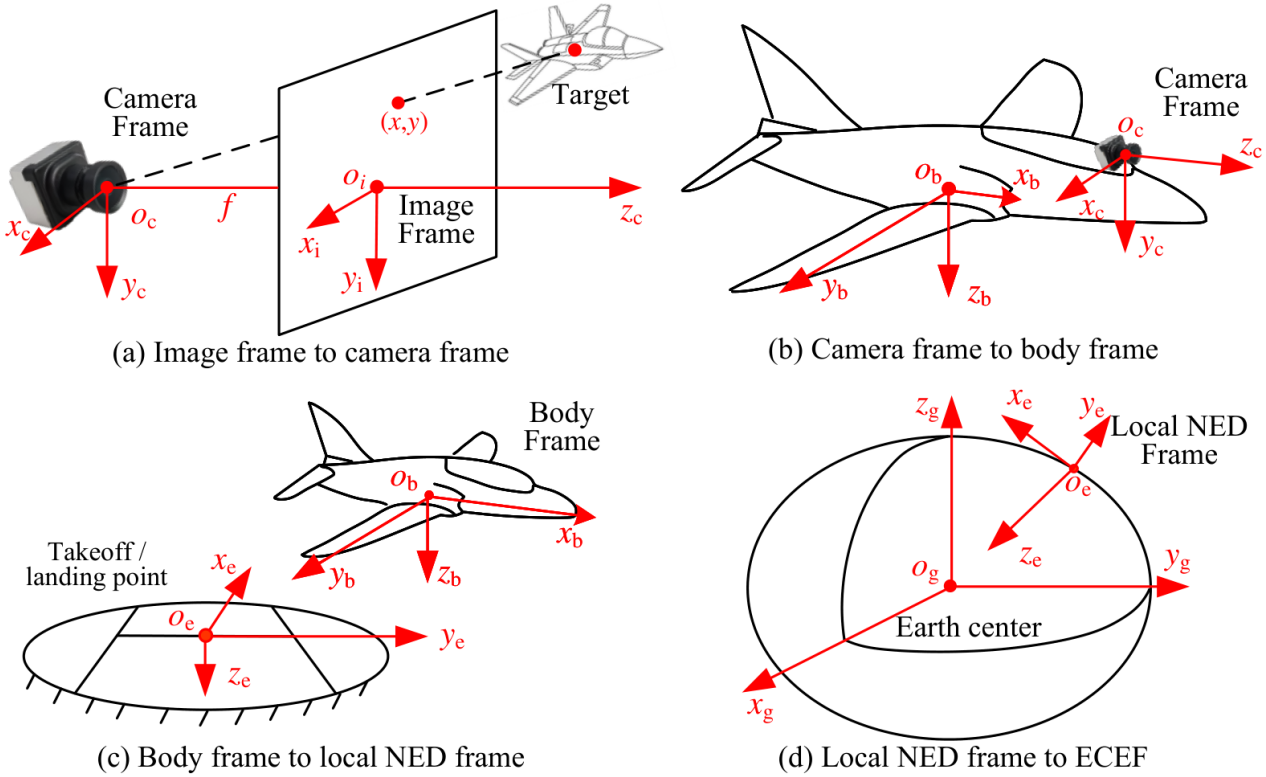


Figure 8.5 Coordinate Transformation Schematic

Pixels and camera: pixels define a sightline; depth supplies its scale.

Body and ground: extrinsics and vehicle pose place the point in the task scene.

Common geographic datum: vehicles and targets share an origin, coordinate convention, and time reference.

8.5.1 Pixel and Image Coordinates: Image and Camera

Pixel Localization and Camera Mounting Geometry

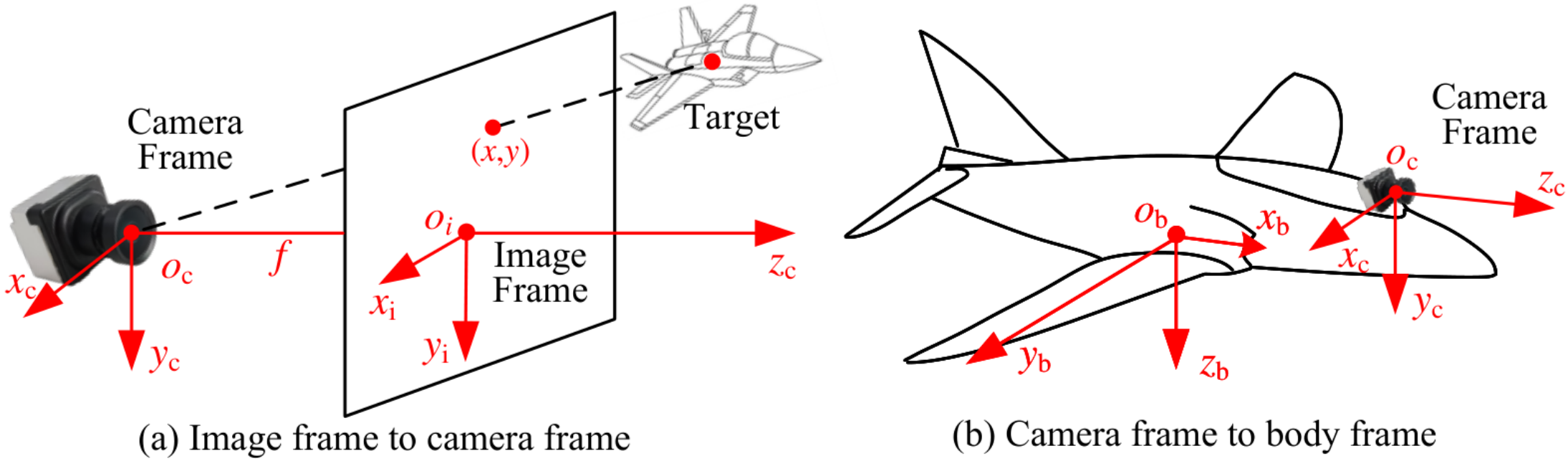


Figure 8.5 (upper half) Imaging geometry and sensor mounting

Left: pixels, intrinsics, and depth. Right: camera mounting position and orientation.

8.5.1 Pixel and Image Coordinates: Pixel Frame

Resolution Defines Data Size and Coordinate Limits

Origin and axes: top-left pixel origin; u right, v down; units: px.

Image Size	Zero-Based Integer Index Range	Pixel Count
640×480	$0 \le u \le 639,\ 0 \le v \le 479$	307 200
1280×720	$0 \le u \le 1279,\ 0 \le v \le 719$	921 600 (3×)

Frame notation: ${}^{i}\mathbf{p}$ image; ${}^{c}\mathbf{p}$ camera; ${}^{b}\mathbf{p}$ body; ${}^{e}\mathbf{p}$ ground; ${}^{g}\mathbf{p}$ geographic.

Resolution gives size; calibration gives the principal point. Target centers may be subpixel.

8.5.1 Pixel and Image Coordinates: Image Frame

Principal-Point Offsets Convert Pixels Into Physical Lengths

$[u_0, v_0]^{\mathrm{T}}$ is the principal point; d_x, d_y are pixel sizes. Here: zero skew, right/down positive.

$$x_{\mathrm{i}} = (u - u_0)d_x \tag{8.1}$$

$$y_{\mathrm{i}} = (v - v_0)d_y$$

Pixel (480,320), principal point (320,240), and $d_x = d_y = 0.01$ mm/px.

$$x_{\mathrm{i}} = 160(0.01) = 1.6\,\mathrm{mm}, \quad y_{\mathrm{i}} = 80(0.01) = 0.8\,\mathrm{mm}$$

Both components are positive: the target projects right and below the principal point.

8.5.1 Pixel and Image Coordinates: Camera Intrinsics

Intrinsics Link Physical and Pixel Focal Lengths

$$[x_{\mathrm{i}}, y_{\mathrm{i}}, f]^{\mathsf{T}} = f\mathbf{K}^{-1}[u, v, 1]^{\mathsf{T}} \tag{8.2}$$

$$\mathbf{K} = \begin{bmatrix} f_x & s & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{8.3}$$

$$f_x = f/d_x, \quad f_y = f/d_y$$

Use the same length unit for f and d_x, d_y. Focal lengths f_x, f_y are in px; s is pixel-axis skew, set to 0 here.

8.5.1 Pixel and Image Coordinates: Resolution Changes

Scale Intrinsics When Resizing an Image

Item	640×480 Original	320×240 Resized
Target pixel	(480,320)	(240,160)
Principal point	(320,240)	(160,120)
Pixel focal length	320px	160px
Normalized ray	$[0.5,0.25,1]^{\mathsf{T}}$	$[0.5,0.25,1]^{\mathsf{T}}$

$$\mathbf{q} = \mathbf{K}^{-1}[u, v, 1]^{\mathsf{T}}$$

Keeping the old intrinsics gives $[-0.25,-0.25,1]^{\mathsf{T}}$: the direction changes.

8.5.1 Pixel and Image Coordinates: Calibration and Distortion

Multiple Views Estimate Real-Camera Parameters

Parameter / Tool	Inputs and Outputs
Calibration board and multiple views	Known geometry and image corners yield intrinsics and distortion.
Radial / tangential distortion	Radial barrel/pincushion offsets; tangential offsets from mounting misalignment.
Camera Calibrator OpenCV / Kalibr	Visual calibration, function-based calibration, and camera–inertial calibration.

Companion reading: 2.CameraCalcDemo / Readme.pdf
Acquire: OneCameraCal.py; calibrate offline: bord-Calibration.py.

A calibrated ray still needs a depth constraint.

Chapter Outline

Multimodal Perception Coordinate Transformations

Chapter 8 Multimodal Perception and Intelligent Decision-Making

8.1 Image Processing Fundamentals

8.2 Recognition, Tracking, Optical Flow

8.3 Localization, Mapping, and Planning

8.4 Common Perception Sensors

8.5 Coordinate Transformations

8.6 Fusion and Control Decision-Making

8.7 Tools and Framework Interfaces

8.8 Experiments and Real-World Transfer

8.9 Exercises and Classroom Discussion

4 Units in This Section

8.5.1 Pixel and Image Coordinates
Textbook: Sec. 8.1.5

8.5.2 From Image to Camera Coordinates
Textbook: Sec. 8.1.5

8.5.3 From Camera to Body and Ground Frames
Textbook: Sec. 8.1.5

8.5.4 Geographic Coordinates and a Complete Worked Example
Textbook: Sec. 8.1.5

8.5.2 From Image to Camera Coordinates: Camera Frame

A Pinhole Camera Projects 3D Points Onto the Image Plane

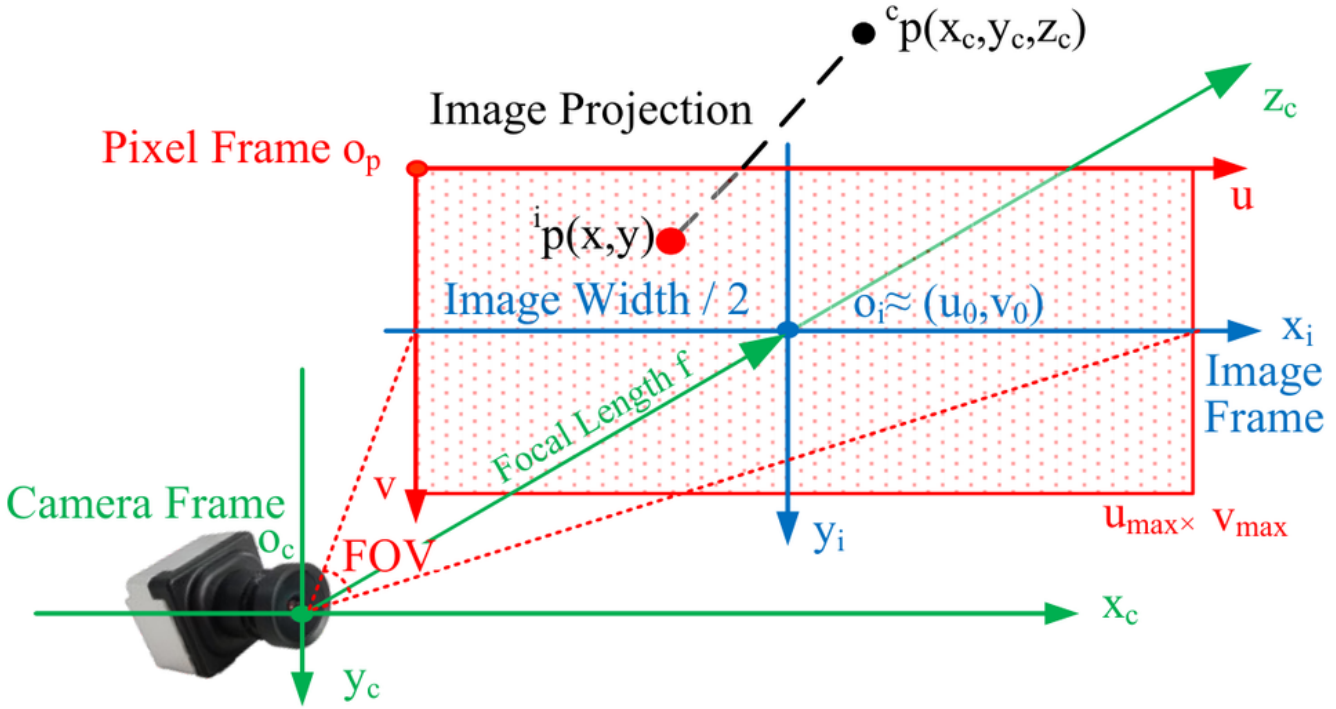


Figure 8.6 Pixel Frame to Camera Frame Transformation

Optical-center origin: x_c right, y_c down, z_c forward along the optical axis.

Focal length: optical center to image plane. Depth: target component along the optical axis.

Assumptions: corrected pixels, square orthogonal pixels, zero skew.

8.5.2 From Image to Camera Coordinates: Field of View

Horizontal Field of View Sets Pixel Focal Length

Ideal undistorted projection, square pixels; field of view (FOV) is horizontal here.

$$\frac{\text{FOV}}{2} = \arctan\left(\frac{d_x u_{max}/2}{f}\right) \tag{8.4}$$

$$f_x = f_y = \frac{f}{d_x} = \frac{u_{max}}{2\tan(\text{FOV}/2)} \tag{8.5}$$

$$u_{max} = 640 \text{ px}, \quad \text{FOV} = 90^\circ \implies f_x = f_y = 320 \text{ px}$$

$90^\circ/2 = 45^\circ$ and $\tan 45^\circ = 1$. A 640 px width gives a 320 px focal length.

8.5.2 From Image to Camera Coordinates: Ideal Camera Intrinsics

Complete Intrinsics for a 640×480 Camera With 90° FOV

$$\mathbf{K} = \begin{bmatrix} \frac{u_{max}}{2\tan(\mathrm{FOV}/2)} & 0 & u_{max}/2 \\ 0 & \frac{u_{max}}{2\tan(\mathrm{FOV}/2)} & v_{max}/2 \\ 0 & 0 & 1 \end{bmatrix} \tag{8.6}$$

$$\mathbf{K} = \begin{bmatrix} 320 & 0 & 320 \\ 0 & 320 & 240 \\ 0 & 0 & 1 \end{bmatrix}$$

Pixel focal length: 320 px; principal point: (320,240). Calibrate real-camera principal points.

$$\mathrm{FOV}_y = 2\arctan(240/320) \approx 73.74^\circ$$

8.5.2 From Image to Camera Coordinates: Back-Projection

Depth Sets the Scale of a Camera Ray

$$[x_c, y_c, z_c]^\top = z_c \mathbf{K}^{-1} [u, v, 1]^\top \tag{8.7}$$

Inputs: corrected pixel (480,320), intrinsics $\mathbf{K} \in \mathbb{R}^{3\times3}$, and axial depth $z_c = 4$ m.

$$\mathbf{q} = [0.5, 0.25, 1]^\top, \qquad {}^{c}\mathbf{p} = 4\mathbf{q} = [2,1,4]^\top \text{ m}$$

Camera point: 2 m right, 1 m down, 4 m forward. It is a 3×1 vector.

8.5.2 From Image to Camera Coordinates: Slant Range and Axial Depth

Slant Range Follows the Sightline; Depth Follows the Optical Axis

$$\mathbf{q} = \mathbf{K}^{-1}[u, v, 1]^{\mathsf{T}}, \qquad {}^{c}\mathbf{p} = r\frac{\mathbf{q}}{\|\mathbf{q}\|}$$

$$z_c = \frac{r}{\|\mathbf{q}\|}, \quad \|\mathbf{q}\| = \sqrt{1.3125} \approx 1.14564$$

Measurement Handling	Axial Depth / m	Recovered Camera Point / m
Correct: convert $r = \sqrt{21}$ m	4.0000	(2.0000,1.0000,4.0000)
Incorrect: use range as depth	4.5826	(2.2913,1.1456,4.5826)

Slant range equals axial depth only when the target lies on the optical axis.

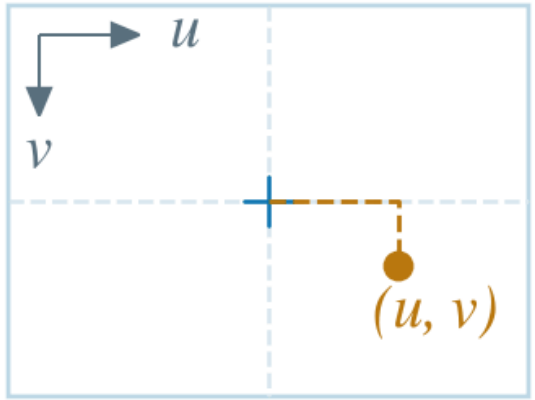
8.5.2 From Image to Camera Coordinates: Projection and Back-Projection

A Pixel Defines a Ray; Depth Selects a Spatial Point

Diagram notation: $c_x = u_0$, $X = x_c$, $Z = z_c$, $\mathbf{P} = {}^{c}\mathbf{p}$

One pixel, many points

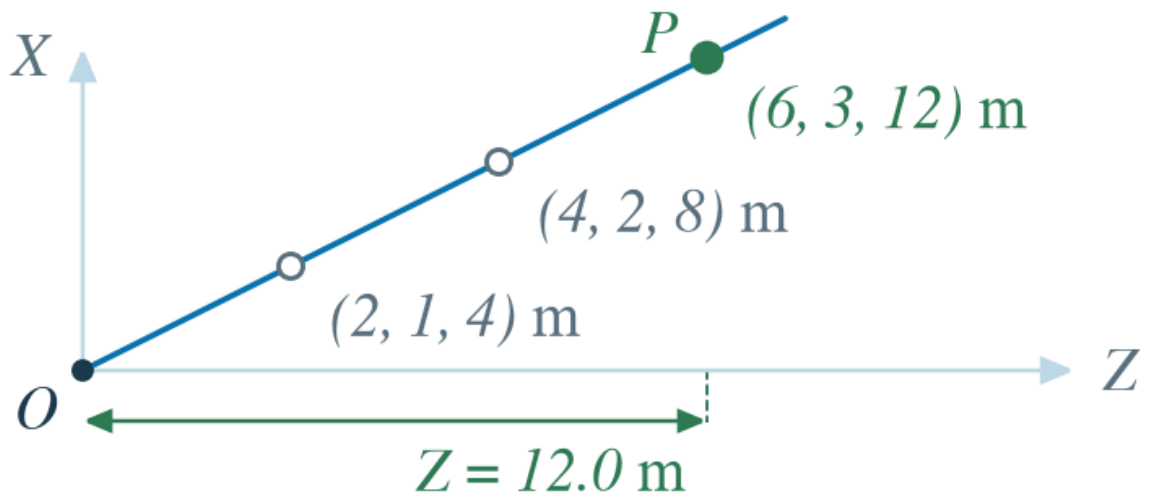
640×480



$(u, v) = (480, 320)$

Depth selects a point

X-Z view (Y omitted)



$f_x = f_y = 320$ px $\quad c_x = 320, \; c_y = 240$

Figure: one pixel defines a spatial ray; axial depth selects a point on it.

8.5.2 From Image to Camera Coordinates: Back-Projection Check

Forward/Back-Projection Consistency Checks Units and Scale

$$u = f_x x_c/z_c + u_0, \quad v = f_y y_c/z_c + v_0$$

Axial Depth / m	Back-Projected 3D Point / m	Reprojection / px
4	(2,1,4)	(480,320)
8	(4,2,8)	(480,320)
12	(6,3,12)	(480,320)

Same-pixel round trip checks geometry; depth needs range, stereo, or known-size constraints.

Geometric round trips can reveal direction and scale errors.

Chapter Outline

Multimodal Perception Coordinate Transformations

Chapter 8 Multimodal Perception and Intelligent Decision-Making

8.1 Image Processing Fundamentals

8.2 Recognition, Tracking, Optical Flow

8.3 Localization, Mapping, and Planning

8.4 Common Perception Sensors

8.5 Coordinate Transformations

8.6 Fusion and Control Decision-Making

8.7 Tools and Framework Interfaces

8.8 Experiments and Real-World Transfer

8.9 Exercises and Classroom Discussion

4 Units in This Section

8.5.1 Pixel and Image Coordinates
Textbook: Sec. 8.1.5

8.5.2 From Image to Camera Coordinates
Textbook: Sec. 8.1.5

8.5.3 From Camera to Body and Ground Frames
Textbook: Sec. 8.1.5

8.5.4 Geographic Coordinates and a Complete Worked Example
Textbook: Sec. 8.1.5

8.5.3 From Camera to Body and Ground Frames: Body Frame

Camera-to-Body Axis Permutation

Convention	Positive Axis Directions	Choice in This Example
Camera	x right, y down, z forward	Origin: optical center
Body: FRD	x/y/z: forward/right/down (FRD)	Origin: body center
Body: FLU	x/y/z: forward/left/up (FLU)	Negate both FRD y and z

$$\mathbf{R}_c^b = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{R}_c^b[2,1,4]^\mathsf{T} = [4,2,1]^\mathsf{T}$$

Optical axis along nose; no relative roll. Recompute the matrix if mounting rotation changes.

8.5.3 From Camera to Body and Ground Frames: Camera Extrinsics

Rotation and Mounting Offset Determine the Body-Frame Point

$$ {}^{b}\mathbf{p} = \mathbf{R}_{c}^{b} \quad {}^{c}\mathbf{p} + \quad {}^{b}\mathbf{t}_{c} \tag{8.8} $$

$$ {}^{b}\tilde{\mathbf{p}} = \mathbf{T}_{c}^{b} \quad {}^{c}\tilde{\mathbf{p}} $$

${}^{b}\mathbf{t}_{c} = [0.2,0.1,0.05]^{\mathsf{T}}$ m is the camera optical-center offset from the body origin.

$$ {}^{b}\mathbf{p} = [4,2,1]^{\mathsf{T}} + [0.2,0.1,0.05]^{\mathsf{T}} = [4.2,2.1,1.05]^{\mathsf{T}} \text{ m} $$

Rotation is dimensionless; translation and point coordinates use meters. Direction: camera to body.

8.5.3 From Camera to Body and Ground Frames: Homogeneous Transform

Homogeneous Points Combine Rotation and Translation

$$\mathbf{T}_c^b = \begin{bmatrix} \mathbf{R}_c^b & {}^b\mathbf{t}_c \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \tag{8.9}$$

$${}^c\tilde{\mathbf{p}} = [\, z_c(\mathbf{K}^{-1})^{\mathsf{T}} \quad \mathbf{e}_3 \,]^{\mathsf{T}}[u, v, 1]^{\mathsf{T}}, \quad \mathbf{e}_3 = [0,0,1]^{\mathsf{T}}$$

$$\mathbf{T}_c^b = \begin{bmatrix} 0 & 0 & 1 & 0.2 \\ 1 & 0 & 0 & 0.1 \\ 0 & 1 & 0 & 0.05 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Append 1 to a 3D point. Rigid transform: 4×4; pixel-to-point lifting matrix: 4×3.

8.5.3 From Camera to Body and Ground Frames: Extrinsic Direction

Invert a Known Body-to-Camera Transform Before Using It

$$\mathbf{T}_c^b = (\mathbf{T}_b^c)^{-1} = \begin{bmatrix} (\mathbf{R}_b^c)^\mathsf{T} & -(\mathbf{R}_b^c)^\mathsf{T} & {}^c\mathbf{t}_b \\ \mathbf{0}_{1\times 3} & & 1 \end{bmatrix}$$

Invert to obtain camera-to-body extrinsics; then recover the original camera point as a check.

$${}^c\mathbf{p} = (\mathbf{R}_c^b)^\mathsf{T}({}^b\mathbf{p} - {}^b\mathbf{t}_c) = [2,1,4]^\mathsf{T}\ \mathrm{m}$$

Calculation	Body Point / m
Correct camera-to-body transform	(4.2,2.1,1.05)
Wrong: transpose rotation, add old offset	(1.2,4.1,2.05)

For OpenCV calibration rotation/translation, check the source and destination frames first.

8.5.3 From Camera to Body and Ground Frames: Ground Frame

Local NED and ENU Describe the Same Physical Position

Local origin: takeoff point, base station, or mission center. A flat local model has limited spatial scope.

Coordinate Convention	Positive Axes	Vertical Position
North–East–Down (NED)	North, east, down	Negative Z: above origin
East–North–Up (ENU)	East, north, up	Positive Z: above origin

$$\mathbf{p}_{\mathrm{ENU}} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \mathbf{p}_{\mathrm{NED}}$$

8.5.3 From Camera to Body and Ground Frames: Body to Ground

A 90° Yaw Rotates the Nose Toward East

$$ {}^{e}\mathbf{p} = \mathbf{R}_b^e \ {}^{b}\mathbf{p} + \ {}^{e}\mathbf{t}_b \iff {}^{e}\tilde{\mathbf{p}} = \mathbf{T}_b^e \ {}^{b}\tilde{\mathbf{p}} \tag{8.10} $$

$$ \mathbf{R}_b^e = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad {}^{e}\mathbf{t}_b = [10{,}20, -30]^{\mathrm{T}}\ \mathrm{m} $$

$$ {}^{e}\mathbf{p} = [-2.1{,}4.2{,}1.05]^{\mathrm{T}} + [10{,}20, -30]^{\mathrm{T}} = [7.9{,}24.2, -28.95]^{\mathrm{T}}\ \mathrm{m} $$

Level vehicle, yaw 90°: target is 7.9 m north, 24.2 m east, and 28.95 m above the origin.

Chapter Outline

Multimodal Perception Coordinate Transformations

Chapter 8 Multimodal Perception and Intelligent Decision-Making

8.1 Image Processing Fundamentals

8.2 Recognition, Tracking, Optical Flow

8.3 Localization, Mapping, and Planning

8.4 Common Perception Sensors

8.5 Coordinate Transformations

8.6 Fusion and Control Decision-Making

8.7 Tools and Framework Interfaces

8.8 Experiments and Real-World Transfer

8.9 Exercises and Classroom Discussion

4 Units in This Section

8.5.1 Pixel and Image Coordinates
Textbook: Sec. 8.1.5

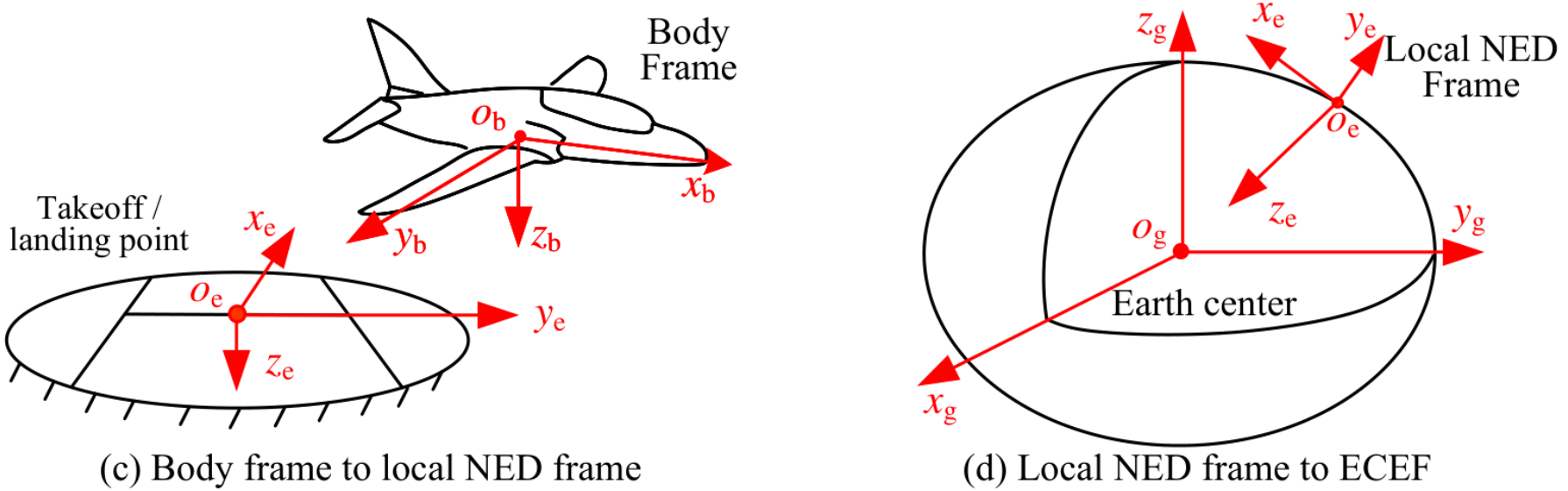
8.5.2 From Image to Camera Coordinates
Textbook: Sec. 8.1.5

8.5.3 From Camera to Body and Ground Frames
Textbook: Sec. 8.1.5

8.5.4 Geographic Coordinates and a Complete Worked Example
Textbook: Sec. 8.1.5

8.5.4 Geographic Coordinates and a Complete Worked Example: Geographic Coordinates

LLA and ECEF Represent Position Differently



(c) Body frame to local NED frame

(d) Local NED frame to ECEF

Figure 8.5 (lower half) Local ground frame and Earth reference

Latitude–longitude–altitude (LLA): latitude ±90°, longitude ±180°, ellipsoidal height in m. Earth-centered, Earth-fixed (ECEF): all axes in m. Ellipsoidal and orthometric heights use different datums.

8.5.4 Geographic Coordinates and a Complete Worked Example: Local to Geographic Coordinates

NED Conversion Uses a WGS84 Reference Point

$$ {}^{g}\mathbf{p}_{\mathrm{LLA}} = ned2lla(\ \ {}^{e}\mathbf{p}_{\mathrm{NED}}, \mathrm{origin}_{\mathrm{LLA}}) \tag{8.11}$$

WGS84 (World Geodetic System 1984) origin: $\mathrm{origin}_{\mathrm{LLA}} = [30°, 114°, 50\ \mathrm{m}]^{\mathsf{T}}$. Height is ellipsoidal.

$$\tilde{\mathbf{p}} = [n, e, d, w]^{\mathsf{T}} \implies \mathbf{p} = [n/w, e/w, d/w]^{\mathsf{T}}$$

Here w=1; pass the first three components. The installed software development kit (SDK) does not check or normalize general w. Native MATLAB also requires flat/ellipsoid.

8.5.4 Geographic Coordinates and a Complete Worked Example: Near-Field Geodetic Calculation

Check Latitude, Longitude, and Height Separately

At latitude 30°, WGS84 meridional radius $M_0 \approx 6351377.104$ m; prime-vertical radius $N_0 \approx 6383480.918$ m.

$$\phi \approx \phi_0 + \frac{n}{M_0 + h_0}, \quad \lambda \approx \lambda_0 + \frac{e}{(N_0 + h_0)\cos\phi_0}, \quad h \approx h_0 - d$$

Angular increments are in radians; convert output latitude/longitude to degrees. Input NED=(7.9,24.2,-28.95) m.

Calculation	Latitude / °	Longitude / °	Ellipsoidal Height / m
First-order local	30.000071265	114.000250811	78.950000
Exact via ECEF	30.000071265	114.000250810	78.950051

8.5.4 Geographic Coordinates and a Complete Worked Example: Overall Transformation

Matrix Order and Dimensions in the Complete Chain

$$ {}^{g}\mathbf{p}_{\mathrm{LLA}} = ned2lla(\pi_h(\mathbf{T}_b^e \mathbf{T}_c^b \mathbf{B}_c [u, v, 1]^{\mathsf{T}}), \quad \mathrm{origin}_{\mathrm{LLA}}) \tag{8.12} $$

$$ \mathbf{B}_c = [\, z_c(\mathbf{K}^{-1})^{\mathsf{T}} \quad \mathbf{e}_3 \,]^{\mathsf{T}} \in \mathbb{R}^{4\times 3}, \quad \mathbf{e}_3 = [0,0,1]^{\mathsf{T}} $$

$$ \pi_h([n, e, d, w]^{\mathsf{T}}) = [n/w, e/w, d/w]^{\mathsf{T}} \in \mathbb{R}^3, \quad w = 1 $$

$$ (4\times 4)(4\times 4)(4\times 3)(3\times 1) = (4\times 1) \overset{\pi_h}{\rightarrow} (3\times 1) $$

Inputs: corrected pixel, intrinsics, depth, mounting extrinsics, vehicle pose at the observation time, and shared geographic origin.

8.5.4 Geographic Coordinates and a Complete Worked Example: Complete Worked Example

One Pixel Produces a Checkable Global Position

Stage	Input / Operation	Result
Pixel to camera	Pixel (480,320), Z=4 m 640×480, horizontal FOV 90°	(2,1,4)m
Camera to body	Permute axes; mounting offset (0.2,0.1,0.05) m	(4.2,2.1,1.05)m
Body to NED	Yaw 90°, position (10,20,-30) m	(7.9,24.2,-28.95)m
NED to LLA	Origin (30°,114°,50 m)	(30.000071265°, 114.000250810°,78.950051m)

LLA → NED → inverse rigid transforms → projection returns pixel (480,320).

8.5.4 Geographic Coordinates and a Complete Worked Example: Error Analysis

Angle, Depth, and Timing Errors Propagate Along the Chain

Error Source	First-Order Calculation	Effect in This Example
Pixel error: 1 px	$z_c/f_x = 4/320$	Lateral: 0.0125 m
Depth error: 0.2 m	$0.2[0.5,0.25,1]^{\mathrm{T}}$	(0.10,0.05,0.20)m
Yaw error: 1°	$\sqrt{4.2^2 + 2.1^2}\,\pi/180$	Lateral: about 0.082 m
Time offset: 50 ms	10 m/s × 0.05 s	Position: about 0.5 m

Intrinsics, extrinsics, mounting, attitude, satellite-navigation origin, geographic model, and registration affect global position.

Share uncertainty and time references along with coordinates.

8.5.4 Geographic Coordinates and a Complete Worked Example: Error Analysis and Fusion

Shared Observations Retain Time and Uncertainty

Image / point cloud observations → **Common coordinates and time** → **Position, target ID, uncertainty**

Shared fields: geographic origin and height datum, observation time, target ID, position covariance, and source ID.

Fusion Method	Role in the Coordinate Chain
Extended Kalman filter (EKF) / Bayesian filtering	Recursively correct state using motion predictions and new observations.
Factor-graph optimization	Combine geometric constraints across times, sensors, or vehicles.
Deep fusion network	Learn multimodal feature relationships; check data distribution and confidence calibration.

Exercises and Discussion

Recalculate All Local Coordinates at a New Depth

Given: 640×480, horizontal FOV 90°, pixel (480,320); change axial depth to 8 m.

$$\mathbf{R}_c^b = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \qquad {}^b\mathbf{t}_c = [0.2, 0.1, 0.05]^{\mathsf{T}}\ \mathrm{m}$$

Level vehicle, yaw 90°, local position (10,20,-30) m; geographic origin (30°,114°,50 m).

1. Find camera, body, and local NED points; label their directions.
2. Decide whether mounting offset, slant-range handling, or LLA order should change.

Suggested Answers

Coordinate Results for One Pixel at Twice the Depth

Step	Reference Result	Direction Interpretation
Camera point	(4,2,8)m	4 m right, 2 m down, 8 m forward
Body point	(8.2,4.1,2.05)m	8.2 m forward, 4.1 m right, 2.05 m down
Local NED point	(5.9,28.2,-27.95)m	5.9 m north, 28.2 m east, 27.95 m up

Translations stay fixed: mounting and vehicle position do not double with target depth.
Range / order: $r = \sqrt{84}$ m; LLA is latitude, longitude, ellipsoidal height.

Section Summary

Geometry, Scale, Pose, and Datum Together Determine Position

Geometry and scale: pixels and intrinsics define sightlines; depth or other geometric constraints define 3D points.

Rigid pose: mounting extrinsics handle camera to body; body pose handles body to ground.

Common reference: local origin, NED/ENU, WGS84 ellipsoidal height, and observation time define position.

Common errors: range as depth, mixed units, reversed extrinsics, direct LLA addition, and stale pose.