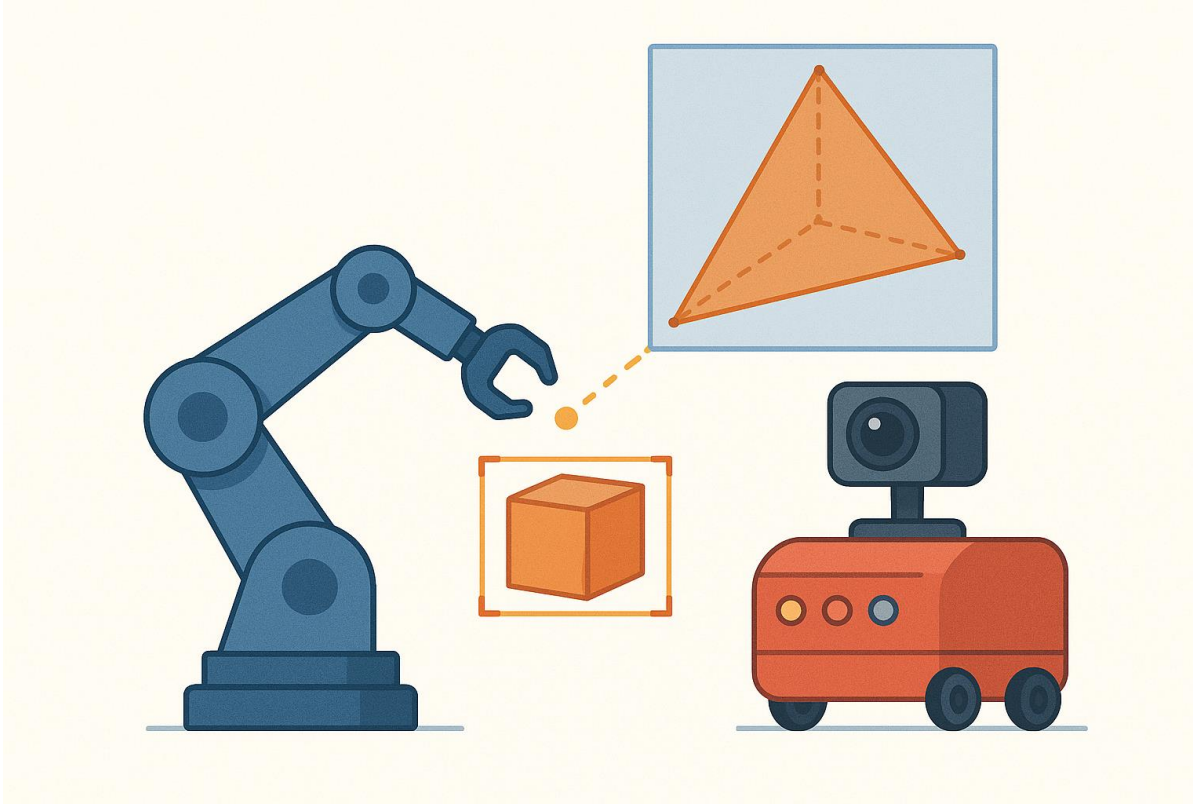
CSIC6843 – Lecture Unit 5

Computer Vision for Robotics



Object Detection, Feature Extraction, and Perception Strategies for Autonomous Systems

Table of Contents

List of Tables

List of Figures

Unit 05: Computer Vision for Robotics: Object Detection & Feature Extraction

Introduction to Robotic Vision

Robotic vision is a cornerstone technology enabling autonomous machines to perceive, interpret, and interact with the physical world. For humans, visual understanding feels instantaneous—we effortlessly transform light patterns into an awareness of objects, their positions, and their relationships. Robots, however, must achieve this through deliberate computational processes, starting from raw sensor data and progressing to structured, actionable information.

In robotics, vision serves two complementary purposes:

1. **Semantic Understanding** — Identifying *what* is present in a scene and classifying it into meaningful categories.

   Example: Detecting that a red cube is present and recognising it as a manipulable object.

2. **Geometric Reasoning** — Understanding *where* objects are in space and how they are oriented or moving.

   Example: Estimating the cube's position, size, and motion trajectory so it can be safely grasped.

These capabilities are central to a wide range of robotic tasks—from autonomous navigation and object manipulation to safe human–robot collaboration. However, achieving them in real-world environments introduces unique challenges:

- **Real-Time Performance** — Processing visual data quickly enough to guide immediate decision-making, often on resource-limited embedded systems.
- **Environmental Variability** — Handling changes in lighting, weather, occlusions, and clutter.
- **Sensor Imperfections** — Overcoming noise, calibration errors, and incomplete data.
- **Safety-Critical Reliability** — Ensuring that perception-driven actions are consistent, accurate, and safe.

This unit focuses on two foundational pillars of robotic perception—**object detection** and **feature extraction**—and examines how they integrate into broader robotic vision systems. You will explore sensing modalities, geometric principles, feature representation, scene understanding, tracking, sensor fusion, and system-level considerations such as robustness, real-time performance, and safety. By the end, you will understand not only how robots "see" but how they transform vision into informed, reliable actions in complex and dynamic environments.

5.1. Sensing Modalities in Robotic Vision

Robots perceive the world through a variety of sensors, each offering a different "slice" of reality. No single sensor can capture *everything* a robot needs to know—so combining multiple sensing modalities often results in more reliable and complete perception.

Monocular RGB Cameras

- **What they do:** Capture standard colour images with rich detail and texture.
- **Advantages:** Low cost, lightweight, and widely available.
- **Limitations:** Cannot directly measure depth; robots must infer spatial layout by analysing motion over time (structure from motion), comparing multiple viewpoints, or using prior scene knowledge.

Example use: A warehouse robot identifying labels and reading barcodes on packages.

Stereo Cameras

- **What they do:** Use two horizontally spaced lenses to mimic human binocular vision. Depth is estimated by measuring *disparity*—how far an object's image shifts between the left and right views.
- **Advantages:** Can measure depth without extra active sensors.
- **Limitations:** Works best with textured surfaces and in well-lit conditions; requires precise calibration.

Example use: A self-driving car gauging the distance to other vehicles on the road.

RGB-D Sensors

- **What they do:** Combine a colour camera (RGB) with active depth sensing (D) using techniques like structured light or time-of-flight.
- **Advantages:** Produce dense depth maps along with colour information, making object detection and manipulation easier.
- **Limitations:** Struggle outdoors in direct sunlight and with shiny or transparent surfaces.

Example use: A service robot mapping a living room for cleaning or object retrieval.

Event Cameras

- **What they do:** Instead of capturing full images at fixed intervals, they record changes in brightness *as they happen* at each pixel.
- **Advantages:** Extremely high temporal resolution (microsecond-level) and wide dynamic range; excel in high-speed motion and tricky lighting.
- **Limitations:** Unconventional data format requires specialised processing algorithms.

Example use: A drone avoiding collisions during high-speed flight through a forest.

Additional Modalities

- **LiDAR (Light Detection and Ranging):** Emits laser pulses to create accurate 3D maps—excellent for mapping and obstacle avoidance but can be bulky and costly.
- **Radar:** Uses radio waves to detect objects, works well in rain, fog, or dust, though with lower resolution.
- **IMU (Inertial Measurement Unit):** Measures acceleration and rotation at high frequency, aiding motion tracking when visual data is unreliable.

Robotic Vision Sensors at a Glance

Sensor Type	Main Strengths	Main Limitations	Typical Use Cases
Monocular RGB Camera	Low cost, light weight, rich colour/texture detail	No direct depth measurement: depth must be inferred	Object recognition, barcode reading, visual inspection
Stereo Camera	Provides depth from passive vision, similar to human eyes	Needs good texture, sensitive to lighting; precise calibration required	Autonomous driving, robotic grasping
RGB-D Sensor	Colour + dense depth maps; good for close-range tasks	Poor performance in sunlight; issues with shiny/transparent objects	Indoor navigation, service robots, object manipulation
Event Camera	Ultra-high temporal resolution; works well in fast motion or high contrast scenes	Unconventional data; requires special algorithms	High-speed drones, industrial inspection of moving parts
LiDAR	Highly accurate 3D mapping over long distances	Expensive, bulkier, may struggle with transparent/absorptive materials	Autonomous vehicles, outdoor mapping
Radar	Works in fog, rain, dust; long range	Lower spatial resolution	All-weather obstacle detection, vehicle safety systems
IMU	High-frequency motion sensing, independent of vision	Drift over time without visual correction	Stabilization in drones, motion estimation in dark or GPS-denied areas

Table 1 - Sensing Modalities in Robotic Vision Comparison

Choosing the Right Combination

The ideal sensor setup depends on the robot's task and environment:

- **Aerial drones** benefit from lightweight, fast-updating sensors like monocular or event cameras.
- **Factory robots** often rely on precise, short-range sensing like RGB-D cameras.
- **Autonomous vehicles** typically combine LiDAR, cameras, radar, and IMUs for redundancy and safety.

By thoughtfully combining these sensing modalities, robots can move beyond partial snapshots of their surroundings to a more complete and reliable understanding of the world.

5.2. Fundamentals of Camera Geometry and Multi-View Perception

For robots to see and understand the world, they must bridge the gap between **3D reality** and **2D images**. This involves understanding how scenes are projected onto an image and how multiple viewpoints can be combined to recover depth and spatial relationships.

Basic Camera Models
The **pinhole camera model** is the simplest way to describe image formation. Imagine a tiny hole letting light from a 3D point pass through and land on a flat surface (the image plane). This projection turns real-world coordinates into pixel coordinates.

- **Why it matters:** This model helps us reason about how objects move in an image when the camera or the object changes position.

Example: Predicting where a ball will appear in the frame as a robot moves toward it.

Multi-View Geometry
When multiple cameras—or the same camera from different positions—observe the same scene, the points they capture are linked by **geometric constraints**. By finding these matching points between images (known as **correspondences**), a robot can:

- **Reconstruct 3D positions** of objects.
- **Estimate its own motion** through the environment.

Example: A mobile robot uses two sequential images to figure out how far it has moved and how the environment has changed.

Depth Perception
Depth is the missing "third dimension" in a single image, but it's essential for safe navigation and precise manipulation. Robots estimate depth using:

- **Stereo triangulation** – Calculating distances by comparing disparities (shifts) in the same object's position across two images.
- **Active depth sensing** – Using RGB-D cameras or LiDAR to directly measure distance to surfaces.

Example: A robotic arm gauging how far to reach to grab an object without colliding with nearby obstacles.

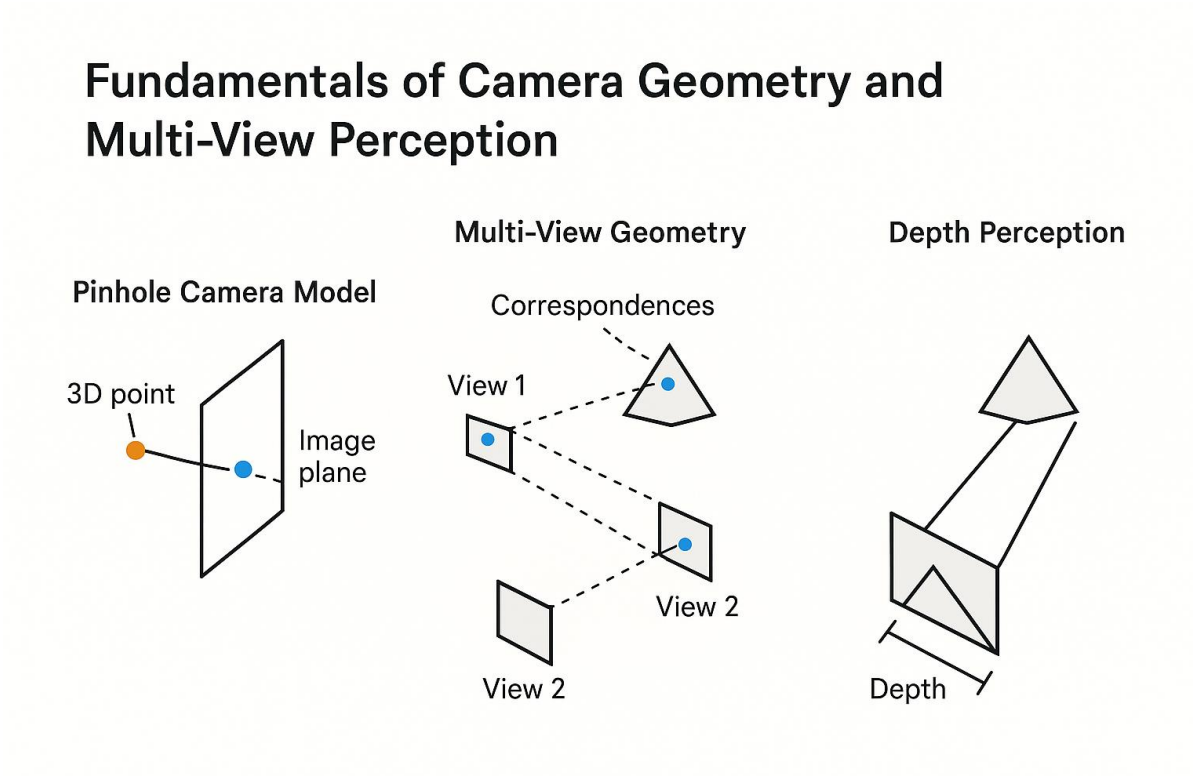


Figure 1 - "Key concepts in camera geometry: the pinhole model for image formation, multi-view geometry for 3D reconstruction, and depth estimation for spatial understanding in robotics."

Practical Use in Robotics

While the mathematics behind these concepts can be complex, robots don't require a human-level mastery of them. Instead, they rely on:

- **Software frameworks** (e.g., OpenCV, ROS) to perform geometric calculations.
- **Sensor calibration** to ensure accurate results.
  This abstraction lets engineers focus on designing behaviour while the system handles the heavy lifting of vision geometry.

5.3. Feature Detection and Description

In robotic vision, **features** are distinctive and repeatable patterns in an image—such as **corners**, **edges**, or textured regions—that can be reliably identified across different frames or viewpoints. They act like "landmarks" in an image, enabling robots to recognise and track elements over time, even when the scene changes.

Feature Detectors
Feature detectors identify *interest points* that remain stable despite changes in **viewpoint**, **lighting**, or **scale**.

- **Classical detectors** include:
    - **Harris** and **Shi–Tomasi corner detectors** – respond to strong intensity changes in multiple directions, ideal for identifying corners and textured points.
    - **FAST (Features from Accelerated Segment Test)** – uses simple binary tests for rapid detection, making it suitable for real-time applications.
- **Learned detectors** use **deep learning** to train on large datasets, enabling higher robustness under challenging conditions (e.g., extreme lighting or motion blur).

Example: A warehouse robot detects the same shelf corner from different angles as it moves through aisles.

Descriptors
Once features are detected, **descriptors** encode the local image patch around each feature into a compact numerical representation.

- **Traditional descriptors**:
    - **SIFT (Scale-Invariant Feature Transform)** – robust to scale, rotation, and lighting changes.
    - **ORB (Oriented FAST and Rotated BRIEF)** – designed for speed and efficiency on embedded hardware.
- **Learned descriptors** optimise feature matching under complex conditions but may require more processing power.

Example: Encoding a detected corner so it can be matched with the same corner in another camera view, even if the camera rotated.

Feature Matching and Tracking
Descriptors are compared between images to **find correspondences**—pairs of points that represent the same real-world location.

- **Matching** is used for tasks like object recognition or loop closure in mapping.
- **Tracking** updates these correspondences frame by frame, enabling real-time motion estimation (visual odometry) and detection of moving objects.

Example: A drone tracking visual features on the ground to estimate its own flight path.

Why Feature Quality Matters

High-quality features—those that are distinctive, stable, and efficiently computed—are crucial for the accuracy of **localisation**, **mapping**, and **navigation**. Designing feature extraction for robotics is a balancing act between:

- **Robustness** – handling lighting changes, motion blur, and viewpoint shifts.
- **Speed** – meeting real-time constraints.
- **Resource efficiency** – operating on embedded processors with limited power and memory.

5.4. Object Detection and Scene Understanding

For a robot to act intelligently, it must not only see objects but also **identify what they are** and **where they are**. Object detection assigns semantic labels (e.g., *"chair"*, *"box"*) and precise locations to objects within an image, providing the foundation for decision-making, navigation, and manipulation.

Detection Paradigms

- **Two-Stage Detectors**
  These systems first **propose candidate regions** where objects might be, then **classify and refine** those regions.
  - **Strengths:** High accuracy, especially for small or overlapping objects.
  - **Trade-off:** Slower due to the two-step process.

  Example: Faster R-CNN detecting multiple tools in a workbench scene.

- **One-Stage Detectors**
  Instead of generating proposals, these detectors **predict object locations and classes directly** over a dense image grid.
  - **Strengths:** Faster, suitable for real-time tasks.
  - **Trade-off:** May lose accuracy for small or hard-to-distinguish objects.

  Example: YOLO or SSD enabling real-time pedestrian detection for autonomous vehicles.

- **Transformer-Based Methods**
  These reframe detection as a **set prediction** problem using attention mechanisms, allowing the model to reason about relationships between objects.
  - **Strengths:** Flexible, promising performance in complex scenes.
  - **Trade-off:** More complex training and longer convergence times.

  Example: DETR detecting multiple interacting objects in a cluttered kitchen.

Segmentation — Beyond Bounding Boxes

Bounding boxes are useful but limited. Segmentation provides **pixel-level understanding** of scenes:

- **Semantic Segmentation** – Labels each pixel by class (*road*, *tree*, *robot arm*). Useful for finding drivable areas or identifying obstacles.
- **Instance Segmentation** – Distinguishes between **individual objects** of the same class (*three separate chairs* instead of one "chair" blob).
- **Panoptic Segmentation** – Combines both semantic and instance segmentation for a complete scene breakdown.

Example: A delivery robot identifies the exact shape and boundary of a package to grasp it without touching fragile surroundings.

3D Detection and 6D Pose Estimation
Many robotic tasks—especially **manipulation**—require knowing not just where an object appears in 2D, but also its **3D position and orientation** (the full six degrees of freedom: x, y, z, roll, pitch, yaw).

- **3D Detection:** Finds volumetric bounding boxes in space.
- **6D Pose Estimation:** Aligns 3D object models to sensor data (RGB, depth, LiDAR) for precise positioning.

Example: A robotic arm aligning itself to pick up a wrench lying at an angle on a shelf.

Why This Matters in Robotics

By combining **semantic understanding** (what objects are) with **geometric understanding** (where they are in space and how they're oriented), robots can plan safe and effective interactions—whether it's a factory robot assembling parts, a drone avoiding obstacles, or an autonomous car navigating traffic.

5.5. Tracking and Data Association

For robots to operate effectively, they must maintain **persistent awareness** of objects and features in their environment—not just in a single frame, but over time. This capability allows them to reason about motion, predict future states, and interact safely with dynamic surroundings.

Tracking
While object detection identifies what is present in each frame, **tracking** ensures that the same object is recognised and followed from one frame to the next.

- Tracking algorithms combine **motion models** (predicting where an object will move next) and **appearance models** (recognising an object's visual characteristics).
- This persistence allows robots to:
    - **Predict trajectories** (e.g., estimating where a pedestrian will be in two seconds).
    - **Plan safe manoeuvres** (e.g., adjusting a drone's flight path to avoid a moving obstacle).

Example: A warehouse robot following the same pallet as it moves between shelves, even when briefly out of view.

Data Association
Data association is the process of **matching detections across frames** to maintain correct object identities.

- Challenges include:
    - **Occlusions** – objects temporarily hidden behind others.
    - **Missed detections** – objects not detected in a frame due to sensor noise or lighting changes.
    - **Lookalikes** – multiple similar-looking objects causing confusion.
- Solutions often blend:
    - **Motion predictions** from filters like Kalman or particle filters.
    - **Visual embeddings**—feature representations that help tell objects apart.

Example: An autonomous car distinguishing between two cyclists riding close together.

Why It Matters

Reliable tracking and robust data association are the bridge between **moment-to-moment perception** and **long-term scene understanding**. They enable higher-level reasoning such as:

- **Intent prediction** – anticipating whether a vehicle will turn or stop.
- **Long-term environment modelling** – building a consistent map of dynamic elements in the scene.

By keeping track of "who's who" in the environment, robots can plan actions that are both **proactive** and **safe**.

5.6. Simultaneous Localisation and Mapping (SLAM) and Visual-Inertial Odometry (VIO)

For a robot to navigate autonomously, it must know **where it is** and **what the environment looks like**—even in places it has never been before. **SLAM** and **VIO** achieve this by integrating **visual perception** with **motion sensing**, enabling a robot to **localise itself** while **building a map** of its surroundings in real time.

Feature-Based SLAM

- Detects and tracks distinctive image features (corners, edges, textures) across multiple views.
- Estimates the robot's trajectory while building a **sparse map** of landmark positions in 3D.
- **Strengths:** Works well in visually rich environments with many trackable points.

Example: A drone mapping a construction site by tracking windows, corners, and edges of buildings.

Direct SLAM

- Avoids discrete feature detection, instead using **raw pixel intensity differences** between images to estimate motion.
- **Strengths:** More robust in texture-poor or low-feature environments (e.g., white walls).

Example: A service robot navigating a smooth-walled hallway with few distinct visual features.

Visual-Inertial Odometry (VIO)

- Combines visual SLAM techniques with **Inertial Measurement Unit (IMU)** data.
- **Advantages:**
    - Improves accuracy during fast motion, where motion blur might reduce visual reliability.
    - Resolves **scale ambiguity** in monocular setups.

Example: An autonomous rover driving over rough terrain while keeping precise location tracking despite sudden jolts and turns.

Why It Matters

SLAM and VIO provide **foundational spatial awareness**—a continuous understanding of a robot's location and the surrounding map. This is essential for:

- **Navigation** – planning safe and efficient routes.
- **Interaction** – aligning tools or grippers with objects in 3D space.
- **Mapping** – documenting environments for future missions.

Without SLAM or VIO, robots would be effectively "blindfolded," unable to move safely or reliably in unfamiliar spaces.

5.7. Sensor Fusion and Calibration

Modern robots rarely rely on a single sensor. By **combining data from multiple sensors**, they can achieve richer, more reliable perception. This process—known as **sensor fusion**—is only effective when the sensors are precisely **calibrated,** so their data aligns both in **space** and **time**.

Spatial Calibration

- Ensures all sensors share a **common spatial reference frame**.
- Involves finding the exact **rigid-body transform** (position and orientation offset) between sensors.

Example: If a robot's camera is mounted 20 cm to the left of its LiDAR, spatial calibration ensures that objects detected by the LiDAR are correctly overlaid on the camera image.

Temporal Calibration

- Aligns sensor data streams in **time**, so events from different sensors refer to the same moment.
- Even small timing errors—on the order of milliseconds—can cause mismatches in **pose estimation**, **obstacle localisation**, or **motion tracking**.

Example: An autonomous car braking too late because the camera detected a pedestrian slightly after the radar did, due to unsynchronised timestamps.

Fusion Algorithms

- Process **asynchronous** and **noisy** sensor readings to generate accurate, low-latency estimates of the robot's state and environment.
- Methods include:
    - **Kalman filters** for smoothly combining measurements.
    - **Particle filters** for handling non-linear, multi-modal uncertainties.
    - **Bayesian fusion** for probabilistic reasoning over sensor data.

Example: Combining LiDAR's accurate range measurements with a camera's rich texture data to create a precise, visually detailed 3D map.

Why Calibration Matters

Robust calibration is the foundation for successful sensor fusion. Without it, even the most advanced algorithms will combine **misaligned** or **mistimed** data, leading to unreliable perception. With proper calibration, robots gain a **coherent, trustworthy view** of the world, enabling safer navigation, better manipulation, and more accurate decision-making.

5.8. Data Annotation, Real-World Challenges, and Robustness

The performance of a robotic perception system depends not only on advanced algorithms but also on the **quality and diversity of the data** it learns from. Real-world environments introduce complexities that must be addressed through careful dataset design, annotation, and robustness strategies.

Annotation Types
High-quality labels are essential for training and evaluating perception models. Common types include:

- **2D Bounding Boxes** – Rectangles marking object locations in images.
- **Dense Semantic Masks** – Pixel-by-pixel classification of scene elements.
- **3D Annotations** – Volumetric bounding boxes or point cloud labels.
- **Pose Alignments** – Six-degree-of-freedom (6DoF) object positions and orientations.

Example: Labelling every pedestrian in a street scene with both a 2D box and their 3D position relative to the robot.

Challenges in Real Data
Real-world conditions introduce problems that complicate both annotation and learning:

- **Occlusions** – Parts of an object are hidden behind other objects.
- **Truncations** – Objects partially cut off at image edges.
- **Small or Distant Objects** – Reduced pixel detail makes recognition harder.
- **Ambiguous Materials** – Transparent glass or reflective metal can confuse both sensors and human annotators.

Example: An autonomous car trying to detect a cyclist partly hidden behind a parked van.

Synthetic and Simulated Data
Virtual environments allow safe, scalable creation of training scenarios, especially for rare or hazardous events.

- **Strengths:** Controlled conditions, infinite variation, and safety in simulating crashes or extreme weather.
- **Limitations:** "Reality gap" — differences in texture, lighting, and physics from real-world data.
- **Solution:** Domain adaptation techniques bridge simulated and real data distributions.

Example: Training a delivery robot in a simulated city before fine-tuning it with real street data.

Handling Edge Cases

Maintaining robust performance in unpredictable conditions requires strategies at multiple levels:

- **Hardware:** High Dynamic Range (HDR) imaging for extreme lighting; LiDARs that are less affected by glare.
- **Algorithms:** Motion-blur compensation, adaptive thresholds for weather effects.
- **Uncertainty Awareness:** Models that estimate their own confidence to flag uncertain predictions.

Example: A drone maintaining stable obstacle detection in both bright sunlight and fog.

Comparing Data Annotation, Real-World Challenges, and Robustness

Challenge	Impact on Perception	Example Scenario	Mitigation Strategies
Occlusion	Objects are partially hidden, leading to incomplete detection or tracking	Pedestrian walking behind a parked car	Use temporal tracking to recover hidden objects; employ multiple sensor viewpoints
Truncation	Loss of object context when partially outside the frame	Cyclist entering from image edge	Expand camera field of view; predict object continuation beyond visible area
Small or Distant Objects	Reduced pixel detail causes missed or false detections	Drone spotting a faraway bird	Use high-resolution imaging; super-resolution techniques
Ambiguous Materials (Transparent/Reflective)	Sensors misinterpret object boundaries or fail to detect	Glass door mistaken for open space	Fuse LiDAR or radar data with vision; train with material-specific examples
Adverse Illumination	Overexposure or underexposure hides features	Robot navigating under flickering streetlights	HDR imaging; adaptive exposure control
Motion Blur	Loss of fine details during fast movement	Drone filming while making rapid turns	Motion-compensated algorithms; high-speed shutters
Weather Effects (Fog, Rain, Snow)	Reduced visibility, noisy sensor data	Autonomous car in heavy rain	Sensor fusion with radar/LiDAR; train models with weather-augmented data
Domain Shift (Synthetic → Real)	Model underperforms when real data differs from training data	Robot trained in simulation misidentifies real objects	Domain adaptation; mixed synthetic and real training datasets
Uncertainty in Predictions	Incorrect confident outputs can cause unsafe actions	Robot misclassifies object but acts without checking	Confidence estimation; human-in-the-loop review in safety-critical settings

Table 2 - "Real-world challenges in robotic perception and how to address them."

Why It Matters

Systems that **anticipate and manage these challenges** are better equipped to handle the unpredictability of real-world deployment. Robust perception enables robots to remain safe, reliable, and effective—whether navigating city streets, exploring disaster zones, or working in busy factories.

5.9. Evaluation Metrics for Robotics Vision

Assessing the performance of robotic vision systems is not just about accuracy—it also involves **efficiency**, **robustness**, and the system's ability to perform reliably in real-world conditions. A balanced evaluation considers multiple aspects of perception, from raw detection quality to resource usage.

Detection Metrics
Used to measure how well the system identifies and localises objects in images.

- **Precision / Recall** –
    - *Precision:* The proportion of detected objects that are correct.
    - *Recall:* The proportion of actual objects successfully detected.
    - *Example:* A warehouse robot detecting 9 out of 10 packages (90% recall) but misidentifying one box as a package (lowering precision).
- **Average Precision (AP)** – Summarises precision–recall performance across different confidence thresholds; commonly used in benchmarks like COCO.
- **Intersection over Union (IoU)** – Measures the spatial overlap between predicted and ground-truth bounding boxes.

Pose Estimation Metrics
Evaluate how accurately the system predicts **3D position** and **orientation** (rotation).

- **Position Error:** Distance between predicted and actual object location.
- **Orientation Error:** Difference in angular alignment.

Example: A robotic arm failing to align its gripper precisely with a bolt due to a 2° rotation error.

Tracking Metrics
Assess the ability to **maintain consistent object identities** and accurate positions over time.

- **ID Switches:** Number of times the tracker confuses one object for another.
- **Multiple Object Tracking Accuracy (MOTA)** – Combines missed detections, false positives, and ID switches into a single score.

Example: An autonomous vehicle loses track of a pedestrian after they pass behind a lamppost.

SLAM and VIO Metrics
Measure accuracy in **trajectory estimation** and **map quality**.

- **Absolute Trajectory Error (ATE):** How far the estimated path deviates from the ground truth.
- **Relative Pose Error (RPE):** Accuracy of short-term motion estimates.
- **Map Consistency:** How well landmarks align across revisited locations.

Computational Metrics
Determine if the system can meet real-world constraints for deployment.

- **Latency:** Time to process each frame.
- **Throughput:** Number of frames processed per second.
- **Memory and Power Usage:** Critical for embedded and mobile platforms.

Example: A drone's vision system failing mid-flight due to excessive power consumption.

Why Balanced Evaluation Matters

A robotic vision system that scores high on accuracy but fails to run in real time—or consumes too much power—may be impractical in real deployments. **Holistic evaluation** ensures the system meets the **complex demands** of robotics: accurate, fast, efficient, and robust in the face of real-world challenges.

5.10. Embedded Systems and Real-Time Considerations

Robotic vision systems must operate under **strict real-time constraints**, where delays or inconsistencies can directly impact safety and performance. These constraints influence everything from **algorithm design** to **hardware selection**.

Latency

- **Definition:** The delay from capturing a sensor frame to making a decision based on it.
- **Impact:** High latency means the robot is acting on outdated information, which can degrade control quality or cause unsafe actions.

Example: An autonomous drone avoiding an obstacle 200 milliseconds too late because its vision pipeline lagged.

Throughput and Jitter

- **Throughput:** Number of frames processed per second.
- **Jitter:** Variability in processing time between frames.
- **Impact:** Even if average speed is high, unpredictable timing can destabilise control loops.

Example: A robotic arm performing precise assembly becomes unstable due to occasional slow vision updates.

Resource Constraints

- **Challenge:** Embedded processors have limited **CPU/GPU performance**, **memory**, and **energy budgets**.
- **Solutions:**
    - **Model quantisation** – Reducing numerical precision to speed up inference.
    - **Pruning** – Removing redundant network weights.
    - **Specialised accelerators** – Using GPUs, TPUs, or FPGAs for parallel processing.

Example: An agricultural robot using a quantised vision model to run for an entire day on battery power.

Software Pipelines

- Vision systems often use **asynchronous, pipelined architectures** to keep sensors, processing units, and control systems working in parallel.
- This reduces idle time and increases efficiency, even on limited hardware.

Example: While one image is being processed, the next is already being captured and queued.

Why It Matters

Real-time performance isn't just a **speed issue**—it's about **predictability** and **reliability**. Designing for low latency, consistent throughput, and efficient resource use ensures that robotic vision systems respond to the world **as it is now**, not as it was a fraction of a second ago.

5.11. Application-Specific Considerations Across Robot Types

While the core principles of robotic vision remain the same, **each robot type has different priorities** based on its platform, operating environment, and task requirements. Designing perception systems with these differences in mind greatly improves performance and safety.

Mobile Ground Robots

- **Priorities:** Broad environmental awareness, reliable obstacle detection, and path planning in dynamic settings.
- **Approach:**
    - Wide field-of-view cameras or LiDAR for maximum coverage.
    - Multi-sensor fusion to compensate for sensor weaknesses (e.g., radar in low visibility).

Example: An autonomous delivery robot navigating crowded sidewalks, avoiding pedestrians, and detecting curbs or steps.

Manipulators and Collaborative Robots (Cobots)

- **Priorities:** Precise 3D understanding of the workspace and safe interaction with humans.
- **Approach:**
    - High-accuracy 6DoF pose estimation of known objects.
    - Shape completion to reconstruct partially visible items.
    - Safety zone monitoring with depth sensors to pause or slow movement near people.

Example: A cobot assembling electronics while dynamically adjusting its movement when a human hand enters its workspace.

Aerial Robots (Drones)

- **Priorities:** Stability, fast decision-making, and obstacle avoidance in open and cluttered airspace.
- **Approach:**
    - Ultra-low-latency visual–inertial odometry for rapid movement.
    - Vibration-tolerant sensors to handle rotor-induced motion.
    - High-resolution vision for detecting small or distant hazards like wires or tree branches.

Example: A powerline inspection drone spotting thin cables against a bright sky.

Logistics and Warehouse Robots

- **Priorities:** Accurate detection in visually challenging industrial settings and reliable indoor localisation.
- **Approach:**
    - Specialised algorithms for reflective or transparent surfaces (e.g., shrink-wrapped packages).
    - Robust detection in repetitive textures like shelving units.
    - Use of fiducial markers or visual tags in feature-sparse areas.

Example: An automated pallet mover identifying transparent plastic wrap on packages and navigating accurately in a warehouse with uniform shelving.

Why It Matters

Tailoring robotic vision to **platform-specific needs** ensures that each system is not only functional but also **optimised for its operational context**—whether it's navigating busy streets, working alongside humans, flying through tight spaces, or streamlining warehouse logistics.

5.12. Safety, Compliance, and Ethical Considerations

As robots increasingly share workspaces, roads, and public spaces with humans, **safety, trust, and ethical responsibility** become as important as technical performance. Robotic vision systems play a central role in meeting these requirements.

Functional Safety

- **Goal:** Prevent harm even in the presence of faults or unexpected conditions.
- **Approach:**
    - **Redundancy:** Multiple sensing modalities (e.g., camera + LiDAR) to ensure continued perception if one fails.
    - **Fault detection:** Continuous self-checks for sensor or software malfunctions.
    - **Failsafe states:** Safe stop, slowdown, or retreat behaviours when critical failures are detected.

Example: An autonomous forklift stops automatically if its vision and proximity sensors disagree on obstacle detection.

Human–Robot Interaction (HRI)

- **Goal:** Enable safe, predictable, and comfortable cooperation between robots and humans.
- **Approach:**
    - Detect and track human presence in real time.
    - Maintain safety zones or dynamic separation distances.
    - Gate or adjust robot actions based on system confidence levels.

Example: A collaborative robot pauses its arm movement when a worker reaches into its workspace.

Privacy and Data Governance

- **Challenge:** Vision systems often capture identifiable human data, raising legal and ethical issues.
- **Approach:**
    - Limit or anonymise stored image data.
    - Maintain **audit trails** to trace decision-making steps.
    - Ensure reproducibility of perception outputs for compliance reviews.

Example: A public-service robot using onboard processing to detect people without storing raw facial images.

Why It Matters

Integrating safety, compliance, and ethics **from the earliest stages of design** fosters public trust, eases regulatory approval, and ensures robots behave responsibly in complex human environments. Systems that meet both **technical** and **ethical** standards are more likely to be accepted, adopted, and sustained in real-world use.

5.13. Future Directions in Robotic Vision

Robotic vision is evolving rapidly, driven by advances in AI, sensing technologies, and computing hardware. Several key trends are shaping the next generation of perception systems:

Learning and Adaptation

- **Goal:** Enable robots to function reliably in diverse, changing environments without extensive retraining.
- **Approach:**
    - **Domain adaptation** to handle differences between training and deployment conditions (e.g., moving from sunny outdoor scenes to dim indoor spaces).
    - **Continual learning** to integrate new experiences without forgetting past knowledge.
    - **Online adjustment** for adapting to sudden changes like lighting shifts or moving furniture.

Example: A delivery robot learning to navigate new neighbourhood layouts without manual reprogramming.

Explainability

- **Goal:** Make vision system decisions transparent and interpretable to humans.
- **Approach:**
    - Visualising which image regions influenced a decision.
    - Providing human-readable reasoning for detection or classification outputs.

Example: A safety inspector reviewing why a factory robot flagged an object as hazardous.

Multi-Modal Fusion

- **Goal:** Combine vision with other sensory modalities to achieve richer perception and reasoning.
- **Approach:**
    - Integrating **language understanding** for natural interaction (e.g., “pick up the red box”).
    - Adding **audio cues** to detect alarms, voices, or environmental sounds.
    - Using **tactile sensing** to confirm grasp quality or detect surface textures.

Example: A household robot using both vision and touch to fold laundry without damaging delicate fabrics.

Edge Intelligence

- **Goal:** Perform advanced perception directly on embedded devices, reducing reliance on cloud processing.
- **Approach:**
    - Leveraging next-generation **AI accelerators** and efficient model architectures.
    - Minimising latency and energy consumption while maintaining accuracy.

Example: An agricultural drone performing crop health analysis entirely onboard, without network connectivity.

Looking Ahead

These developments point toward robotic vision systems that are not only **more intelligent and adaptable**, but also **safer, more interpretable, and more capable of operating autonomously** in complex real-world environments. The future will see robots that can learn continually, explain their actions, and seamlessly integrate multiple senses—bringing them closer to human-like perception and reasoning.

Summary

In this unit, we explored the principles and practices that enable robots to transform raw visual input into a structured, actionable understanding of their environment. We began by examining **sensing modalities** and the importance of selecting and combining appropriate sensors for reliable perception. We then covered the **fundamentals of camera geometry**, multi-view perception, and depth estimation, which provide the geometric foundation for spatial reasoning.

A major focus was placed on **feature detection and description**, the process of identifying and representing distinctive visual patterns that support recognition, localisation, and mapping. We also examined **object detection and scene understanding**, moving from basic categorisation to precise pixel-level segmentation and 3D pose estimation—key capabilities for intelligent interaction with complex environments.

Building on these foundations, we studied **tracking and data association** for maintaining persistent awareness of dynamic scenes, as well as **SLAM and visual-inertial odometry** for continuous localisation and mapping. We addressed **sensor fusion and calibration** as critical enablers of robust perception, and discussed **data annotation, real-world challenges, and robustness** strategies to maintain performance under variable and unpredictable conditions.

Evaluation was framed holistically, considering not just accuracy but also **real-time constraints**, computational efficiency, and application-specific needs across diverse robot types. Finally, we examined **safety, compliance, and ethical considerations**, along with **emerging trends** such as explainability, continual learning, and multi-modal fusion.

Together, these concepts form a comprehensive toolkit for designing robotic vision systems that are **accurate, efficient, reliable, and adaptable**—capable of moving beyond simply “seeing” to truly **understanding and acting within** their environments.