

Human-Machine Interaction

Interaction Modalities &
Technologies

Motion Interaction

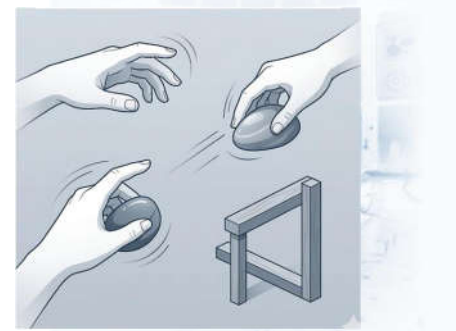
Dr. Patrick Chan
patrickchan@scut.edu.cn

South China University of Technology

1

Traditional Input Devices

- Requires **physical contact with an interface surface**
 - **Keyboard** → symbolic, discrete input
 - **Mouse** → indirect spatial control
 - **Touchscreen** → direct manipulation on surface
- However, **humans** interact with the **physical world without surfaces**



Motor Skill Mismatch

- **Mapping gap** between **action** and **intention**
 - **Humans** evolved for **grasping**, **pointing**, **rotating**, and **throwing**. Hand has 27 DoF
 - **Traditional interfaces** require **typing** symbols, **cursor** translation, and **tapping** icons. A mouse has 2 DoF
- Example: Rotating a 3D object
 - Mouse: drag + modifier keys
 - Human instinct: rotate hand



Surface Dependency

- Keyboard / mouse / touch **need surface**
 - E.g. **desk**, **screen**, **reachable panel**, and **clean environment**
- **Interaction breaks** when **surface disappears**
 - E.g. **driving**, **surgery**, **public displays**, and **VR/AR**



Attention Demand

- Precise input needs **visual confirmation**:
 - Typing accuracy
 - Cursor positioning
 - Button targeting
- Causes **danger** in **vision demanding tasks**: driving, piloting, and walking navigation



Need for Touchless Interaction

- **Sterile Operating Room**
 - Surgeons browse MRI/CT
 - Avoids touching screens → preserves sterility
- **Industrial Maintenance (AR)**
 - Technician confirms steps
 - No need to release tools
- **Smart Kitchen**
 - Cook flips recipe pages
 - Gands covered in flour → preserves sterility



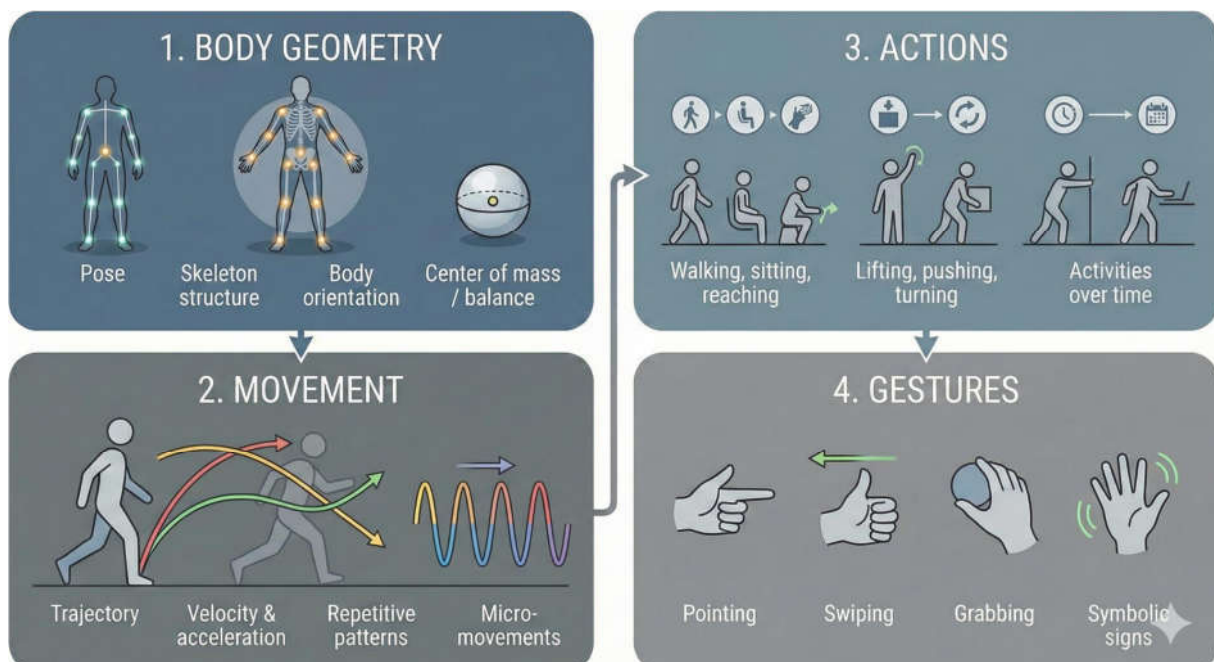
Motion Capture (Mocap)

- Process of recording the **movement** of objects or people
- **Goal**: To map **physical spatial changes** into a **digital coordinate system**
- Key Parameters
 - **Position** (x, y, z coordinates)
 - **Orientation** (Roll, Pitch, Yaw)
 - **Dynamics** (Velocity and Acceleration)



Motion Capture

What can be captured?



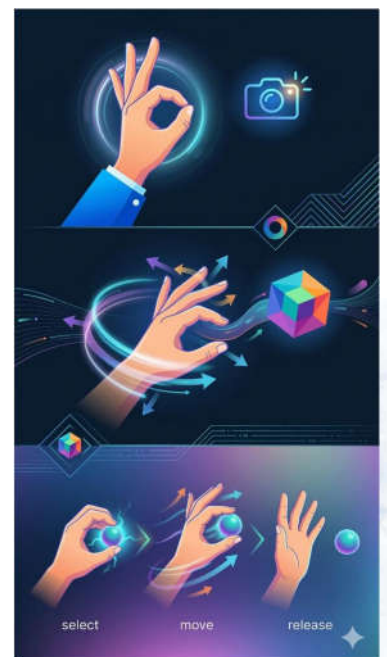
Levels of Motion Understanding

- **Body Geometry** → Raw physical **pose** data
 - Joint **positions**, **skeleton**, **posture**
- **Movement** → How the **body moves**
 - **Trajectory**, **speed**, **acceleration**
- **Actions** → What the person is **doing**
 - **Activities** over time (walking, sitting, lifting)
- **Gestures** → What the person wants to **communicate**
 - **Meaningful motions** (point, swipe, thumbs-up)



Static vs Continuous Gestures

- **Static Gestures**: Focus on **Spatial arrangement**
 - **One-time command**. Trigger event
 - E.g. “OK” sign → take photo
- **Dynamic Gestures**: Focus on **Velocity**, **Direction**, and **Rhythm**.
 - System **tracks motion continuously**
 - E.g. moving a virtual joystick
- **Hybrid Interaction**: Combine **both** types
 - E.g. “Grab” (start) → Move hand (control) → Release (stop)



Hand vs Body Gestures

- **Body Gestures**

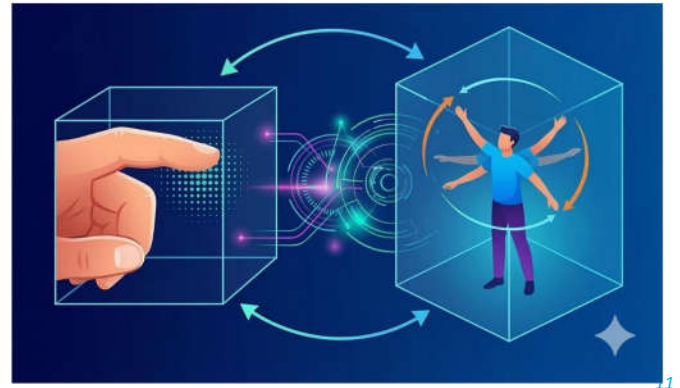
- Large-scale, lower precision
- Navigation, gaming, robot signaling at distance

- **Resolution Challenge**

- System must handle very different scales
 - ~1 cm finger movement
 - ~1 m body movement

- **Hand Gestures**

- Small-scale, high precision
- Fine control (virtual sliders, typing, object manipulation)



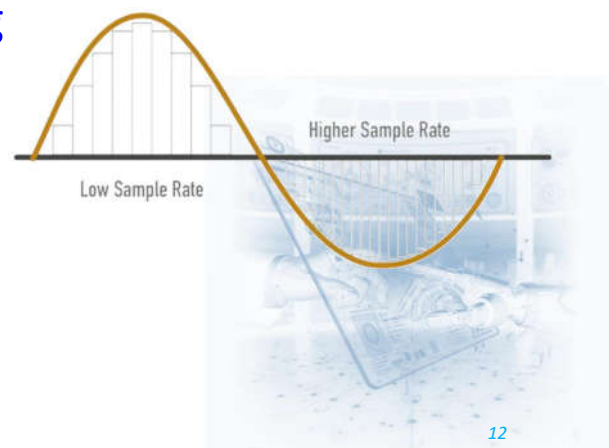
Quantization

- Human **continuous motion** is **converted** into small **discrete steps**

- **Sampling Rate**

Faster motion needs faster sensing

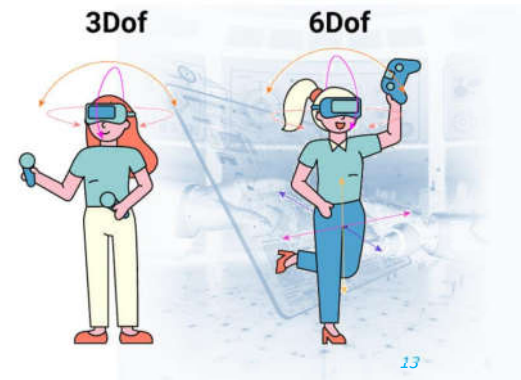
- **~30 Hz** → smooth enough for UI interaction
- **+120 Hz** → needed for fast motion analysis (sports, medical)



Quantization

- Human **continuous motion** is **converted** into small **discrete steps**
 - **Degrees of Freedom (DoF)**
More complicated actions need larger DoF

- **3-DoF:**
rotation only
(look / tilt / turn)
- **6-DoF:**
rotation + position
(move forward, backward, sideways)



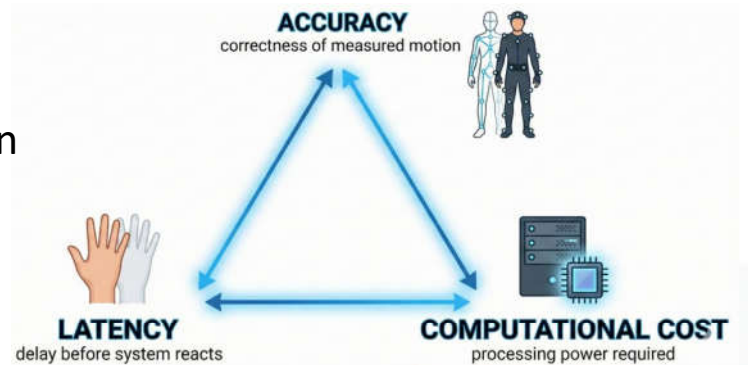
Offline vs Real-Time

- **Offline (Post-processing)**
 - Priority: **precision**
 - Used in **filmmaking** (e.g., performance capture)
 - **Errors** can be **corrected manually**
- **Real-Time (Low Latency)**
 - Priority: **speed & responsiveness**
 - Used in **VR/AR and robotics control**
 - If latency > ~20 ms → users feel delay

Trade-off

• Constraints Triangle

- **Accuracy**
correctness of measured motion
- **Latency**
delay before system reacts
- **Computational Cost**
processing power required



• Trade-off example:

- High accuracy often requires heavy calculation
- Low latency may result in bad quality



15

Devices & Sensors

- **RGB camera**
standard video tracking
- **Infrared (IR) camera**
low-light / marker tracking
- **RGB-D camera**
color + depth
- **Stereo camera**
depth from two cameras
- **Laser / LiDAR**
precise distance scanning
- **Time-of-Flight (ToF)**
measures light travel time
- **IMU**
accelerometer + gyroscope + magnetometer
- **Radar (mmWave)**
- **Ultrasonic tracking**
- **Magnetic tracking systems**
- **RF / Wi-Fi sensing**

16

RGB Camera

- A standard color camera that captures images in visible light
- **How It works**
 - Captures RGB lights by channels
 - Combines channels to form a full-color 2D image
- **What It Can Detect**
 - Visible light
 - Body pose & skeleton (+AI)
 - Hand gestures (+AI)
 - Facial expressions (+AI)

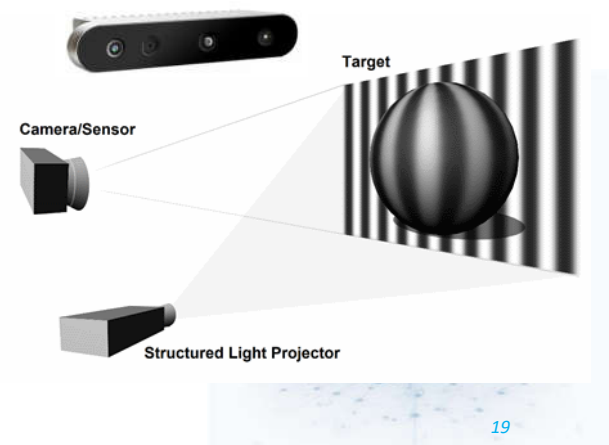


RGB Camera

- **Strengths**
 - Low cost & widely available
 - No wearable devices required
 - Rich visual information
- **Limitations**
 - Sensitive to lighting conditions
 - Occlusion problems (blocked view)
 - Depth estimation is indirect

RGB-D (Color+Depth) Camera

- A camera that captures both **color image and depth distance**
- **How It Works**
 - Structured **light** or **Time-of-Flight** measures distance to each pixel
 - Produces a **3D scene map**
- **What It Can Detect**
 - **All RGB camera can do**
 - **Object interaction depth**
 - Accurate body **skeleton tracking** (+AI)
 - Hand **position** in **3D space** (+AI)



RGB-D (Color+Depth) Camera

- **Strengths**
 - Direct **3D measurement**
 - **More reliable** than RGB alone
 - Works in **moderate lighting**
- **Limitations**
 - Limited range
 - Sensitive to **sunlight/reflective surfaces**
 - Higher cost than RGB camera

Inertial Measurement Unit (IMU)

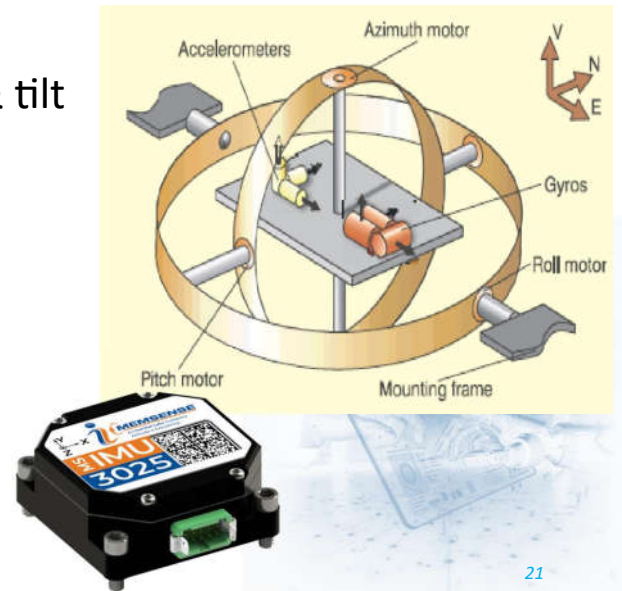
- A small sensor that measures motion using **internal physical forces**

- **How It Works**

- **Accelerometer** → linear movement & tilt
- **Gyroscope** → rotation
- **Magnetometer** → orientation

- **What It Can Detect**

- Device orientation (roll, pitch, yaw)
- Steps & gait
- Hand and body motion
- Sudden impacts / falls



Inertial Measurement Unit (IMU)

- **Strengths**

- Works **without cameras**
- **Not affected by lighting**
- **High speed & low latency**

- **Limitations**

- **Drift over time**
- Needs **calibration**
- **Limited absolute position tracking**

Optical Marker

- **Reflective "ping-pong" balls** placed on anatomical joints to reduce the dependence on object recognition
- **Pros**
 - Extreme sub-millimeter accuracy
 - High frame rates
- **Cons**
 - Expensive setup cost
 - Setup required



23

Wearable Device

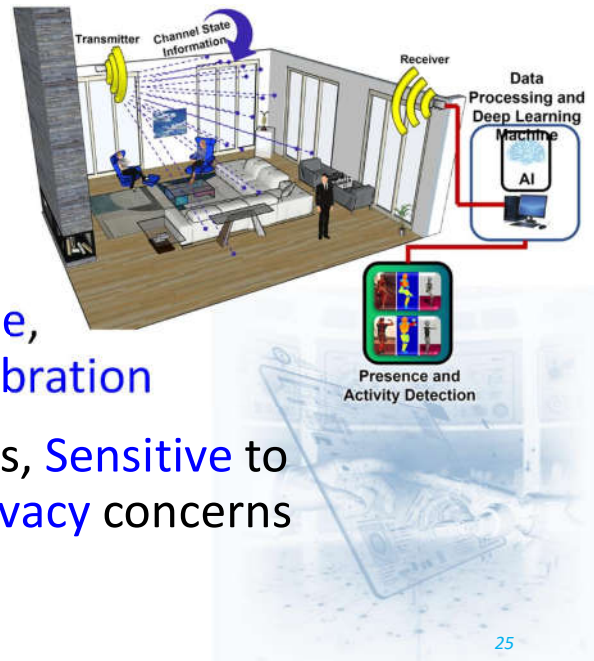
- Devices **worn on the body**
- **Pros**
 - High precision & stable tracking
 - Works without cameras or lighting
 - Robust to occlusion
- **Cons**
 - Must wear equipment
 - Comfort & battery constraints
 - Setup/calibration required



24

Unobtrusive Contactless Detection

- **User behaves normally**
 - No physical touch
 - No markers or wearable devices
 - No explicit instruction required
- **Pros:** Natural interaction, Comfortable, Hygienic, Minimal user effort, No calibration
- **Cons:** Lower precision than wearables, Sensitive to environment (lighting, occlusion), Privacy concerns



Application-Driven Selection

- **No “best”** motion capture device
- **Different applications** require different priorities: Precision, Latency, Robustness, User comfort, Budget,
- **Environment constraints**
 - **Lighting:** Bright sunlight / Indoor controlled
 - **Occlusion:** Crowded / Clear view
 - **Mobility:** Moving user / Fixed space



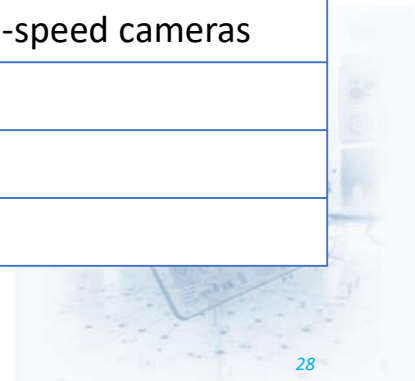
Application-Driven Selection

- Human **comfort** → **contactless** sensing
- **Precision** → **wearables / markers**
- **Safety-critical** → **low latency** sensors
- **Long-term** monitoring → **unobtrusive** sensing



Application-Driven Selection

Application	Priority	Suitable Devices
VR/AR interaction	Low latency	IMU, RGB-D
Medical surgery	Sterile & precise	IR, RGB-D, Optical tracking
Sports analysis	High accuracy	Optical MoCap, High-speed cameras
Smart home	Convenience	RGB camera, Radar
Industrial work	Hands occupied	Wearables, Radar
Public kiosk	No contact	RGB-D, LiDAR



Application-Driven Selection

• Scenario A

- Tracking a professional athlete's gait
- Optical Markers



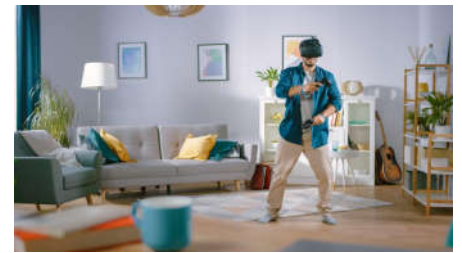
• Scenario B

- Controlling a smart TV from a sofa
- Depth + Vision



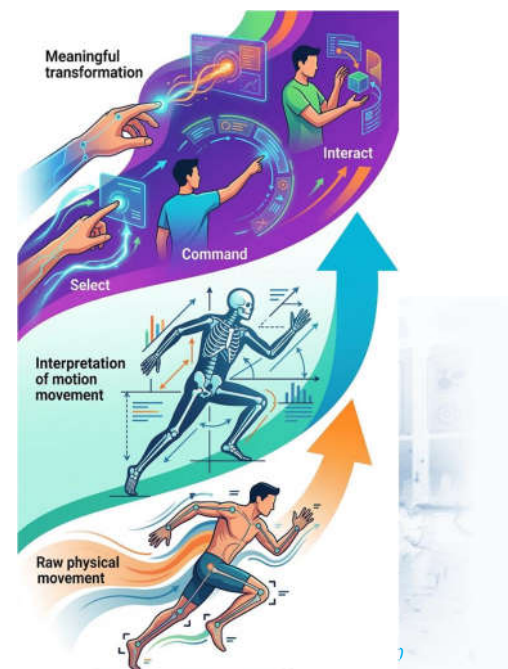
• Scenario C

- VR gaming in a living room
- IMU + Vision



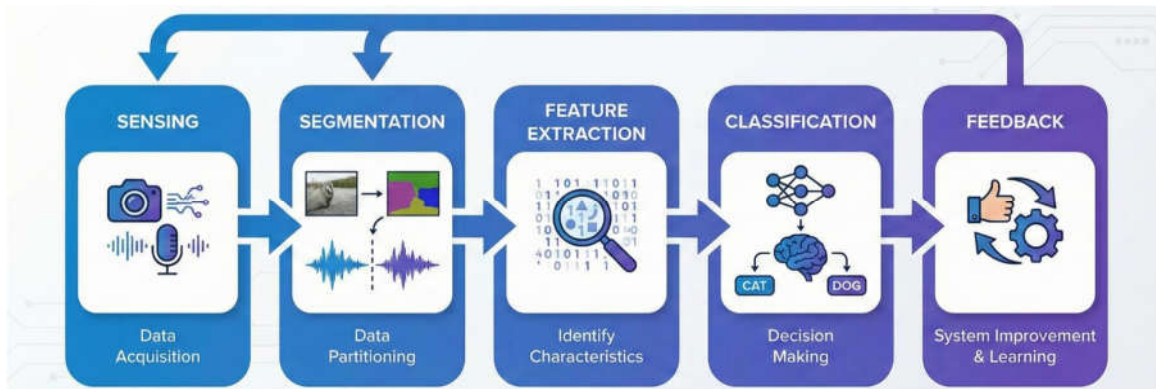
Recognition Pipeline

- Motion Capture is **more than detection**
 - It does not only detect movement
It interprets human intention
- AI **transforms** movement into **meaning**



Recognition Pipeline

- **Recognition Pipeline**

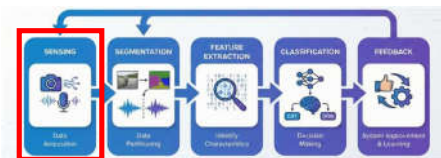


- Which part is the hardest?



31

Recognition Pipeline **Sensing**



- Sensors capture raw motion data
 - Not only visual information
- **Output**
Large continuous data stream
- **Challenge**
Noise, occlusion, drift



32

Sensing (Hardware Design)

- In practice, the **hardest part** is not the algorithm but **data collection**
 - Garbage In, Garbage Out
- Common Problems: **Noise, misuse**
- Even advanced **AI fails** if input data is **unreliable**
- How can we collect **high-quality motion data**?



33

Sensing (Hardware Design)

- **Sensing Volume**
 - Define the **interaction zone** clearly
 - User must stay **inside reliable detection region**
 - **Too near / too far** reduces accuracy
- **User Factors**
 - **Wearing position consistency**
 - **Body occlusion**
 - **Multiple users in space**
 - **Natural movement variability**

34

Sensing (Hardware Design)

• Calibration

- Coordinate alignment
- Orientation reference
- Initial pose capture
- Periodic recalibration needed

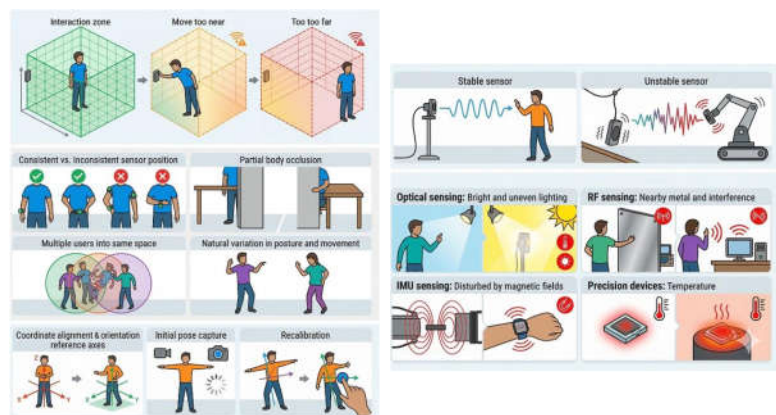
• Mounting & Placement

- Stable mounting reduces noise
- Align sensor with expected motion direction
- Avoid vibration and moving supports

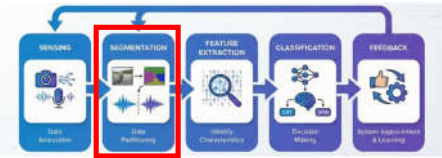
Sensing (Hardware Design)

• Environment Conditions

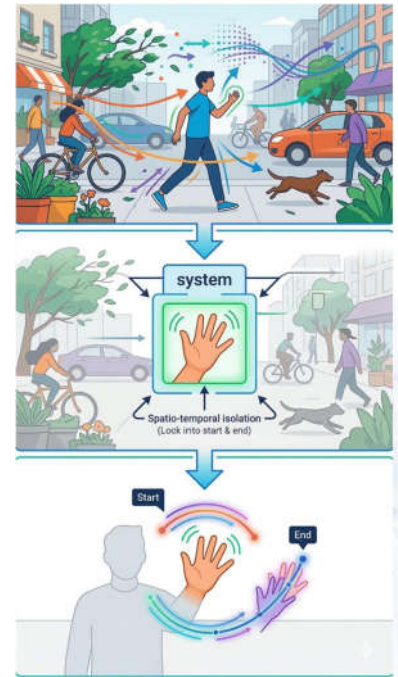
- Lighting affects optical sensors
- Metal & interference affect RF sensors
- Magnetic disturbance affects IMU
- Temperature may affect precision devices



Segmentation

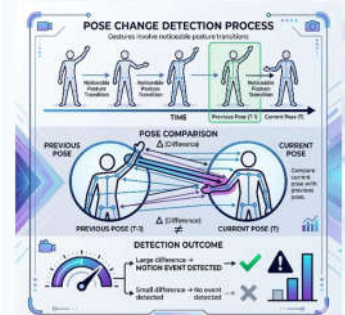
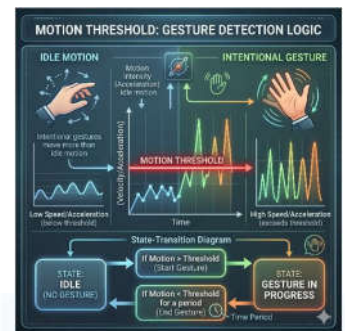


- Extract the **related information**
 - **Temporal**: when motion starts and ends
 - **Spatial** : Region of Interest (ROI), Position
 - Without segmentation, the system reacts to too much irrelevant information
- **Goal**: **Separate** meaningful motion from background



Temporal Segmentation Method

- **Motion Threshold (speed/acceleration)**
 - Intentional gestures **move more than idle motion**
 - **Monitor velocity / acceleration**
 If motion > threshold → start gesture
 If motion < threshold for a period → end gesture
- **Pose Change Detection**
 - Gestures involve **noticeable posture transitions**
 - **Compare current pose with previous pose**
 Large difference → motion event detected



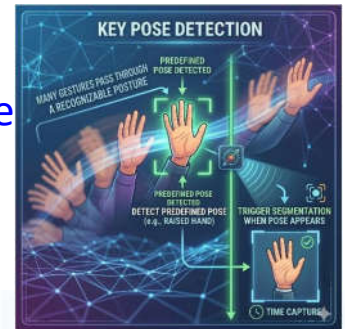
Temporal Segmentation Method

• Key Pose Detection

- Many gestures pass through a recognizable posture
- Detect predefined pose (e.g., raised hand)
Trigger segmentation when pose appears

• Wake Gestures (raise hand, hold still)

- System should only listen after explicit intention
- Wait for special gesture (hold hand up, open palm)
Enable recognition mode until inactivity timeout



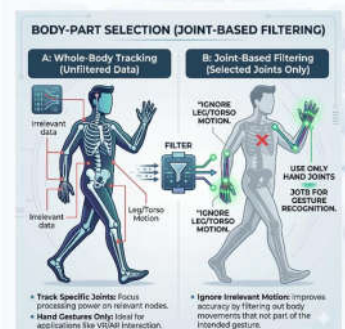
Spatial Segmentation Method

• Region of Interest (ROI) Filtering

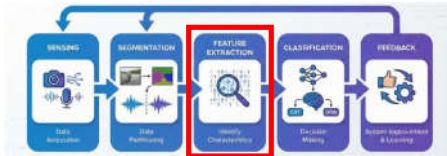
- Only monitor motion inside a predefined interaction area
- Track only the hand zone in front of a screen
Ignore background or body movement

• Body-Part Selection (Joint-Based Filtering)

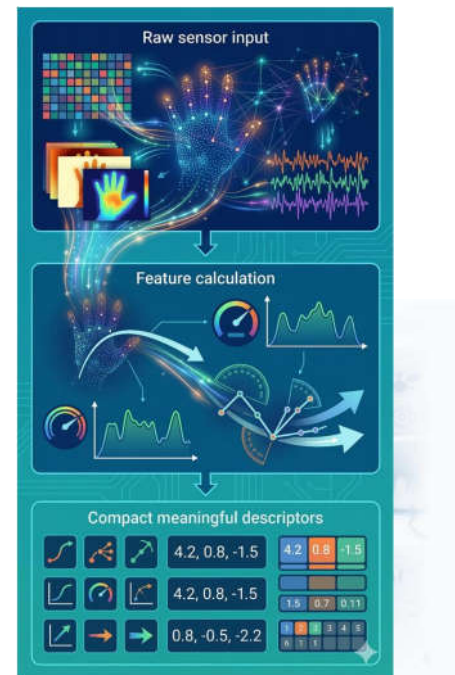
- Track specific joints instead of the whole body
- Use only hand joints for gesture recognition
Ignore legs and torso motion



Feature Extraction



- **Convert raw data into measurable characteristics**
 - E.g. Hand trajectory, Speed, Angle change, Motion direction
- Reduce data size but keep important information
- Raw pixels → numerical descriptors



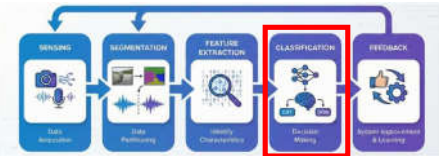
Example

- **RGB-D Camera**
 - Raw data: 3D joint positions
 - Extracted features: hand trajectory, joint angles, reach distance
- **Radar Sensor**
 - Raw data: reflected signal over time
 - Extracted features: velocity profile, presence & motion pattern, micro-movement

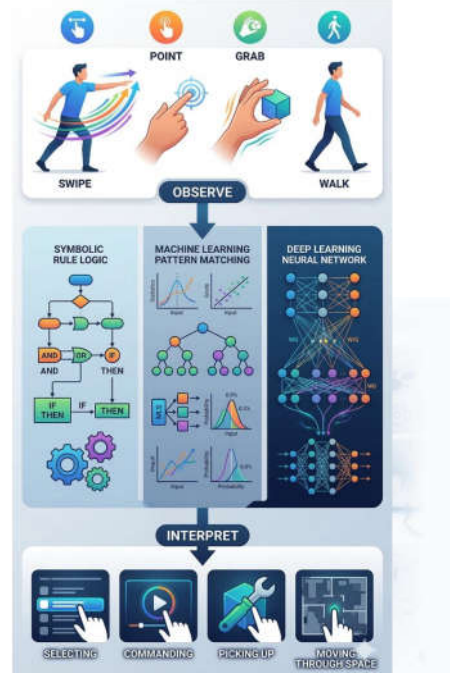


Recognition Pipeline

Classification

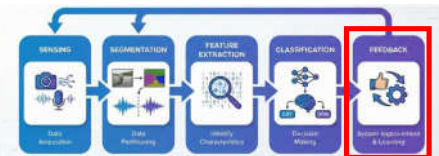


- Identify the meaning of the motions
 - E.g. Swipe, Point, Grab, Walk
- Methods
 - Rule-based
 - Machine learning
 - Deep learning

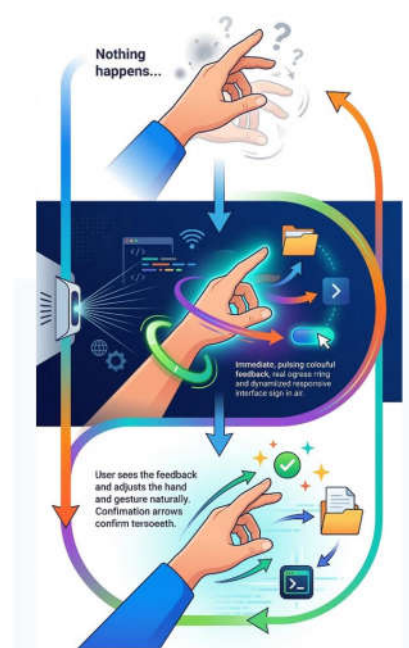


Recognition Pipeline

Feedback



- When the user performs a gesture but nothing happens, the user doesn't know why
- Close Loop Control
 - Movement → Sensing → Recognition → Feedback → User Adjustment
- Low-Latency Feedback
 - Attempt to recognize the gesture immediately, even before it's finished



Recognition Pipeline

Feedback

- **Visual Feedback**

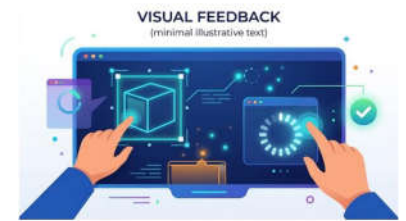
- E.g. highlight object, progress indicators

- **Auditory Confirmation**

- E.g. Tone on success, warning sound

- **Haptic Cues**

- E.g. Vibration on activation



Recognition Pipeline

Feedback & Interaction Loop

- Response when recognition failed

- **Diagnostic Feedback**

- Tell the user why it failed: “Too fast”, “Too far”

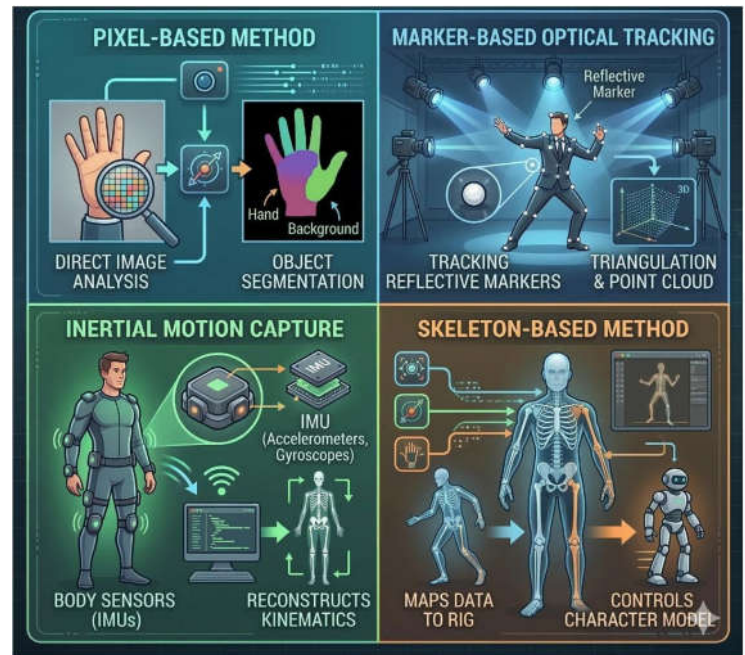
- **Confidence Scores**

- Execute command only if confidence $> 90\%$
- Otherwise show hint: “Low confidence — try again”



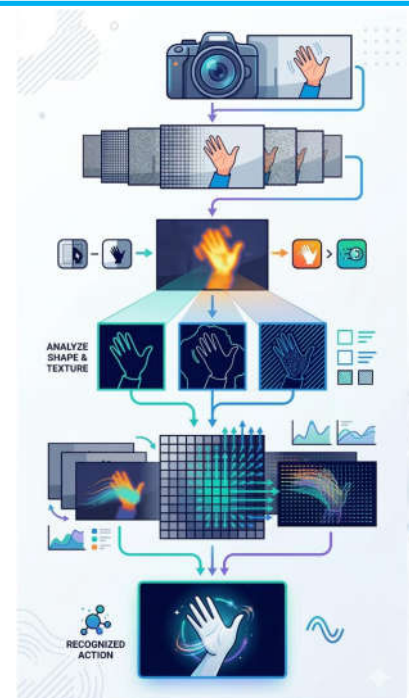
Common Methods

- Pixel-Based Method
- Marker-Based Optical Tracking
- Inertial Motion Capture
- Skeleton-based Method



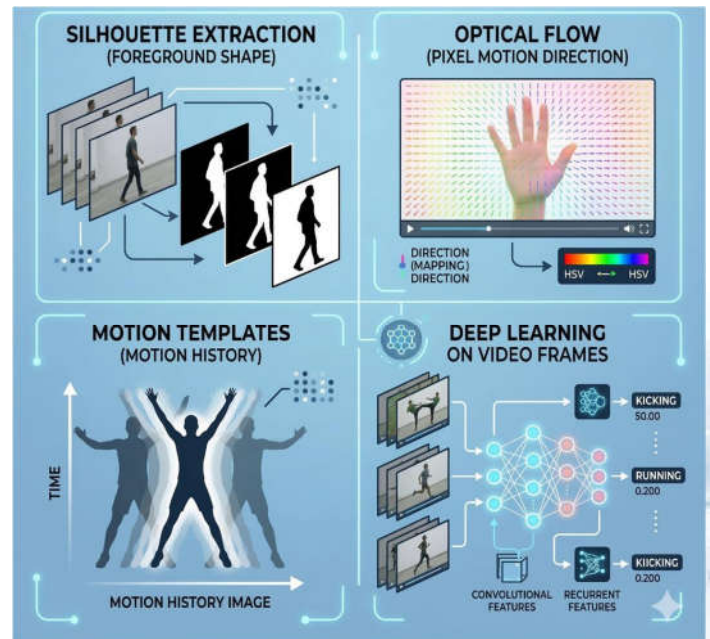
Pixel-Based Method

- Recognize motion **directly from image pixels**
- Process:
 - Detect motion regions
 - Analyze shape or texture change
 - Track pixel movement
- **Output:** **action** or **gesture** label



Typical Techniques

- **Silhouette Extraction**
(foreground shape)
- **Optical Flow**
(pixel motion direction)
- **Motion Templates**
(motion history)
- **Deep Learning on video frames**



49

Characteristics

- **Pros**
 - No body model required
 - Works with ordinary cameras
 - Captures full-body appearance cues
- **Cons**
 - Sensitive to lighting & background
 - Affected by clothing variation
 - Hard to understand precise joint motion

50

Applications

- **Best for**

- Activity recognition
- Surveillance behavior detection
- Crowd analysis

- **Not ideal for**

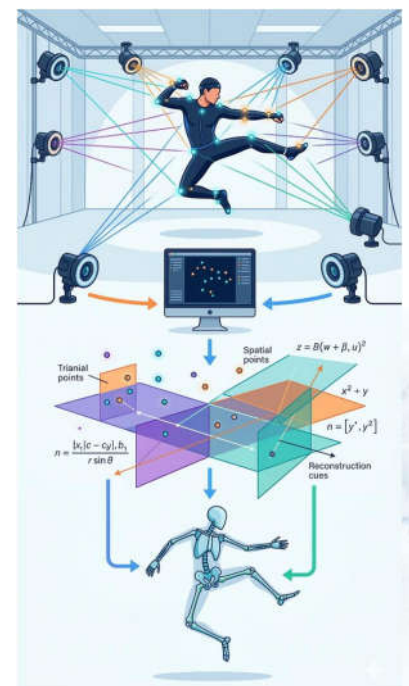
- Fine hand gesture control
- Precise interaction

Marker-Based Optical Tracking

- **Track reflective markers** attached to the body using **multiple cameras**

- **Process**

- Attach markers to **key body locations**
- Cameras **detect marker positions**
- **Triangulate 3D coordinates**
- **Reconstruct body motion**



Characteristics

• Pros

- **Gold standard precision**
- Reliable for biomechanics analysis
- Suitable for animation & sports science

• Cons

- Requires special suit & setup
- Expensive multi-camera system
- Not suitable for everyday environments

53

Applications

• Best for

- Film & game animation (highest motion realism)
- Clinical gait & rehabilitation analysis
- Professional sports biomechanics research
- Laboratory human movement studies

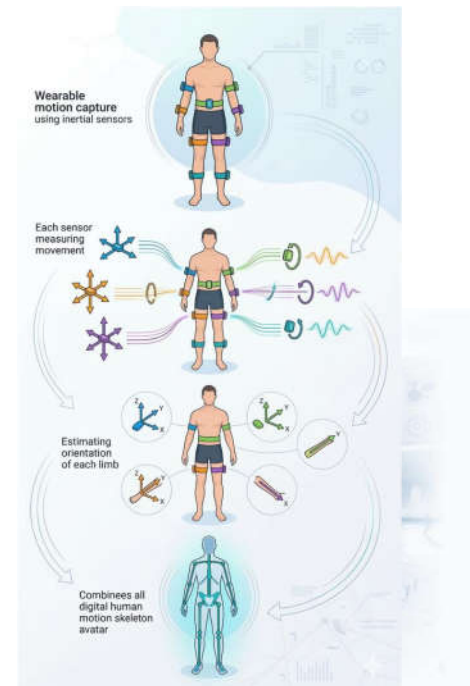
• Not ideal for

- Daily consumer interaction (too intrusive)
- Outdoor or large public spaces
- Mobile AR/VR use
- Quick setup environments (factories, homes)

54

Inertial Motion Capture

- Capture human motion using **wearable inertial sensors** (IMUs)
- Process
 - **Attach IMUs** to body segments
 - **Measure** acceleration & rotation
 - **Estimate orientation** of each limb
 - **Combine segments** into full-body motion



Characteristics

• Pros

- **Portable** and **mobile**
- Works **indoors** & **outdoors**
- **Low latency** response

• Cons

- **Position drift** over time
- Requires **calibration**
- Wearing equipment can be **inconvenient**

Applications

• Best for

- VR motion capture
- Outdoor sports analysis
- Rehabilitation monitoring
- Robotics teleoperation

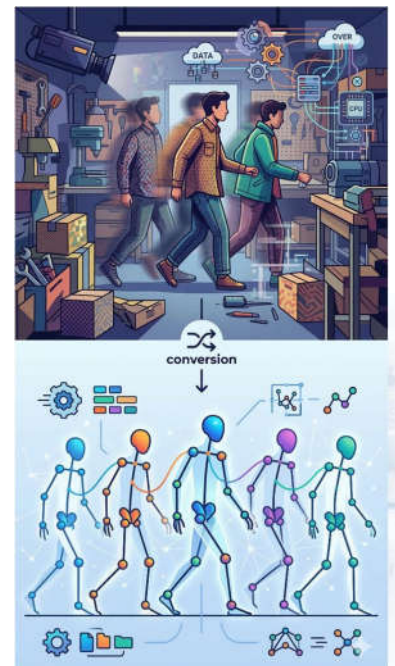
• Not ideal for

- High-precision position measurement
- Casual public interaction
- Short spontaneous use (setup required)

57

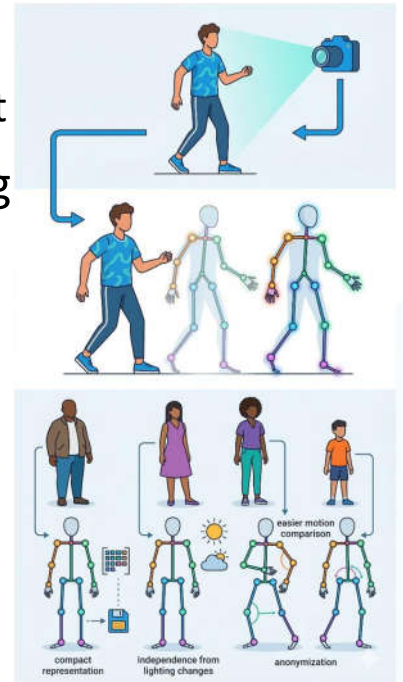
Problem with Raw Sensor Data

- **Images** contain **background**, **lighting** changes, **clothing** variation
 - Hard for algorithms to directly interpret motion
 - High computational cost
- **Skeleton** methods **remove visual complexity** so the system understands movement **structure**



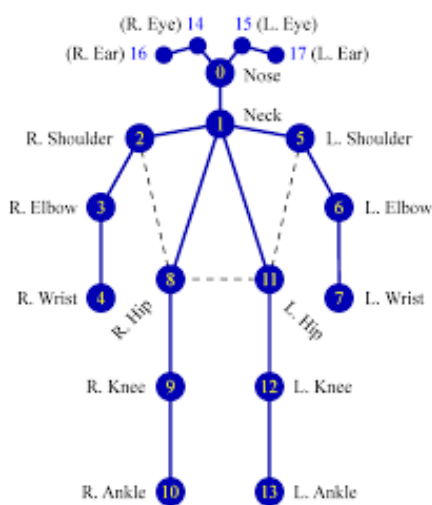
Skeleton-based Method

- **Convert** the human **body** into **key joints**
E.g head, shoulders, elbows, hands, hips, knees, feet
- A **skeleton** is **estimated** from sensor data using **pose estimation algorithms**
- **Input:** RGB images, Depth images, Multi-camera views, Motion sensors (IMU suits)
- **Output:** Set of body joints
head, shoulders, elbows, hands, hips, knees, feet

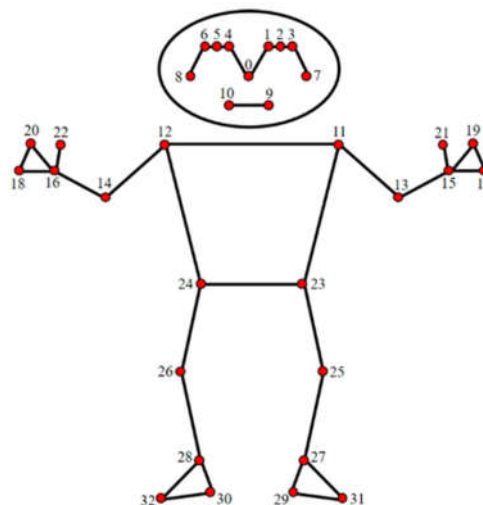


Recognition: Skeleton-based Method

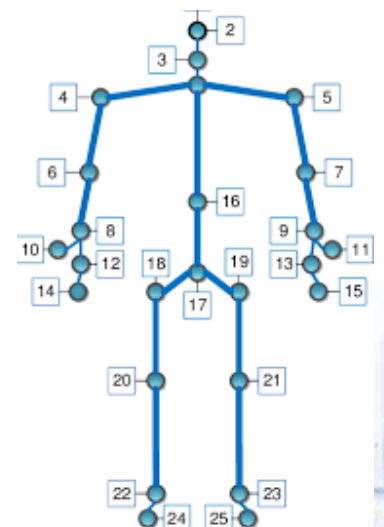
Standard Frameworks (Body)



COCO, Microsoft
(17 points)

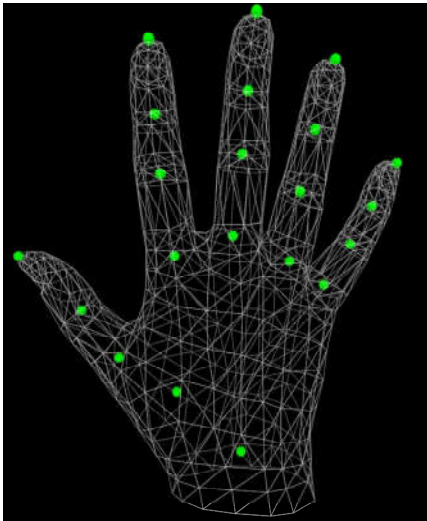


Media Pipe, Google
(33 points)

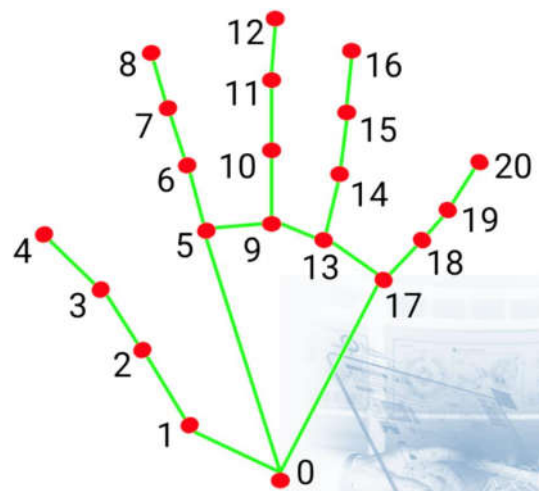


Kinect, Microsoft
(25 points)

Standard Frameworks (Hand)



Mano, Max Planck Institute, Germany
(21 keypoints / mesh)



MediaPipe, Google
(21 keypoints)

Joint Constraints

- Human **joints** have **limited Degrees of Freedom (DoF)**
 - **Hinge joints (knee, elbow):** 1-DoF, Flexion / extension only
 - **Ball-and-socket joints (shoulder, hip):** 3-DoF, Rotation in all directions
- The constraints **improve estimation**
 - E.g. Elbow cannot bend backward



Characteristics

• Pros

- Compact representation
- Robust to clothing & appearance
- Easier motion comparison
- Efficient for recognition algorithms

• Cons

- Sensitive to occlusion
- Reduced finger detail

63

Applications

• Best for

- Gesture interaction
- Human–robot interaction
- VR body tracking
- Action recognition

• Not ideal for

- Heavy crowd occlusion scenes
- Exact biomechanical measurement

64

Common Failure Cases

- **Self-Occlusion**

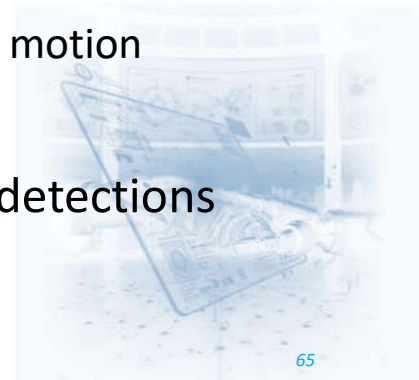
- Parts of a body **block each other**

- **Motion Blur**

- Moves **faster** than sensor **capture rate**
 - E.g. Typical cameras (~30 fps) cannot track fast motion

- **Environmental Noise**

- **Shadows** or patterned **clothing** create false detections



Dynamic Gesture

- **Static**: recognize a **single** pose

- **Dynamic**: recognize **motion over time**

- Extra Challenges

- **Temporal Segmentation**: Must detect **start and end** of motion (not only spatial location)
- **Speed Variation**: Different users move at **different speeds**
- **Variable Duration**: Same gesture may last **longer or shorter**



Pre-processing Methods

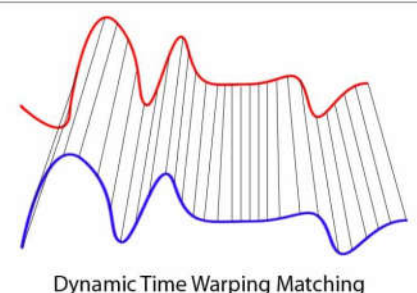
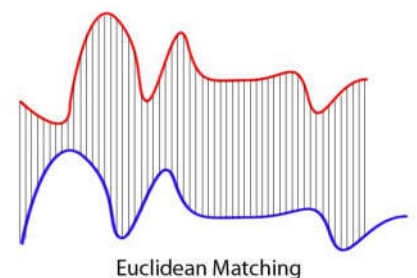
- **Normalization**
 - **Adjust time scale** so motions become comparable
 - **Align sequences** before classification
- **Fitting Variable-Length Data into Models**
 - **Zero-Padding**: Add empty frames to short motions
 - Simple but may add noise
 - **Interpolation**: Resample motion to fixed length
 - Smoother but may distort details



67

Trajectory-Based Recognition

- **Dynamic Time Warping (DTW)**
 - Motion sequence comparison method
 - Still works even if one motion is faster or slower
 - **Spatiotemporal Curves** are used
 - Show motion as a path over time
 - like a line moving through space
 - **Measure**
 - **Curvature**: how much the path bends
 - **Path length**: how far the motion travels



68

Generalization

- **Person-Independent Recognition**
 - Works across age, gender, body size, mobility
- **Speed Invariance**
 - Recognize gesture regardless of duration
Example: swipe in 0.2 s or 0.8 s
- **Background Noise Resistance**
 - Ignore other people or movements in the scene



Annotation Cost

- **Labeling** videos requires marking start/end of gestures
 - Extremely time-consuming and expensive at scale
- **Solution: Synthetic Data**
 - Generate motion using game engines (Unity / Unreal) or LLM
 - Produce large amounts of clean data



Latency

- **Human Perception Limit**

- 100 ms recognition delay → feels sluggish
- AR/VR target often <20 ms

- **Computation Budget**

- Deep networks improve accuracy
- But limited by mobile/embedded hardware power



Early Detection

- **Recognize** the gesture **before** it **finishes**

- Model continuously estimates gesture probability
- If confidence > 90% → trigger command early
- Example: detect “Circle” at ~75% completion

- Trade-off: **Accuracy** vs **Responsiveness**



Gesture Design

- Assign **meaning** to a **detected gesture**
 - e.g., "Rotate hand" -> "Adjust Volume"
- Simplest Rule on Mapping
 - **Distinguishability**
Gestures must be **easy to tell apart**
 - E.g. Two fingers ↑ vs Three fingers ↑



Gesture Design

Natural vs Learned Gestures

- **Natural (Intuitive)**
 - Mimicking physical world interactions.
 - **Benefit:** Zero learning curve
 - E.g. Turning a virtual knob to increase volume.
 - **Learned (Abstract)**
 - Symbolic movements that must be memorized.
 - **Benefit:** Highly efficient for "power users" once mastered
 - E.g. Drawing an "S" in the air to Save a file.
- **Trade-off:** "Ease of Entry" vs "Expert Efficiency"

Avoiding Overload & Fatigue

• Cognitive Load

- Keep the gesture set **small** and **intuitive**
 - Users can remember only 5–7 abstract gestures



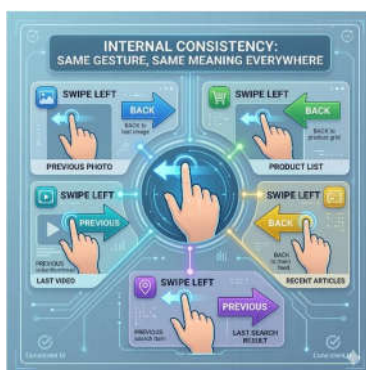
• Ergonomic Comfort

- Avoid high-effort movements (e.g., arms raised overhead)
 - Prevent fatigue during repeated use
- **Relaxation Zone:** Prefer gestures near the body
 - E.g. Elbows tucked or resting on a surface

Consistency and Memorability

• Internal Consistency

- Same Gesture, Same Meaning everywhere
- E.g. swipe left = back



• Metaphorical Mapping

- Use familiar real-world actions
- E.g. tossing an item



Gesture Design Inclusivity

• Cultural Nuance

- Some gestures carry different meanings across regions



• Body Diversity

- Users vary in arm length, hand size, and mobility

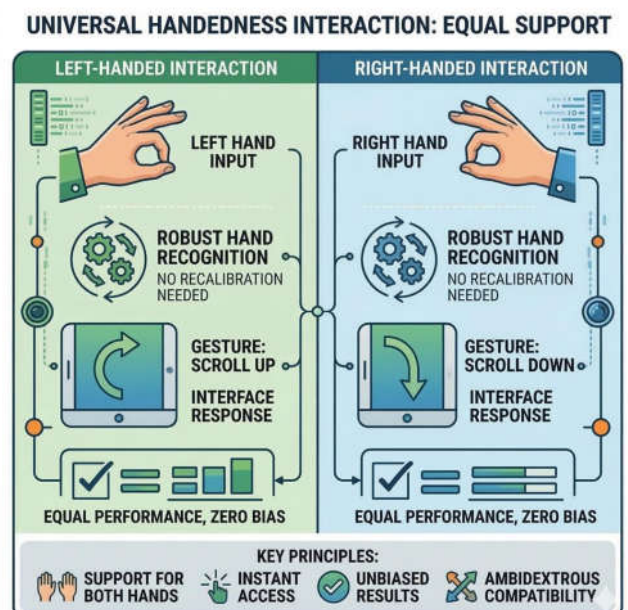


77

Gesture Design Inclusivity

• Handedness

- Support both left- and right-handed interaction
- No recalibration or performance bias



78

Gesture Design

Mapping

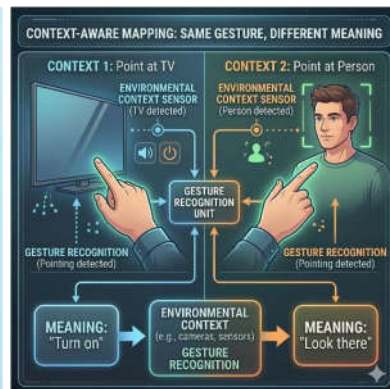
• One-to-One Mapping

- One gesture, one meaning



• Context-Aware Mapping

- Same gesture, different meaning
- Combine gesture recognition with environmental context
- Example: Pointing
 - Point at TV → “Turn on”
 - Point at person → “Look there”



79

Unintentional Movements

• Passive vs Active Movement

- **Gesticulation**: natural hand motion while talking
- **Manipulation**: intentional command to the system

• Accidental Trigger

- E.g. Reaching for a coffee cup triggers “Delete”

• Thresholding Techniques

- **Dwell time**: action must be held briefly
- **Minimum velocity**: ignore slow casual motion



Confirmation

- **Two-Step Activation: Prime → Execute**
 - Engagement gesture activates system (e.g., wake gesture)
 - User performs actual command
- **Critical Action Barriers**
 - For high-risk commands (Erase Data / Shut Down), require hold gesture (~3 s)
 - “Undo” Gesture



Context Gating

- **Gaze-Gating**
 - Accept gestures only when user is looking at the interface
 - Combines eye-tracking + gesture sensing
- **Proximity Gating**
 - Enable interaction only within a “sweet spot” distance
Example: 0.5 m – 1.5 m
- **State Awareness**
 - Disable certain gestures in specific modes
Example: no “Delete” in View-Only mode



Fail-Safe Interaction Protocols

- **“Freeze” Protocol**

- If tracking is lost (occlusion / sensor failure)
 - Keep last safe state
 - Do NOT guess the user’s intent

- **Emergency Stop**

- Universal high-priority gesture (e.g., crossed-arms “X”)
- Overrides all commands immediately

