README.md

Ubiquitous Computing

Table of Contents

- Ubiquitous Computing
 - Table of Contents
- Lecture 1 on Ubiquitous Computing (UC)
 - Introduction to Ubiquity
 - What is Ubiquitous Computing (UC)?
 - Key Features of UC 🥄
 - The Three Eras of Modern Computing
 - 1 Mainframe Era (1950s-1970s)
 - 2 Personal Computing Era (1980s-Present) 🖳
 - B Ubiquitous Computing Era (Present-Future) •
 - The Vision of Ubiquitous Computing
 - The Current State of UC
 - 1. Smart Mattresses
 - 12. Taste Manipulation with Temperature
 - •• 3. Gaze Tracking for Online Meetings
 - 3 4. Silent Speech Reading
 - **4** 5. Smart Cushions for Relaxation
 - Windows Computing in the Industry
 - Mean of the second of the secon
 - → Example: Fall Detection in Smartphones
 - - 1 Healthcare
 - 2 Smart Cities
 - 3 Retail 🛒
 - 4 Automotive ____
 - 5 Education
- Subjuitous Computing Course Plan (DES535)
 - Course Schedule Breakdown
 - January: Laying the Foundation
 - 1 Week 1-2: Introduction to Ubiquitous Computing
 - Week 3-4: Aspects of Ubiquitous Computing
 - February: Context Awareness & Location-Based Computing
 - Week 5-6: Ambient & Context-Aware Computing
 - 4 Week 7: Project Discussion & Formulation
 - March: Physiological Sensing & Human Behavior Analysis
 - 6 Week 10-11: Motion & Activity Sensing
 - Week 12-13: Physiological Sensing & Biometric Computing
 - P April: Affective Computing, Smart Systems & Final Project Work
 - 8 Week 14-15: Affective Computing (Emotion-Aware Systems)

- 9 Week 16-17: Wearable Computing & Smart Systems
- May: Final Examinations & Project Presentations
 - Week 18-19: Project Demonstrations & End-Semester Exams
- @ Conclusion
- Secture 2: Ubiquitous Computing (UC)
 - Core Characteristics of Ubiquitous Computing (UC) Systems
 - 11 Computers Need to Be Networked, Distributed & Transparently Accessible ◆
 - U Human-Computer Interaction (HCI) Should Be More Implicit than Explicit
 - ③ Context-Awareness: Systems Should Adapt to Their Environment 🏡
 - 4 Computers Should Operate Autonomously with Self-Governance ❖
 - Is Handling Multiple Dynamic Interactions via Intelligent Decision-Making
 - Prontext-Awareness in UC: The "Five W's" Framework
 - 1 WHO? (Identity) D
 - 2 WHAT? (User Activity) 🏃
 - 3 WHERE? (Location) •
 - 4 WHEN? (Time) X
 - 🖪 WHY? (Reason) 🤔
 - Farly and Modern Examples of Context-Awareness
 - 1 Georgia Tech's Aware Home Project 🏡
 - 2 Modern Context-Aware Applications
 - Additional UC-Related Concepts
 - 1 Calm Technology 🧘
 - 2 Invisibility in UC 🌫
 - Final Thoughts
- Secture 3: Ubiquitous Computing (UC) Wearable Sensors & Privacy
 - Q Overview of the Lecture
 - La Wearable Sensors: Enhancing Human Capabilities
 - Property Types of Wearable Sensors
 - Motion Sensors @ (Tracking Movement & Orientation)
 - ② Bioelectric Sensors 🗲 (Monitoring Body Signals)
 - **③** Biometric Sensors **½** (Health & Performance Tracking)
 - ■ Environmental Sensors (External Conditions Monitoring)
 - ⑤ Optical & Chemical Sensors / (Monitoring Body Fluids & Blood Circulation)
 - Applications of Wearable Sensors
 - 1 Healthcare & Medical Monitoring
 - 2 Wellness & Fitness ½
 - 3 Smart Fashion
 - 4 Business & Security Representation
 - A Privacy Concerns in Ubiquitous Computing
 - P Definition of Privacy (Alan Westin)
 - Privacy in UC
 - Porders of Privacy Breaches
 - ∘ **X** Activity: Building a Wearable Sensor Data App
 - © Conclusion: The Future of Wearable Sensors & Privacy
- Secture 4: Privacy in Ubiquitous Computing (UC)

- Privacy in Ubiquitous Computing
 - What is Privacy?
 - ¶ The Fundamental Problem
- Solove's Privacy Taxonomy
 - ¶ 1. Information Collection 📡 (How Data is Collected)
 - ? 2. Information Processing (How Data is Stored & Analyzed)
 - ¶ 3. Information Dissemination 💣 (How Data is Shared)
 - ¶ 4. Invasion of Privacy 🗗 (Unwanted Intrusions)
- O People Care About Privacy?
- Framework for Designing Fair Ubiquitous Computing Systems
- @ Conclusion: Privacy in Ubiquitous Computing
- ► Lecture 5: Ambient & Context-Aware Computing in Ubiquitous Computing (UC) ○ Overview of the Lecture
 - Context-Aware Computing in UC
 - What is Context-Aware Computing?
 - - What is Context Modeling?
 - Context-Aware Logic Space
 - Punderstanding Logic in Context-Aware Applications
 - * Requirements of Context-Aware Applications
 - Q Key Functional Requirements

 - 2 Context Aggregation (Data Processing & Storage)
 - ③ Context Consistency ✓ (Ensuring Data Accuracy)

 - 5 Context Querying & Adaptation in (Real-Time Decision Making)
 - **6** Context Reasoning (AI-Driven Insights)
 - 🔬 Demonstration: Sensing Context with Arduino Sensors
 - Il Ultrasonic Proximity Sensor \(\sqrt{\sq}}}}}}}}}}}}} \signtimes\sintititex{\sintity}}}}}}}} \endittinmides\sintititith{\sintitta}\sintititit{\sintitta}}}}}}}} \endittinmides\sintititititit{\sintititit{\sintititit{\sintititit{\sint{\sintiin}}}}}}}} \endittinmides\sintitititititititititititititit
 - 2 Soil Moisture Sensor 🌱
 - 3 Proximity Sensor Q
- Sensors, Capacitive Sensing & Context-Aware Computing in UC
 - Q Overview of the Lecture
 - - P Three Major Types of Sensors in Context-Aware Computing
 - 1 Physical Sensors
 - 2 Virtual Sensors 🖳
 - B Logical Sensors
 - Ш Hierarchy of Context Representation
 - Q Context Data is Processed in Different Layers:
 - The Architecture of Context-Aware Systems
 - Rey Components of a Context-Aware System
 - Wall++: Room-Scale Interactive & Context-Aware Sensing
 - Mall++ Works?

- \(\neq \) Capacitive Sensing: The Foundation of Touch Screens
 - What is Capacitive Sensing?
 - How It Works?
 - 2 Evolution of Capacitive Sensing: SmartSkin (2002)
- Applications of Capacitive Sensing
- * Demonstrations: Implementing Capacitive Sensing with Arduino & MIT App Inventor
 - I Demonstration: Wall++ (Capacitive Paint & Sensors) 🏡
 - 2 Activity: Creating a Mobile App for Capacitive Touch Sensing
- Sensing in Ubiquitous Computing (UC)
 - Q Overview of the Lecture
 - Case Study 1: SwitchBack Eye-Tracking for Attention Guidance
 - Motivation: The Problem of Divided Attention in Mobile Use
 - **%** SwitchBack: The Solution
 - How SwitchBack Works: Eye-Tracking Mechanism
 - Case Study 2: RadarFoot mmWave Sensing for Smart Shoes
 - Motivation: Enhancing Ground Surface Context Awareness
 - mmWave Sensing Technology: How It Works
 - **%** How mmWave Radar Detects Ground Surfaces
 - mmWave Radar: Signal Processing & Object Detection
 - Applications of mmWave Sensing
 - The Gait Cycle: Understanding Walking Patterns
 - * Activity: Implementing Capacitive Sensing in a Mobile App
 - © Conclusion: The Future of Eye-Tracking & mmWave in UC
- In-Depth Analysis of FMCW mmWave Radar Sensing
 - Q Overview of the Document
 - Introduction to mmWave Sensing: FMCW Radars
 - Fundamentals of FMCW Radar Operation
 - What is a Chirp?
 - How FMCW Radar Measures Range
 - Step-by-Step Process
 - Calculating Object Distance
 - III Range Resolution in FMCW Radar
 - Measuring Multiple Objects Using Fourier Transforms
 - Welocity Measurement in FMCW Radar
 - 1 Doppler Shift in Radar
 - Measuring Velocity Using Multiple Chirps
 - Angle Estimation (Angle of Arrival AoA)
 - Key Formula:
 - © Conclusion: The Future of FMCW mmWave Radar
- Lecture 8: RadarFoot Fine-Grain Ground Surface Context Awareness for Smart Shoes
 - Q Overview of the Lecture
 - The Gait Cycle & Smart Shoe Sensing
 - Punderstanding the Gait Cycle
 - Mean How mmWave RadarFoot Works
 - P Sensing Principle of RadarFoot

- Mhy Traditional Sensors (IMUs) Are Not Enough
- III Feature Extraction & Machine Learning for Surface Classification
 - P How Machine Learning Improves Surface Detection
- © Conclusion: The Future of Smart Shoes with mmWave Sensing
- P Lecture 9: Location Sensing in Ubiquitous Computing (UC)
 - Q Overview of the Lecture
 - Location Representation in Ubiquitous Computing
 - 1 Physical vs. Symbolic Locations
 - 2 Absolute vs. Relative Locations
 - Location Sensing Techniques
 - 1 Proximity Sensing
 - 2 Trilateration \u224
 - 🖪 Hyperbolic Lateration 📡
 - 4 Triangulation **>**
 - **5** Fingerprinting **4**
 - 6 Dead Reckoning 1
 - 🔽 Scene Analysis 🕏
 - M Revisiting Applications of Location Sensing
- Lecture 10: Location Sensing Systems in Ubiquitous Computing (UC)
 - Q Overview of the Lecture
 - Global Positioning System (GPS)
 - What is GPS?

 - Active Bat: Ultrasonic-Based Indoor Location Tracking
 - Cricket: RF + Ultrasound Hybrid Localization
 - Graded Activity: Implementing Trilateration in MIT App Inventor
 - Step 1: Create 4 Fixed Reference Points
 - Step 2: Allow Users to Select a Point on the Screen
 - Step 3: Compute Distances
 - P Step 4: Find Intersection Point
- P Lecture 11: Queries & Models in Location Sensing (UC)
 - Q Overview of the Lecture
 - Queries to Location Models
 - ¬ ¬ Types of Queries in Location Models
 - 1 Position Queries 📌
 - Nearest Neighbor Queries Q
 - B Range Queries
 - A Navigation & Road Topology in Location Models
 - Ш Requirements for Location Models
 - P Case Study: Microsoft GeoLife Dataset
 - ∘ **Ⅲ** Real-World Applications of Location Queries
- P Lecture 12: Mining Location Histories & Travel Sequences in Ubiquitous Computing (UC)
 - Q Overview of the Lecture
 - M Extracting Interesting Locations from GPS Trajectories

- Understanding GPS Logs & Stay Point Detection
 - What is a Stay Point?
- P Location History Modeling with Hierarchical Graphs
- Q HITS-Based Inference for Travel Patterns
- Ш Mining Classical Travel Sequences
- Personalized Travel Recommendations
- Graded Activity: Designing a UC System for Clinical Leg Injury Patients
- © Conclusion: The Future of GPS-Based Location Sensing
- Lecture 13: Motion & Activity Sensing in Ubiquitous Computing (UC)
 - Q Overview of the Lecture
 - Understanding Accelerometers & Earth's Gravity (g)
 - Proper Acceleration vs. Coordinate Acceleration
 - **X** Types of Accelerometers
 - 1 Capacitive Accelerometers
 - Piezoelectric Accelerometers
 - 3 Piezoresistive Accelerometers
 - In the second of the second of
 - Magnetoresistive Accelerometers

 - Applications of Accelerometers
 - Syroscopes: Measuring Rotational Motion
- Lecture 14: Motion & Activity Sensing in Ubiquitous Computing (UC)
 - Q Overview of the Lecture
 - G Gyroscopes: Measuring Orientation & Angular Velocity
 - Mechanical Gyroscopes
 - NEMS Gyroscopes & the Coriolis Effect
 - ∘ III Orientation Axes: Roll, Pitch & Yaw
 - man Activity Recognition (HAR)
 - Key Characteristics of HAR Systems
 - O Challenges in HAR
 - Sensor Modalities for HAR
 - S Activity Recognition Chain (ARC)
 - Applications & Concrete Examples
 - Seferences for Further Reading
- Lecture 15: Motion & Activity Sensing in Ubiquitous Computing (UC) Module V (Part II)
 - Q Overview of the Lecture
 - 🔟 UCI-HAR Dataset: Experimental Design & Feature Extraction
 - Servation & Drawback

 - WiFi Sensing Fundamentals
 - | WiFi-Based HAR & Fine-Grained Movement Detection
 - SelectionReferences for Further Reading
- Lecture 16: Introduction to WiFi-Based Motion & Activity Sensing
 - Q Overview of the Lecture
 - MiFi Channels & Propagation

- Q Channel State Information (CSI)
- © Enabling Technologies: MIMO & OFDM
 - 1 MIMO (Multiple Input Multiple Output)
 - ② OFDM (Orthogonal Frequency Division Multiplexing)
- Properties of the Comparison with Video-Based Sensing

- \begin{align*}
 \text{ Yes Takeaways & Future Directions.}
- • Lecture 17: Physiological Sensing in Ubiquitous Computing (UC) Module VI (Part I)
 - Q Overview of the Lecture
 - 1. What Are Physiological Signals?
 - ∘ ∠ 2. How to Record Physiological Signals?
 - Wearables & Devices
 - \$ 3. Applications of Physiological Sensing
 - - 4.1 Principle of PPG
 - 4.2 PPG Probe Types
 - 4.3 Arterial Waveform Structure & Derived Metrics
 - 4.4 Measuring HR vs. BP: Parameter Guidelines
 - III 5. Case Study: OneMind—Detecting Divided Attention in Mobile MOOC Learning
 - Problem Context
 - OneMind Contribution
 - Solution Pipeline
 - Key Results
 - Strain Class Activity: Critical Evaluation
- 📡 Lecture 18: Electrodermal Activity (EDA) in Physiological Sensing
 - Q Overview of the Lecture
 - 1. What Is Electrodermal Activity?
 - © 2. Measuring EDA: Exosomatic vs. Endosomatic
 - 🗠 3. EDA Signal Structure: Tonic & Phasic Components
 - **1** 4. Key EDA Parameters
 - Applications of EDA
 - **a** 6. Commonly Used Devices & Confounds
 - 7. Psychological Constructs Underpinning EDA
 - ♠ 8. Threats to Validity
- 🞽 Lecture 19: Wearable Computing & Smart Systems
 - Q Overview of the Lecture
 - © 1. Key Attributes of Wearable Computing
 - 2. Smart Clothing: Scenario & Challenges
 - 🗵 3. SeamFit: A Seam-Based Smart Clothing Solution
 - o 🕶 4. Early Prototypes: From Wearable Multimedia to Modern Smartwear
 - ⊕ 5. Broader Applications of Wearable Computing
 - 🛕 6. Fundamental Challenges of Wearable Systems
- — Lecture 20: Electroencephalography (EEG) in Physiological Sensing Module VI (Part II)
 - Q Overview of the Lecture

- ⊕ 2. EEG Signal Generation & Noise
- 3. EEG Frequency Bands & Cognitive States

- 4 6. Graded Activity: Designing a Distraction-Monitoring EEG Device
- * Lecture 21: Affective Computing Module VII (Part I)
 - Q Overview of the Lecture
 - man 1. What Is Affective Computing?
 - \$2. Emotions: Cognitive vs. Physical

 - - 4.1 Facial Expressions 😂 😂 🖼 😯
 - 4.2 Vocal Intonation
 - 4.3 Motor Forms & Essentic Gestures
 - 4.4 Physiological Responses 💵
 - 5. Designing Emotion-Aware UC Systems

 - J 7. Human Expressions & Facial Action Coding (FACS)
 - \$\forall 8\$. Case Study: ExpresSense
 - 8.1 System Architecture
 - 8.2 Experimental Setup & Results
 - 8.3 Usability Study
 - 9. Opportunities & Challenges
- • EEGlass: Toward Everyday EEG-Eyewear for Brain-Computer Interaction
 - Context & Motivation
 - * Prototype Design & Hardware
 - Signal Acquisition & Processing
 - Performance & Findings
 - \mathbb{R} Strengths & Innovations
 - A Limitations & Challenges
 - Puture Directions

Lecture 1 on Ubiquitous Computing (UC)



Introduction to Ubiquity

The concept of **ubiquity** refers to the state of being **present everywhere** or being **very common**. In computing, it means integrating technology so seamlessly into daily life that users hardly notice it.

• Think about it:

- Do you ever consciously think about streetlights, Wi-Fi networks, or voice assistants like Alexa? 🤖
- These technologies blend into our routines and function without demanding active attention.

Definition of Ubiquity:

"The fact of appearing everywhere or of being very common."



What is Ubiquitous Computing (UC)?

Mark Weiser, the father of UC, stated:

The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it."

This means that computing should **blend naturally** into the environment, operating in the background while enhancing user experience.

Key Features of UC Q

- Always Present but Unnoticed Devices work in the background.
- Minimal Cognitive Load 🧠 The user doesn't need to think about technology to use it.

Example:

- Traditional computing requires users to sit at a **desktop PC** and perform tasks.
- UC enables **smart assistants (Siri, Google Assistant)** to execute commands based on **voice recognition**, making it hands-free.

📜 The Three Eras of Modern Computing 🏛

Computing has **evolved** significantly over the decades, moving toward **Ubiquitous Computing**:

- **1** Mainframe Era (1950s-1970s)
 - · Few computers, many users.
 - Large, expensive machines used in government and research centers.
 - Example: IBM 360 Mainframe

2 Personal Computing Era (1980s-Present) 🖳

- One computer per person.
- Laptops, desktops, and smartphones became common.
- Example: Windows PCs, MacBooks, and iPhones

☑ Ubiquitous Computing Era (Present-Future) ◆

- Multiple devices per person, seamlessly integrated.
- IoT devices, Al assistants, and smart environments.
- Example: Google Home, Apple HomeKit, Smartwatches 🔮
- UC is the next step, where computing "disappears" into daily life.

🔮 The Vision of Ubiquitous Computing

To understand how UC enhances daily life, students were given a **thought experiment**:

- Example: Waking Up in a Smart Home
- Current Scenario:
 - Wake up to a **loud alarm clock** 💆.
 - Walk to the kitchen to make coffee .

UC-Enhanced Scenario:

- A **smart mattress detects body movement** and wakes you up naturally.

This **context-aware computing** improves comfort while minimizing effort. **

The Current State of UC

UC is not just a **theoretical concept**—it is already influencing various industries!

1. Smart Mattresses

Example: Sleep Number 360 Smart Bed

- Monitors sleep patterns and adjusts firmness automatically.
- Wakes you up **gently** with vibrations instead of alarms.

■ 2. Taste Manipulation with Temperature

- Research shows that altering **lip temperature** changes taste perception. ③
- Future Applications: Personalized dining experiences.

90 3. Gaze Tracking for Online Meetings

- Al detects where users are **looking .**
- Ensures better engagement in virtual calls.
- Example: Al-powered **Zoom eye-contact correction**.

🤫 4. Silent Speech Reading

- Uses muscle movement tracking to detect words without speaking.
- Could help in military communication or silent AI assistants.

■ 5. Smart Cushions for Relaxation

- **Detects posture** and adjusts seating position for comfort.
- Example: Posture-correcting chairs for offices.

Ubiquitous Computing in the Industry

Big tech companies are investing heavily in **UC-powered smart solutions**:

- **♠ Google Nest** Adjusts home temperature based on user behavior.
- Apple HomeKit Controls smart devices through Siri voice commands.
- Google Fit Tracks fitness data from wearables.
- Empatica Smartwatch Monitors stress and sleep patterns in real time.

Q Why it matters?

- These systems allow for hands-free automation.
- Users get **seamless experiences** without directly interacting with devices.

How UC Works: Processing & Components

A **Ubiquitous Computing System** consists of four **key components**:

♦ Component	Q Function
[★] Humans/Animals	End-users interacting with technology.
Sensing & Data Collection	Devices that monitor user behavior.
♦ Connectivity & Context Awareness	Al that interprets user data.
■ Objects & Computing Devices	Smart gadgets, sensors, and wearables.

Example: Fall Detection in Smartphones

A **simple algorithm** can detect falls using an **accelerometer**:

lıl Formula:

- If sudden **X, Y, Z-axis changes** exceed a threshold, classify as **FALL**.
- Otherwise, classify as **normal movement**.

Real-life use cases:

• Smartwatches like Apple Watch 📱 detect falls and alert emergency contacts! 🚨



Innovations in UC

The **future** of UC will shape multiple industries:

🔟 Healthcare 🏥

- Wearables track **blood pressure** and **ECG data** in real time.
- Smart insulin delivery for diabetics.

2 Smart Cities

- Al-driven traffic lights that reduce congestion.
- Sensors adjust streetlights based on real-time data.

■ Retail ■

- Amazon Go stores use UC for checkout-free shopping!
- Al cameras track items without a cashier.

4 Automotive

- Self-driving cars rely on UC to navigate safely.
- Tesla's Autopilot system adjusts driving based on real-world data.

⑤ Education **⋄**

- Al-powered **personalized learning** adapts to student needs.
- Smart whiteboards that record lectures for remote access.
- UC will continue shaping the way we live and work!

© Conclusion

Ubiquitous Computing is revolutionizing our relationship with **technology**. The **ultimate goal** is to **embed technology seamlessly** into our lives so that it feels natural and intuitive.

- 📌 Key Takeaways: 🔽 UC enables hands-free, context-aware computing.
- We are moving from personal computers to integrated smart systems.
- Industry leaders like Google, Apple, and Amazon are shaping the UC future.
- ▼ Future innovations will impact healthcare, retail, education, and cities.

💡 Final Thought:

As Mark Weiser said, "The most advanced technology is the one we no longer notice—because it simply works."



Ubiquitous Computing Course Plan (DES535)

The Course Plan for DES535: Ubiquitous Computing outlines the structured progression of the subject, covering fundamental concepts, context-awareness, sensing mechanisms, physiological computing, affective computing, and smart systems. The schedule integrates theoretical learning, hands-on experimentation, assignments, and project work to provide students with a comprehensive understanding of Ubiquitous Computing (UC).

Course Schedule Breakdown

The course spans from **January to May**, divided into **four months of learning** and concluding with **end-semester examinations**. The structure ensures **progressive complexity**, introducing fundamental concepts first and then diving into advanced topics.

9

January: Laying the Foundation

1 Week 1-2: Introduction to Ubiquitous Computing

★ Key Topics:

- Understanding **what UC** is and its historical background.
- Exploring Mark Weiser's Vision of computing seamlessly blending into daily life.
- Identifying real-world applications such as smart homes, wearable tech, and AI-driven automation.

Why is this important?

This introduction provides students with the necessary background to appreciate **how computing has** evolved from mainframes to personal computing and now to Ubiquitous Computing.

2 Week 3-4: Aspects of Ubiquitous Computing

★ Key Topics:

- Studying pervasive computing principles.
- Understanding interaction models between humans, environments, and smart devices.
- Introduction to privacy and security challenges in UC.

📌 Examples:

- Google Assistant and Amazon Alexa Smart assistants that integrate seamlessly into daily activities.
- IoT-based automation Smart thermostats like Nest that adapt to user behavior.

💡 Takeaway:

Students start grasping how UC systems need to balance intelligence with user privacy and convenience.

February: Context Awareness & Location-Based Computing

3 Week 5-6: Ambient & Context-Aware Computing

📌 Key Topics:

- What is **Context Awareness**?
- How devices gather, process, and utilize user context to make intelligent decisions.
- Types of context data Location, time, user activity, emotions, and environmental factors.
- Real-life Applications Smart homes, healthcare monitoring, and adaptive UI/UX.

📌 Example:

- Smart lights that adjust brightness based on ambient conditions and user presence.
- Google Maps predicting traffic conditions using real-time data from smartphones.

Pakeaway:

This section builds the foundation for **designing adaptive and responsive UC systems**.

4 Week 7: Project Discussion & Formulation

Focus:

- Students form **groups of 3-4** and brainstorm potential UC applications.
- Initial **project ideas** are discussed, and feasibility is evaluated.
- Hands-on experimentation begins with context-aware computing applications.

💡 Takeaway:

This marks the shift from **theory to practical implementation**, ensuring students apply their knowledge to real-world use cases.

5 Week 8-9: Location Sensing & Activity Monitoring

★ Key Topics:

- GPS-based location sensing.
- Indoor Positioning Systems (IPS) Wi-Fi, Bluetooth, Infrared-based positioning.
- Motion detection using accelerometers & gyroscopes.
- Human Activity Recognition (HAR) through sensors.

★ Examples:

- Fitness apps like Google Fit & Apple Health tracking steps using smartphone accelerometers.
- Location-based reminders & geofencing (e.g., "Remind me to buy milk when I reach the grocery store").
- Fall detection in Apple Watch Recognizing sudden movement changes to detect accidents.

Practical Experimentation:

• Students demonstrate real-world systems using location sensing in IoT environments.

Pakeaway:

This section enables students to design **location-aware applications** that provide **personalized services based on movement & positioning**.

March: Physiological Sensing & Human Behavior Analysis

6 Week 10-11: Motion & Activity Sensing

Key Topics:

- Analyzing motion data from sensors.
- Gesture-based control systems in UC.
- Wearable technology and its role in activity tracking.

📌 Examples:

- Microsoft Kinect & Al cameras recognizing body movements.
- Gesture-controlled smart TVs & gaming consoles (e.g., Nintendo Wii, Xbox Kinect).
- VR Headsets tracking user movements for immersive experiences.

Pakeaway:

Students learn how motion-based interactions can redefine human-computer interaction in everyday life.

Week 12-13: Physiological Sensing & Biometric Computing

★ Key Topics:

- Monitoring physiological signals like heart rate, EEG, and muscle activity.
- **Biometric authentication** in UC (e.g., fingerprint, facial recognition).
- Emotion detection using Al-driven sensors.

Examples:

- Apple Watch ECG feature detecting heart rhythm anomalies.
- Smart glasses with AI eye-tracking to detect emotions.

🔬 Hands-on Experiment:

• Students experiment with physiological sensing using wearables & biometric sensors.

💡 Takeaway:

This section emphasizes how **Ubiquitous Computing can revolutionize healthcare and security through biometric data analysis.**

📍 April: Affective Computing, Smart Systems & Final Project Work

8 Week 14-15: Affective Computing (Emotion-Aware Systems)

★ Key Topics:

- Emotion recognition using AI & sensors.
- Applications of mood-adaptive computing.
- Challenges in affective computing.

★ Examples:

- Al detecting customer emotions in retail (Amazon Al cameras).
- Spotify & Netflix recommendations based on mood.
- Mental health monitoring through wearable EEG sensors.

<u>★</u> Experimentation:

• Building emotion-classification systems using facial expressions, voice tone, and body language.

Pakeaway:

Students learn how AI can enhance user experience by making devices emotionally aware.

19 Week 16-17: Wearable Computing & Smart Systems

📌 Key Topics:

- Integration of smart wearables with IoT.
- Al-powered decision-making in smart systems.
- · Case studies of modern smart systems.

***** Examples:

- Smart clothing with embedded health sensors.
- Empatica Smartwatch detecting stress levels.
- · Self-adjusting climate control in smart buildings.

Experimentation:

· Hands-on implementation of smart wearable prototypes.

Pakeaway:

This section deepens the understanding of wearable technology's role in real-world Ubiquitous Computing applications.

May: Final Examinations & Project Presentations

10 Week 18-19: Project Demonstrations & End-Semester Exams

Key Activities:

- Final project submission & presentations showcasing UC-based innovations.
- End-semester exams assessing theoretical and practical knowledge.

Examples of Potential Projects:

- Smart sleep tracking system.
- AI-based gesture control interface.
- Emotion-aware virtual assistants.

💡 Final Takeaway:

The culmination of this course ensures that students leave with:

- A strong grasp of UC principles.
- Hands-on experience in sensors, IoT, and AI-based applications.
- The ability to design & implement real-world UC solutions.

© Conclusion

The **DES535: Ubiquitous Computing course** is **well-structured** to gradually build **knowledge, hands-on experience, and problem-solving abilities** in modern computing paradigms.

📌 Key Highlights:

- ✓ Theoretical foundation in UC principles.
- ✔ Practical projects involving motion, location, and biometric sensing.
- ✓ Real-world applications of AI, IoT, and wearables.
- ✓ Final project allowing students to showcase innovative UC applications.

Looking Ahead:

With Ubiquitous Computing shaping **smart cities, healthcare, and AI-driven automation**, this course equips students for **the future of computing.**



Lecture 2: Ubiquitous Computing (UC)

This lecture builds upon the **fundamentals of Ubiquitous Computing (UC)** by exploring its **core** characteristics, context-awareness, modern applications, and emerging UC-related concepts like Calm Technology and Invisibility.

Let's break it down in detail. 🚀



A **Ubiquitous Computing system** must possess the following **five key characteristics** to be effective:

🔟 Computers Need to Be Networked, Distributed & Transparently Accessible 🏵



UC systems are **not standalone devices** but are instead **part of an interconnected network**.

📌 Example:

- Smart home ecosystems (like Google Home and Apple HomeKit) allow multiple devices (lights, thermostats, cameras, speakers) to work together **seamlessly**.
- A **smart office** where the lighting and air conditioning adjust based on the number of people present.
- 2 Human-Computer Interaction (HCI) Should Be More Implicit than Explicit 🤖

Traditional computing requires **explicit user commands** (typing, clicking, swiping). UC aims for "Implicit HCI," where the system understands user intentions without requiring direct input.

Example:

- Smart sensors in cars adjusting seat position based on previous preferences.
- Face ID unlocking your phone without requiring a password.

💡 Key takeaway: UC systems should operate autonomously in the background without requiring constant user interaction.

Context-Awareness: Systems Should Adapt to Their Environment h

UC systems analyze their surroundings and adjust behaviors accordingly.

- Example:
 - Adaptive smartphone brightness that adjusts based on ambient light.
 - Location-based reminders (e.g., "Pick up groceries when near a supermarket").
- Why is this important?

Without **context-awareness**, a UC system is just another **static** computer.

4 Computers Should Operate Autonomously with Self-Governance 🌣

Instead of constantly needing user input, UC systems must be self-sufficient and make intelligent decisions.



- Google Nest Thermostat learns user preferences and adjusts temperature automatically.
- Roomba robotic vacuum cleans without human intervention.

💡 Key takeaway: UC devices should not require micromanagement and should learn user behavior over time.

互 Handling Multiple Dynamic Interactions via Intelligent Decision-Making 🧠

UC systems process multiple interactions simultaneously and use AI-driven decision-making.

📌 Example:

- A smartwatch tracking heart rate, sleep patterns, and movement at the same time.
- Amazon's Just Walk Out stores automatically detect when an item is taken from a shelf and charge the customer without needing checkout.
- 💡 **Key takeaway:** UC systems must be able to **handle multiple tasks efficiently** in real-time.

Context-Awareness in UC: The "Five W's" Framework

Context-awareness helps smart systems understand their users through five dimensions:

1 WHO? (Identity)

- UC systems identify who is interacting with them.
- Current systems mainly recognize a single user but rarely other people in the environment.

📌 Example:

- Face recognition systems (e.g., Face ID unlocking a phone).
- Smart door locks that only unlock for specific users.

Challenge:

Future UC systems must recognize multiple people at once and adjust interactions accordingly.

2 WHAT? (User Activity) 🏃

The system detects what the user is doing and responds accordingly.

Example:

- Fitness trackers detecting if you are running, walking, or sitting.
- Smart fridges detecting when food items are running low and suggesting a grocery list.

💡 Challenge:

It is **difficult to interpret human activity accurately** due to **variability in movement and behavior**.

■ WHERE? (Location)

• UC devices use GPS, Wi-Fi, Bluetooth to track user location.

• Location is a critical factor in making decisions.

P Example:

- Google Maps predicting traffic conditions based on real-time movement data.
- Smart hotel rooms adjusting AC and lighting when a guest enters.

Key takeaway:

Many modern UC applications heavily rely on location data.

- Time-based actions help UC systems predict user behavior.
- Systems track time-sensitive actions and adjust accordingly.

📌 Example:

- Smart home systems lowering the lights at bedtime automatically.
- Wearable health monitors tracking daily habits to detect anomalies (e.g., Apple Watch detecting irregular heartbeats).

💡 Challenge:

Most UC applications do not fully utilize time-based analysis, leaving room for improvement.

5 WHY? (Reason) 🤔

- Understanding user intent is the most complex challenge in UC.
- The system must sense physiological indicators like heart rate, body temperature, and skin response to detect emotions and behavior.

📌 Example:

- Wearable stress monitors detecting anxiety levels.
- Smart home assistants detecting user mood through voice tone analysis.

💡 Challenge:

Accurately understanding why a person is doing something requires advanced AI and data analysis.

PE

Early and Modern Examples of Context-Awareness

🔟 Georgia Tech's Aware Home Project 🏡

- An early UC research project focusing on smart homes and elderly care.
- Key features:
 - Smart floor sensors detecting movements.

• Lost object tracking using RFID technology.

₱ Why was this significant?

It was one of the first projects that used context-aware computing in a real-world environment.

2 Modern Context-Aware Applications 🚀

★ Wall++ (2018)

- A smart interactive wall that detects touch, gestures, and electrical signals.
- Potential uses: Home automation, security, and gaming.

DriveR (2024)

- A road safety system that detects dangerous driving behavior using context-aware sensors.
- Helps create **dynamic real-time maps** for safer driving.

What's next?

UC applications are becoming **more sophisticated**, integrating **AI, IoT, and predictive analytics**.

Additional UC-Related Concepts

☐ Calm Technology ♣

- UC should **blend into the background** without demanding user attention.
- Instead of being **disruptive**, it should act **subtly**.

📌 Examples:

- Non-intrusive notifications (e.g., silent phone vibrations instead of loud ringtones).
- Peripheral awareness features (e.g., video call blur background to reduce distractions).

Invisibility in UC ≈

UC should "disappear" into daily life.

Just like **printing technology** doesn't interrupt **reading a book**, UC should operate **seamlessly**.

📌 Example:

- Automatic subway card scanning without requiring manual input.
- **Voice-activated smart assistants** that respond without pressing buttons.

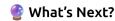
Future Outlook:

We are moving towards technology that we don't "see" but still benefits from daily.

📝 Final Thoughts

💡 Key Takeaways

- ✓ UC systems must be **networked**, **intelligent**, **and context-aware**.
- ✓ The "Five W's" framework helps design better user-centric UC applications.
- ✓ UC should be calm, invisible, and non-intrusive.
- ✓ Future UC applications will involve AI-driven emotion detection, motion-aware environments, and predictive analytics.



With advances in AI, IoT, and wearable computing, Ubiquitous Computing is shaping the future of smart living. 🚀

Lecture 3: Ubiquitous Computing (UC) – Wearable Sensors & Privacy

Q Overview of the Lecture

This lecture dives into wearable sensors, their applications in various fields, and the privacy concerns associated with UC-based devices. It provides a detailed classification of different sensor types, their functions, and real-world implementations in healthcare, fitness, business, and even fashion.

Wearable Sensors: Enhancing Human Capabilities

Wearable sensors are compact, **body-mounted devices** that continuously collect **biological**, **motion**, environmental, and chemical data. They enable seamless monitoring and interaction with the surrounding world.

Types of Wearable Sensors

Wearable sensors are classified into six major categories:

■ Motion Sensors (Tracking Movement & Orientation)

Motion sensors capture acceleration, angular velocity, and magnetic field changes to monitor physical activity and orientation.

Types of Motion Sensors:

- ACC (Accelerometers): Detect movement intensity (e.g., fall detection, step counting).
- GYRO (Gyroscopes): Measure angular rotation (e.g., detecting ankle sprains, VR head tracking).
- MAG (Magnetometers): Detect surrounding magnetic fields (e.g., navigation systems).
- IMU (Inertial Measurement Unit): Combines ACC & GYRO for precise motion tracking.
- MIMU (Magneto-Inertial Measurement Unit): Adds MAG sensors for enhanced activity classification.

Applications:

Smartwatches tracking daily activity & sports performance.

- Wearable **fall-detection systems** for elderly care.
- VR gaming headset motion tracking (e.g., Meta Quest, PlayStation VR).

2 Bioelectric Sensors f (Monitoring Body Signals)

Bioelectric sensors detect electrical activity generated by muscles, heart, brain, and skin conductance.

Types of Bioelectric Sensors:

- Acoustic Sensors (Microphones): Convert sound waves into electrical signals (e.g., voice assistants, smart gloves).
- ECG (Electrocardiography): Measures heart rhythms (e.g., Apple Watch ECG, Fitbit Sense).
- **EEG (Electroencephalography):** Captures **brain activity** (e.g., **Neurosky, Muse Brain-Sensing Headbands**).
- EOG (Electrooculography): Tracks eye movement (e.g., VR eye-tracking technology).
- EMG (Electromyography): Measures muscle activity (e.g., wearable prosthetics, sports performance tracking).
- EDA/GSR (Electrodermal Activity / Galvanic Skin Response): Measures stress & anxiety levels (e.g., stress-monitoring smartwatches).
- Applications:
- Sleep monitoring systems analyzing brain activity.
- **Magnetic Proof** Biofeedback therapy using muscle and heart rate analysis.
- Smart prosthetics responding to brain or muscle signals.

■ Biometric Sensors (Health & Performance Tracking)

Biometric sensors monitor hydration levels, lactate accumulation, and other metabolic markers.

Types of Biometric Sensors:

- **Hydration Sensors:** Measure **body fluid balance** using **impedance**, **light reflection**, **or sweat** analysis.
- Lactate Sensors: Detect lactic acid buildup during intense physical activity (used in sports & fitness training).
- Applications:
- Smart sportswear tracking hydration & fatigue levels.
- Athlete training optimization using metabolic stress monitoring.

4 Environmental Sensors (External Conditions Monitoring)

Environmental sensors detect **temperature**, **pressure**, **and atmospheric changes**.

Types of Environmental Sensors:

• Temperature (TEMP) Sensors: Monitor body temperature in real time (e.g., smart rings, fitness bands).

• Pressure Sensors: Analyze gait, grip strength, and physical force (e.g., smart insoles for walking analysis).

- Applications:
- Smart clothing detecting weather changes.
- Rehabilitation devices monitoring patient movement.

5 Optical & Chemical Sensors / (Monitoring Body Fluids & Blood Circulation)

These sensors analyze **light reflection & chemical compositions** in the body.

Types of Optical & Chemical Sensors:

- PPG (Photoplethysmography) Sensors: Measure blood flow & heart rate (e.g., Apple Watch, Garmin smartwatches).
- Continuous Glucose Monitoring (CGM) Sensors: Track blood sugar levels (e.g., Freestyle Libre, GlucoWise).
- Applications:
- Seizure detection & blood pressure monitoring.
- Diabetes management using CGM systems.

Applications of Wearable Sensors

Wearable sensors have revolutionized multiple fields:

- 🔟 Healthcare & Medical Monitoring 🏥
 - Qardio, Neurosky, Abbott Diabetes Care Smart ECG monitors, Al-powered diabetes tracking.
 - **iTBra, ADAMM** Wearable breast cancer detection systems.
- 🙎 Wellness & Fitness 🏃
 - LUMOBack, Netatmo JUNE Posture correction & UV exposure monitoring.
 - StretchSense Tracks muscle movement for rehabilitation.
- 3 Smart Fashion 👗
 - Light-sensitive & motion-responsive dresses (e.g., Rainbow Winters fashion line).
- 4 Business & Security 🔐
 - Nymi Band, NFC Ring Wearable authentication devices for secure logins & transactions.
- 📌 Impact:
- ✔ Real-time health insights.
- ✓ Improved security with biometric authentication.
- Enhanced athlete performance tracking.

Privacy Concerns in Ubiquitous Computing

With great power comes great responsibility—privacy challenges in UC are a major concern.

P Definition of Privacy (Alan Westin)

"Privacy is the claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated to others."

- 📌 Types of Privacy in UC
- 1 Territorial Privacy Prevents unauthorized access to private spaces.
- Example: A smart wall that collects home data leaking information to third parties.
- 2 Bodily Privacy Protects biometric data & health information.
- Example: A **smart shirt sending health data to a doctor** without consent.
- **Communication Privacy** Protects **personal conversations & interactions**.
- Example: A smartwatch analyzing user messages for mood tracking.
- Borders of Privacy Breaches

UC redefines privacy boundaries by continuously collecting data. Privacy breaches can be classified into:

- ✓ Natural Borders: Physical barriers like walls, doors, and clothing.
- Social Borders: Expectations of confidentiality among professionals (e.g., doctors, lawyers).
- Spatial Borders: Users expect their personal and professional lives to remain separate.
- Transitory Effect Borders: Users expect temporary actions (e.g., old messages) to be forgotten.
- Example of a Privacy Violation:

A fitness app sharing location data with advertisers without consent.

★ Activity: Building a Wearable Sensor Data App

Objective:

Develop an app that records instant sensor data and uploads it to Google Forms.

- 📌 Steps:
- 1 Read **motion & bioelectric sensor data** from a smartwatch or smartphone.
- 2 Send live data to a Google Form.
- 3 Store sensor logs for analysis.
- Real-World Use Case:
- 💡 This activity simulates **how fitness apps collect & store user data** for analytics.
- © Conclusion: The Future of Wearable Sensors & Privacy

Wearable sensors are transforming healthcare, security, fitness, and business, but privacy must be protected. Future Ubiquitous Computing systems must balance innovation with ethical data collection.

- What's Next?
- AI-powered wearable assistants.
- Biometric security replacing passwords.
- Ethical UC frameworks for privacy protection.
- **Final Thought:**
- Wearable sensors should empower users, not exploit them.

Lecture 4: Privacy in Ubiquitous Computing (UC)

This lecture **deeply explores privacy concerns** in Ubiquitous Computing (UC), focusing on **Solove's Privacy Taxonomy**, **different perspectives on privacy**, and **designing fair UC systems**.

Given that UC systems continuously **collect, process, and transmit personal data**, privacy protection is **one of the biggest challenges** in modern computing.

Privacy in Ubiquitous Computing

In **Ubiquitous Computing**, devices like **smartphones**, **wearables**, **smart assistants**, **and IoT systems** collect massive amounts of **user data**. However, **without proper security** and ethical data handling, **privacy violations can occur**.

What is Privacy?

Alan Westin defines privacy as:

- The claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated to others."
- The Fundamental Problem

Most data collection in UC is voluntary (e.g., fitness apps tracking steps).

- **However**, when data is **collected secretly, without consent, or misused**, it leads to **privacy breaches**.
- Example:
 - Google Assistant or Alexa always "listening" even when not activated.
 - Fitness apps selling user health data to insurance companies without consent.
- Why is Privacy Important?
- Protects personal identity.
- 🔽 Prevents data misuse (e.g., unauthorized surveillance).
- 🖊 Ensures trust in technology.

Paniel Solove proposed a privacy framework that classifies different types of privacy violations.

📍 1. Information Collection 📡 (How Data is Collected)

- Types of Violations:
- ✓ Surveillance: Tracking an individual's actions (e.g., CCTV cameras, GPS tracking).
- ✓ Interrogation: Collecting sensitive personal details (e.g., forced biometric registration).

📌 Example:

- Smart homes monitoring user presence.
- Websites collecting browsing habits without consent.

Privacy Risk:

Users may lose control over their personal data.

📍 2. Information Processing 🖥 (How Data is Stored & Analyzed)

- Types of Violations:
- ✓ Aggregation: Linking multiple data sources about a person (e.g., social media data + bank records).
- ✓ **Identification:** Connecting anonymous data to a specific person.
- ✓ Insecurity: Poor data protection, leading to hacks & breaches.
- ✓ Secondary Use: Using data for a purpose other than what was originally agreed.

📌 Example:

- Facebook tracking users across the web even after they log out.
- A health app collecting weight data and selling it to insurance companies.

Privacy Risk:

Users may not even know how their data is being used.

📍 3. Information Dissemination 📤 (How Data is Shared)

- Types of Violations:
- ✔ Breach of Confidentiality: Sharing data that was supposed to be private.
- ✓ **Disclosure:** Exposing sensitive information without permission.
- ✓ Exposure: Revealing intimate private details (e.g., medical history, nude pictures).
- ✔ Blackmail: Threatening to reveal private data for extortion.
- ✓ **Distortion**: **Spreading false information** about someone.
- 📌 Example:
 - A company exposing employee salary details.
 - A hacking group leaking personal photos of celebrities.

Privacy Risk:

Users **lose trust** in organizations handling their data.

4. Invasion of Privacy (Unwanted Intrusions)

- Types of Violations:
- ✓ Intrusion: Forcing someone into unwanted interactions (e.g., spam calls, unsolicited ads).
- ✓ Decisional Interference: Governments or companies controlling user decisions (e.g., China's social credit system monitoring citizens' behaviors).

★ Example:

- Companies tracking political views to manipulate voter behavior.
- Apps forcing users to opt-in to data collection without an alternative.

Privacy Risk:

Leads to unethical control over users.

Do People Care About Privacy?

Different people perceive privacy differently.

- Privacy Fundamentalists:
- Highly distrustful of organizations that collect personal data.
- Prefer complete control over their data.
- Example: People who disable location services, avoid social media, and use encrypted apps (e.g., Signal, ProtonMail).

Privacy Pragmatists:

- Weigh the benefits vs. privacy risks.
- Open to sharing data in exchange for convenience.
- Example: Users who accept cookies but use ad-blockers selectively.

Privacy Unconcerned:

- Trust companies & governments with their data.
- Believe privacy concerns are overhyped.
- Example: People who share personal details freely online.
- 🖈 Which category do you fall into? 🤔

Framework for Designing Fair Ubiquitous Computing Systems

To ensure privacy protection, developers must follow ethical design principles.

- 🔽 1. Transparency: Users must know what data is being collected & how it's used.
- **2. Informed Consent:** Users should **explicitly agree** to data collection.
- 3. Anonymization: Remove personally identifiable information (PII) from datasets.
- 4. Security Measures: Use encryption, authentication, and access controls.
- **5. Data Minimization:** Collect **only necessary data**, avoid excessive tracking.
- ★ Example of a Fair UC System:
- ✓ Apple's privacy labels on apps showing what data is collected.

- ✓ Incognito mode in browsers preventing tracking.
- ✓ GDPR regulations requiring websites to ask for cookie consent.

% Activity: Building a Privacy-Aware ML Model

- Task:
 - Develop an ML model that makes predictions while ensuring data privacy.
 - Use Python, Flask, and MIT App Inventor for deployment.

📌 Steps:

- Train an ML model to predict user behavior (e.g., fitness trends, financial patterns).
- **2** Ensure privacy-focused design (no excessive data storage).
- 3 Deploy using Flask API for secure data transmission.
- This experiment teaches how to build AI models while respecting user privacy.

© Conclusion: Privacy in Ubiquitous Computing

- Ubiquitous Computing brings massive privacy risks due to continuous data collection.
- Solove's Taxonomy helps classify different privacy violations.
- Users have varying privacy concerns (Fundamentalists, Pragmatists, Unconcerned).
- Fair UC Systems must follow ethical principles to protect users.
- Final Thought:
- Privacy should be a fundamental right, not an afterthought in Ubiquitous Computing.
- What's Next?
- More regulations on data privacy (e.g., GDPR, CCPA).
- Stronger privacy-focused AI and blockchain solutions.
- Ubiquitous Computing evolving towards user-controlled privacy.
- 🔐 "Privacy isn't dead, but it needs protection more than ever."

Lecture 5: Ambient & Context-Aware Computing in Ubiquitous Computing (UC)

Q Overview of the Lecture

This lecture introduces **Ambient and Context-Aware Computing**, focusing on how computing systems adapt to their environment by sensing **contextual data** and making **intelligent decisions** in real time.

Key topics covered:

- ✓ Context-aware computing fundamentals
- ✓ Context modeling and logic space

- ✔ Requirements of context-aware applications
- ✓ Hardware demonstrations using Arduino sensors

Context-Aware Computing in UC

★ What is Context-Aware Computing?

A **Context-Aware Computing system** is a computing system that:

- Senses & processes user/environmental data
- Adapts behavior automatically in real-time
- Makes decisions based on dynamic inputs
- ***** Example Use Cases:
- ✓ Smart Classrooms : Adjusts lighting based on instructor presence and tracks student attention.
- \checkmark Smart Offices \blacksquare : Detects CO₂ levels to estimate the number of occupants and suggests stress-relieving exercises.
- ✓ Smart Vehicles ←: Alerts drowsy drivers and suggests optimal driving routes based on historical driving patterns.

\bigcirc Why is Context Awareness Important?

- Reduces user effort by automating decisions.
- Enhances personalization in smart environments.
- Enables real-time decision-making using Al-driven insights.

M Context Modeling in UC

What is Context Modeling?

Context modeling is the **process of identifying, structuring, and utilizing contextual data** to improve system behavior.

Key Questions in Context Modeling:

- 1 Which contextual information is relevant? (E.g., user behavior, location, environmental conditions)
- 2 How do different context elements relate?
- 3 How should the system react to context changes?
- racking in a Smart Environment

Context Element	Example
Time (t)	"Night"
Location (l)	"Bedroom"
User Activity (a)	"Writing on a laptop"
System Response	"Silence all notifications"



Context-aware systems should **predict user needs** and respond **proactively**.

Context-Aware Logic Space

Understanding Logic in Context-Aware Applications

Logic space refers to how UC systems process context and make intelligent decisions.

- 📌 Example: Smart Plant Watering System 🌱
- ✓ Context Modeling: Identifies when the plant needs water.
- ✔ Pervasiveness: Determines if simple data models or knowledge reasoning is needed.
- ✓ System Behavior: Decides whether to be loosely context-aware or fully dependent on contextual data.
- Why is this important?

A well-defined logic space ensures that UC applications act intelligently and efficiently.

***** Requirements of Context-Aware Applications

Q Key Functional Requirements

- ✓ **Sources:** Sensors, user inputs, IoT devices.
- ✓ Example Sensors:
 - Ambient light sensors (for adaptive brightness).
 - IMUs (Inertial Measurement Units) (for motion detection).
 - **Noise sensors** (for adjusting volume based on environment).

📌 Example:

A smart thermostat collects room temperature data and adjusts AC settings accordingly.

- 2 Context Aggregation (Data Processing & Storage)
- ✓ Combines multiple context sources to provide a unified perspective.
- ✓ Ensures data integrity when merging contextual information.
- 📌 Example:

A fitness tracker aggregates heart rate, movement, and sleep data to provide holistic health insights.

- **③** Context Consistency ✓ (Ensuring Data Accuracy)
- ✓ Maintains **reliability** of dynamically changing context models.
- ✓ Ensures data updates reflect real-world conditions.

* Example:

If an air quality sensor detects high CO₂ levels, the ventilation system should respond immediately.

- Context Discovery

 ¬ (Finding Relevant Contextual Data)
- ✓ Locates and retrieves **useful contextual data** from various sources.
- ✓ Ensures **seamless interaction** between different UC systems.
- Example:

A classroom attendance app detects the instructor's presence by sensing Bluetooth devices and ambient CO₂ levels.

- 5 Context Querying & Adaptation in (Real-Time Decision Making)
- ✓ Users or systems should retrieve specific context data using queries.
- ✓ The system automatically adapts based on new contextual changes.
- Example:

If a user leaves a conference room, the system automatically reduces AC usage and turns off lights.

- 6 Context Reasoning (Al-Driven Insights)
- ✓ Uses machine learning & AI to infer hidden patterns.
- ✓ Enables **predictive behavior** based on past data.
- Example:

If students are leaving a lecture early, the system infers the lecture is not engaging and suggests improvements.

Demonstration: Sensing Context with Arduino Sensors

- 🔟 Ultrasonic Proximity Sensor 📡
- ✓ Measures distance using sound waves.
- ✓ Detects obstacles, movement, or human presence.
- * Example Application:

A security system that alerts when unauthorized movement is detected.

🔧 Code Example (Arduino):

```
const int trigPin = 9;
const int echoPin = 10;
long duration;
int distance;
```

```
void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
}
void loop() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.034 / 2;
  Serial.print("Distance: ");
  Serial.println(distance);
}
```


Used in parking sensors, robotics, and security systems.

- 2 Soil Moisture Sensor 🌱
- ✔ Detects soil moisture levels.
- ✓ Controls automatic irrigation systems.
- ***** Example Application:

A smart irrigation system that waters plants when soil is dry.

Code Example (Arduino):

```
#define sensorPower 7
#define sensorPin A0

void setup() {
    pinMode(sensorPower, OUTPUT);
    digitalWrite(sensorPower, LOW);
    Serial.begin(9600);
}

void loop() {
    Serial.print("Moisture Level: ");
    Serial.println(readSensor());
    delay(1000);
}

int readSensor() {
    digitalWrite(sensorPower, HIGH);
```

```
delay(10);
int val = analogRead(sensorPin);
digitalWrite(sensorPower, LOW);
return val;
}
```


Used in agriculture, gardening, and climate monitoring.

■ Proximity Sensor ¬

- ✓ Detects object presence without physical contact.
- ✓ Used in touchless door sensors, automated lighting, and smart appliances.
- ★ Example Application:

A **touchless elevator button** that detects hand motion instead of physical press.

Code Example (Arduino):

```
int IRSensor = 9;
int LED = 13;
void setup() {
  Serial.begin(115200);
  pinMode(IRSensor, INPUT);
  pinMode(LED, OUTPUT);
}
void loop() {
  int sensorStatus = digitalRead(IRSensor);
  if (sensorStatus == 1) {
    digitalWrite(LED, LOW);
    Serial.println("Motion Detected!");
  } else {
    digitalWrite(LED, HIGH);
    Serial.println("Motion Ended!");
  }
}
```

🚀 Key Takeaway:

Used in home automation, security systems, and industrial automation.

© Conclusion: The Future of Context-Aware UC Systems

- ✓ UC systems are shifting towards real-time adaptation & AI-driven automation.
- ✓ Context-aware applications must balance efficiency & ethical concerns.
- ✓ Sensors like IMUs, proximity, and environmental sensors will power smart cities & industries.

- What's Next?
- 🚀 Al-powered fully autonomous context-aware environments.
- Hyper-personalized user experiences based on context reasoning.
- Ethical privacy-first computing solutions for UC.
- 💡 Final Thought: "Smart environments should adapt to humans—not the other way around."

Lecture 6: Sensors, Capacitive Sensing & Context-Aware Computing in UC

Q Overview of the Lecture

This lecture focuses on:

- ✓ Types of sensors for context sensing
- ✓ Context-aware system architecture
- ✓ Capacitive sensing technology
- ✓ Wall++: A room-scale interactive sensing system
- ✔ Applications & real-world implementations



Types of Sensors for Context Sensing

Three Major Types of Sensors in Context-Aware Computing

1 Physical Sensors 🜣

- Capture real-world environmental and motion data.
- Examples:
 - GPS (Global Positioning System): Determines location.
 - Accelerometers: Detect movement, shaking, and orientation.
 - Heart Rate Monitors: Measure pulse rate for health tracking.

💡 Use Case:

Smartphones using GPS & accelerometers for location tracking and step counting.

2 Virtual Sensors

- Software-based sensors that extract contextual information from applications.
- Examples:
 - **Social media check-ins** (e.g., sharing your location on Instagram).
 - Al-powered emotion detection from text messages.

💡 Use Case:

• Google Maps predicting traffic congestion based on user movement data.

🖪 Logical Sensors 🔄

- Combination of physical & virtual sensors for advanced context inference.
- Examples:
 - **Wearable health bands** tracking **heart rate & stress levels** while linking data with an AI-based virtual assistant.
 - Smart buildings adjusting room temperature & lighting based on motion and ambient conditions.

Use Case:

• Smart homes adjusting climate control based on user presence & preferences.

Ш Hierarchy of Context Representation

Q Context Data is Processed in Different Layers:

- 1 Raw Data: Sensor readings (e.g., temperature, movement, GPS location).
- 2 Low-Level Features: Derived from raw data (e.g., "user is walking").
- 3 High-Level Features: Al-driven inference (e.g., "user is exercising").
- **4 Context-Based Actions:** System responses based on inference (e.g., "pause notifications when user is in a meeting").

💡 Example:

• A smartphone detecting low light → Increasing screen brightness automatically.

Architecture of Context-Aware Systems

Context-aware systems consist of **multiple components** working together.

★ Key Components of a Context-Aware System

- ✓ Context Sensors: Collect real-time data (e.g., GPS, heart rate, temperature).
- ✓ Context Processing Engine: Analyzes sensor data to infer meaning.
- ✓ Context Storage & Aggregation: Merges data from multiple sources for better accuracy.
- ✓ Context Reasoning Module: Uses AI/ML to predict patterns & make decisions.
- ✓ **User Interface:** Displays the processed data to the user (e.g., notifications, alerts, recommendations).

Example:

Google Assistant recognizing daily routines and suggesting relevant reminders.

■ Wall++: Room-Scale Interactive & Context-Aware Sensing

Wall++ is a **smart wall system** that uses **capacitive sensing & electromagnetic signal detection** to **track human movement, gestures, and touch interactions** on a large surface.

Developed by:

• Disney Research & Carnegie Mellon University (CHI 2018 Best Paper Award 🖫).

- ♣ How Wall++ Works?
- ✓ Uses special conductive paint to create capacitive sensors on a wall.
- ✓ Detects hand proximity, touch, and even body movement near the wall.
- ✓ Can track gestures, poses, and movements in real-time.
- Applications of Wall++:
- Smart homes: Wall-based gesture control for lights & appliances.
- **Mealthcare:** Tracking patient movement for fall detection.
- Gaming & AR: Creating interactive gaming surfaces.

Foundation of Touch Screens Capacitive Sensing: The Foundation of Touch Sensing Capacitive Sensing: The Foundation of Touch Sensing Capacitive Sensing: The Foundation of Touch Sensing Capacitive S

What is Capacitive Sensing?

Capacitive sensing is a **technology used in touchscreens, smart surfaces, and gesture recognition systems**.

- 1 How It Works?
 - A capacitive sensor consists of **grid-shaped transmitter and receiver electrodes**.
 - When a finger touches the sensor, it disrupts the electric field, which the system detects as input.
- Example:
 - Touchscreens on smartphones & tablets work based on capacitive sensing.
- 2 Evolution of Capacitive Sensing: SmartSkin (2002)
- Developed by: Jun Rekimoto
- ✓ First freehand gesture tracking system.
- ✓ Used conductive grids to detect hand movements above a surface.
- P Legacy:
 - Inspired modern smart touchscreens & interactive surfaces.

Applications of Capacitive Sensing

- ✓ Touchscreens: Found in smartphones, tablets, and laptops.
- ✓ Smart Clothing: Fabric-based sensors for gesture-based wearable tech.
- ✓ Gesture Control: Detecting hand movements in smart home automation.
- ✓ Biometric Authentication: Fingerprint sensors in smartphones.



Smart papers with capacitive sensing for next-gen interactive learning materials.

Demonstrations: Implementing Capacitive Sensing with Arduino & MIT App Inventor

🔟 Demonstration: Wall++ (Capacitive Paint & Sensors) 🏡

Phases of Wall++ Implementation:

- **1** Applying conductive paint (nickel-based) on the wall.
- 2 Using brushes, sprays, or rollers to apply multiple coatings.
- 3 Adding copper traces with vinyl stickers for better sensing.
- 4 Configuring electrode patterns to optimize tracking.

? Challenges:

• Balancing **electromagnetic sensitivity & resolution** for precise tracking.

Activity: Creating a Mobile App for Capacitive Touch Sensing

- Objective: Develop a simple mobile application that displays touch points on the screen.
- Tool: MIT App Inventor (No-code app development platform).

Steps to Implement:

- 1 Use the Canvas element in MIT App Inventor.
- 2 Detect user touch events.
- 3 Display touch coordinates on screen.

Real-World Use Case:

• This activity simulates how smart touchscreens detect touch points & gestures.

© Conclusion: Future of Context-Aware & Capacitive Sensing Technologies

- Context-aware systems are evolving towards AI-driven automation.
- Wall++ and capacitive sensing will play a major role in future smart environments.
- Wearable & IoT-based capacitive sensors will transform human-computer interaction.
- What's Next?
- ✓ Gesture-controlled smart homes.
- ✓ Capacitive-sensing walls for next-gen UI.
- ✓ Al-powered context-aware assistants.
- Final Thought: "The future of computing is invisible—where our environments understand and respond to us effortlessly."

Lecture 7: Eye-Tracking & mmWave Sensing in Ubiquitous Computing (UC)

Q Overview of the Lecture

This lecture introduces two major topics:

- 1 SwitchBack: Using Focus and Saccade Tracking to Guide Users' Attention for Mobile Task Resumption
- 2 RadarFoot: Fine-Grain Ground Surface Context Awareness for Smart Shoes using mmWave Sensing

Both studies explore how Ubiquitous Computing (UC) can enhance human-computer interaction through eye-tracking technology & millimeter-wave radar sensing.

Case Study 1: SwitchBack - Eye-Tracking for Attention Guidance

♣ Motivation: The Problem of Divided Attention in Mobile Use

Many users frequently switch attention between their mobile devices and their surroundings.

- **✓ Example:** A pedestrian checking emails while crossing the street **must look up** to avoid accidents.
- Issues caused by divided attention:
 - Safety risks 🚷 (e.g., distracted walking).
 - **Reduced productivity** (losing track of reading progress).

★ SwitchBack: The Solution

- ✓ SwitchBack detects when a user returns to their phone after a distraction.
- ✓ It guides the user back to the last reading position using Focus and Saccade Tracking (FAST).
- ✓ Automatically scrolls the text when the user reaches the bottom of the screen.
- 💡 Real-World Impact:
- Useful when touchscreen controls don't work (e.g., gloves in winter).
- Improves reading experience for multitaskers.

How SwitchBack Works: Eye-Tracking Mechanism

- Q Step 1: Detecting User Attention
 - SwitchBack uses Focus and Saccade Tracking (FAST) to determine whether a user is looking at the screen.
- Q Step 2: Tracking Eye Movements
 - It measures the user's pupil movement relative to the eye to track reading position.
- Step 3: Detecting Saccades (Eye Jumps)

- When reading text, users make **small rapid eye movements** (saccades).
- SwitchBack tracks saccades to detect when a user moves to a new line.

Step 4: Automatically Scrolling the Screen

When a user reaches the end of visible text, SwitchBack auto-scrolls based on eye movement.

Challenges:

- ✓ Noise & false positives: Misreading eye jumps can lead to incorrect scrolling.
- ✓ Different reading speeds: Needs to adjust dynamically to user behavior.
- How it Improves User Experience
- No need for manual scrolling.
- Helps users quickly resume tasks after interruptions.
- Supports accessibility for users with disabilities.
- Demo Video: SwitchBack Eye-Tracking

Case Study 2: RadarFoot - mmWave Sensing for Smart Shoes

Motivation: Enhancing Ground Surface Context Awareness

RadarFoot is an intelligent shoe technology that uses millimeter-wave (mmWave) radar to:

- ✓ Identify ground surfaces (e.g., wet roads, grass, snow, sand).
- ✓ Improve safety by detecting hazardous walking conditions.
- Enhance athlete performance tracking (e.g., measuring running efficiency on different terrains).

Developed by:

Monash University (Australia) & University of New South Wales (ACM UIST 2023).

Why It Matters?

- Smart footwear can improve fall detection & terrain adaptation.
- Could be used in assistive walking devices for the elderly.
- Enhances AI-driven personal training in sports.

ammWave Sensing Technology: How It Works

- Millimeter-wave (mmWave) sensors operate in the 30-300 GHz frequency range.
- They use Frequency Modulated Continuous Wave (FMCW) radar to detect objects & movement.

***** How mmWave Radar Detects Ground Surfaces

- ✓ The radar transmits a **chirp signal** (a sinusoidal wave with increasing frequency).
- ✓ The signal reflects off the ground and returns to the receiver.
- ✓ The time delay & frequency shift in the reflected signal helps determine:
 - **Distance** to the surface.
 - Material properties (e.g., wetness, hardness, friction).

• Walking conditions (e.g., smooth road vs. rough terrain).

📌 Example:

- Walking on ice → Radar detects high reflectivity → Alerts user to be cautious.
- Walking on soft grass → Radar detects absorption → Adjusts running recommendations.

Technical Reference: FMCW Radar Basics

mmWave Radar: Signal Processing & Object Detection

Key Equations:

✔ Round-Trip Time Delay (τ):

- $(\tau = \frac{2d}{c})$
- d = distance to object, c = speed of light.

✓ Identifying Multiple Objects

- Multiple objects create multiple reflected signals.
- The system uses Fourier Transform to distinguish them.
- 💡 Challenge:
- Objects at the same distance but different materials may have similar reflections.
- Requires AI-based classification models to improve accuracy.

Applications of mmWave Sensing

- 💡 mmWave radar is already used in autonomous vehicles & smart homes.
- ✓ **Self-driving cars** (Tesla, Waymo) use mmWave for **collision detection**.
- ✓ Gesture control in smart homes (Google Nest Hub uses mmWave for hand gestures).
- ✓ Health monitoring (mmWave radars in sleep tracking devices like Google Soli).
- RadarFoot expands these applications to smart shoes.

RadarFoot & The Gait Cycle: Understanding Walking Patterns

• The **gait cycle** is the sequence of movements during walking or running.

- ✓ RadarFoot analyzes gait data to assess user balance & step efficiency.
- ✓ Helps in detecting early signs of mobility disorders (e.g., Parkinson's Disease, Stroke Recovery).
- **Demo Video:** RadarFoot mmWave Smart Shoes

* Activity: Implementing Capacitive Sensing in a Mobile App

- **Objective:** Build a simple mobile app that **visualizes touch interactions** using capacitive sensing.
- Tool: MIT App Inventor (No-code platform).

- 📌 Steps:
- 1 Use a canvas element to track touch points.
- **2** Detect multi-touch gestures (e.g., pinch, swipe).
- 3 Display real-time touch coordinates on-screen.

💡 Real-World Use Case:

- Simulates **touchscreen input processing** in smartphones.
- Can be extended to **gesture recognition in AR/VR applications**.

Conclusion: The Future of Eye-Tracking & mmWave in UC

- SwitchBack shows how AI can improve digital reading experiences.
- RadarFoot demonstrates the power of mmWave in mobility analysis.
- AI-powered smart shoes could revolutionize healthcare & sports.
- Gesture & eye-tracking will redefine human-computer interaction.
- ✓ AR & VR applications for eye-tracking.
- ✓ Smart clothing with built-in radar sensors.
- ✓ AI-powered navigation for visually impaired users.
- Final Thought:
- "The future of computing will be invisible—seamlessly woven into everyday life."

In-Depth Analysis of FMCW mmWave Radar Sensing

Q Overview of the Document

This document, titled "Introduction to mmWave Sensing: FMCW Radars", authored by Sandeep Rao from Texas Instruments, presents an in-depth exploration of Frequency-Modulated Continuous Wave (FMCW) radars for millimeter-wave (mmWave) sensing.

The key areas covered include:

- ✓ Fundamentals of FMCW radar operation
- ✓ Measuring the range of multiple objects
- ✓ Intermediate Frequency (IF) signal & bandwidth ◆
- ✓ Range Resolution & Velocity Estimation ←
- ✔ Fourier Transforms for signal processing IIII

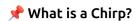
Introduction to mmWave Sensing: FMCW Radars

- mmWave radars operate in the 30 GHz 300 GHz frequency range.
- FMCW radars are used for object detection, velocity measurement, and motion tracking.
- Key applications include:

- **Self-driving cars** (collision avoidance & lane detection).
- Industrial automation (object positioning & robotics).
- **Smart home automation a** (gesture control & security).

📌 FMCW radars work by transmitting a "chirp" signal and analyzing its reflection from objects.

Fundamentals of FMCW Radar Operation



A chirp is a sinusoidal signal whose frequency increases linearly over time.

Ш Key Chirp Parameters:

- **Start Frequency (fc):** Initial frequency of the chirp.
- **Bandwidth (B):** The frequency range swept by the chirp.
- **Duration (Tc):** The total time for one chirp.
- **Slope (S):** The rate at which frequency increases ((S = \frac{B}{Tc})).

💡 Example Calculation:

• If (B = 4) GHz and (Tc = 40) µs, then $(S = \frac{4 GHz}{40 \mu s} = 100 MHz/\mu s).$

📡 How FMCW Radar Measures Range

- Step-by-Step Process
- 1 A synthesizer generates a chirp signal .
- 2 The TX antenna transmits the chirp 1.
- The chirp reflects off objects and returns to the RX antenna .
- 4 The received signal is mixed with the transmitted chirp •.
- \square A low-frequency Intermediate Frequency (IF) signal is generated \square .



The IF signal frequency is proportional to object distance.

📏 Calculating Object Distance

The distance **d** to an object is determined using the **round-trip time delay** (τ) of the reflected chirp.

📌 Formula:

 $[\lambda = \frac{2d}{c}][f_{IF} = S \lambda = \frac{2d}{c}]$

ш Example Calculation:

• If ($S = 100 \text{ MHz/}\mu s$) and an object is **5m away**, then $\frac{2(5m)}{3 \times 10^8 \text{ m/s}} = 33.3 \text{ ns}.$

The IF frequency is
 (f_{IF} = (100 MHz/\(\mu\)s)\times (33.3 ns) = 3.33 MHz).

🚀 Takeaway: Higher IF frequency means a farther object.

Ш Range Resolution in FMCW Radar

📌 Definition: The minimum distance between two objects that allows them to be resolved separately.

📌 Formula:

 $[d_{res}] = \frac{c}{2B}$ where **B** is the chirp bandwidth and **c** is the speed of light.

ш Example:

Bandwidth (B)	Range Resolution ((d_{res}))
4 GHz	3.75 cm
2 GHz	7.5 cm
1 GHz	15 cm
600 MHz	25 cm

- Larger bandwidth → Better resolution.
- A higher chirp bandwidth improves the ability to distinguish close objects.

Measuring Multiple Objects Using Fourier Transforms

reach one produces a distinct IF signal.

- ш Processing Steps:
- 1 Convert the time-domain IF signal into the frequency domain using Fourier Transform.
- 2 Identify peaks in the frequency spectrum, where each peak corresponds to an object.
- 3 The frequency of each peak determines object distance.

₫ Example:

- If two objects are at 5m and 7m, their IF signals might be 3.33 MHz & 4.67 MHz.
- The Fourier Transform separates these signals into distinct peaks.

🚀 Takeaway:

A high-resolution Fourier Transform improves object detection accuracy.

Velocity Measurement in FMCW Radar

- **Problem:** How do we detect object **motion & speed**?
- 📌 Solution: Doppler Effect in FMCW radar.

1 Doppler Shift in Radar

- If an object moves **toward** the radar, the received chirp is **compressed** (higher frequency).
- If an object moves **away**, the received chirp is **stretched** (lower frequency).
- The **Doppler shift** $((f_D))$ is proportional to object velocity (v).

 \nearrow Formula for Velocity Estimation: [v = \frac{\lambda f_D}{2}] where λ is the radar wavelength.

ш Example:

If f_D = 300 Hz and (\lambda = 3.9 mm),
 [v = \frac{3.9 mm \times 300 Hz}{2} = 0.585 m/s.]

🚀 Takeaway:

Faster objects create larger Doppler shifts.

Measuring Velocity Using Multiple Chirps

Key Concept: By transmitting multiple chirps at different times, we can estimate velocity more accurately.

ш Steps:

- Transmit two chirps separated by (T_c).
- 2 Compare the phase difference ((\Delta\phi)) between them.
- 3 Estimate velocity using:

 $[v = \frac{\lambda \cdot T_c}{\lambda \cdot T_c}]$

ш Example:

If (\Delta \phi = 90^\circ) and (T_c = 40 μs),
 [v = \frac{3.9 mm \times 90^\circ}{4\pi \times 40 μs} = 0.78 m/s.]

🚀 Takeaway:

Phase-based velocity estimation is highly accurate for slow-moving objects.

TANGLE Estimation (Angle of Arrival - AoA)

- Problem: How do we determine the direction of an object?
- 📌 Solution: Multiple antennas using phase differences.
- 📌 Key Formula:

[\theta = $\sinh^{-1} \left(\frac{\phi \right) }{2\pi d} \right)$ where **d** is the antenna spacing.

ш Example:

• If (d = $\frac{1}{2}$) and (\Delta \phi = 45^\circ), [\theta = \sin^{-1} \left(\frac{\lambda \times 45^\circ}{2\pi \times \lambda/2} \right) = 22.5^\circ.]



More antennas → More accurate angle estimation.

Conclusion: The Future of FMCW mmWave Radar

- ✓ Self-driving cars (Autonomous Navigation).
- ✓ Healthcare (Remote heart rate sensing).
- ✓ Security & surveillance (Motion detection).
- ✓ Robotics & industrial automation.
- Final Thought:
- "FMCW radar is revolutionizing real-time object sensing with extreme precision."

Lecture 8: RadarFoot – Fine-Grain Ground Surface Context Awareness for Smart Shoes

Q Overview of the Lecture

This lecture presents **RadarFoot**, a **smart shoe technology** that uses **millimeter-wave (mmWave) radar sensing** to classify **different ground surfaces** based on **reflected radar signals**.

- Developed by:
 - · Monash University, Australia
 - University of New South Wales, Australia
 - CSIRO's Data61 (Australia)
 - Presented at UIST '23 (ACM Symposium on User Interface Software & Technology)
- Core topics covered:
- ✓ The Gait Cycle & Ground Surface Classification
- ✔ How mmWave RadarFoot Works
- ✓ Why Traditional Sensors (IMUs) Are Not Enough
- ✔ Feature Extraction & Machine Learning for Surface Classification

🚶 The Gait Cycle & Smart Shoe Sensing

Understanding the Gait Cycle

The **gait cycle** refers to the **pattern of movement** when walking.

It consists of **different phases** that can be **analyzed for identifying surface interactions**.

- ★ Key Phases of the Gait Cycle:
- 1 Heel Strike Foot makes initial contact with the ground.
- 2 Loading Response Body weight shifts onto the foot.

- **3** Midstance Foot is flat, supporting the full body weight.
- 4 Terminal Stance & Toe-Off Foot pushes off the ground.
- Why is this important?
 - Different surfaces (grass, ice, sand, concrete) affect these phases differently.
 - mmWave radar can detect these changes by analyzing the reflected signals.
- Reference Video: Gait Cycle Explanation

📡 How mmWave RadarFoot Works

📌 Sensing Principle of RadarFoot

RadarFoot detects surfaces by analyzing how radar signals reflect off the ground.

Two key factors affect reflected radar signals:

- ✓ Signal Travel Distance Longer distance reduces intensity.
- ✓ Reflection Coefficient (r) Determined by the refractive index of the surface.
- Refractive Index & Permittivity:
 - Different materials allow or block electromagnetic waves differently.
 - Wet surfaces absorb more signals \ref{main} , while hard surfaces reflect more \ref{main} .
- How RadarFoot Classifies Surfaces:
- Tracks amplitude & phase changes in reflected signals Ⅲ.
- Analyzes absorption rates & reflection patterns <a>s

Why Traditional Sensors (IMUs) Are Not Enough

- ★ Issue with IMUs (Inertial Measurement Units):
 - No significant change in IMU readings when walking over different surfaces.
 - IMUs only measure acceleration & angular velocity (e.g., foot movement) but cannot differentiate surface types.
- Why mmWave Works Better?
- mmWave radar detects subtle differences in signal reflection & absorption.
- ✓ Tracks how waves interact with different ground materials.
- ✓ 14 key features extracted from reflected signal amplitude.
- 🚀 Takeaway:
- mmWave offers a richer dataset than IMU-based motion tracking.

Ш Feature Extraction & Machine Learning for Surface Classification

♣ How Machine Learning Improves Surface Detection

RadarFoot extracts 14 key features from reflected mmWave signals to classify surfaces.

- Best Performing Algorithm:
- 🔽 Random Forest Algorithm 🌳
 - Provided highest accuracy for classifying ground materials.

ш Why Random Forest?

- ✓ Handles non-linear relationships in data.
- ✓ Works well for noisy signals.
- ✔ Provides high classification accuracy.
- **?** Comparison with Other Models:

Model	Performance	
Random Forest 🌳	✓ Best Accuracy	
SVM (Support Vector Machine)	× Less accurate	
KNN (K-Nearest Neighbors)	× Slower & less robust	



AI-powered classification enables real-time surface detection in smart shoes.

© Conclusion: The Future of Smart Shoes with mmWave Sensing

RadarFoot is a major breakthrough in footwear-based sensing, enabling ground-aware smart shoes.

- Potential Applications:
- ✓ **Sports & Fitness** Detect running efficiency on different terrains.
- ✓ Healthcare Assistive walking for elderly & disabled users.
- ✓ Robotics Ground-aware foot placement for legged robots.
- What's Next?
- ✓ Combining mmWave with AI for advanced terrain adaptation.
- ✓ Integrating RadarFoot into commercial smart shoes.
- Final Thought:
- 🔮 "Future footwear will not just be worn—it will sense, adapt, and enhance movement in real-time." 🚶



Lecture 9: Location Sensing in Ubiquitous Computing (UC)

Q Overview of the Lecture

This lecture, part of **Module IV** (**Part I**), discusses **location sensing**—a critical component in **Ubiquitous Computing (UC)** that enables devices to determine **position, movement, and spatial awareness** in real-time.

- Key Topics Covered:
- ✓ Types of Location Representation
- ✓ Location Sensing Techniques
- ✔ Real-World Applications
- ✔ Advanced Methods like Trilateration, Hyperbolic Lateration, Triangulation & Fingerprinting
- References for Further Reading:
 - Location Preview Draft University of Washington
 - IEEE Location Sensing Paper Stanford

Location Representation in Ubiquitous Computing

A **location** can be represented in multiple ways:

- 1 Physical vs. Symbolic Locations
- ✔ Physical Location: Expressed using exact coordinates like latitude, longitude, and altitude.
 - ★ Example: "47°39′17" N, 122°18′23" W at 20.5m elevation."
 - ✓ Symbolic Location: Expressed using human-readable terms.
 - 🖈 Example: "Inside a kitchen," "Near the mailbox," "On a train approaching Denver."
- Pakeaway:
- Physical locations are essential for GPS & autonomous navigation.
- Symbolic locations are better for human interactions & AI assistants.

2 Absolute vs. Relative Locations

- ✓ Absolute Location: A fixed coordinate that does not change.
 - P Example: "The Eiffel Tower is at 48.8584° N, 2.2945° E."
 - ✓ Relative Location: Positioning based on distance from a reference point.
 - PExample: "The lost hiker is 200m northwest of the base camp."
- Why It Matters?
- Absolute location is critical for global navigation & mapping.
- Relative location is useful for indoor navigation & augmented reality.

Location Sensing Techniques

There are multiple ways to sense and compute location, each with unique strengths & challenges.

Proximity Sensing

- ✓ Determines location based on closeness to a reference point.
- ✓ Three main approaches:
 - Physical Contact Sensors: Detects direct interaction (e.g., pressure sensors in smart floors).
 - Wireless Networks: Identifies location based on nearby WiFi/Bluetooth devices.
 - Automatic ID Systems: Uses RFID, NFC, or credit card transactions for location detection.
- 📌 Example:
 - Your phone detects a home WiFi network → Assumes you're at home.
- Why It Matters?
- Proximity sensing is crucial for indoor positioning.

Trilateration \

- ✓ Uses distances from three or more reference points to determine location.
- ✓ Works similar to GPS satellite positioning.
- **★** How It Works:
- 1 A device measures its distance from multiple known points.
- **2** Each distance defines a circle around the reference point.
- 3 The intersection of the circles reveals the device's location.

Ш Example:

- If you are 5 km from Tower A, 3 km from Tower B, and 4 km from Tower C, your location is at the
 intersection of those three circles.
- Used in GPS-based navigation & outdoor tracking.

3 Hyperbolic Lateration

- ✓ Uses the difference in signal arrival times to estimate location.
- ✓ Instead of measuring distance, it calculates time difference of arrival (TDOA).
- 📌 Example:
 - Mobile networks use hyperbolic lateration for phone location tracking.
- Why It Matters?
- Works well in GPS-denied environments (e.g., deep urban areas).

4 Triangulation

- ✓ Uses angles of arrival (AOA) from reference points to determine location.
- ✓ Requires directional antennas to estimate position.

- How It Works:
- 1 Two base stations measure the angle at which they receive the signal.
- 2 The angles determine the position of the device.

ш Example:

• Radar & GPS augmentation use triangulation for precise positioning.

- Used in military, defense, and autonomous vehicles.

5 Fingerprinting **Q**

- ✓ Uses pattern matching techniques to estimate location.
- ✔ Relies on two key properties:
 - Temporal Stability: A radio signal remains stable over time at a specific location.
 - Spatial Variability: A radio signal varies between different locations.
- **★** How It Works:
- 1 A database of signal fingerprints is created for a location (training phase).
- 2 The system compares real-time signals with stored fingerprints.

ш Example:

• WiFi-based indoor positioning systems use fingerprinting.

- 🚀 Why It Matters?
- Most accurate for indoor location tracking (malls, airports, hospitals).

6 Dead Reckoning 🚶

- ✓ Estimates location based on movement speed & direction from the last known position.
- ✓ Used when GPS signals are lost (e.g., inside tunnels).
- 📌 Example:
 - A car continues estimating its position after losing GPS in a tunnel.
- Why It Matters?
- Essential for underground and submarine navigation.

🗷 Scene Analysis 🛡

- ✓ Uses AI & visual features to infer location.
- ✓ Types:
 - Static Scene Analysis: Matches observed features to a pre-existing database.
 - **Differential Scene Analysis:** Compares consecutive frames to track movement.



• Google Lens recognizes landmarks to estimate user location.



Powers modern AR & Al-driven navigation systems.

M Revisiting Applications of Location Sensing

- Key Real-World Applications:
- ✓ **GPS Navigation & Mapping** Google Maps, Waze.
- ✓ Autonomous Vehicles Self-driving car localization.
- ✓ **Smart Homes** Location-aware home automation.
- ✓ Retail & Marketing Proximity-based ads & offers.
- ✓ Healthcare Patient tracking & fall detection.
- **Demo Video:** Location Sensing Applications

© Conclusion: The Future of Location Sensing in UC

- Key Trends:
- ✓ Fusion of multiple sensing techniques for higher accuracy.
- ✓ Al-powered indoor positioning without GPS.
- ✔ Privacy-first location tracking (GDPR-compliant systems).
- ✓ Improved energy efficiency for always-on location sensing.
- Final Thought:
- 🧣 "In the future, location sensing will be seamless, ultra-precise, and privacy-aware."

Lecture 10: Location Sensing Systems in Ubiquitous Computing (UC)

Q Overview of the Lecture

This lecture builds upon **location sensing** concepts by exploring **different location tracking systems**, including **GPS**, **Active Badge**, **Active Bat**, **and Cricket Systems**. It also includes a **graded activity** using **MIT App Inventor to visualize trilateration**.

- Key Topics Covered:
- ✓ Global Positioning System (GPS) & its Components
- ✓ Active Badge (IR-Based Indoor Location Tracking)
- ✓ Active Bat (Ultrasonic-Based Location Tracking)
- ✔ Cricket System (RF + Ultrasound Hybrid Localization)
- ✓ Graded Activity: Implementing Trilateration in MIT App Inventor
- References for Further Reading:

- University of Washington Location Sensing Notes
- Stanford IEEE Location Sensing Paper

Global Positioning System (GPS)

★ What is GPS?

✓ GPS is a satellite-based navigation system that provides positioning, navigation, and timing (PNT) services.

- ✓ It consists of **24+ geosynchronous satellites** orbiting the Earth.
- ✓ Each GPS satellite transmits signals that contain:
 - Satellite location data
 - Timestamp information
- **★** How GPS Works?
- 1 Each satellite transmits a signal containing a pseudorandom ID, ephemeris data, and almanac data.
- 2 The GPS receiver calculates the time difference between the transmitted and received signal.
- The Time Difference of Arrival (TDOA) is used in hyperbolic lateration to compute location.
- 4 A minimum of 4 satellites is required to determine 3D location (latitude, longitude, and altitude).
- ✓ Used in smartphones, vehicles, aviation, and defense applications.
- ✓ Works outdoors, but struggles with indoor positioning due to signal blockage.
- Example of GPS Usage:
 - Google Maps using GPS to determine user location.

% Active Badge: IR-Based Indoor Location Tracking

- Peveloped by: Olivetti Research Lab (1992)
- ✓ One of the first indoor tracking systems.
- ✓ Designed to track employees & visitors inside buildings.
- How It Works?
- **1** Each user wears an infrared (IR) badge that emits a unique ID code.
- 2 Networked IR sensors detect the badge signal.
- 3 The location of the badge is determined based on the sensor detecting the strongest IR signal.
- 4 The system updates user location every 10-15 seconds.
- ✓ First step towards modern indoor tracking.
- ✓ Used for security & access control in corporate environments.
- Limitations:
- × IR signals cannot pass through walls (requiring multiple sensors).
- \times Limited to a 6m range from sensors.

- Example of IR-Based Location Tracking:
 - Remote control IR communication with a TV.

📡 Active Bat: Ultrasonic-Based Indoor Location Tracking

- Poveloped by: AT&T Cambridge (1997)
- ✓ Uses ultrasonic pulses to measure location inside buildings.
- **★** How It Works?
- 1 Each user wears a "Bat" device that emits ultrasonic pulses.
- These pulses are detected by ceiling-mounted ultrasonic receivers.
- 3 The system calculates time-of-flight (ToF) of the ultrasonic waves to determine distance.
- 4 Triangulation is used to find the user's exact location.
- ✓ Higher accuracy (~3 cm) compared to Active Badge.
- ✓ More reliable than IR-based systems (since ultrasound reflects off surfaces).
- Limitations:
- \times Requires ceiling-mounted receivers across large areas.
- \times Slower update rates (compared to RF-based systems).
- Example of Ultrasonic Tracking:
 - Parking sensors in cars detecting nearby objects using ultrasound.

🔄 Cricket: RF + Ultrasound Hybrid Localization

- Property Developed by: MIT (2000s)
- ✓ A hybrid RF (radio frequency) & ultrasonic positioning system.
- ✓ Designed for indoor navigation & robotics.
- A How It Works?
- 1 Beacons (fixed nodes) transmit both RF and ultrasonic signals.
- 2 Mobile Cricket tags detect the signals & calculate distance using time-of-flight (ToF).
- 3 Trilateration is used to estimate the tag's relative position.
- Why It Matters?
- ✓ More scalable than Active Bat.
- ✓ Works without requiring complex wired infrastructure.
- Limitations:
- × Requires multiple beacons for high accuracy.
- Example of Hybrid RF & Ultrasonic Tracking:
 - Amazon warehouse robots using RF-based localization.



Graded Activity: Implementing Trilateration in MIT App Inventor

Objective:

Develop a graphical simulation of trilateration in MIT App Inventor.

- Step 1: Create 4 Fixed Reference Points
- ✔ Place 4 fixed reference nodes on a graphical interface.
- ₱ Step 2: Allow Users to Select a Point on the Screen
- ✓ User touches the screen to create a "device" point.
- ★ Step 3: Compute Distances
- ✓ The app draws circles around reference points with radii equal to the computed distances.
- ₱ Step 4: Find Intersection Point
- ✓ Trilateration determines the location of the device based on circle intersections.
- Bonus Task:
- ✓ Modify the app to automatically track the device's real-world latitude & longitude using GPS sensors.
- **Reference for MIT App Inventor:** MIT App Inventor Tutorials
- Why This Activity Matters?
- ✓ Hands-on understanding of trilateration.
- ✔ Practical implementation of GPS concepts.

© Conclusion: The Future of Location Sensing

- Key Trends:
- ✓ AI-Powered Indoor Positioning (combining WiFi, Bluetooth, and RF).
- ✔ Privacy-Preserving GPS Technologies (Secure Multi-Party Computation).
- ✓ Energy-Efficient Location Tracking (using ML to optimize GPS power consumption).
- Final Thought:
- 🔮 "In the future, location sensing will be ultra-precise, seamless, and privacy-aware." 🚀

Lecture 11: Queries & Models in Location Sensing (UC)

Q Overview of the Lecture

This lecture builds on **location sensing models**, focusing on **query types**, **navigation**, **range queries**, **and data modeling** in **Ubiquitous Computing (UC)**. It also discusses **Microsoft's GeoLife Dataset**, which

contains real-world GPS data for mobility research.

- Key Topics Covered:
- ✓ Types of Queries in Location Models
- ✓ Navigation & Road Topology for UC
- ✓ Range Queries & Geocasting in Context-Aware Systems
- ✓ Requirements for Location Models
- ✓ GeoLife Dataset: Real-World GPS Trajectories for Research
- References for Further Reading:
 - Springer Location Query Models
 - Microsoft GeoLife GPS Trajectory Dataset

Queries to Location Models

In **Ubiquitous Computing (UC)**, applications use **location models** to **retrieve spatial data** and enhance context awareness.

Q Types of Queries in Location Models

1 Position Queries 📌

- ✓ Definition: Determines the position of static or mobile objects.
- ✓ Example Use Cases:
 - Tracking users, buildings, vehicles, or bus stops.
 - Finding the closest available parking spot.
 - Identifying a moving target's location in real-time.

***** Key Feature:

• Supports multiple coordinate reference systems (global GPS & local indoor maps).

Why It Matters?

Enables navigation, industrial planning, and smart city development.

2 Nearest Neighbor Queries Q

- ✓ **Definition:** Finds the **closest objects** to a reference position.
- ✔ Requires:
 - Object positions
 - A distance function to measure proximity
- ₱ Geometric vs. Symbolic Distance:
- ✓ Geometric: Uses Euclidean distance for precise positioning.
- ✓ Symbolic: Defines custom distance rules (e.g., "room X is next to room Y").

ш Example Scenarios:

Scenario	Direct Distance	Real Distance
Restaurant across a highway	100m	1km (due to road network)
Nearest hospital via road	3km	4.5km (due to traffic routes)

- Real-world movement is affected by obstacles like roads, rivers, and restricted areas.

3 Range Queries 🕥

- ✓ Definition: Finds all objects within a geographic boundary.
- ✓ Example Use Cases:
 - Checking if a room is empty before locking.
 - · Emergency evacuation monitoring.
 - Smart messaging (Geocasting): Sending messages to users in a specific area.
- **★** How It Works:
- ✓ For geometric coordinates, the system calculates whether a point is inside a defined boundary.
- ✓ For symbolic coordinates, predefined relationships determine spatial containment.
- Enables geofencing, smart IoT automation, and emergency response systems.

Navigation & Road Topology in Location Models

- ✓ Navigation requires a model of the transportation network (roads, train lines, etc.).
- ✓ Interconnected locations define possible routes from point A to B.
- ★ Example Components:
- ✓ Road Geometry: Defines the physical structure of roads.
- Road Topology: Maps how roads connect at intersections.
- Essential for GPS navigation, ride-sharing, and smart traffic management.

M Requirements for Location Models

For **effective location sensing**, models must support:

- ✓ Object Positions Store object locations using geometric (GPS) or symbolic coordinates.
- ✓ **Distance Functions** Calculate **travel distances**, not just straight-line distances.
- ✓ Topological Relations Define spatial containment (room in building) & connectivity (road networks).
- ✔ Orientation Track object direction & rotation for better positioning.
- Accuracy Ensure real-world data consistency & frequent updates.

Supports smart city planning, indoor navigation, and automated transportation systems.

Case Study: Microsoft GeoLife Dataset

Microsoft Research Asia collected the GeoLife dataset, which provides:

- ✓ Real-world GPS trajectory data from 178 users (2007-2011).
- ✓ 17,621 trajectories covering 1.25 million km (total duration: 48,203 hours).
- ✓ Data logged at high resolution (every 1-5 seconds or 5-10 meters per point).

♠ Dataset Features:

Feature	Description
Latitude & Longitude	GPS coordinates
Altitude	Elevation data
Timestamp	Logs time of movement
Transportation Mode	Walking, driving, bus, bike

- Enables research in mobility analysis, location privacy, and transportation optimization.
- ★ Example Research Applications:
- ✓ Smart travel route predictions.
- ✓ Traffic pattern analysis.
- ✓ Al-powered ride-sharing optimization.

Ш Real-World Applications of Location Queries

- Key Use Cases:
- ✓ Navigation Apps (Google Maps, Waze) Uses position & nearest neighbor queries.
- ✓ Smart Home Automation Uses range queries to detect presence & trigger actions.
- ✓ Ride-Sharing (Uber, Lyft) Uses nearest neighbor search to match drivers & passengers.
- ✓ Emergency Response Systems Uses range queries for disaster management.
- ✓ Retail & Marketing (Geofencing Ads) Uses range-based targeting.
- What's Next?
- ✓ AI-powered predictive location modeling.
- ✔ Privacy-preserving GPS techniques.
- ✓ Energy-efficient location tracking for IoT.
- Final Thought:
- 🔮 "Location sensing in UC is evolving towards ultra-precise, privacy-first, and AI-driven systems." 🚀

Lecture 12: Mining Location Histories & Travel Sequences in Ubiquitous Computing (UC)

Q Overview of the Lecture

This lecture explores location history modeling, travel sequence mining, and recommendation systems using GPS trajectory data. It is based on the GeoLife dataset, which helps in analyzing human mobility patterns.

- Key Topics Covered:
- ✓ Extracting Interesting Locations from GPS Trajectories
- ✓ Stay Point Detection in GPS Logs
- ✓ Location History Modeling with Hierarchical Graphs
- ✓ HITS-Based Inference for Travel Patterns
- ✓ Mining Classical Travel Sequences
- ✔ Personalized Travel Recommendations
- ✓ Graded Activity: Designing a UC System for Users with Clinical Leg Injuries
- References for Further Reading:
 - GeoLife GPS Trajectory Dataset Microsoft Research
 - Mining Interesting Locations and Travel Sequences

Ш Extracting Interesting Locations from GPS Trajectories

- ✓ People want to know which locations are the most interesting in a given region.
- ✓ They also want to explore **common travel sequences** between these locations.
- Example Use Cases:
- ✓ Tourists identifying the most visited places in a city.
- ✓ Urban planners analyzing mobility patterns for traffic management.
- Why It Matters?
- Helps in building recommendation systems for personalized travel suggestions.

Understanding GPS Logs & Stay Point Detection

- GPS Log Definition:
 - A **GPS log** is a collection of **GPS points** ($P = \{p_1, p_2, ..., p_n\}$).
 - These points are **sequentially connected into a curve** based on timestamps.
 - The curve is **split into GPS trajectories**, which define **travel paths**.
- What is a Stay Point?

- ✓ A stay point (S) is a geographic region where a user stayed for a certain time interval.
- ✓ Extracting stay points depends on two threshold parameters:
 - **Time threshold** Minimum time spent at a location to be considered significant.
 - **Distance threshold** Maximum distance traveled while staying in the same location.

ш Example Calculation:

If a user remains within a 200m radius for at least 10 minutes, the system marks it as a stay point.

- Why Stay Points Matter?
- Helps in identifying commonly visited locations for clustering & mobility analysis.

7

Location History Modeling with Hierarchical Graphs

- ✓ Location history records the sequence of places a person has visited.
- ✓ A tree-based hierarchical graph (TBHG) groups stay points into clusters.
- **★** How TBHG Works?
- 1 Each node in the graph represents a cluster of stay points.
- **2** Edges define directed transitions between locations.
- 3 Clusters represent semantically important locations (e.g., landmarks, popular spots).

ш Example:

- A node could represent "New York Central Park".
- Edges could represent travel between the park & nearby locations.
- Helps in predicting mobility patterns & optimizing location-based services.

Q HITS-Based Inference for Travel Patterns

- ✓ HITS Algorithm (Hyperlink-Induced Topic Search) is used to analyze travel importance.
- ★ Key Concepts:
- ✓ Hub Score: Measures how many routes pass through a location.
- Authority Score: Measures how important a location is based on connections.

ш Example:

- "Times Square" has a high authority score because it is a major landmark.
- "New York Subway Station" has a high hub score since many routes pass through it.
- 🚀 Why It Matters?
- Helps in ranking locations based on importance & accessibility.

Ш Mining Classical Travel Sequences

- ✓ The classical score of a travel sequence is calculated based on:
- **1 Sum of hub scores** for users who took this route.
- 2 Authority scores of locations in the sequence.
- 3 Probability that people follow this sequence in the future.

ш Example Calculation:

Travel Route	Hub Score	Authority Score	Probability Weight	Final Score
$A \rightarrow B \rightarrow C$	50	70	0.8	96
$D \rightarrow E \rightarrow F$	40	60	0.6	72



Helps in predicting preferred travel paths for smart recommendations.

Personalized Travel Recommendations

- ✓ Using mined travel sequences, a recommendation system can:
- ✓ Suggest optimal routes based on user preferences.
- ✔ Avoid high-traffic areas by analyzing mobility patterns.
- ✔ Recommend tourist attractions based on popularity & historical visits.

ш Example Application:

Google Maps suggesting personalized routes based on past travels.

Makes location-based services more intelligent & user-friendly.

Graded Activity: Designing a UC System for Clinical Leg Injury Patients

📌 Objective:

Design a **Ubiquitous Computing System** that helps **users with clinical leg injuries** by:

- ✓ Tracking their location trajectories.
- ✔ Recommending alternate/optimal travel routes.
- Submission Requirements:
- 1 Concept Sketch of the Interface (Visual Representation).
- 2 2 Use Cases (How the system benefits users).
- **3 2-4 Stakeholders** (Who benefits from the system?).
- 4 2-4 Key Features (Important functionalities).
- **5** List of Sensing Modalities (Which sensors will be used & why?).

@ Conclusion: The Future of GPS-Based Location Sensing

- Key Trends:
- ✓ AI-Powered Location Insights Combining GPS & AI for mobility forecasting.
- ✓ Privacy-Aware Location Tracking Secure location-based services.
- ✓ Wearable Tech & Location Sensing Personalized healthcare mobility tracking.
- ✓ Energy-Efficient GPS Technologies Reducing battery drain in location-aware apps.
- 💡 Final Thought:

Lecture 13: Motion & Activity Sensing in Ubiquitous Computing (UC)

Q Overview of the Lecture

This lecture focuses on **Motion & Activity Sensing** using **accelerometers and gyroscopes**, key components in **Ubiquitous Computing (UC)** that enable **motion tracking**, **tilt sensing**, **and activity recognition**.

- Key Topics Covered:
- ✓ Accelerometer Fundamentals & Earth's Gravity (g)
- ✓ Types of Accelerometers (Capacitive, Piezoelectric, Hall Effect, etc.)
- ✓ MEMS-Based Accelerometers & Their Working Principle
- ✔ Applications of Accelerometers (Vibration Detection, Impact Sensing, Smart Vehicles, etc.)
- ✓ Gyroscopes: Measuring Rotational Motion
- References for Further Reading:
 - Accelerometer & Gyroscope Working Principles
 - Accelerometer Lecture Notes UNC Charlotte

Understanding Accelerometers & Earth's Gravity (g)

- ✓ An accelerometer measures linear acceleration, which is the rate of change of velocity in one, two, or three axes (X, Y, Z).
- ✓ The acceleration due to Earth's gravity (g) at sea level is 9.81 m/s².

Ⅲ Reference Points for Acceleration (g-force) in Different Scenarios:

Scenario	Acceleration (g)
Earth's Gravity	1g
Passenger Car in Corner	2g
Race Car Driver in Corner	3 g
Bobsled Rider in Corner	5g

Scenario	Acceleration (g)
Human Unconsciousness	7 g
Space Shuttle Acceleration	10g

Example Question:

Q: What will be the reading of an accelerometer placed on a table vs. a free-falling object?

- \checkmark On the table: 9.81 m/s² (1g) due to gravitational force.
- \checkmark In free fall: 0 m/s² (0g) since both the object and accelerometer are falling at the same rate.
- Why It Matters?
- Accelerometers provide critical data for navigation, impact detection, and motion tracking.

📡 Proper Acceleration vs. Coordinate Acceleration

- ✔ Proper Acceleration: The actual acceleration experienced by an object relative to its own rest frame.
- ✓ Coordinate Acceleration: The change in velocity of an object relative to a specific reference frame.

📌 Example:

• An astronaut in free fall experiences **0g** (Proper Acceleration = 0), but an **observer on Earth** sees the astronaut accelerating due to gravity.

Why It Matters?

Essential for designing motion-tracking systems, aviation safety, and free-fall detection.

* Types of Accelerometers

Accelerometers use **different sensing mechanisms** to measure motion:

Capacitive Accelerometers

- ✓ Uses micromachined features that create capacitance changes when moving.
- ✓ Common in smartphones, game controllers, and wearables.
- Example:
 - iPhone tilt control (screen orientation switching).

2 Piezoelectric Accelerometers

- ✓ Uses piezoelectric crystals that generate voltage when subjected to motion or pressure.
- ✓ Ideal for high-frequency vibration sensing.
- 📌 Example:
 - Crash detection systems in airbag deployment.

3 Piezoresistive Accelerometers

- ✓ Uses beam-like structures whose electrical resistance changes with acceleration.
- ✓ More robust for shock sensing & industrial applications.
- Example:
 - Black box recorders in aircraft.

4 Hall Effect Accelerometers

- ✓ Uses magnetic field changes to detect motion.
- ✓ Highly stable for industrial motion sensing.
- 📌 Example:
 - · Magnetically guided factory robots.

5 Magnetoresistive Accelerometers

- ✓ Relies on changes in resistivity under a magnetic field.
- ✓ Less common but useful in specialized applications.
- 📌 Example:
 - Advanced automotive safety systems.
- Why It Matters?
- Each accelerometer type has specific advantages for different UC applications.

MEMS-Based Accelerometers

- ✓ MEMS (Microelectromechanical Systems) Accelerometers are the most widely used due to miniaturization & high precision.
- **#** How It Works:
- 1 Contains a proof mass (seismic mass) tethered to a substrate.
- 2 Sense fingers extend from the proof mass and move when acceleration is applied.
- 3 Stationary electrodes detect changes in capacitance due to motion.
- ш Example Applications:
- ✓ Fitness wearables detecting step count & running speed.
- ✓ Automotive crash sensors for impact detection.
- MEMS accelerometers are compact, cost-effective, and used in nearly all modern UC applications.

Applications of Accelerometers

- ✓ Tilt & Roll Detection Adjusting smartphone screens based on orientation.
- ✓ Vehicle Skid Detection Enabling smart braking systems (e.g., ABS).
- ✓ Impact Detection Airbag deployment in vehicles.
- ✓ Active Suspension Systems Keeping vehicles stable on rough roads.
- Example in Vehicles:
 - Tesla uses accelerometers for collision detection & self-driving calibration.
- ₱ Example in Smartphones:
 - Google Pixel uses accelerometers for motion-based gestures (flip to silence).
- Why It Matters?
- Accelerometers power motion-aware experiences in modern technology.

🔄 Gyroscopes: Measuring Rotational Motion

- ✓ Gyroscopes measure angular velocity (rotation speed) around an axis.
- ✓ Used for orientation tracking & motion stabilization.
- ★ Gyroscopes vs. Accelerometers:

Feature	Accelerometer	Gyroscope
Measures	Linear Acceleration	Rotational Motion
Detects	Tilt, Vibration, Impact	Yaw, Pitch, Roll
Example	Step Counting	VR Head Tracking

- Example Applications:
- ✓ Smartphone screen rotation.
- ✓ VR motion tracking (Oculus, PlayStation VR).
- ✓ Self-balancing robots & drones.
- Why It Matters?
- Gyroscopes are essential for stabilizing motion-sensitive UC applications.

© Conclusion: The Future of Motion Sensing in Ubiquitous Computing

- ✓ Al-powered activity recognition (smartwatches predicting workouts).
- ✓ Sensor fusion (combining accelerometers, gyroscopes, and magnetometers).
- ✓ Miniaturization of MEMS accelerometers for wearables.
- ✓ Advanced gesture recognition for AR/VR applications.
- Final Thought:
- "Motion sensing is transforming Ubiquitous Computing—powering everything from fitness trackers

to autonomous vehicles." 🚀



Q Overview of the Lecture

This lecture delves into two intertwined pillars of motion sensing in Ubiquitous Computing (UC): the **gyroscope**—a key device for measuring orientation and angular velocity—and **Human Activity Recognition** (HAR), which transforms raw motion data into semantic understanding of user behavior. We explore the physical principles behind mechanical and MEMS gyroscopes, define rotational axes (roll, pitch, yaw), and then unpack the end-to-end pipeline of HAR systems, including their characteristics, challenges, sensor modalities, and processing chain.

6 Gyroscopes: Measuring Orientation & Angular Velocity

Mechanical Gyroscopes

"A gyroscope is a device used for measuring or maintaining orientation and angular velocity."

A classical mechanical gyroscope comprises a spinning mass mounted on gimbals, allowing it to resist external torques and maintain its axis direction in space. This precession behavior (a change in orientation of the rotational axis) underpins its ability to serve as a stable reference for navigation and orientation maintenance in aerospace and marine applications.

NEMS Gyroscopes & the Coriolis Effect

"A Micro-Electro-Mechanical Systems (MEMS) gyroscope measures the angular rate by applying the theory of the Coriolis effect..."

- 1. A **proof mass** is suspended on micro-fabricated springs and driven to oscillate along the **x-axis**.
- 2. When the device experiences an angular velocity (ω) about the **z-axis**, the **Coriolis force** deflects the mass along the **y-axis**.
- 3. This minute displacement is detected capacitively, converted into an electrical signal proportional to rotation rate, enabling compact, low-power gyroscopes in smartphones, drones, and wearable devices.

Ⅲ Orientation Axes: Roll, Pitch & Yaw

"Rotation around the front-to-back axis is called roll. Rotation around the side-to-side axis is called pitch. Rotation around the vertical axis is called yaw."

- Roll (φ): Tilting side-to-side (e.g., an airplane banking).
- Pitch (0): Nodding up-and-down (e.g., raising/lowering a smartphone's front edge).
- Yaw (ψ): Turning left-to-right (e.g., a rotating platform).

These three **degrees of freedom** form the backbone of orientation tracking in navigation systems, augmented reality (AR), and mobile gaming.



Human Activity Recognition (HAR)

HAR leverages motion sensors to automatically infer the type and context of human activities (walking, typing, driving, etc.).

🏷 Key Characteristics of HAR Systems

1. Execution Mode

- Offline: Data is logged and processed in batches (e.g., daily activity summaries at midnight).
- Online: Real-time processing for interactive feedback (e.g., step counters notifying at 5 000 steps).

2. Generalisation

- User-Independent: Designed for many users with fixed thresholds (e.g., stress detection via EDA in a smart desk).
- User-Specific: Calibrated per individual, yielding higher accuracy but lower cross-user portability.

3. Recognition Mode

- Continuous: Automatically spots activities in streaming data (e.g., blink detection from ultrasound echoes).
- Isolated (Segmented): Assumes known start/end windows for gestures (e.g., 2 s window to distinguish blink vs. drowsiness).

4. Activity Types

- Periodic: Repetitive patterns (walking, cycling) analyzed with sliding windows and frequencydomain features.
- Sporadic: Occasional gestures interspersed within other activities; requires robust segmentation.
- **Static:** Postures or holding-still gestures (e.g., sitting vs. standing).

5. System Model

- Stateless: Recognizes primitives solely from sensor signals (e.g., reach, grasp).
- Stateful: Maintains contextual/environmental models (e.g., location map) to boost recognition at the cost of complexity.

Challenges in HAR

- Intraclass Variability: The same activity looks different across or even within individuals (e.g., walking speed variance).
- Interclass Similarity: Distinct activities produce similar sensor patterns (e.g., jogging vs. vigorous dancing).

Sensor Modalities for HAR

1. Ambient Sensors: Fixed-location devices (e.g., WiFi receivers, RFID readers) that infer activity from environmental signals.

2. Wearable Sensors: Body-mounted or handheld (e.g., accelerometers, gyroscopes, magnetometers, barometers) providing high-fidelity motion data.

Wearables dominate UC applications due to portability, lower setup cost, and personalized data.

Activity Recognition Chain (ARC)

An **ARC** transforms raw data into activity labels through:

- 1. **Data Acquisition:** Collect sensor streams (D).
- 2. **Signal Preprocessing & Segmentation:** Filter noise, split into windows (W_i).
- 3. Feature Extraction & Selection: Compute statistical, frequency, or time-domain features (Xi).
- 4. **Model Training:** Learn parameters (θ) for classification algorithms.
- 5. **Classification:** Assign confidence scores for each activity class Y_i.

Each stage must be carefully tuned to balance **latency**, accuracy, and **power consumption**—critical in battery-constrained UC devices.

Applications & Concrete Examples

- Fitness Trackers: Combine accelerometer & gyroscope data to detect running vs. cycling and count calories burned. 🏃 🚴
- Smart Vehicles: MEMS gyroscopes trigger airbag deployment on crash detection, while accelerometers adjust active suspension.
- Gesture Control: AR/VR headsets fuse gyroscope and magnetometer for smooth orientation tracking in virtual worlds.
- Emotion Sensing (): Analyzing micro-movements via smartphone gyroscopes to infer speaker's emotional state based on tremor patterns. ⊕ ⊗

References for Further Reading

- Mechanical & MEMS Gyroscopes: PMC Article on Gyroscope Principles
- Human Activity Recognition Surveys: "A Survey on Human Activity Recognition Using Wearable Sensors," ACM

Final Thoughts:

Motion and activity sensing form the foundation of context-aware UC systems. By mastering gyroscope physics and the full spectrum of HAR design—from sensor selection to real-time classification—designers can craft **seamless**, **intelligent** applications that respond fluidly to human behavior.

📡 Lecture 15: Motion & Activity Sensing in Ubiquitous Computing (UC) — Module V (Part II)

Q Overview of the Lecture

This lecture builds on foundational HAR concepts by examining a real-world smartphone dataset (UCI-HAR) and then broadening the sensing modality to WiFi-based activity detection. We will:

- Dissect the **UCI-HAR experimental protocol**, preprocessing steps, and feature extraction.
- Offer a **critical observation** and identify a **drawback** of this approach.
- Design an **exercise** to explore whether fine-grained activities (e.g., typing) can be sensed via accelerometers/gyroscopes.
- Introduce WiFi sensing fundamentals—MIMO, OFDM, and Channel State Information (CSI)—and survey its applications in human detection and fine-grained movement recognition.

Ⅲ UCI-HAR Dataset: Experimental Design & Feature Extraction

"Each person performed six activities (WALKING, WALKING_UPSTAIRS, WALKING_DOWNSTAIRS, SITTING, STANDING, LAYING) wearing a smartphone ... Using its embedded accelerometer and gyroscope, we captured 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50 Hz ... The sensor signals ... were pre-processed by applying noise filters and then sampled in fixedwidth sliding windows. ● 561-features were derived with time and frequency domain variables."

Activity Label	Description
WALKING	Level-ground ambulation
WALKING_UPSTAIRS	Ascending stairs
WALKING_DOWNSTAIRS	Descending stairs
SITTING	Seated posture
STANDING	Upright stationary
LAYING	Supine or prone stationary

- 1. Participants: 30 volunteers (19–48 years)
- 2. **Sensor Placement:** Samsung Galaxy S II on the waist
- 3. **Sampling:** 3-axis accel. & gyro @ 50 Hz, video-recorded for manual labeling
- 4. **Partitioning:** 70% subjects for training, 30% for testing
- 5. **Preprocessing:** Noise filtering + fixed-width sliding windows
- 6. **Features:** 561 features combining time- and frequency-domain descriptors



🤔 Critical Observation & Drawback

While the UCI-HAR dataset provides a rich feature set and a controlled labeling process, it suffers from limited ecological validity:

- Homogeneous Sensor Placement: Always on the waist—real users carry phones in pockets, handbags, or hand.
- Controlled Laboratory Environment: Activities performed in isolation, without real-world confounders (e.g., carrying objects, interacting with others).
- Participant Demographics: Only 30 volunteers; limited diversity in age, body-types, and movement styles.

Drawback: Such constraints can lead to **overfitting** on laboratory patterns and **poor generalization** to in-the-wild scenarios where sensor orientation and activity context vary widely.

Exercise: Detecting Typing Activity via ACC/GYRO

Can we extend HAR to **fine-grained activities** like typing?

- 1. Data Collection App: Build a simple MIT App Inventor app to stream and log accelerometer/gyroscope readings during typing vs. not-typing periods.
- 2. **Labeling:** Manually annotate each time window as **Typing** or **Not Typing**.
- 3. **Clustering Analysis:** Apply **K-means** to the collected feature vectors.

Cluster	Dominant Activity	Potential Overlap
Cluster 1	Typing	Minimal hand vibrations
Cluster 2	Not Typing	Ambient movements (e.g., shifts in posture)

- Inference: If clusters are well-separated, accelerometer/gyroscope data can capture the microvibrations of keystrokes. 🖳
- Caveat: Typing force and hand stabilization differ across users; clustering may blur boundaries without personalization.

📡 WiFi Sensing Fundamentals

'One important technology for the success of WiFi is Multiple-Input Multiple-Output (MIMO) ... Along with Orthogonal Frequency-Division Multiplexing (OFDM), MIMO provides Channel State Information (CSI) for each transmit and receive antenna pair at each carrier frequency."

- MIMO: Leverages multiple antennas to increase throughput and spatial diversity.
- OFDM: Splits a wideband channel into many orthogonal subcarriers, each carrying a portion of the data stream.
- Channel State Information (CSI): Fine-grained measurements of amplitude and phase across subcarriers—sensitive to multipath changes caused by human movement.

By monitoring real-time CSI variations, systems can infer human presence, position, and gestures without wearable devices—a paradigm known as Device-Free Passive (DfP) sensing.



WiFi-Based HAR & Fine-Grained Movement Detection

- 1. Human Presence & Counting: CSI fluctuations reveal occupancy levels in rooms (e.g., people entering/exiting).
- 2. Activity Recognition: Classify broad activities (walking, sitting, falling) by pattern-matching CSI time
- 3. Fine-Grained Gestures: Use high-resolution CSI to detect subtle motions—hand gestures, breathing patterns, even typing on a keyboard (with sub-millimeter displacement).

📌 Example Project:

- CSI-Enabled Gesture Control: IAMMOTIONS, which distinguishes eight hand gestures via SVM classifiers on CSI features.
- Vital Sign Monitoring: Extract respiratory and heartbeat rates by filtering low-frequency CSI oscillations induced by chest motion.

References for Further Reading

- UCI HAR Dataset:
 - https://archive.ics.uci.edu/dataset/240/human+activity+recognition+using+smartphones
- Wang et al., "Understanding CSI-Based Human Activity Recognition," ACM Computing Surveys

Key Takeaway:

By combining wearable-based HAR (accelerometers/gyroscopes) with infrastructure-based sensing (WiFi CSI), UC systems can achieve **ubiquitous**, **non-intrusive**, and **fine-grained** understanding of human motion unlocking next-generation applications in smart homes, healthcare monitoring, and human–computer interaction. 🚀

Lecture 16: Introduction to WiFi-Based Motion & **Activity Sensing**

Q Overview of the Lecture

This lecture transitions from wearable sensors to infrastructure-based sensing, demonstrating how commodity WiFi hardware can be repurposed for motion and activity detection. We cover:

- 1. WiFi Channels & Propagation Phenomena
- 2. Channel State Information (CSI) and its geometric significance
- 3. Key Enabling Technologies: MIMO & OFDM
- 4. **Comparison** with video-based sensing
- 5. CSI-Driven Sensing Applications, culminating in the WIFITUNED case study for online meeting engagement monitoring

📶 WiFi Channels & Propagation

"Wi-Fi most commonly uses the 2.4 GHz and 5 GHz radio bands. The frequency band is divided into multiple channels."

- **Channels:** Each band is partitioned into 20 MHz (or wider) channels to allow multiple networks to coexist.
- **Propagation Effects:** As radio waves travel, they undergo **reflection**, **diffraction**, and **scattering** off walls, furniture, and human bodies. These create **multiple propagation paths**, which superimpose at the receiver:

"The received superimposed signals carry the physical information of the signal propagation space."

• **Implication:** Any movement in the environment (e.g., a person walking) perturbs these paths, altering the composite signal received.

Channel State Information (CSI)

"CSI describes the propagation process of the wireless signal and therefore contains geometric information of the propagation space."

- **Definition:** CSI is a complex-valued matrix representing **amplitude** and **phase** shifts for each **subcarrier** and each **Tx–Rx antenna pair**.
- Static vs. Dynamic Scatterers:
 - **Static:** Walls, furniture (constant over time)
 - **Dynamic:** Human limbs, gestures (time-varying)

"The CSI observed by the receiver is added up with the portions contributed by the static (furniture, walls, etc.) and dynamic (arms, legs, etc.) scatterers."

- **Virtual Transmitters:** Each scatterer can be modeled as a "virtual Tx," enabling a rich multipath profile.
- **Sensing Potential:** Subtle changes in phase/amplitude across subcarriers can be mapped back to **motion trajectories, breathing rates,** or **gesture patterns**.

Enabling Technologies: MIMO & OFDM

MIMO (Multiple Input Multiple Output)

"Employs multiple antennas at both the transmitter and receiver to send and receive multiple data streams ... MIMO uses spatial multiplexing, which splits the data into smaller streams that are sent simultaneously over different antennas."

- Throughput Gain: Parallel streams boost data rates.
- **Spatial Diversity:** Different antenna paths experience independent fading; combining them improves reliability.

• Sensing Benefit: More Tx–Rx pairs yield higher-dimensional CSI, enhancing sensitivity to spatial movements.

2 OFDM (Orthogonal Frequency Division Multiplexing)

- Subcarriers: Divides the wideband channel into orthogonal narrowband carriers, each modulated separately.
- **Robustness:** Resilient to frequency-selective fading and inter-symbol interference.
- Sensing Benefit: Per-subcarrier CSI granularity allows resolution of fine temporal and spatial variations in the channel.

Comparison with Video-Based Sensing

Aspect	Video-Based	WiFi-Based
Intrusiveness	Requires cameras •	Uses existing WiFi infrastructure 🔽
Coverage	Line-of-sight required	Penetrates walls; non-line-of-sight
Privacy	High privacy concerns ♡	Lower visual privacy concerns ☺
Cost & Deployment	Cameras + lighting	Commodity WiFi NICs; minimal deployment
Data Modality	Pixel intensities	RF amplitude & phase (CSI)

"Non-intrusive, Pervasive, Low-cost"

Both pipelines share stages—data capture, preprocessing, feature extraction, learning/classification but WiFi sensing excels when privacy and cost are paramount.

CSI-Based WiFi Sensing Applications

1. Human Presence & Counting

CSI fluctuations correlate with the number of occupants; useful for smart buildings.

2. Activity Recognition

• Macro activities (walking, sitting, falling) distinguished via time-series CSI patterns.

3. Fine-Grained Gesture Detection

• Hand gestures (swipes, pinches) resolved with sub-millimeter precision by tracking phase variations. 💅

4. Vital Sign Monitoring

 Minute chest movements from breathing or heartbeat modulate CSI low-frequency components.

A non-intrusive system for monitoring online meeting engagement via head movements:

"Monitoring engagement in online meetings in indoor environment"

- Modality Correlation: Correlates CSI changes with head nods, turns, and facial orientation, alongside audio signals.
- Pipeline:
 - 1. **CSI Acquisition:** AP continuously samples CSI.
 - 2. **Signal Correlation:** Align CSI dips/peaks with audio cues and known head-movement signatures.
 - 3. **Engagement Metric:** Computes engagement scores in real time, enabling feedback or moderation tools.

Live demos have shown robust detection of engagement shifts—ideal for remote education and large-scale webinars. ▶



Key Takeaways & Future Directions

- Infrastructure Reuse: WiFi NICs double as motion sensors—zero additional hardware cost.
- **Multi-Modal Fusion:** Combining CSI with audio, IMU, or camera data can boost accuracy for complex tasks.
- **Edge Deployment:** Lightweight CSI processing on edge devices enables real-time feedback in smart homes and offices.
- Research Frontiers:
 - **Deep Learning on CSI:** End-to-end CNNs/Transformers for robust feature learning across varied environments.
 - **Privacy-Preserving Sensing:** Ensuring that no personally identifiable information (e.g., gait, identity) can be reconstructed.

"WiFi sensing transforms everyday access points into pervasive motion-and-activity sensors—unlocking a new dimension of context-aware computing."

♥ Lecture 17: Physiological Sensing in Ubiquitous Computing (UC) — Module VI (Part I)

Q Overview of the Lecture

This lecture introduces **physiological signals**—biological processes generating measurable electrical, mechanical, or chemical changes—and explores how they can be captured and leveraged in UC systems. We cover:

- 1. **Definitions & Examples** of key physiological signals
- 2. **Recording Modalities**: wearables, smartphones, and novel devices
- 3. **Applications** of physiological sensing in health, wellness, and HCI
- 4. Deep Dive: Photoplethysmography (PPG)
 - Principles of PPG waveforms

- Probe types: transmission vs. reflection
- Derived physiological parameters
- Guidelines for HR and BP measurement

5. Case Study: Detecting Divided Attention in Mobile MOOC Learning

1. What Are Physiological Signals?

Physiological signals reflect the body's internal state through measurable changes:

- **Electrocardiography (ECG):** Cardiac electrical activity
- **Electroencephalography (EEG):** Brainwave electrical activity
- Electromyography (EMG): Muscle electrical potentials
- **Electrooculography (EOG):** Eye movement potentials
- Photoplethysmography (PPG): Blood-volume-induced light modulation
- **Respiratory Signals:** Chest/abdominal movement or airflow
- Blood Oxygen (SpO₂): Oxygen saturation via optical absorption
- Continuous Glucose Monitoring (CGM): Interstitial glucose levels
- Body Temperature: Core or skin temperature variations
- **Sweat Analysis:** Electrolyte/biomarker concentrations

∠ 2. How to Record Physiological Signals?

Wearables & Devices

- Rings & Watches: Ultrahuman Ring, Empatica Embrace
- Chest Straps: Polar H10 heart-rate monitor
- **Smartphones:** Cameras (PPG), IMUs (EOG inference via blink detection)
- **Emerging Form Factors:** Smart gloves (EMG), smart masks (respiration), smart shoes (gait), smart glasses (EOG)

These modalities trade off **accuracy**, **comfort**, and **ubiquity**—key considerations when integrating physiological sensing into UC applications.

\$ 3. Applications of Physiological Sensing

- **Health Monitoring:** Arrhythmia detection (ECG), sleep apnea screening (respiratory), glucose regulation (CGM)
- **Fitness & Wellness:** Heart-rate zones (PPG), stress evaluation (skin conductance + HRV), breathing exercises
- Emotion & Cognitive State: Stress, workload via EDA/HRV; attention, drowsiness via EEG/EOG
- Authentication & Security: Gait patterns (IMU + PPG), cardiac signature biometrics
- Human-Computer Interaction: EMG-driven prosthetic control, EOG-based gaze interfaces

These applications illustrate the **ubiquity** and **transformative potential** of physiological data in personalized, context-aware UC systems.



4. Deep Dive: Photoplethysmography (PPG)

4.1 Principle of PPG

"The photoplethysmographic (PPG) signal is defined as oscillations in light transmission through a tissue, which are created by heart beats (Systoles and Diastoles)."

- **Optical Path:** LED illuminates tissue → light is **absorbed** and **scattered** by blood/tissue → photodiode measures **attenuated intensity** → converted to voltage waveform.
- Frequency Components:
 - High-frequency (AC): Heartbeat-induced blood-volume changes
 - Low-frequency (DC): Baseline tissue properties, respiration, thermoregulation

4.2 PPG Probe Types

Mode	Configuration	Use Case		
Transmission	LED → tissue → photodiode on opposite sides (clip-based)	Medical SpO₂ on finger, earlobe (arterial sites)		
Reflection	LED & photodiode side-by-side, millimeters apart	Wearables (wrist, forehead), non- contact PPG		
• Transmission Probes:				

- Pros: High signal quality for arterial oxygen saturation
- Cons: Limited to clip-able sites (finger, earlobe)
- Reflection Probes:
 - Pros: Versatile placement (wristbands, smartwatches)
 - Cons: More susceptible to motion artifacts

4.3 Arterial Waveform Structure & Derived Metrics

A typical PPG waveform contains:

- 1. Systolic Peak (P_1) : Maximum upstroke of each cardiac cycle
- 2. **Dicrotic Notch & Wave (P₂):** Valve closure reflections
- 3. Baseline Drift: Slow variations from respiration/venous flow

Extractable Parameters:

- **Heart Rate (HR):** Inter-beat intervals from P₁ timestamps
- Heart Rate Variability (HRV): Time- and frequency-domain variability metrics
- Atrial Fibrillation Detection: Irregular P₁ patterns
- Respiratory Rate: Modulation of baseline amplitude
- Blood Pressure Estimation: Pulse Transit Time (PTT) between ECG & PPG peaks
- Arterial Stiffness: Pulse wave velocity analysis

4.4 Measuring HR vs. BP: Parameter Guidelines

Parameter	Heart Rate Monitoring	Blood Pressure Estimation
Location	Wrist	Finger (or ear)
Mode	Reflectance	Transmission
Sampling Rate	Low (~30–40 Hz)	High (>500 Hz for waveform fidelity)
Wavelength	Green/Red LED (525–600 nm)	Infrared (~940 nm)
Recording Time	Short (10–30 s)	Long (5–20 min for cuff-less BP models)

• Why? HR detection tolerates lower sampling rates and motion artifacts, while reliable BP estimation requires high-fidelity waveform features and stable optical coupling.

Problem Context

Mobile MOOCs offer flexibility but increase **multitasking** and **distractions**, impairing learning outcomes.

OneMind Contribution

Proposes a system to **detect presence, type, and intensity of divided attention** via implicit sensing on unmodified smartphones.

- **Physiological Signals:** Combines PPG-derived HRV, gyroscope/accelerometer patterns of device handling, and screen-interaction logs.
- Learning Impact: Correlates attention lapses with quiz performance and retention metrics.

Solution Pipeline

1. Data Acquisition:

- PPG via camera+flash
- IMU streams for device motion
- Touch-event timestamps

2. Feature Extraction:

- HRV indices (SDNN, RMSSD)
- Motion entropy, device-posture changes
- Interaction frequency (taps/swipes per minute)

3. Modeling & Detection:

- Machine learning classifiers trained on **focused vs. distracted** sessions
- Real-time attention score output to adaptive UI interventions

Key Results

- **Divided Attention** significantly degrades quiz accuracy by up to **15%**.
- OneMind achieves 85% detection accuracy for high vs. low attention states.



Class Activity: Critical Evaluation

Evaluate OneMind on:

- 1. **Usability:** Intrusiveness of camera-based PPG? Battery impact?
- 2. Accuracy: Robustness across lighting conditions and device models
- 3. Accessibility: Can users with disabilities (e.g., tremor) leverage the system?
- 4. **Ubiquity:** Dependence on smartphone sensors—availability across device classes
- 5. Scalability & Robustness: Performance under network latency, large-scale MOOCs

For each limitation, propose improvements (e.g., fusing additional low-power sensors, adaptive calibration, privacy-preserving on-device ML).



Final Thoughts:

Physiological sensing unlocks deep insights into human health, cognition, and behavior. From PPG's rich cardiovascular signals to camera-free EMG/EOG inference, UC designers can create context-aware systems that adapt to our biological states—ushering in a truly human-centric computing era.



📡 Lecture 18: Electrodermal Activity (EDA) in Physiological Sensing

Q Overview of the Lecture

This lecture dives deep into Electrodermal Activity (EDA), also known as Galvanic Skin Response (GSR)—a window into the **sympathetic nervous system's** involuntary arousal. We will explore:

- What is EDA? physiology & stimulus drivers
- How EDA is measured exosomatic vs. endosomatic methods
- **Signal structure** tonic vs. phasic components
- **Key parameters** SCL, SCR, latency, amplitude, NS.SCR
- Applications from VR stress sensing to biofeedback gaming
- **Devices & confounds** hardware choices and environmental factors
- Psychological constructs arousal, stress, orienting response, cognitive load
- Threats to validity electrode placement, motion artifacts, skin prep



1. What Is Electrodermal Activity?

- Definition: EDA reflects changes in skin conductance due to sweat gland activity on palms and soles —driven by **sympathetic arousal** in response to sensory or emotional stimuli.
- Autonomic Response: Completely involuntary, modulated by touch, sight, sound, odor, or taste triggers.

② 2. Measuring EDA: Exosomatic vs. Endosomatic

1. Exosomatic Approach:

- Applies a constant voltage (DC or AC) across skin electrodes.
- Measures skin resistance/conductance (DC) or admittance/impedance (AC).

2. Endosomatic Approach:

- Records the skin's **natural electrical potential** without external excitation.
- Uses one electrode on an **active** site (e.g., palm) and one on an **inactive** site (e.g., forearm).

∠ 3. EDA Signal Structure: Tonic & Phasic Components

- Tonic Component (Skin Conductance Level, SCL):
 - Baseline conductance in absence of discrete events.
 - Varies slowly with hydration, temperature, and psychological state.
- Phasic Component (Skin Conductance Responses, SCR):
 - Short-term bursts ("peaks") following stimuli (anticipation, decision-making).
 - Characterized by **amplitude**, **latency**, **rise time**, and **recovery time**.
- Non-specific SCR (NS.SCR):
 - Phasic peaks not clearly linked to an identifiable stimulus.

③ 4. Key EDA Parameters

Parameter	Definition
SCL	Mean tonic conductance over a rest period
SCR Amplitude	Peak conductance increase after a stimulus
Latency	Time from stimulus onset to SCR onset
Rise Time	Time from SCR onset to peak
Recovery Time	Time from peak back to baseline
SCR Frequency	Number of phasic peaks per minute (tonic arousal measure)

5. Applications of EDA

- VR & Gaming: Monitor anxiety, presence, and tune game mechanics via biofeedback loops.
- **User Experience Research:** Assess emotional enjoyment and orienting responses in multi-sensory environments.
- **Stress & Workload Monitoring:** NS.SCR frequency as a real-time stress index; SCRs track cognitive resource demands.
- Adaptive Interfaces: Games where players intentionally modulate their EDA to influence in-game events.

6. Commonly Used Devices & Confounds

- Hardware: Biopac GSR amplifiers, Empatica E4, Shimmer3 GSR+ modules
- Environmental & Physiological Effects:
 - Caffeine/Medication, Temperature & Humidity impact EDA baselines.
 - Movement Artifacts: Grasping, speech, running can destabilize electrodes and heat skin, skewing conductance.

7. Psychological Constructs Underpinning EDA

- Arousal: State of being awake and reactive; indexed by SCL and SCR frequency.
- Stress: Sympathetic-driven; NS.SCR frequency correlates with stress levels.
- Orienting Response: Novelty detection via phasic SCRs to new stimuli—amplitude reflects stimulus intensity.
- Cognitive Load: Task-imposed demand; SCR is a reliable proxy for mental resource consumption.

\triangle 8. Threats to Validity

- Skin Preparation & Electrode Placement:
 - Wrist recordings may reflect **thermoregulatory** rather than emotional sweating—validation studies report **poor reliability** on the wrist.
- **Physical Activity:**
 - Electrode movement and heat from exercise (walking, instrument play) can produce spurious conductance changes.

Final Thought:

Electrodermal Activity offers a non-invasive, continuous measure of autonomic arousal, with vast applications from affective computing to adaptive user experiences. By mastering EDA's physiological basis, signal structure, and confounds, designers can build truly **emotion-aware** systems in ubiquitous computing.



Lecture 19: Wearable Computing & Smart Systems

Q Overview of the Lecture

Lecture 19 introduces the domain of wearable computing and smart textiles, exploring:

- 1. Key Attributes that define successful wearables
- 2. Smart Clothing Scenarios & their unique design challenges
- 3. **SeamFit**—a state-of-the-art smart T-shirt system for exercise logging
- 4. Early Wearable Prototypes and their evolution
- 5. **Broad Applications** of wearables beyond health
- 6. Fundamental Challenges—power, heat, networking, privacy, and interfaces

1. Key Attributes of Wearable Computing

"Persist and provide constant access to information services ... Sense and model context ... Adapt interaction modalities ... Augment and mediate interactions with the user's environment."

A truly wearable system must:

- Be Always-On & Effortless: Daily use without conscious donning; seamless integration into clothing or accessories.
- **Sense & Model Context:** Continually observe the user's environment, physical/mental state, and internal device status to remain relevant.
- Adaptive Interaction: Shift modalities (visual, auditory, haptic) based on situational context—e.g., silent alerts during meetings.
- Augment & Mediate: Automatically gather location-relevant resources and filter information to match user preferences.

 ↑

These attributes differentiate wearables from mobile devices by emphasizing **passive**, **continuous**, and **context-aware** operation.



👕 2. Smart Clothing: Scenario & Challenges

Use Case: Kate's Smart T-Shirt logs workouts and survives laundry just like a regular shirt.

Design Challenges include:

- **Generalizability Across Fits:** Garments stretch and drape differently on each body—sensor placement must remain accurate for joints regardless of size.
- Washing Durability: Electronics and conductive threads must endure repeated cycles of water, detergent, and mechanical agitation without degrading.
- **User Comfort & Breathability:** Sensor integration must not compromise fabric hand, stretch, or moisture wicking—critical for active wear.

3. SeamFit: A Seam-Based Smart Clothing Solution

SeamFit repurposes existing seam lines in garments as electrodes, embedding conductive threads invisibly:

- 1. **Invisible Integration:** TPU-coated silver-nylon threads run along shoulders, sleeves, and torso seams —preserving aesthetics and fabric properties.
- 2. **Seam-Joint Correlation:** Seam placement aligns naturally with body joints, enabling **fine-grained motion sensing** without bulky sensors.
- 3. **Minimal Fabric Impact:** Breathability and stretch remain unchanged, maintaining user comfort during exercise.

In a 15-participant study covering 14 exercises, SeamFit achieved **89% detection accuracy**, **93.4% classification accuracy**, and an average count error of **0.9 reps**, **independent** of user body type, wash cycles, and garment fit.

Critical Evaluation:

· Strengths:

- **Robustness:** High accuracy across real-world variations (fits, washes).
- Transparency: Users interact with familiar garments—no learning curve.
- **Durability:** TPU coating protects conductive threads during laundering.

• Limitations & Improvements:

- **Electrode Wear:** Over many washes, silver-nylon may fatigue—consider plating or encapsulation enhancements.
- **Power & Data Modules:** SeamFit offloads processing to removable units; future work could miniaturize and embed energy harvesting (e.g., textile solar cells).
- **Scalability:** Custom seam patterns per garment design; a standardized seam-sensor kit could reduce manufacturing complexity.

4. Early Prototypes: From Wearable Multimedia to Modern Smartwear

The pioneering **wearable multimedia computer** integrated sensors and displays into eyeglasses, coupled with a head-mounted screen and wireless connectivity—an ancestor of today's AR glasses.

- Lessons Learned:
 - **Ergonomics & Weight:** Early rigs were bulky; miniaturization has been key.
 - **Interface Design:** From on-lens keyboards to voice and gesture controls, wearables demand novel I/O paradigms.

5. Broader Applications of Wearable Computing

Beyond fitness and AR, wearables now span:

- Healthcare Monitoring: Continuous ECG, SpO₂, glucose sensing for chronic disease management. •
- Occupational Safety: Real-time posture and fatigue alerts for industrial workers. 🦺
- Interactive Entertainment: Haptic suits and gesture-sensing gloves for immersive VR/AR. 🙈
- Environmental Sensing: Smart jackets that adjust insulation or filter air based on pollution levels.

(See demonstrations: j8uLddsKROU, EQ-_bWjDKC4, giYfG9DmNxE on YouTube)

\triangle 6. Fundamental Challenges of Wearable Systems

"Power use, Heat Dissipation, Networking, Privacy, Interface Design" — Leonard Foner.

- **Power & Heat:** Continuous sensing and wireless comms demand energy; thermal comfort must be maintained.
- **Networking & Latency:** Seamless handoff between BLE, Wi-Fi, and cellular to ensure low-latency data.

 data.
- **Privacy & Ethics:** Wearables often collect sensitive biometric and location data—designers must minimize data collection and secure personal information by default. **9**

• **Interface Innovation:** Micro-displays, bone conduction audio, and haptics require new design patterns to avoid information overload.

🚀 Final Thought:

Wearable computing fuses **textiles**, **electronics**, and **machine intelligence** into everyday garments and accessories. By addressing the twin pillars of **user comfort** and **robust sensing**, innovators can create truly **pervasive**, **invisible** computing platforms that augment human abilities—ushering in a future where technology is not worn on the body, but woven into it.

Lecture 20: Electroencephalography (EEG) in Physiological Sensing — Module VI (Part II)

Q Overview of the Lecture

This lecture delves into **Electroencephalography (EEG)**—the non-invasive measurement of scalp electrical potentials generated by neuronal populations. We will examine:

- 1. Action Potentials & Postsynaptic Potentials
- 2. **EEG Signal Generation** & noise considerations
- 3. **EEG Frequency Bands** and their cognitive correlates
- 4. **Electrode Montages** (10–20 system) for spatial coverage
- 5. Commercial EEG Systems
- 6. **Design Exercise**: Conceptualize a distraction-monitoring EEG device

1. From Action Potentials to EEG

"An action potential is a series of quick changes in voltage across a cell membrane."

- **Resting Membrane Potential (RMP):** Neuronal cell bodies maintain approximately **–60 to –70 mV** internally relative to the extracellular space.
- Action Potential (AP): When depolarization reaches threshold, a rapid voltage spike (~ +40 mV) travels along the axon—triggering excitatory postsynaptic potentials (EPSPs) or inhibitory postsynaptic potentials (IPSPs) in downstream neurons.
- **Summation Requirement:** Only **synchronous activity** of millions of neurons can produce scalp-measurable potentials due to skull attenuation (~100×).

2. EEG Signal Generation & Noise

- **Source:** Predominantly large pyramidal cells in the **cerebrum**, oriented perpendicular to the cortical surface.
- **Attenuation & Filtering:** The skull and scalp dampen high-frequency components; low-amplitude signals (< 20 µV) require high-gain, low-noise amplifiers.
- Noise Sources:
 - Internal: Muscle activity (EMG), cardiac signals (EKG)

- External: Power-line interference (50/60 Hz), electrode movement artifacts
- Implication: Rigorous shielding, impedance matching, and artifact rejection are critical for reliable EEG.

3. EEG Frequency Bands & Cognitive States

Band	Range (Hz)	Functional Correlate
Delta	0.5–4	Deep sleep, unconscious states (may appear in waking pathology)
Theta	4–7.5	Drowsiness, meditation, memory retrieval, creative insight
Alpha	8–13	Relaxed wakefulness (eyes closed), inhibitory cortical idling
Beta	14–30	Active thinking, focused attention; high-beta (> 30 Hz) in anxiety/panic
Gamma	30–100+	Sensory binding, high-level cognitive processing, consciousness integration

These rhythms reflect **population-level oscillations** arising from thalamo-cortical and cortico-cortical loops.

7 4

4. Electrode Positions: The 10–20 System

- Labels:
 - Fp (Frontopolar), F (Frontal), C (Central), T (Temporal), P (Parietal), O (Occipital), A
 (Auricular), Z (Midline)
- **Placement:** Electrodes are spaced at 10% or 20% intervals of skull landmarks (nasion, inion, preauricular points) to ensure reproducibility.
- Montages:
 - **Referential:** Each active site referenced to a common site (e.g., mastoid).
 - **Bipolar:** Adjacent electrodes compared—useful for localizing focal activity.

☆ 5. Commercial EEG Devices

- **Emotiv EPOC:** 14-channel wireless headset—consumer-grade, good for BCI prototyping.
- g.tec g.Nautilus: 32–64 channels, medical-grade, supports online processing.
- **Enobio 8:** 8-channel wearable, dry electrodes for ambulatory monitoring.

Each balances **channel count**, **portability**, and **signal fidelity**—key trade-offs when selecting hardware for specific applications.

6. Graded Activity: Designing a Distraction-Monitoring EEG Device

Objective: Create a wearable EEG system that detects user distraction during tasks (e.g., studying, driving).

- 1. Sensor Array & Placement:
 - Frontal (Fz, F3, F4) and central (Cz) electrodes to capture theta/beta shifts associated with attentional lapses.

2. Signal Processing Pipeline:

- **Preprocessing:** Bandpass filter (1–40 Hz), notch filter at mains frequency (50 Hz).
- Feature Extraction: Theta/Beta ratio, event-related potentials (e.g., P300 amplitude).
- Classification: Lightweight ML model (e.g., SVM) on edge device (e.g., embedded ARM MCU).

3. User Feedback Loop:

- **Haptic Alert:** Gentle vibration when distraction threshold exceeded.
- Adaptive Interface: Pause content playback or adjust difficulty when sustained distraction detected.

4. Differentiators vs. Traditional Systems:

- Form Factor: Comfortable, behind-the-ear dry electrodes—no gel, low maintenance.
- **On-Device Processing:** Protects privacy by avoiding raw EEG transmission.
- Context Integration: Fuse IMU data (head orientation) to distinguish head nodding vs. device movement.

Why Users Will Adopt: Unlike lab-bound EEG, this system offers seamless wearability, real-time alerts, and privacy-preserving on-device processing—empowering users to maintain focus without cumbersome setups. 🚀

Key Takeaway:

EEG translates microscopic neuronal events into macroscopic oscillations measurable at the scalp. By understanding action potentials, signal generation, frequency bands, and practical electrode montages, designers can craft wearable brain-computer interfaces that monitor cognitive states—paving the way for attention-aware, adaptive ubiquitous systems.

🐈 Lecture 21: Affective Computing — Module VII (Part I)

Q Overview of the Lecture

Lecture 21 lays the foundation for **Affective Computing** by exploring how technology can **detect**, **interpret**, and **respond** to human emotions. We examine:

- 1. **Definition & Scope** of Affective Computing
- 2. Emotions as Cognitive & Physical Phenomena
- 3. **Sentic Modulation**: the natural, subtle expressions of emotion
- 4. Types of Sentic Modulation: facial, vocal, motor, physiological
- 5. **Design Considerations** for emotion-aware UC systems
- 6. Cognitive Appraisal of Emotions
- 7. Human Expressions & Facial Action Coding
- 8. Case Study ExpresSense: Smartphone-based facial expression sensing via acoustics
- 9. Experimental Findings & Usability
- 10. Opportunities & Challenges for real-world deployment

1. What Is Affective Computing?

Affective Computing is "the expanding intersection between technology and emotion," focused on **detection** and **response** to human feelings.

- **Cognitive Aspect:** Emotions involve appraisal, comparison, and inference—how situations trigger emotional states.
- **Physical Aspect:** Emotions manifest in **body signals** (heart rate changes, skin conductance) and **sentic modulations** (facial expression, voice inflection).

🥦 2. Emotions: Cognitive vs. Physical

- **Cognitive Emotions:** Involve mental processes—e.g., interpreting that one's wheelchair collision was accidental shifts anger to sympathy.
- **Physical Emotions:** Encompass bodily responses—posture changes, facial micro-expressions, vocal tone shifts—often subconscious.



3. Sentic Modulation: The Body's Emotional Language

"Sentic modulation, such as voice inflection, facial expression, and posture, is the primary means of communicating human emotion." — Manfred Clynes, 1977

- Natural & Subconscious: We rarely control micro-expressions or subtle posture shifts consciously.
- Basic Emotions & Unique Patterns: Each basic emotion (joy, anger, sadness, surprise, fear, disgust) has signature muscle activations.

🎨 4. Types of Sentic Modulation

4.1 Facial Expressions © ⊕ ⊕ ⊕

- Governed by Facial Action Units (AUs) (e.g., AU12: lip corner puller for smiling).
- **Contextual Rules:** Social display rules modulate expression intensity—e.g., in a classroom vs. sports arena.
- **Real-World Challenge:** Sports contexts may amplify expressions, risking false positives if a model is trained only on neutral settings.

4.2 Vocal Intonation 🎤

- "How" Over "What": Affective systems must parse pitch, energy, and rhythm beyond lexical content.
- Applications: Synthetic voices (e.g., virtual assistants) can project empathy by modulating tone.
- Challenge: Background noise and speaker variability complicate robust intonation analysis.

4.3 Motor Forms & Essentic Gestures 🕺

- Arbitrary motor outputs (e.g., hand flutters) can express emotion.
- **Sentograph Devices** capture pressure trajectories over time, mapping spatiotemporal gestural "sentic forms."

4.4 Physiological Responses □□

- Heart Rate Variability, EDA, respiration co-occur with emotional states.
- **Integration:** Physiological signals can disambiguate expressions that look similar (e.g., surprise vs. fear).

5. Designing Emotion-Aware UC Systems

When building real-world affective systems, account for:

- 1. **Emotion Intensity:** Subtle vs. overt expressions demand different sensitivity thresholds.
- 2. **Emotion Type Granularity:** E.g., distinguishing affectionate vs. passionate love.
- 3. Induction Context: Real vs. imagined vs. cinematic scenarios influence expression authenticity.
- 4. Display Rules: Cultural or situational norms may encourage suppression or exaggeration.
- 5. User Factors: Mood, medication, diet, and device context (lighting, background noise).

6. Cognitive Appraisal of Emotions

- Primary Emotions: Immediate responses to stimuli (e.g., fear at loud noise).
- **Secondary Emotions:** Cognitive evaluations layered atop primary reactions (e.g., embarrassment following fear).
- **System Implication:** Models must incorporate context and user history to avoid misclassification (e.g., startled by tech failure vs. malicious attack).

7. Human Expressions & Facial Action Coding (FACS)

- **Ekman's Six Basic Emotions:** Happiness, Sadness, Anger, Fear, Disgust, Surprise (+ Contempt later added).
- Microexpressions vs. Macroexpressions:
 - Micro: <0.5 s, involuntary, reveal true emotion under suppression.
 - Macro: >0.5 s, under conscious control.
- **FACS:** Decomposes any facial movement into combinations of AUs, enabling fine-grained expression analysis.

★ 8. Case Study: ExpresSense

Goal: Infer user engagement from facial expressions via acoustic sensing on an unmodified smartphone.

8.1 System Architecture

- Acoustic Capture: Smartphone microphone records near-ultrasonic reflections from facial movements.
- 2. Signal Processing:
 - Bandpass filtering (<20 kHz) to isolate subtle facial-motion echoes.
 - Feature extraction of amplitude and frequency shifts.
- 3. **Classification Pipeline:** ML models map acoustic features to expression labels (Angry, Happy, Sad, Surprise).

4. Integration: Results feed into a video-streaming app interface, displaying engagement metrics in real time.

8.2 Experimental Setup & Results

- Controlled Expressions: 10 participants, 3 sessions per emotion, 6 880 samples total.
- Accuracy: Individual expression classification achieved up to 90%+ for some emotions.
- Environmental Sensitivity:
 - **Distance & Elevation:** Minor performance dip beyond 50 cm or >15° tilt.
 - Motion & Finger Interference: Gross head movements degrade accuracy more than ambient noise.
- Natural Expressions: For spontaneous reactions, accuracy gravitates around 75–80%, reflecting realworld complexity.

8.3 Usability Study

- **Participants:** 72 users evaluating the video-streaming prototype.
- **SUS Score: 85.3/100**—indicating high perceived usability and acceptability.



🌱 9. Opportunities & Challenges

- Opportunities:
 - **Device-Free Sensing:** Leverages existing hardware—no wearables required.
 - **Cross-Modal Fusion:** Combining acoustics, vision, and physiology can boost robustness.
- Challenges:
 - Interference: Speech and ambient sounds can mask ultrasonic echoes.
 - **Obstructions:** Glasses, masks, or facial hair alter reflection patterns.
 - **Privacy:** Inferring emotion acoustically raises consent and data-use concerns.

🚀 Final Thought:

Affective Computing bridges the gap between **cold technology** and **human warmth**. By harnessing **sentic** modulation, physiological cues, and innovative sensing (like ExpresSense), ubiquitous systems can become truly **emotion-aware**, fostering richer, more empathetic human–computer interactions.

EEGlass: Toward Everyday EEG-Eyewear for **Brain-Computer Interaction**



Context & Motivation

EEGlass emerges at the intersection of two trends: the rapid miniaturization of head-mounted displays (HMDs) into socially acceptable eyewear, and the quest to take EEG out of the lab and into daily life. Rather than bulky EEG caps or hidden lab rigs, EEGlass embeds electrodes into normal-looking frames, aiming to capture meaningful brain signals—resting rhythms, motor-intention desynchronizations, and eye-movement artifacts—while you go about your routine.

☆ Prototype Design & Hardware

• Form Factor: Consumer-style plastic frames that can house a Google Glass-type module for familiarity and social comfort.

- Electrode Layout:
 - Central Bridge Contact: One electrode at the glabella (just above the nose) captures frontal signals.
 - Mastoid Contacts: Two electrodes behind each ear tap into temporal/parietal activity.
 - Reference & Ground: Nasal-bridge pads serve as common reference points.
- Electronics: An open-source OpenBCI Cyton board provides 8 channels, an inertial sensor, local storage, Bluetooth connectivity, and Arduino-style programmability.

Together, these choices trade off full-cap spatial resolution for wearability and cost-effectiveness, while still targeting key EEG bands and eye-movement potentials.



Signal Acquisition & Processing

- Resting State Protocol: Alternating eyes-open vs. eyes-closed windows reveal classic alpha peaks around 8–12 Hz in both EEGlass and a medical-grade 8-channel system.
- Motor Intention Task: Using a two-class hand-movement paradigm, EEGlass detects event-related desynchronization (ERD) in alpha/beta bands, albeit at lower amplitude compared to scalp-centered electrodes—confirming weaker but usable signals.
- EOG Detection: Subtle shifts in skin potential from blinks and saccades manifest cleanly in the ultrasonic-free EEG channels, opening doorways to integrated eye-tracking.

Data flows from raw biopotential \rightarrow high-pass and notch filtering \rightarrow power spectral estimation for rhythms → ERD computation for movement.

Performance & Findings

- Alpha Rhythm Capture: EEGlass matches baseline systems in identifying the alpha peak during rest, showing that even peripheral contacts can sense global cortical oscillations.
- Motor-Action ERD: Though ERD magnitude is attenuated, EEGlass reliably distinguishes left vs. right hand preparation through lateral electrode pairs—laying groundwork for simple motor BCIs.
- Eye-Movement Signals: The single-degree-of-freedom EOG channel registers directional eye motions and blinks with sufficient clarity for blink detection or coarse gaze estimation.

Together, these results validate EEGlass's core promise: unobtrusive capture of multiple neural signatures in real time.

🙎 Strengths & Innovations

- Social Acceptability: Frames look like regular glasses—wearable in public without stigma.
- Open Platform: Leverages community-driven OpenBCI hardware/software, encouraging hackability and transparent research.

• Multi-Modal Potential: Combines EEG, EOG, and IMU in one form factor, enabling richer interfaces and context-aware applications.

△ Limitations & Challenges

- **Spatial Resolution:** Skull-peripheral electrodes cannot localize deep or fine-scale cortical activity—limiting advanced BCI paradigms.
- **Signal Attenuation:** Brain signals must traverse more tissue before reaching electrodes, reducing signal-to-noise and demanding careful filtering.
- **Fit & Stability:** Everyday movement or ill-fitting frames can break contact, creating artifacts or data loss
- **User Diversity:** Head shapes and skin properties vary; electrode placement may need adaptive sizing or adjustable mounts.

Future Directions

- **Machine Learning Calibration:** Train personalized models that map EEGlass inputs to lab-grade signals, compensating for attenuation and noise.
- **Miniaturized Electronics:** Integrate ultra-low-power amplifiers and battery into the temple arms, eliminating external modules.
- **Hybrid Sensing:** Fuse eye-motion, brain rhythms, and head orientation to create intuitive hands-free controls for AR/VR interfaces.
- **Longitudinal Studies:** Deploy across diverse users and use cases—cognitive load monitoring, wellness feedback, adaptive gaming—to assess robustness in the wild.

🚀 Takeaway:

EEGlass charts a path toward **everyday brain–computer interfaces** by embedding EEG into familiar eyewear. Though resolution and signal strength are trade-offs, the prototype's proof-of-concept success in capturing alpha rhythms, motor-related ERD, and EOG artifacts suggests that socially integrated neuroadaptive systems may soon become a part of our daily wearable toolkit.