NOKIA Bell Labs

GLAMAR: Geo-Location Assisted Mobile Augmented Reality for Industrial Automation

Mostafa Uddin, **Sarit Mukherjee**, Murali Kodialam, TV Lakshman
Nokia Bell Labs

About the Authors

- Mostafa Uddin, https://mostafau.github.io/
- Sarit Mukherjee, https://www.bell-labs.com/about/researcher-profiles/saritmukherjee/
- Murali Kodialam, https://www.bell-labs.com/about/researcher-profiles/muralikodialam/
- TV Lakshman, https://www.bell-labs.com/about/researcher-profiles/tvlakshman/

Overview

- **Background**
  - Mobile AR (MAR)
- GLAMAR
  - Components
  - Coordinate Systems (environment accurately affects AR and vice versa)
  - Regenerative Particle Filter (estimating the location of targets)
    - Location, Acceleration, Motion Updates
- Evaluation
  - BLE vs. Wi-Fi
  - Computer Vision Comparison
  - Edge-based MAR Comparison

Mobile Augmented Reality (MAR) for Industrial Automation

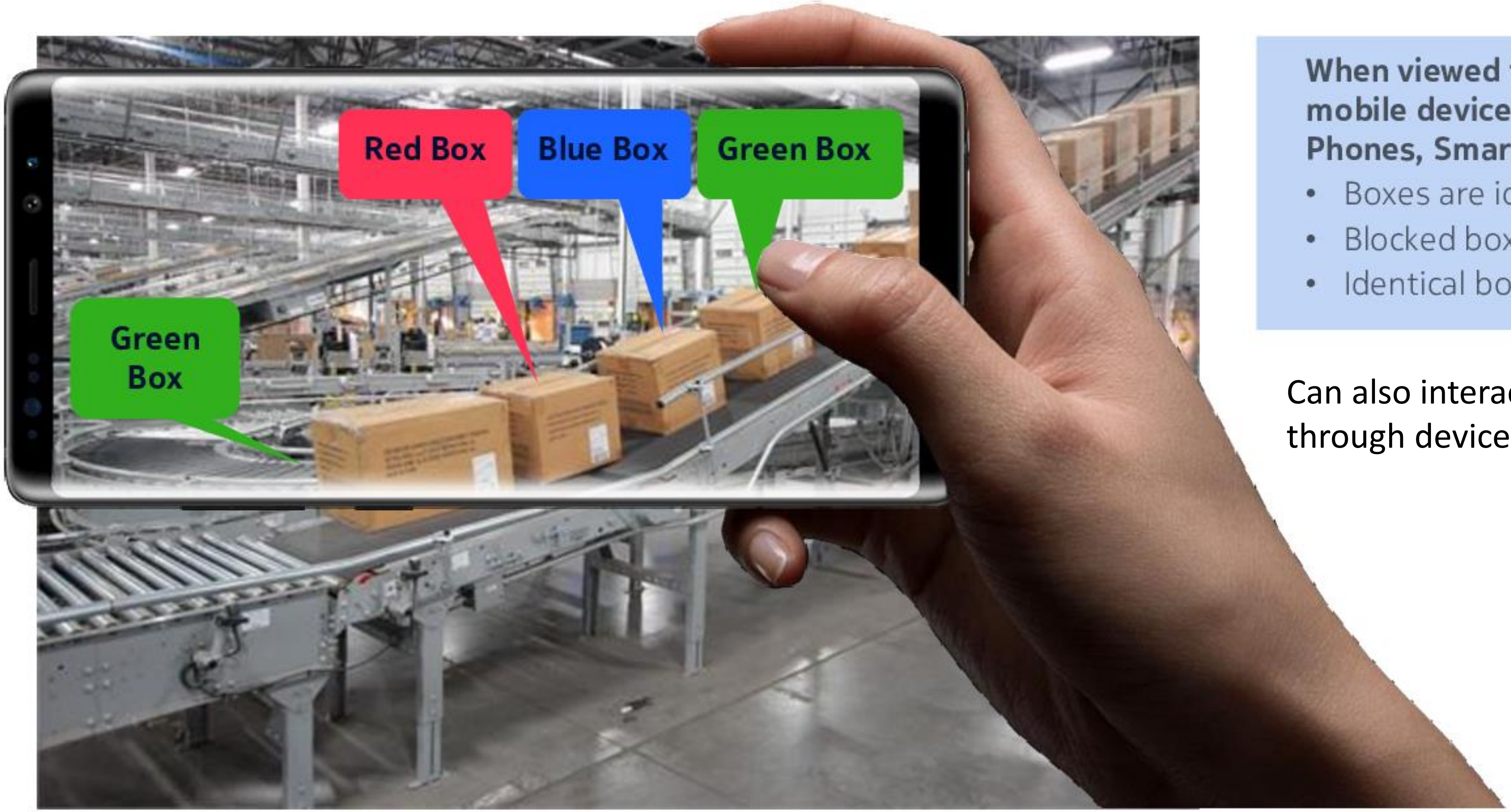
- Augmented Reality (AR) is going to play a significant role in transforming and automating Future Industry.
- Consumer mobile devices (e.g., Smart Phone, Smart Glasses) will bring in widespread adoption of AR in industry.
- Mobile AR (MAR) has some practical limitations due to the constrained capabilities of the devices (processing, battery).
- We propose a framework for efficient support of MAR on smart devices by leveraging the enhanced facilities installed in Future Industry.

Role of Mobile Augmented Reality (MAR) for Industrial Automation

Identification and control of objects in a digital warehouse is challenging when they are

- Moving,
- View is blocked, or
- Identical looking

Role of Mobile Augmented Reality (MAR) for Industrial Automation

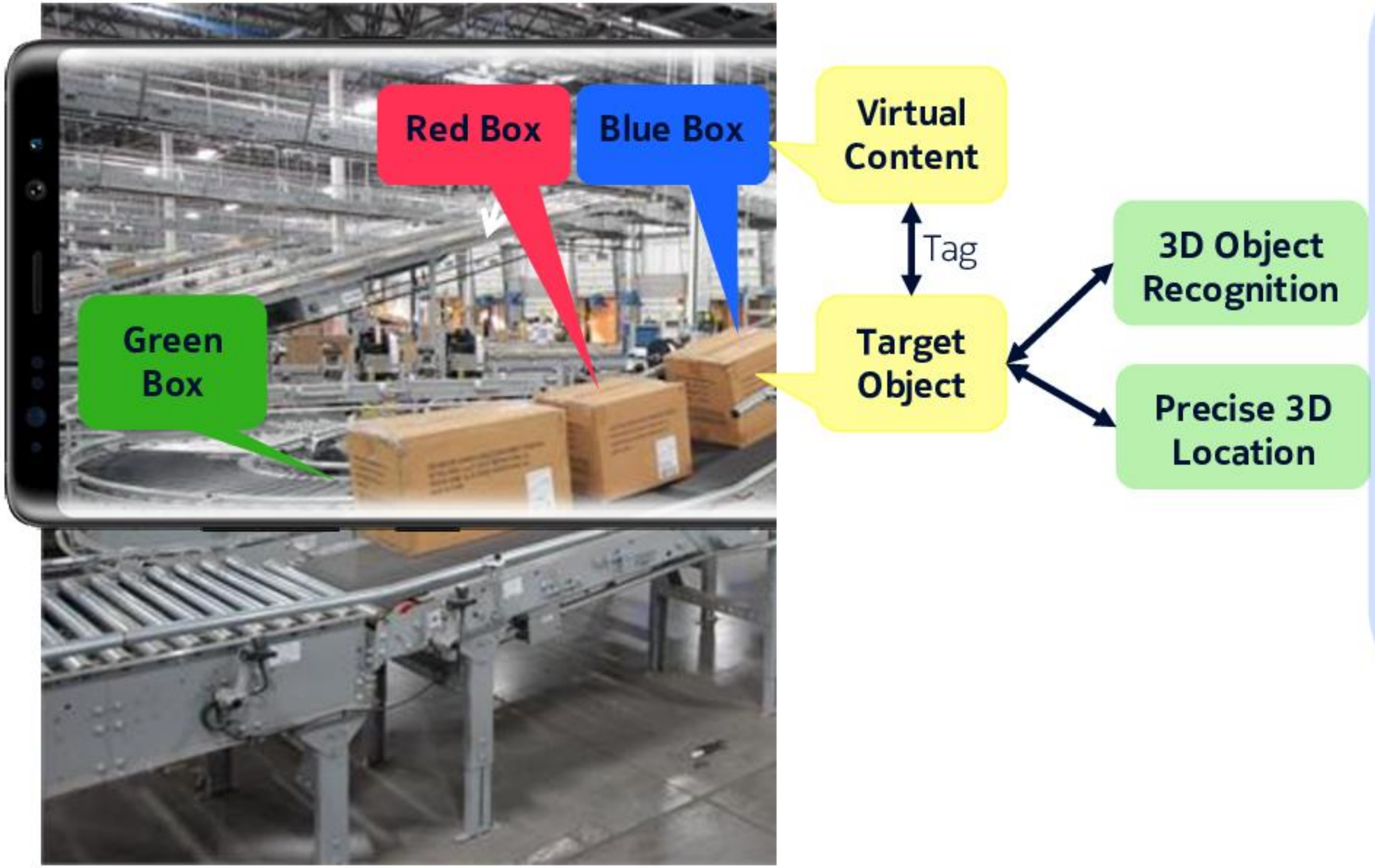


When viewed through AR enabled mobile device (e.g., Smart Phones, Smart Glasses)

- Boxes are identified/tagged,
- Blocked boxes are tagged
- Identical boxes differentiated

Can also interact with environment through device

Challenges in Supporting MAR for Industrial Automation



- State of the art computer vision-based techniques are impractical for mobile devices.
- In addition, vision-based techniques hindered by
  - Distance to object
  - Object occlusion
  - Similar looking objects
- 3D point cloud based solutions are expensive for MAR.
- Recently proposed MAR-based solutions have limited application.

Our Approach: Geo-Location Assisted MAR (GLAMAR)

Inertial measurement unit (IMU)

Leverage the future industry infrastructure to support MAR-based application efficiently on smart devices

- Location Service,
- Smart sensor
- Low latency communication,
- Edge compute

Claim that the active tags will become commonplace on some objects of interest (e.g. robots)

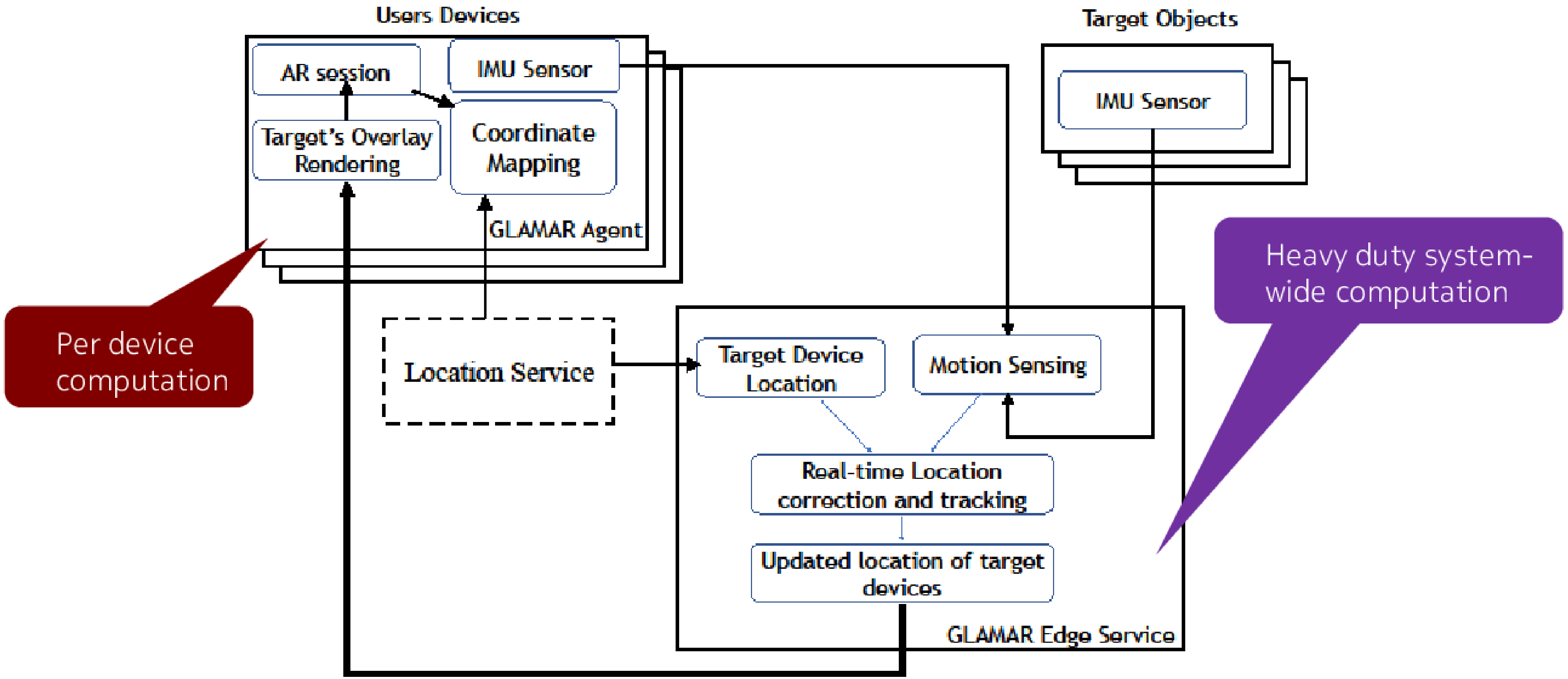
Advantages over Computer Vision

- stationary and mobile objects
- don't need line of sight
- works at any distance

Overview

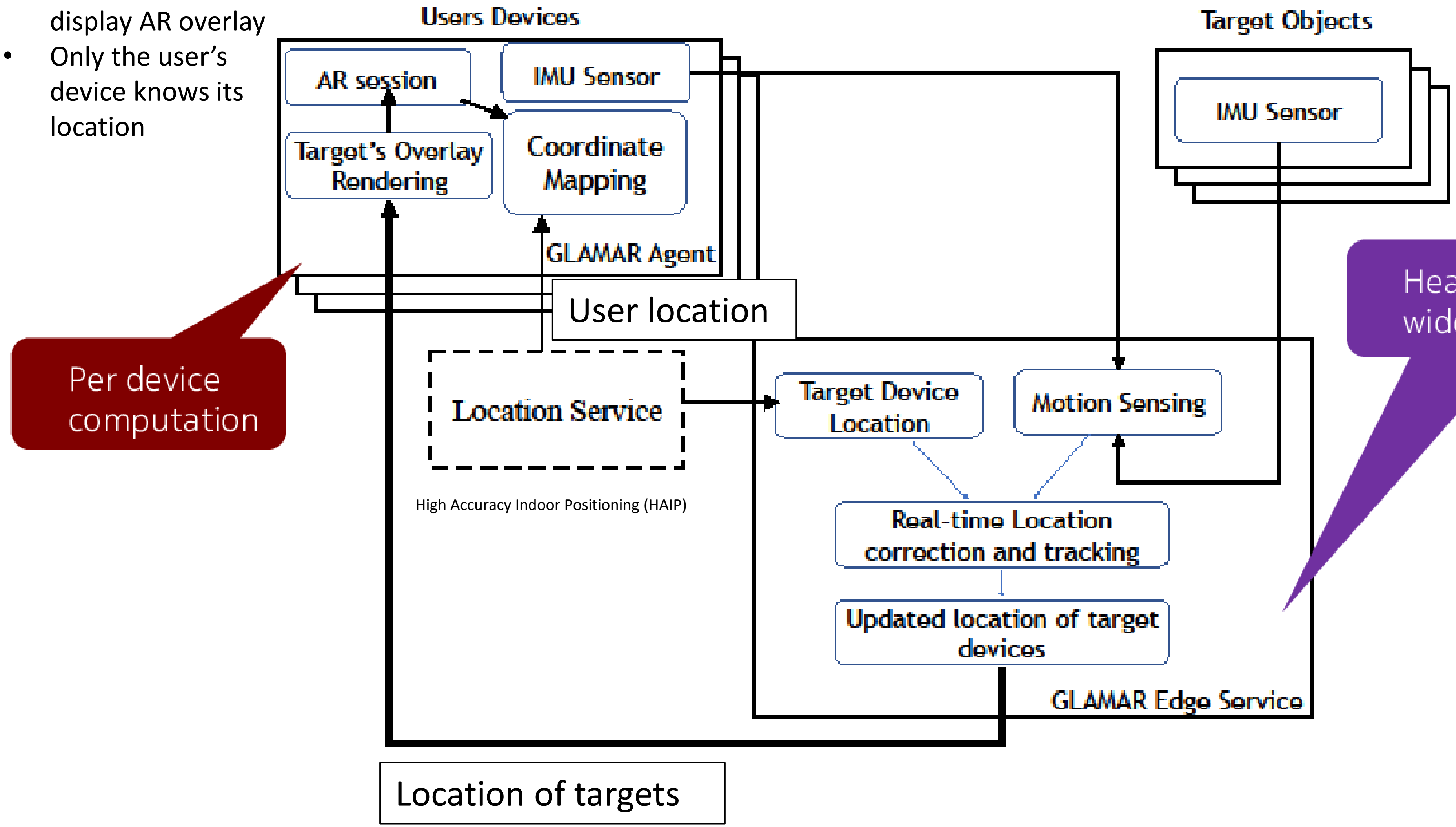
- Background
  - Mobile AR (MAR)
- **GLAMAR**
  - Components
  - Coordinate Systems (environment accurately affects AR and vice versa)
  - Regenerative Particle Filter (estimating the location of targets)
    - Location, Acceleration, Motion Updates
- Evaluation
  - BLE vs. Wi-Fi
  - Computer Vision Comparison
  - Edge-based MAR Comparison

GLAMAR System Overview



GLAMAR System Overview

- User devices display AR overlay
- Only the user's device knows its location



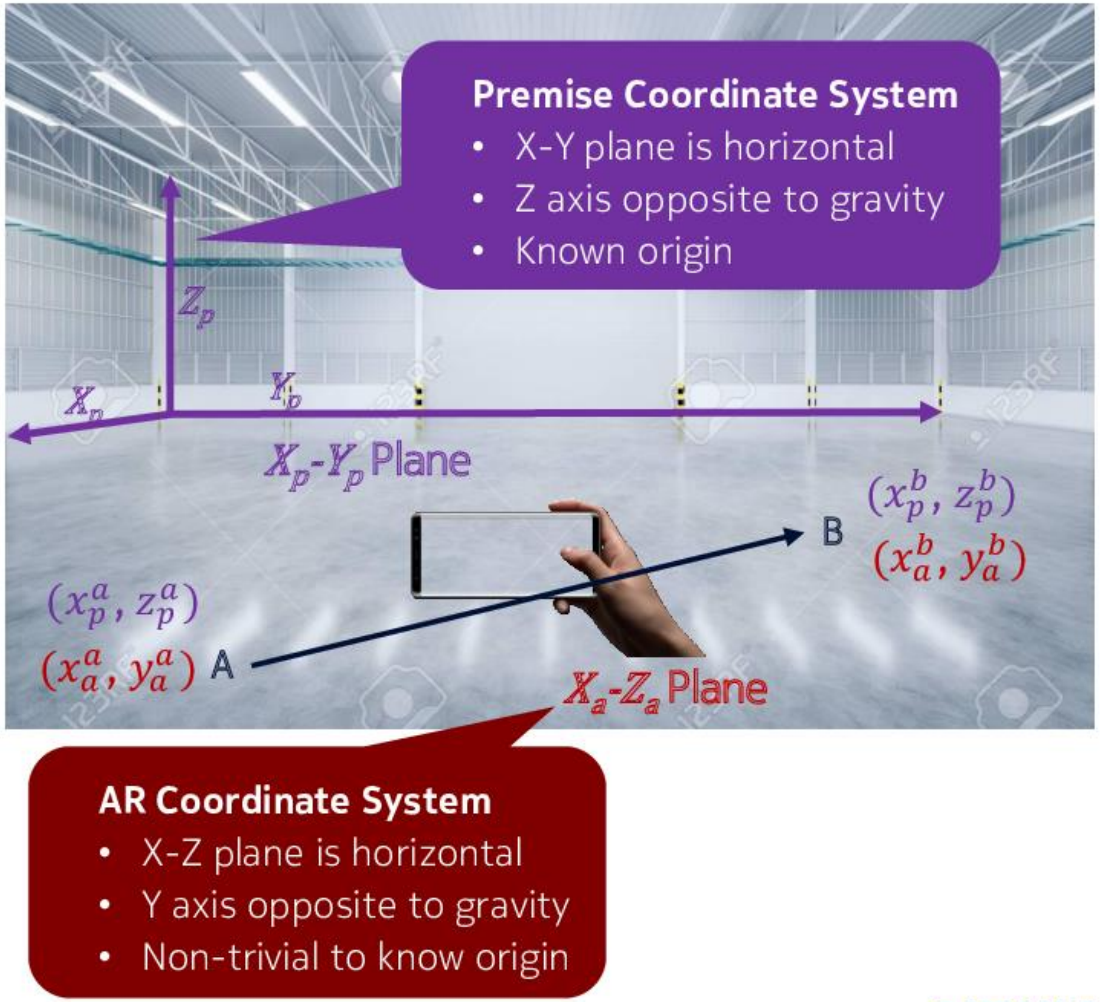
- Each target object (e.g. package) has Bluetooth tag (for HAIP location service) and IMU sensor (for edge service updates)

- Edge Service tracks location of each target object (e.g. package)
- Updates estimated location when triggered
- Sends target object locations to users

Coordinate Systems

- Four coordinate systems:

1. AR display coordinate system
   - real/virtual objects displayed here
   - 1 per user
2. Phone coordinate system
   - orientation of phone
3. Premise coordinate system
   - location of object in deployment site
4. Reference (Earth) coordinate system
   - cardinal directions
   - used as reference to find (3)

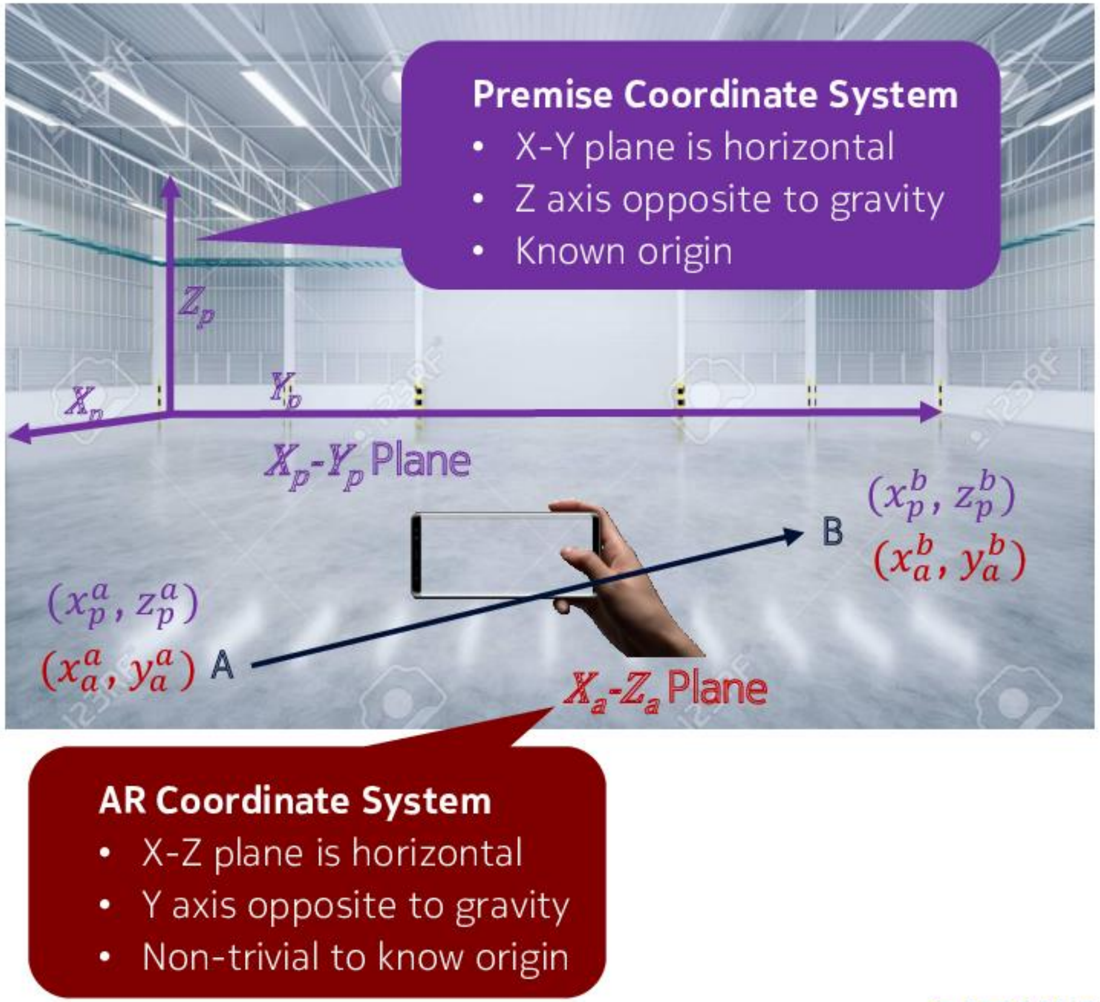


Focus on the mapping from (3) to (1)

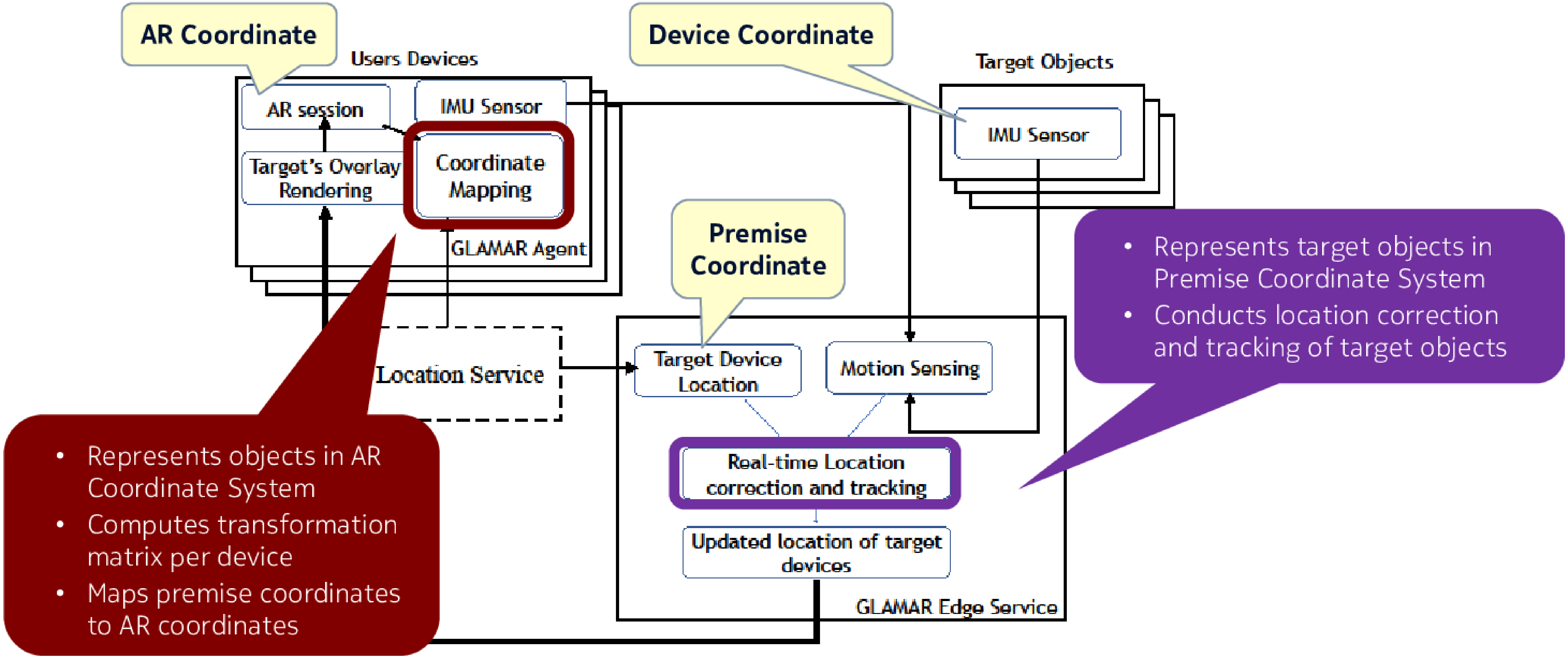
Coordinate Systems

- Four coordinate systems:

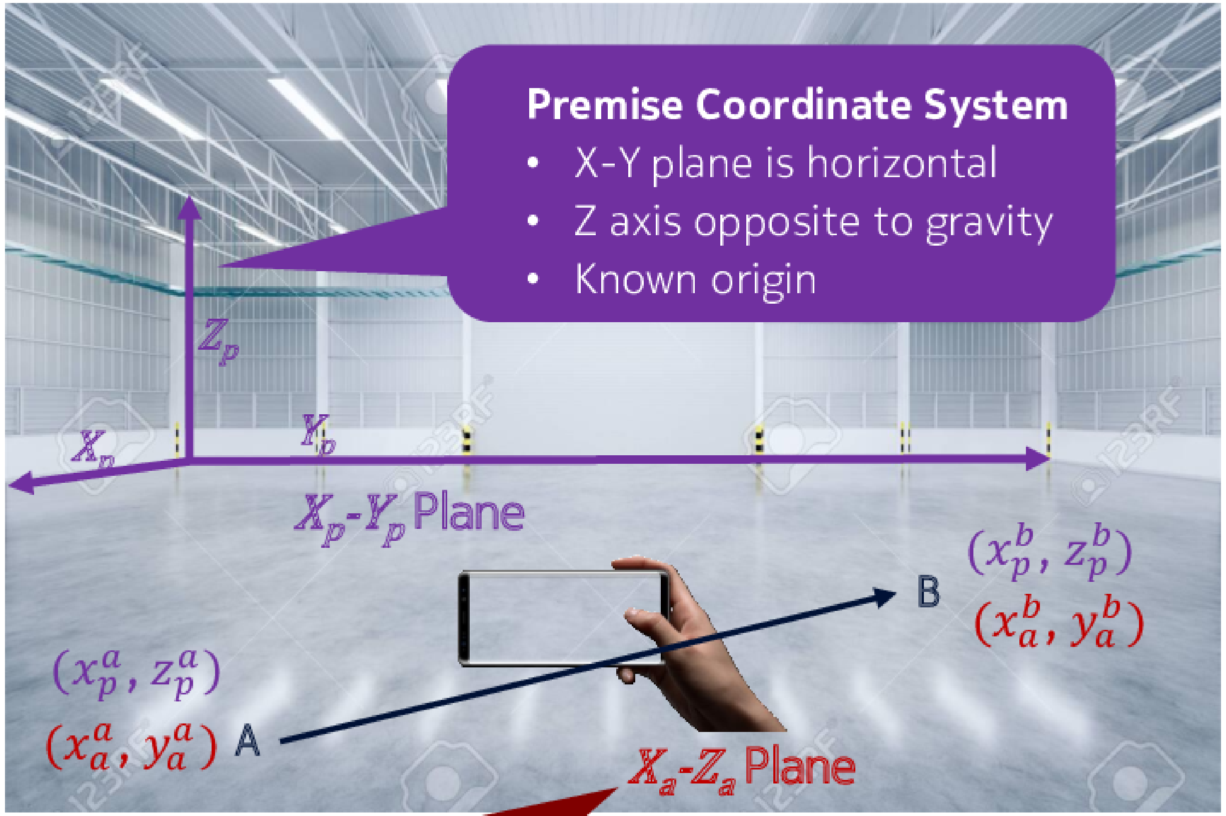
1. AR display coordinate system
   - real/virtual objects displayed here
   - 1 per user
2. Phone coordinate system
   - orientation of phone
3. Premise coordinate system
   - location of object in deployment site
4. Reference (Earth) coordinate system
   - cardinal directions
   - used as reference to find (3)



GLAMAR System Overview: Mapping Coordinate Systems

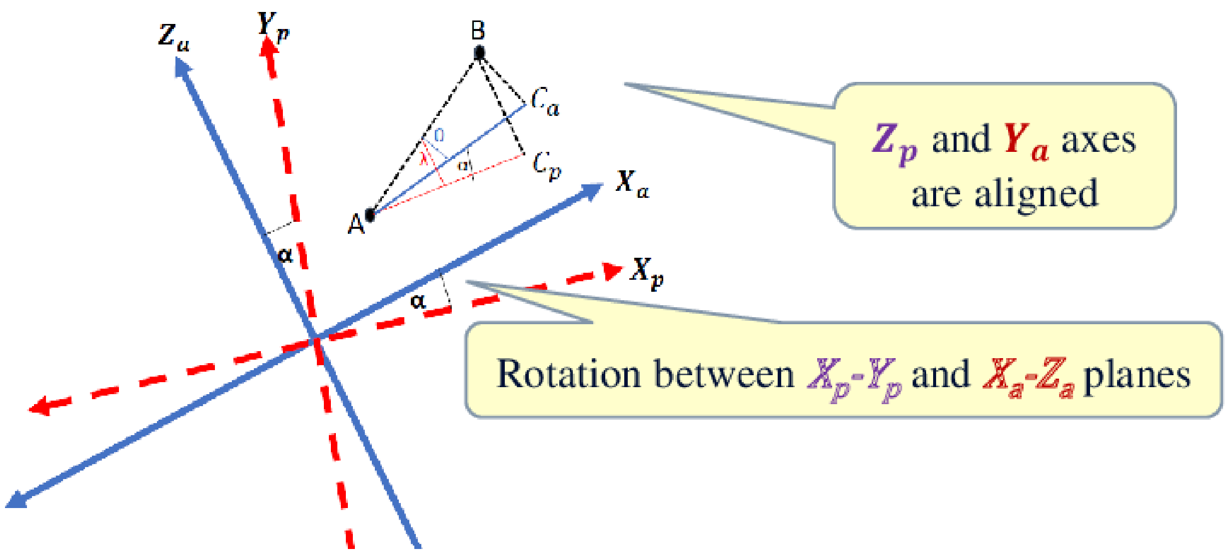


Mapping Coordinate Systems: Premise Coordinates to AR Coordinates



AR Coordinate System
- X-Z plane is horizontal
- Y axis opposite to gravity
- Non-trivial to know origin

Premise Coord x (Rotation & Translation) = AR Coord



$$\alpha = \arctan\left(\frac{y_p^b - y_p^a}{x_p^b - x_p^a}\right) - \arctan\left(\frac{z_a^b - z_a^a}{x_a^b - x_a^a}\right)$$

- Calculate rotation (α) first
  - Measure two points (a and b) in AR and Premise coordinates
- Then find translation by finding the difference between the points in rotated system
- Updates needed because
  1. Location service (premise coordinates) sometimes wrong
  2. AR coordinates change

Mapping Coordinate Systems: Premise Coordinates to AR Coordinates

R Coord

Y_a axes
aligned

X_a-Z_a planes

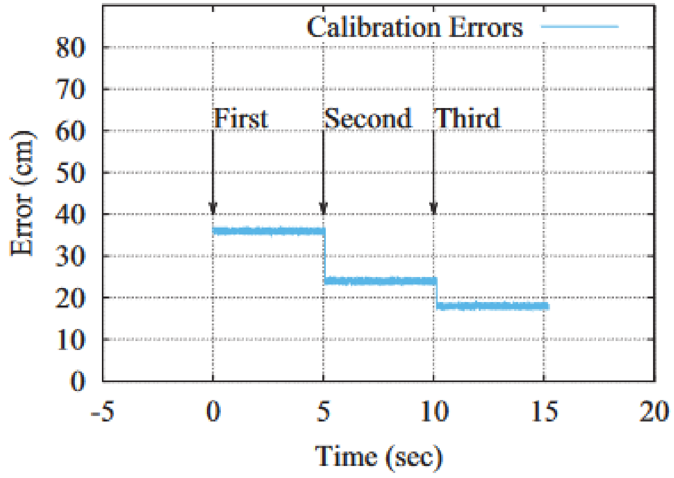


Fig. 7: Continuous coordinate calibration helps in reducing the error of tracking target objects in AR coordinate.

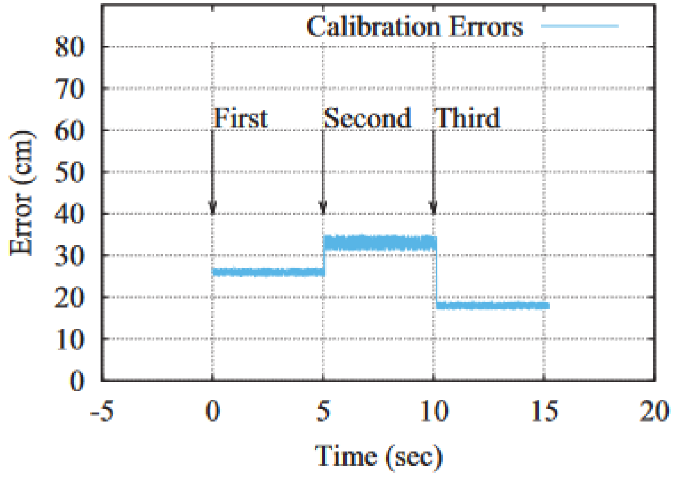


Fig. 8: Due to error in location readings, second calibration step increases the error in tracking target objects in AR coordinate.

$$\alpha = \arctan\left(\frac{\cdots}{x_p^b - x_p^a}\right) - \arctan\left(\frac{\cdots - z_a^a}{x_a^b - x_a^a}\right)$$

AR Coordinate System

- X-Z plane is horizontal
- Y axis opposite to gravity
- Non-trivial to know origin

- Calculate rotation (α) first
  - Measure two points (a and b) in AR and Premise coordinates
- Then find translation by finding the difference between the points in rotated system
- Updates needed because
  1. Location service (premise coordinates) sometimes wrong
  2. AR coordinates change

Regenerative Particle Filter

- Models location of a target as the average position of a set of n (= 4096) particles
- Each particle has position, velocity, and acceleration
- Update location of each particle based on sensor readings
- Resampling: remove unlikely particles and reinforce likely particles
- Regeneration: when the package is stationary, replace particles with particles drawn from a normal distribution centered at the current estimated location
  - Replacement particles have velocity and acceleration 0

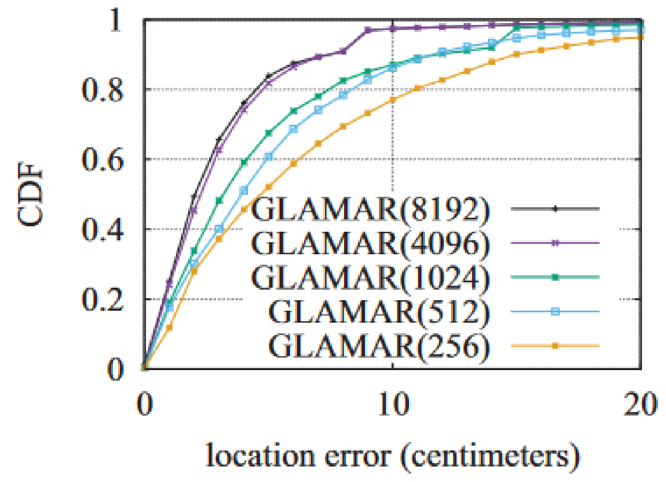
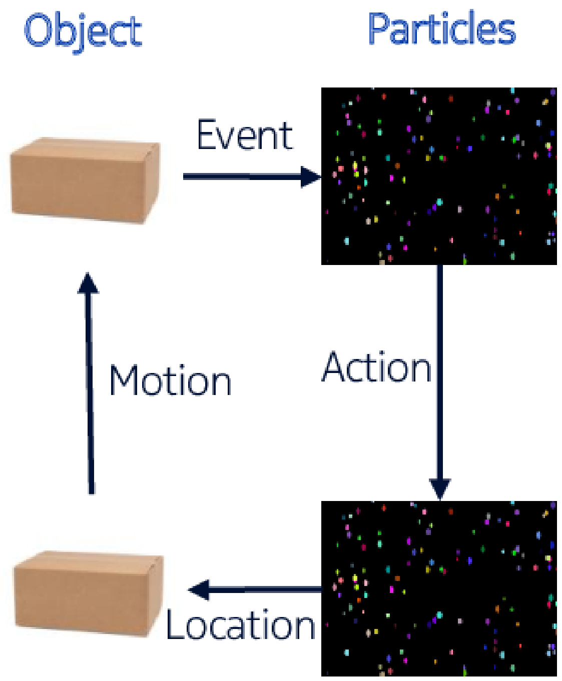


Fig. 15: Statistical distribution of absolute location error for different number of particles used in particle filter implementation.

Real-time Location Correction and Tracking: Regenerative Particle Filter

Event	Property	Action
Acceleration Update	Noisy, non-zero when stationary	• Update location of *n* particles based on the noise model of the acceleration • Update acceleration to new value
Location Update	Noisy, prone to interference	• Update location of *n* particles based on the noise model of the acceleration • **Resample**: Generate new maximum likely *n* particles from old particles based on new location value
Motion Update (stationary)	Stable and high confidence	• Calculate expected location of the object • **Regeneration**: Generate *n* stationary particles around that location

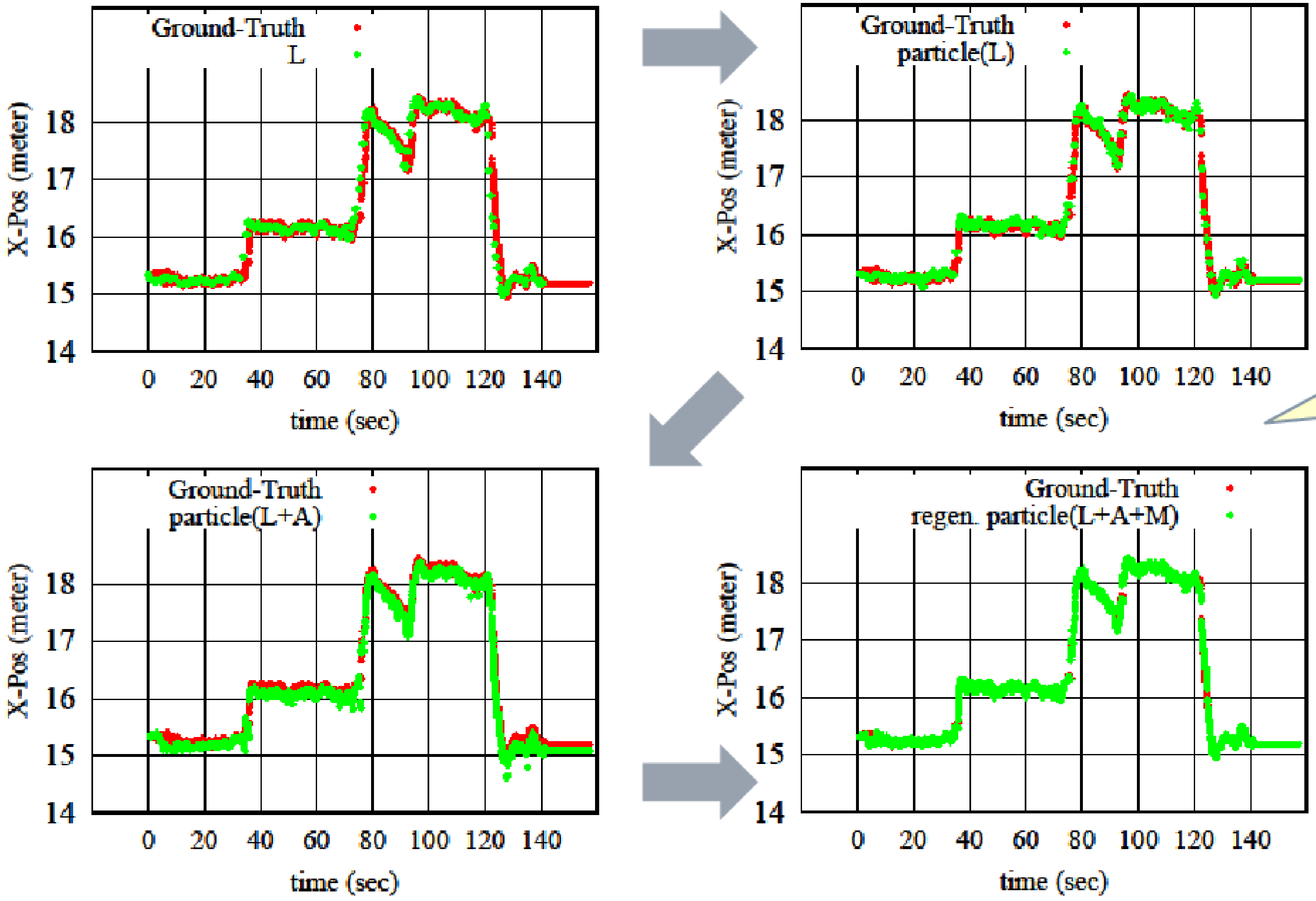


Freezing particles does not work

If particles have nonzero velocity or acceleration, they inaccurately model a stationary target

NOKIA Bell Labs

Effect of Regenerative Particle Filter



Accuracy improves by our regenerative particle filter in tracking a moving target object as more types of events are used

Event
Acceleration Update
Location Update
Motion Update (stationary)

Overview

- Background
  - Mobile AR (MAR)
- GLAMAR
  - Components
  - Coordinate Systems (environment accurately affects AR and vice versa)
  - Regenerative Particle Filter (estimating the location of targets)
    - Location, Acceleration, Motion Updates
- **Evaluation**
  - BLE vs. Wi-Fi
  - Computer Vision Comparison
  - Edge-based MAR Comparison

Evaluation

- Each target given a High Accuracy Indoor Positioning (HAIP) Tag and IMU sensor (Android phone)
- GLAMAR client in Android using ARCore
- GLAMAR edge server with a ring queue for each target and 3 processes:
    1. Producer: Adds HAIP location data (Acceleration, Location)
    2. Producer: Determine whether package has stopped moving (Motion)
    3. Consumer: Particle filter
- Ground truth for accuracy determined by sending location of target's phone to client phone through Google Firebase

Evaluation

- BLE vs. Wi-Fi
  - Compare efficiency and performance of BLE and Wi-Fi for communication between target, edge server, and client
  - BLE better in both efficiency and performance
- Computer Vision Comparison
  - Compare efficiency against alternative approach implemented as an Android App using OpenCV on a Google Pixel 4
  - 83.3% lower CPU utilization, 200% framerate increase
- Edge-based MAR Comparison
  - Unable to gather data due to proprietary alternatives, differing metrics
  - Comparison table of claimed features

Evaluation: BLE vs. Wi-Fi

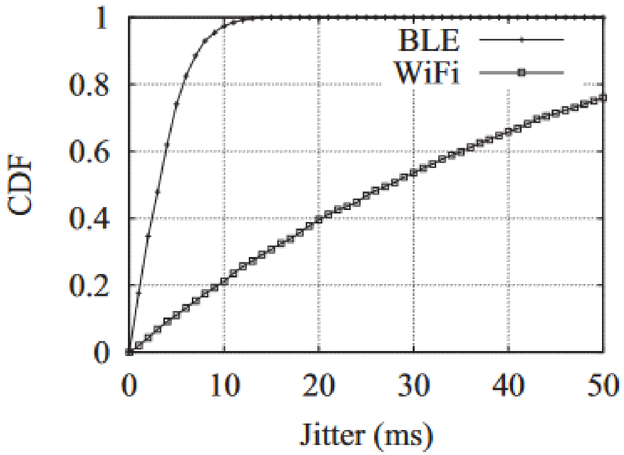


Fig. 9: Observed jitter in Wi-Fi and BLE in industrial setting.

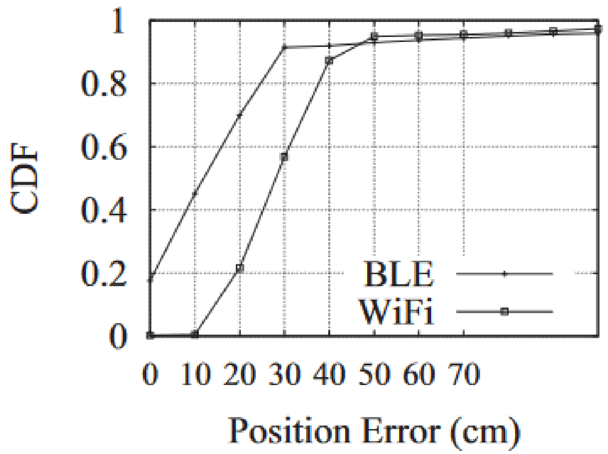
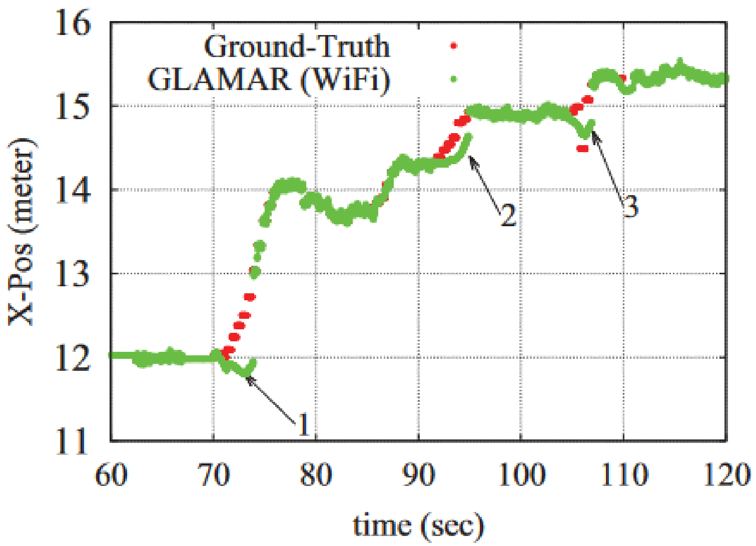
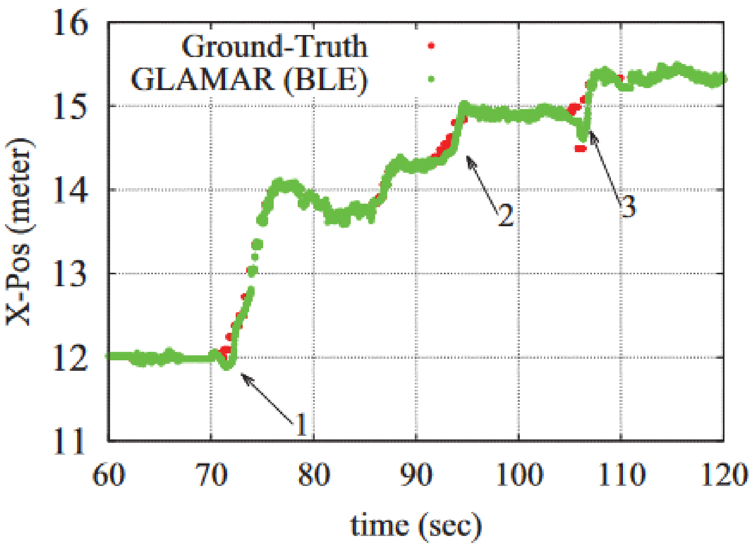


Fig. 11: Position error comparison between using BLE and Wi-Fi.





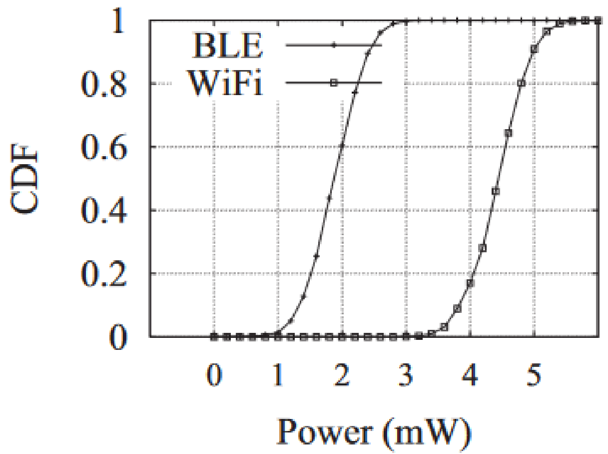
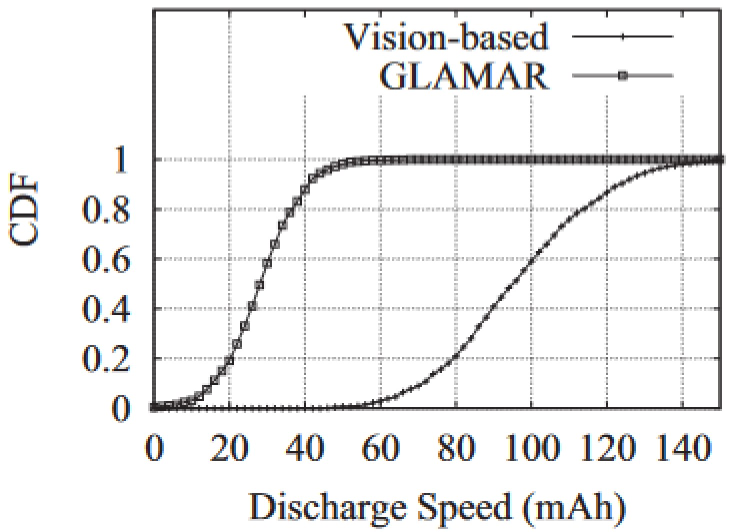


Fig. 12: Power consumption measurement using BLE and Wi-Fi.

Evaluation: Computer Vision Comparison

83.3% lower average CPU utilization, 200% framerate increase, longer battery life

	Vision-based	GLAMAR	
CPU utilization for stationary target object (%)	43.1±14	8±3	81% lower
CPU utilization for in-motion target object (%)	62.7±7	9.3±4	85% lower
Average frame rate (fps)	11±1.2	33±2.1	200% higher



Evaluation: Edge-based MAR Comparison

Features \ MAR System	MARVEL [36]	Jaguar [14]	[10]	MARLIN [9]	GLAMAR
Object tracking	Stationary	Stationary	Stationary+Moving	Stationary+Moving	Stationary+Moving
Detected object type	N/A	Planar	Planar+Non-planar	Planar+Non-planar	Planar+Non-planar
Detected object position (coordinate)	3D Real-world	3D Real-world	2D Image frame	2D Image frame	3D Real-world
Solution Architecture	Edge-assisted	Edge-assisted	Edge-assisted	On-device	Edge-assisted
Integrable with external AR platforms	No	Yes	No	No	Yes
On-device computation	Moderate	Moderate	Moderate	High	Low
Privacy preservation	No	No	No	Yes	Yes

TABLE II: Comparison of GLAMAR with recently proposed MAR systems.

- They claim their solution is the best across all given features

Evaluation: Edge-based MAR Comparison

MAR System / Features	MARVEL [36]	Jaguar [14]	[10]	MARLIN [9]	GLAMAR
Object tracking	Stationary	Stationary	Stationary+Moving	Stationary+Moving	Stationary+Moving
Detected object type	N/A	Planar	Planar+Non-planar	Planar+Non-planar	Planar+Non-planar
Detected object position (coordinate)	3D Real-world	3D Real-world	2D Image frame	2D Image frame	3D Real-world
Solution Architecture	Edge-assisted	Edge-assisted	Edge-assisted	On-device	Edge-assisted
Integrable with external AR platforms	No	Yes	No	No	Yes
On-device computation	Moderate	Moderate	Moderate	High	Low
Privacy preservation	No	No	No	Yes	Yes

TABLE II: Comparison of GLAMAR with recently proposed MAR systems.

- Only system capable of tracking non-planar moving objects in a 3D coordinate system

Positive/Negative Points

Positive

- Evaluated GLAMAR implementation against computer vision alternative
- System works with different methods of communication between components

Negative

- Vague experimental procedure, mostly focused on setup
- Unable to quantitatively compare to other edge-based systems
- Did not compare accuracy of GLAMAR to computer vision approach

Questions

- Is it really practical to assume the tracker cost is low?
- Are there any other cases where we might have an environment where we can set up the location service and tag everything beforehand?