



INTEGRATED SENSING AND COMMUNICATION

October 14, 2026 | E. Zandi, E. Pasetto, G. Cavallaro | Forschungszentrum Jülich

ISAC Operational Modes & Earth Observation Equivalents

1. Active Radar Imaging

Active Microwave Sounding

- Low-altitude atmospheric profiling
- Obstacle and hazard tracking

3. Bistatic / Multistatic Network

Bi-static Satellite Radar

- 3D surface topography reconstruction
- Wide-area disaster imaging

2. Ground-Based SAR (GB-SAR)

Terrestrial InSAR

- Volcanic flank deformation tracking
- Landslide and slope stability monitoring

4. CSI Anomaly Detection

Electromagnetic Wave-Environment Interaction

- Early wildfire smoke/plasma detection
- Severe weather and storm tracking

Possible Use Cases: 6G/5G Advanced

- 1 3GPP TR 22.870 (UC 7.3): Vulnerable pedestrian safety

Possible Use Cases: 6G/5G Advanced

- 1 3GPP TR 22.870 (UC 7.3): Vulnerable pedestrian safety
- 2 3GPP TR 22.870 (UC 7.4): High-res topographical maps

Possible Use Cases: 6G/5G Advanced

- 1 3GPP TR 22.870 (UC 7.3): Vulnerable pedestrian safety
- 2 3GPP TR 22.870 (UC 7.4): High-res topographical maps
- 3 3GPP TR 22.870 (UC 7.5): Low-altitude UAV supervision

Possible Use Cases: 6G/5G Advanced

- 1 3GPP TR 22.870 (UC 7.3): Vulnerable pedestrian safety
- 2 3GPP TR 22.870 (UC 7.4): High-res topographical maps
- 3 3GPP TR 22.870 (UC 7.5): Low-altitude UAV supervision
- 4 3GPP TR 22.870 (UC 7.6): Environment object reconstruction

Possible Use Cases: 6G/5G Advanced

- 1 3GPP TR 22.870 (UC 7.3): Vulnerable pedestrian safety
- 2 3GPP TR 22.870 (UC 7.4): High-res topographical maps
- 3 3GPP TR 22.870 (UC 7.5): Low-altitude UAV supervision
- 4 3GPP TR 22.870 (UC 7.6): Environment object reconstruction
- 5 3GPP TR 22.870 (UC 7.7): Road digitalization

Possible Use Cases: 6G/5G Advanced

- 1 3GPP TR 22.870 (UC 7.3): Vulnerable pedestrian safety
- 2 3GPP TR 22.870 (UC 7.4): High-res topographical maps
- 3 3GPP TR 22.870 (UC 7.5): Low-altitude UAV supervision
- 4 3GPP TR 22.870 (UC 7.6): Environment object reconstruction
- 5 3GPP TR 22.870 (UC 7.7): Road digitalization
- 6 3GPP TR 22.870 (UC 7.9): Ship detection and tracking

Possible Use Cases: 6G/5G Advanced

- 1 3GPP TR 22.870 (UC 7.3): Vulnerable pedestrian safety
- 2 3GPP TR 22.870 (UC 7.4): High-res topographical maps
- 3 3GPP TR 22.870 (UC 7.5): Low-altitude UAV supervision
- 4 3GPP TR 22.870 (UC 7.6): Environment object reconstruction
- 5 3GPP TR 22.870 (UC 7.7): Road digitalization
- 6 3GPP TR 22.870 (UC 7.9): Ship detection and tracking
- 7 3GPP TR 22.837 (UC 5.5): Sensing for flooding in smart cities

Possible Use Cases: 6G/5G Advanced

- 1 3GPP TR 22.870 (UC 7.3): Vulnerable pedestrian safety
- 2 3GPP TR 22.870 (UC 7.4): High-res topographical maps
- 3 3GPP TR 22.870 (UC 7.5): Low-altitude UAV supervision
- 4 3GPP TR 22.870 (UC 7.6): Environment object reconstruction
- 5 3GPP TR 22.870 (UC 7.7): Road digitalization
- 6 3GPP TR 22.870 (UC 7.9): Ship detection and tracking
- 7 3GPP TR 22.837 (UC 5.5): Sensing for flooding in smart cities
- 8 Fire and early fire detection

Possible Use Cases: 6G/5G Advanced

- 1 3GPP TR 22.870 (UC 7.3): Vulnerable pedestrian safety
- 2 3GPP TR 22.870 (UC 7.4): High-res topographical maps
- 3 3GPP TR 22.870 (UC 7.5): Low-altitude UAV supervision
- 4 3GPP TR 22.870 (UC 7.6): Environment object reconstruction
- 5 3GPP TR 22.870 (UC 7.7): Road digitalization
- 6 3GPP TR 22.870 (UC 7.9): Ship detection and tracking
- 7 3GPP TR 22.837 (UC 5.5): Sensing for flooding in smart cities
- 8 Fire and early fire detection
- 9 Wildlife detection and tracking (beavers, geese, deer at FZJ)

Possible Use Cases: 6G/5G Advanced

- 1 3GPP TR 22.870 (UC 7.3): Vulnerable pedestrian safety
- 2 3GPP TR 22.870 (UC 7.4): High-res topographical maps
- 3 3GPP TR 22.870 (UC 7.5): Low-altitude UAV supervision
- 4 3GPP TR 22.870 (UC 7.6): Environment object reconstruction
- 5 3GPP TR 22.870 (UC 7.7): Road digitalization
- 6 3GPP TR 22.870 (UC 7.9): Ship detection and tracking
- 7 3GPP TR 22.837 (UC 5.5): Sensing for flooding in smart cities
- 8 Fire and early fire detection
- 9 Wildlife detection and tracking (beavers, geese, deer at FZJ)
- 10 RAN parameter design and adaptation (needs private network)

Possible Use Cases: 6G/5G Advanced

- 1 3GPP TR 22.870 (UC 7.3): Vulnerable pedestrian safety
- 2 3GPP TR 22.870 (UC 7.4): High-res topographical maps
- 3 3GPP TR 22.870 (UC 7.5): Low-altitude UAV supervision
- 4 3GPP TR 22.870 (UC 7.6): Environment object reconstruction
- 5 3GPP TR 22.870 (UC 7.7): Road digitalization
- 6 3GPP TR 22.870 (UC 7.9): Ship detection and tracking
- 7 3GPP TR 22.837 (UC 5.5): Sensing for flooding in smart cities
- 8 Fire and early fire detection
- 9 Wildlife detection and tracking (beavers, geese, deer at FZJ)
- 10 RAN parameter design and adaptation (needs private network)
- 11 Many more

Network Variables for ISAC

Variables measurable from mobile networks for sensing:

- **CSI:** Channel matrices H , path gains, phases
- **AoA:** Azimuth/elevation of paths
- **AoD:** Angle of departure
- **Delay (ToA):** Path delay, RMS delay spread
- **Doppler:** Radial velocity, micro-Doppler
- **RSS:** Received signal strength
- **RSRP:** Reference signal received power

Network Variables for ISAC

Variables measurable from mobile networks for sensing:

- **CSI:** Channel matrices H , path gains, phases
- **AoA:** Azimuth/elevation of paths
- **AoD:** Angle of departure
- **Delay (ToA):** Path delay, RMS delay spread
- **Doppler:** Radial velocity, micro-Doppler
- **RSS:** Received signal strength
- **RSRP:** Reference signal received power

Of course, it is a very complex signal processing task!

Channel Model: Channel Frequency Response (CFR)

The CFR from Tx antenna i_t to Rx antenna i_r over subcarrier j and OFDM symbol k :

$$H_{i_t, i_r, j, k} = \beta_k \times \underbrace{e^{-j[2\pi(\Delta f_j \Delta \tau^{TO} + \Delta f_k^{CFO} k T_s) + \phi_{i_t, i_r}]}}_{\text{Phase Offsets}} \times \sum_{l=1}^{N_p} \left[\underbrace{\rho_{i_t, i_r, j, k}[l]}_{\text{Path Loss}} \underbrace{e^{-j2\pi \Delta f_j \tau[l]}}_{\text{Delay}} \underbrace{e^{-j2\pi f^D[l] k T_s}}_{\text{Doppler}} \right. \\ \left. \times \underbrace{e^{-j2\pi \frac{f_c}{c} (i_r - 1) \Delta d_r \sin \theta^r[l]}}_{\text{AoA}} \underbrace{e^{-j2\pi \frac{f_c}{c} (i_t - 1) \Delta d_t \sin \theta^t[l]}}_{\text{AoD}} \right]$$

Key components: β_k (power variation), ϕ (hardware offset), ρ (path loss), τ (delay), f^D (Doppler), θ^r/θ^t (AoA/AoD), f_c (carrier freq.), Δd (antenna spacing)

Vital Sign Monitoring: Respiration and Heart Rate

Principle: Periodic chest/heart movement causes small path length changes
→ phase variations in received signal (micro-Doppler).

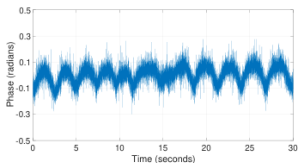
Process:

- 1 Static target localization:** Use Doppler dimension to identify low-velocity components (chest movement)
- 2 Phase extraction:** Extract CSI phase from the selected Doppler-Delay-AoA bin
- 3 Phase differencing:** Suppress slow drifts and body sway
- 4 Bandpass filtering:** Respiration: 0.1–0.5 Hz, Heartbeat: 0.8–2 Hz
- 5 Frequency estimation:** FFT or MUSIC to extract dominant frequencies

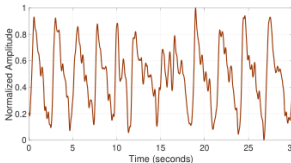
Experimental setup: 5 GHz WiFi, 2m Tx-Rx distance

Vital Sign Monitoring: Respiration and Heart Rate

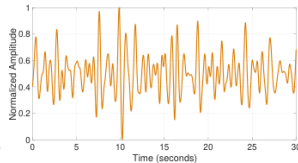
Result: Successfully extracted both respiration and heartbeat waveforms from residual CSI phase.



(a) Raw phase feature



(b) Extracted respiration waveform



(c) Extracted heartbeat waveform

Experimental results from field campaign at Parramatta River, Sydney, Credit: Wu et al. ISAC: From Human to Environmental Sensing, Jul. 18, 2025

Environmental Sensing: Water Level Monitoring

Water level monitoring via mobile network signals:

Physical mechanism: Water surfaces affect signal propagation through:

- Reflection paths (especially at low grazing angles or NLOS)
- Doppler spread and phase coherence
- CSI magnitude fluctuations and time-of-flight

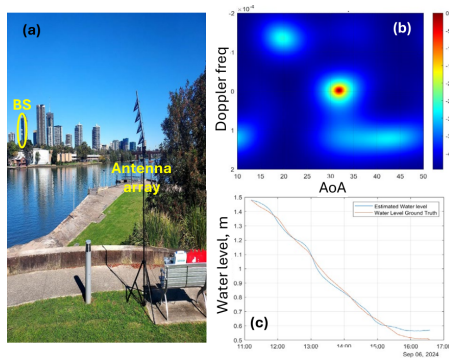
Field campaign: Parramatta River, Sydney (NLOS, indoor reception)

- SDR receiver with two monopole antennas
- Signals from 7 LTE cells (Bands 1, 3, 5, 28)
- Sonar-based water level sensor as ground truth

Water Level Monitoring: Experimental Results

Results:

- Linear regression with RSRP from 7 LTE carriers: RMSE = **7.36 cm**
- Joint Doppler-AoA processing: sensing resolution of 10^{-4} m^2



Experimental results from field campaign at Parramatta River, Sydney, showing water level estimation using LTE signals. Credit: Wu et al. ISAC: From Human to Environmental Sensing, Jul. 18, 2025.

Environmental Sensing: Soil Moisture Measurement

Physical mechanism: Soil moisture changes dielectric permittivity ε :

- **Phase:** $\Delta\phi \propto \sqrt{\varepsilon(\theta)}$ where θ = volumetric water content
- **Delay:** Path delay variations with permittivity
- **Reflectivity:** Interface reflectivity Γ changes with moisture

Environmental Sensing: Soil Moisture Measurement

Physical mechanism: Soil moisture changes dielectric permittivity ε :

- **Phase:** $\Delta\phi \propto \sqrt{\varepsilon(\theta)}$ where θ = volumetric water content
- **Delay:** Path delay variations with permittivity
- **Reflectivity:** Interface reflectivity Γ changes with moisture

Implementations:

- **LoRa-based:** RSSI-phase correlation achieves **sub-4%** moisture error
- **WiFi-based (SoilTAG):** Moisture-sensitive subcarrier profiles
- **LTE-based:** Passive sensing with matched filtering
- **GNSS-R:** Surface and root-zone moisture, RMSE < **0.05 cm³/cm³**

Environmental Sensing: Soil Moisture Measurement

Physical mechanism: Soil moisture changes dielectric permittivity ε :

- **Phase:** $\Delta\phi \propto \sqrt{\varepsilon(\theta)}$ where θ = volumetric water content
- **Delay:** Path delay variations with permittivity
- **Reflectivity:** Interface reflectivity Γ changes with moisture

Implementations:

- **LoRa-based:** RSSI-phase correlation achieves **sub-4%** moisture error
- **WiFi-based (SoilTAG):** Moisture-sensitive subcarrier profiles
- **LTE-based:** Passive sensing with matched filtering
- **GNSS-R:** Surface and root-zone moisture, RMSE < **0.05 cm³/cm³**

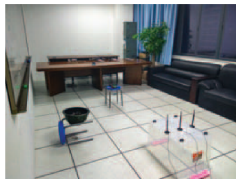
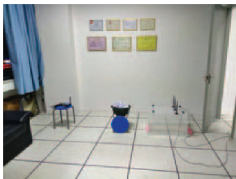
Challenges

- Soil type heterogeneity
- Changing surface conditions
- **Question:** Can remote-sensing (as additional modality) help?

Fire Detection: WiFi-Based Sensing (WiFire)

Device-free fire detection using WiFi networks:

Key contribution: First work to leverage CSI from commercial WiFi (802.11n) for fire detection—no additional sensors required.



System overview of WiFire for device-free fire detection using WiFi CSI. Credit: Zhong, et al., Wi-fire: Device-free fire detection using WiFi networks, ICC, 2017

Fire Detection: A Long Journey

How fire affects radio signals:

- **Plasma Attenuation:** Thermal ionization (K^+ , Na^+) absorbs RF energy.
- **Refractive Bending:** Thermal gradients (∇n) deflect rays.
- **Particulate Scattering:** Soot/ash cause Rayleigh/Mie extinction.
- **Phase Scintillation:** Turbulence induces phase/Doppler jitter.
- **Missing Channel Models:** No standardized model for dynamic $\epsilon_r(t)$.

Fire Detection: A Long Journey

How fire affects radio signals:

- **Plasma Attenuation:** Thermal ionization (K^+ , Na^+) absorbs RF energy.
- **Refractive Bending:** Thermal gradients (∇n) deflect rays.
- **Particulate Scattering:** Soot/ash cause Rayleigh/Mie extinction.
- **Phase Scintillation:** Turbulence induces phase/Doppler jitter.
- **Missing Channel Models:** No standardized model for dynamic $\epsilon_r(t)$.

Key Modeling Challenges

- **Lack of Reliable Channel Models:** No standardized model yet exists
- **High Sensitivity to Fire Type:**
 - *Forest/Bushfires:* High plasma absorption from vegetation alkali.
 - *Industrial/Chemical:* Dense, conductive soot scattering.
 - *Structural/Indoor:* Severe multipath interference with thermal fading.

Our Current Activities

- Multimodal Environment Reconstruction
 - Relevant to **environment object reconstruction** (3GPP TR 22.870 UC 7.6)

Our Current Activities

- Multimodal Environment Reconstruction
 - Relevant to **environment object reconstruction** (3GPP TR 22.870 UC 7.6)
- **Core idea:** The process of creating a digital twin of physical spacial region of interest. **Multimodal** means that the construction happens by combining data from **multiple, heterogenous data sources**

Our Current Activities

- Multimodal Environment Reconstruction
 - Relevant to **environment object reconstruction** (3GPP TR 22.870 UC 7.6)
- **Core idea:** The process of creating a digital twin of physical spacial region of interest. **Multimodal** means that the construction happens by combining data from **multiple, heterogenous data sources**
- The modular design ensures that each new modality can be added to the existing framework as an independent layer without the need to re-design the whole pipeline

Our Current Activities

- Multimodal Environment Reconstruction
 - Relevant to **environment object reconstruction** (3GPP TR 22.870 UC 7.6)
- **Core idea:** The process of creating a digital twin of physical spacial region of interest. **Multimodal** means that the construction happens by combining data from **multiple, heterogenous data sources**
- The modular design ensures that each new modality can be added to the existing framework as an independent layer without the need to re-design the whole pipeline
- Application
 - Environment aware ray-tracing
 - Static/dynamic RAN parameter desing and adaptation
 - V2X communication and planning
 - Smart city and urban planning

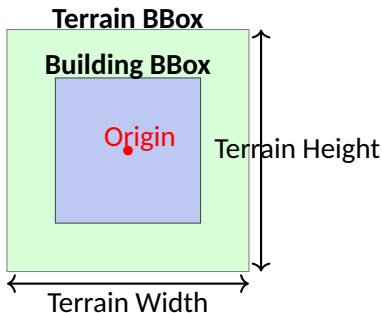
Edoardo starts here

Example of current implementation of the Environment reconstruction



The workflow starts by providing as input center point (either by manually selecting it in the map viewer or providing the coordinates explicitly) and a bounding box size. The workflow then creates a Sionna scene centered in the point with the selected bounding box.

Scene Definition: Bounding Box Setup



Key Parameters

- **Building BBox:** Defines region for building extraction
- **Terrain BBox:** Expanded region for complete terrain coverage
- **Origin Point:** Reference for coordinate transformation (from world coordinates to scene coordinates)

Modalities currently employed in the project

- **3D-BAG:** The official 3-D model of every Dutch address and building, offering geometry at multiple levels of detail for visualisation, spatial analysis and planning.
- **AHN (Actueel Hoogtebestand Nederland):** High-resolution digital elevation model (DEM) for terrain reconstruction.
- **OpenStreetMap:** A crowd-sourced, openly licensed geographic database that provides vector data (roads, buildings, land use, POIs, etc.) for into GIS and simulation pipelines (used to find road-facing walls).
- **Mapillary:** A collaborative platform that aggregates street-level photos and video from contributors worldwide. The street-level images are then used to enrich the facade's level of detail.

Buildings reconstruction from 3D-BAG dataset

3D-BAG Dataset

3D Basis Administratieve Gebieden

- Netherlands nationwide 3D building dataset
- Format: CityJSON (Level of detail 2.2)
- Each building has a unique identifier and has 3 associated geometries: Walls, Roofs, Ground
- Coordinate reference system (CRS): EPSG:28992 (RD/NAP)

Common CRS: EPSG:28992 (Dutch National Coordinate System)
Allows direct spatial alignment of building and terrain data

3D-BAG dataset overview

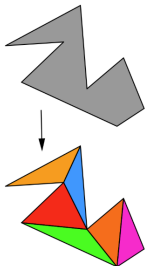
- Data for each building are stored in a dictionary-like structures with the keys being unique building identifiers
- Each building in dataset has associated a set of surfaces, that in turn are just a list of vertices. Moreover, each surface has associated one of 3 possible labels: 'WallSurface', 'RoofSurface', 'GroundSurface'
- Example:

$$\text{building_id} : \begin{cases} \text{WallSurface} : \{[v_j, v_k, v_l, \dots], [v_a, v_b, v_c, \dots], \dots\} \\ \text{RoofSurface} : \{[v_n, v_m, v_p, \dots], \dots\} \\ \text{GroundSurface} : \{[v_i, v_j, v_k, \dots], \dots\} \end{cases}$$

- Each 3D-BAG tile has associated a mapping of vertices indices to actual coordinates (according to the global CRS):
$$\begin{cases} v0 : (x0, y0, z0) \\ v1 : (x1, y1, z1) \\ \vdots \end{cases}$$

3D-BAG data processing

- The workflow iterates through each building in the region of interest and extracts all the surfaces of types WallSurface and RoofSurface. For each polygon the normal and the centroid are saved (more on that later)
- Each of the polygons making up the surfaces are then triangulated



Example of polygon triangulation

Polygon Triangulation

From wikipedia definition: "In computational geometry, polygon triangulation is the partition of a polygonal area (simple polygon) P into a set of triangles, i.e., finding a set of triangles with pairwise non-intersecting interiors whose union is P ".

Why Triangles?

- This has to do with Sionna's graphical engine, Mitsuba, that represents every object in the virtual environment as a set of triangular primitives
- Triangular meshes are also commonly used in several engineering simulation softwares.



Example of a complex object represent in the graphical rendering engine Mitsuba as a set of triangle

Addition of building objects to the Sionna scene

After triangulation, the building objects are added to a Sionna scene. For convenience, the coordinates of all the vertices are expressed relative to a selected origin point within the BBox before being added to the Scene



Buildings only scene (3D-BAG)

Coordinate System Transformation

World to Scene Coordinate Conversion

$$\mathbf{P}_{\text{scene}} = \mathbf{P}_{\text{world}} - \mathbf{O}_{\text{origin}}$$

Where:

- $\mathbf{P}_{\text{scene}}$: Point in scene coordinates (Sionna)
- $\mathbf{P}_{\text{world}}$: Point in EPSG:28992 (RD/NAP)
- $\mathbf{O}_{\text{origin}}$: Scene origin point (x_0, y_0, z_0) . The BBox center or the bottom-left corner are usually selected as origin points

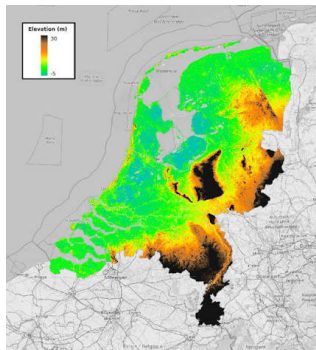
Why Transform?

- Sionna works with local coordinates
- Avoids numerical precision issues
- Large coordinate values in Dutch system

Terrain Rendering with the AHN Dataset

AHN (Actueel Hoogtebestand Nederland) Dataset

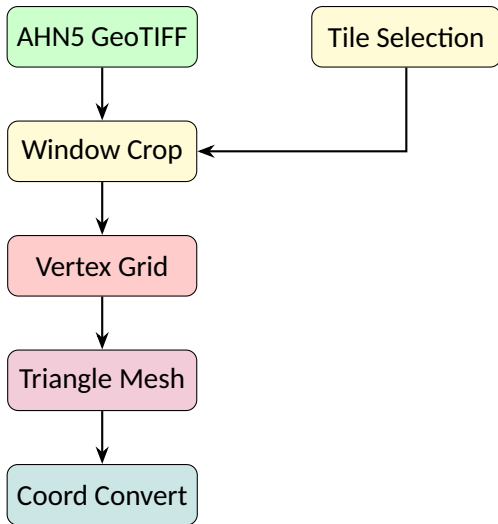
- High-resolution LiDAR-derived elevation data
- Format: GeoTIFF (georeferenced raster grid)
- Coordinate system: EPSG:28992 (RD New) with NP vertical datum
- Resolution: $\approx 0.5\text{m}$ (AHN4)
- Point density: $\approx 8\text{-}10$ points/m²
- Vertical accuracy: $\approx 5\text{cm}$ RMSE



AHN digital model overview

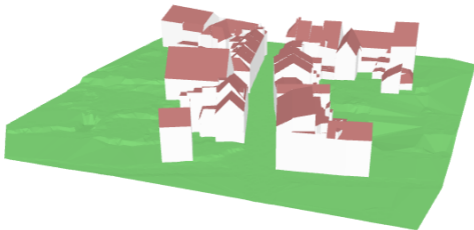
Source: www.ahn.nl

Terrain Processing Pipeline



Addition of terrain information to the Sionna scene

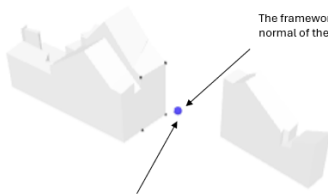
After the terrain processing a single object for the whole terrain is obtained and added to the Sionna scene. Also here the vertices are mapped to the internal sionna scene coordinates.



Buildings (3D-BAG) and terrain (AHN)

Facade material attribution with street-level images

This modality is currently under development. The main idea is to enhance the buildings' facades with semantic maps obtained from street-level view images



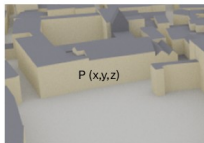
The framework automatically extracts the centroid and the normal of the façade planes facing the streets

The position is then mapped to a geographical coordinate reference system. This information is then used to query the mapillary database to obtain façade images



The images are then used to carry out a semantic segmentation of the materials making up the facade. The map is then projected back onto the 3d model of the façade to enrich the level of detail of the scene

Semantic segmentation workflow



The 3d point coordinates (x,y,z) , the building id and the face id are stored

↓
Query RS database and
obtain the façade image



→
Perform semantic
segmentation
using a pre-trained dedicated
model



Obtain the segmentation map
in which each pixel is assigned
to a specific material

↓
Map the point $P(x,y,z)$ to the
corresponding pixel in the
segmentation map and assign the
correct material to the facade

Wall surfaces filtering

Before querying the image database the walls actually facing a road must be selected. A first filtering eliminates the walls that are 'shared' between adjacent buildings



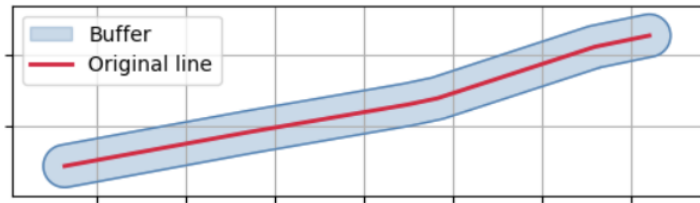
First filtering of the facade walls. For visualization purposes a black dot is placed at the centroid of each selected wall

Wall surfaces filtering

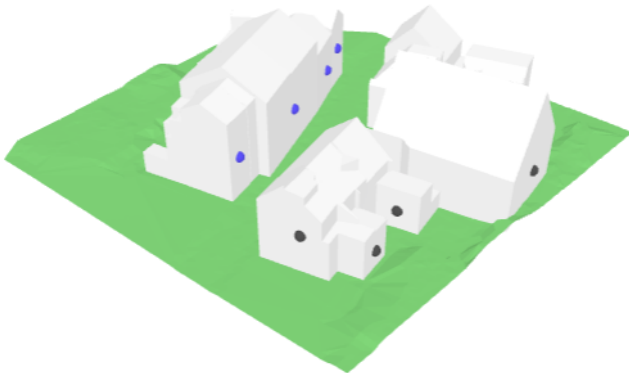
- The first filtering of walls is not enough, since we need to find the walls that are actually facing a road. To do this another modality is integrated in the workflow: OpenStreetMap data.
- The idea is simple: For each wall we find a candidate observation point by moving from the centroid along the normal direction. By using coordinates transformation we can easily go from the scene coordinates to the real world coordinates.
- We can then use this information to query OpenstreetMap databases to see if the point lies on a road or not
- Since we only care about the position on the map and not the elevation (z coordinate) we can simply work with the 2d coordinates of the points and therefore it is possible to change CRS easily during the querying process

Wall surfaces filtering

Side note: OSM provides geometrical information of street as MultiLine objects, i.e. a sequence of consecutive segments. Therefore, for carrying out a geospatial analysis the street polygons are obtained by 'buffering' each segment making up the line



Wall surfaces filtering

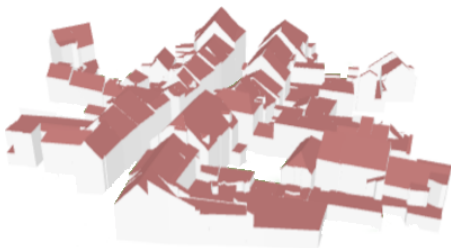


Example of the road-facing filtering with OSM data: Walls that are recognized as facing the road have a blue sphere in the centroid, walls found not facing the road have a black sphere in the centroid

Big picture current/future modalities

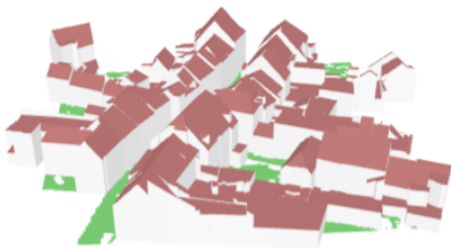
Data Source	What it adds
3D-BAG dataset	Buildings' geometries
AHN dataset	Terrain topology
Mapillary images	Semantic segmentation
Google street view images	Semantic segmentation
LiDAR point clouds f urban environments	even finer geometrical detail of buildings' facades
Object reconstruction (UC 7.6)	even finer geometrical detail of buildings' facades
Regional building information (e.g., municipalities)	even finer geometrical detail of buildings' facades
Street level video cameras	Real time update
3GPP type sensing (ISAC)	Real time update

Effect of Adding different modalities to the Sionna scene



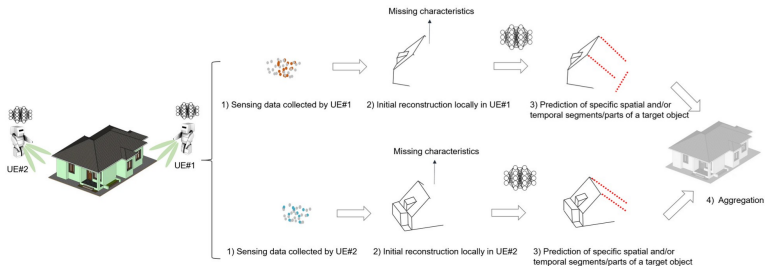
Buildings (D-BAG)

Effect of Adding different modalities to the Sionna scene



Buildings (D-BAG) + Terrain (AHN)

Environment Object Reconstruction



6G UC 7.6: Environment Object Reconstruction (Image credit: 3GPP TR 22.870)

Discussion & Questions

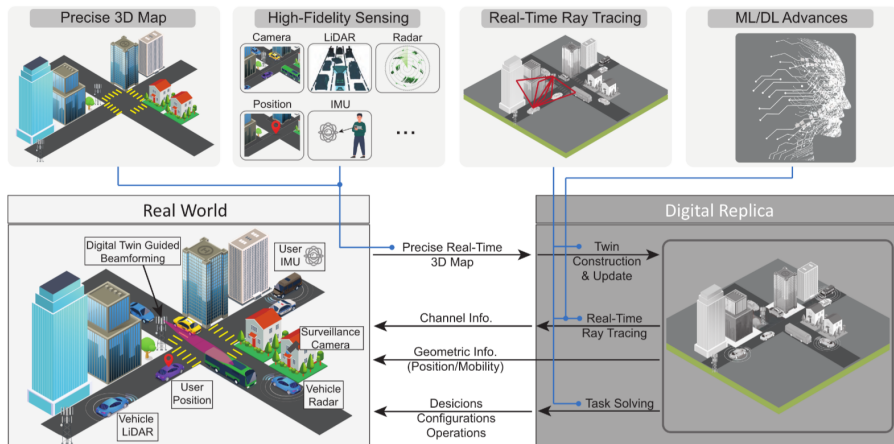
- 1 Which UCs are more interesting for Ericsson?**
- 2 What is the SOTA of hardware, e.g., mmW antenna?**
- 3 Do we need to create a measurement setup using MIMO USRPs?
Or can Ericsson provide mmW antenna for creating datasets?**
- 4 Can we build a p-5G or p-6G at FZJ**

Thanks for your attention

Mobile Network: Digital Twin Networks

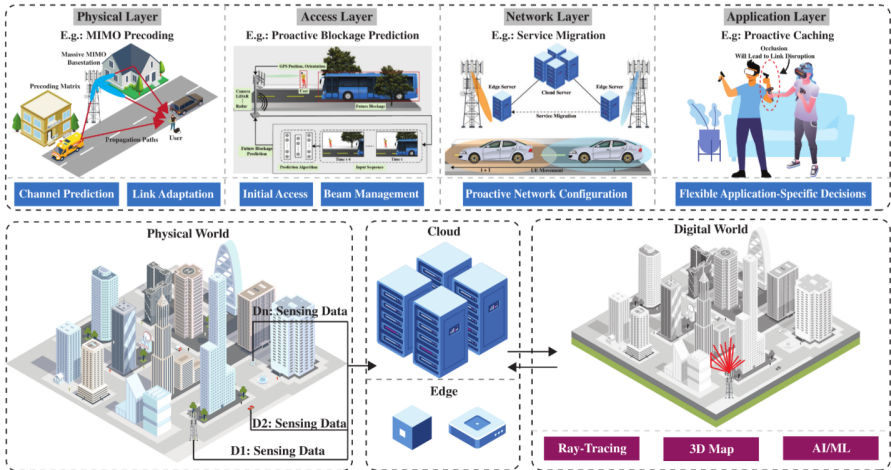
- **Real-time Network Mirroring:** DTNs provide a virtual replica of the physical network, enabling real-time monitoring and optimization.
- **Predictive Management:** Continuous data synchronization allows proactive fault detection and predictive maintenance.
- **AI-driven Optimization:** Digital twins enable AI models to test, train, and refine control strategies without risking live performance.
- **Resource Efficiency:** Facilitates dynamic spectrum, energy, and computing resource allocation — key for sustainable 6G.
- **Closed-loop Automation:** Supports self-learning and self-healing networks, aligning with 6G's zero-touch operation vision.

Mobile Network: Digital Twin Networks



Credit: AI Khateeb et al., 2023, Real-Time Digital Twins: Vision and Research Directions for 6G and Beyond

Applications of Digital Twin Networks



Credit: Al Khateeb et al., 2023, Real-Time Digital Twins: Vision and Research Directions for 6G and Beyond

Ray-tracer at the heart of DTN

A predictive, deterministic techniques that models the received signals by

- Following different copies of the signal
- Follows propagation rules (electromagnetic wave)
- Can be very precise
- But computationally expensive
 - Exponential increase of interactions in complex environments
 - Reflection, refraction and diffuse scattering

Ray-tracer at the heart of DTN

There are plenty of Ray-tracing tools:

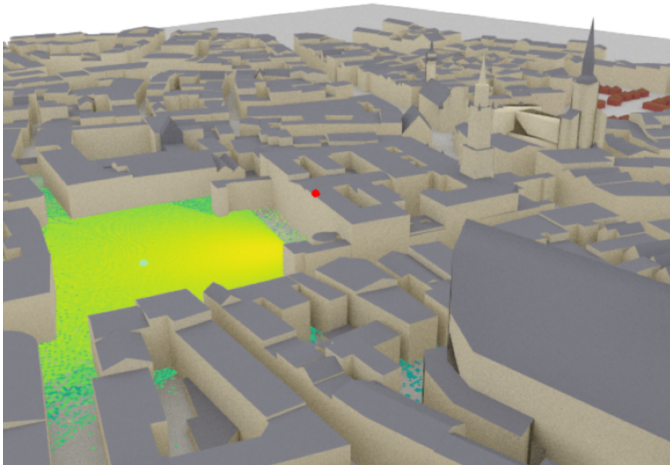
- Not open source (proprietary)
- Slow (not hardware optimized, e.g., **WirelessInSite**)
- Neglect the impact of vegetations and water
- Simplified implementation
- No usage of high-fidelity sensing data (low LoD)

Sionna: Nvidia's Ray-tracer

A tool developed by Nvidia lab Paris:

- ✓ Open source
- ✓ Hardware optimized (a few thousand images in a fraction of sec)
- ✓ Differentiable ray-tracer
- ✗ Neglects the impact of vegetations and water
- ✗ Overlooks the scene details
- ✗ No usage of high-fidelity sensing data (low LoD)

Sionna: Nvidia's Ray-tracer



Credit: Sionna's official web page (<https://nvlabs.github.io/sionna/rt/tutorials/Introduction.html>)

Sionna's path solver

Path solving (finding interaction points) grows, exponentially with increasing number of objects in scene, e.g., buildings, cars, trees, etc.

- Sionna has a fast (heuristic) path solver
- The type of interactions (specular reflection, refraction, scattering) is decided based ϵ_r and μ_r

$$p(\mathcal{R}), p(\mathcal{S}) \sim \frac{|r_{\perp}|^2 + |r_{\parallel}|^2}{|r_{\perp}|^2 + |r_{\parallel}|^2 + |t_{\perp}|^2 + |t_{\parallel}|^2}$$
$$p(\mathcal{T}) \sim \frac{|r_{\perp}|^2 + |r_{\parallel}|^2}{|r_{\perp}|^2 + |r_{\parallel}|^2 + |t_{\perp}|^2 + |t_{\parallel}|^2}$$

where $r_{\perp}$, $r_{\parallel}$, $t_{\perp}$, $t_{\parallel}$ functions of ϵ and θ_i .

Sionna's path solver

$$r_{\perp} = \frac{\cos(\theta_i) - \sqrt{\epsilon_r - \sin^2(\theta_i)}}{\cos(\theta_i) + \sqrt{\epsilon_r - \sin^2(\theta_i)}}$$

$$r_{\parallel} = \frac{\epsilon_r \cos(\theta_i) - \sqrt{\epsilon_r - \sin^2(\theta_i)}}{\epsilon_r \cos(\theta_i) + \sqrt{\epsilon_r - \sin^2(\theta_i)}}$$

$$t_{\perp} = \frac{2 \cos(\theta_i)}{\cos(\theta_i) + \sqrt{\epsilon_r - \sin^2(\theta_i)}}$$

$$t_{\parallel} = \frac{2\sqrt{\epsilon_r} \cos(\theta_i)}{\epsilon_r \cos(\theta_i) + \sqrt{\epsilon_r - \sin^2(\theta_i)}}.$$

Permittivity of Different Material

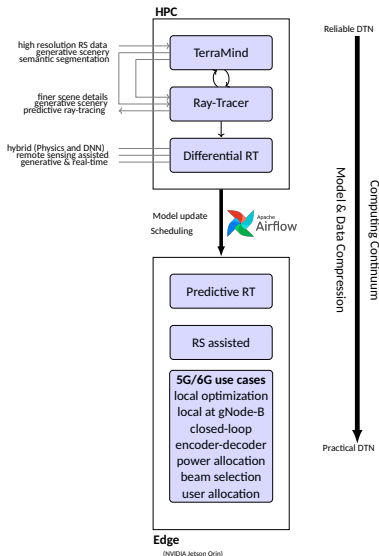
Material	Relative Permittivity (ϵ_r)
Asphalt	2-4
Concrete	5.2
Brick	3.9
Wood (dry)	2.0
Glass	6.3
Polystyrene / Styrofoam	about 2.6

Our Approach

Deploying high-fidelity remote sensing data & semantic segmentation to classify material in the scene

- High resolution satellite data (Maxar) for top view (roof, street)
- Point cloud and data from civil engineering & urban design field for facade classification
 - Includes complicated scene matching techniques, e.g., **multi-view gemoetry**
- Train the big models (accuracy) and compress for edge deployment

Our Vision: From HPC to Edge



Key Challenges and Solutions

Challenges:

- Classical **ray tracers** (RTs) are computationally expensive and inaccurate
- Lack of fine-grained scene details beyond **OpenStreetMap** (OSM)
- Missing physical phenomena (diffuse scattering, refraction)
- Edge devices have limited computational power

Our Solutions:

- High-resolution remote sensing & point cloud data
- **TerraMind** for multimodal generative **foundation model** (FM)
- Hybrid & differentiable RT with artificial intelligence (AI)-augmented physics
- Model compression, pruning, and knowledge distillation

Implementation Roadmap

Our ambition is to implement the **first-of-its-kind** digital twin on a **graphics processing unit (GPU)**-embedded edge device, such as **Jetson AGX Orin**.

Key milestones:

- 1 Achieve ~ 1 – 2 meters per pixel resolution with high spatial resolution remote sensing data
- 2 Higher-quality **channel state information** (CSI) estimation using advanced RT
- 3 Generative ahead-of-time (predictive) ray tracing
- 4 Deployment on embedded GPUs as practical **digital twin network** (DTN)
- 5 Deploy differentiable properties for network optimization
- 6 **machine learning** (ML)OPs using **Apache Airflow** for automation

While full-scale TerraMind runs on HPC, **TerraMind-Edge** will bring compressed, distilled variants to gNode-B.

Glossary / Abbreviations

AI artificial intelligence. 68

CSI channel state information. 69

DTN digital twin network. 69

FM foundation model. 68

GPU graphics processing unit. 69

ML machine learning. 69

OSM OpenStreetMap. 68

RT ray tracer. 68, 69