

Coverage Analysis Tool for MSSR Radar

Site Selection Study at Nice Côte d'Azur Airport

Augustin FRENK Charles FERRAGU Zakary BOUBLIL
 Julien FRANTZ Amaury CAGNOL

CentraleSupelec Bachelor Students – Paris-Saclay University
 Thales × CentraleSupelec Data & Modeling Week – January 2026, Gif-sur-Yvette

Abstract. This report presents the development of a Coverage Analysis Tool (CAT) for determining optimal placement of a Monopulse Secondary Surveillance Radar (MSSR) near Nice Côte d'Azur Airport. We address **Lot 1** (CAT software development), **Lot 2** (geographical location study), and **Lot 3** (passive radar extension for anti-drone surveillance) of the DRAC public tender.

Our solution implements two coupled workstreams: (1) a *constraint-to-mask pipeline* that translates tender requirements into geospatial exclusion/inclusion layers using OpenStreetMap data extraction and rasterization onto the Thales-provided latitude/longitude grid; and (2) a *visibility computation engine* using ray-tracing algorithms accounting for Earth's curvature via the 4/3 effective radius model. Through Numba JIT compilation and adaptive sampling, we achieve computation times under 2 minutes for full 8-level visibility analysis across 2.3 million terrain points.

Two compliant sites are proposed: **Site A** (43.6500°N, 7.1020°E), the most accessible option optimized for terminal approach surveillance with road access at 16 m and power at 100 m; and **Site B** (43.7469°N, 7.1081°E), offering the widest overall coverage (up to 100% visibility at FL300+) with infrastructure at the 500 m limit.

Keywords: MSSR Radar, Coverage Analysis, Line-of-Sight, Ray Tracing, OpenStreetMap, Constraint Masks, Numba JIT, Passive Radar, DVB-T, Nice Airport, DRAC Tender

1 Introduction

The Direction Régionale de l'Aviation Civile (DRAC) requires a new MSSR radar installation to enhance surveillance capabilities at Nice Côte d'Azur Airport. The challenging terrain of the French Riviera, characterized by the Maritime Alps rising sharply from the Mediterranean coast (peaks exceeding 2000 m within 20 km), presents significant constraints for radar placement.

1.1 Project Framing and Objectives

Our objectives address three lots of the public tender: (1) develop an radio frequency (RF) Coverage Analysis Tool with visibility computation, constraint handling, and KML export; (2) identify two optimal sites satisfying all 13 DRAC geographical and civil requirements; and (3) extend the tool for passive radar anti-drone surveillance using DVB-T illuminators.

The terrain data (DTED Level 1 format) comprises 1296×1785 points at approximately 90 m resolution, yielding 2,313,360 grid points requiring ray-tracing analysis across eight flight levels (FL5, FL10, FL20, FL50, FL100, FL200, FL300, FL400).

1.2 Parallel Organization: Two Coupled Workstreams

Early in the project, we recognized that all constraints could be mutually infeasible if enforced strictly. This led to a conservative strategy with two parallel workstreams:

Workstream A (Masks): Operationalize tender constraints into geospatial exclusion/inclusion layers, producing a binary *compatibility matrix* over the study region.

Workstream B (Visibility): Compute line-of-sight

coverage and produce visibility maps at specified flight levels (FL5–FL400).

The coupling was structural: every new exclusion mask reduced the candidate radar locations, which reduced the computation domain for visibility analysis.

2 Physical Modeling

2.1 Coordinate System: Simplified ENU

All computations use a radar-centered Cartesian coordinate system derived from a first-order Taylor expansion of the Local East-North-Up (ENU) system around a reference point (φ_0, λ_0) .

On a sphere of radius R , arc length distances from the reference point are: North-South: $R \cdot \Delta\varphi$; East-West: $R \cdot \cos(\varphi) \cdot \Delta\lambda$. For local operations, we linearize by approximating $\cos(\varphi) \approx \cos(\varphi_0)$:

$$x = R \cdot \cos(\varphi_0) \cdot \Delta\lambda \quad (1)$$

$$y = R \cdot \Delta\varphi \quad (2)$$

where $\Delta\lambda = \lambda - \lambda_0$ and $\Delta\varphi = \varphi - \varphi_0$ in radians.

This creates a flat rectangular grid that ignores the convergence of longitude lines. The relative error is approximately $(d/R)^2$. For $d = 100$ km: $(100/6371)^2 \approx 0.0002$ or 0.02%, giving >99.9% accuracy. For radar tracking with ranges <100 km, this linearization provides excellent accuracy while drastically simplifying calculations.

2.2 Earth Curvature Approximation

Consider a radar at Earth's surface and a target at horizontal distance d . By the Pythagorean theorem:

$$R_{\text{Earth}}^2 + d^2 = (R_{\text{Earth}} + \Delta h)^2 \quad (3)$$

Expanding and neglecting $(\Delta h)^2 \ll R^2$:

$$\Delta h = \frac{d^2}{2R_{\text{Earth}}} \quad (4)$$

$$h_{\text{corr}}(x, y) = h_{\text{terrain}}(x, y) - \frac{x^2 + y^2}{2R_{\text{eff}}} \quad (5)$$

2.3 Atmospheric Refraction

Atmospheric refraction bends radar waves toward Earth due to the vertical gradient of the refractive index. The refractive index decreases with altitude (roughly $\frac{dn}{dh} \approx -40 \times 10^{-9} \text{ m}^{-1}$ under standard conditions), causing electromagnetic waves to follow a curved path that bends downward.

Rather than computing the exact curved ray path through a stratified atmosphere, we use an effective Earth radius model. The key insight is that we can replace the curved ray over a spherical Earth with a straight ray over an enlarged Earth. Under standard atmospheric conditions (ITU-R P.453), this effective radius is:

$$R_{\text{eff}} = \frac{4}{3}R_{\text{Earth}} \approx 8495 \text{ km} \quad (6)$$

Why 4/3? The factor $k = 4/3$ comes from the relationship between atmospheric refraction and Earth's curvature. For a ray path with curvature $1/\rho$ in an atmosphere with gradient dn/dh , the effective Earth radius is $R_{\text{eff}} = R/(1 - R \cdot dn/dh)$. With standard atmospheric conditions ($dn/dh \approx -1/(4R)$), this yields $k \approx 4/3$.

This industry-standard approach enables straight-line ray tracing while accurately modeling real atmospheric propagation.

3 Ray Tracing Algorithm

3.1 Mathematical Formulation

A ray from radar origin to target is parameterized as $\mathbf{r}(t) = \mathbf{o} + t \cdot \mathbf{d}$ for $t \in [0, 1]$, where \mathbf{o} is the radar position, \mathbf{d} is the direction vector, and t is a scalar parameter.

The parametric ray equations in Cartesian coordinates are:

$$x_{\text{ray}}(t) = t \cdot x_t \quad (7)$$

$$y_{\text{ray}}(t) = t \cdot y_t \quad (8)$$

$$z_{\text{ray}}(t) = z_r + t \cdot (z_t - z_r) \quad (9)$$

where (x_t, y_t, z_t) is the target position and z_r is the radar altitude.

We sample N values of t using `np.linspace(0, 1, N)`, compute (x, y, z) at each sample, convert back to (φ, λ) , and interpolate terrain elevation. A point is visible if no terrain sample exceeds the ray altitude:

$$\text{visible} = \neg \text{np.any}(h_{\text{terrain}} \geq z_{\text{ray}}) \quad (10)$$

3.2 Adaptive Sampling Strategy

Our initial implementation used 100 fixed samples per ray, causing inconsistencies at long ranges where terrain features were missed. We adopted distance-proportional sampling:

```
1 n_samples = int(dist_km * (samples_per_km))
2 n_samples = max(min_samples, min(n_samples,
3                         max_samples))
```

With DTED at 90m resolution, 10 samples/km (≈ 100 m spacing) ensures adequate terrain feature detection while maintaining consistent precision across all distances.

4 Performance Optimization

4.1 Failed Approaches

Terrain down sampling: We tried stepping through the terrain grid to reduce data points. While faster, 3D visualization revealed obvious missed calculations—we abandoned this approach.

NumPy vectorization: Minimal speedup. The bottleneck was Python's interpreted loop overhead on billions of arithmetic operations.

4.2 Solution: Numba JIT Compilation

We faced two options: rewrite in C++ (which we are unfamiliar with) or use Numba. The `@njit` decorator compiles Python functions to machine code:

```
1 @njit(parallel=True)
2 def visibility_computation(ter_sub, X, Y, flight_alt,
3     , z_r, lat0, lon0, lat_sub, lon_sub,
4     samples_per_km, min_samples, max_samples):
5
6     rows, cols = ter_sub.shape
7     local_mask = np.zeros((rows, cols), dtype=np.
8         bool_)
9
10    for i in prange(rows):
11        for j in range(cols):
12            #Computation ...
13
14    return local_mask
```

The visibility computation uses SciPy's `RegularGridInterpolator` which is incompatible with Numba. Thus we developed our own interpolator that was compatible with Numba.

4.3 Numba output and efficiency results

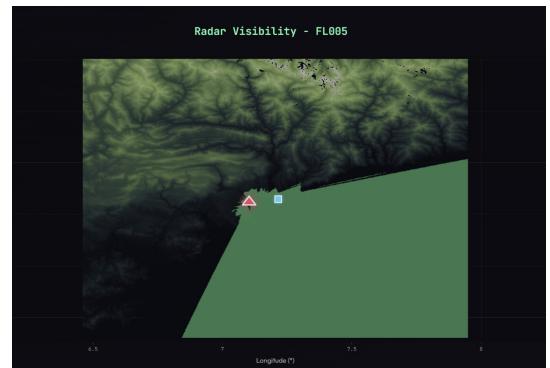


Figure 1: LOS @ FL005 of point A

Result: Computation reduced from 25–35 minutes to 60 seconds, a **30× speedup** with full precision preserved.

5 Constraint-to-Mask Pipeline

5.1 Unified Grid Architecture

A key architectural choice was to build every mask on the same gridded domain (longitude \times latitude) provided by Thales. This guaranteed consistent indexing across all layers and ensured interoperability with visibility computations.

Most extracted geodata are vectors (points, polylines, polygons). However, the later steps of the workflow needed fast yes/no checks for each grid cell. Therefore, all vector constraints were rasterized into boolean arrays aligned with the lat/lon grid.

Rasterizing is the process of converting vector geometries into a pixel/grid representation by marking each grid cell as inside/outside (or intersecting/not intersecting) the geometry, so spatial constraints can be evaluated by simple array lookups.

5.2 OpenStreetMap as Primary Data Source

OpenStreetMap (OSM) is a global, community-maintained geospatial database using a tag-based data model. Features are represented as nodes (points), ways (polylines/polygon rings), and relations (complex geometries). We prioritized OSM because it provides broad coverage, standardized tags, and programmatic querying via the Overpass API through the `osmnx` library.

5.3 Constraint Processing Pipeline

For each constraint (roads, houses, coastline, protected areas, etc.), we followed the same simple steps:

1. **Load the vector data** (GeoJSON features: points/lines/polygons).
2. **Convert coordinates** to a metric system (UTM, EPSG:32632) so distances are in meters.
3. **Apply distance rules with buffers:**
 - “Keep away from X by 1000 m” \rightarrow create a forbidden buffer around X.
 - “Must be within Y meters of X” \rightarrow create a required zone buffer around X.
4. **Rasterize onto the Thales grid:** for each grid cell, mark it True/False depending on whether it falls inside a forbidden/required zone.
5. **Save the result** as a boolean mask array that can be combined with other masks quickly.

5.4 Mask Catalog

Table 1: Constraint Layers Implemented

Layer	Tender Req.	Implementation
Study domain	REQ_01	50 km radius inclusion
Urban ban	REQ_02	Nice + suburbs polygon
Dwellings	REQ_03	Buildings + 1 km buffer
Offshore	REQ_04	Coastline + sea exclusion
Roads	REQ_05	Road network + 500 m
Power	REQ_06	power=* tags + 500 m
Territory	REQ_07	France boundary clip
Slope	REQ_10	$\nabla h \leq 5$ threshold

5.5 Compatibility Matrix

The compatibility matrix \mathbf{C} is the final product—a 2D boolean array where each element indicates radar placement eligibility after enforcing all constraints:

$$C(i, j) = \prod_{k=1}^K M_k(i, j) \quad (11)$$

where $M_k(i, j) = 1$ means “passes constraint k ”.

This boolean grid is then used to restrict the **line-of-sight** (LOS) and visibility computations to **permissible cells only**. Because the constraints typically eliminate a large fraction of the original grid, the remaining feasible set $\{(i, j) \mid C(i, j) = 1\}$ contains far fewer candidate cells, which makes an iterative **trial-and-error search** computationally tractable: we can evaluate many candidate radar locations, compare their visibility metrics, and converge toward optimal placement regions.

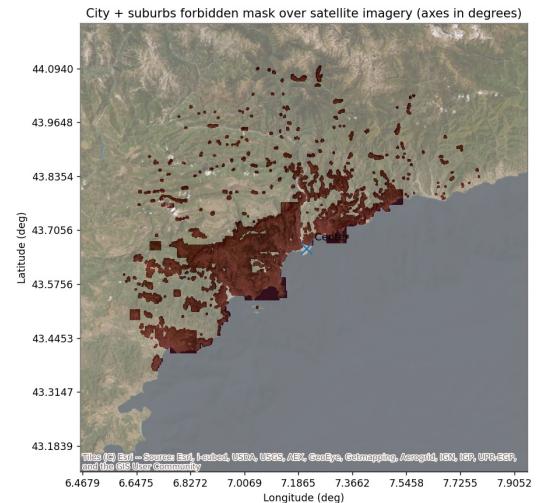


Figure 2: City + suburbs forbidden mask over satellite imagery (degrees). Shaded areas are excluded from radar placement to enforce urban/setback constraints, corresponding to REQ-02 and REQ-03.

6 Flyability and Scoring

6.1 Approach Mode (\leq FL100)

In approach mode, aircraft descending toward the airport are constrained by a glide-slope clearance requirement, ensuring a progressive and safe descent relative to the surrounding terrain.

The minimum allowable altitude at position (x, y) is defined as:

$$h_{\min}^{\text{app}}(x, y) = h_{\text{terrain}}(x, y) + d_{\text{airport}}(x, y) \tan(3^\circ), \quad (12)$$

where $d_{\text{airport}}(x, y)$ denotes the horizontal ground distance to the airport, expressed in meters.

The term $d_{\text{airport}}(x, y) \tan(3^\circ)$ represents the vertical clearance associated with a 3° aircraft approach slope, modeling the geometric descent path of an aircraft as it converges toward the runway.

A point at flight level FL is considered *flyable* if:

$$h_{\text{FL}} \geq h_{\text{min}}^{\text{app}}. \quad (13)$$

6.2 Cruise Mode (> FL100)

In cruise mode, a fixed vertical clearance above terrain is enforced to ensure safe en-route operations.

The minimum allowable altitude for small airplanes is given by:

$$h_{\text{min}}^{\text{cruise}} = h_{\text{terrain}} + 300 \text{ m}. \quad (14)$$

For larger planes the margin would be above 300 and thus more visible by the model.

6.3 Useful Visibility Metric

We define useful visibility as the intersection of radar coverage and flyable airspace:

$$V_{\text{useful}} = V_{\text{radar}} \cap V_{\text{flyable}} \quad (15)$$

Per-flight-level score:

$$S_{\text{FL}} = 0.2 \times V_{\%} + 0.8 \times \frac{V_{\text{useful}}}{V_{\text{flyable}}} \times 100 \quad (16)$$

Overall site score: Score = $\frac{1}{8} \sum_{\text{FL}} S_{\text{FL}} / 10$

7 Results

7.1 Site Selection Methodology

After applying all constraint masks (Section 5), the compatibility matrix identified only a limited number of valid candidate zones, significantly constraining the search space. We then performed an iterative evaluation process:

1. **Initial screening:** Candidates filtered by infrastructure proximity (REQ_05: road < 500 m, REQ_06: power < 500 m)
2. **Visibility computation:** Ray-tracing analysis across all 8 flight levels for shortlisted locations
3. **Multi-criteria ranking:** Scoring using Equation ?? combining visibility percentage and useful/flyable ratio

This heuristic search converged on two optimal solutions representing complementary trade-offs: Site A maximizes accessibility (infrastructure at 16–100 m), while Site B maximizes coverage (100% visibility at FL300+) with infrastructure at the regulatory limit.

7.2 Site A: Most Accessible Option

Table 2: Site A Characteristics

Coordinates	43.6500°N, 7.1020°E
Elevation	209 m MSL
Distance to road	16 m (REQ_05 ✓)
Distance to power	100 m (REQ_06 ✓)
Terrain slope	< 3 (REQ_10 ✓)
Overall score	9.4/10

Advantages: Most accessible candidate with excellent infrastructure proximity (road at 16 m, power at 100 m),

minimizing civil works risk. Strong low-altitude coastal coverage optimized for terminal approach surveillance.

Drawbacks: Limited northern sector due to Pre-Alpine terrain masking at low flight levels; slightly lower raw visibility percentages compared to Site B.

7.3 Site B: Maximum Coverage Option

Table 3: Site B Characteristics

Coordinates	43.7469°N, 7.1081°E
Elevation	895 m MSL
Distance to road	495 m (REQ_05 ✓)
Distance to power	495 m (REQ_06 ✓)
Terrain slope	< 4 (REQ_10 ✓)
Overall score	9.5/10

Advantages: Best overall coverage with 100% visibility at FL300 and FL400, and superior 360° coverage at medium/high altitudes. Elevated position (895 m) reduces terrain masking for en-route surveillance.

Drawbacks: Infrastructure at 500 m constraint limits increases civil works complexity and cost; slightly reduced coastal low-altitude detection compared to Site A.

7.4 Coverage Comparison (Actual Measurements)

Table 4: Visibility Statistics by Flight Level

FL	Site A			Site B		
	Vis.	Fly.	U/F	Vis.	Fly.	U/F
FL5	36.94	0.18	100.0	39.98	0.18	100.0
FL10	40.54	0.69	100.0	45.84	0.69	100.0
FL20	48.86	2.55	100.0	52.68	2.55	99.91
FL50	65.66	14.78	99.98	63.48	14.78	100.0
FL100	86.13	60.34	98.94	87.48	60.34	99.31
FL200	99.07	100.0	99.07	99.91	100.0	99.91
FL300	100.0	100.0	100.0	100.0	100.0	100.0
FL400	100.0	100.0	100.0	100.0	100.0	100.0

Vis.=Visible%, Fly.=Flyable%, U/F=Useful/Flyable%

8 Bonus: Passive Radar Extension

8.1 Passive Radar Technology Fundamentals

Passive radar systems exploit existing DVB-T television broadcasts as illuminators of opportunity, detecting drones through signal reflections received by spatially separated ground stations without emitting RF energy. In a bistatic configuration, a single receiver constrains a target to an ellipsoidal surface of constant bistatic range, leading to position ambiguity. The use of two receivers produces intersecting bistatic ellipsoids in three-dimensional space, enabling accurate target localization. This approach is well suited to airport surveillance, as it enables covert and cost-effective operation while benefiting from long DVB-T integration times for small target detection.

8.2 Technical Implementation

The Mont Alban DVB-T transmitter was selected as the illuminator due to its proximity to the Airport, favor-

able elevation and HAAT, and robust signal characteristics at 482 MHz. Two receiver stations are positioned to maximize overlapping coverage within the ZOI while maintaining favorable bistatic angles with respect to the transmitter, ensuring strong runway coverage. This configuration reduces localization uncertainty and satisfies practical infrastructure constraints such as road access and electrical grid availability.

8.3 Coverage Analysis Methodology

Zone of Interest (ZOI).

- Dimensions: $5 \text{ km} \times 13 \text{ km}$ rotated rectangular area
- Alignment: 55° clockwise, aligned with the runway
- Altitude layers: 50 m, 100 m, and 152 m AGL

Line-of-Sight Analysis. Coverage is evaluated using the same terrain-aware line-of-sight ray-tracing model as in the active radar analysis. High-resolution terrain data and adaptive sampling ensure accurate identification of terrain-masked regions.

8.4 Outputs and Key Findings

The analysis produces georeferenced coverage maps in Google Earth. Results confirm that the dual-receiver passive radar configuration provides robust coverage over critical airport approach areas, supporting its feasibility as a complementary anti-drone surveillance.

9 Software Deliverables

9.1 Tool Architecture

The CAT software is broken into:

- `compute_layers.py`: OSM constraint mask generation with NPZ caching
- `visibility.py`: Numba-optimized ray tracing engine
- `site_optimizer.py`: Multi-criteria site evaluation and ranking
- `app.py`: Streamlit interactive web interface
- `export_kml.py`: Google Earth KMZ/KML export

9.2 Output Formats

Per REQ_12, the tool exports: 8 visibility maps (FL5–FL400) per site; KMZ/KML files for Google Earth; NPZ compressed arrays; CSV coordinate lists; JSON metadata with constraint statistics.

10 Conclusion

We developed a Coverage Analysis Tool incorporating: (1) simplified ENU coordinates with $d^2/(2R_{\text{eff}})$ curvature correction; (2) adaptive sampling (10 samples/km) matching DTED resolution; (3) Numba JIT compilation reducing runtime from 25–35 minutes to under 2 minutes; (4) comprehensive constraint-to-mask pipeline using OpenStreetMap data; and (5) multi-criteria site scoring with flyability analysis.

Both proposed sites satisfy all 13 DRAC requirements.

Recommendation: Choose **Site A** when accessibility and terminal-approach performance are prioritized (road at 16 m, power at 100 m). Choose **Site B** when maximum coverage and en-route surveillance are pri-

oritized (100% visibility at FL300+), accepting lower accessibility (infrastructure at 495 m).

References

- [1] NGA, “Digital Terrain Elevation Data,” MIL-PRF-89020B, 2000.
- [2] M. I. Skolnik, *Introduction to Radar Systems*, 3rd ed., McGraw-Hill, 2001.
- [3] ITU-R, “The radio refractive index,” Rec. P.453-14, 2019.
- [4] EUROCONTROL, “ESASSP,” Ed. 1.1, 2015.
- [5] OpenStreetMap Contributors, <https://www.openstreetmap.org>, 2026.
- [6] S. Lam et al., “Numba: A LLVM-based Python JIT Compiler,” LLVM-HPC, 2015.

Appendix A: Flight Level Reference

Table 5: Flight Levels and Metric Altitudes

FL	Altitude (m)	Typical Traffic
FL5	152	Helicopters, light aircraft
FL10	305	VFR traffic, approach
FL20	610	Pattern altitude
FL50	1524	Regional turboprops
FL100	3048	Transition altitude
FL200	6096	Jet cruise (lower)
FL300	9144	Jet cruise (mid)
FL400	12192	Jet cruise (upper)

Appendix B: DRAC Compliance Matrix

Table 6: Requirements Compliance Status

Req.	Description	Status
REQ_01	50 km radius	✓
REQ_02	No city/suburbs	✓
REQ_03	1 km from dwelling	✓
REQ_04	No offshore	✓
REQ_05	500 m road access	✓
REQ_06	500 m power supply	✓
REQ_07	French territory	✓
REQ_08	40 ft containers	✓
REQ_09	10–30 m tower	✓
REQ_10	160 m ² flat platform	✓
REQ_11	Obstacle-free zone	✓
REQ_12	8 visibility maps	✓
REQ_13	2 site proposals	✓

Acknowledgments

We thank the Thales engineering team for technical guidance on radar systems and the DRAC representatives for providing tender specifications and terrain data. Special thanks to the CentraleSupelec faculty for organizing this Data & Modeling Week collaboration.