BALROG.run

VALIDATION & BENCHMARK SUMMARY · FOR TECHNICAL REVIEW

Longley-Rice ITM on GPU: what was measured, and against what

A full-area coverage map of 5.95 million receiver points in **92 ms** on a consumer GPU, checked against the NTIA/ITS reference implementation to **0.000 dB on smooth ground** and **1.122 dB worst case** on 108 m of relief.

Measured 4–5 August 2026 · GeForce RTX 3060 (Ampere, sm_86, CUDA 12.9) against an Intel i5-8600K in the same desktop · SRTM1 tile N48E002, 0.6° domain, 800 MHz · every figure from a logged run. This document may be circulated freely.

1 · The baseline, and why it is not ours

The yardstick is the **ITM reference implementation published by the U.S. NTIA's Institute for Telecommunication Sciences** — the code the FCC's OET-69 procedure rests on. It is public source, it is buildable, and it is not ours. A speed ratio or an agreement figure means nothing when the vendor controls both ends of it.

SAME GRID: 5.95 M POINTS, 0.6°, 800 MHZ, SRTM1 30 M	TIME	VS GPU
BALROG GPU — RTX 3060, full pipeline	92 ms	—
BALROG GPU — further scenario, same terrain	51 ms	—
NTIA reference, sharded over 6 cores	11.4 s	124×
NTIA reference, 1 core — the published ITM code	55.0 s	598×
For reference: BALROG on a Tesla P100 (2016)	*522 ms*	—

The reference is compiled `-O3 -march=native` from unmodified public source and handed its terrain profiles **pre-built**, so what is timed is the model's arithmetic and not its file handling. Sharding across all six physical cores is the best a user could realistically do with it: **124× is the figure that survives that objection**. Sharding efficiency is 4.8× on six cores, as expected once memory traffic is shared.

> **A correction that ran against us.** An earlier version of these figures quoted 38.9 s for the reference on one core. That was measured on profiles 851 points long; the run it was compared against uses profiles of 1 336, and a Longley-Rice call costs more the longer its profile. Re-measured properly: **55.0 s** (median of five). The error made the reference look faster than it is — the comfortable direction, which is exactly why it is published.

2 · Agreement with the reference

Smooth ground: the model alone

Flat ground removes terrain from the comparison, so what remains is the model itself. Sixteen distances from 3.0 to 25.5 km, 800 MHz, receiver at 2 m, at the median quantile:

TRANSMITTER HEIGHT	WORST GAP TO THE REFERENCE, 16 DISTANCES
1000 m	0.000 dB — identical at every distance
23 m	0.055 dB

Real relief: the comparison that matters

Feeding the reference its own reading of the terrain tile would measure a difference in terrain *sampling*, not in model. So the engine dumps the elevation profile it actually uses — 851 points at 30 m spacing, elevations 44 to 152 m — and the reference is driven point-to-point on exactly that array. Same relief, same parameters, only the model differs.

SIXTEEN PROBES, 108 M OF RELIEF	FIRST RESULT	CURRENT
Worst gap to the reference	15.104 dB	1.122 dB
Median gap	−1.01 dB	−0.01 dB
Probes within 0.2 dB	4 of 16	14 of 16
Δh values of zero on real relief	11 of 851	0

Two residuals remain unexplained: +1.12 dB at 13.5 km and +0.40 dB at 4.5 km, both with the receiver in the clear. They are stated rather than absorbed into the median.

3 · What the first honest comparison exposed

The 15 dB above is included deliberately: it is the difference between a validation and a press release. The bias was systematic — the engine under-predicted loss almost everywhere — which points at a term rather than at numerical noise. The irregularity parameter Δh was **20 to 45 m low at every distance**, and exactly 0.0000 at 25.5 km on terrain carrying 108 m of relief. Two distinct defects were behind it.

The estimator, replaced rather than patched

The reference's `ComputeDeltaH.cpp` does three things the GPU version did not: it **resamples** the interval to n points (35 to 245); it fits a straight line and **subtracts it**, so deciles are taken on departure from the slope and a steadily rising valley does not read as roughness; and it takes the deciles **exactly**, without iterating. All three are transcribed. The exact order statistics fit in a bitonic sort of 256 elements — 36 block barriers against up to a hundred passes for the loop it replaced. The replacement is **188 lines shorter and faster**; it was not a trade.

A border defect in the block fill

Δh is computed one point in `incr`; the fill copied, across each block, the value taken at the *start of the next block*. At the edge of the domain that point does not exist. Measured: the domain ends at index 853, the block covering 848–855 reaches for 856, and Δh collapsed to zero on three points where the neighbourhood reads 116.7 m. **That single fault was the whole of the 15 dB.** The fill now falls back to the block's own start, which is always a computed point.

> **Method note.** The first attempt to dump the elevation profile used a device-side `printf`. That silently loses data — the FIFO flushes per block and drops the rest — returning 270 lines out of 1333, in clumps of about thirty, with no warning. The array is copied back host-side instead.

4 · Measurement hygiene

Consecutive runs of the same binary on the same data gave 126, then 202, then 62, then 62 ms for one stage — bimodal and unrelated to run order. Recording `clocks.sm` before each launch explained it: the card idles at 210 MHz, nine times below its working clock, and the first run after a pause spends its heaviest stage ramping rather than computing.

CLOCK BEFORE LAUNCH	TOTAL	ΔH STAGE
210 MHz (idle)	267.3 ms	214.6 ms
540 MHz	113.7 ms	62.9 ms
1942 MHz (working clock)	107–116 ms	57–65 ms

The rule every figure here follows: **wake the card, take the median of several runs, verify the clock, never publish a first run** — both sides of any comparison in the same sitting.

An unplanned cross-check

A warm NVIDIA T4 returns 34.3 ms for 1 488 291 points — 23.0 ns per point. The RTX 3060 does 92 ms for 5 950 000, or 15.5 ns. The ratio between the cards is **1.49**; their published fp32 throughput ratio is **1.57**. Nothing is left unexplained by the silicon, which is the strongest available evidence that the port is sound: the engine is limited by the hardware, not by anything clever or anything broken.

5 · Caveats, stated rather than waited for

- **Scaling is not linear.** Cost grows as points$^{1.2}$: doubling the receiver points costs about 2.3×. Figures for other domain sizes must be scaled, not interpolated.
- **Terrain preparation is paid once.** Of the first 92 ms, 41 ms is terrain. Every further frequency, mast height or reliability quantile over the same ground costs 51 ms — the figure that makes coverage volumes and moving transmitters affordable at all.
- **ITM only, and not as a roadmap gap.** What was parallelised is the ITM algorithm itself — the method filed with the INPI (June 2017, May 2018) is specific to how this model walks its terrain profiles, not a wrapper any propagation model can be dropped into. Porting another model to the GPU is original mathematical work, not an engineering task with a date. If your study must be ITWOM or ITU-R P.528, this is not the engine.
- **Two residuals above 0.2 dB are unexplained**, as stated in section 2.

6 · Reproducing this

Nothing above requires trusting this document. The baseline is public source; the tile is a standard SRTM1 tile; the parameters are five plain-text files. Anyone repeating the benchmark needs three things: the real profile length (1 336 points, not 851), the same terrain, and `-march=native` — dropping to `-march=x86-64-v2` costs the reference about 15 %, in its disfavour.

The **90-day evaluation** exists to repeat every comparison here on your terrain, your machine, and whatever tool you already trust. It is the production engine with an expiry date and a watermark on its output; the attenuation figures themselves are untouched. Licensing is a single Ed25519-signed file verified offline — no activation server, no callback, no network access at any point, and nothing is metered or reported back.
