PaveSplat:

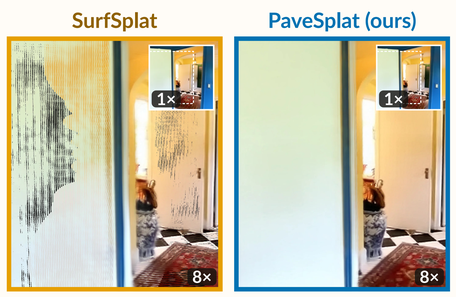
Resolution-Robust Feed-Forward Gaussian Splatting with Residually Corrected Surface-Aligned Covariances

Janghun Kim ✉ · Hyun-Cheol Kim · Eunbin An · Myeonghun Oh · Hyon-Gon Choo
Electronics and Telecommunications Research Institute (ETRI), South Korea

ACCV 2026 · Osaka, Japan · Dec 14–18, 2026

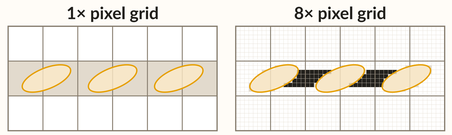
Feed-forward 3DGS predicts covariances (scale, rotation) that work well at the training resolution — but re-render the same prediction at a higher resolution and holes open. PaveSplat keeps the analytic surface-aligned frame and learns one identity-initialized rotation residual Δq, so surfels keep paving the surface.

1 The problem: holes at high resolution



Same prediction re-rendered at 8× (2048^2)

- Most feed-forward 3DGS methods predict pixel-aligned Gaussians — typically **one per pixel** — from a few input views in a single pass.
- At 1×, each surfel's **one-pixel minimum footprint** hides the gaps between loosely placed surfels.

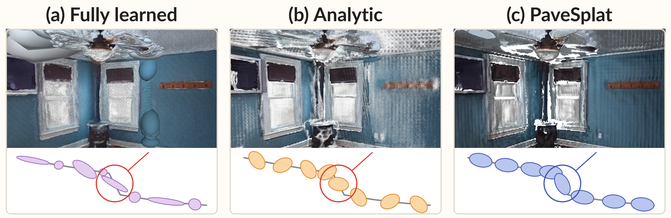


Schematic: at 1× the one-pixel minimum footprint fills the gaps; at 8× the same surfels leave uncovered pixels.

- Re-render the same prediction on a 2×–8× denser grid and the gaps **show up as holes**.
- Root cause: depth sets each surfel's size from its pixel footprint, but that size only covers the surface if the orientation is right — and analytic surfels have **no learned orientation** to fix mis-aimed ones at depth edges, thin structures and grazing surfaces.

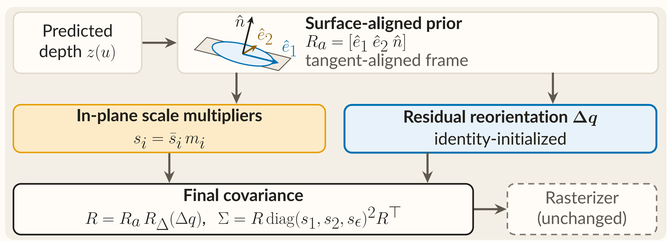
2 Method: a correctable surface-aligned covariance

Three ways to cover a surface with surfels



Why "PaveSplat"?

A metaphor, not an acronym: flat, elongated surfels tile the visible surface like paving stones — aligned with its tangents, turned by Δq where needed — and keep paving it across rendering resolutions.

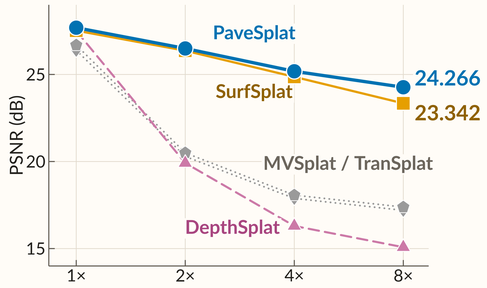


$q = q_a \otimes \Delta q$ $\quad \Delta q$ = identity at initialization

- **Surface-aligned prior:** in-plane axes follow the measured tangents, so the anisotropic radii act where they were measured.
- **Residual reorientation Δq:** the only new learned covariance parameter — 9,512 parameters (<0.05% of the model), +2% encoder time.
- **Identity-initialized:** training starts exactly at the analytic frame and rotates a surfel only where rendering asks for it.

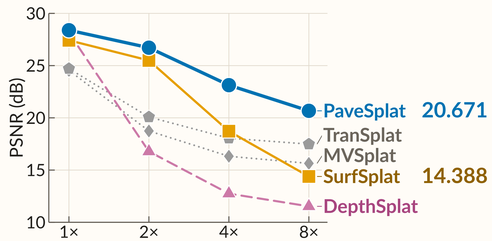
3 Results: robust high-resolution re-rendering

RealEstate10K — PSNR vs. render scale



1×–8× best PSNR, SSIM & LPIPS among **single-Gaussian-per-pixel** methods at every scale

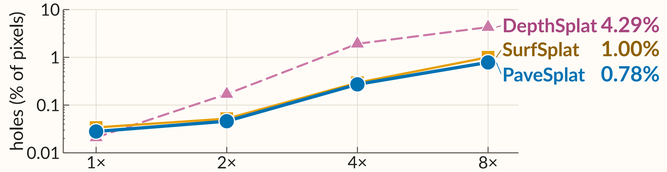
Zero-shot DL3DV (trained on RealEstate10K)



PaveSplat **leads PSNR** among single-Gaussian-per-pixel methods at every scale. Zero-shot ScanNet: slightly behind at 1× (−0.10 dB), ahead at 8× (+2.511 dB).

4 Why it works: mechanism evidence

Fewer coverage holes when re-rendered



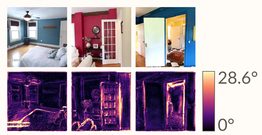
Hole = pixel rendered near-black (max RGB < 0.02) where the ground truth has content; mean over 6,474 test scenes.

Design path (identical recipe per row)

Configuration	$s_{\min}/s_{\max}$ ↓	holes 8× ↓	PSNR 8× ↑
SurfSplat (same recipe, normal-only frame)	0.667	1.00%	23.342
+ tangent-aligned frame	0.580	1.11%	23.850
+ residual Δq (ours)	**0.585**	**0.78%**	**24.266**

The last step changes only Δq, so it isolates the residual (step 1 also adopts our camera-space formulation).

Δq is used — it does not collapse to identity



Mean residual angle **0.160 rad (≈9.1°)**, p99 1.073 rad (≈61.5°), concentrated at **depth discontinuities and grazing surfaces**.
top: input · bottom: residual angle $|\Delta q|$

Scope & limits

- 2×–8× re-rendering is a **coverage stress test**, not native high-resolution capture.
- Native-resolution gain is **small**; RealEstate10K metrics are near saturation.
- Far-domain outdoor **ACID**: tied at native (+0.04 dB), behind at 4× (−1.40 dB).
- Δq adapts **orientation, not depth**: zero-shot ScanNet normal error is modestly lower (median 36.32° → 35.07°).

This work was supported by an ETRI grant funded by the Korean government [26ZC1100, Development of Spatial Media Technology and Interaction Technology for Convergence of the Real and Virtual World].
Builds on SurfSplat (ICLR 2026) and DepthSplat (CVPR 2025); surfel formulation of 2DGS (SIGGRAPH 2024).

Project page
paper · code · model
janghunn.github.io/pavesplat
✉ janghun@etri.re.kr