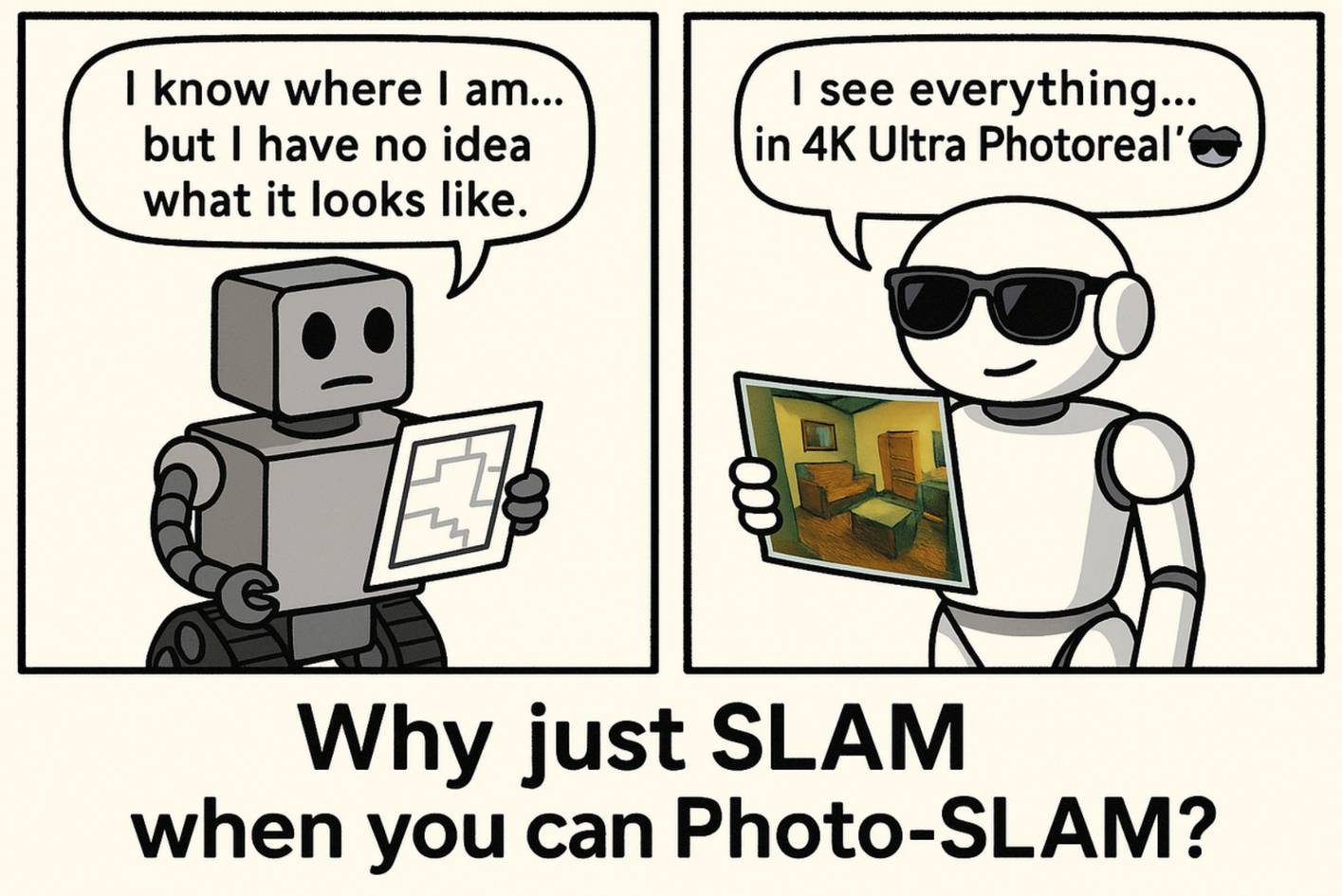
PHOTO SLAM

REAL-TIME SIMULTANEOUS LOCALIZATION AND PHOTOREALISTIC MAPPING FOR MONOCULAR, STEREO, AND RGB-D CAMERAS

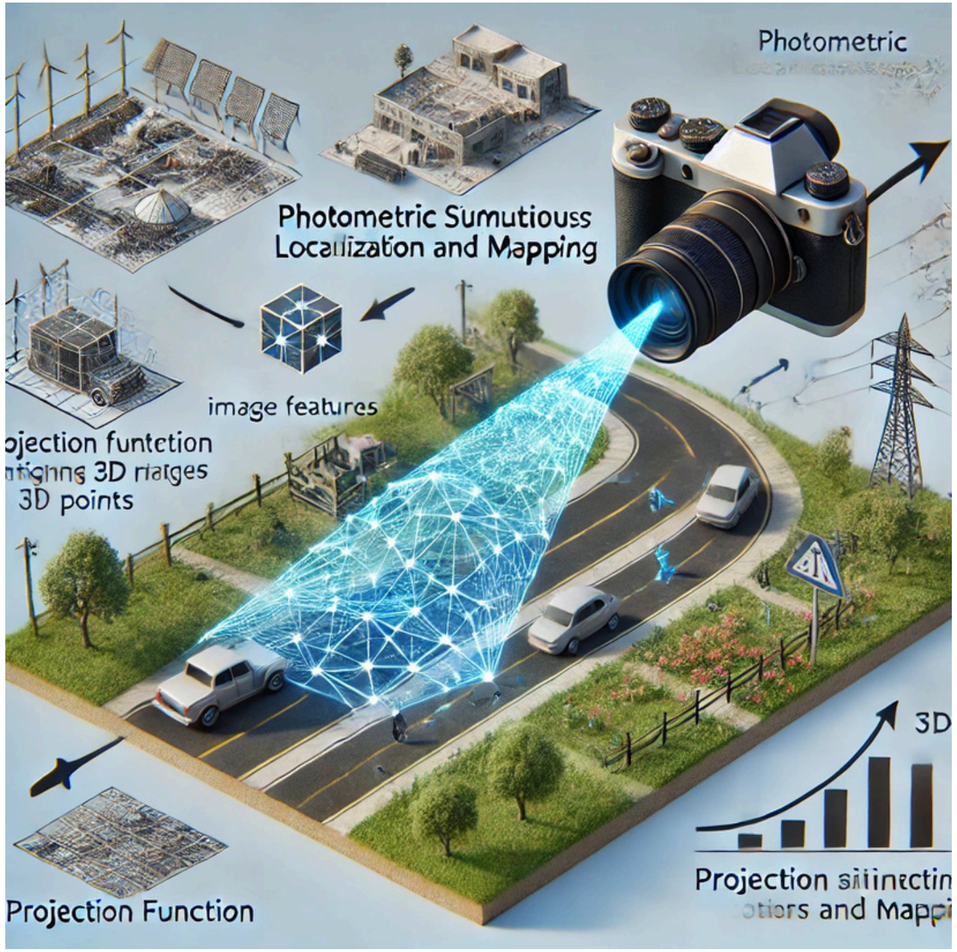
By Akshitha Poreddy
Sri Vaagdevi Bangari
Sowmya Koneti

WHY AND HOW IS IT DIFFERENT ?



INTRODUCTION

- Photo-SLAM (Photometric Simultaneous Localization and Mapping) is a vision-based technique for estimating camera motion and creating 3D maps.
- It minimizes photometric error instead of relying on keypoint detection like traditional SLAM.
- Hybrid localization combines explicit geometry-based tracking with implicit radiance field optimization



IMPLICIT VS EXPLICIT

🔷 Explicit Methods

(e.g., KinectFusion, BundleFusion)

- Use **structured data** like depth maps or point clouds
- Build **dense 3D surfaces** in real time
- But: Images are often **not visually realistic**

🔶 Implicit Methods

(e.g., NeRF, iMAP, Nice-SLAM)

- Use **neural networks (MLPs)** to learn the scene's shape and look
- Generate **photorealistic images** from new viewpoints
- But:
  - Are often **slow and computationally heavy**
  - Require **depth input or extra models** for good results

- Photoslam uses explicit geometric features (for fast and accurate tracking) and implicit textures (for realistic appearance)

FEATURES OF PHOTO-SLAM

✅ Works in featureless and low-texture environments
✅ Uses pixel intensity for accurate localization and mapping
✅ Robust to geometric distortions and lighting changes

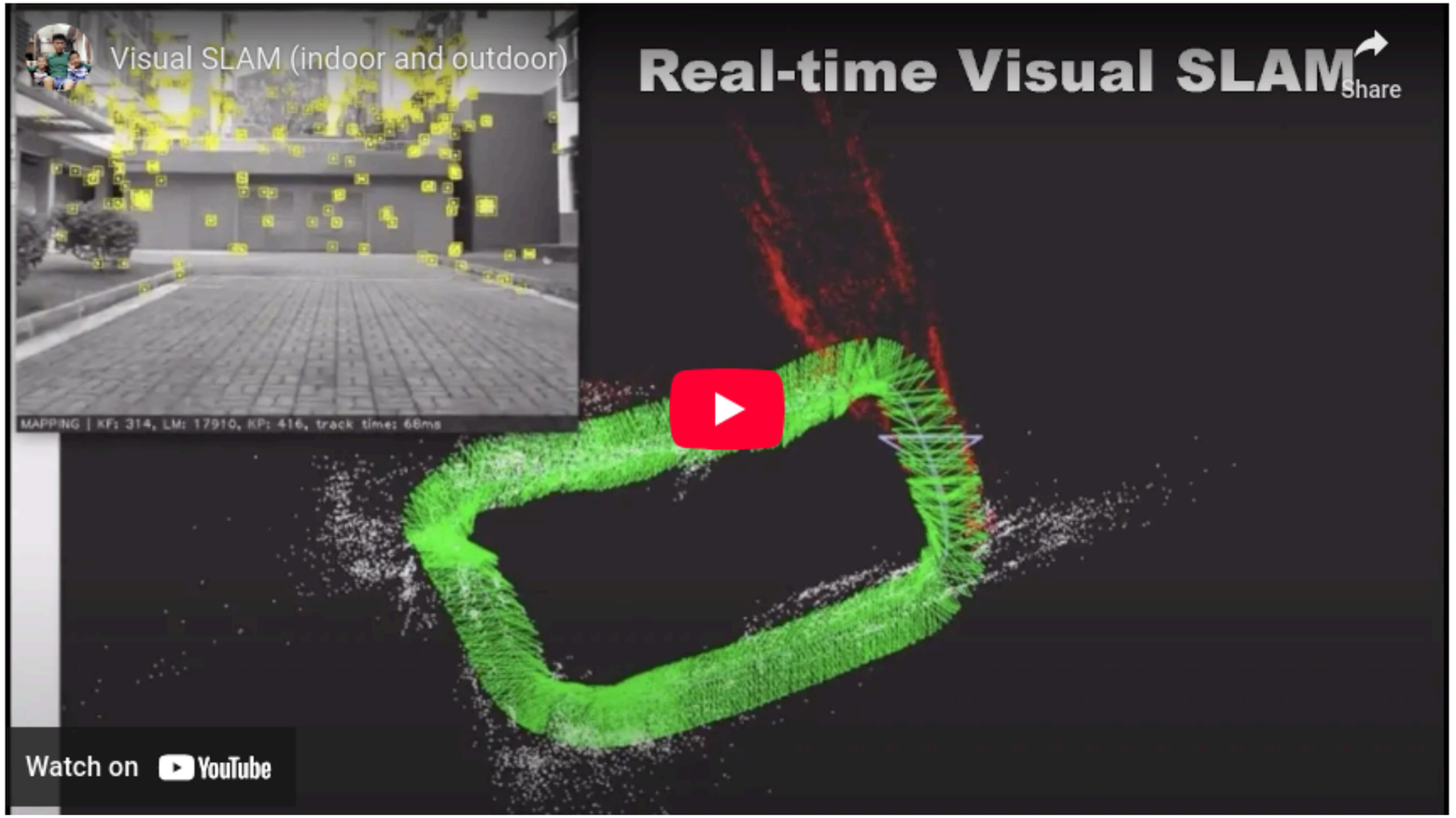
KEY INNOVATIONS

- 3D Gaussian Splatting for faster rendering
- Gaussian Pyramid-based learning: teaches the system to learn from low to high detail
- No need for dense depth maps (works even with just a regular camera!)

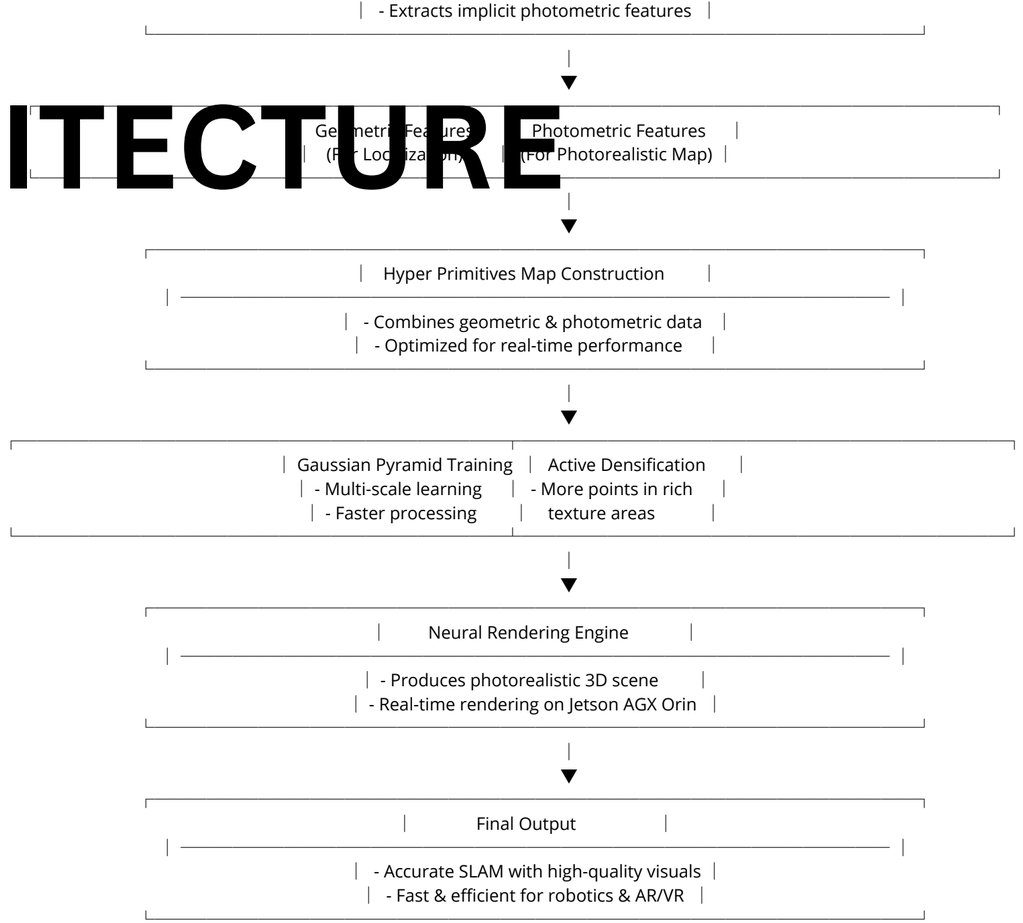
Practical Use & Applications

Robotics: robots navigating in real-world environments.
Drones: drones capturing photorealistic maps for inspection or surveillance.
AR/VR: devices generating realistic environments for augmented reality experiences.
Autonomous vehicles: helping vehicles understand and visualize their surroundings efficiently.

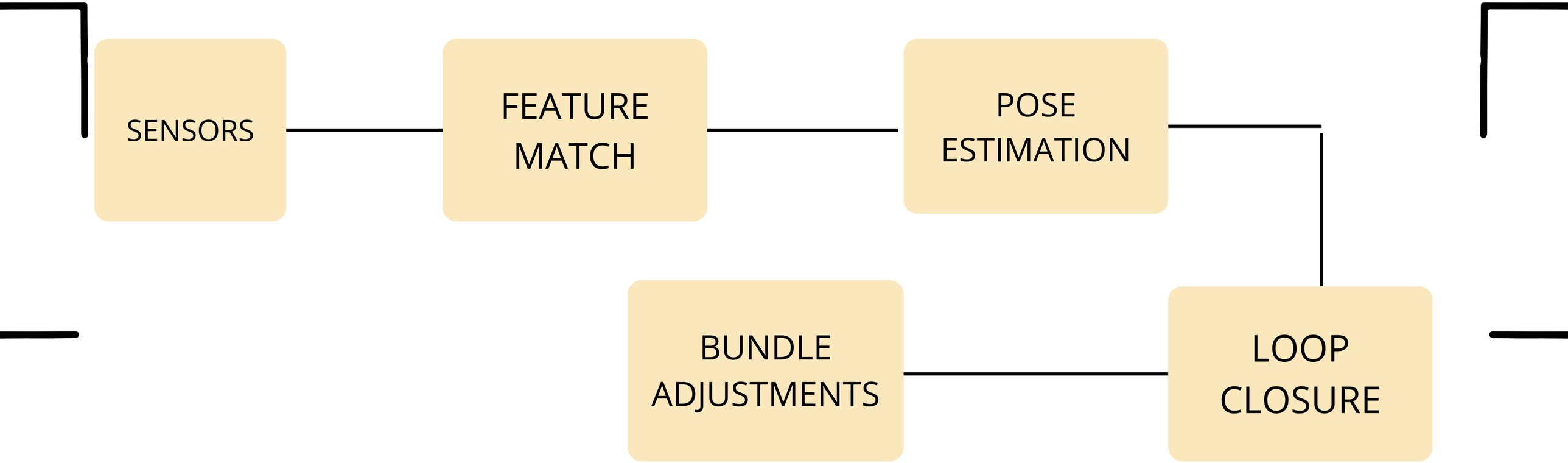
DEMO


Visual SLAM (indoor and outdoor)
MAPPING | KF: 314, LM: 17910, KP: 416, track time: 68ms
Real-time Visual SLAM
Share
Watch on YouTube

ARCHITECTURE



METHODOLOGY



LOOP CLOSURE AND POSE OPTIMIZATION

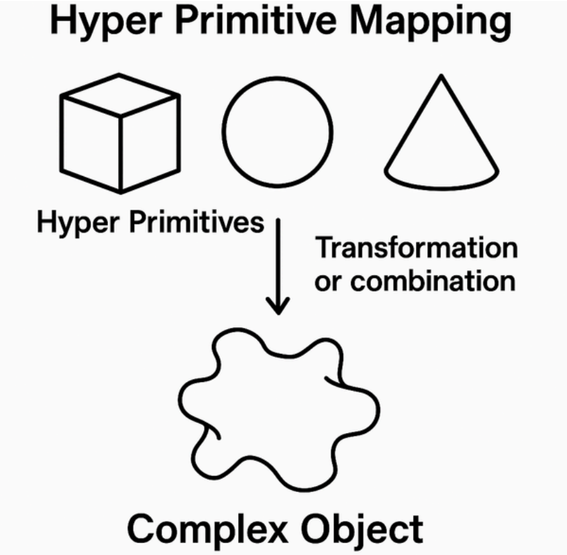
- Over time, small pose estimation errors accumulate (drift).
- The system performs loop closure by identifying when the camera returns to a previously mapped area using feature similarity.
- Once a loop is detected, it runs pose graph optimization to align past and current poses and reduce accumulated errors.
- This process ensures global consistency of the 3D map.

CONTRIBUTIONS

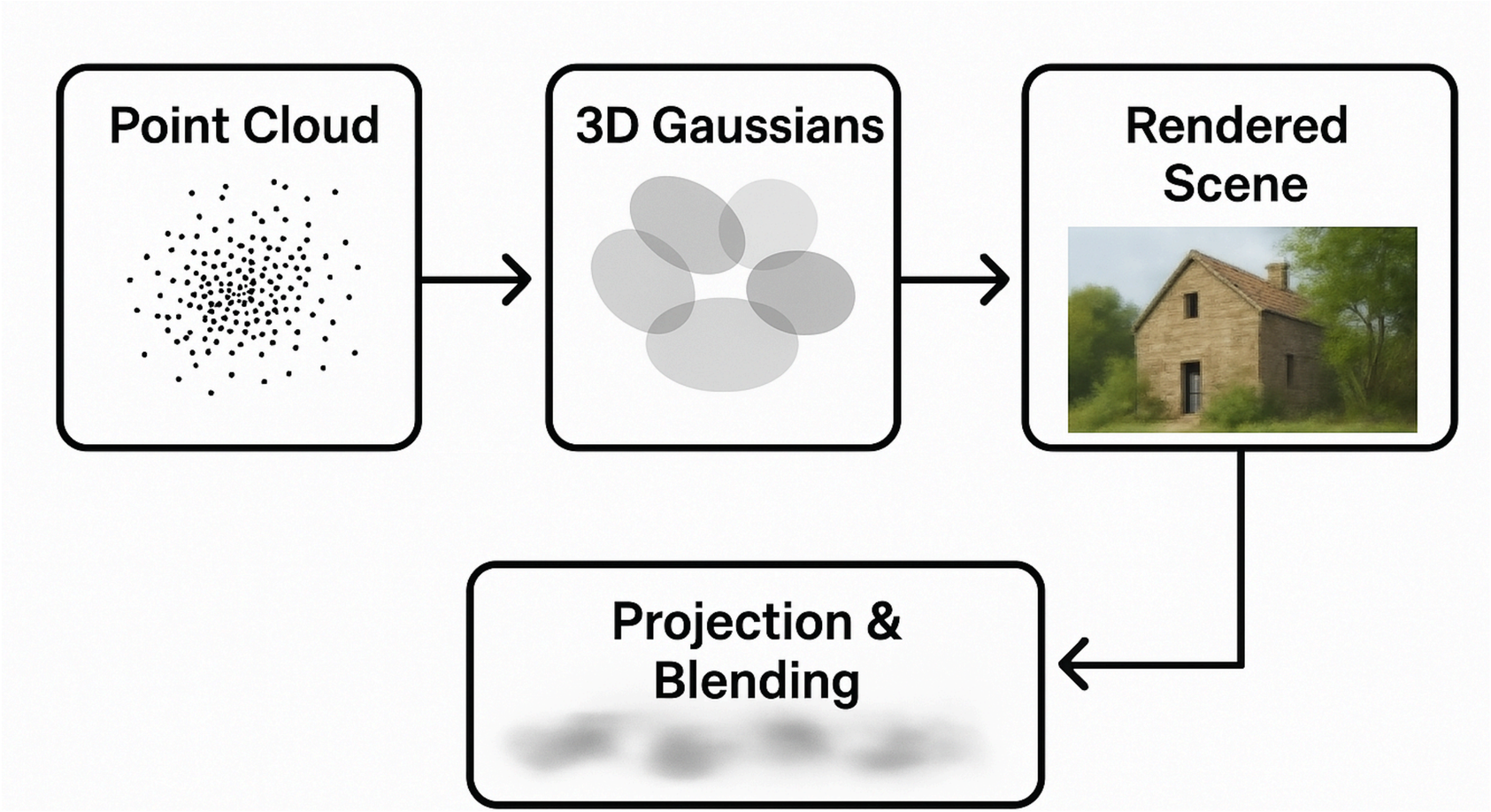
- **Novel SLAM System (Photo-SLAM):** Introduces Photo-SLAM as the first system to combine hyper primitives for geometry and appearance with real-time performance.
- **Hyper Primitives Mapping:** A point-based representation that encodes position, scale, rotation, color, and lighting for efficient mapping and rendering.
- **Gaussian-Pyramid Learning:** A multi-scale training strategy that guides the model using low- to high-resolution image supervision for better optimization.
- **Cross-Camera Generalization:** Enables seamless operation across monocular, stereo, and RGB-D camera inputs by adapting its mapping and learning methods

HYPER PRIMITIVES MAP

- Each map element is a 3D Gaussian primitive containing:
  Position: 3D location in the world
  Scale and Rotation: Defines the orientation and shape of Gaussian blob
  Color and Density: Captures appearance and visibility
  Spherical Harmonics: Encodes lighting behavior
- These primitives allow the system to render realistic views and track accurately by jointly optimizing geometry and texture.
- Acts as a shared representation for both localization and photorealistic mapping.



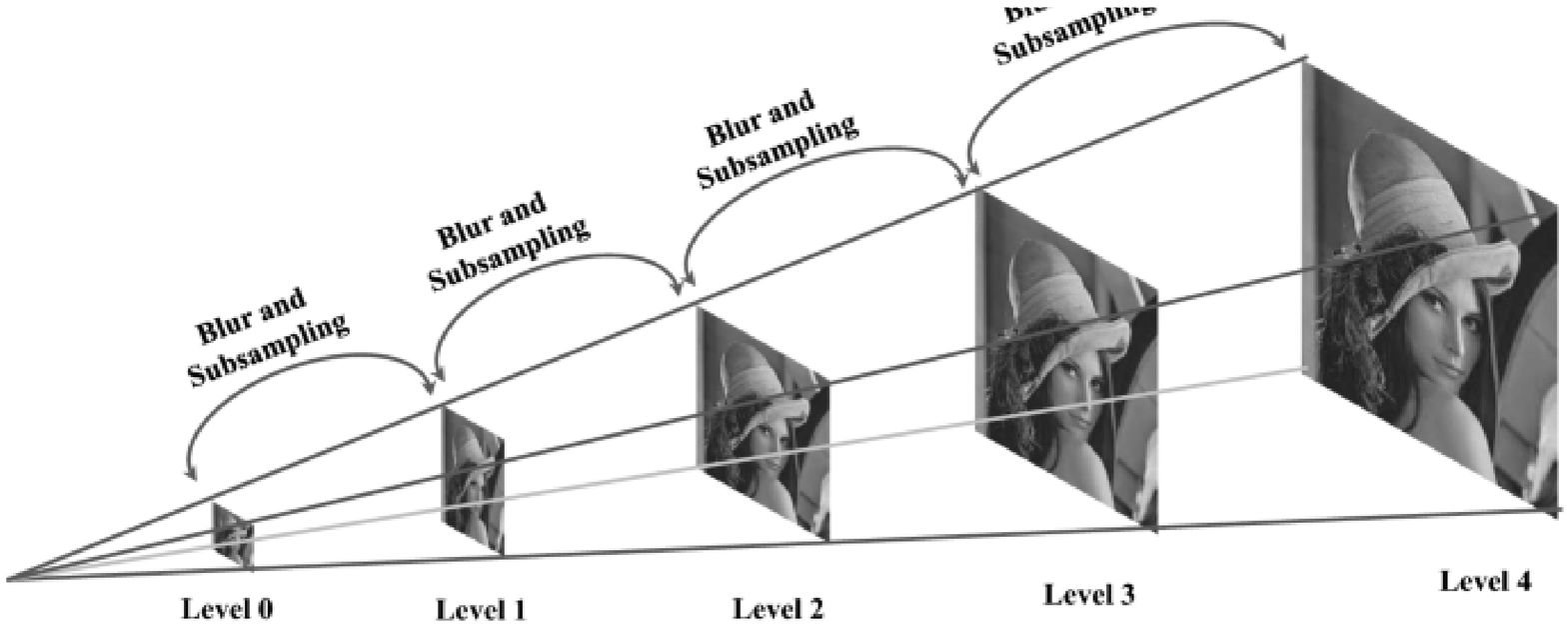
3D GAUSSIAN SPLATTING



GAUSSIAN-PYRAMID LEARNING

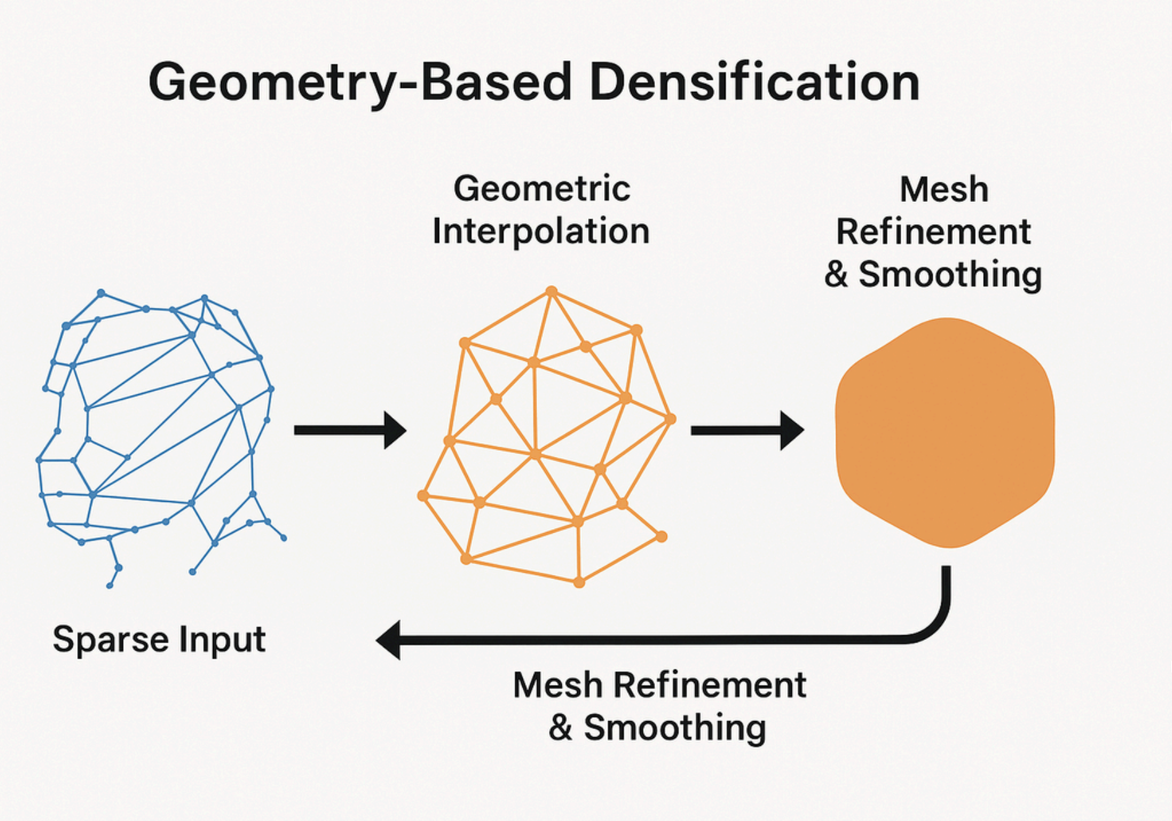
A progressive training strategy that begins with low-resolution images and gradually moves to higher-resolution versions.

- Stabilize training and improve convergence speed.
- Avoid overfitting to fine details too early
- Loss functions (e.g., photometric loss, perceptual loss) are applied at multiple scales to supervise rendering quality.



GEOMETRY-BASED DENSIFICATION

- In monocular or stereo setups, some regions of the map may be under-sampled.
- The system automatically detects low-density or poorly reconstructed regions and inserts new Gaussian primitives based on interpolated depth and image features.
- This ensures the map has sufficient resolution to support both accurate tracking and high-quality rendering.



SETUP

- **Datasets:**
  - **Replica:** Photorealistic synthetic indoor dataset
  - **TUM RGB-D:** Real-world indoor dataset with RGB and depth
  - **EuRoC MAV:** Real-world stereo dataset captured by drones
- **Evaluation Metrics:**
  - **Localization:** RMSE(Root Mean Squared Error), Absolute Trajectory Error (ATE)
  - **Rendering Quality:** PSNR(Peak Signal-to-Noise Ratio), SSIM(Structural Similarity Index), LPIPS(Learned Perceptual Image Patch Similarity)
  - **Efficiency:** Tracking/rendering FPS, memory usage

QUANTITATIVE RESULTS

Localization Performance:

- Better results than ORB-SLAM3 and DROID-SLAM in low texture and featureless environments.

Rendering Quality:

- Up to 30% higher PSNR than NeRF-based systems
- Much faster rendering (900–1000 FPS on desktop, 100+ FPS on Jetson)

Efficiency:

- GPU memory usage lower than other neural SLAM systems
- Scales well across different hardware

RESULTS COMPARISON (DATASETS)

On Replica Dataset		Localization (cm)		Mapping			Resources			
Cam	Method	RMSE ↓	STD ↓	PSNR ↑	SSIM ↑	LPIPS ↓	Operation Time ↓	Tracking FPS ↑	Rendering FPS ↑	GPU Memory Usage ↓
Mono	ORB-SLAM3 [2]	3.942	3.115	-	-	-	<1 mins	**58.749**	-	0
	DROID-SLAM [34]	0.725	**0.308**	-	-	-	<2 mins	35.473	-	11 GB
	Nice-SLAM* [46]	99.9415	35.336	16.311	0.720	0.439	>10 mins	2.384	0.944	12 GB
	Orbeez-SLAM [4]	-	-	23.246	0.790	0.336	<5 mins	49.200	1.030	6 GB
	Go-SLAM [44]	71.054	24.593	21.172	0.703	0.421	<5 mins	25.366	0.821	22 GB
	Ours (Jetson)	1.235	0.756	29.284	0.883	0.139	<5 mins	18.315	95.057	**4 GB**
	Ours (Laptop)	**0.713**	0.524	33.049	**0.926**	0.086	<5 mins	19.974	353.504	**4 GB**
	Ours	1.091	0.892	**33.302**	**0.926**	**0.078**	<2 mins	41.646	**911.262**	6 GB
RGB-D	ORB-SLAM3 [2]	1.833	1.478	-	-	-	<1 mins	**52.209**	-	0
	DROID-SLAM [34]	0.634	0.248	-	-	-	<2 mins	36.452	-	11 GB
	BundleFusion [6]	1.606	0.969	23.839	0.822	0.197	<5 mins	8.630	-	5 GB
	Nice-SLAM [46]	2.350	1.590	26.158	0.832	0.232	>10 mins	2.331	0.611	12 GB
	Orbeez-SLAM [4]	0.888	0.562	32.516	0.916	0.112	<5 mins	41.333	1.401	6 GB
	ESLAM [16]	**0.568**	0.274	30.594	0.866	0.162	<5 mins	6.687	2.626	21 GB
	Co-SLAM [36]	1.158	0.602	30.246	0.864	0.175	<5 mins	14.575	3.745	**4 GB**
	Go-SLAM [44]	0.571	**0.218**	24.158	0.766	0.352	<5 mins	19.437	0.444	24 GB
	Point-SLAM [27]	0.596	0.249	34.632	0.927	0.083	>2 hrs	0.345	0.510	24 GB
	Ours (Jetson)	0.581	0.289	31.978	0.916	0.101	<5 mins	17.926	116.395	**4 GB**
	Ours (Laptop)	0.590	0.289	34.853	**0.944**	0.062	<5 mins	20.597	396.082	**4 GB**
	Ours	0.604	0.298	**34.958**	0.942	**0.059**	<2 mins	42.485	**1084.017**	5 GB

Quantitative results on the Replica dataset. We mark the best two results with **first** and second . Nice-SLAM* means

On TUM Dataset		fr1-desk				fr2-xyz				fr3-office			
Cam	Method	RMSE (cm) ↓	PSNR ↑	SSIM ↑	LPIPS ↓	RMSE (cm) ↓	PSNR ↑	SSIM ↑	LPIPS ↓	RMSE (cm) ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Mono	ORB-SLAM3 [2]	**1.534**	-	-	-	0.720	-	-	-	1.400	-	-	-
	DROID-SLAM [34]	78.245	-	-	-	36.050	-	-	-	154.383	-	-	-
	Go-SLAM [44]	33.122	11.705	0.406	0.614	28.584	14.807	0.443	0.572	105.755	13.572	0.480	0.643
	Ours (Jetson)	1.757	18.811	0.681	0.329	**0.558**	21.347	0.727	0.187	1.687	18.884	0.672	0.289
	Ours (Laptop)	1.549	20.515	0.733	0.241	0.852	**21.575**	**0.739**	**0.157**	1.542	19.138	0.680	0.259
	Ours	1.539	**20.972**	**0.743**	**0.228**	0.984	21.072	0.726	0.166	**1.257**	**19.591**	**0.692**	**0.239**
RGB-D	ORB-SLAM3 [2]	**1.724**	-	-	-	0.385	-	-	-	1.698	-	-	-
	DROID-SLAM [34]	91.985	-	-	-	41.833	-	-	-	160.141	-	-	-
	Nice-SLAM [46]	19.317	12.003	0.417	0.510	36.103	18.200	0.603	0.313	25.309	16.341	0.548	0.386
	ESLAM [16]	3.359	17.497	0.561	0.484	31.448	22.225	0.727	0.233	25.808	19.113	0.616	0.359
	Co-SLAM [36]	3.094	16.419	0.482	0.591	31.347	19.176	0.595	0.374	25.374	17.863	0.547	0.452
	Go-SLAM [44]	2.119	15.794	0.531	0.538	31.788	16.118	0.534	0.419	26.802	16.499	0.566	0.569
	Ours (Jetson)	4.571	18.273	0.663	0.338	0.360	**23.127**	**0.780**	**0.149**	1.874	19.781	0.701	0.235
	Ours (Laptop)	1.891	20.403	0.728	0.251	0.361	22.570	0.777	0.158	1.315	21.569	0.749	0.184
	Ours	2.603	**20.870**	**0.743**	**0.239**	**0.346**	22.094	0.765	0.169	**1.001**	**22.744**	**0.780**	**0.154**

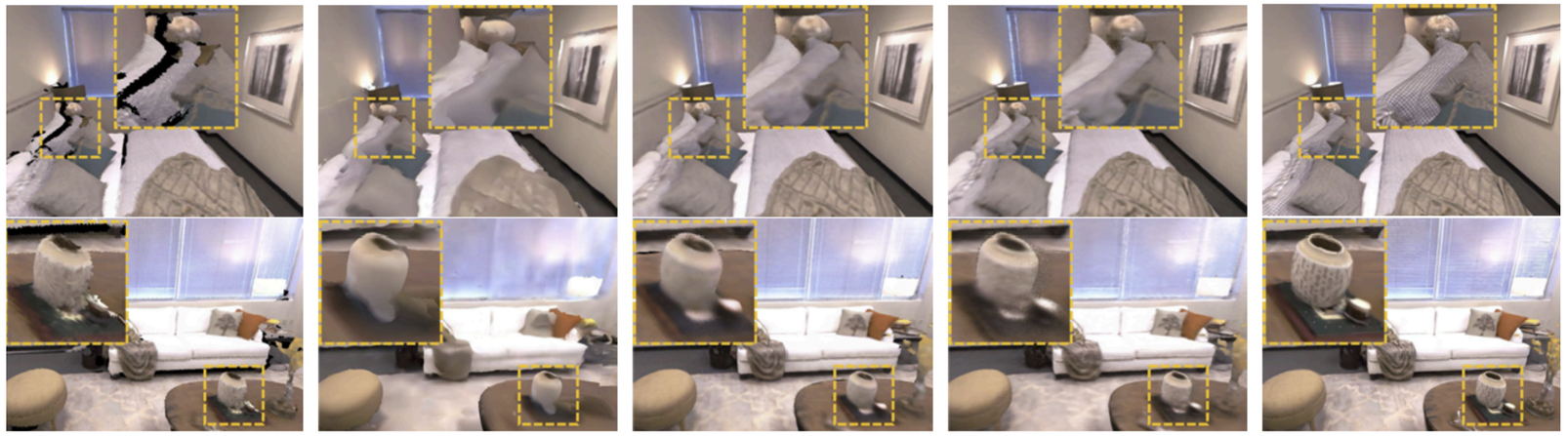
ıble 2. Quantitative results on the TUM RGB-D dataset. We mark the best two results with **first** and second

On Euroc Stereo		ORB-SLAM3	DROID-SLAM	**Ours (Jetson)**	**Ours (Laptop)**	**Ours**
MH-01	RMSE (cm) ↓	4.379	39.514	4.207	**4.049**	4.109
	PSNR ↑	-	-	**13.979**	13.962	13.952
	SSIM ↑	-	-	**0.426**	0.421	0.420
	LPIPS ↓	-	-	0.428	0.378	**0.366**
MH-02	RMSE (cm) ↓	4.525	39.265	**4.193**	4.731	4.441
	PSNR ↑	-	-	14.210	**14.254**	14.201
	SSIM ↑	-	-	**0.436**	**0.436**	0.430
	LPIPS ↓	-	-	0.447	0.373	**0.356**
V1-01	RMSE (cm) ↓	8.940	21.646	8.830	8.836	**8.821**
	PSNR ↑	-	-	16.933	17.025	**17.069**
	SSIM ↑	-	-	**0.626**	0.622	0.618
	LPIPS ↓	-	-	0.321	0.284	**0.266**
V2-01	RMSE (cm) ↓	26.904	**15.344**	26.643	26.736	26.609
	PSNR ↑	-	-	16.038	**16.052**	15.677
	SSIM ↑	-	-	**0.643**	0.635	0.622
	LPIPS ↓	-	-	0.347	**0.314**	0.323

ıle 3. Quantitative results on the EuRoC MAV dataset, using

QUALITATIVE RESULTS

- Sharper, more detailed reconstructions than ORB-SLAM and Nice-SLAM
- More consistent color and lighting under varying views
- Better handling of occlusions and fine structures
- Works well in cluttered indoor environments with complex lighting



(a) BundleFusion [6] (b) Nice-SLAM [46] (c) ESLAM [16] (d) Co-SLAM [36] (e) Ours

COMPARISION

CONCLUSION

- Photo-SLAM is a novel real-time SLAM system that bridges the gap between classical geometric SLAM and modern photorealistic rendering.
- Key innovations include the hyper primitives map, 3D Gaussian splatting, and multi-scale learning.
- Demonstrated strong performance in both tracking accuracy and visual fidelity.


THANK YOU
SO MUCH!!