Optimizing Drone Trajectory Planning for Capturing Large-Scale 3D Gaussian Splatting Scenes via the MAVLink Protocol

Cheng-Yuan Wu
(chengyuanwu83@gmail.com)

Advisor: **Cheng-Hsin Hsu**

AIINS@NTHU
Ambient Intelligence for
Immersive Networked Systems Lab

國立清華大學
NATIONAL TSING HUA UNIVERSITY

Different Interaction Modes with Videos

DoF = Degrees of Freedom

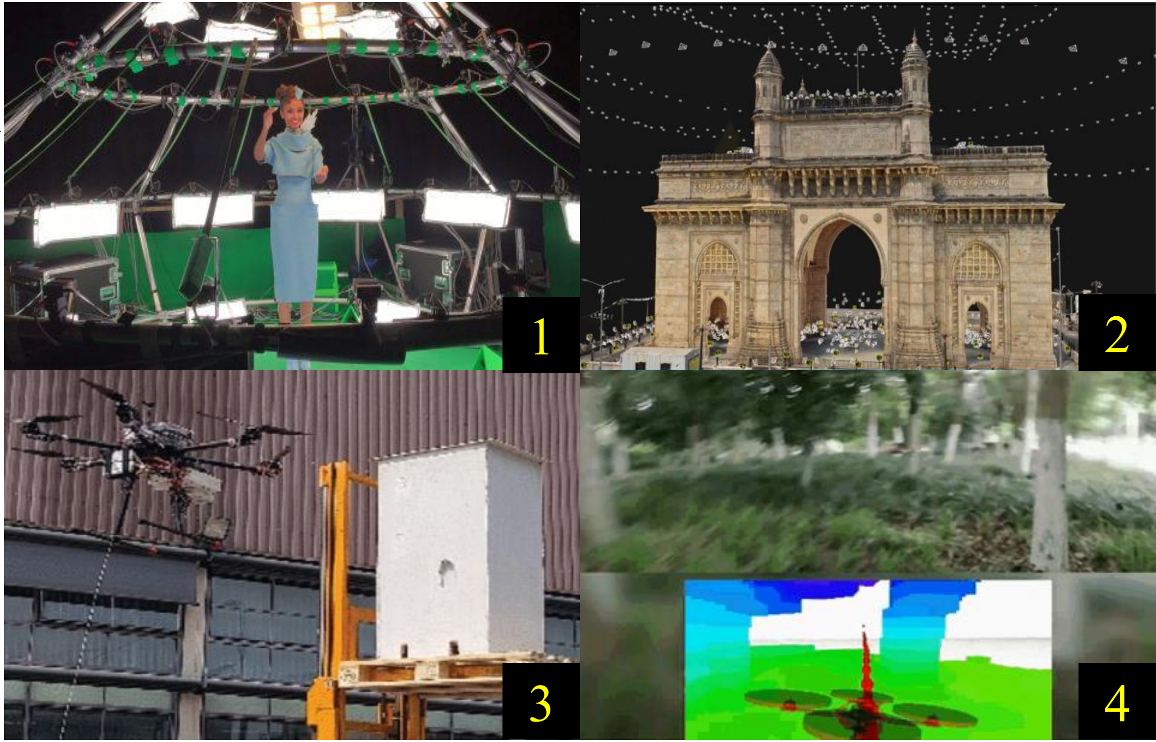
2D Videos
0-DoF

360-degree Videos
3-DoF

Volumetric Video
6-DoF

6DoF (Degrees of Freedom) Applications

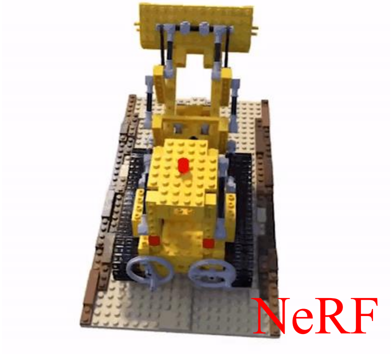
- Film production
- Heritage preservation
- Defect detection
- Robotic navigation



1. https://www.capturingreality.com/RealityCapture-In-Ghost-In-The-Shell
2. https://www.capturingreality.com/cultural-heritage
3. Marchisotti, Daniele, and Emanuele Zappa. "Feasibility study of drone-based 3-D measurement of defects in concrete structures." IEEE Transactions on Instrumentation and Measurement 71 (2022): 1-11.
4. Zhou, Xin, et al. "Ego-planner: An esdf-free gradient-based local planner for quadrotors." IEEE Robotics and Automation Letters 6.2 (2020): 478-485.

Enabling Representations of 6DoF Applications

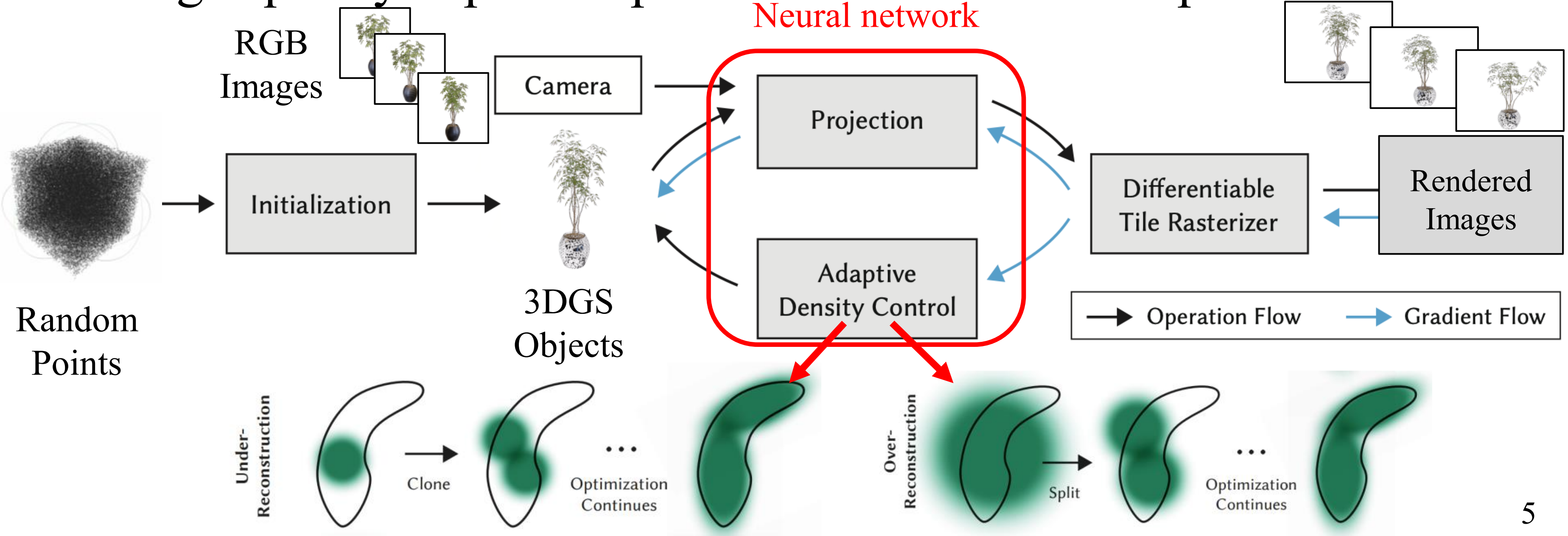
- 3D meshes and point clouds are more suitable to objects generated by computer graphics ← not good for real-life objects
- Compared to DIBR (Depth-Image-Based-Rendering), NeRF (Neural Radiance Fields) [ECCV'20] and 3DGS (3D Gaussian Splatting) [SIGGRAPH'23] produce better synthesized views
- NeRF rendering, however, is much slower than 3DGS rendering



3D Gaussian Splatting (3DGS) [1]

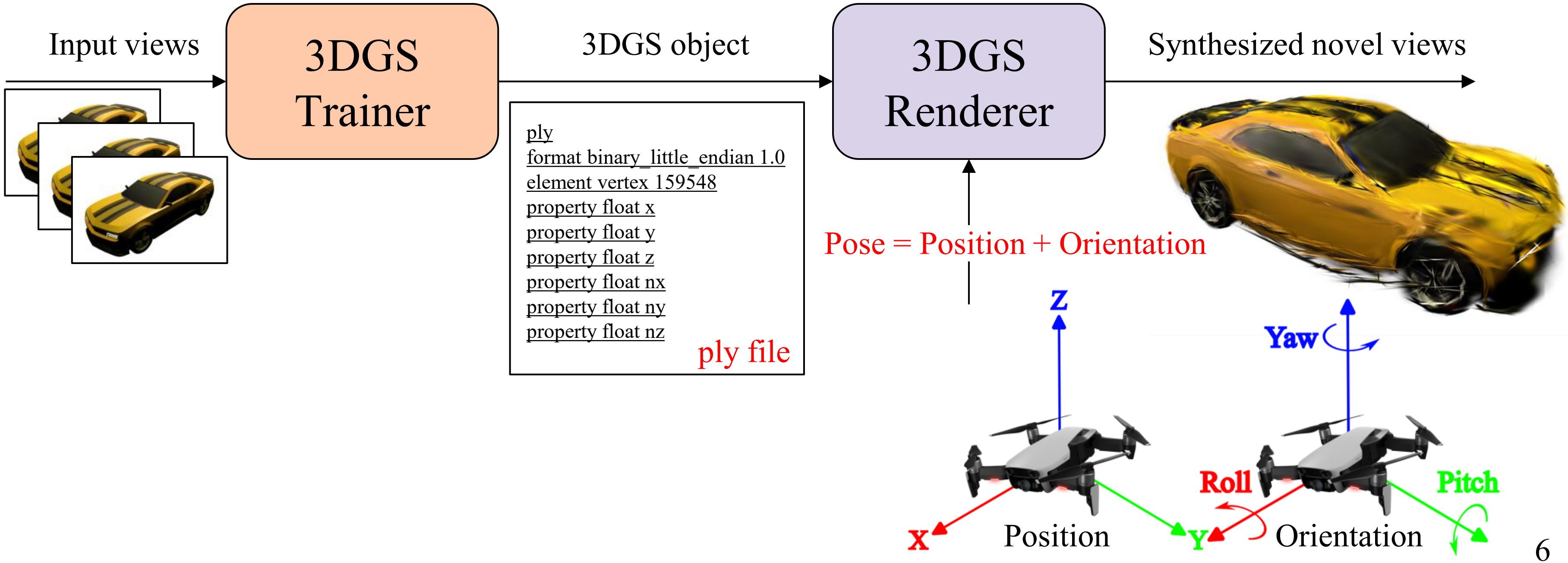
- Simple learning-based generation pipeline
- Real-time rendering
- High-quality explicit representation ⟵ enriched point clouds

Neural network



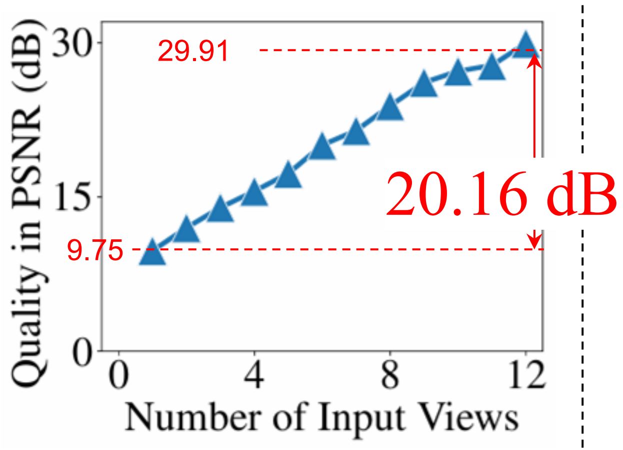
(Novel) View Synthesis Using 3DGS

- Input views: Ground truth images captured from real scenes
- Synthesized views: New perspectives generated by 3DGS objects

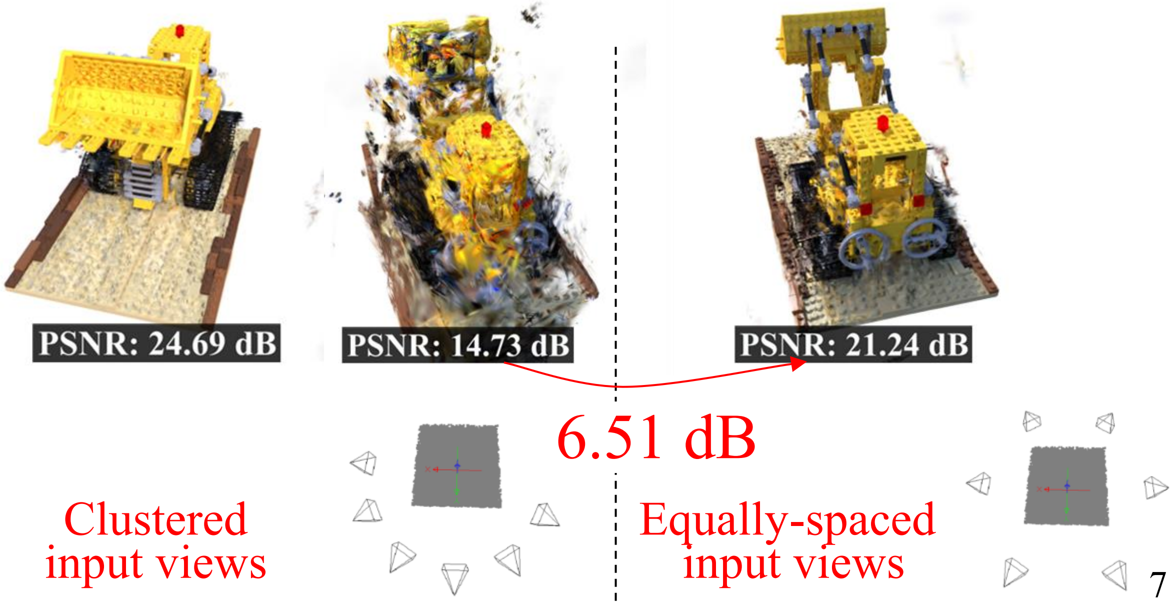


Selection of Input Views is Crucial: Pilot Tests

- More input views improve synthesized view quality
- With the same number of input views equally-spaced setup boost the quality



Increasingly more input views

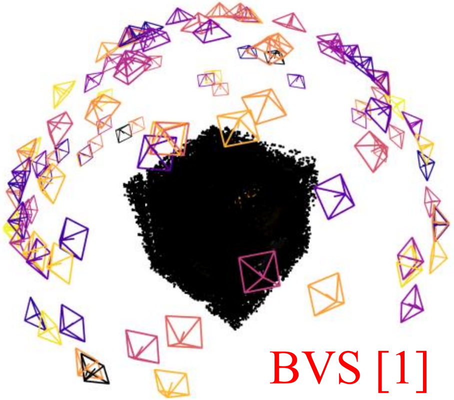


Clustered input views

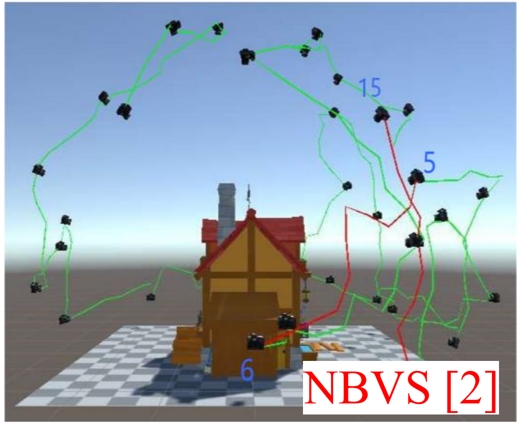
Equally-spaced input views

View Selection Problems

- Two problem variants
  - Best View Selection (BVS): selects a subset of images from already captured views
  - Next Best View Selection (NBVS): select the next few poses on-the-fly for additional input views



BVS [1]



NBVS [2]

1. Jiang, Wen, Boshu Lei, and Kostas Daniilidis. "Fisherrf: Active view selection and mapping with radiance fields using fisher information." European Conference on Computer Vision. Cham: Springer Nature Switzerland, 2024.
2. Ran, Yunlong, et al. "Neurar: Neural uncertainty for autonomous 3D reconstruction with implicit neural representations." IEEE Robotics and Automation Letters 8.2 (2023): 1125-1132.

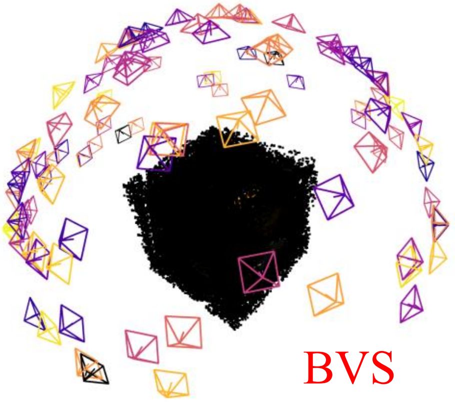
Outlines

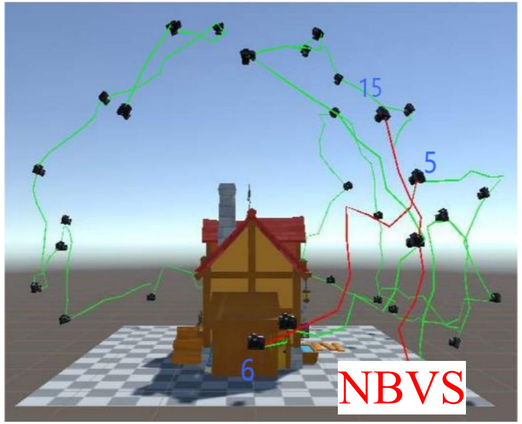
- Motivation
- Related Work
- System Architecture
- Research Problem & Algorithms
- Experiments
- Conclusion & Future Work

Related Work

- Best View Selection [1]: Not suitable for online scenario
- Next Best View Selection [2, 3, 4]: Built upon a memory-hungry 3D occupant map and focus on unobserved areas
- No consideration of both network conditions and actual protocols like MAVLink

1. Jiang, Wen, Boshu Lei, and Kostas Daniilidis. "Fisherrf: Active view selection and mapping with radiance fields using fisher information." European Conference on Computer Vision. Cham: Springer Nature Switzerland, 2024.
2. Jin, Rui, et al. "Gs-planner: A gaussian-splatting-based planning framework for active high-fidelity reconstruction." 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2024.
3. Jin, Liren, et al. "Activegs: Active scene reconstruction using gaussi splatting." IEEE Robotics and Automation Letters (2025).
4. Zeng, Jing, et al. "Multi-robot autonomous 3D reconstruction using Gaussian splatting with Semantic guidance." IEEE Robotics and Automation Letters (2025).



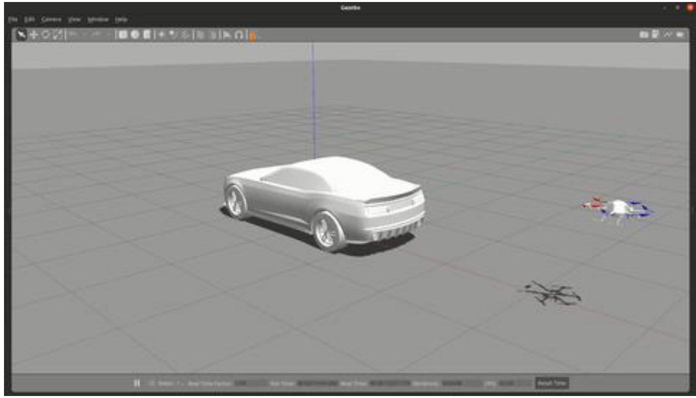


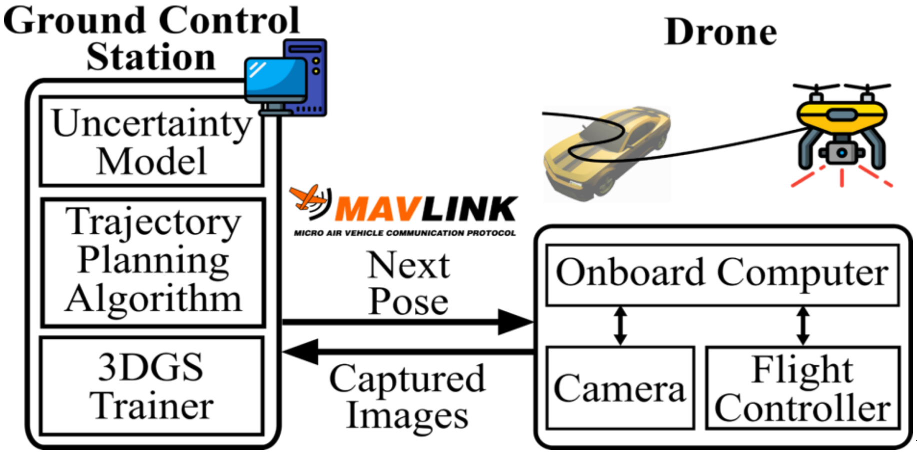
Outlines

- Motivation
- Related Work
- System Architecture
- Research Problem & Algorithms
- Experiments
- Conclusion & Future Work

Considered NBVS Scenario

- Drone: Capturing images as candidate input views
- Ground Control Station (GCS):
  - Planning drone trajectory on-the-fly
  - Training 3DGS objects from input views
- MAVLink: Enabling communication between drone and GCS

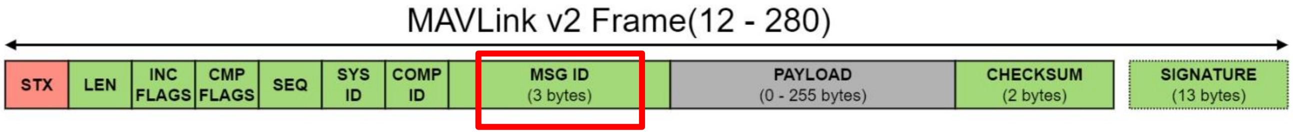




MAVLink (Micro Air Vehicle Link)

- Lightweight messaging protocol [1, 2] for communicating with drone and GCS
- Decode the payload through a predefined message

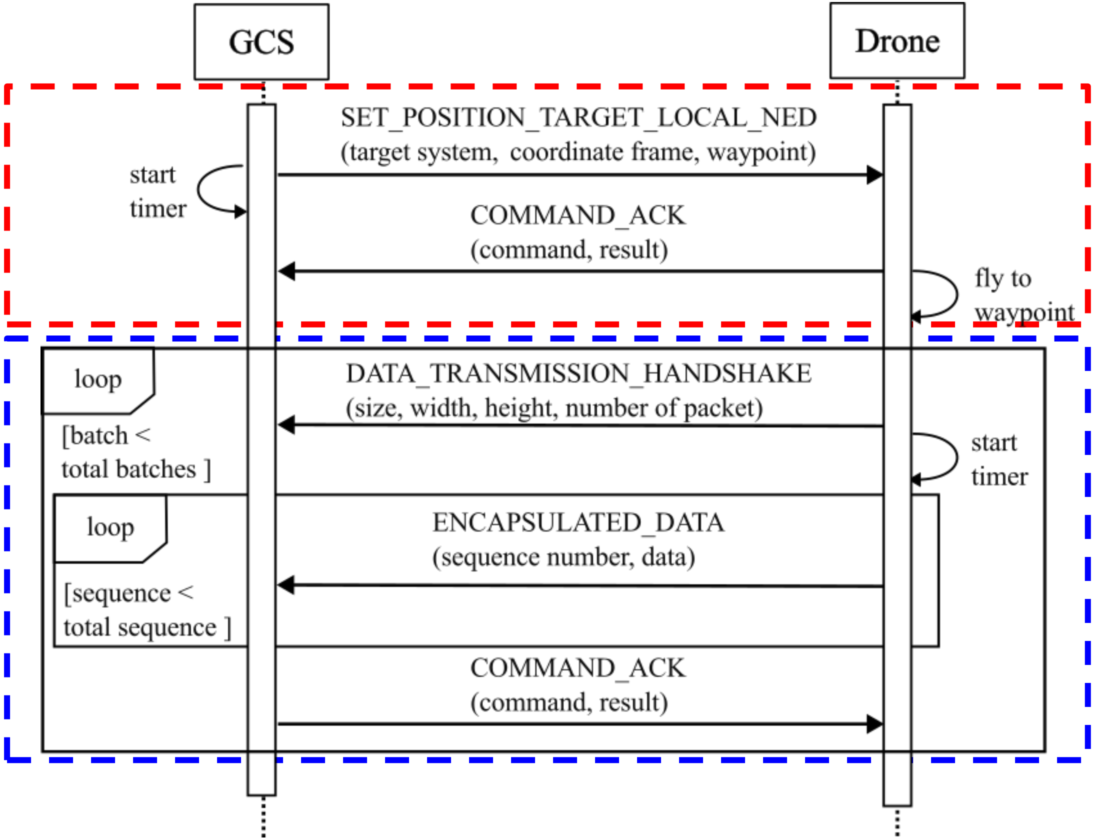
Application	MAVLink
Transport	UDP
Network	IP
Data Link	802.11 MAC
Physical	802.11g



Indicating the message type

1. https://mavlink.io/en/
2. Koubâa, Anis, et al. "Micro air vehicle link (mavlink) in a nutshell: A survey." IEEE Access 7 (2019): 87658-87680.

Capture image through MAVLink

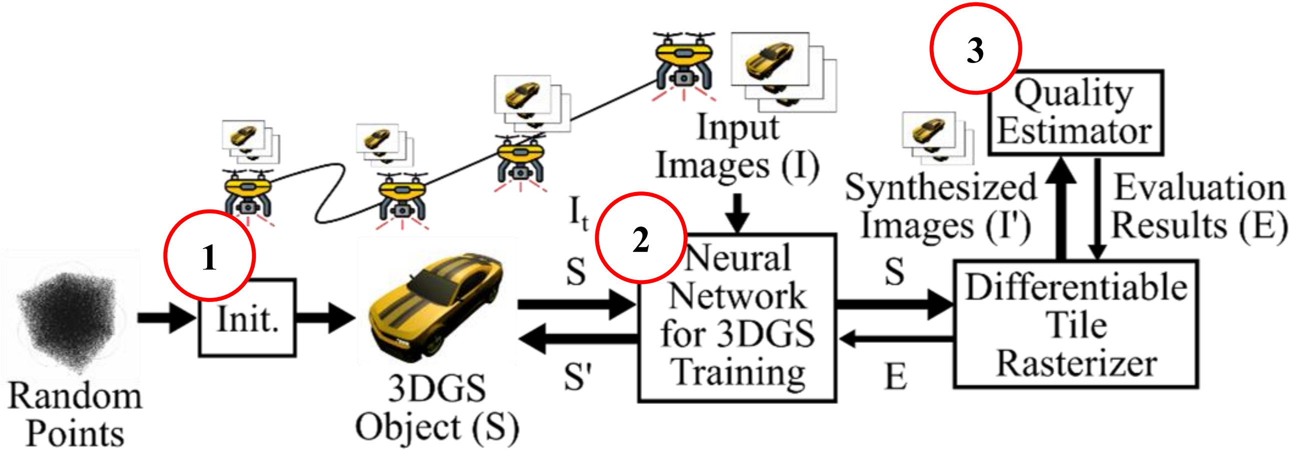


Transmit desired waypoint

Transmit captured image

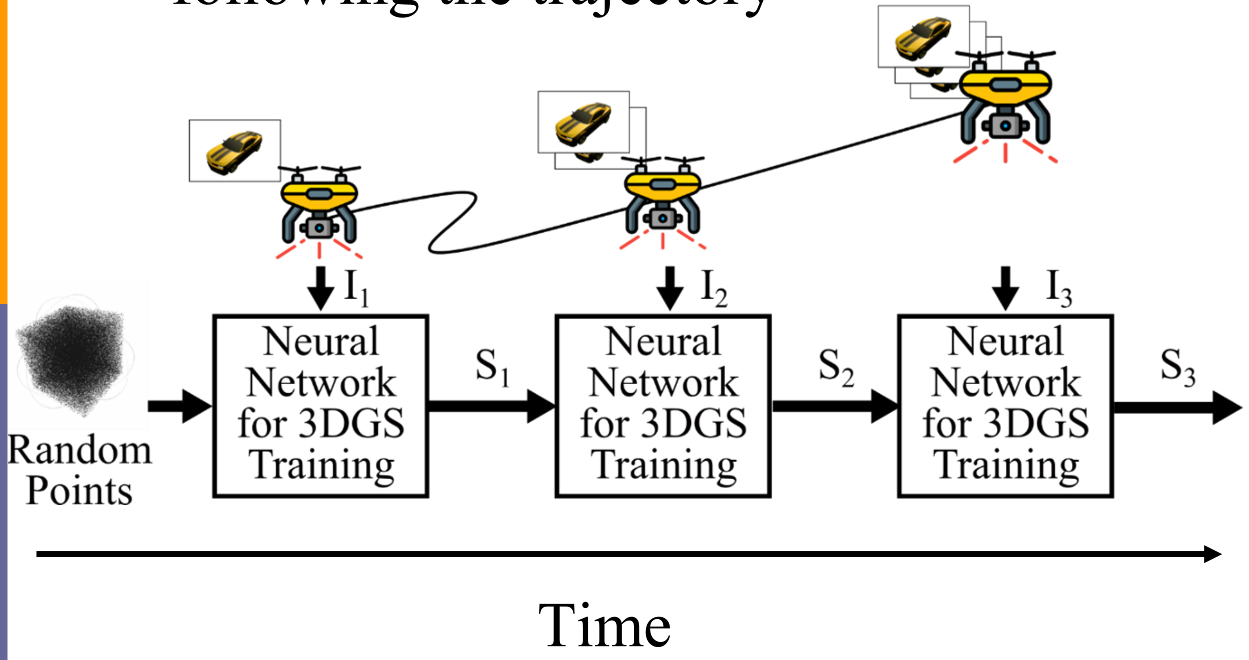
3DGS Training Process

- Initialize the 3DGS object with a random point cloud 1
- Go through multiple epochs, where in each epoch
  - Employ a neural network that take 3DGS object (S) and set of input images (I) as input 2
  - Evaluate synthesized view (I') quality to optimize 3DGS object 3



Incrementally Generated 3DGS Object

- 3DGS is incrementally constructed for increasingly better quality when each drone gradually captures input views following the trajectory



Time t_1: 13.69 s
Epoch e_1: 1412
PSNR q_1: 8.91 dB

Time t_2: 67.68 s
Epoch e_2: 8109
PSNR q_2: 12.64 dB

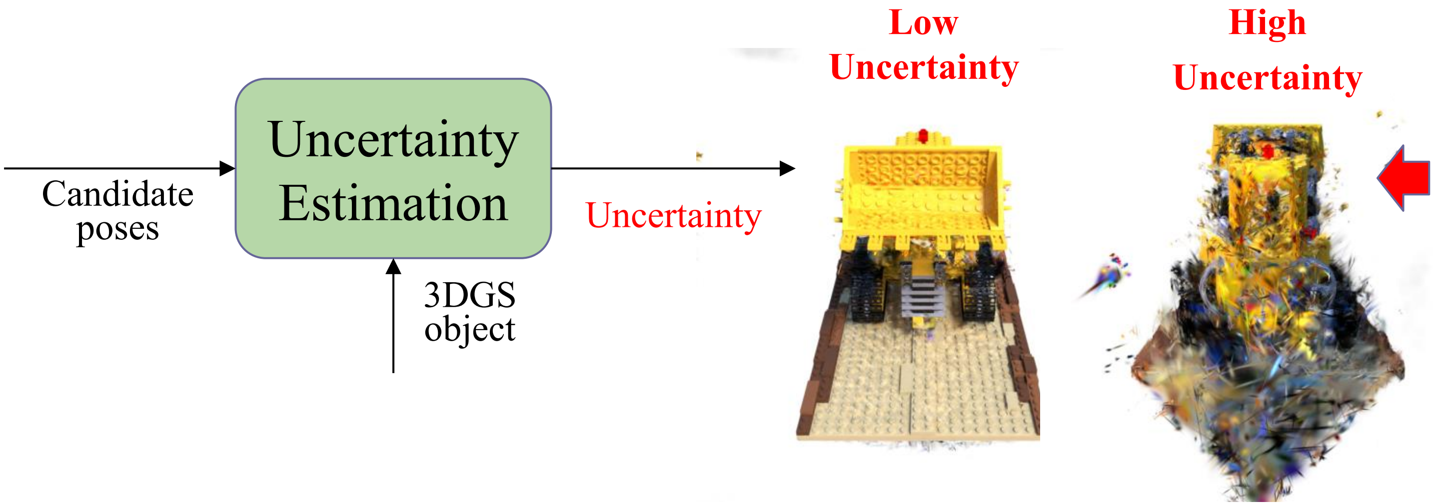
Time t_3: 350.0 s
Epoch e_3: 37040
PSNR q_3: 21.14 dB

Outlines

- Motivation
- Related Work
- System Architecture
- Research Problem & Algorithms
- Experiments
- Conclusion & Future Work

Challenge 1: Quantify Information

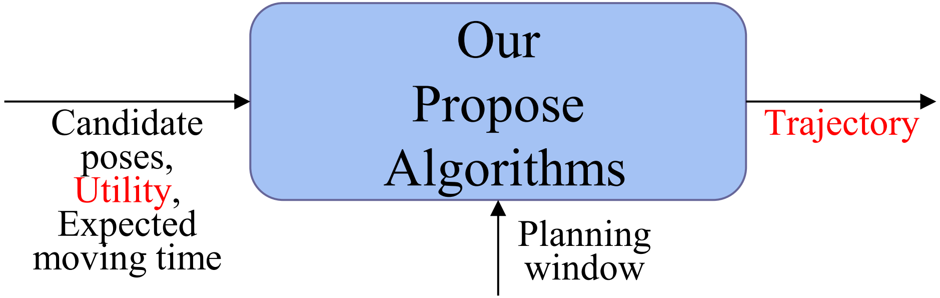
- It is not easy to quantify the information amount brought by each potential, or candidate pose to an existing 3DGS object
- Solution: Employ uncertainty to quantify the contributions of a candidate pose given 3DGS object



We should capture image of the pose with high uncertainty to improve the quality of synthesized view

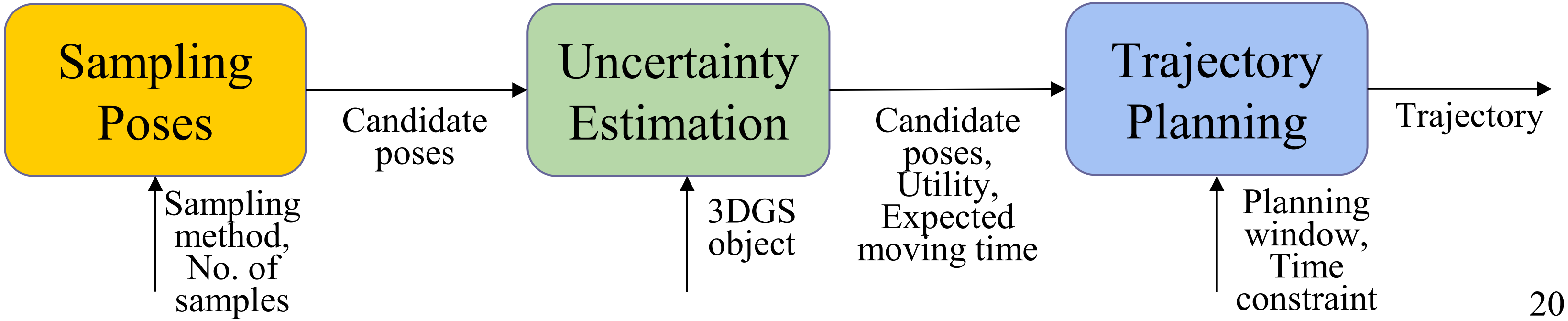
Challenge 2: Optimize Drone Trajectory

- It is non-trivial to systematically compute a drone trajectory to maximize the overall quality of final synthesized views
- Solution: formulate and solve an optimization problem to compute the drone trajectory
  - Utility: the potential contribution of each pose, which is quantified in Challenge 1
  - Trajectory: a sequence of poses
  - We propose two trajectory planning algorithms to solve the problem



NBVS Framework

- We propose a framework for producing a drone trajectory
- Sampling poses: Discretizing the huge search space to control the completing
- Uncertainty estimation: To evaluate potential contribution (utility) of current 3DGS
- Trajectory planning: To maximize the utility of the resulting trajectory



Challenge 1

Optimization Criteria

- Fisher information [1] indicates how much information is captured under a candidate pose for a given 3DGS object
- We use Fisher information to evaluate potential contribution of each pose

x: candidate pose
y: synthesized view of x
w: Gaussians' parameters

$$\mathcal{I}[\mathbf{w}^*; \{\mathbf{y}_i^{acq}\} | \{\mathbf{x}_i^{acq}\}, D^{train}]$$ Potential contribution of given pose

$$= \boxed{H[\mathbf{w}^* | D^{train}]} - \boxed{H[\mathbf{w}^* | \{\mathbf{y}_i^{acq}\}, \{\mathbf{x}_i^{acq}\}, D^{train}]}$$

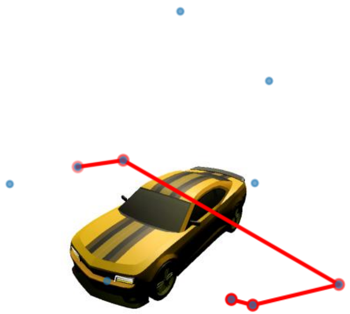
Conditional entropy of train set (D^{train})

Conditional entropy of the set contain train set (D^{train}) and given pose (x^{acq}, y^{acq})

1. Jiang, Wen, Boshu Lei, and Kostas Daniilidis. "FisherRF: Active view selection and uncertainty quantification for radiance fields using Fisher information." *arXiv preprint arXiv:2311.17874* (2023).

Optimization Problem

- Planning a trajectory to maximize the total utility within the planning window duration W



$$\text{maximize}_{\boldsymbol{P}} \sum_{i=1}^{M} \sum_{j=1}^{M} h_i \cdot v_{ij}$$

$$\text{subject to}: \sum_{i=1}^{M} v_{ij} = \sum_{k=2}^{M} v_{jk} \ \forall j \in \{p_2, p_3, \ldots, p_{N-1}\}$$

Flow conservation

$$\sum_{j=2}^{M} v_{1j} = 1;$$

Starting pose is chosen

$$\sum_{j=2}^{M} v_{ij} \leq 1; \forall i = 2, 3, \ldots, M;$$

Each pose is visited once

$$\sum_{\forall v_{ij}=1} \boldsymbol{A}[\boldsymbol{x_i}][\boldsymbol{x_j}] \leq W;$$

Expected moving time within time limit

$$v_{ij} \in \{0, 1\}, \ \forall i, j = 1, 2, \ldots, M.$$

v_{ij} show that if pose j is visited after pose i

Challenge 2

DPC (Dynamic Programming with Constraint)

- Core idea: Find the trajectory with the maximal utility within time constraint using Dynamic Programming
- Key steps
  - Explore all possible candidate pose combinations as trajectories
  - Use Dynamic Programming to avoid repeated combinations
  - Stop adding new pose into trajectory when the expected flying time exceeds planning window W
  - Return the best-known trajectory when time constraint C is used up

utility function: $\boxed{U_o(x_i, S)} = \boxed{h_i}$ Fisher Information

Utility of given pose

Challenge 2

AUM (A*-inspired Utility Maximization)

- Core idea: Add the pose with highest utility, which has highest Fisher information and subsequent weighted Fisher Information in a greedy fashion
- Key steps
  - Add the pose with highest utility as the first pose
  - Find the highest utility of next two poses x_i and x_j beyond a given trajectory $P*$
  - Add the x_i and start next iteration until the expected flying time exceeds planning window W
  - Return the best-known trajectory

Remaining moving time

utility function: $\boxed{U_A(x_i, x_j, S, t)} = h_i + \boxed{h_j/A[x_i][x_j]} \cdot \boxed{[(W - (t + A[x_i][x_j]))]}$

Utility of given poses

Next poses' weighted Fisher Information

Comparison and Recommendations

M = the number of samples pose

- DPC:
  - Time complexity: $O(M^2 2^M \log M)$
  - Suitable for applications that best quality is needed
  - Used with few sampling poses and short planning window
- AUM:
  - Time complexity: $O(M^3)$
  - Suitable for applications that real-timeness is crucial
  - Used with plenty sampling poses and long planning window

AUM

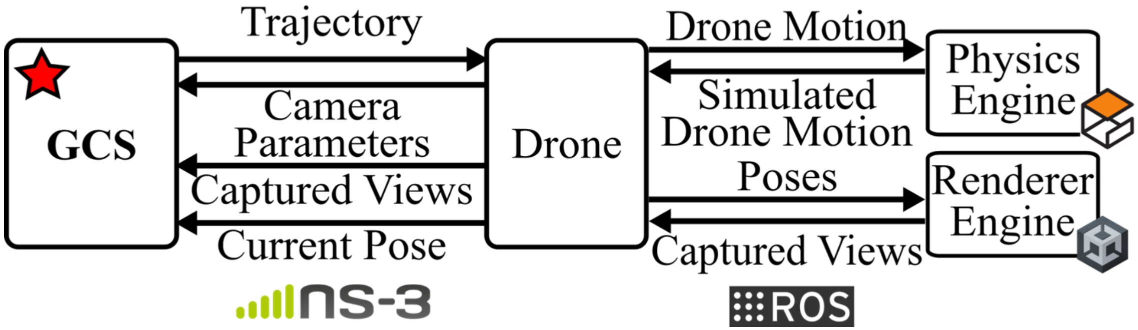
DPC

Outlines

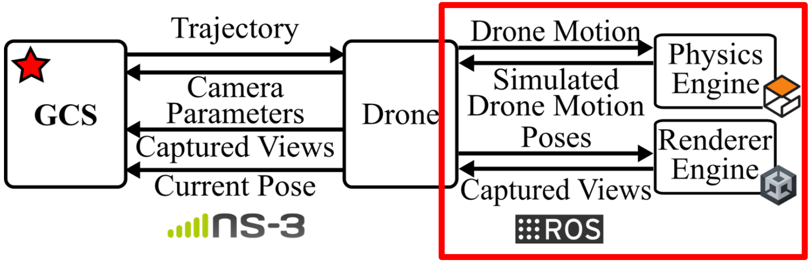
- Motivation
- Related Work
- System Architecture
- Research Problem & Algorithms
- Experiments
- Conclusion & Future Work

Implementations

- Use simulations to facilitate fair comparison and better reproducibility
- Our testbed should support:
  - Realistic physics simulation
  - Photorealistic rendering
  - Actual network with MAVLink implementation

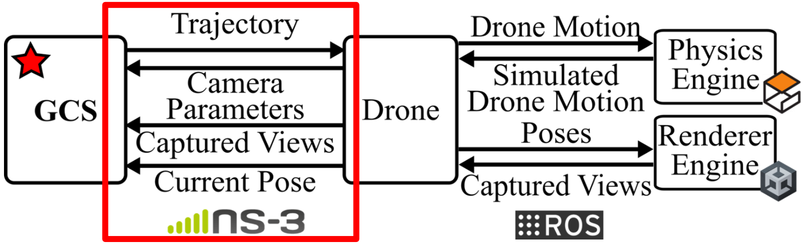


Drone Simulator



- **Physic Engine** (Gazebo) offers:
  - Realistic physic effects: gravity, wind, and robotics dynamic…
  - Various sensor plugins: GPS, IMU, and LiDAR…
  - Not capable of photo-realistic rendering
- **Renderer Engine** (Unity) offers photo-realistic rendering effects: real-time dynamic shadows, directional lights and spotlights

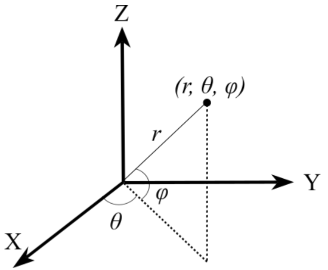
Network Simulator



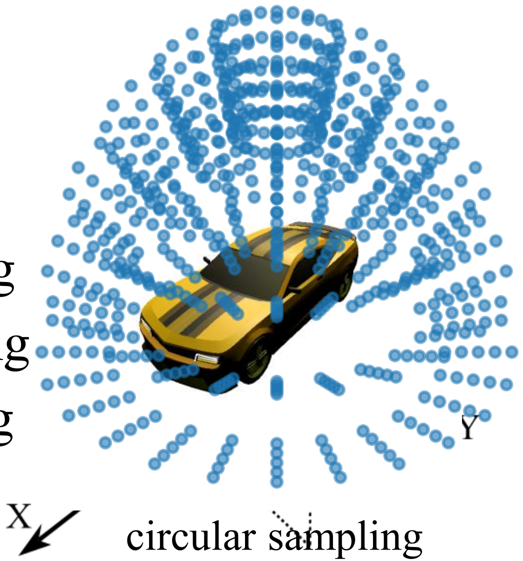
- Network simulator (NS-3) offers:
  - Multiple network protocol: Wi-Fi, DSRC, and LoRa…
  - Signal propagation effects: obstacle penetration , path loss models…
- We implemented a protocol with the following configuration:
  - Protocol: UDP over WiFi
  - Mode: Unicast
  - Message format: MAVLink

Application	MAVLink
Transport	UDP
Network	IP
Data Link	802.11 MAC
Physical	802.11g

Algorithm Parameters



- Candidate pose sampling
  - $r \in [4, 10]$, radius for random and circular sampling
  - $\phi \in \{15, 30, 45, 60\}$, longitude for circular sampling
  - $\theta \in \{0, 20, . . . , 340\}$, latitude for circular sampling
- $M \in \{5, \underline{10}, 20, 40, 80\}$, the number of samples
- $W \in \{25, \underline{50}, 75, 100\}$, the planning window size
- $C \in \{0.25, 0.5, \underline{1}, 2, 4\}$, the running time constraint



circular sampling

Sampling Poses → Candidate poses → **Uncertainty Estimation** → Candidate poses, Utility, Expected moving time → **Trajectory Planning** → Trajectory

Sampling Poses input: Sampling method (r, θ, ϕ), No. of samples (M)

Uncertainty Estimation input: 3DGS object

Trajectory Planning input: Planning window (W), time constraint (C)

Setup

Car

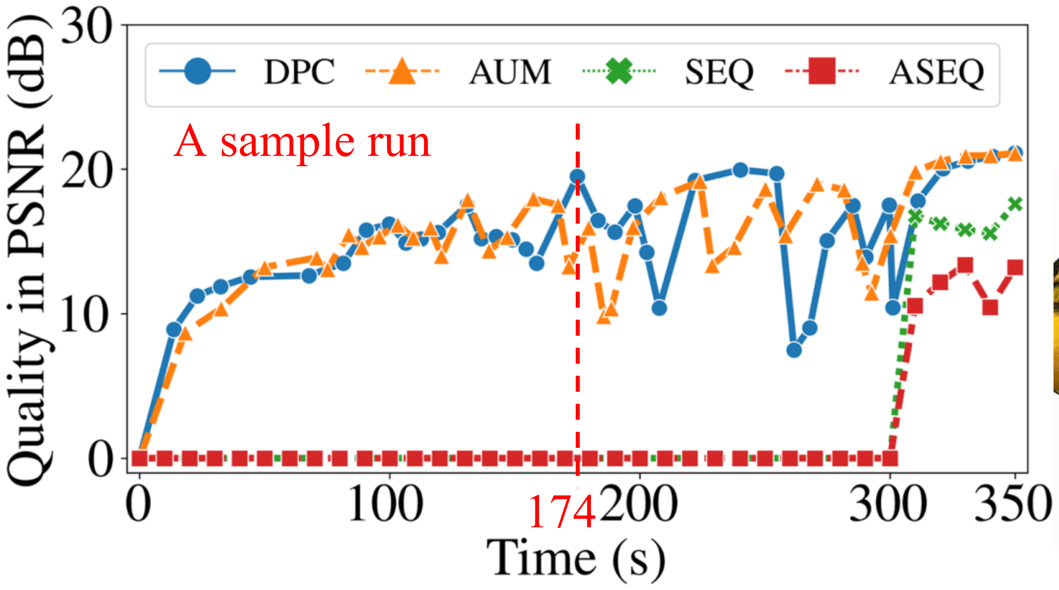
Ship

Cabin

- Sampling method
  - Random
  - Circular
- Baselines ← There was no NBVS baseline available at the time of writing
  - Using all already captured views (SEQ)
  - Selecting representative images, with the number capped at the number of input views in our algorithms (ASEQ)
- Metrics
  - Visual quality: PSNR in dB, SSIM
  - No. input views
- Evaluation results
  - Timeliness: How much time can we save by on-the-fly training? **Timeliness**
  - Performance: How much quality improvement can we achieve? **Performance**
  - Parameters: How do the parameters impact the results? **Parameters**

We Generate 3DGS Objects On-the-Fly

- DPC provides high-quality synthesized views at the 174-th second, compare to the 350-th second of SEQ/ASEQ

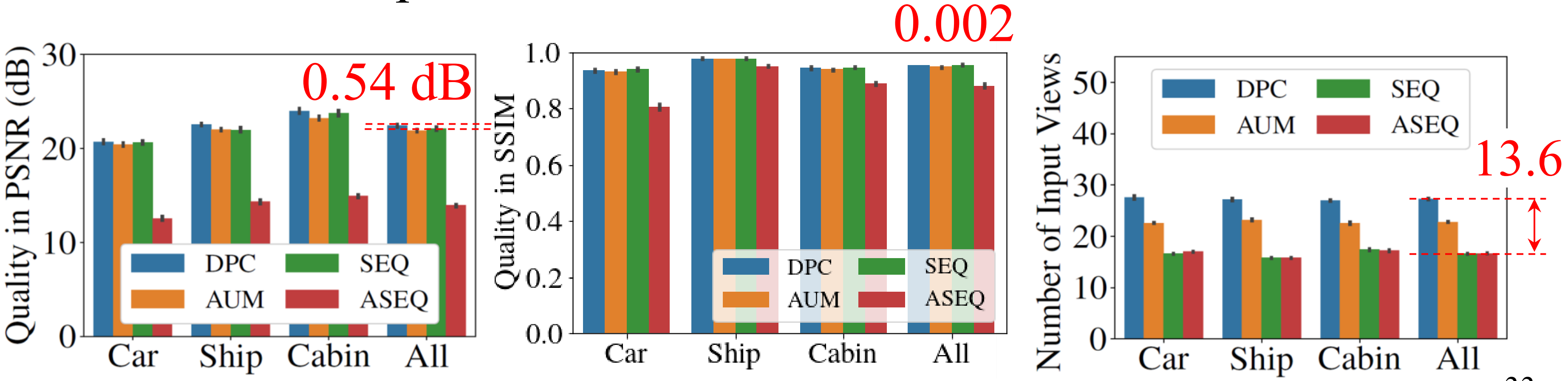


SEQ

DPC (Ours)

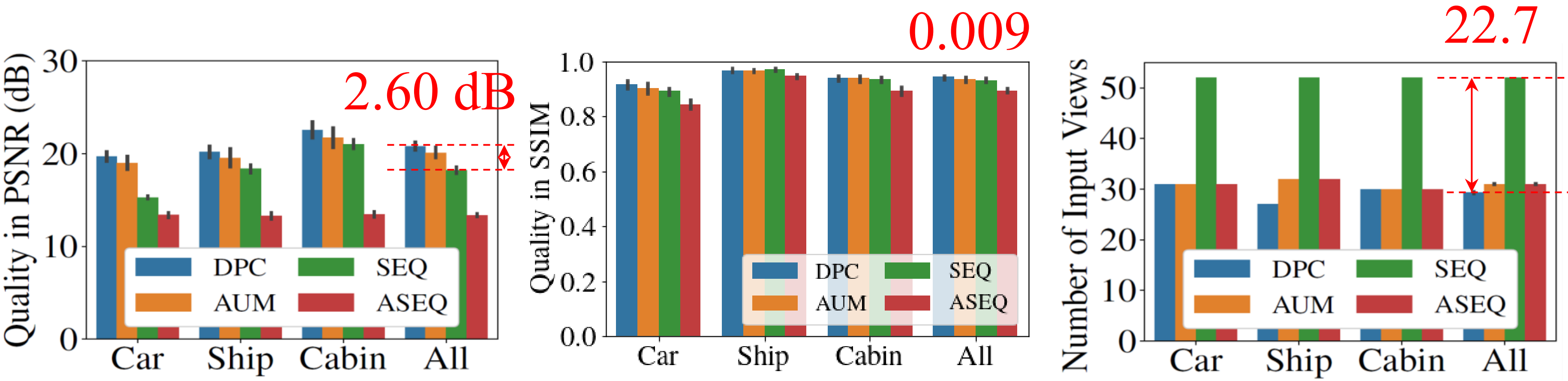
Our Algorithms Outperform the Baselines (Random Sampling)

- Although AUM underperforms SEQ by 0.45 dB in PSNR, but it run faster than the DPC (approximately 1000-fold difference, which will be shown later)
- DPC outperforms SEQ by 0.54 dB in PSNR, and captures 13.6 more input views



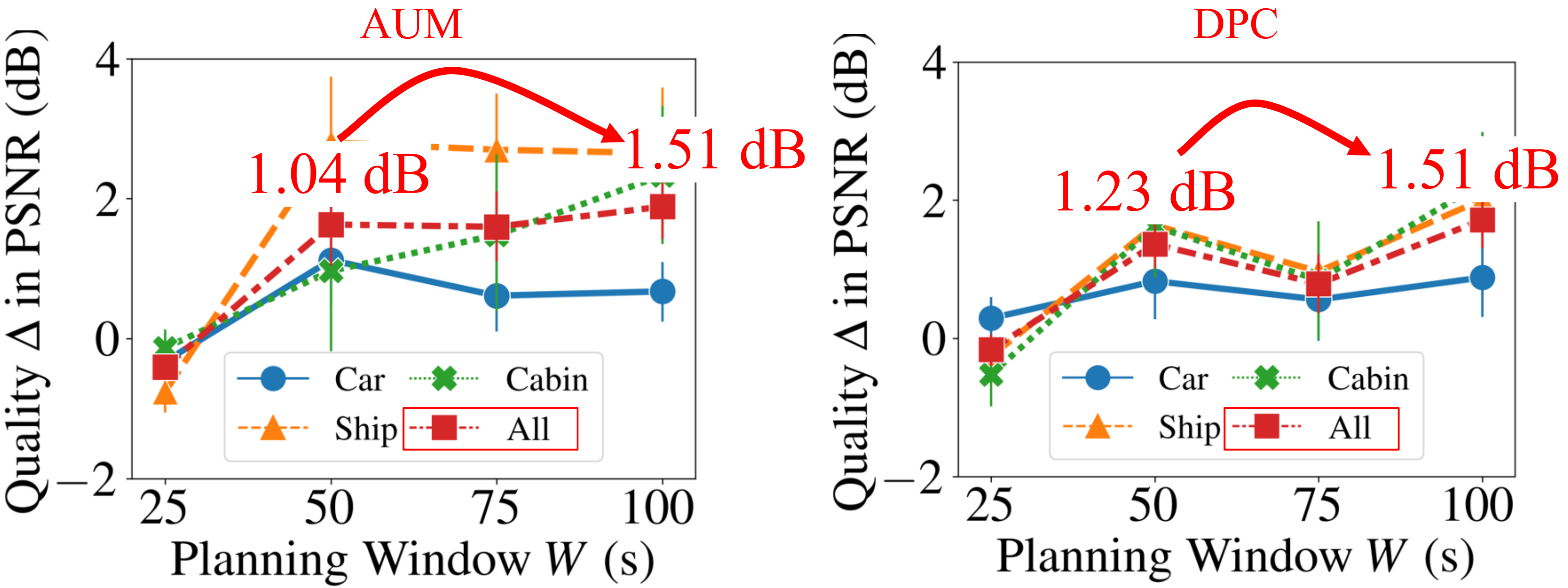
Our Algorithms Outperform the Baselines (Circular Sampling)

- AUM outperforms SEQ by 1.87 dB in PSNR, and captures 21.0 less input views
- DPC outperforms SEQ by 2.60 dB in PSNR, and capturs 22.7 less input views



Random Sampling Performs Better than Circular Sampling

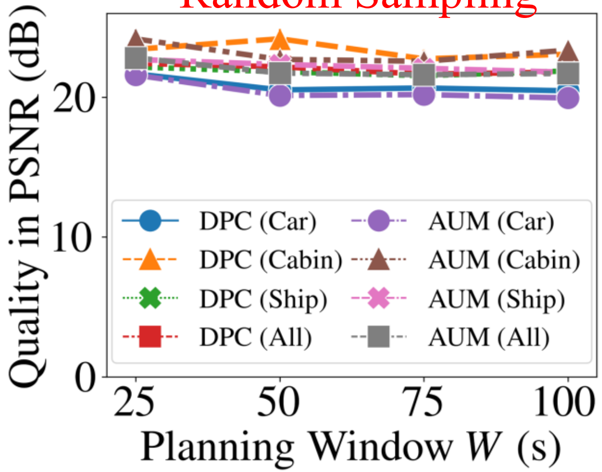
- Random sampling leads to better visual quality
- Bigger window size results in larger gap



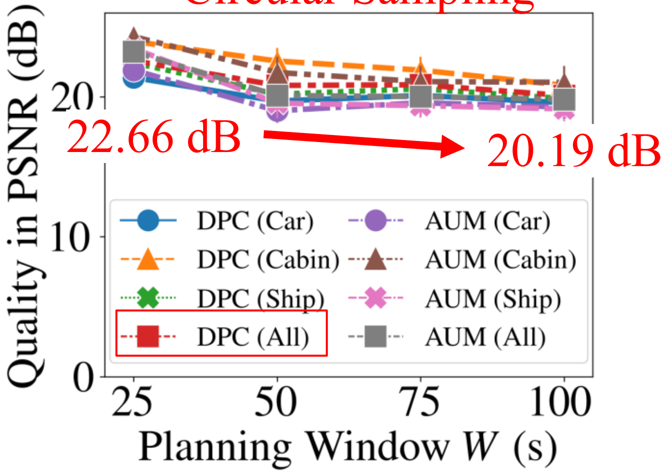
Impact of Window Size

- Random sampling: Visual quality is not affected by window size ← It is already fairly good
- Circular sampling: Visual quality drops with bigger window size from 22.66 dB to 20.19 dB in PSNR ← Larger W results in fewer re-planning opportunities

Random Sampling

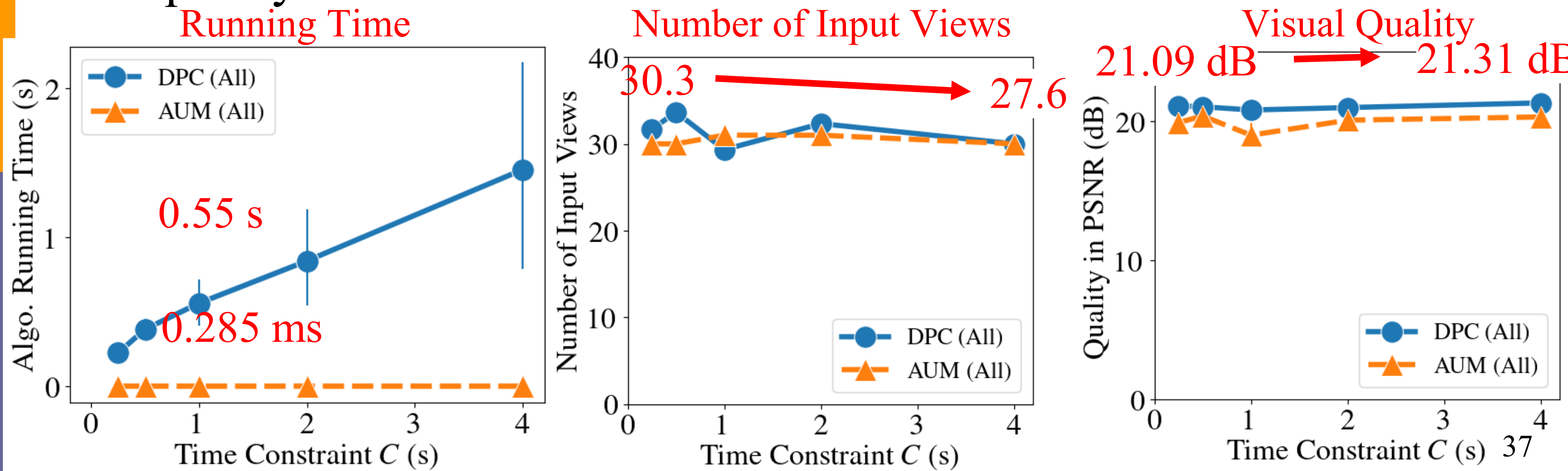


Circular Sampling



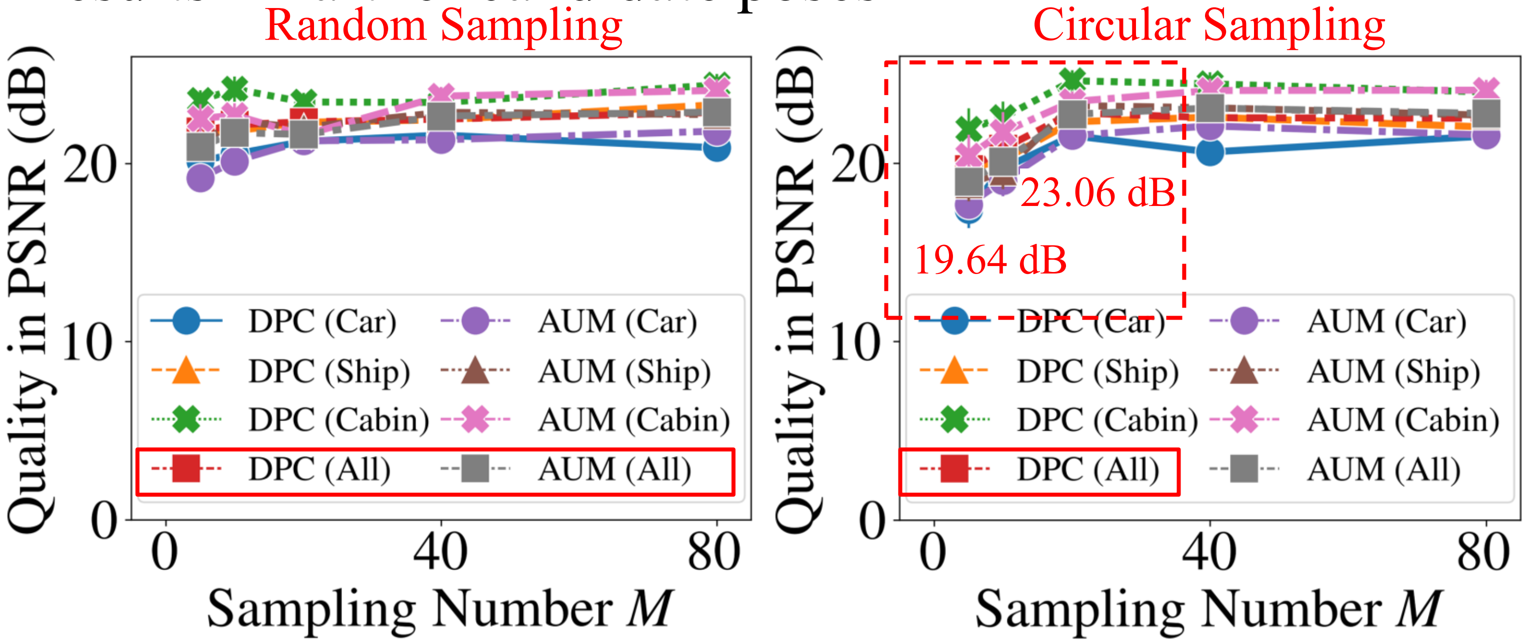
Impact of Time Constraint

- AUM runs 1000 times faster than DPC under default setting ($C = 1$ s)
- Larger C reduces the number of input views and improves the quality



Impact of Different Number of Samples

- Random sampling: It is rather stable ← It is already fairly good
- Circular sampling: More samples lead to better quality ← Larger M results in farther candidate poses



Summary of Experiments

- Achieve the final synthesized view quality in a shorter time (reduce 175.4 s on sample result) **Timeliness**
- Improve the visual quality of 3DGS objects by up to 5.90 dB in PSNR with fewer input views captured **Performance**
- Different parameters can be chosen to better suit the usage scenarios **Parameters**
  - For high visual quality, we recommend DPC with random sampling
  - For real-timeness, we recommend AUM

Outlines

- Motivation
- Related Work
- System Architecture
- Research Problem & Algorithms
- Experiments
- Conclusion & Future Work

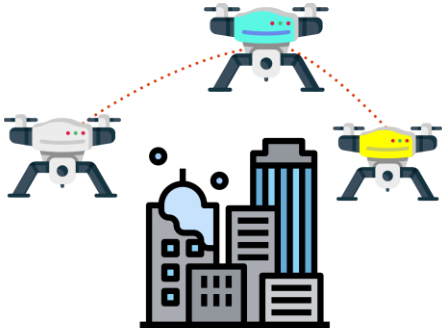
Conclusion

- 6DoF is important for many applications, including urban planning, smart agriculture, and search and rescue among others
- Selecting input views is critical for high quality view synthesis
- We are the first to propose NBVS algorithms for 3DGS objects
- Compared to the prior arts, our solution
  - Improved the visual quality (up to 5.90 dB in PSNR)
  - Achieved the final synthesized view quality without incurring long running time (reduce 175.4 s)

Future Work

Reliable Protocols

Network-Aware Trajectory Planning



Cooperation Systems

Alternative Networks

Thank you for listening!

Thanks for the help of Prof. Hsu, Yuan-Chun Sun, Cheng-Tse Lee, and all labmates

Publications:

1. **C. Wu**, Y. Sun, C. Lee, and C. Hsu, “Optimally planning drone trajectories to capture 3D Gaussian splatting objects,” in Proc. of International Conference on Multimedia Modeling (MMM’25), Nara, Japan, January 2025.
2. **C. Wu** and C. Hsu, “ FlyGS : Online 3DGS Scene Construction from MAVLink Drone Feeds” Proceedings of the 3rd Workshop on UAVs in Multimedia: Capturing the World from a New Perspective. 2025 (UAVM 2025) (**Under preparation**)

More Training Time

- The PSNR drops slightly due to overfitting