A System for Multi-View Mapping of Dynamic Scenes Using Time-Synchronized UAVs

Aniket Gupta[1], Dennis Giaya[2], Vishnu Rohit Annadanam[2], Mithun Diddi[2], Huaizu Jiang[1], Hanumant Singh[2]

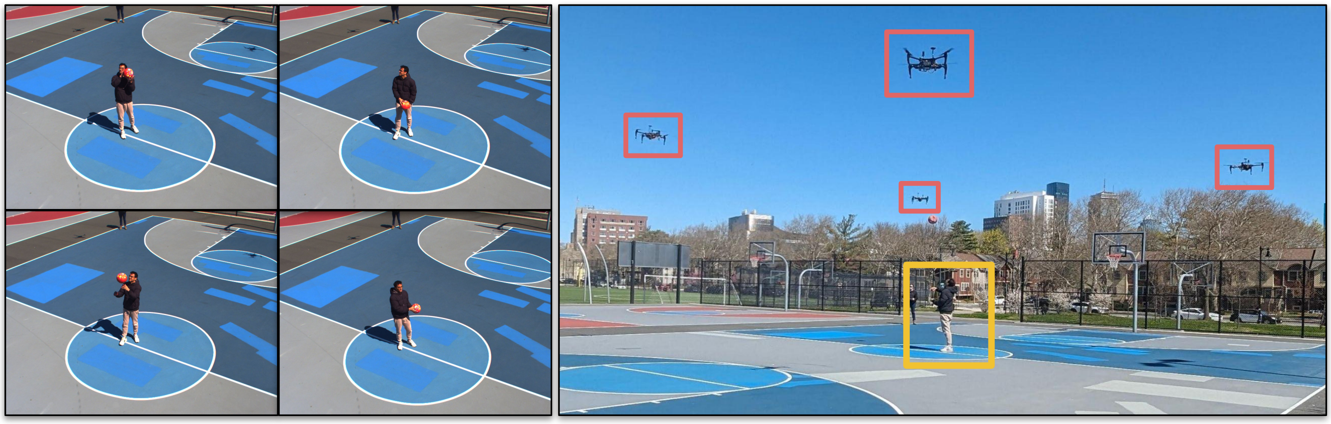


Fig. 1: We propose a novel approach to collect time-synchronized images with a fleet of UAVs for mapping dynamic scenes in unstructured outdoor environments. In our system, we utilize GNSS-PPS to synchronize cameras across UAVs and collect two dataset sequences to generate 4D maps of dynamic scenes. (a) shows the data from one out of four UAVs spanning multiple timesteps and (b) shows the multi-view data collection system at a single timestep from the sequence.

***Abstract*— Recent advances in 3D scene reconstruction, such as Neural Radiance Fields (NeRF) and 3D Gaussian Splatting, have demonstrated remarkable results in novel view synthesis and dynamic scene representation. Despite these successes, existing approaches rely on time-synchronized multi-view imagery captured using specialized camera rigs in controlled environments. This reliance limits their applicability in uncontrolled, unbounded dynamic scenes. In this work, we propose a novel Unmanned Aerial Vehicle (UAV) based multi-view capture system that leverages GNSS Pulse Per Second (PPS) signals for precise frame synchronization across multiple cameras. Our system eliminates the need for fixed infrastructure, enabling flexible and scalable data collection for dynamic scene reconstruction in diverse environments. In addition to the system architecture, we also introduce a dataset of synchronized multi-view images captured in unbounded outdoor scenes from four synchronized UAVs, each carrying a stereo camera rig. We benchmark several 3D and 4D representation methods on our dataset and highlight the challenges associated with data collection in unstructured outdoor settings such as sparse views, varied lighting conditions, visual degradation etc. Our hardware configuration details, software details and dataset is available at `https://github.com/neufieldrobotics/Dynamic_Mapping`.**

I. Introduction

Recent advances in computer vision have led to the development of powerful models like Neural Radiance Fields (NeRF) [1] and 3D Gaussian Splatting [2], which enable high-fidelity 3D scene reconstruction and novel view synthesis from multi-view images. The capabilities of these models is further extended in representing dynamic scenes by capturing multi-view images over multiple timesteps in [3], [4], [5]. However, applying these methods on dynamic scenes requires time-synchronized images from multiple viewpoints. [6], [7] are examples of some of such datasets and their data collection methods.

Traditionally, acquiring suitable datasets required specialized camera rigs or controlled studio environments with precisely synchronized cameras. The major limitation of these systems is the inability to truly capture multi-view images in unstructured outdoor scenes. Applications in both the computer vision community (e.g., human motion capture in the wild, Computer Generated Imagery (CGI), sports analysis, AR/VR content generation) and the scientific community (e.g., environmental monitoring, sea-wave mapping, animal motion capture) are often constrained by traditional data acquisition methods, limiting the practical deployment of advanced 3D reconstruction models. Consequently, there is a pressing need for flexible, scalable systems that can collect synchronized multi-view imagery without requiring careful instrumentation of the scene in advance.

[1]Khoury College of Computer Sciences, Northeastern University, Boston, MA 02115. HJ is supported by the National Science Foundation under Award IIS-2310254. [2]Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115. {gupta.anik, giaya.d, annadanam.v, h.jiang, ha.singh}@northeastern.edu

In this paper, we introduce a novel system that leverages a fleet of Unmanned Aerial Vehicles (UAVs) equipped with stereo cameras synchronized using the Global Navigation Satellite System (GNSS) Pulse Per Second (PPS) signal. With the GNSS PPS signal, we achieve precise time synchronization across multiple UAVs, ensuring that all cameras capture images simultaneously. By eliminating the need for specialized camera rooms or tedious synchronization mechanisms, our system enables the collection of time-synchronized, multi-view datasets in outdoor environments, overcoming the constraints associated with traditional methods. Moreover, the mobility of UAVs allows for adaptable positioning and coverage of large or inaccessible areas, facilitating the capture of complex scenes from diverse perspectives. This capability is particularly beneficial for feeding data into models like NeRFs and Gaussian Splatting, which require high-quality, synchronized imagery for optimal performance.

To the best of our knowledge, this is the first time wireless synchronized UAVs have been used to collect high-quality time-synchronized multi-view images to support reconstruction of a dynamic environment in a distributed manner. Using our system of four UAVs, we collect two *long* sequences of dynamic scene with 360º and 180º scene coverage, respectively. This dataset enables new research directions in scene representation, allowing exploration of reliable and geometrically accurate scene representation methods and highlighting real-world data collection challenges such as sparse views, large viewpoint shifts, and varying lighting conditions.

Our contributions can be summarized as follows:

- We present the system design of a dynamic multi-view capture system consisting of a GNSS PPS-based synchronization system for coordinating multiple UAV-mounted cameras, enabling precise time alignment without specialized infrastructure.
- We release two long dataset sequences with varying configurations capturing a dynamic scene. Both sequences are captured using 4 UAVs with a stereo camera onboard. We also release our hardware and software configuration details.
- We have benchmarked several state-of-the-art algorithms across 3D and 4D representation methods and present a comparison among the algorithms both qualitatively and quantitatively highlighting fundamental issues that need the entire community's effort to tackle.

II. Related Work

A. Marker Based Motion Capture

Marker-based motion capture (mocap) systems involve attaching reflective or active markers to the subject, which are then tracked by multiple cameras or sensors. The cameras emit or detect specific wavelengths of light reflected by these markers to triangulate their precise positions in 3D space. The resulting data provides accurate joint trajectories for applications in film production, video games, sports science, and medical rehabilitation [8], [9], [10] . A key advantage of marker-based systems is their high accuracy and reliability, which has made them an industry standard in many professional environments. However, these systems require controlled conditions (such as studios with calibrated cameras), significant preparation time, and may constrain the performer's natural movement if many markers are attached. Our system on the other hand does not require any markers, and is not limited to studio settings. Moreover, since UAVs are mobile, our system can be extended to track objects across large scenes.

B. Markerless Motion Capture

Markerless motion capture aims to track human movement without the need for special markers or suits. Instead, these methods rely on computer vision techniques, such as feature detection, silhouette analysis, or deep learning-based pose estimation, to infer body joint positions and orientations [6], [7]. The removal of markers simplifies the mocap pipeline and reduces preparation time. It also allows for more natural movement as participants are not encumbered by markers. However, markerless approaches require synchronized cameras, and thus they are limited to studio settings where wired synchronization can be easily implemented. In contrast, our system is suitable for deploying an effectively unlimited number of synchronized nodes in outdoor scenes globally without any prior environmental preparation required.

C. Sensor Synchronization

In multi-sensor motion capture systems—whether marker-based or markerless—synchronizing data streams from different sensors (cameras, inertial measurement units, depth sensors, etc.) is critical for accurate spatiotemporal alignment. The general concept in wireless synchronization[11] is to transmit a low-frequency synchronization signal over a wireless channel from a centralized source with a precise clock and recover the synchronization pulse for disciplining local oscillators at multiple receiver nodes. The [12], [13], [14] outlines popular wireless synchronization methods and associated challenges. Some of them in [15], [16], [17], [18] require proprietary, closed source hardware at the transmitter and receiver nodes. At the same time, software protocols in most methods [19], [20] are closed-sourced, with exceptions being [12]. Most of these methods use dedicated infrastructure to relay the synchronization signal instead of using existing physical layers or links on robots. Our system on the other hand uses only commercially available hardware and open-source software to achieve synchronization across multiple nodes.

III. Data Collection System

A. UAV System Description

Our UAV data collection system was designed to facilitate the synchronized and geo-referenced acquisition of images across multiple platforms in an outdoor environment. In addition to the standard flight components, each UAV comprises a multi-band GNSS receiver, a synchronization

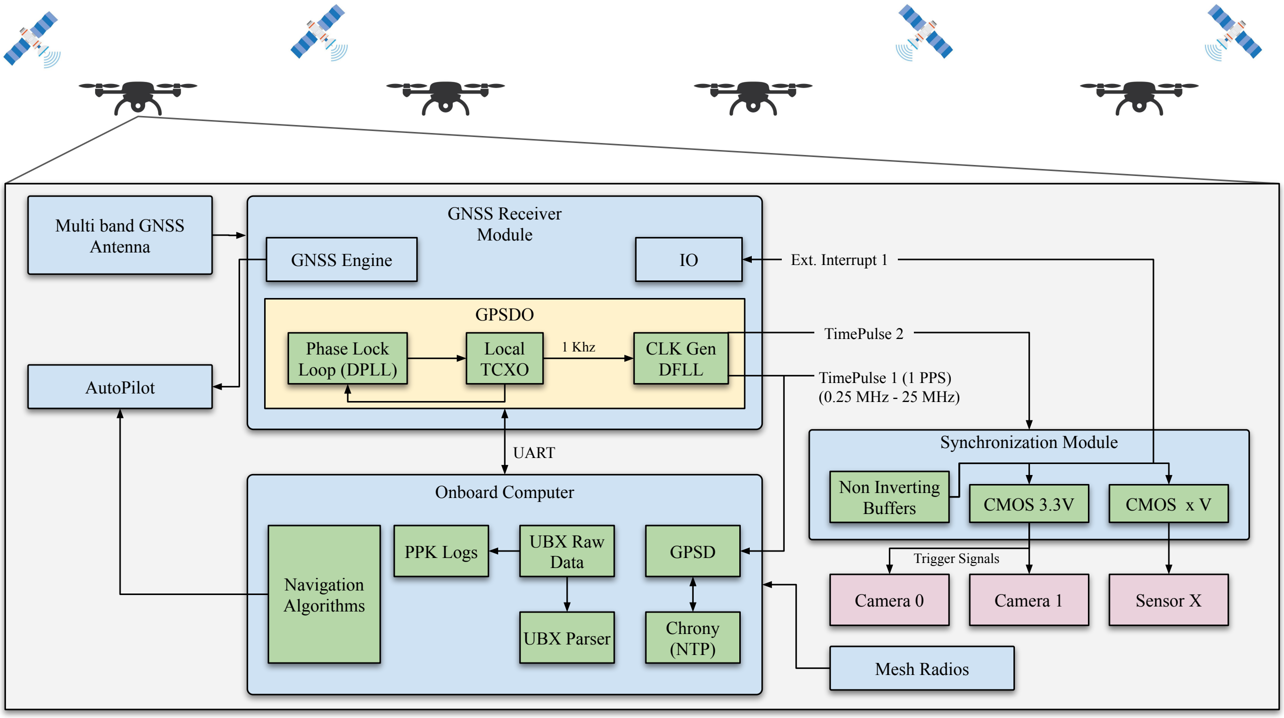


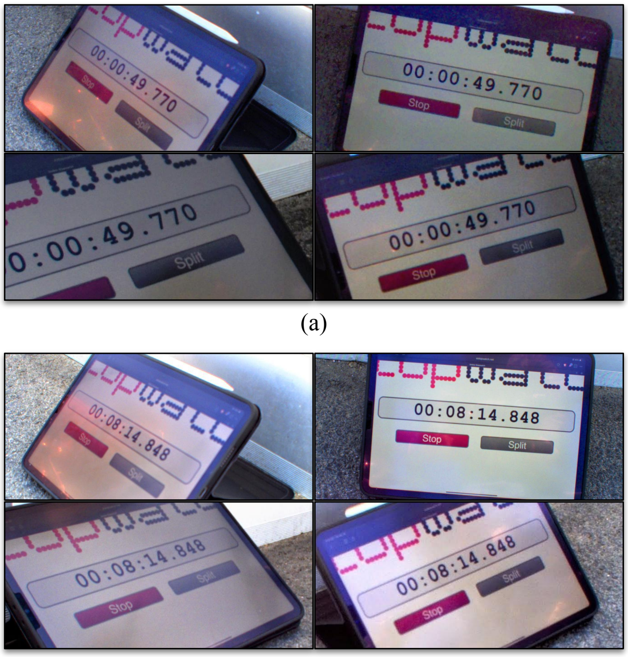
Fig. 2: Hardware system architecture of the multi-UAV data collection system. Each UAV consists of (1) a GNSS receiver module for navigation and time-synchronization, (2) an onboard computer for running navigation algorithms, monitoring, and logging data, and (3) a synchronization module that generates trigger signals for sensors. Mesh radios facilitate communication between UAVs and the Ground Control Station (GCS).

trigger distribution board, an onboard payload computer for data processing and storage, and a stereo camera rig for image acquisition. Fig. 2 shows an overview of our complete hardware system.

1) GNSS Receiver: In our design, the Ublox Zed-F9T GNSS IC is integrated along with a lightweight helical multi-band GNSS antenna. The resulting multi-band GNSS solution provides accurate positioning and timing information for subsequent data processing.

The Ublox Zed-F9T GNSS IC is a developer-friendly, low-power multi-band GNSS receiver with a 5ns timing accuracy and a user configurable GPS disciplined oscillator (GPSDO) capability. The temperature-controlled local oscillator (TCXO) on F9T outputs a 1kHz reference, which is disciplined using an estimated 1*PPS* reference from the GNSS engine. The TCXO is used to generate two variable-frequency user-configurable time-pulse (TP) outputs, TP1 and TP2. These reference signals are aligned to the top of the second via the GNSS receiver and therefore aligned across multiple UAVs. The time pulse output signals can be configured from 0.25Hz to 25MHz.

For our purposes, TP1 is configured to 1Hz and enables precise synchronization of the onboard computer with UTC time. TP2 is configured to the desired camera frame rate of 10Hz. The signal from TP2 is distributed to both cameras through the synchronization board.



(a)

(b)

Fig. 3: Visualization of time synchronization across all UAVs. (a) shows the images captured from all four UAVs before starting data collection. (b) shows the images captured right after data collection.

2) Synchronization board: The synchronization board consists of Texas Instruments SN74LS07 IC with six non-inverting hex buffers and connectors for power, debugging, input sync signal, and trigger outputs.

The synchronization board takes the time-pulse output from the GNSS as input and distributes it to a variety of end devices. In doing so, it sources additional current to buffer the signal and can also facilitate logic inversion and level shifting to meet the requirements of each particular device. For example, the FLIR Blackfly S cameras used for this application require a 3.3V tolerant Transistor-Transistor Logic (TTL) for their external trigger.

3) Onboard Computer: An Nvidia Jetson Orin Nano 8GB serves as the onboard computer for each UAV. This computer receives and stores sensor data. The integrated GPU provides the capability to evaluate algorithms onboard and a serial connection to the UAV flight controller enables execution of autonomous flight behaviors. The computer software stack includes the Robot Operating System (ROS)[21] which provides a flexible framework to implement vehicle behaviors as ROS nodes and also to log timestamped sensor data from multiple sensors using the provided rosbag utility.

4) Stereo Camera: Each UAV is equipped with a calibrated stereo camera rig which is triggered by the GNSS TP2 signal provided through the synchronization board. The rig consists of two Blackfly S BFS-U3-16S2C cameras mounted at a baseline of 20cm and configured to capture RGB images at 1440 × 1080 resolution. Images are collected at 10 frames per second (FPS), as dictated by the frequency of TP2. Fig 4 shows the complete hardware setup on one UAV.

B. Ground Control Station System Description

The Ground Control Station (GCS) includes a Linux computer and a WiFi access point. The primary role of the GCS is to provide situational awareness to the UAV operators during a mission. By connecting to the UAV OBCs through the GCS, it is possible to verify time synchronization and data collection processes are running. Additionally, a GStreamer process running on each UAV streams images over UDP to the GCS in real-time. To optimize bandwidth usage, these streamed images are transmitted at a reduced resolution and a configurable downsampled frame rate, typically 1/10th of actual acquisition rate.

C. Time synchronized and geo-referenced data collection

We combine four of the described UAV systems with a GCS to implement a synchronized multi-view capture system.

The Ublox Zed F9T is the backbone of our time synchronization system. By synchronizing all UAVs and GCS to GNSS UTC time and aligning all sensor trigger signals with the phase of the GNSS 1PPS signal, we are able to achieve sub microsecond frame synchronization across all cameras, as demonstrated by Figure 3.

1) Synchronization of system time: The GPS daemon, gpsd, running on the GCS ingests GNSS data in the form of binary UBX messages. It publishes the global time information parsed from these messages to a shared memory interface for the Linux network time protocol (NTP) daemon: chrony. Chrony, receiving both UTC time from GPSD and a precise 1 PPS top-of-second reference from a general purpose input/output pin connected directly to the GNSS IC, is able to precisely synchronize system time with UTC. The GCS then functions as a stratum 1 NTP time source for all of the UAV OBCs. The GCS and UAVs communicate over a WiFi local area network provided by the GCS access point.

2) Monitoring binary UBX messages: The UBX-TIM-TM2 and UBX-MON-RF are monitored by each UAV to (a) Obtain and log the timestamps of output trigger pulses from the synchronization board in UTC time and (b) Detect the presence of an active jammer in the vicinity.

3) Logging raw GNSS data: The raw GNSS data on all UAVs and on the GCS is logged and can be processed to obtain a post-processed kinematic (PPK) GNSS solution. It can serve as a ground truth reference for all UAV positions. The raw UBX data can be processed by the RTKCONV and RTKPOST apps in the open-source GNSS processing library RTKLIB[22], to produce the PPK GNSS solution. Additionally, the GPSD-based Robot Operating System (ROS) driver[23] is used to publish GNSS positioning data to navigate the UAVs to desired waypoints in real time.

4) Image post-processing scripts: The images captured across multiple UAVs are temporally correlated using the ROS message timestamp consistent across rosbags from all OBCs. Using rosbags from multiple UAVs, the post-processing scripts output synchronized image frames from multiple viewpoints.

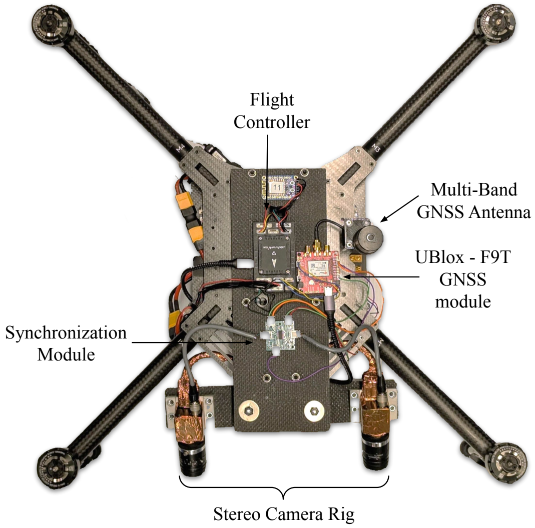


Fig. 4: UAV integrated with our proposed stack for synchronous Image acquisition across multiple nodes

IV. Dataset

We collect two sequences using our system in outdoor environments. In both datasets, four UAVs hover in formation

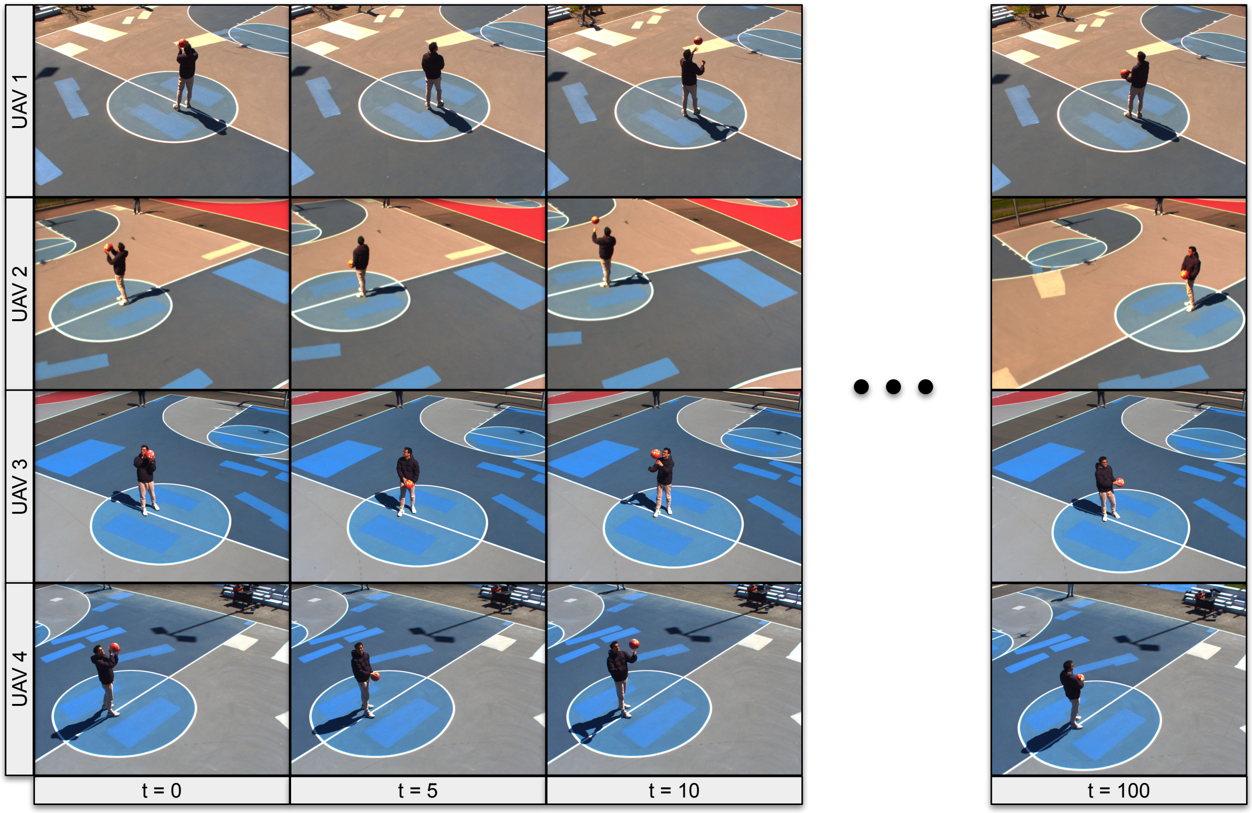


Fig. 5: Visualization of data samples from the 360 sequence from different viewpoints at different timesteps.

Label	Duration (s)	Number of Images	Size	Description
360	302	24160	92GB	symmetric scene, varying lighting, large view-point change
180	321	25680	97GB	unbounded scene, moderate view-point change

TABLE I: List of sequences in our datasets and their description

surrounding a human playing with a ball. Table I summarizes key properties of each sequence.

A. 360 Sequence

For the 360 sequence, the four UAVs are uniformly distributed across a full 360º of potential subject viewing angles, allowing for complete coverage of the subject located at the center of a basketball court. The court's largely symmetrical layout introduces geometric ambiguities, challenging reconstruction methods that rely on distinctive feature points. Additionally, because two UAVs face the sun while two do not, the onboard camera processing results in visible color and brightness inconsistencies—specifically, the court's gray surface appears yellowish in some views. These lighting and color variations highlight the challenges of collecting data in the real-world. Fig 5 shows some example images from this sequence.

B. 180 Sequence

In contrast, the 180 sequence has the four UAVs distributed uniformly across only approximately 180º of the potential subject viewing angles. This captures a more limited frontal sector of the scene, but offers a higher degree of overlap between adjacent views. Recorded on a football field, this setting provides more texture and features than the basketball court, although reliably establishing correspondences across the wider area remains a challenge. Despite its relatively denser coverage, the dataset still tests the resilience of 3D reconstruction methods to real-world factors such as partial occlusions, minor lighting shifts, and variations in camera orientation.

V. Benchmarking the SOTA

To demonstrate the quality and usefulness of the dataset, we benchmark across a set of well-known state-of-the-art 3D representation methods. The selected algorithms are chosen to provide comprehensive coverage of the field, including NeRFs [1], 3D Gaussian splatting [2], Sparse-NeRFs [24], [25], [26] and Sparse 3D Gaussian Splatting methods [27], [28], [29]. We also investigate 4D representation methods such as 4D Gaussian Splatting [4] and DynNeRF [30].

A. Evaluation

Both NeRF and 3DGS type methods require camera poses for each image which are generally obtained using Structure from Motion (SFM) techniques. Because our dataset contains

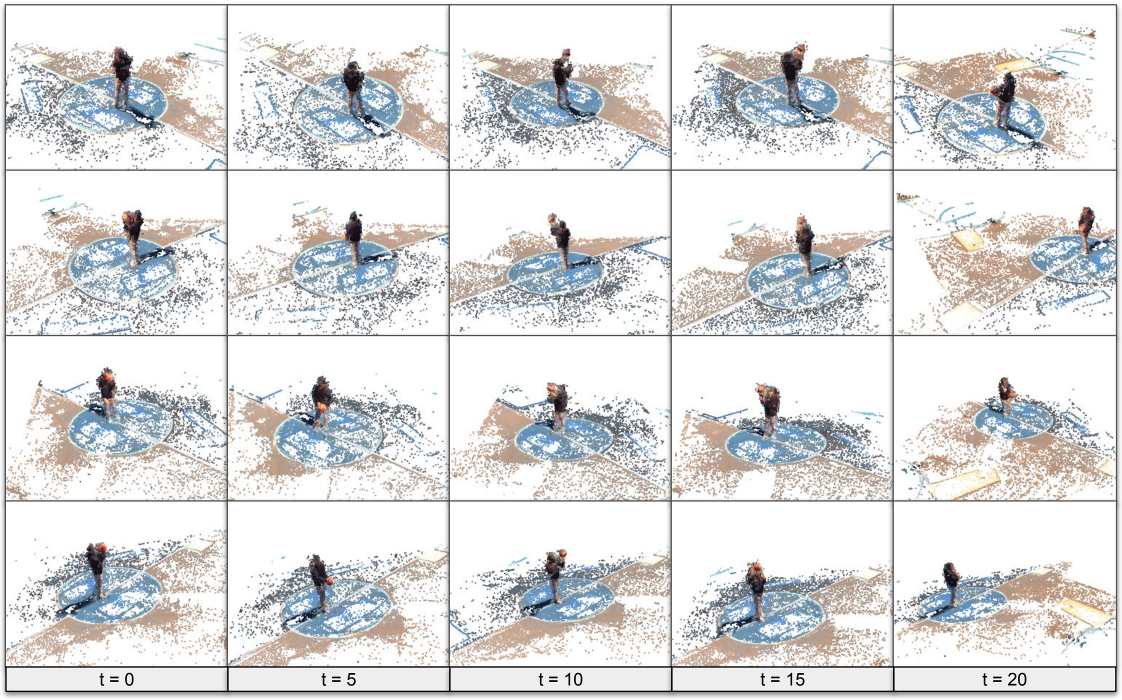


Fig. 6: Visualization of sparse dynamic reconstruction from different timesteps and from different view points generated using VGGSFM [31]. *For a more comprehensive representation of the dynamic point cloud, please refer to the accompanying video submission.*

images with large viewpoint changes, classical SFM tools such as COLMAP [32], [33] fail to align images across UAVs. The only SFM method able to do so is VGGSFM [31]. We generate a sparse point cloud and corresponding camera poses for each timestep using [31] and use that to initialize all 3D methods for evaluation. For 4D reconstruction, we first register images from each drone along the time dimension and use those pose estimates to align the sparse point clouds from each timestep. Fig. 6 shows snapshots of the sparse dynamic point cloud across different timesteps. Each timestep contains a total of 8 images: 2 from each UAV. We use 7 for training and hold out 1 for evaluation. For the 4D representation methods, we use images at all timesteps from 7 cameras and hold out images from 1 camera for evaluation. The evaluation results are summarized in Table II and Table III. Fig. 7 shows the reconstruction of the hold out image from 3D representation models.

	360 Sequence			180 Sequence		
Method	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓
3DGS	0.80	23.74	0.37	0.34	17.37	0.47
SparseGS	0.74	17.90	0.37	0.41	18.95	0.40
InstantSplat	0.78	21.26	0.37	0.43	19.81	0.43
2DGS	0.73	18.07	0.46	0.24	14.94	0.55
NeRF	0.47	16.11	0.42	0.23	15.35	0.58
SparseNeRF	0.48	16.74	0.30	0.21	15.97	0.54
Zip-NeRF	0.93	25.67	0.08	0.75	21.67	0.17

TABLE II: List of sequences in our datasets and their description

	360 Sequence			180 Sequence		
Method	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓
4DGS	0.67	15.17	0.53	0.61	16.37	0.57
DynNERF	0.58	16.25	0.47	0.54	15.92	0.44

TABLE III: List of sequences in our datasets and their description

B. Discussion

1) Sparse Views: While our system theoretically scales to any number of UAVs, logistical concerns and cost will ultimately place an upper bounds on the number used. Due to resource constraints, we only utilized four UAV systems for our data collection. With the 360 sequence in particular, we want to highlight the problem of using existing 3D representation methods with only sparse views. As evident from Tab. II, even the latest state-of-the-art methods suffer when the viewing angles are sparse. This can be primarily attributed to the fact that these methods are not grounded in scene geometry and only minimize photometric loss over the given images. In our experiments, only Zip-NeRF [26] performs reasonably well on both sequences, likely due to its use of a multi-scale optimization strategy.

2) Dynamic Reconstruction in the Wild: In both sequences, our dataset targets the challenge of representing dynamic scenes from sparse, multi-view data. Most existing work in this domain focuses on monocular camera rigs [34], [35], leveraging different timesteps and camera poses in a shared optimization to train a radiance field over an

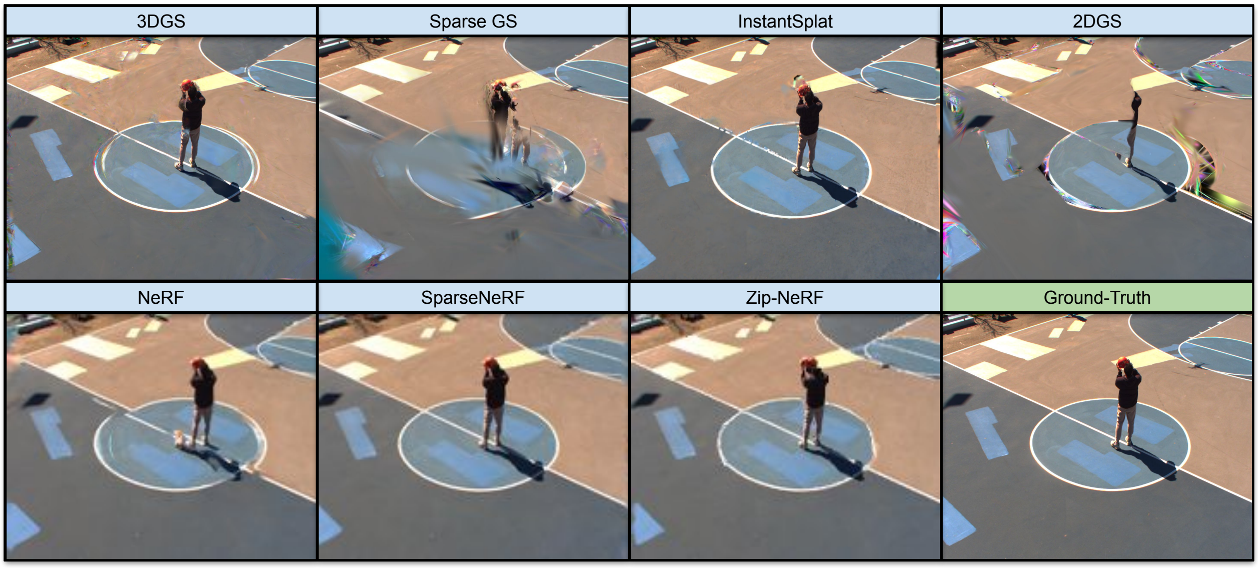


Fig. 7: This figure shows the output mesh generated by the evaluated 3D representation methods.

entire video. To evaluate multi-view dynamic scene representation, we tested the methods of [4] and [30], but since these approaches are not designed to handle sparse camera viewpoints, they exhibit suboptimal performance on both sequences in our dataset.

3) Potential Usage: The previous discussion clearly shows the current limitations of 3D and 4D representation methods. Despite the challenges of sparse camera coverage, varying illumination from viewpoint shifts, and limited geometric grounding—the system and dataset we present offer an invaluable testbed for pushing the boundaries of 3D/4D reconstruction. Researchers can use our real-world scenarios to develop and validate new approaches that integrate richer geometry priors and more robust optimization strategies. Furthermore, applications in both computer vision community (CGI, AR/VR, sports analytics etc) and science community (environmental monitoring, sea-wave mapping, animal motion capture etc) stand to benefit from solutions that can handle such imperfect, field-collected data and still produce reliable reconstructions. One potential research direction is to use diffusion models to improve reconstruction with sparse views as presented in [36], [37]. These approaches could also be extended to dynamic scenes by using video diffusion models instead of image-diffusion.

VI. Conclusion and Future Work

In this work, we tackled the challenge of capturing time-synchronized, multi-view imagery for dynamic scene reconstruction outside controlled studio environments. Our novel multi-UAV system, which leverages GNSS PPS-based synchronization, circumvents the need for specialized infrastructure or wired synchronization configurations and enables data collection in unstructured outdoor settings. We contribute a comprehensive dataset of two extended sequences, each captured with four UAVs equipped with stereo cameras. This dataset inherits all the complexities of data from an uncontrolled environment, including sparse viewpoints, symmetries, featureless regions, and dynamic elements. Our experimental evaluations on state-of-the-art scene representation approaches highlight both the promise and shortcomings of current static/dynamic scene representation methods. With this dataset, we aim to drive the development of 3D/4D representation methods that can operate reliably in the real-world.

Our system's potential extends beyond conventional 3D/4D reconstruction tasks, as it paves the way for a range of applications for the computer vision and the science community. In ongoing and future work, we plan to expand our dataset and incorporate additional scenes with dynamic object tracking, formation flying, and other advanced data collection maneuvers. We also intend to pursue new algorithmic solutions—both independently and through collaborations—to further push the boundaries of real-world 3D and 4D reconstruction methods.

References

[1] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *CoRR*, vol. abs/2003.08934, 2020. [Online]. Available: https://arxiv.org/abs/2003.08934

[2] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," 2023. [Online]. Available: https://arxiv.org/abs/2308.04079

[3] J. Luiten, G. Kopanas, B. Leibe, and D. Ramanan, "Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis," in *3DV*, 2024.

[4] Z. Yang, H. Yang, Z. Pan, and L. Zhang, "Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting," in *International Conference on Learning Representations (ICLR)*, 2024.

[5] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer, "D-NeRF: Neural Radiance Fields for Dynamic Scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[6] H. Joo and et al., “Panoptic studio: A massively multiview system for social motion capture,” 2015.

[7] T. Li, M. Slavcheva, M. Zollhoefer, S. Green, C. Lassner, C. Kim, T. Schmidt, S. Lovegrove, M. Goesele, R. Newcombe, and Z. Lv, “Neural 3d video synthesis from multi-view video,” 2022. [Online]. Available: https://arxiv.org/abs/2103.02597

[8] M. Plappert, C. Mandery, and T. Asfour, “The KIT motion-language dataset,” *Big Data*, vol. 4, no. 4, pp. 236–252, dec 2016. [Online]. Available: http://dx.doi.org/10.1089/big.2016.0028

[9] M. Trumble, A. Gilbert, C. Malleson, A. Hilton, and J. Collomosse, “Total capture: 3d human pose estimation fusing video and inertial sensors,” in *2017 British Machine Vision Conference (BMVC)*, 2017.

[10] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber, “Documentation mocap database hdm05,” Universität Bonn, Tech. Rep. CG-2007-2, June 2007.

[11] Y.-C. Wu, Q. Chaudhari, and E. Serpedin, “Clock synchronization of wireless sensor networks,” *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 124–138, 2011.

[12] K. Alemdar, D. Varshney, S. Mohanti, U. Muncuk, and K. Chowdhury, “Rfclock: Timing, phase and frequency synchronization for distributed wireless networks,” in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom ’21. New York, NY, USA: Association for Computing Machinery, 2021, p. 15–27. [Online]. Available: https://doi.org/10.1145/3447993.3448623

[13] S. Youn, “A comparison of clock synchronization in wireless sensor networks,” *International Journal of Distributed Sensor Networks*, vol. 9, no. 12, p. 532986, 2013. [Online]. Available: https://doi.org/10.1155/2013/532986

[14] B. Sundararaman, U. Buy, and A. D. Kshemkalyani, “Clock synchronization for wireless sensor networks: a survey,” *Ad Hoc Networks*, vol. 3, no. 3, pp. 281–323, 2005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570870505000144

[15] A. Dongare, P. Lazik, N. Rajagopal, and A. Rowe, “Pulsar: A wireless propagation-aware clock synchronization platform,” in *2017 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*. IEEE, 2017, pp. 283–292.

[16] O. Abari, H. Rahul, D. Katabi, and M. Pant, “Airshare: Distributed coherent transmission made seamless,” in *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2015, pp. 1742–1750.

[17] V. Yenamandra and K. Srinivasan, “Vidyut: Exploiting power line infrastructure for enterprise wireless networks,” *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4, pp. 595–606, 2014.

[18] D. H. Nguyen, A. Paatelma, H. Saarnisaari, N. Kandasamy, and K. R. Dandekar, “Sub-microsecond network synchronization for distributed wireless phy protocols,” in *Proceedings of the 9th ACM Workshop on Wireless of the Students, by the Students, and for the Students*, 2017, pp. 3–5.

[19] H. Katabi, “Megamimo: Scaling wireless capacity with user demands,” 2012.

[20] H. Rahul, H. Hassanieh, and D. Katabi, “Sourcesync: A distributed wireless architecture for exploiting sender diversity,” *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 4, pp. 171–182, 2010.

[21] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, “Ros: an open-source robot operating system,” in *2009 IEEE International Conference on Robotics and Automation (ICRA) Workshop on Open-Source Software*, 2009, pp. 1–6. [Online]. Available: https://robotics.stanford.edu/∼ang/papers/icraoss09-ROS.pdf

[22] “Open source rtk processing library.” [Online]. Available: https://github.com/tomojitakasu/RTKLIB

[23] “Ros wrapper for gpsd library.” [Online]. Available: https://github.com/ros-drivers/gps_umd

[24] G. Wang, Z. Chen, C. C. Loy, and Z. Liu, “Sparsenerf: Distilling depth ranking for few-shot novel view synthesis,” in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[25] P. Truong, M.-J. Rakotosaona, F. Manhardt, and F. Tombari, “Sparf: Neural radiance fields from sparse and noisy poses,” 2023. [Online]. Available: https://arxiv.org/abs/2211.11738

[26] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, “Zip-nerf: Anti-aliased grid-based neural radiance fields,” 2023. [Online]. Available: https://arxiv.org/abs/2304.06706

[27] H. Xiong, S. Muttukuru, R. Upadhyay, P. Chari, and A. Kadambi, “Sparsegs: Real-time 360° sparse view synthesis using gaussian splatting,” 2023.

[28] Z. Fan, W. Cong, K. Wen, K. Wang, J. Zhang, X. Ding, D. Xu, B. Ivanovic, M. Pavone, G. Pavlakos, Z. Wang, and Y. Wang, “Instantsplat: Unbounded sparse-view pose-free gaussian splatting in 40 seconds,” 2024.

[29] B. Huang, Z. Yu, A. Chen, A. Geiger, and S. Gao, “2d gaussian splatting for geometrically accurate radiance fields,” in *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024.

[30] C. Gao, A. Saraf, J. Kopf, and J.-B. Huang, “Dynamic view synthesis from dynamic monocular video,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2021.

[31] J. Wang, N. Karaev, C. Rupprecht, and D. Novotny, “Vggsfm: Visual geometry grounded deep structure from motion,” 2023.

[32] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, “Pixelwise view selection for unstructured multi-view stereo,” in *European Conference on Computer Vision (ECCV)*, 2016.

[33] J. L. Schönberger and J.-M. Frahm, “Structure-from-motion revisited,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[34] B. Park and C. Kim, “Point-dynrf: Point-based dynamic radiance fields from a monocular video,” in *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2024, pp. 3159–3169.

[35] Q. Wang, V. Ye, H. Gao, W. Zeng, J. Austin, Z. Li, and A. Kanazawa, “Shape of motion: 4d reconstruction from a single video,” 2024.

[36] R. Wu, B. Mildenhall, P. Henzler, K. Park, R. Gao, D. Watson, P. P. Srinivasan, D. Verbin, J. T. Barron, B. Poole, and A. Holynski, “Reconfusion: 3d reconstruction with diffusion priors,” *arXiv*, 2023.

[37] R. Cai, J. Y. Zhang, P. Henzler, Z. Li, N. Snavely, and R. Martin-Brualla, “Can generative video models help pose estimation?” 2024. [Online]. Available: https://arxiv.org/abs/2412.16155