Multi-View Synthesis with Dual-Tap Coded-Exposure Camera Array

Muhammad Ahmed Mansoor, Jainish Shailesh Solanki, and Karanpreet Raja

Abstract—Dynamic Multi-View Synthesis aims to reconstruct dynamic 3D scenes from 2D images. While methods like Dynamic Neural Radiance Fields (NeRFs) and 4D Gaussian Splatting have achieved photorealistic results, they rely on the assumption that scene motion is negligible during the exposure time of a single frame. When motion outpaces the exposure rate, reconstruction degrades due to motion blur. Traditional high-speed cameras can mitigate this but are prohibitively bulky, expensive, and difficult to synchronize. In this work, we propose a compact, low-cost, and low-power multi-view capture system utilizing dual-tap coded-exposure cameras. By enabling per-pixel exposure control, these cameras allow us to time multiplex the pixel array and trade spatial resolution for temporal resolution or temporal extent on demand, but without the drawbacks of traditional high-speed cameras. We present a hardware implementation of a synchronized 3-camera rig, a comprehensive evaluation of calibration techniques, and results for static novel-view synthesis for images captured from this rig. Furthermore, we develop a synthetic coded-exposure dataset and use it to study how coded-exposure high-speed imaging impacts the performance of HexPlanes, a popular Dynamic NeRF implementation. Our results show that trading off spatial resolution for more high-speed frames improves temporal interpolation capabilities, although naive spatial multiplexing introduces geometric artifacts.

Index Terms—Novel-View Synthesis, Neural Radiance Fields, Coded Exposure

1 Introduction

The reconstruction of photorealistic 3D scenes from 2D images has seen a paradigm shift with the advent of Neural Radiance Fields (NeRF) [1] and 3D Gaussian Splatting [2]. These techniques allow for free-viewpoint rendering of static scenes with high fidelity. These methods have also been successfully extended to dynamic scenes through approaches such as HexPlanes [3] and 4D Gaussian Splatting [4]. However, these representations are typically trained on standard video input (30-60 fps). Consequently, rapid motion results in blur or temporal aliasing that the reconstruction network cannot resolve, leading to loss in quality of the scene representation. While high-speed cameras can resolve this motion, deploying them in arrays is logistically complex due to bandwidth constraints, bulky form factor, power considerations, and difficulty of synchronization.

Standard camera sensors have a limited frame rate because they must integrate light over the entire pixel-array at the same time, and then go through a slow analog-to-digital conversion (ADC) and read-out cycle before they can capture another frame. High-speed cameras choose to speed-up these bottlenecks at the cost of high power consumption and high output bandwidth which lead to overall bulk and complexity. Software-defined coded-exposure sensors [5] bridge this gap in a unique way. These sensors allow per-pixel exposure control using a software programmable bit-stream. This enables us to expose a subset of the pixels and have them retain the photo-generated charge as we expose another subset, essentially dividing an exposure in time into multiple "sub-exposures" which are then all read-out after a single ADC conversion cycle. Since each sub-exposure captures the scene at a different point in time, we obtain a high-speed video sequence without going through the ADC and read-out multiple times, although at the cost of reduced spatial resolution. This technique is illustrated in Figure 1.

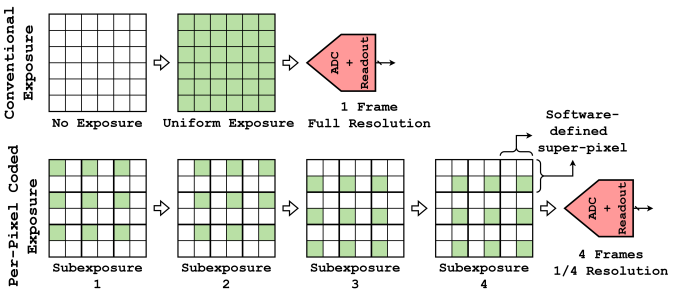

Fig. 1. An illustration of high-speed video capture using a single coded-exposure frame.

The goal of this project is to evaluate the impact of this tradeoff on dynamic novel-view synthesis. To that end, we first assemble a fixed-rig consisting of three synchronized coded-exposure cameras and demonstrate static novel-view synthesis as a proof-of-concept. We conclude that with only 3 views, the problem is under-constrained. So for dynamic scenes, we build a synthetic coded-exposure dataset with 5 views and using the HexPlane [3] dynamic NeRF implementation, find that trading off spatial resolution for more high-speed frames improves temporal interpolation results, although our analysis of spatial novel-views in HexPlane was hindered by unresolved artifacts.

- *M. A. Mansoor is with the Department of Electrical and Computer Engineering, University of Toronto.*
- *J. S. Solanki is with the University of Toronto Institute of Aerospace Studies.*
- *K. Raja is with the Department of Electrical Engineering and Computer Science, York University.*

2 Related Work

Many alternatives to traditional high-speed cameras have been suggested and implemented to mitigate the prohibitive power consumption and output bandwidth associated with standard high frame-rate imaging. Prominent technologies in this domain include event cameras, Time-of-Flight (ToF) sensors, and coded exposure cameras. Event cameras operate asynchronously, capturing per-pixel brightness changes (events) rather than full intensity frames, while ToF sensors generally focus on depth estimation by measuring the travel time of light pulses.

Coded exposure cameras, however, address the fundamental bottlenecks of conventional imaging. Standard camera sensors have a limited frame rate because they must integrate light over the entire pixel-array simultaneously, followed by a slow analog-to-digital conversion (ADC) and read-out cycle. While traditional high-speed cameras solve this by simply speeding up hardware at the cost of bulk and complexity, software-defined coded-exposure sensors [5] bridge this gap via per-pixel exposure control using a programmable bit-stream. This allows the sensor to expose pixel subsets sequentially, retaining photo-generated charge while others are exposed, effectively creating multiple "sub-exposures" within a single time window. These are read out after a single ADC conversion cycle, yielding a high-speed video sequence without the read-out bottleneck, albeit at the cost of reduced spatial resolution.

Coded exposure offers distinct advantages over its counterparts, particularly for dense 3D reconstruction. Standard NeRF pipelines rely on consistent, sharp inputs and fail catastrophically in dynamic high-speed scenes because conventional cameras necessitate long exposure times that introduce motion blur, destroying the photometric consistency required for training. Event-based approaches such as EventNeRF [6] have been proposed to overcome this by using asynchronous event streams from event cameras to train NeRFs. However, they struggle with the intrinsic limitations of event sensing, i.e. noise, challenges in robust threshold tuning, and a lack of absolute intensity (texture) information, which can lead to geometric artifacts and color drift. To the best of our knowledge, ToF or software-defined coded-exposure sensors have not yet been utilized for dynamic NeRF reconstruction.

3 Methodology

3.1 Hardware Implementation

3.1.1 Fixed Rig Assembly

Our acquisition system assembly consists of three custom coded-exposure camera modules sourced from the Intelligent Sensory Microsystems Laboratory at the University of Toronto. These have been mounted on a rigid optical breadboard, such that they have slightly different heights, and a largely overlapping field of view as shown in Figure 2.

3.1.2 Frame Rate

Ordinarily, when we use coded-exposure cameras for high-speed imaging using spatio-temporal multiplexing, we divide the total exposure into sub-exposures such that the total exposure time (i.e. sum of sub-exposure periods) remains constant. In such a case, we say that we are trading spatial resolution for temporal resolution or frame rate. However, the change in subexposure period changes the captured scene brightness drastically. Here we opt for a different approach. We set the camera exposure time to 500μs for every sub-exposure whether total or partial (and so the LED in Figure 2 is set to a fixed brightness), and we limit ourselves to only one ADC/readout cycle. So all captured frames will be high-speed (2000 FPS) but, we can trade spatial resolution for more of such frames. Thus, in this case, it is more accurate to say that we are trading spatial information for temporal information. As we will see later, this is also more consistent with the synthetic dataset we create.

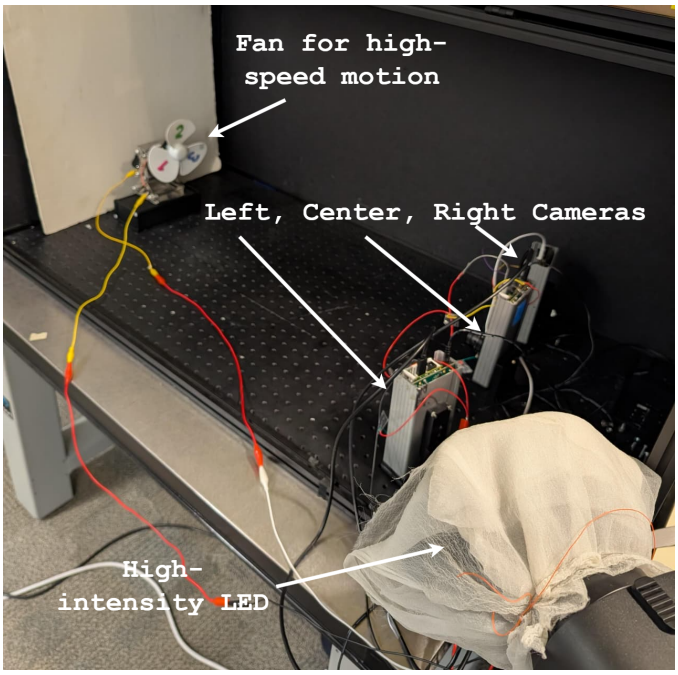

Fig. 2. Our 3-coded-exposure camera rig, with a high-intensity LED for illumination and a fan as a high-speed scene

3.1.3 FPGA Synchronization

Precise temporal alignment is critical for a stable NeRF representation, even millisecond offsets can prevent convergence in dynamic scenes. We implemented a synchronization logic on the on-board FPGAs using a single-leader/multi-follower architecture.

- **Leader:** Generates the master clock and exposure trigger codes.
- **Followers:** Listen to the trigger line and reset their internal exposure counters upon signal edge detection.

We verified this synchronization using an oscilloscope, confirming alignment within the microsecond range. A functional validation was performed by capturing a fan spinning at high RPM with 500μs sub-exposures (2000 FPS equivalent), yielding visually consistent frozen motion across views as shown in Figure 3.

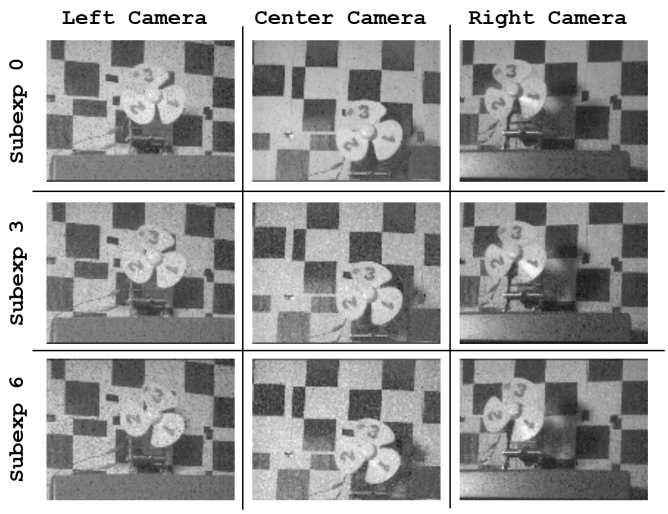

Fig. 3. Synchronized captures of a rotating fan from our coded-exposure camera rig

3.2 Camera Calibration

Accurate estimation of camera intrinsics and rig extrinsics is a prerequisite for NVS. Because the cameras used are the product of a research project, they have nonidealities like dead-pixels, lens misalignment etc. so we expected calibration to be non-trivial. Therefore, we attempted calibration using multiple techniques, and used static NVS as a benchmark to test the quality of parameters obtained from each method.

3.2.1 Method 1: Checkerboards and MATLAB

In this method, we captured checkerboards from each camera individually, and then used MATLAB camera calibration toolbox to determine intrinsics. The intrinsics so obtained have 6 parameters: f_x, f_y represent the focal length, c_x, c_y represent the principal point, and k_1, k_2 model lens distortion.

For extrinsics, MATLAB has a stereo calibration toolbox that calculates a translation vector $\mathbf{t}$ and rotation matrix $\mathbf{R}$ for one camera with respect to the other. So, we divided our rig into two stereo pairs: center-right and center-left. We captured overlapping checkerboards from each pair, and using the stereo toolbox, determined $\mathbf{t}$ and $\mathbf{R}$ for the side cameras with respect to the center camera. If we assume the center camera to be our origin, we can treat these results as our complete extrinsics.

3.2.2 Method 2: Photogrammetry Based

In this method, we used COLMAP which is a Structure-from-Motion (SfM) based photogrammetry tool. Since COLMAP requires multiple diverse views of the scene to approximate calibration parameters, we introduced a fourth, moving camera that captured a static texture-rich scene from diverse angles alongside the 3 fixed cameras. This provided COLMAP with sufficient parallax to perform robust Bundle Adjustment. We also attempted to use a second photogrammetry tool called MetaShape, but it was not able to converge.

3.2.3 Method 3: Hybrid

In this method, we provided COLMAP the results of our MATLAB calibration and allowed it to perform bundle adjustment using those as starting points. We explored 3 further possibilities:

- Using only the intrinsics from MATLAB
- Using both intrinsics and extrinsics from MATLAB

3.3 Static Novel-View Synthesis

For static NVS we used a popular off-the-shelf program called Nerfstudio, and wrote scripts to enable our custom dataset and calibration results be used with it. Nerfstudio bundles together many NVS implementations, but the two we used are:

- **Nerfacto:** A NeRF implementation that combines techniques from many published works to ensure good results for real-world scenes
- **Splatfacto:** A 3D Gaussian Splatting implementation that also combines many state-of-the-art techniques

We used both of these techniques to evaluate our calibration techniques and choose the best one. The detailed results of this comparison will be discussed later, but we concluded that with only 3 cameras, especially with all the aforementioned nonidealities, the problem of NVS is severely underconstrained. So, for dynamic NVS, we decided to create a synthetic coded-exposure dataset.

3.4 Synthetic Coded-Exposure Dataset

As a starting point, we took a sequence from the PKU-DyMVHumans [7] dataset. The original intent was to use many camera views, but because this idea was conceived much later in the project timeline, we decided to use five camera views for the sake of training time. Views from the four cameras in between each pair of the five training cameras were used as test cameras. Additionally, to test the temporal interpolation capabilities of dynamic NeRFs, we set each even frame as training data and odd frames as test data.

To simulate coded exposure, each frame was downsampled by a factor of N with a shift of one pixel between two consecutive frames. Every even-odd frame pair was downsampled in the same way since the odd pairs were to be only used for testing. Since we are simulating a single coded exposure capture, we keep only N^2 frames for training and an additional N^2 for test (so a total of $2N^2$ frames). The focal length and principal point intrinsics were also scaled by N, whereas the extrinsics remained unchanged. Figure 4 illustrates the downsampling and test-train split for $N = 2$.

3.5 Dynamic Novel-View Synthesis

For the dynamic NVS, we used the 4D NeRF implementation from the HexPlanes paper [3]. We trained HexPlanes on three different versions of our synthetic dataset, i.e. $N = 2, 3,$ and 4. For each training run, we performed two tests:

- **Temporal:** From the training cameras, have the model predict the test frames

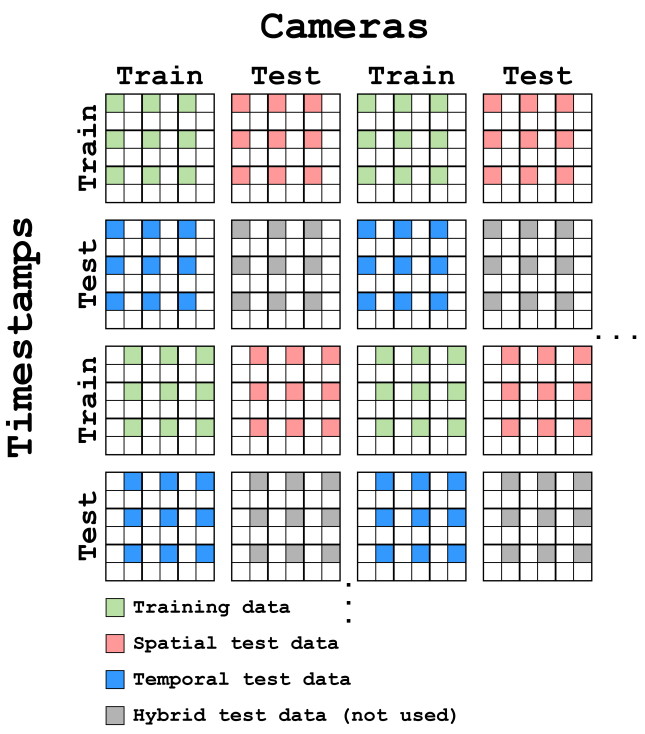

Fig. 4. Illustration of synthetic coded-exposure data creation and its test-train split for $N = 2$

TABLE 1
Results of novel-view synthesis for static scene

Calibration Method	Nerfacto PSNR (dB)	Splatfacto PSNR (dB)
MATLAB Only	N/A (No ground truth)	
COLMAP + MATLAB intrinsics & Extrinsics	Convergence failure	
COLMAP Only	14.41	18.12
COLMAP + MATLAB intrinsics	14.38	16.9

- **Spatial:** From the test cameras, have the model predict the frames corresponding to training time-stamps

Hybrid test (prediction of test time-stamps from test cameras) could not be attempted due to time constraints. Figure 4 also illustrates this classification of tests for $N = 2$.

4 Results and Discussion

4.1 Static Novel-View Synthesis

4.1.1 Results

Static NVS using the Nerfacto NeRF implementation and Splatfacto 3D Gaussian Splatting implementation was performed to compare the various calibration techniques and choose the best one. The results are summarized in Table 1.

For the MATLAB-only calibration, since no ground-truths were available, we generated a video from the trained scene representation of the viewing camera flying between the three training cameras and judged it qualitatively. We observed that if viewed from any angle other than the training cameras, the result was fuzzy and indiscernible similar to the results for MATLAB+COLMAP intrinsics as shown in Figure 5.

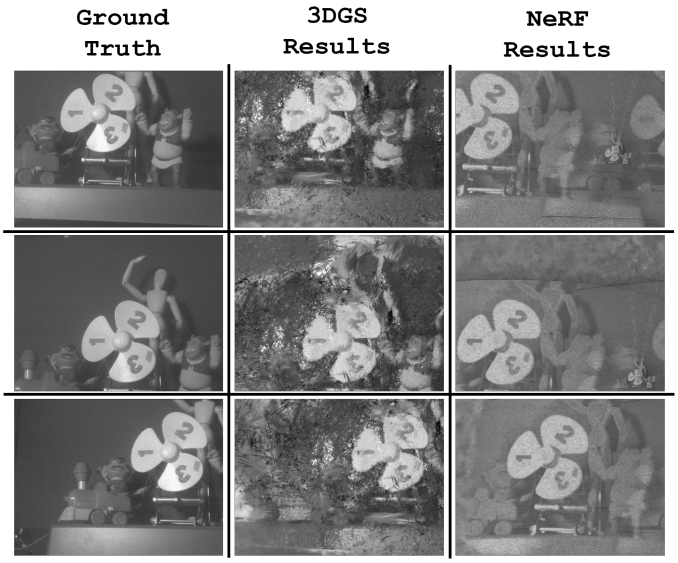

Fig. 5. Results from Nerfacto and Splatfacto when using calibration from COLMAP initialized with intrinsics from MATLAB

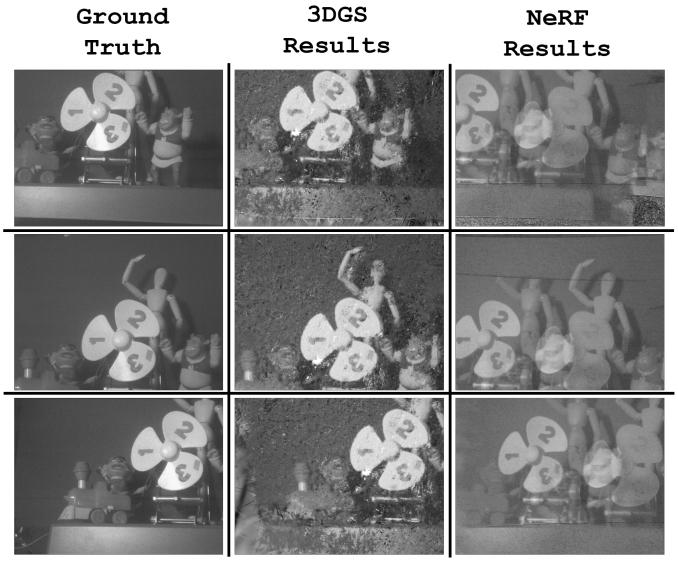

Fig. 6. Results from Nerfacto and Splatfacto when using COLMAP-only calibration

Since we captured additional views from a 4th camera for the COLMAP method, we were able to use the same as ground truths for tests and calculate PSNR. As discussed previously, we attempted to use COLMAP in 3 different ways:

- COLMAP initialized with the intrinsics and extrinsics from MATLAB failed to create a dense enough point cloud to be able to extract usable calibration data
- Using only the intrinsics from our MATLAB checkerboard analysis yielded better results as shown in Figure 5
- The best results were from COLMAP only as shown in Figure 6

4.1.2 Discussions

In both the convergent cases, 3D Gaussian Splatting produces quantitatively better results than the NeRF. Qualitatively, while the 3DGS does not have the duplication artifacts we see in the NeRF results, it suffers from the 'needle' artifacts that are characteristic of 3DGS when calibration is imperfect and training views are sparse.

We also observe that injecting MATLAB intrinsics paradoxically reduces synthesis quality. This is likely due to model mismatch: COLMAP defaults to a constrained 4-parameter "simple radial" model, which is robust for sparse data, whereas MATLAB solves for a 6-parameter model. When these complex 6-parameter intrinsics are enforced, the bundle adjustment becomes over-parameterized relative to the available views, increasing the likelihood of getting stuck in local minima.

When both intrinsics and extrinsics from MATLAB are used, COLMAP fails to converge. We attribute this to a coordinate system mismatch between the two initialization methods. Our MATLAB-derived extrinsics define the world origin at the center camera. In contrast, COLMAP's incremental mapper typically establishes a coordinate system based on an arbitrarily selected initial image pair, which in our case is statistically most likely to be from the ten images from the uninitialized moving camera. Because the MATLAB-initialized poses and the COLMAP-inferred scene structure likely differ significantly in scale, translation, and rotation, the static cameras are effectively initialized at disjoint locations relative to the reconstructed scene. This geometric inconsistency prevents the bundle adjustment from successfully converging.

The qualitative and quantitative results indicate that the current hardware configuration is insufficient for high-fidelity view synthesis. This is likely due to physical sensor defects and lens misalignments exceeding the capability of simple camera models, evidenced by physically impossible calibration artifacts like principal points diverging outside the sensor boundaries. Furthermore, with only three input views, the novel view synthesis problem remains fundamentally underconstrained. Given these significant limitations, we conclude that extending this specific physical rig to dynamic scenes is impractical. Consequently, we decided to shift our experimental focus to a synthetic dataset.

4.2 Dynamic Novel-View Synthesis on Synthetic Dataset

4.2.1 Results

We generated the synthetic dataset and performed tests for temporal and spatial reconstruction as described earlier using HexPlanes for different values of N. The quantitative results for the temporal test are shown in Table 2 and some qualitative samples are shown in Figure 7.

Unfortunately, the spatial test was marred with artifacts which could not be diagnosed and corrected in time. Figure 8 shows an example.

4.2.2 Discussion

In the temporal interpolation analysis, we observe a positive correlation between the number of sub-exposures (N) and reconstruction quality (PSNR/SSIM). This suggests that increasing the number of temporal samples enables the HexPlane model to better resolve complex motion dynamics. However, qualitative validation is hindered by a pervasive over-smoothing in the rendered outputs. This loss of high-frequency detail likely stems from training view sparsity, though it may also indicate spatial aliasing artifacts introduced by the shifting that is characteristic of our spatio-temporal multiplexing process.

Regarding the spatial interpolation failure, the root cause remains undetermined. It may be attributed to fundamental calibration, data handling, or configuration errors, or again to the shift resulting from spatio-temporal multiplexing.

TABLE 2
Quantitative comparison of reconstruction quality across different values of N.

N	PSNR	SSIM
2	28.78	0.9448
3	32.58	0.9596
4	33.05	0.9604

Fig. 7. Results from HexPlanes temporal test (zoomed-in onto the subject)

5 Conclusion

This project explored the intersection of high-speed coded-exposure imaging and neural novel-view synthesis. We successfully developed a hardware platform capable of synchronized multi-view capture to validate this approach. While calibration issues and material constraints hindered high-fidelity dynamic reconstruction on physical data, our comparative analysis identified 3D Gaussian Splatting with calibrations via photogrammetry as the most resilient pipeline for such unconstrained and nonideal hardware setups. Furthermore, experiments on our synthetic dataset demonstrate a positive correlation between the number of sub-exposures and improved temporal interpolation quality. These results suggest that increasing the quantity of temporal observations effectively enhances motion representation. However, the presence of geometric artifacts in spatial

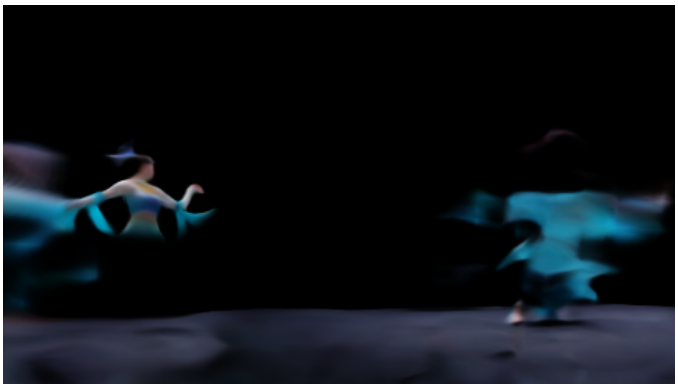

Fig. 8. Result from HexPlanes spatial test with artifacts

interpolation tests indicates that naive spatio-temporal demultiplexing of coded-exposure data introduces aliasing that current pipeline architectures cannot automatically resolve, suggesting a need for explicit ray-mask modeling in future work.

6 Future Work

6.1 Hardware Refinements

The primary bottleneck in our physical experiments was calibration instability, stemming from the imperfect mechanical assembly and the prevalence of dead pixels in the sensor array. Future iterations of this system should prioritize the fabrication of precision-machined lens and camera mounts to minimize extrinsic geometric drift. Furthermore, implementing a rigorous quality control process to select sensor modules with minimal pixel defects would significantly enhance intrinsic calibration reliability. Finally, increasing the camera count beyond the current triad would significantly alleviate the under-constrained nature of the novel-view synthesis.

6.2 Algorithmic Improvements

Our synthetic experiments revealed that spatial shifting introduces artifacts. Current NeRF and Gaussian Splatting pipelines assume a dense, consistent pixel grid. To address this, future work should integrate the coded-exposure mask directly into the ray-generation process. By explicitly modeling the mask during training, the network can be trained directly on the coded-exposure capture rather than the downsampled and shifted versions. Finally, this evaluation should be extended to 4D Gaussian Splatting [4], which may offer superior training speeds and robustness to sparse views compared to implicit field methods like HexPlanes.

Acknowledgments

The authors would like to thank Professor David Lindell for his invaluable guidance and support throughout this project.

Statement on the Use of Large Language Models

An LLM was used to rewrite some parts of this report in a more academic tone.

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," in *European Conference on Computer Vision (ECCV)*, 2020.

[2] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics (TOG)*, vol. 42, no. 4, 2023, (SIGGRAPH).

[3] A. Cao and J. Johnson, "Hexplane: A fast representation for dynamic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[4] G. Wu, T. Yi, J. Fang, L. Xie, X. Zhang, W. Wei, W. Liu, Q. Tian, and X. Wang, "4d gaussian splatting for real-time dynamic scene rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[5] R. Rangel, X. Sun, A. Barman, R. Gulve, S. Bajic, J. Wang, H. Wang, D. B. Lindell, K. N. Kutulakos, and R. Genov, "23,000-exposures/s 360fps-readout software-defined image sensor with motion-adaptive spatially varying imaging speed," in *IEEE Symposium on VLSI Technology and Circuits*, 2024.

[6] V. Rudnev, L. von Stumberg, M. Usvyatsov, A. Geiger, and M. Lambers, "Neural radiance fields from a single colour event camera," in *European Conference on Computer Vision (ECCV)*, 2022.

[7] X. Zheng, L. Liao, X. Li, J. Jiao, R. Wang, F. Gao, S. Wang, and R. Wang, "Pku-dymvhumans: A multi-view video benchmark for high-fidelity dynamic human modeling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.