Temporal-controlled Frame Swap for Generating High-Fidelity Stereo Driving Data for Autonomy Analysis

Yedi Luo[1]

Xiangyu Bai[1]

Le Jiang[1]

Aniket Gupta[2]

Eric Mortin[3]

Hanumant Singh[2]

Sarah Ostadabbas[1]
https://web.northeastern.edu/ostadabbas

[1] Augmented Cognition Lab,
Electrical and Computer Engineering
Department, Northeastern University,
Boston, MA, USA

[2] Field Robotics Lab,
Electrical and Computer Engineering
Department, Northeastern University,
Boston, MA, USA

[3] DEVCOM Analysis Center (DAC)
US Army

Abstract

This paper presents a novel approach, TeFS (Temporal-controlled Frame Swap), to generate synthetic stereo driving data for visual simultaneous localization and mapping (vSLAM) tasks. TeFS is designed to overcome the lack of native stereo vision support in commercial driving simulators, and we demonstrate its effectiveness using Grand Theft Auto V (GTA V), a high-budget open-world video game engine. We introduce GTAV-TeFS, the first large-scale GTA V stereo-driving dataset, containing over 88,000 high-resolution stereo RGB image pairs, along with temporal information, GPS coordinates, camera poses, and full-resolution dense depth maps. GTAV-TeFS offers several advantages over other synthetic stereo datasets and enables the evaluation and enhancement of state-of-the-art stereo vSLAM models under GTA V's environment. We validate the quality of the stereo data collected using TeFS by conducting a comparative analysis with the conventional dual-viewport data using an open-source simulator. We also benchmark various vSLAM models using the challenging-case comparison groups included in GTAV-TeFS, revealing the distinct advantages and limitations inherent to each model. The goal of our work is to bring more high-fidelity stereo data from commercial-grade game simulators into the research domain and push the boundary of vSLAM models [1].

[1]The simulated datasets and our code is available at https://github.com/ostadabbas/Temporal-controlled-Frame-Swap-GTAV-TeFS-. Video at: https://www.youtube.com/watch?v=mIfOBIx3HwA&ab_channel=SarahOstadabbas.
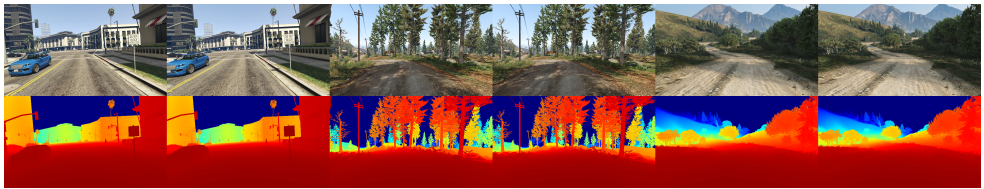
Figure 1: A few snapshots showcasing our GTAV-TeFS dataset, which includes synthetic stereo driving image pairs obtained from GTA V along with various ground truth data such as depth maps, camera poses, GPS coordinates, temporal information, etc.

1 Introduction

Stereo vision in motion is crucial for autonomous-vehicle navigation research, enabling critical visual simultaneous localization and mapping (vSLAM) tasks such as stereo odometry and up-to-scaled map creation. To enhance the robustness and performance of algorithms solving these tasks, large amounts of high-quality stereo driving data are required. Compared with expensive and hard-to-obtain real-world data, synthetic data has been increasingly used as a cost-effective and scalable solution. Popular open-source simulators, such as CARLA [6], SYNTHIA [16], and AirSim [18], provide comprehensive application programming interfaces (APIs) for stereo data collection, significantly mitigating the scarcity of stereo driving data. However, due to their nonprofit nature, their environmental complexity and texture quality are not on par with today's commercial video games. A prime example is Grand Theft Auto V (GTA V) [11], a 265-million-dollar-budget [17] open-world game with superior realism and remarkable diversity that surpasses the capabilities of most open-source simulators.

As a result, many research teams have constructed high-quality datasets from GTA V [1, 8, 9, 19], but, to date, no one has been able to provide a viable large-scale synthetic stereo-driving dataset collected from GTA V. As a commercial video game, GTA V has shortcomings in ease of data collection. Unlike open-source platforms with built-in dual-viewport stereo collection API, closed-source video games like GTA V not only have no native stereo vision support but also pose a strict single viewport limitation to prohibit rendering different camera views simultaneously.

To address this limitation and enable dynamic stereo data collection in all video games with similar restrictions, we introduce a novel method called Temporal-controlled Frame Swap (TeFS). To concretely illustrate the implementation details and effectiveness of our TeFS method, we use GTA V as our demonstration platform, and create the first large-scale GTA V-based stereo-driving dataset, GTAV-TeFS. Our dataset includes over 88,000 1920×1080 high-resolution stereo RGB image pairs along with temporal information, GPS coordinates, camera poses, and full-resolution dense depth maps. Compared to the existing GTA V-based datasets, our dataset enables the evaluation and enhancement of stereo vSLAM tasks within the GTA V environment. Compared to other synthetic stereo datasets, GTAV-TeFS also offers several advantages, including closer-to-realism environmental complexity, scenarios in both city and off-road environments, and comparison groups that feature controlled road segments under various challenging cases. A few snapshots of our GTAV-TeFS dataset is shown in Fig. 1.

Considering TeFS collection process is different from the conventional dual-viewport stereo, we also prove that stereo data collected using TeFS is of similar quality to dual-viewport data. Ideally, we would require a directly comparable dual-viewport stereo dataset

from GTA V. However, no such dataset is currently available. Fortunately, TeFS is universally applicable, so we implemented TeFS in the open-source simulator CARLA [6] to compare TeFS stereo data with native stereo data. Finally, we conduct an extensive evaluation of various vSLAM models, including both feature-based and learning-based approaches, utilizing the challenging-case comparison groups provided by our 'GTAV-TeFS' dataset. Our objective was to push the boundaries of state-of-the-art vSLAM models, effectively showcasing their distinct strengths and limitations. To summarize, the main contributions of our work are as follows:

- Introducing a novel method, Temporal-controlled Frame Swap (TeFS), to collect stereo-driving data from commercial video games with single viewport limitations.

- Creating the first large-scale stereo driving dataset, GTAV-TeFS, based on the high-fidelity video game, GTA V, which includes stereo RGB image pairs, temporal information, GPS coordinates, camera poses, and full-resolution dense depth maps, facilitating up-to-scale stereo odometry tasks in a realistic commercial game environment.

- Validating the quality of stereo data collected using TeFS by demonstrating comparable results in stereo odometry evaluations to that of traditional dual-viewport data.

- Evaluating various vSLAM models using the challenging-case comparison groups offered by our GTAV-TeFS dataset, revealing the distinct advantages and limitations inherent to each model

2 Related Work

Numerous synthetic datasets have been created from the commercial game GTA V due to its high-quality graphics and complex environment. Authors in "Play For Data" offered 25,000 video frames for training semantic segmentation systems [14], while in "Play For Benchmark", they provided over 250,000 high-resolution video frames annotated with ground-truth data for benchmarking multi-level vision tasks, such as tracking and perception. [15]. Another example is the PreSIL dataset [9], which includes 50,000 MonoRGB image frames and corresponding LiDAR point cloud data, has also proven valuable in improving the accuracy of 3D Object Detection Benchmarks. However, these GTA V datasets have a shared limitation: the lack of stereo image pairs, which impedes their use in vSLAM tasks. With only the monocular data, the scaling issue of estimated trajectories is almost inevitable. Additionally, multi-view GTA datasets like GTA SfM [22] and MVS-Synth [8] also suffer from their own shared limitations from a different perspective. While both provide multiple-angle RGB images of static scenes along with ground truth depth maps for 3D scene reconstruction, neither offers dynamic stereo image pairs in motion, rendering them unsuitable for stereo odometry tasks as well.

Apart from commercial games, open-source simulators are widely used for computer vision and autonomous driving research. One notable example is Microsoft's AirSim [18], a high-fidelity simulator that has been used to create the TartanAir dataset [23], offering rich data modalities such as stereo RGB images, depth maps, semantic segmentation masks, and camera poses across diverse environments. Similarly, CARLA [6] simulator provides realistic urban environments with numerous ground truth labels and the ability to configure stereo camera setups for autonomous driving research. One significant advantage of these

simulators is their comprehensive data collection API, as they were specifically designed for research purposes. Nonetheless, even though their data collection tools are more accessible, their geometry complexity and scene diversity do not match the level of realism provided by commercial-grade video games due to their significant budget gaps.

In conclusion, due to the limitations present in commercial games like GTA V and the reduced environmental quality of open-source simulators, there is a definite need for a universal approach that overcomes the single viewport restriction and allows for the acquisition of synthetic stereo driving data in commercial video games and driving simulators.

3   Synthetic Stereo Driving Data

This section provides a comprehensive overview of the implementation of our TeFS method in GTA V. We describe the interaction with the game environment, the stereo camera setup, the TeFS workflow, the process of obtaining depth maps, and the limitations of our approach.

3.1   Modding and Camera Setup in GTA V

Our ability to interact with GTA V largely relies on the highly developed modding community, with the most important plugin being ScriptHook V [2]. It enables us to directly interact with GTA V through API libraries such as ScriptHookVDotNet [5] and GTA V native. Information such as camera poses, GPS locations, and in-game time value are accessed through these libraries. For our camera setup, we attached a pair of parallel cameras to the front hood of our ego vehicle 'Blista', with a stereo baseline of 0.54m. We set the vertical field of view to $59°$ and the horizontal field of view to $90°$. GTA V uses a distortion shader to make the game more realistic for players [4]. However, this shader poses potential issues for research, as the camera's distortion parameters are not publicly disclosed. Therefore, we remove the distortion shader to ensure undistorted stereo data output.

3.2   TeFS: Temporal-controlled Frame Swap

A potential solution for capturing stereo data in motion from GTA V is a simple frame swap method, where the left camera is captured in the first frame and the right camera in the second. However, this method does not account for temporal disparity between the two cameras, which can be up to 16.7ms for simulators running at 60 fps, and can easily double considering the rendering engine's one frame response time. While this issue is negligible for indoor or static scenes, it can cause significant position offsets (56-112cm for a car driving at 120km/h) when collecting stereo data from a moving vehicle. As a result, the captured image pairs are unsuitable for stereo odometry algorithms.

The primary challenge is the lack of per-frame temporal control over the camera-swapping process, similar to the issue noted in Hurl's PreSIL dataset [9]. This limitation is due to the client-server architecture utilized by various GTA V-based data collection frameworks, including PreSIL, which prioritizes real-time analysis and scalability over low communication latency. To address this limitation, our proposed Temporal-controlled Frame Swap algorithm adopts a single-thread architecture that provides precise temporal control during the camera-swapping process, as shown in the flow chart in Fig. 2.

We introduce several important parameters and internal functions for our TeFS method. We use the parameter "cycleTick" to track in-game status changes during the stereo-collection
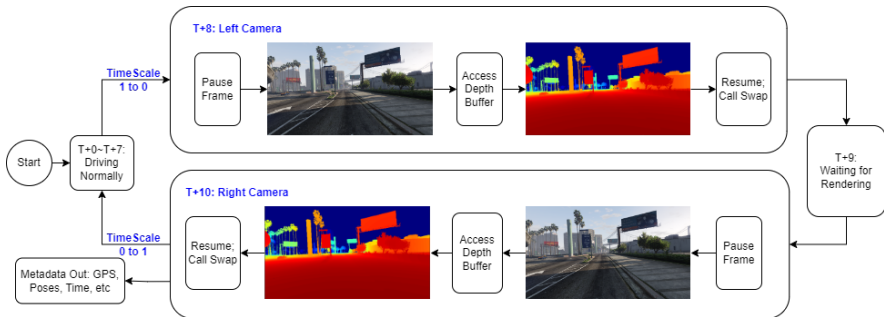
Figure 2: This flow chart outlines the Temporal-controlled Frame Swap (TeFS) algorithm for capturing synthetic stereo data from GTA V. It incorporates a single-thread architecture to precisely control the passage of time during the camera-swapping process.

cycle, which we denote as T throughout the rest of the paper. The "onTick" function is automatically called by the game engine when the script is loaded, providing a fixed time step, or "uniTick," for tracking the global progress of the game engine, regardless of the frame rate. The "timeScale" parameter controls the passage of time in the game world, ranging from 0 to 1 and beyond, with 1 representing normal game speed and 0 representing a pseudo-pause, in which the entire scene remains stationary while the rendering engine still listens to function calls. The concept of pseudo-pause is crucial to our TeFS method since we cannot swap cameras while the game is natively paused. To avoid temporal disparity, pseudo-pause is our best replacement. However, pseudo-pause is not perfect, as many animations in GTA V are hard-coded, causing scenes to appear slowed down instead of completely halted. Therefore, to get the best results, we occasionally use the native pause function as well.

According to the process outlined in Fig. 2, we aim to collect one stereo image pair every 10 uniTicks once the data collection process commences. The ego vehicle drives normally for 7 uniTicks until $T + 7$, at which point we set the timeScale parameter to 0 to achieve a pseudo-pause at the subsequent uniTick. At $T + 8$, we pause the game and capture the RGB image for the left camera. Using the GTAVisionExport [21] repository, we access the DirectX depth buffer for the corresponding RGB frame to obtain depth information. We have modified GTAVisionExport to work seamlessly with our TeFS method. This ensures that we can store depth information with true scale units rather than normalized depth shader. Afterward, we resume the game and prepare for the camera swap.

After resuming the game, GTA V remains in the pseudo-pause state since we set the timeScale to 0 before the initial pause. This allows the rendering engine to execute our camera swap function while the scene remains relatively still. At $T + 9$, we wait for the rendering engine to complete the camera-swapping process. When $T + 10$ is reached, we can natively pause the game again and repeat the process for the right camera. At the end of $T + 10$, we reset the timeScale to its normal speed, allowing the ego car to move forward and start the next data collection cycle.

3.3 Temporal Information Conversion

Capturing stereo data using the TeFS method involves a significant amount of temporal manipulation, so we cannot use real-world time to track the traversal timespan. Instead, we use GTA V's in-game time system to log each image pair's timestamp, as the in-game clock remains consistent even during significant slowdowns or native pauses throughout the data

collection process. Based on our default stereo data capture setup, there is a 2.5 second in-game time difference between consecutive stereo image pairs, which translates to approximately 0.0825 seconds in real-world time. Therefore, we can further calculate our default camera frequency, CamFreq, to be approximately 12 fps using $CamFreq = \frac{1}{T_g \times \left(\frac{D_r}{86400}\right)}$, where T_g is the in-game time passed between frames and D_r is the real-world duration of one in-game day (default value is 48 minutes or 2880 seconds).

3.4   Depth Ground Truth

Depth information retrieved using GTAVisionExport [21] cannot be used directly because the stored information is in normalized device coordinates (NDC) format [13]. It is a coordinate system used in computer graphics to represent positions on the viewport after the perspective projection and transformation have been applied. Followed by the work of Racinsky [13] and PreSIL [9], we convert the given NDC value to each pixel's actual depth value by using $Depth = \frac{Map_{uv}}{NDC + Map_{uv} \times \frac{nc_z}{2 \times fc_z}}$, where $Depth$ represents the actual depth value of one pixel, and NDC denotes the depth value in normalized device coordinates for the same pixel. nc_z and fc_z refer to the near and far clip distances of the camera, respectively. Map_{uv} corresponds to the shape of the generated images and is a 2D array of values, with each element indicating the distance to a point on the near-clipping plane for a specific pixel (u,v) in the image. Furthermore, we also vectorized PreSIL's sequential generation approach, significantly enhancing the depth conversion process.

4   Experiments and Evaluation

Our experiments are conducted for two primary reasons. First, to demonstrate that the stereo data collected using our TeFS method is of similar quality to traditional dual-viewport methods, allowing state-of-the-art vSLAM algorithms to accurately estimate stereo trajectories. Second, to illustrate the value of the challenging-case comparison groups in our GTAV-TeFS dataset by pushing the state-of-the-art vSLAM models to their limits, effectively highlighting their unique strengths and limitations. Across the experiment section, we mainly use three metrics to evaluate the performance of various algorithms: Absolute Pose Error (APE, in both meters and percentages) and Relative Pose Error (RPE). APE(m) measures the global deviation from the ground truth for each pose in the trajectory, while APE(%) helps eliminate the effect of trajectory length. RPE measures local motion accuracy, also known as drift. All metrics favor lower values.

4.1   TeFS Validation with CARLA Implementation

To validate the quality of our TeFS approach in stereo data collection and its universal applicability across various engines and games, we use the state-of-the-art open-source simulator, CARLA [6], as our data comparison platform. Mirroring the steps taken with GTA V, we implemented the TeFS method in CARLA and established a temporal disparity of 0.5ms between the left and right camera, which is the lowest temporal disparity CARLA can support. Although it's not the same as the 0.2-0.3ms temporal disparity found in GTA V, they remain comparable to each other. Afterwards, we proceeded to collect two sets of stereo data: one using TeFS, and the other using the native stereo API. For each method we captured three

subsets in separate maps, each consisting of 3600 images and yielding roughly 1km in trajectory length. With data ready, we performed stereo odometry evaluation on both datasets using ORB-SLAM3[3]. Both TeFS and native datasets illustrate excellent APE scores in all three predicted trajectory results(see Table 1). The maximum difference between two datasets' evaluation results is only 0.2%, confirming that stereo data collected via TeFS are indeed on par with those derived from native stereo API.

Table 1: Stereo odometry evaluation results of datasets collected from CARLA [6] via TeFS method and the official stereo API, using ORB-SLAM3 [3]. In all experiments, we only aligned origin with ground truth. Results of our approach are marked in bold typeface, denoting performances that rival or sometimes surpass CARLA's native method.

Map	Trajectory Distance (m)	Method	Evaluate Metrics		
			APE(m) ↓	APE(%) ↓	RPE(m) ↓
Town 03	954.41	Native	1.67±0.59	0.17%±0.06%	0.39±0.14
		TeFS	**1.82±0.65**	**0.19%±0.06%**	**0.39±0.14**
Town 06	957.97	Native	4.22±3.17	0.44%±0.33%	0.31±0.25
		TeFS	**2.30±1.67**	**0.24%±0.17%**	**0.31±0.25**
Town 10	950.05	Native	0.98±0.64	0.10%±0.07%	0.27±0.22
		TeFS	**1.01±0.63**	**0.10%±0.07%**	**0.27±0.22**

4.2 Challenging Case Evaluation for vSLAM Models

Current state-of-the-art vSLAM models achieve great performance on established datasets such as KITTI[7]. However, challenging cases such as hazardous environments, high noise, and low-light conditions still persist as obstacles for current vSLAM models. In the GTAV-TeFS dataset, we have prepared comparison data groups for three distinct road segments under various weather conditions, including one urban scenario and two off-road scenarios. Sample images can be found in Fig. 3. Each route has three corresponding weather combinations: 'Extra Sunny', 'Cloudy with Rain', and 'Night with Thunderstorm'. 'Extra Sunny' represents the basic condition,' Cloudy with Rain' serves as the moderate noise condition, and 'Night with Thunderstorm 'stands as the most challenging condition, featuring not only low contrast and high noise but also includes overexposure caused by lightning. All of the road segments incorporate loop closures, aiming to test the loop detection capabilities of each visual SLAM model. For models equipped with loop detection features, it should theoretically further enhance their evaluation results.
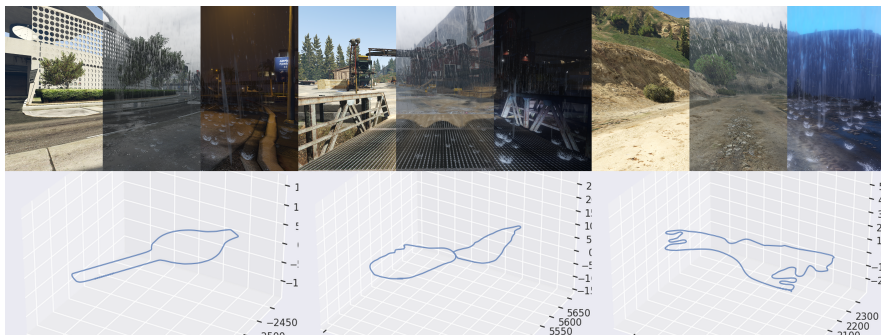

Figure 3: Comparison group samples: Three distinct road segments captured under diverse weather and lighting conditions, together with their corresponding ground truth trajectories.

Table 2: Stereo odometry evaluation results of various state-of-the-art feature-based vSLAM algorithms on the challenging-case comparison groups of our GTAV-TeFS dataset. In all experiments, we only aligned origin with ground truth.

Scene	Length(m)	Weather	Model	Metrics			
				APE(m)↓	APE(%)↓	RPE(m)↓	Loop Detected
City 04	760.0	Extra Sunny	ORBSLAM3	3.20±0.85	0.42±0.11	2.78±0.16	✓
			OV^{2}SLAM	**2.53±1.21**	**0.33±0.16**	**0.06±0.09**	✓
			VINS Fusion	2.95±1.36	0.39±0.18	0.09±0.26	✓
		Cloudy with Rain	ORBSLAM3	**2.13±1.19**	**0.28±0.16**	**0.03±0.04**	✓
			OV^{2}SLAM	21.01±9.33	2.77±1.23	0.29±2.28	✓
			VINS Fusion	3.63±2.00	0.48±0.26	0.21±0.48	✓
		Night Thunderstorm	ORBSLAM3	Lost track of the map			
			OV^{2}SLAM	**54.19±48.40**	**7.11±6.35**	**0.59±4.15**	
			VINS Fusion	Map range unreachable			
Offroad 01	1206.0	Extra Sunny	ORBSLAM3	**3.16±1.65**	**0.26±0.14**	**0.11±0.12**	✓
			OV^{2}SLAM	4.81±2.21	0.39±0.18	0.21±0.19	
			VINS Fusion	4.27±1.73	0.35±0.14	0.16±0.25	✓
		Cloudy with Rain	ORBSLAM3	**4.19±2.39**	**0.35±0.20**	**0.07±0.40**	✓
			OV^{2}SLAM	6.60±4.19	0.54±0.34	0.20±0.25	
			VINS Fusion	49.75±32.98	4.12±2.73	0.38±4.73	✓
		Night Thunderstorm	ORBSLAM3	Lost track of the map			
			OV^{2}SLAM	**91.90±45.50**	**7.57±3.79**	**0.29±2.31**	
			VINS Fusion	Map range unreachable			
Offroad 02	3230.0	Extra Sunny	ORBSLAM3	**8.19±3.95**	**0.25±0.12**	**0.39±0.26**	✓
			OV^{2}SLAM	15.47±12.69	0.48±0.39	0.36±2.07	✓
			VINS Fusion	15.44±8.99	0.48±0.39	0.16±0.21	✓
		Cloudy with Rain	ORBSLAM3	**9.45±2.77**	**0.29±0.09**	**0.16±0.14**	
			OV^{2}SLAM	85.62±65.35	2.65±2.03	0.14±0.21	
			VINS Fusion	72.47±61.96	2.26±1.93	0.22±0.70	
		Night Thunderstorm	ORBSLAM3	Lost track of the map			
			OV^{2}SLAM	**160.33±82.39**	**4.97±2.49**	**0.34±3.54**	
			VINS Fusion	Map range unreachable			

4.2.1 Stereo Odometry Evaluation On Feature-based vSLAM Models

We first undertook an evaluation of the stereo odometry capabilities of three feature-based vSLAM models, including ORB-SLAM3 [6], VINS Fusion [10], and OV^{2}SLAM[2]. As presented in Table 2, all three models show excellent performance in all road segments under the base 'Extra Sunny' weather. Notably, ORB-SLAM3 demonstrated superior robustness and accuracy in the longest road segment, Offroad 02, demonstrating 48% lower APE than that of the other two. Moving to the rainy scenario, ORB-SLAM3 consistently outperformed the other visual SLAM models, with the mean APE score sitting comfortably below 0.35% maintaining its dominant edge across all road segments. This was largely attributed to its advanced loop merging capability; while VINS Fusion and OV2 both detected loop closure, the final merged map did not match the quality produced by ORB-SLAM3.

When subjected to the most challenging scenario, 'Night with Thunderstorm', all models suffered severe degradation in performance. ORB-SLAM3 lost a significant portion of the maps, and the results of the VINS FUSION across all three road segments were unusable. Remarkably, OV^{2}SLAM was the only model that generated complete results for evaluation under these conditions, despite the high APE scores. Based on these evaluation results, we conclude that ORB-SLAM3 provides the most reliable performance and exceptional loop merging capability under normal and moderately noisy environments. Meanwhile, OV^{2}SLAM demonstrated remarkable robustness, consistently delivering comparable maps under hazardous situations.

Table 3: Evaluation results of DROID-SLAM's [20]stereo and RGB-D odometry on the challenging-case comparison groups of our GTAV-TeFS dataset. The table includes both corrected APE/RPE scores with ground truth and raw APE results. The inclusion of raw APE results emphasizes DROID-SLAM's stereo scaling issue, while providing a comprehensive evaluation of its overall performance.

Scene	Weather	Input Mode	Metrics(Corrected)			Metrics(Raw)
			APE(m)↓	APE(%)↓	RPE(m)↓	APE(m)↓
City 04	Extra Sunny	Stereo	1.27±0.58	0.23±0.12	0.63±0.14	73.69±33.71
		RGBD	**0.86±0.45**	**0.11±0.06**	0.63±0.10	**15.77±6.79**
	Cloudy with Rain	Stereo	**1.73±0.95**	**0.23±0.12**	0.63±0.14	73.75±33.65
		RGBD	2.99±1.19	0.39±0.16	**0.62±0.10**	**13.16±5.48**
	Night Thunderstorm	Stereo	6.42±2.09	0.84±0.27	0.63±0.20	74.61±33.79
		RGBD	**4.91±2.39**	**0.64±0.31**	**0.62±0.15**	**10.29±4.77**
Offroad 01	Extra Sunny	Stereo	3.61±1.15	0.30±0.10	0.58±0.19	98.68±49.18
		RGBD	**1.35±0.54**	**0.11±0.04**	0.58±0.18	**19.72±9.85**
	Cloudy with Rain	Stereo	6.23±3.66	0.51±0.30	0.59±0.18	99.44±49.29
		RGBD	**2.64±1.13**	**0.22±0.09**	0.58±0.18	**18.59±9.23**
	Night Thunderstorm	Stereo	18.24±6.11	1.50±0.50	0.62±0.31	100.32±50.09
		RGBD	**6.68±4.42**	**0.55±0.35**	0.62±0.19	**21.48±12.89**

4.2.2 Stereo and RGB-D Odometry Evaluation on Learning-based vSLAM Model

In addition to feature-based vSLAM models, we also evaluated one learning-based model, DROID-SLAM [20]. Despite its claimed stereo support, the research community has reported multiple scaling issues with its stereo trajectory estimation results. Our evaluation, as presented in Table 3, corroborates these findings. When only aligning the estimated trajectory with the origin without any scaling, the DROID-SLAM's stereo raw APE scores are excessively large, as if it is tested on monocular mode. Based on our observations over experiments, we need to match the scale of our prediction with ground truth. We first calculate scalar by dividing ground truth trajectory length with estimated length; then we multiply our estimations by the scalar. Upon scaling with ground truth, all trajectories produced by DROID-SLAM maintain the overall shape and demonstrate great accuracy. It is somewhat understandable that DROID-SLAM might encounter stereo scaling issues, considering the fact that the model was primarily trained on RGB-D data rather than stereo data.

Fortunately, our dataset also includes dense depth maps. Hence, to further substantiate our findings and assess the capabilities of DROID-SLAM, we incorporated RGB-D evaluation results alongside the stereo results. Unsurprisingly, both scaled stereo and RGB-D trajectory estimation under sunny weather conditions were readily handled by DROID-SLAM, with mean APE scores well below 0.3% for both road segments. What surprised us was its performance in rainy and thunderstorm scenarios. Like other models, DROID-SLAM's stereo results suffered a performance downgrade under high-noise and overexposure environments. However, it still manages to produce much more consistent results than any feature-based vSLAM models we evaluated. Especially in the most challenging scenarios. To be specific, in both City 04 and Offroad 01's Night with Thunderstorm scene, previously only OV^{2}SLAM was able to provide complete results with mean APE at 7.11% and 7.57%, respectively. While scaled stereo of DROID-SLAM provides mean APE at 0.84% and 1.50%, respectively. Furthermore, its RGB-D results are even more outstanding, providing accurate trajectory estimation in all rainy and thunderstorm scenarios, with mean APE well below 0.64%. However, DROID-SLAM has its own drawbacks as well, its long evaluation time and hardware requirement(11GB VRAM minimum [20]) are considerably demanding, causing some challenging and lengthy segments such as Offroad 02 in GTAV-TeFS dataset unable

to finish evaluation before our RTX 3090 GPU (24GB VRAM) shut down the process.

5   TeFS Visual Comparisons

We present visual comparisons to showcase actual differences among the resulting outputs of the TeFS, native stereo, and the simple swap method through Fig. 4. The top images compare stereo data collected using TeFS and native API in CARLA. By visual comparison, the two appear identical due to the TeFS's minimal 0.5ms temporal offset, which translates to a 2mm spatial difference at 15km/h. This offset only causes a sub-pixel difference in a 1080p image. The bottom images contrast the stereo data collected via TeFS and the simple frame swap method in GTA V. The latter would have a temporal offset of 16.7ms, which is 80 times larger than the 0.2ms temporal offset in TeFS. This significant discrepancy is causing the right output to lose important information and make depth-related tasks unfeasible.

Figure 4: TeFS vs native stereo in CARLA and TeFS vs simple swap stereo in GTA V

6   Conclusion

This paper introduces Temporal-controlled Frame Swap, a novel method designed to overcome the single viewport limitation in commercial video games like GTA V. By enabling the dynamic collection of high-quality stereo driving data, we creates the GTAV-TeFS dataset, which proves to be a valuable resource for evaluating and enhancing stereo vSLAM algorithms. The effectiveness and generalizability of the TeFS method, as well as the quality of the collected stereo data, have been scientifically analyzed and confirmed. With its precise control over in-game time, TeFS could potentially aid in the development of future surrounding-view data acquisition tools. This expansion of data collection capabilities presents exciting avenues for the advancement of future autonomy models.

7   Acknowledgement

References

[1] Xiangyu Bai, Yedi Luo, Le Jiang, Aniket Gupta, Pushyami Kaveti, Hanumant Singh, and Sarah Ostadabbas. An Evaluation Platform to Scope Performance of Synthetic Environments in Autonomous Ground Vehicles Simulation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.

[2] Alexander Blade. Scripthookv. http://www.dev-c.com/gtav/scripthookv, 2015.

[3] Carlos Campos, Richard Elvira, Juan J. Gomez, Jose M. M. Montiel, and Juan D. Tardos. ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021.

[4] Adrian Courrèges. Gta v graphics study. https://www.adriancourreges.com/blog/2015/11/02/gta-v-graphics-study/, 2015.

[5] Crosire. Scripthookvdotnet. https://github.com/crosire/scripthookvdotnet, 2015.

[6] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017.

[7] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. In *International Journal of Robotics Research (IJRR)*, volume 32, pages 1231–1237. Sage Publications Sage UK: London, England, 2013.

[8] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[9] Braden Hurl, Krzysztof Czarnecki, and Steven Waslander. Precise synthetic image and lidar (presil) dataset for autonomous vehicle perception. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 2444–2451. IEEE, 2019. doi: 10.1109/IVS.2019.8813809.

[10] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. Dtam: Dense tracking and mapping in real-time. In *IEEE International Conference on Computer Vision (ICCV)*, 2011. URL https://ieeexplore.ieee.org/document/6126513.

[11] Rockstar North. Grand theft auto v, 2013. URL https://www.rockstargames.com/V/.

[12] Tong Qin, Shaozu Cao, Jie Pan, and Shaojie Shen. A general optimization-based framework for global pose estimation with multiple sensors, 2019.

[13] Matěj Račinský. 3d map estimation from a single rgb image. Master's thesis, Czech Technical University in Prague, May 2018.

[14] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *European Conference on Computer Vision (ECCV)*, pages 102–118. Springer, 2016.

[15] Stephan R. Richter, Zeeshan Hayder, and Vladlen Koltun. Playing for benchmarks. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2232–2241, 2017. doi: 10.1109/ICCV.2017.243. URL https://doi.org/10.1109/ICCV.2017.243.

[16] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3234–3243, 2016.

[17] ScreenRant. Gta 5: How much it cost to make rockstar's open world game, 2019. URL https://screenrant.com/grand-theft-auto-5-how-much-cost-make/.

[18] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, pages 621–635. Springer, 2018.

[19] Andrew Stewart and Steven L Waslander. The ursa dataset: A large scale dataset for unmanned surface vehicle perception in coastal environments. In *2019 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4603–4609. IEEE, 2019. doi: 10.1109/ICRA.2019.8793719.

[20] Zachary Teed and Jia Deng. DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras. *Advances in neural information processing systems*, 2021.

[21] UM & Ford Center for Autonomous Vehicles (FCAV). GTAVisionExport. https://github.com/umautobots/GTAVisionExport, 2018. GitHub repository.

[22] Kaixuan Wang and Shaojie Shen. Flow-motion and depth network for monocular stereo and beyond. *IEEE Robotics and Automation Letters*, 5(2):3307–3314, 2020. doi: 10.1109/LRA.2020.2975750.

[23] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4909–4916. IEEE, 2020.