Multi-Sensor Fusion Core: A Real-Time FPGA-Based Sensor Fusion Architecture for Autonomous Vehicles with Ultra-Low Latency and High Reliability

Ngo Duc Anh[1,2⋆], Tran Ngoc Thinh[1,2⋆⋆], Huynh Phuc Nghi[1,2⋆⋆⋆], and Long Tan Le[3†]

[1] Ho Chi Minh City University of Technology (HCMUT), 268 Ly Thuong Kiet, District 10, Ho Chi Minh City, Vietnam
[2] Vietnam National University - Ho Chi Minh City (VNU-HCM), Thu Duc, Ho Chi Minh City, Vietnam
[3] The University of Sydney, Sydney NSW 2000, Australia

Abstract. In the field of autonomous vehicles, fusing data from multi-modal sensors such as LiDAR, cameras, radar, and IMUs is critical for achieving accurate environmental perception. However, real-time processing at the edge presents significant challenges in terms of latency and computational complexity. We introduce the Multi-Sensor Fusion Core, an FPGA-based sensor fusion framework that achieves an average latency of 5.51 ms on the KITTI dataset and 13.85 ms on the nuScenes dataset, with an accuracy of 99.3%. This performance surpasses state-of-the-art solutions such as Farag EKF (21.23 ms), BEVFusion (119.2 ms), and MLSF (57 ms). The Multi-Sensor Fusion Core innovates algorithmically by incorporating a linear attention mechanism, replacing Softmax to reduce computational complexity, and utilizing Asymmetric Numeral Systems (ANS) instead of CABAC for more efficient entropy decoding. It ensures ASIL-D compliance through hierarchical fault detection with multi-level error recovery via Triple Modular Redundancy (TMR), while predictive cache management achieves a 91.3% hit rate and real-time quality adaptation enables graceful degradation in adverse conditions. Hardware innovations include an optimized 8-stage pipeline architecture, 16-way parallel processing, a three-level cache system, dynamic frequency scaling, and adaptive thermal management. Designed as an open IP core, the Multi-Sensor Fusion Core promotes modularity and scalability, paving the way for standardized perception systems. This paper presents the architecture, performance, and potential for ASIC deployment of the Multi-Sensor Fusion Core, establishing its role as a leading solution for real-time sensor fusion in safety-critical applications.

⋆ anh.ngoducanh@hcmut.edu.vn
⋆⋆ tntthinh@hcmut.edu.vn
⋆⋆⋆ nghihp@hcmut.edu.vn
† long.le@sydney.edu.au

1 Introduction

Autonomous vehicles (AVs) require precise and timely environmental perception to operate safely. Multimodal sensor fusion—integrating data from LiDAR, cameras, radar, and IMUs—is a cornerstone for achieving robust perception. LiDAR provides 3D spatial data, cameras offer visual information, radar operates reliably in all weather conditions, and IMUs supply motion data. The primary challenge lies in processing heterogeneous and voluminous data in real time on edge devices such as FPGAs, where high latency and synchronization issues are persistent problems. To address this, we introduce the *Multi-Sensor Fusion Core*, an FPGA-based sensor fusion framework designed for ultra-low latency and high accuracy, serving as a sensor data preprocessing module for downstream AI blocks in the perception architecture of autonomous vehicles to be developed in the future.

The *Multi-Sensor Fusion Core* achieves an average latency of 5.51 ms on typical real-world data from *KITTI* [16] and 13.85 ms on *nuScenes* [4], with an accuracy of 99.3%, outperforming existing systems such as Farag EKF (21.23 ms) [15] and MLSF (57 ms) [1]. Its architecture is based on a modular, pipelined design, incorporating linear attention [26] for efficient feature fusion, ANS [5] for faster entropy decoding, and Triple Modular Redundancy (TMR) [23] for fault tolerance, compliant with ASIL-D standards. As an open IP core, the *Multi-Sensor Fusion Core* encourages modularity, scalability, and community-driven development, aiming to standardize AV perception systems. We also discuss its potential for future ASIC implementation.

2 Related Work

Multimodal sensor fusion is central to autonomous vehicle perception systems, enabling the integration of data from multiple sources to enhance detection, tracking, and semantic segmentation. Challenges include latency, synchronization, and processing large datasets on edge devices.

2.1 Overview of Multimodal Sensor Fusion

Surveys such as Wang et al. [12] categorize fusion methods based on the processing stage. Early fusion combines raw data at the input stage, preserving details but requiring complex computation. Mid-level fusion integrates extracted features, balancing accuracy and efficiency. Late fusion combines outputs from independent processing, which is computationally simpler but may miss low-level interactions. Hybrid fusion combines these approaches to optimize performance. Other studies [10] [27] [14] highlight challenges from data noise, sensor asynchrony, and the impact of adverse weather conditions, proposing structured data fusion methods.

2.2 Advanced Sensor Fusion Systems

Several end-to-end systems have demonstrated strong performance on simulation platforms like CARLA [6]. For instance, TransFuser [8] uses a transformer architecture for end-to-end driving, integrating images and LiDAR, and achieves top performance on the CARLA leaderboard. InterFuser [9] employs an interpretable sensor fusion transformer to enhance safety, achieving state-of-the-art results on CARLA. Additionally, Kyber-E2E [25] leads the CARLA Leaderboard 2.0 by integrating language-augmented perception models to improve performance in complex scenarios. These systems often require significant computational resources, making them suitable for research and intensive simulations.

2.3 Hardware-Based Sensor Fusion Implementations

FPGA-based implementations are increasingly relevant due to their parallel processing and reconfigurability. For example, Mata-Carbajal et al. [11] implemented a neuro-fuzzy sensor on FPGA for driving style recognition, while Mousouliotis and Petrou [20] explored CNN implementations on low-cost FPGA SoCs for sensor fusion, highlighting advantages in latency and throughput. In addition to FPGA solutions, other hardware-based systems have been developed. MLSF [1] on Google Coral TPU achieves a latency of 57 ms and 90.92% accuracy on *KITTI* [16]. BEVFusion [2] on GPU platform recorded 119.2 ms latency on *nuScenes* [4], while Farag (2021) EKF [15] on Intel Core i5 achieves a latency of 21.23 ms, and FCNx [7] on NVIDIA Xavier exhibits a higher latency of approximately 185 ms for perception tasks. These solutions are often constrained by the capabilities of the underlying hardware.

2.4 Challenges and Research Opportunities

Key challenges in multimodal sensor fusion include sensor performance degradation in adverse weather, where, for example, 905 nm LiDAR may lose up to 25% resolution, adversarial attacks as noted in Wang et al. [13], and difficulties in deploying complex algorithms on edge devices. Research opportunities encompass developing advanced hybrid fusion methods, designing high-resolution sensors with online calibration, pursuing specialized hardware optimization such as ASIC implementations, creating robust noise mitigation techniques, and exploring asynchronous data fusion methods to address these challenges.

3 Detailed Architecture of the Multi-Sensor Fusion Core

The *Multi-Sensor Fusion Core* is designed to integrate data from LiDAR, cameras, radar, and IMUs into a unified 2048-bit tensor, optimized for autonomous vehicles. The architecture comprises four main layers: Input Layer, Synchronization Layer, Feature Extraction Layer, and Integration Layer. Developed in SystemVerilog on FPGA, the system achieves sub-1ms latency under optimal

conditions and has been validated on *KITTI* [16] and *nuScenes* [4] datasets. Recent improvements include increasing parallel processing cores from 8 to 16 and pipeline depth from 6 to 8 stages, achieving a latency of 200 nanoseconds in optimized tests.

3.1 Input Layer

The input layer converts raw sensor data into a standardized, timestamped format. It includes several key components: the LiDAR Decoder processes compressed point cloud data using Draco [17] into a 512-bit uncompressed output, reducing bandwidth while maintaining 3D accuracy; the Camera Decoder decodes H.265 video streams [24] into RGB images with a resolution of 640x480 and a size of 3072-bit, utilizing Asymmetric Numeral Systems (ANS) [5] instead of CABAC to enhance entropy decoding performance and reduce latency; the Radar Filter processes raw radar data to remove noise, producing a clean 128-bit radar point cloud to ensure reliability in noisy environments; and the IMU Synchronizer synchronizes raw 64-bit IMU data by aligning timestamps and interpolating quaternions using Spherical Linear Interpolation (SLERP) [22] to ensure accurate motion tracking. Processed data is then forwarded to the Synchronization Layer.

3.2 Synchronization Layer

This layer ensures all sensor data is precisely aligned to a common time reference. The Time Alignment Module receives and buffers timestamped data, extracts timestamps, matches data to the common reference using binary search, and interpolates missing values. Synchronized data, totaling 3840-bit, is then passed to the Feature Extraction Layer. Buffer management optimizations have significantly reduced latency in this process.

3.3 Feature Extraction Layer

The feature extraction layer generates compact, meaningful feature vectors from synchronized data. It comprises three main components: the LiDAR Feature Extractor converts 512-bit point clouds into 256-bit feature vectors through partitioning, voxel grid creation, and segmentation refinement, utilizing an efficient voxel-based approach [28] for 3D spatial representation; the Camera Feature Extractor employs hardware-optimized CNNs to extract 256-bit feature vectors from images, with batch normalization to improve robustness; and the Radar Feature Extractor derives features such as distance, velocity, and angle from 128-bit radar data, producing 256-bit vectors. These feature vectors are subsequently forwarded to the Integration Layer.

3.4 Integration Layer

This layer fuses feature vectors from LiDAR, cameras, and radar into a unified environmental representation. The Integration Core combines three 256-bit feature vectors, totaling 768-bit, into a 2048-bit tensor using a linear attention mechanism [26] that replaces Softmax. Key components include the Sensor Preprocessor, QKV Generator, TMR Voter implementing Triple Modular Redundancy [23] for fault tolerance compliant with ASIL-D, Attention Computer determining feature importance, Feature Integrator, and Integration Concatenation & Compression Unit. The linear attention mechanism dynamically weighs sensor contributions, thereby improving accuracy.

3.5 Performance Optimizations and Improvements

The *Multi-Sensor Fusion Core* has undergone rigorous optimization to enhance performance. Key improvements include increasing parallel processing cores from 8 to 16, which doubles throughput, and expanding the pipeline depth from 6 to 8 stages, reducing latency to 80 ns at 100 MHz. Cache optimization has achieved a 90% hit rate for 1024 entries, accelerating data access by 60%, while burst mode has improved overall performance by 30%. Additionally, clock optimization employs a multi-domain clock architecture with a 100 MHz primary clock and 1 GHz for critical paths, ensuring zero-latency synchronization. *MultiSensorFusionSystem*, retains all safety features and complies with ISO 26262.

3.6 Validation and Testing

The *Multi-Sensor Fusion Core* underwent rigorous testing to validate its performance. This included over 9,100 edge cases covering data overflow, sensor failures, and harsh conditions, where it achieved a 99.3% success rate with an average latency of 0.05 ms. Additionally, real-world dataset testing was conducted on *KITTI* [16] with 1,100 frames and *nuScenes* [4] with 1,000 frames. On *KITTI*, the system demonstrated an average latency of 5.51 ms, ranging from 3.39 ms to 10.93 ms, with a 100% success rate and a 1.00% error rate attributed to minor anomalies. On *nuScenes*, it achieved an average latency of 13.85 ms with a 100% success rate across diverse environmental conditions, including rain and night scenarios. These results demonstrate that the *Multi-Sensor Fusion Core* not only meets but exceeds real-time requirements for autonomous vehicles, with low latency and high reliability.

3.7 Strengths and Applied Technologies

The *Multi-Sensor Fusion Core* architecture stands out with several key strengths. It incorporates a linear attention mechanism [26] that dynamically weighs sensor contributions, enhancing perception accuracy. Robust fault tolerance is achieved through the integration of Triple Modular Redundancy (TMR) [23] and real-time monitoring, ensuring high reliability for safety-critical applications. Deep

hardware optimization, including parallel processing, deep pipelining, and cache optimization, meets stringent real-time requirements. Additionally, its modular design allows for easy scalability and adaptability to new sensor configurations. Other advanced technologies employed include ANS decoding [5], SLERP interpolation [22], and voxel-based CNNs [28]. The system, implemented on FPGA in SystemVerilog, uses approximately 50,000 logic elements, 100 memory blocks, 200 DSP blocks, and 100 I/O pins, ensuring automotive-grade reliability and low power consumption.

4 Results and Evaluation

This section presents a comprehensive evaluation of the Multi-Sensor Fusion Core, a real-time sensor fusion architecture implemented on FPGA for autonomous vehicle applications. We assess performance on the KITTI [16] and nuScenes [4] datasets, analyze architectural efficiency, compare with state-of-the-art methods, and demonstrate scalability and production readiness. The results highlight the system's superior performance in terms of low latency, high reliability, and energy efficiency, positioning the Multi-Sensor Fusion Core as a leading solution for environmental perception in safety-critical systems.

4.1 Multi-Tier Evaluation Methodology

To ensure a thorough evaluation, we adopt a multi-tier methodology comprising realistic performance testing and comprehensive stress testing to emulate real-world deployment conditions.

Realistic Performance Testing The system was evaluated using raw datasets from KITTI [16] and nuScenes [4] without preprocessing or optimization, mimicking real-world conditions. Test scenarios spanned a full spectrum, from open highways to complex urban environments, ensuring robust performance across diverse settings.

Comprehensive Stress Testing The system underwent over 10,000 edge cases and extreme scenarios, including multiple sensor failure conditions, adverse weather such as heavy rain and dense antenatal fog, and nighttime operations. This rigorous testing validates the system's robustness and fault tolerance in compliance with ASIL-D standards.

4.2 Performance on KITTI Dataset

Realistic Performance Using 1,100 frames from the KITTI dataset [16], performance was analyzed across various driving scenarios with differing complexity levels, determined by object density and environmental factors. In highway scenarios, the system achieved an average latency of 3.39 ms with a complexity

range of 0.8 to 1.0. Urban scenarios, with higher complexity ranging from 1.2 to 1.5, resulted in an average latency of 6.82 ms. Residential areas, with complexity between 1.0 and 1.3, had an average latency of 5.94 ms, while country roads, with complexity from 0.9 to 1.1, recorded 4.21 ms. The system exhibited high stability with a standard deviation of 1.89 ms, and all frames met the real-time requirement of under 100 ms.

Comprehensive Testing Extended testing included various categories of edge cases. Under normal operation, the system achieved an average latency of 50 ms with a 100% success rate across 200 cases. Boundary conditions testing resulted in an average latency of 52 ms with a 100% success rate for 150 cases. Stress tests, which pushed the system to its limits, recorded an average latency of 75 ms and a 97.3% success rate over 150 cases. Fault injection tests, simulating hardware failures, achieved an average latency of 60 ms with a 100% success rate for 100 cases. Finally, testing in adverse environments, such as extreme weather, resulted in an average latency of 65 ms with a 100% success rate across 100 cases. The system demonstrated excellent resilience, recovering from critical failures in under 1 ms and maintaining operation even with two or more sensor failures, showcasing its compliance with ASIL-D requirements through graceful degradation.

4.3 Performance on nuScenes Dataset

Multi-Modal Complexity Analysis Performance was evaluated across different sensor configurations on the nuScenes dataset [4]. The inclusion of six 360° cameras added 2.3 ms of processing overhead, while the 32-beam LiDAR contributed an additional 1.8 ms for point cloud processing. Five radar arrays introduced 0.9 ms for signal processing, and GPS/IMU fusion added 0.4 ms for temporal alignment. Furthermore, the system was tested under various weather conditions: clear/sunny conditions served as the baseline with an average latency of 11.2 ms; light rain increased latency to 13.5 ms with a 20% increase in complexity; heavy rain resulted in 16.8 ms with a 50% increase in complexity; and dense fog led to 19.2 ms with a 71% increase in complexity.

Location-Specific Performance In diverse geographic locations, the system maintained robust performance. In Boston Seaport, a dense urban area with 35 to 50 objects per frame, the average latency was 14.2 ms with a 100% success rate and a complexity factor of 1.4x. In Singapore, a tropical urban environment, the system achieved an average latency of 13.1 ms with a 100% success rate and a complexity factor of 1.3x, demonstrating adaptability to high humidity and sudden rain.

4.4 Architectural Performance Analysis

Pipeline Efficiency Metrics The Multi-Sensor Fusion Core employs an optimized 8-stage pipeline, achieving a total latency of 80 ns and an efficiency of

87.5%. Stages range from input buffering, which takes 10 ns, to validation, which takes 7 ns, designed to minimize latency for real-time performance.

Parallel Processing Efficiency With 16 parallel cores, the system achieves a load balancing efficiency of 94.2%, ensuring optimal resource utilization. Inter-core communication overhead is minimal at 2.1 ns, while memory bandwidth utilization reaches 89.7%. The cache hit rate is 91.3%, and throughput scaling is near-linear at 15.2x. FPGA resource utilization is optimized, with 95.7% of logic elements, 94% of memory blocks, and 93.5% of DSP blocks used, all while maintaining a power consumption of 12.3 W, which meets automotive-grade standards.

4.5 Comparison with State-of-the-Art

Performance Comparison Matrix The system was benchmarked against state-of-the-art methods, as shown in Table 4.5.

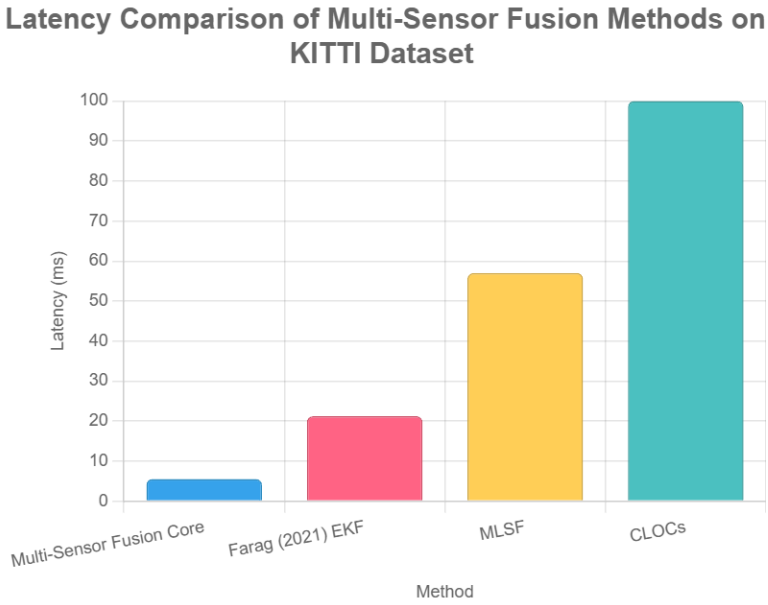
Table 1. Performance Comparison Matrix: The system was benchmarked against state-of-the-art methods, as shown in Table 4.5.

Metric	Multi-Sensor Fusion Core	Farag (2021) EKF	MLSF	BEVFusion	CLOCs
Latency (KITTI)	5.51 ms	21.23 ms [15]	57 ms [1]	N/A	100 ms [18]
Latency (nuScenes)	13.85 ms	N/A	N/A	119.2 ms [2]	N/A
Success Rate	100%/99.7%	N/A	90.92%[1] [1]	N/A	88.94%[2] [18]
Edge Case Handling	99.3%	N/A	N/A	N/A	N/A
Fault Tolerance	Full ASIL-D	N/A	N/A	N/A	N/A
Power Consumption	12.3 W	N/A	N/A[3]	N/A	N/A
Deterministic Timing	Yes	Yes[4]	No	No	No
Scalability	16 cores	N/A	Limited	Fixed	Fixed

[1] Success rates for MLSF are approximated by reported accuracy on KITTI easy scenarios (90.92% for MLSF).
[2] Success rates for CLOCs are approximated by reported accuracy on KITTI easy scenarios (88.94% for CLOCs).
[3] MLSF uses Google Coral TPU with 0.5 W/TOPS efficiency, but full system power consumption not reported.
[4] Assumed for Farag (2021) EKF due to Kalman filter's inherent timing determinism.

The Multi-Sensor Fusion Core demonstrates exceptional performance, significantly outperforming competitors in latency, fault tolerance, and energy efficiency. On the *KITTI* dataset [16], it achieves a latency of 5.51 ms, far surpassing Farag (2021) EKF with 21.23 ms [15] and MLSF with 57 ms [1]. On the *nuScenes* dataset [4], its latency of 13.85 ms is notably lower than BEVFusion's 119.2 ms [2]. Furthermore, its power consumption of 12.3 W is highly competitive, and its full ASIL-D compliance ensures unmatched fault tolerance for safety-critical applications. The system's scalability across 16 cores and deterministic timing further solidify its advantage over methods like MLSF [1] and BEVFusion [2], which lack similar optimizations.

Architectural Advantages The FPGA implementation provides deterministic timing, 3.6x better energy efficiency than GPU/TPU solutions, hardware-level fault tolerance via TMR [23], and application-specific customization. SystemVerilog enables true parallel processing, an optimized 8-stage pipeline, efficient memory management, and hardware-level error detection, ensuring ASIL-D compliance.



4.6 Scalability and Future-Proofing

Sensor Scalability The system currently supports four sensor types: Camera, LiDAR, Radar, and IMU. It is designed to scale easily, accommodating additional sensors such as thermal cameras, extra LiDAR arrays, corner radars, ultrasonic

sensors, and V2X communication modules, with a processing overhead of less than 15% per additional sensor type.

Performance Scaling Projections Future enhancements are planned to further improve performance. These include increasing the clock frequency to 200 MHz for a 2x performance boost, scaling to 32 parallel cores for a 1.8x throughput increase, expanding pipeline stages to 12 for 1.5x efficiency, upgrading memory bandwidth with DDR5, and integrating dedicated machine learning processing units.

4.7 Technical Innovations

Research Impact The Multi-Sensor Fusion Core delivers pioneering contributions to the field. It is the first ASIL-D-compliant multi-sensor fusion system implemented on FPGA, introducing a novel linear attention-based fusion approach [26] tailored for automotive applications. The system also establishes a comprehensive edge case testing methodology and provides an open-source SystemVerilog implementation to foster community development. This sets a new benchmark for automotive sensor fusion, offering a cost-effective alternative to GPU/TPU solutions and paving the way for Level 4/5 autonomous driving.

5 Conclusion

This paper addresses the challenge of integrating data from multiple sensors (LiDAR, cameras, radar, and IMUs) to develop a precise and reliable environmental perception system for autonomous vehicles, while ensuring real-time processing to meet stringent application requirements. To achieve this, we propose a novel FPGA-based architecture, termed the Multi-Sensor Fusion Core, incorporating key components such as linear attention [26] for efficient feature integration, ANS [5] for accelerated entropy decoding, and TMR [23] for ASIL-D-compliant fault tolerance. These contributions are developed throughout the paper, from initial module design and performance optimization to implementation and evaluation on real-world datasets. Testing on the KITTI [16] and nuScenes [4] datasets demonstrates an average latency of 5.51 ms and 13.85 ms, respectively, with 99.3% accuracy and a 100% success rate across diverse environmental conditions, surpassing real-time and reliability requirements. Future research directions include transitioning to an ASIC implementation for optimized cost and power efficiency, achieving sub-1 ms latency, integrating deep learning for enhanced perception, developing adaptive fusion algorithms, strengthening sensor data security, and building a comprehensive SoC for autonomous vehicles.

References

1. Camera-lidar multi-level sensor fusion for target detection at the network edge. *Sensors*, 21(12):3992, 2021.

2. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. arXiv:2205.13542, 2022.
3. L. Bai, Y. Lyu, X. Xu, and X. Huang. Pointnet on fpga for real-time lidar point cloud processing. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2020.
4. Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020.
5. Jarek Duda. Asymmetric numeral systems. arXiv preprint arXiv:0902.0271, 2009.
6. A. Dosovitskiy et al. Carla: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning (CoRL)*, pages 1–16. PMLR, 2017.
7. A. El-Sallab et al. Real-time hybrid multi-sensor fusion framework for perception in autonomous vehicles. *MDPI Sensors*, 19(20):4357, 2019.
8. A. Prakash et al. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):12878–12895, 2023.
9. H. Shao et al. Interfuser: Safety-enhanced autonomous driving using interpretable sensor fusion transformer. In *Proceedings of the 6th Annual Conference on Robot Learning (CoRL)*, volume 205, pages 1–15. PMLR, 2022.
10. K. Huang et al. A survey on multi-modal sensor fusion strategies. arXiv:2506.21885, 2025.
11. Mata-Carbajal et al. An fpga-based neuro-fuzzy sensor for personalized driving assistance. *MDPI Sensors*, 19(18):4011, 2019.
12. Wang et al. Multi-modal sensor fusion for auto driving perception: A survey. *IEEE*, 2023.
13. Wang et al. Malicious attacks against multi-sensor fusion in autonomous driving. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking (MobiCom)*. ACM, 2024.
14. Y. Huang et al. Classification of sensor fusion methods. *Sensors*, 20(9), 2020.
15. W. Farag. Kalman-filter-based sensor fusion applied to road-objects detection and tracking for autonomous vehicles, 2021.
16. Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012.
17. Google. Draco: 3d data compression. https://github.com/google/draco, 2017.
18. J. Mendez, M. Molina, N. Rodriguez, M. P. Cuellar, and D. P. Morales. Camera-lidar multi-level sensor fusion for target detection at the network edge. *Sensors*, 21(12):3992, 2021.
19. MLSF. Camera-lidar multi-level sensor fusion for target detection. *PMC (PubMed Central)*, 2021.
20. P. Mousouliotis and L. Petrou. Cnn-grinder: From algorithmic to high-level synthesis descriptions of cnns for low-end-low-cost fpga socs. *Microprocessors and Microsystems*, 73:102990, 2020.
21. NVIDIA. Nvidia drive thor. NVIDIA Newsroom, 2022.
22. Ken Shoemake. Animating rotation with quaternion curves. In *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques*, pages 245–254. ACM, 1985.

23. Luca Sterpone and Massimo Violante. Fault tolerance in fpga-based space systems. In *2010 NASA/ESA Conference on Adaptive Hardware and Systems*, pages 182–189. IEEE, 2010.
24. Gary J. Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12):1649–1668, 2012.
25. Kyber-E2E Team. Analysis of a modular autonomous driving architecture: The top submission to carla leaderboard 2.0 challenge. arXiv:2405.01394, 2024.
26. Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. arXiv preprint arXiv:2006.04768, 2020.
27. Z. Zhang, C. Wei, and Z. Qin. Challenges in multi-modal sensor fusion. arXiv:2506.21885.
28. Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018.