

UC San Diego

UC San Diego Electronic Theses and Dissertations

Title

Dynamic Vision: Unlocking Motion Insights and Multimodal Event Creation

Permalink

<https://escholarship.org/uc/item/0x4490jv>

Author

Zhang, Zhongyang

Publication Date

2025

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA SAN DIEGO

Dynamic Vision: Unlocking Motion Insights and Multimodal Event Creation

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy

in

Data Science

by

Zhongyang Zhang

Committee in charge:

Professor Tauhidur Rahman, Chair
Professor Gert Cauwenberghs
Professor Haojian Jin
Professor Hava Siegelmann
Professor Edward Wang

2025

Copyright

Zhongyang Zhang, 2025

All rights reserved.

The Dissertation of Zhongyang Zhang is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2025

DEDICATION

To those who stood beside me when the path was steep:
My parents, who held me up;
My advisor, who lit the way;
And my friends, who made the climb worthwhile.

EPIGRAPH

If you just wait, the only thing that happens
is that you get older.

Haruki Murakami, *Dance Dance Dance*

What breaks fate isn't miracles — it's human will.

Touma Kamijou, *A Certain Scientific Railgun*

Even the most mundane of days can become interesting,
if you look at them closely enough.

Houtarou Oreki, *Hyouka*

TABLE OF CONTENTS

Dissertation Approval Page	iii
Dedication	iv
Epigraph	v
Table of Contents	vi
List of Figures	ix
List of Tables	xi
Acknowledgements	xii
Vita	xviii
Abstract of the Dissertation	xx
Introduction	1
Chapter 1 Neuromorphic High-Frequency 3D Dancing Pose Estimation in Dynamic Environment	6
1.1 Introduction	6
1.2 Related Work	10
1.3 Design Consideration	13
1.4 Proposed System	16
1.4.1 Event Preprocessing and TORE Volume	16
1.4.2 Event Filtering with Human Body Mask Prediction Network	17
1.4.3 Human Pose Estimation Network	19
1.4.4 Loss	21
1.5 Synthetic Data Generation with Motion-to-Event Simulator	22
1.5.1 Advantages of Synthetic Data	23
1.5.2 Tools used in the simulator	24
1.5.3 Comprehensive Motion-to-Event Simulator	25
1.5.4 Synthetic Dataset Description	26
1.6 Human Subject Study with Dynamic Lighting and Background	27
1.6.1 Real-World Dataset in Motion Capture Facility	27
1.7 Results	29
1.7.1 Training Details	29
1.7.2 Evaluation Metrics	30
1.7.3 Evaluation	32
1.7.4 Result Statistics and Analysis	33
1.7.5 Impact of Pretraining on Synthetic Data	35

1.7.6	Ablation Study	35
1.7.7	Occlusion	36
1.7.8	Systems Benchmarking	37
1.8	Comparison between Different Modalities	39
1.8.1	RGB Camera	39
1.8.2	RGB-Depth Camera	39
1.9	Limitation and Future Works	43
1.10	Conclusion	44
Chapter 2	V2CE: Video to Continuous Events Simulator	45
2.1	Introduction	45
2.2	Proposed System	48
2.2.1	Stage1: Motion-Aware Event Voxel Prediction	48
2.2.2	Stage2: Voxels to Continuous Events Sampling	52
2.3	Experiments and Results	54
2.3.1	Stage1: Motion-Aware Event Voxel Prediction	54
2.3.2	Stage2: Voxel to Continuous Event Sampling	59
2.4	Conclusion	61
Chapter 3	Labits: Layered Bidirectional Time Surfaces Representation for Event Camera-based Continuous Dense Trajectory Estimation	63
3.1	Introduction	63
3.2	Related Works	66
3.3	Labits: Layered Bidirectional Time Surface	67
3.4	Method	71
3.4.1	Labits-to-APLOF Net	71
3.4.2	Pipeline	74
3.5	Results	76
3.6	Conclusion	81
3.7	Supplementary Material	81
3.7.1	Model Specifications	81
3.7.2	Instantaneous Speed Prediction Error Estimation Analysis	82
3.7.3	Analysis of the Bidirectional Structure in Labits	83
3.7.4	Ablation Study on Time Bin Size	84
3.7.5	Additional Comparison Experiments	85
3.7.6	Limitations and Future Directions	86
Chapter 4	Larver: Learning Alignment and Representation for Versatile Event Regeneration	93
4.1	Introduction	93
4.2	Related Work	95
4.3	Proposed Method	97
4.3.1	Overview	97
4.3.2	Representation Learning: FLER	98

4.3.3	Unified Embedding for Events	100
4.3.4	Event Generation via Diffusion Models	100
4.3.5	Reconstruction of Event Voxel Grids	101
4.3.6	Summarization	102
4.4	Conclusion & Future Work	102
Chapter 5	Discussion and Future Work	104
5.1	Summary of Dissertation	104
5.2	Event-base Vision and Real-time Applications	106
5.3	Event-based Vision and Human Computer Interaction	107
5.4	Generalizability in Event-based Vision	109
5.5	Event Representations	110
5.6	Areas of Future Work	111
	Bibliography	112

LIST OF FIGURES

Figure 1.1.	The schematic representation of the TORE volume.	16
Figure 1.2.	The pipeline of the proposed YeLan.	18
Figure 1.3.	Synthetic data generation pipeline.	24
Figure 1.4.	Settings of human subject data collection in an indoor laboratory setting. .	25
Figure 1.5.	RGB frame samples from the synthetic data generated by the comprehensive motion-to-event simulator.	27
Figure 1.6.	RGB frame samples from the Yelan-Real-Dataset collected in motion capture laboratory.	29
Figure 1.7.	Sample results from Yelan-Syn-Dataset in different lighting conditions with dynamic backgrounds.	31
Figure 1.8.	The performance across lighting conditions, dynamic backgrounds, subjects, and camera views in our synthetic data.	33
Figure 1.9.	Impact of lighting condition on the performance (MPJPE) on our synthetic (left) and real-world (right) dataset.	34
Figure 1.10.	AUC curve with dynamic and static background on Yelan-Real-Dataset. ...	34
Figure 1.11.	Samples of occluded inputs, predicted mask, TORE volume representation, and inferred human poses.	36
Figure 1.12.	Time-saving and performance rising percentage with regard to confidence score threshold.	37
Figure 1.13.	Joint-wise average MPJPE analysis on Yelan-Syn-Dataset.	38
Figure 1.14.	HPE comparison between RGB and DVS in both high and low-lighting conditions.	40
Figure 1.15.	2D MPJPE Comparison between DVS and RGB.	40
Figure 1.16.	Stable human pose estimation of the neuromorphic camera on different FPS settings.	41
Figure 1.17.	Qualitative comparison between the RGB-Depth camera and YeLan in terms of the distance between the camera and the human subject.	43
Figure 2.1.	K-nearest neighbor graph comparison from events in (x, y, t) space.	46

Figure 2.2.	Proposed motion-aware event voxel prediction pipeline and hybrid loss. . .	47
Figure 2.3.	Visualization for two stages in LDATI.	52
Figure 2.4.	Event frame comparison between V2CE and baseline methods	55
Figure 2.5.	Zoom-in comparison on event frame details.	57
Figure 2.6.	Voxel Cube comparison.	58
Figure 2.7.	Qualitative comparison between all voxel-to-event sampling methods given a randomly generated event stream.	60
Figure 3.1.	Labits generation schematic and Labits’ comparison with other event representations.	65
Figure 3.2.	Labits-RAFT and Labits-to-APLOF Net architecture.	74
Figure 3.3.	Visualization of detailed inputs and outcomes from our model.	77
Figure 3.4.	Trajectory predictions on the MultiFlow dataset by our proposed model and baseline methods.	79
Figure 3.5.	Visual comparison of Labits and one-way Labits.	82
Figure 3.6.	Comparison of trajectory predictions.	89
Figure 3.7.	Comparison of optical flow estimations.	90
Figure 3.8.	Visualization of intermediate prediction results.	91
Figure 3.9.	Comparative visualization of event representations.	92
Figure 4.1.	Pipeline and the FLER definition loss set of stage 1.	97
Figure 4.2.	Pipeline of the stage 2 and 3.	98

LIST OF TABLES

Table 1.1.	A comparison between existing 3D HPE datasets and our collected datasets.	9
Table 1.2.	Test set results comparison on three metrics for Yelan-Syn-Dataset and Yelan-Real-Dataset.	32
Table 1.3.	The comparison between RGB, RGBD, and neuromorphic cameras. For FOV, H, V, and D stand for horizontal, vertical, and diagonal FOV.	41
Table 2.1.	The quantitative comparison between our method and all baseline methods on the testing set.	56
Table 2.2.	The ablation study results for all our proposed losses.	58
Table 2.3.	Testing set event voxel-to-event stream results comparison.	60
Table 2.4.	Testing set video-to-event stream results comparison.	61
Table 3.1.	A comparison of event representation methods used in deep learning.	66
Table 3.2.	Results on MultiFlow dataset.	78
Table 3.3.	DCT-RAFT and Labits-RAFT model size and performance metrics	81
Table 3.4.	Final APLOF losses comparison between Labits and one-way Labits.	84
Table 3.5.	Performance comparison of Labits and one-way Labits.	84
Table 3.6.	Performance comparison of DCF-RAFT and Labits-RAFT using different bin setups.	84
Table 3.7.	Performance Comparison of Labits-to-APLOF and Voxel-to-APLOF Net. .	85
Table 3.8.	Ablation study results on Labits and voxel-to-APLOF Net.	85
Table 3.9.	Comparison of more recent layered event representations applied to the RAFT architecture for trajectory estimation.	86

ACKNOWLEDGEMENTS

Nothing is more satisfying than the clarity of hindsight—like right now, as I’m about to look back on my six-year PhD journey and casually remark, “Hey! Getting a PhD wasn’t so tough after all!”

Almost before realizing it, the ending theme song has started to play. Thinking back to my middle school self—eagerly presenting hastily derived conclusions to my family as if unveiling some groundbreaking scientific discovery—I feel a gentle pang of nostalgia. Back then, “I want to be a scientist” was the common aspiration among many children, and I was certainly no exception. Though it would be premature to confidently label myself a “scientist” today, I genuinely take pride in what I’ve accomplished and the things I’m about to achieve. Dreams are always worth having—what if they actually come true?

Humming the melodies of grand dreams, making one decision after another, navigating countless days and nights filled with diverse experiences—this is how I’ve arrived where I am today. Looking back, the path I traveled was filled with unpredictability and uncertainty, and even now, revisiting some of those pivotal moments leaves me in awe of life’s serendipity. Yet, beyond thanking the divine, I especially want to extend my heartfelt gratitude to the numerous mentors and companions who have graced my journey, transforming dreams into reality and turning daunting obstacles into bridges I could cross.

Foremost, I wish to thank my PhD advisor, Professor Tauhidur Rahman. Throughout these years of studying and working together, he taught me invaluable lessons and provided me with courage and strength. Not only did he guide me in research—teaching me how to identify problems and find solutions—but he also became an exemplary role model in life. He always greeted students with a warm, comforting smile, and whenever we encountered difficulties, whether academic or personal, he was invariably the first to step forward with genuine concern and practical assistance. When I first arrived at UMass Amherst and realized my research interests did not align with my initially chosen advisor, only Professor Rahman stepped up, offering me a position in his lab. He went to great lengths to help me switch programs, giving my

PhD journey a renewed sense of hope. It's no exaggeration to say that Professor Rahman was my savior—without his support at that critical moment, I wouldn't have completed my PhD. Later, when he received a new job offer from UCSD, he did not abandon his existing students like many other professors might have done. Instead, he patiently guided all lab members wishing to transfer alongside him, facilitating a complicated and demanding relocation process. Rather than merely being an advisor, he has been like a father figure to me. For Professor Rahman, no amount of gratitude could ever truly be enough.

Next, I would like to thank the two prestigious universities and their respective departments where I spent my PhD journey. During my initial transfer from the ECE department to the College of Information and Computer Sciences (CICS) at UMass Amherst, both deans and the PhD advisors from these two colleges offered invaluable support. They patiently helped me navigate through complexities, laid out all feasible solutions, and ultimately facilitated my transition into Professor Rahman's lab. Their supportive attitudes and generous policies laid the foundation for many happy memories. Similarly indispensable was the support from UCSD and the Halicioğlu Data Science Institute (HDSI) during my subsequent move from UMass Amherst to UCSD. At the time, HDSI had just begun recruiting PhD students, and many procedures and policies were still under development. It was HDSI's PhD advisor Kimber Worthy, along with many other staff members, who ensured a smooth and seamless transfer for all of us. Even after officially joining UCSD, Kimber continued to solve countless issues for me, effortlessly unraveling every complex problem—she truly seemed omnipotent! Their help made my years in San Diego exceptionally wonderful and irreplaceable.

I also wish to sincerely thank my labmates: Bhanu Gullapalli, Yunfei Luo, Md Farhan Tasnim Oshim, Manasa Kalanadhabhatta, and Forsad Al Hossain, particularly my senior colleagues. As the first cohort of PhD students in our lab, they set great examples for me and provided tremendous support both academically and personally. I still vividly remember the morning when I suffered a sudden acute gastric perforation, in excruciating pain and extremely weakened, urgently needing medical attention. Farhan immediately rushed me to the hospital,

and Forsad soon arrived to help facilitate my transfer for emergency surgery. Emergencies can occur abruptly and unexpectedly, but these kind-hearted individuals pulled me back from the brink, quite literally saving my life. For this, I feel endless gratitude toward them. During the time of surgery and recovery, other friends from afar in China, such as Xiaoyue Zhang and Ying Wang, also expressed their sincere concern and shared valuable knowledge that greatly aided my recovery. Their support made the hospital stay feel less arduous and much warmer.

In addition to my seniors, I also want to sincerely thank all the graduate and undergraduate students with whom I have collaborated on various projects. Jiacheng Qiu, Shuyang Cui, Kaidong Chai, Haowen Yu, and Zhiyang Xu provided immense assistance on many projects related to event-camera vision and hyperspectral vision. Additionally, Jinhe Wen and Zixi Chen contributed significantly to the gaming project, dedicating countless hours as we painstakingly refined our project together. We brainstormed, experimented, revised manuscripts, and fought alongside each other on the battlefield of research. I recall vividly the nights we stayed up together, connected by calls and working tirelessly until the very last minute of paper deadlines. Each time we successfully submitted our work, we felt an incomparable sense of satisfaction. Regardless of the eventual outcomes, I am profoundly thankful to all my colleagues who fought alongside me. It is because of you all that each of these achievements became possible; because of you, my PhD life was made complete. You are more than just academic collaborators—you are friends whom I hold dearly.

Moreover, I'm deeply grateful to the professors who mentored or collaborated with me on various projects: Professors Asif Salekin, Upal Mahbub, Hava Siegelmann, Haojian Jin, Edward Wang, and others provided significant guidance in deep learning and human-computer interaction. Professor Adam Grabell generously shared invaluable domain expertise on integrating physiology in preschool children with computer vision, and Bijan Arbab contributed significant insights in the video game domain. Throughout my PhD, challenges and obstacles were ever-present. Yet these respected professors served as beacons in darkness, illuminating paths forward. Often, even the faintest glimmer of light can rescue a lost, desperate traveler.

I also extend my heartfelt thanks to colleagues, mentors, and managers who supported me during my internships, offering insights into the world beyond academia. Your enthusiasm, supportiveness, and helpfulness filled me with admiration for the industry, inspiring me to venture forth and apply my knowledge meaningfully. Specifically, I want to thank my mentor Keli Cheng at Qualcomm, whose patience, respectfulness, and collaborative spirit provided me an unforgettable internship. Additionally, I greatly appreciate Yuda Feng for recommending my internship at Samsung Research America. There, I was fortunate to meet the energetic, knowledgeable, and responsive manager Hao Chen, whose timely guidance, quick responses to various needs, and exciting research ideas greatly enriched my understanding of large language models and culminated in a deeply satisfying project. Both internships blessed me with warm, loving, and wise colleagues, filling my post-graduation outlook with optimism.

Furthermore, I want to extend my heartfelt thanks to my friends who have made my PhD life extraordinarily colorful and enriched my spirit throughout these years. Those relaxing weekend meals and joyful conversations with Hongyu Lu, Mila Tan, and Wen-Chin Liu, greatly relieved my stress. Together, we explored every corner of San Diego, indulging in diverse cuisines; together, we passionately sang anime classics with more emotion than accuracy, songs filled with our collective memories; together, we etched unforgettable moments into our lives during our journey to the Grand Canyon. Although after this graduation, we are finally parting ways to different corners of the world, I sincerely hope that someday, when we gather again, we can once more belt out “God Knows” to the strumming of your guitar.

In addition to Hongyu and the others, I’d also like to express my gratitude to Yingyu Lin, Pengrun Huang, and Wenqin Liu. Although our acquaintance did not begin early, in the final year before graduation, we’ve ventured to many remarkable places. We caught sunsets by the ocean in the twilight hours, chased the glowing bioluminescent tides and the dazzling Comet streaking across the night sky, immersed ourselves in “Les Misérables” at the theatre, and captured nature’s stunning wonders in Arizona. I have always been somewhat introverted, but you all have painted my monotonous canvas with vibrant colors, making UCSD more than just

another UC, but a place where I truly experienced the beauty of San Diego.

Other than the friends I met in San Diego, I'm equally unable to forget the wonderful companions I made during my time at UMass Amherst. Minhao Cui, Zhiyang Xu, Yongzhi Xiao, Stephen Chung, Chaofan Jiang—it was our weekly Friday night Smash Bros tournaments that I looked forward to the most during that years. Hotpot bubbling, controllers clicking, our single Nintendo Switch eventually multiplied to one in each person's hands. No matter how time passes, those thousand-plus hours of shared memories will forever be our bond.

Beyond gaming, the anime enthusiasts I befriended at UMass Amherst also gave me uniquely precious memories. We crammed together to watch Evangelion movies and “Liz and the Blue Bird,” spent time playing rhythm games in the arcade, hiked and picnicked in the mountains, and even flew from all over the country to attend Anime Expo in Los Angeles afterwards. These moments, some raw and some delicate, some distinctly Japanese, and others purely American, remain deeply special to me. Hanling Guo, Jikun Wang, Yanliang Chen, Yikang Gong, Kaidong Chai, Zhenyu Lei—thank you all for helping me live a Japanese-inspired dream in America, for making this niche passion something we could pursue and share together. Hontouni, Arigatou.

Finally, I hope to conclude with the deepest gratitude to my family. Ever since I chose the challenging path of pursuing a PhD, no matter the obstacles, my parents have comforted me, analyzed situations with me, and unwaveringly supported my decisions. Their love and trust consistently inspire confidence and inner strength. I just want to say: I love you!

To everyone who has walked this path with me—whether for a long time or just a short while—thank you. You have all made this journey extraordinary.

Chapter 1, in full, is based on the paper from Zhongyang Zhang, Kaidong Chai, Haowen Yu, Ramzi Majaj, Francesca Walsh, Edward Wang, Upal Mahbub, Hava Siegelmann, Donghyun Kim, and Tauhidur Rahman, “Neuromorphic high-frequency 3D dancing pose estimation in dynamic environment” published in *Neurocomputing*, Volume 547, 2023, 126388, ISSN 0925-2312. The dissertation author is the primary investigator and author of this paper.

Chapter 2, in full, is based on the paper from Zhongyang Zhang, Shuyang Cui, Kaidong

Chai, Haowen Yu, Subhasis Dasgupta, Upal Mahbub, and Tauhidur Rahman, “V2CE: Video to Continuous Events Simulator” published in 2024 IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 2024, pp. 12455-12461. The dissertation author is the primary investigator and author of this paper.

Chapter 3, in full, is based on the paper from Zhongyang Zhang, Jiacheng Qiu, Shuyang Cui, Yijun Luo, and Tauhidur Rahman, “Labits: Layered Bidirectional Time Surfaces Representation for Event Camera-based Continuous Dense Trajectory Estimation” submitted for publication in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2025. The dissertation author is the primary investigator and author of this paper.

Chapter 4, in part, is based on the paper from Zhongyang Zhang, Jiacheng Qiu, Gert Cauwenberghs, Upal Mahbub, and Tauhidur Rahman, “Larver: Learning Alignment and Representation for Versatile Event Regeneration” currently being prepared for submission. The dissertation author is the primary investigator and author of this paper.

VITA

- 2019 Bachelor of Engineering, Huazhong University of Science and Technology
- 2019–2020 PhD Student of Electrical and Computer Engineering, University of Massachusetts Amherst (Transferred to the Computer Science Department)
- 2020–2022 PhD Student of Computer Science, University of Massachusetts Amherst (Transferred to University of California San Diego)
- 2019 Research Assistant, University of Massachusetts Amherst
- 2020–2021 Teaching Assistant, University of Massachusetts Amherst
- 2022 Research Assistant, University of Massachusetts Amherst
- 2022–2025 Research Assistant, University of California San Diego
- 2025 Doctor of Philosophy in Data Science, University of California San Diego

PUBLICATIONS

- [1] **Z. Zhang**, J. Qiu, S. Cui, Y. Luo, and T. Rahman, “Labits: Layered Bidirectional Time Surfaces Representation for Event Camera-based Continuous Dense Trajectory Estimation,” Dec. 12, 2024, arXiv: arXiv:2412.08849.
- [2] **Z. Zhang**, J. Wen, Z. Chen, D. Arbab, S. Sahani, W. Lewis, K. Giard, B. Arbab, H. Jin, and T. Rahman, “Predicting Quality of Video Gaming Experience Using Global-Scale Telemetry Data and Federated Learning,” Dec. 21, 2024, arXiv: arXiv:2412.08950.
- [3] **Z. Zhang**, S. Cui, K. Chai, H. Yu, S. Dasgupta, U. Mahbub, and T. Rahman, “V2ce: Video to continuous events simulator,” in 2024 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2024, pp. 12455–12461.
- [4] J. Wen, **Z. Zhang**, T. M. Tran, L. Mu, T. Rahman, and H. Jin, “Folk Models of Loot Boxes in Video Games,” Proc. ACM Hum.-Comput. Interact., vol. 8, no. CHI PLAY, pp. 1–23, Oct. 2024.
- [5] **Z. Zhang**, K. Chai, H. Yu, R. Majaj, F. Walsh, E. Wang, U. Mahbub, H. Siegelmann, D. Kim, and T. Rahman, “Neuromorphic high-frequency 3D dancing pose estimation in dynamic environment,” Neurocomputing, vol. 547, p. 126388, 2023.
- [6] Y. Xiao, H. Sharma, **Z. Zhang**, D. Bergen-Cico, T. Rahman, and A. Salekin, “Reading Between the Heat: Co-Teaching Body Thermal Signatures for Non-intrusive Stress Detection,” Proc. ACM Interact. Mob. Wearable Ubiquitous Technol., vol. 7, no. 4, pp. 1–30, Dec. 2023.
- [7] M. Kalanadhabhatta, A. M. Santana, **Z. Zhang**, D. Ganesan, A. S. Grabell, and T. Rahman, “EarlyScreen: Multi-scale Instance Fusion for Predicting Neural Activation and Psychopathology in Preschool Children,” Proc. ACM Interact. Mob. Wearable Ubiquitous Technol., vol. 6, no. 2, pp. 1–39, Jul. 2022.

- [8] A. S. Grabell, A. M. Santana, K. N. Thomsen, K. Gonzalez, **Z. Zhang**, Z. Bivins, and T. Rahman, “Prefrontal modulation of frustration-related physiology in preschool children ranging from low to severe irritability,” *Developmental Cognitive Neuroscience*, vol. 55, p. 101112, 2022.
- [9] **Z. Zhang**, Z. Xu, Z. Ahmed, A. Salekin, and T. Rahman, “Hyperspectral image super-resolution in arbitrary input-output band settings,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 749–759.
- [10] L. Zhang, **Z. Zhang**, C. Huang, H. Deng, H. Lin, B.-C. Tseng, J. Drewniak, and C. Hwang, “Decoupling Capacitor Selection Algorithm for PDN Based on Deep Reinforcement Learning,” in *2019 IEEE International Symposium on Electromagnetic Compatibility, Signal Power Integrity (EMC+SIPI)*, Jul. 2019, pp. 616–620.
- [11] **Z. Zhang**, L. Zhang, Z. Sun, N. Erickson, R. From, and J. Fan, “Solving Poisson’s Equation using Deep Learning in Particle Simulation of PN Junction,” in *2019 Joint International Symposium on Electromagnetic Compatibility, Sapporo and Asia-Pacific International Symposium on Electromagnetic Compatibility (EMC Sapporo/APEMC)*, IEEE, 2019, pp. 305–308.

ABSTRACT OF THE DISSERTATION

Dynamic Vision: Unlocking Motion Insights and Multimodal Event Creation

by

Zhongyang Zhang

Doctor of Philosophy in Data Science

University of California San Diego, 2025

Professor Tauhidur Rahman, Chair

Event cameras, or dynamic vision sensors, offer distinct advantages over frame-based cameras, such as a high dynamic range, exceptional temporal resolution, and heightened sensitivity to motion. These characteristics make them particularly suitable for tasks like human pose estimation and dense trajectory estimation.

However, effectively utilizing the spatiotemporal features in event stream data remains challenging. This dissertation addresses this by tackling high-frequency 3D human pose estimation under dynamic lighting conditions, enabling the accumulation and flow of temporal information across time bins. Additionally, for dense continuous-time trajectory estimation, which requires precise pixel-level instantaneous speed understanding, a novel layered bidi-

rectional time surface event representation is introduced. This improved event representation preserves abundant local speed cues, significantly enhancing the feature tracking process.

Throughout the exploration of these event camera-based vision tasks, a critical challenge emerges: the lack of sufficient data. As a novel vision modality, there are few large datasets with high-quality labels and paired data from other modalities. To advance this field, it is crucial to address this issue. This dissertation presents three solutions: generating event data using 3D simulation software, converting video data to event streams, and directly generating event streams using a generative model. These methods explore promising directions for event data generation and complement each other. In the final method, the event modality is aligned with mainstream modalities, allowing events to be generated from other modalities' inputs and vice versa. These efforts help bridge the domain gap between events and other modalities.

Introduction

When discussing vision sensors, people often think of RGB frame-based cameras, infrared cameras [62], depth cameras [163], and even hyperspectral cameras [80]. Each has its unique features and serves specific purposes. The vision sensors in everyday devices like smartphones or DSLR cameras are typically RGB frame-based, capturing red, green, and blue channels similar to the human eye, thus accurately storing and reproducing the world as we see it. In contrast, infrared cameras capture radiation in the infrared spectrum, hyperspectral cameras more densely segment and store information across different spectral ranges, and depth cameras measure the distance between the camera and object surfaces for each pixel. Each type of sensor has its strengths and weaknesses, playing a unique role in various tasks and applications.

Beyond these common vision sensors, a novel type is gaining attention: the event camera [95, 45]. Unlike other sensors, event cameras do not rely on frames or require synchronized clocks and pixels. Each pixel operates independently and asynchronously, monitoring changes in brightness. When a pixel detects a change in the logarithmic brightness value exceeding a preset threshold, it generates an “event” to record this change. Each event e comprises four values: x, y, t, p , representing the spatial coordinates on the image sensor plane, the time of occurrence, and the polarity of the event. The polarity p , a binary value, indicates whether the brightness increased or decreased.

This hardware design gives event cameras their distinct advantages. Without waiting for all pixels to periodically transmit brightness values and form frames, they can deliver significantly more events per unit time under the same bandwidth. This principle grants event cameras their first characteristic: ultra-high temporal resolution. A typical DAVIS346 event

camera by Inivation can theoretically handle 12 million events per second, whereas smartphones or cameras often operate at 30 or 60 frames per second. Additionally, because event cameras record changes in brightness rather than absolute values, they have a much higher dynamic range than frame-based cameras. For instance, the DAVIS346 event camera achieves a dynamic range of 120 dB, compared to 56.7 dB for its built-in RGB CMOS sensor, despite sharing the same lens and light intake.

Moreover, event cameras are inherently active sensors; pixels trigger events only when brightness changes meet specific criteria, rather than being clock-driven to transmit brightness values periodically. This feature provides a third advantage: sensitivity to motion and filtering of relatively static objects. In stable lighting conditions, a stationary object generates no events, effectively filtering it out, while only moving objects relative to the camera produce event streams. This characteristic is advantageous for tracking moving objects or inferring camera motion through relative motion. Lastly, because event cameras generate few events when there are no moving objects in the field of view, they consume low power, typically between 10 mW and 30 mW (DAVIS346), compared to 140 mW for the built-in RGB CMOS sensor. Thus, event cameras are well-suited for low-power applications, such as use in edge devices.

These attributes—high temporal resolution, high dynamic range, low power consumption, and sensitivity to motion—open up diverse applications for event cameras. The high temporal resolution makes them ideal for feature tracking [108], human pose estimation [105, 22, 187], and video frame interpolation [151, 142, 100]. Coupled with their high dynamic range, event cameras can be used for image deblurring [30, 166, 82] and quality enhancement [154]. Their sensitivity to motion provides unique advantages in tasks like optical flow estimation [21, 98], simultaneous localization and mapping (SLAM) [76, 17], and object segmentation [8].

These features ultimately enhance the performance of event cameras in motion capture and understanding. Higher temporal resolution aids in capturing finer motion trajectories, greater dynamic range reduces the impact of lighting conditions on capturing moving targets, and sensitivity to moving objects is tailor-made for motion capture. My series of works first focus on

motion capture, specifically high-frequency human pose estimation under challenging lighting conditions. This research leverages the spatiotemporal continuity of events, using historical and near-future information to improve the accuracy of human keypoint recognition, even in extremely low-lighting conditions [173].

In my first project, I utilized the TORE (Time-Ordered Recent Event) representation proposed by Baldwin et al. [12] as the input model for event representation. It effectively uses historical events to predict human keypoints at the current time. However, this representation struggles to predict instantaneous velocities at intermediate time points and suffers from severe temporal occlusion issues. While effective for human pose estimation, it falls short for dense continuous-time trajectory estimation with event streams. Therefore, I proposed a new event representation Labits [175], constructing a bidirectional time surface at evenly-spaced intermediate time points to more accurately capture the instantaneous velocity of objects at each pixel. Ideally, the spatial derivative on this event representation corresponds to the inverse of instantaneous velocity, enabling more precise pixel-level tracking.

Despite their suitability for motion-related problems, event cameras face a significant challenge: data availability. Compared to other visual modalities, event cameras are still relatively new, with limited high-quality labeled event data available [45, 24, 25]. There is also a lack of datasets paired with other mainstream modalities like text, audio, infrared, and depth images. As a result, the event modality remains relatively isolated, and many data- and label-hungry tasks cannot rely solely on event camera, hindering large-scale applications in general computer vision tasks.

Three of my four PhD projects aim to address this issue by facilitating event generation and conversion between event and other modalities. In the first project [173], we used 3D modeling software like Blender [18] to create 3D scenes and load motion scripts, rendering them into videos with animation software. We then generated event streams using existing video-to-event (v2e) [69] tools. This method allows for highly customized event stream generation, offering significant flexibility. Simulated events generated this way have proven effective in

augmenting real data, with models pre-trained on these simulations showing improved accuracy in human keypoint recognition compared to those trained solely on real data. However, this approach is time-consuming and labor-intensive, requiring meticulous scene modeling, including characters, environment, lighting, camera angles, actions, and background videos.

In traditional computer vision, researchers have accumulated extensive labeled image and video data [176, 111, 6, 96]. Reusing these resources could alleviate the data scarcity in event-based vision domain. However, existing video-to-event conversions are rough, struggling with overexposed and underexposed areas in videos, and suffering from temporal inaccuracies and event count discrepancies [69, 180, 122, 47]. To address these issues, my second project proposed V2CE [174], which combines deep learning and probabilistic local dynamic-aware time resampling to significantly improve the accuracy of video-to-event conversion, particularly in temporal precision and continuity. By considering recent past and future temporal information and optimizing a task-specific loss set, I resolved the problem of sparse voxel grids in deep learning-generated event voxel grids. This approach enables more accurate and continuous temporal conversion when transforming event voxel grids into event streams.

While V2CE effectively addresses video-to-event conversion, it only supports this specific transformation scheme and cannot integrate multiple modalities or allow text prompts to directly modify generated events. Similarly, event streams are challenging to align with other modalities for downstream tasks or be used for generating other modalities [160, 177]. In my final project, I proposed a comprehensive solution to align event modalities with other mainstream modalities, enabling event streams to generate content in other modalities or vice versa. This approach uses generative models to create events under various conditions, completing my series of solutions to the event data scarcity problem.

This thesis will unfold chronologically across several projects. The first chapter introduces the application of event cameras in high-frequency, low-light human pose estimation. The second chapter details efforts to enhance the accuracy of converting videos into time-continuous events. In the third chapter, I present a new representation, Labits: Layered Bidirectional Time

Surface Representation, which better captures instantaneous velocity changes over time, aiding in dense continuous-time trajectory estimation. The fourth chapter discusses my latest work, Larver: Learning Alignment and Representation for Versatile Event Regeneration. This work is still under active development, so the chapter does not cover the results section. Finally, the thesis concludes with a summary of this series of works.

Chapter 1

Neuromorphic High-Frequency 3D Dancing Pose Estimation in Dynamic Environment

1.1 Introduction

Technology-Mediated Dancing (TMD), which leverages digital systems to facilitate remote, engaging, and health-promoting physical activities in dance form, has always been a part of gaming. More and more so, TMD is becoming a central part of the immersive experience as the digital experience becomes more and more blended with the physical reality. [102, 33, 107]. Ranging from remote control-based gaming console games to Virtual Reality (VR) platforms, various TMD forms are intertwining with users' living spaces. Human pose estimation (HPE) serves a crucial function in TMDs, as it discerns users' unique, intricate, and rapidly evolving dance poses, enabling interaction with computers. To accommodate the broadest range of players, TMD necessitates high-fidelity HPE that functions reliably in diverse, challenging, and realistic indoor environments, such as dynamic lighting and background conditions.

Contemporary HPE systems predominantly rely on depth and RGB cameras; however, both standard RGB camera-based and depth-based monocular HPE systems [31, 32, 59] struggle to generate ultra-fast, high-speed human pose inferences due to their limited frame rates—a critical requirement for numerous real-world applications (e.g., virtual reality dance games, high-

frequency motion characterization for tremor monitoring applications). The FPS of mainstream RGBD cameras (e.g., Microsoft Azure Kinect and Intel RealSense) is typically limited to 30 FPS due to a combination of hardware and software constraints, such as the time overhead introduced by IR projectors. Although some high-spec RGB cameras support relatively high FPS, they are usually fixed and have an attainable upper limit. In contrast, neuromorphic cameras can achieve a bandwidth of over 10 million events per second with a minimum latency of approximately $20 \mu s$ (DAVIS 346). Provided that computational resources are available, neuromorphic cameras can attain extremely high-frequency inference. A detailed analysis can be found in the Results section surrounding Fig. 1.16. Moreover, while RGB-based HPE falters in low-light conditions due to substantial motion blur-related issues, depth cameras can only operate within a limited depth range. Furthermore, neither camera inherently distinguishes between static and moving objects, leading to both static and dynamic objects in the background being captured alongside the target human body. These shortcomings compromise HPE robustness in dynamic settings. Depth cameras, which employ active infrared light, also demand considerable power consumption. Consequently, our proposed work aims to develop a high-speed, low-latency, and low-power 3D human pose estimation framework capable of functioning in dynamic lighting and motion environments.

Neuromorphic cameras [95], also known as Dynamic Vision Sensors (DVS) or event cameras, have a silicon retina design based on a core mechanism of mammalian vision, rendering them particularly sensitive to moving targets and fluctuating lighting conditions. These cameras operate on the premise that, for mammals, moving objects often contain more valuable information for hunting and evading predators, whereas relatively static backgrounds deserve less attention for continuous monitoring and processing. Emulating this dynamic vision characteristic, each neuromorphic camera pixel operates asynchronously; independently monitoring logarithmic minute brightness changes, thereby ensuring sensitivity to motion in both high and low-light conditions. This mechanism inherently filters out static backgrounds without necessitating the transmission of detailed, bulky frames each time an event occurs. Instead, moving targets can

rapidly activate numerous events with exceptional time resolution, enhancing neuromorphic camera sensitivity to dynamic targets. The wide dynamic range renders neuromorphic cameras resilient across various situations: from nighttime to glaring noon, from tunnels to night driving. Beyond their advantages in low-light conditions, neuromorphic cameras also exhibit reduced sensitivity to skin color and brightness changes [119, 45].

DVS HPE has garnered considerable interest recently due to the aforementioned advantages, prompting the collection of several noteworthy datasets [22, 133]. Despite these datasets utilizing fixed patterns for motion guidance rather than dances, they also exhibit significant limitations concerning the development of pragmatic systems. Data acquisition in these studies is conducted in ideal laboratory settings, devoid of any background motion to serve as interference. Lighting conditions are optimal and stable, precluding the exploration of low and/or variable lighting scenarios. These uncertainties and noise, reflective of real-world environments, present genuine challenges that must be addressed to render this technology more practical and generalizable. Consequently, we introduce two novel datasets: one featuring a real-world dynamic background under both high and low-light conditions, and the other comprising a simulated environment with a fully controllable and customizable pipeline for generating new data samples on demand. These datasets not only suit the needs of this research but also benefit the broader community as they will be made publicly available. A comparison between the existing neuromorphic camera-based HPE datasets and our assembled datasets are presented in Table 1.1.

Not only are existing datasets overly idealistic, previous DVS HPE efforts are further constrained by the “missing torso” problem, an intrinsic limitation stemming from the dynamic properties of the neuromorphic camera. When certain human body parts remain stationary, neuromorphic cameras solely capture other moving components, disregarding these static portions. Consequently, during such periods, the event representation contains minimal or no information about these parts, leading to substantial estimation inaccuracies.

To construct a DVS HPE system compatible with more realistic environments, we propose

Table 1.1. A comparison between existing 3D HPE datasets and our collected datasets. †: “Modality” indicates the modality used in the proposed pipeline rather than all paired modalities. ‡: “tight” refers to Mocap-specific tight clothing, “casual” to everyday attire, and “arbitrary” to any clothing type, even if it may cause significant occlusion. MMHPSD dataset is not publicly available.

Study	Human3.6M[71]	MKV[186]	DHP19[22]	MMHPSD[187]	YeLan	
Modality [†]	RGB	RGBD	DVS	DVS	DVS	
Inference Rate	50 FPS	10 FPS	Arbitrary, 50 FPS verified	15FPS	Arbitrary, 150 FPS verified	
Lighting	Ideal	Ideal	Ideal	Ideal	Low to High	
Data Type	Real	Real	Real	Real	Synthetic	Real
Background	Static	Static	Static	Static	Dynamic	Both
View	Arbitrary	Arbitrary	Arbitrary	Arbitrary	Arbitrary	Front
Clothing [‡]	Casual	Casual	Tight	Casual	Arbitrary	Tight
Data Size	3600000	22406	350860	240000	3958169	446158

a two-stage system, *YeLan*¹, which accurately estimates human poses under low-light conditions amid noisy background elements. In the first stage, an early-exit-style mask prediction network is implemented to eliminate moving background objects while maximizing energy efficiency. The second stage employs a BiConvLSTM to facilitate information flow between frames, mitigating the missing torso issue. Additionally, TORE volume is utilized to construct denser and more informative input tensors, addressing the low event rate problem in low-light settings. We conducted extensive experiments comparing our approach with baseline DVS HPE methods, achieving state-of-the-art results on the two proposed new datasets.

In summary, the core contributions of this paper are as follows:

1. YeLan represents the first neuromorphic camera-based 3D human pose estimation solution tailored for dance motion, functioning robustly in challenging conditions such as low lighting and occlusion, and overcoming neuromorphic camera limitations while harnessing their strengths.
2. Our end-to-end simulator enables precise, low-level control over generated events and produces the first and largest neuromorphic camera dataset for dance HPE. This synthetic

¹Derived from a character in the video game Genshin Impact, the name *YeLan* signifies “night orchid” and can also be construed as the Chinese term for “night viewing.”

dataset (*Yelan-Syn-Dataset*) surpasses existing resources in quantity and variability.

3. We carry out a human subject study and collect a real-world dance HPE dataset (*Yelan-Real-Dataset*), taking into account low-light conditions and mobile background content.

Both datasets and code will be made publicly available following publication.

1.2 Related Work

Dance and Its Effects: Spanning various age groups, dance has a rich history and is a beloved form of communication and physical activity with a rich history. Due to its cultural diversity and practicality, dance has evolved into numerous variations across regions and time periods, encompassing diverse styles, rhythms, intensities, and steps. Substantial research evidence supports the notion that dance exerts a significant positive influence on physical and mental health [131]. A recent study [148] revealed that dance can enhance neuroplasticity and stimulate neural activation in multiple brain regions. Consequently, dance can serve as a rehabilitative tool for an array of brain-related pathologies [131]. Recent literature indicates that dance therapy can have a considerable positive impact on depression [2], schizophrenia [33], Parkinson’s [58], fibromyalgia [102, 38], dementia [19], cognitive deterioration [107, 185], stress [118], and chronic stroke [117].

Technology-mediated dance has gained popularity in recent years [66], rendering dance more accessible and facilitating learning [130, 128]. Dance has been integrated into various video games [66, 84, 130, 1], including the *Just Dance Series* [158], *Dance Dance Revolution* [37], and *Dance Central* [35]. Additionally, movement-based VR rhythm games such as *Beat Saber* [14], *Synth Riders* [143], and *Dance Collider* [36] have emerged.

High-fidelity 3D human pose estimation is a critical component of technology-mediated dance, as it enables the translation of dance gestures into input or commands [3, 4] for AR/VR or dance video games. These 3D human pose estimators facilitate performance evaluation, personalized feedback, and choreography [66]. However, conventional RGB and depth camera-based 3D human pose estimations often struggle to capture rapid or high-speed pose changes

during dance performances in challenging conditions (e.g., low light, dynamic background). In this work, we introduce a neuromorphic camera-based 3D human pose estimation system designed to support technology-mediated dance in these dynamic and demanding settings.

Human Pose Estimation: Depending on the taxonomy, existing 3D human pose estimation methods can be classified by modality, the number of sensors used in tracking, and result forms. Most approaches rely on RGB-based [172, 93, 164, 59, 32] or RGB-Depth-based modalities [126, 140, 109, 186]. Generally, RGB image-based methods have lower equipment requirements and are more extensively explored. However, RGB-Depth cameras benefit from the additional depth information they provide, aiding in detection, segmentation, and parts localization. Still, depth cameras depend heavily on IR projectors for depth map construction, which consume significant power and are sensitive to bright environmental light. This limits the working distance and FOV of RGBD cameras and hinders their outdoor application.

For the second classification method, the resulting two categories are: monocular [171, 32, 186] and multi-view [138, 150, 125] methods. The multi-view method requires substantially more power and involves capturing subjects from multiple views/cameras positioned at different angles around a capturing area [46]. Moreover, setting up a multi-view neuromorphic camera data collection system is complex and costly, making applications outside the lab very challenging. The monocular method estimates human pose from a single camera view.

An alternative classification approach for human pose estimation techniques involves determining if they are model-based methods, as described by Omran et al. [115], or skeleton-based methods, as outlined by Chen et al. [31]. While the model-based method seeks to reconstruct the full 3D body shape of a human model [59], skeleton-based methods utilize a bone skeleton as an intermediate representation and regress the joint locations in 3D space. In this work, we use a monocular skeleton-based DVS HPE approach to achieve a more efficient and practical solution for 3D HPE in real-world settings.

Dynamic Vision Sensor (DVS), or neuromorphic camera, was originally proposed by Lichtsteiner et al. [95]. In recent years, the neuromorphic camera has increasingly garnered

attention and has been applied to various computer vision tasks, including object recognition [94, 81], segmentation [8], corner detection [169, 110], gesture recognition [22, 157, 53], optical flow estimation [21, 98], depth estimation [48, 63], Simultaneous Localization And Mapping (SLAM) [76, 17], autonomous driving [91, 29], and human pose estimation [105, 22, 187]. While RGB cameras struggle due to motion blur, neuromorphic cameras are highly sensitive to lux variations in both extremely overexposed and underexposed scenes [16] by design.

Neuromorphic Camera-based Human Pose Estimation: While neuromorphic cameras have been previously proposed for human pose estimation in existing literature, the focus of prior work has mainly been on designing algorithms for relatively noise-free, background-activity-less, and well-lit settings. For instance, the recent neuromorphic camera HPE dataset, DHP19 [22], features event recordings of human movements, poses, and moving objects, and [187] further collected a multi-modality HPE dataset. Additionally, some neuromorphic camera datasets have been employed in gesture recognition [9] and action recognition [70]. However, these datasets capture a limited number of motion trajectory types in controlled, noise-free environments, which hampers the development of a robust neuromorphic camera-based sensing system.

The lack of a diversified dataset in realistic conditions under low and dynamic light is a major bottleneck for creating a more robust neuromorphic camera-based sensing system. In this work, we address this issue by demonstrating how a neuromorphic camera-based mobile sensing platform can effectively capture high-frequency 3D human poses during a dance performance while being highly resilient to various real-world challenging conditions. We aim to develop a versatile and robust system for human pose estimation that can be successfully applied in a wide range of real-world scenarios by incorporating resilience to low light; dynamic moving backgrounds; higher sensor fields of view; longer distances between the sensor and target human; and diverse outfits or clothing.

1.3 Design Consideration

Challenges in Neuromorphic Camera-based Dancing HPE: In contrast to everyday movements or gestures such as walking, jumping, and waving hands, dance encompasses a far greater diversity of movements that are unlikely to be included in existing fixed-activities-based HPE datasets, rendering dancing HPE a more challenging and generalized problem. Furthermore, compared to gesture recognition-level daily actions, a dancer constantly alters their pose at a significantly faster pace. The average joint location change rate is generally much larger, imposing a higher demand on the temporal resolution of pose estimation. This is particularly true for high-fidelity and professional dancing recordings employed in the gaming industry or virtual dance video production.

Neuromorphic cameras, with their inherent high-frequency and motion-sensitive characteristics, are well-suited for addressing these challenges. However, they also face unique obstacles compared to RGB-based methods. For instance, in noisy and dynamic environments, their extreme motion sensitivity can result in numerous events from environmental signals rather than the actual dancer’s motion. Sometimes, these events even overwhelm those generated by the dancer, leading to a severe event filtering problem. Moreover, low-light conditions produce fewer events for the same movement and hence further lower the signal-to-noise ratio in the event representation.

To address these challenges, YeLan proposes an early-exit-style event filtering mechanism that predicts a binary mask over the target human body and rejects all events not generated due to the user’s body movement. This helps to improve the accuracy and reliability of the human pose estimation model in various environments.

Missing torso problem: Another unique problem for DVS HPE is the missing torso problem. During physical activity, not all body parts move equally. For instance, the upper body may be in motion while the lower body remains stationary. This partial immobility results in the silence of corresponding pixels within the neuromorphic camera, leading to a higher error rate in

predicting these missing joints. To solve this problem, we need to focus on the following points:

1. *Appropriate spatiotemporal representation:* An adeptly designed spatiotemporal representation can efficiently compile information derived from an extensive event stream, preserve vital historical data, and eliminate noise events. These factors collectively enhance the understanding of human body part locations and mitigate the missing torso issue.
2. *Information flow between frames:* Neighboring frames exhibit similarities in historical information and human joint locations. By considering the feature maps of these frames as a time series and employing a robust temporal method to facilitate information flow between frames, it becomes possible to gather insights regarding the missing torso and generate more persuasive joint location estimates.

Prior research on neuromorphic cameras has introduced various representations, such as event frames [123], constant-count frames [106], time surfaces [15], and voxel grids [182]. A time surface is a 2D representation in which each pixel encodes the timestamp of the most recent event as its pixel value. Each pixel retains the timestamp of the latest event [86] occurring at that location. A voxel grid, meanwhile, constitutes a 3D histogram of events, where each voxel denotes the number of events within a specified interval at a particular pixel location. By retaining the spatiotemporal information of the entire event history, voxel grids circumvent the loss of information associated with collapsing the history into a 2D grid representation. However, event frames, event counts, and voxel grids do not effectively conserve distantly related historical information, deteriorating the missing torso problem. Time surfaces, conversely, may discard temporal information and are unable to maintain information from multiple events at identical pixel locations. Furthermore, these existing representations are hindered by a low signal-to-noise ratio (SNR) in poorly illuminated conditions.

In this study, we employ a modified Time Ordered Recent Event (TORE) volume representation [12] that concurrently preserves both the most recent and relevant historical information,

thereby helping alleviate the missing torso issue. This representation also functions as a noise filter for noise types such as salt and pepper without impacting other significant signals.

Spatiotemporal representation: Learning a representation that preserves meaningful spatiotemporal information properly about the human pose is a fundamental challenge. Especially given that neuromorphic cameras can generate a massive event sequence within a short period of time.

Prior work on neuromorphic cameras proposed different representations, including event frame [123], constant-count frame [106], time surface [15], and voxel grid [182]. Time surface is a 2D representation where each pixel records the most recent event’s timestamp as its pixel value. Each pixel stores the timestamp of the last event [86] in that location. A voxel grid is a 3D histogram of events. Each voxel represents the number of events in an interval at a pixel location. Voxel grid prevents information loss by preserving spatial-temporal information of the whole history instead of collapsing the history into a 2D grid representation. However, event frame, event count, or voxel grid do not preserve relatively distant historical information, which gives rise to the missing torso problem. Time surface, on the other hand, can discard the temporal information and cannot keep the information from multiple events at the same pixel location. Moreover, these existing representations suffer from a low signal-to-noise ratio (SNR) in low-lighting conditions.

In this work, we use a modified Time Ordered Recent Event (TORE) volume representation [12] that can simultaneously preserve both the latest information and recent historical information, which can then compensate for the missing torso problem. It also serves as a noise filter for noises like salt and paper without affecting other informative signals.

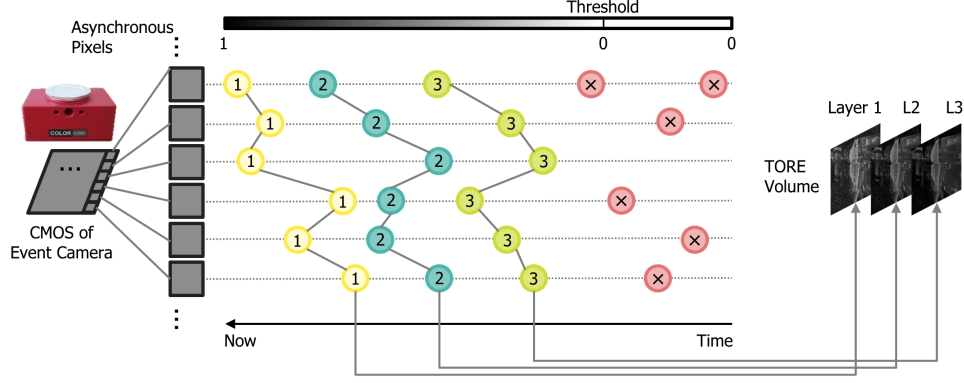


Figure 1.1. The schematic representation of the TORE volume. As each pixel in the neuromorphic camera operates asynchronously, we establish a First-In-First-Out (FIFO) queue for each pixel corresponding to each event polarity (polarity denotes whether the event is triggered by an increase or decrease in brightness). The queue possesses a depth of K (here $K=3$), and when a queue becomes full and subsequent events arrive, the oldest event is removed from the queue. The i^{th} layer of TORE is produced by extracting the i^{th} event stored in the FIFO queues of all pixels.

1.4 Proposed System

1.4.1 Event Preprocessing and TORE Volume

TORE [12] attempts to mimic the human retina by preserving the membrane’s potential properties. A fixed-length(e.g., K) First-In-First-Out (FIFO) queue is adopted to record the relative timestamp of the most recent K events. When a new event enters a pixel’s queue, its relative timestamp is inserted, and the oldest event in the queue is expelled. TORE calculates the logarithm of these timestamps in the FIFO buffer. TORE transforms the sparse event stream into a dense, bio-inspired representation with minimal information loss, achieving state-of-the-art results in various DVS tasks (e.g., classification, denoising, human pose estimation). A comprehensive comparison of different representations can be found in [12].

In contrast to the original TORE volume, we modify TORE through normalization, 0-1 flip, and range scaling. We first perform the missing normalization and then invert the normalized maximum and minimum values. In the original paper [12], the oldest events and pixels with no recorded events are assigned the maximum value $\log(\tau)$, which is counterintuitive and may

hinder the convergence speed. We can readily address this issue by setting the value v in TORE to $1 - v/\log(\tau)$. With this modification, older events have lower values, while newer events have higher values, and pixels with no recorded events are set to 0. Lastly, since the logarithm is utilized in TORE calculations, the most recent $4.63\mu s$ occupies the $[0.7, 1]$ range of the entire $[0, 1]$, and $4.63\mu s$ is shorter than the DVS camera’s sensitivity time (150ms, as used in the original TORE paper [12]). Consequently, a rescaling is introduced to mitigate this issue. Our modification formula is as follows:

$$v' = (1 - v/\log(\tau))|_{[0,0.7]}/0.7 \quad (1.1)$$

where v and v' denote the original and modified TORE values, while the notation $x|_{[a,b]}$ indicates the value of x is clamped within the range $[a, b]$.

TORE maintains a first-in-first-out queue individually for each pixel, with values in the queue decreasing over time. TORE can preserve pixel history for up to five seconds. These attributes render it suitable for dynamic conditions. In high-lighting conditions, TORE can discard redundant histories for the same pixel, alleviating the model’s computational burden. In low-lighting conditions, past event histories can compensate and aid in determining the position of some joints that have not moved significantly during the last time window, thereby helping address the missing torso issue. Fig. 1.1 illustrates the process of generating modified TORE volumes.

1.4.2 Event Filtering with Human Body Mask Prediction Network

To filter out events triggered by background activities and neuromorphic camera hardware noise (e.g., hot pixel and leak noise [69]), a mask prediction network is employed, capable of predicting a human body mask from the TORE volume representation.

For the mask prediction network, we adopt a modified version of U-Net [129] proposed by Olaf et al. Unlike the original version, our modified iteration can predict the mask for the

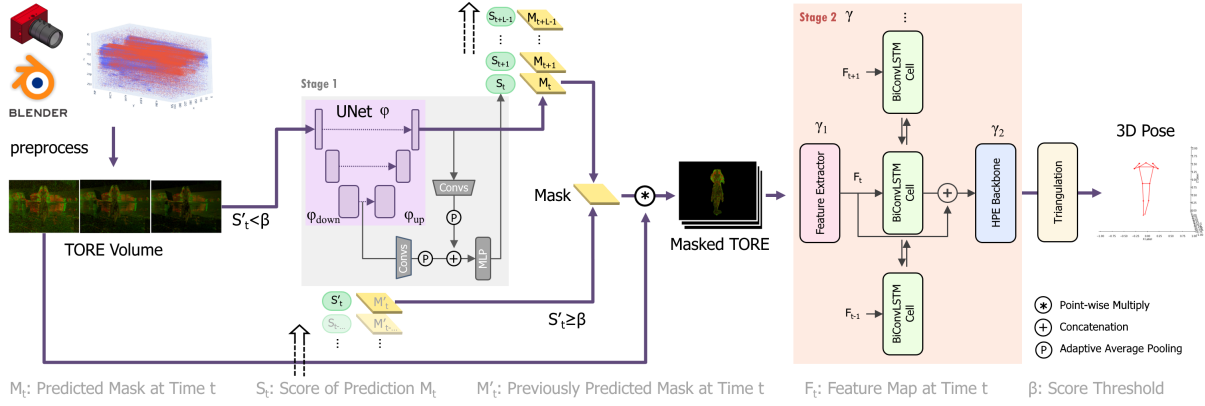


Figure 1.2. The pipeline of the proposed *YeLan*. It initially processes the event stream into TORE volumes, which are subsequently sent to the stage one human body mask prediction network. This network predicts a series of masks for the ensuing frames, accompanied by quality-assessment scores to minimize computation costs. The estimated human mask undergoes point-wise multiplication with the original TORE volume before advancing to the next stage. Stage two encompasses the human pose estimation network, where BiConvLSTM and three hourglass-like refinement blocks are employed to estimate the heatmap of joints’ projections on three orthogonal planes. The precise 3D coordinates of these joints are determined through a triangulation method based on these heatmaps.

current input frame and a series of masks following this frame. Each predicted mask is generated in conjunction with a confidence score. The primary rationale behind predicting the human body mask of future frames is that future motion trajectories of different human body parts are generally predictable with information about current and previous motion trajectories. Time-Ordered Recent Event (TORE) volume representation efficiently captures current and previous motion history. This enables the mask prediction network to estimate human body masks of the current and future frames and their corresponding confidence scores. The confidence scores offer an early-exit-like mechanism, allowing the computational pipeline to bypass mask prediction for a specific frame if the mask predicted based on a previous frame’s TORE volume achieves a high confidence score. This mechanism significantly reduces the computational cost and energy consumption (further results in Section 1.7.3). For the predicted mask, the U-Net generates floating point numbers between 0 and 1, with a binarization process employing a minimal threshold of 0.1. The small threshold ensures that the generated mask does not exclude any

parts of the human body, as failure to encompass the entirety would result in greater error than permitting a slight amount of noise.

Denote the input TORE representation as X_t , and the U-Net here as φ , with the first half denoted as φ_{down} and the latter half as φ_{up} . Then, we have:

$$\varphi(X_t) = \begin{cases} M'_t, S'_t \text{ exists \& } S'_t \geq \beta \\ [M_t, M_{t+1}, \dots, M_{t+L-1}], \text{ Otherwise} \end{cases} \quad (1.2)$$

where M_t represents the predicted human mask at timestamp t , and M'_t signifies the human mask of this timestamp previously predicted. The confidence scores of M_t and M'_t are denoted as S_t and S'_t , respectively, where:

$$[S_t, \dots, S_{t+L-1}] = MLP[P[Conv_1[\varphi_{down}(X_t)]] + P[Conv_2[\varphi(X_t)]]] \quad (1.3)$$

when the corresponding S'_t does not exist. Here, P represents adaptive pooling layers, Conv denotes convolutional layers, and MLP signifies a multi-layer perceptron.

A significant challenge in training the mask prediction network with an end-to-end approach is the absence of a realistic neuromorphic camera dataset containing labeled human body mask sequences for recorded event streams. Nevertheless, as our proposed motion-to-event simulator (outlined in Section 1.5) can generate paired pixel-level human masks at a high frame rate, they can be utilized to comprehensively train the mask prediction network.

1.4.3 Human Pose Estimation Network

The human pose estimation network comprises a ResNet-based feature extractor, a Bidirectional Convolutional LSTM (BiConvLSTM) layer, an HPE backbone, and a triangulation module. The initial portion of ResNet34 functions as the feature extractor, succeeded by a BiConvLSTM layer incorporating a skip connection. Given that the human body undergoes no abrupt changes within a relatively brief temporal window, adjacent frames typically exhibit

similar ground truth labels. This continuity in human joint movements renders it advantageous to reference neighboring frames when estimating joint positions.

BiConvLSTM represents a bidirectional variant of ConvLSTM [137], in which ConvLSTM constitutes a form of recurrent neural network designed for spatiotemporal prediction, incorporating convolutional structures within both input-to-state and state-to-state transitions. Subsequently, the HPE backbone comprises three hourglass-like CNN blocks, each producing a series of marginal heatmaps to reconstruct the coordinates of human joints in 3D space. All intermediate outputs from these three blocks are utilized to compute the loss with the ground truth heatmaps, while the final two blocks can be regarded as refinement networks. The feature extractor and backbone network architecture have been developed based on a model proposed in [133]. Denoting the feature extractor as γ_1 and the generated feature map as F_t , we have:

$$F_t = \gamma_1(X_t \cdot M_t) \quad (1.4)$$

where the symbol $\cdot$ denotes pixel-wise multiplication. Consequently, the generated joint heatmaps H_t can be derived as follows:

$$H_t = \gamma_2[\text{Conv}[\text{BiConvLSTM}(F_t), F_t]] \quad (1.5)$$

For each joint, YeLan generates three heatmaps depicting the probability of its projected position on the xy , xz , and yz planes (denoted as $H_t^{xy}, H_t^{xz}, H_t^{yz}$). Subsequently, a soft-argmax operator is applied to extract the normalized coordinates of the joint. Ultimately, predictions from the xy plane serve as the final estimations for x and y coordinates, while values for z are

computed by averaging the yz and xz predictions. The formula can be expressed as follows:

$$[x_{tj}^{xy}, y_{tj}^{xy}] = \text{soft-argmax}(H_{tj}^{xy}) \quad (1.6)$$

$$[x_{tj}^{xz}, z_{tj}^{xz}] = \text{soft-argmax}(H_{tj}^{xz}) \quad (1.7)$$

$$[y_{tj}^{yz}, z_{tj}^{yz}] = \text{soft-argmax}(H_{tj}^{yz}) \quad (1.8)$$

$$p_{tj}^{xyz} = [x_{tj}^{xy}, y_{tj}^{xy}, \frac{z_{tj}^{yz} + z_{tj}^{xz}}{2}] \quad (1.9)$$

where x_t^{xy} denotes the estimated x at time t for a specific joint j 's predicted xy -plane heatmap, and p_{xyz} represents the predicted 3D coordinates for joint j at time t .

The ground truth labels employed during training and testing are normalized prior to being input into the network. For a particular joint, we initially project it onto a plane parallel to the camera's image plane, maintaining the same depth as the depth reference. The head joint's depth value serves as this reference. Subsequently, the 3D space within the DVS camera's view is mapped to a cube with a range of $[-1, 1]$. Finally, as the network does not directly predict the 3D coordinates of a joint but instead forecasts its marginal heatmaps, we extract the joints' projections on three orthogonal faces of the normalized space cube to generate the ground truth for marginal heatmaps. The ultimate marginal heatmaps are computed using a Gaussian filter applied to these projection images [133].

1.4.4 Loss

For the mask prediction network, the loss function comprises three components. The initial component is the Binary Cross Entropy (BCE) loss, computed between the predicted mask series $\hat{M}$ and the corresponding ground truth masks M . This loss is implemented to ensure the precision of all generated masks. Subsequently, a Mean Square Error (MSE) loss is calculated over the predicted confidence scores S and their ground truth. The ground truth score represents the Mean Absolute Error (MAE) between a predicted mask and its respective ground truth mask. Finally, although the objective is to predict the mask series for the current frame and

subsequent frames, they are not uniformly significant. It is imperative to ensure that masks for proximate frames receive greater weighting, particularly for the current frame. Consequently, an additional BCE loss is computed between the predicted mask for the current frame ($\hat{M}_0$) and its corresponding ground truth M_0 . The cumulative loss is:

$$Loss_{mask} = BCE(M, \hat{M}) + BCE(M_0, \hat{M}_0) + MSE(S, MAE(M, \hat{M})) \quad (1.10)$$

We employed the loss presented in [133] for the human pose estimation network. As the marginal heatmaps can be interpreted as probability distributions of joint locations, Jensen-Shannon Divergence (JSD) can be applied between the predicted heatmaps by each block ($\hat{H}^i$, where i denotes the block index) and the ground truth heatmaps H on each projection plane (xy, xz, zy). Additionally, a geometrical loss is computed between the reconstructed 3D joint coordinates $\hat{p}_{xyz}$ and their ground truth p_{xyz} . The loss for this stage can be expressed as follows:

$$Loss_{HPE} = \sum_i (||\hat{p}_{xyz}^i - p_{xyz}||_2 + JSD(H_{xy}, \hat{H}_{xy}) + JSD(H_{xz}, \hat{H}_{xz}) + JSD(H_{zy}, \hat{H}_{zy})) \quad (1.11)$$

1.5 Synthetic Data Generation with Motion-to-Event Simulator

First and foremost, while some neuromorphic camera-based HPE datasets exist [22], they primarily focus on fixed everyday movements (e.g., walking, jumping, waving hands), causing models trained on them to falter when faced with intricate movements. Dance performances generally encompass swift and complex gestures, which are scarce in a typical everyday gesture dataset such as [22]. Furthermore, these datasets have been collected under optimal lighting conditions with blank or static backgrounds, failing to represent real-world environments. Additionally, the extant dataset suffers from a lack of diversity in participants, motion dynamics,

clothing styles, and types of background activities. Concerning clothing, all participants don identical, form-fitting black attire. To address this critical gap, we propose generating synthetic data with a comprehensive motion-to-event simulator.

Regarding simulators, though several existing neuromorphic camera simulators are available [112, 122, 69, 78], they share a common and pivotal issue. Almost all exclusively convert existing images or videos into event streams, rather than creating highly customized event streams tailored to specific research problem requirements. Furthermore, for existing simulators, if the source video lacks human joint labels, the generated event stream is also devoid of labels. Nonetheless, in this work, the simulator we develop employs a physics-aware rendering system and renders all relevant parameters fully controlled and customized, including lighting, motion, human gender, body shape, skin color, clothing and accessories, background, and scenes. This feature allows us to implement complex dance movements, and all generated data is paired with accurate labels.

1.5.1 Advantages of Synthetic Data

Numerous advantages of employing a synthetic data generator have been emphasized in recent work [57], with a car driving-based DVS event simulator also proposed [43]. Acquiring real-world data is both costly and time-consuming, as high costs per participant limit sample size, scene scale, and diversity in physical and environmental attributes. The synthetic dataset generation process affords comprehensive control, enabling precise management of a wide array of parameters. The resulting video file can be rendered at an exceptionally high frame rate (FPS), mitigating blur issues stemming from under or over-exposure in dynamic lighting conditions. Finally, the ground truth of human joints' 3D locations can be extracted within the software for various dances and human models.

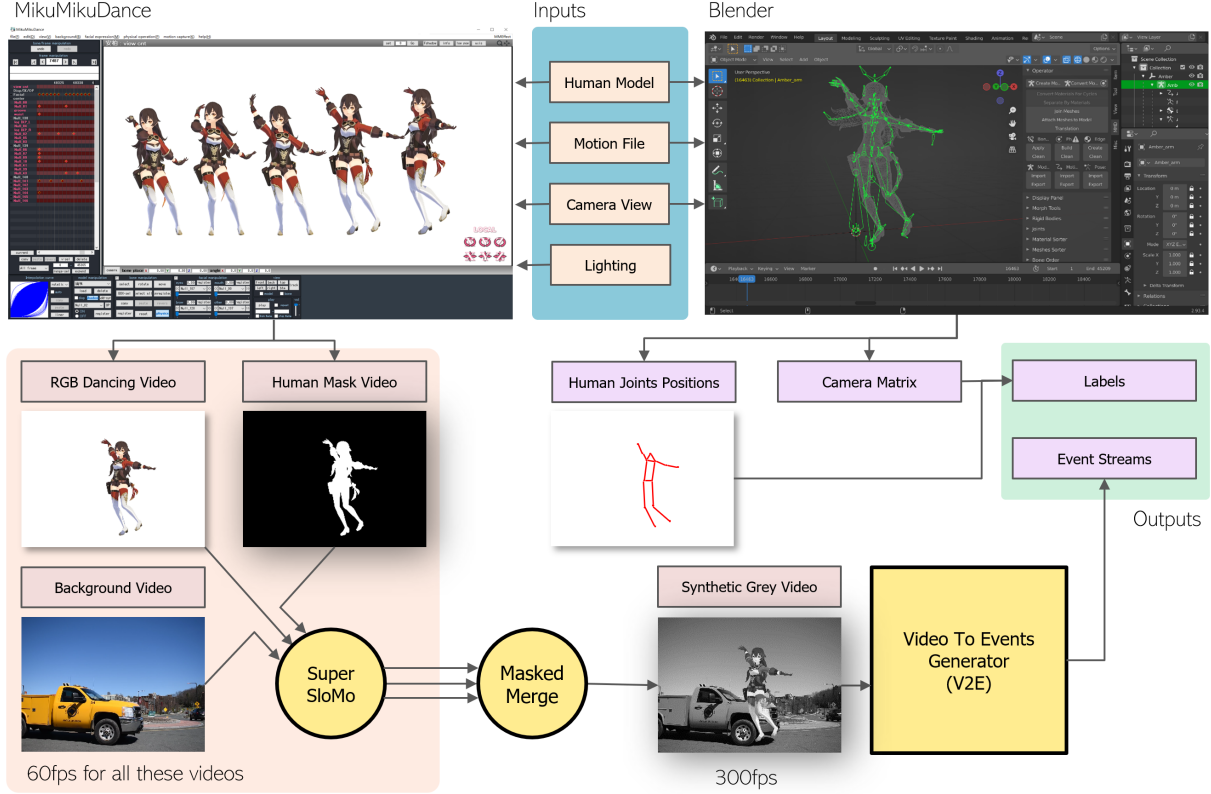


Figure 1.3. The pipeline for synthetic data generation utilizing the comprehensive motion-to-event simulator comprises motion files, human models, camera views, lighting, and other settings as input. These inputs are rendered into RGB and human mask videos in MMD, subsequently merged with background videos. The merged videos undergo processing as event streams using V2E, while the corresponding labels are computed in Blender.

1.5.2 Tools used in the simulator

Synthetic data generation consists of RGB dance video rendering, human joints' position extraction, and events generation. MikuMikuDance (MMD), Blender, and V2E are chosen for each step, respectively.

MMD is a freeware animation program that lets users animate and creates 3D animated movies. This software is simple but powerful, with a long history and a big open-source community behind it. Plenty of human models, scenes, and movement data can be easily accessed for free. In addition, it can automatically handle clothing physics and interaction with the body in a sophisticated manner with minimal manual adjustment.

Software *Blender* is adopted to generate human joints' ground truth labels and camera matrix. *Blender* is a free and open-source 3D computer graphics software for creating animated films, motion graphics, etc. It is highly customizable, and all essential information can be accessed during rendering, including the 6 degrees of freedom coordinates of human joints and the camera center. The 13 key points' ground truth coordinates are extracted at 300 frames per second (FPS).

Lastly, *Video to Event (V2E)* is used for event generation. V2E is a toolset released in 2021 by Delbruck et al. [69]. It can synthesize realistic event data from any conventional frame-based video using an accurate pixel model that mimics the neuromorphic camera's nonidealities. According to its author, V2E supports an extensive range of customizable parameters and is currently the only tool to model neuromorphic cameras realistically under low illumination conditions.

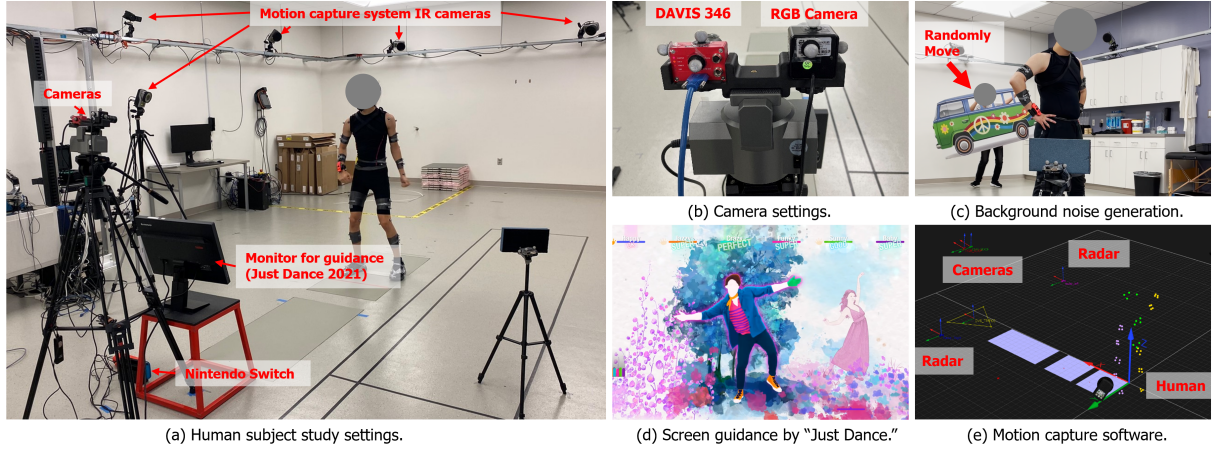


Figure 1.4. Settings of human subject data collection in an indoor laboratory setting.

1.5.3 Comprehensive Motion-to-Event Simulator

As Fig. 1.3 shows, our proposed motion-to-event generator takes 3D character models, motion files, camera views, and lighting conditions as inputs. MMD renders RGB dance videos and their paired human mask videos given these inputs, while Blender generates corresponding ground truths. For each camera view, a camera intrinsic and extrinsic matrix is calculated by

Blender as well.

Then our simulator combines these rendered dance videos with collected background videos by referring to the paired mask videos. According to [69], if a video’s temporal resolution is low, generated event stream will be less realistic. However, due to the software limitation and background video quality, the synthesized videos are at 60 FPS. This gap in FPS is compensated using SuperSlowMo[73], which can interpolate videos to a high FPS with convincing results. To reduce the time and computation cost, we only apply SuperSlowMo to dance and background videos before the merging. This way, we do not have to interpolate synthesized videos for each human and background combination.

After the RGB video rendering, merged videos are sent to the V2E events stream generator. Many parameters such as event trigger thresholds, noise level, and slow-motion interpolation scale, can be modified in detail. These features enable us to generate many highly customizable DVS event streams at a meager cost in a short time. To simulate situations in the real world, we increase the noise as the brightness decreases.

For human joints’ ground truth, as mentioned above, we write our customized scripts and inject them into Blender to collect the exact position of all joints while rendering scenes at 300 FPS. Also, scripts help extract the camera’s intrinsic and extrinsic matrix used in the label pre-processing. Besides the advantages mentioned above, 3D human models have even more initial flexibility. Skin color, height, body style, clothing, hair color, style, and accessories are all easily modifiable - which is very difficult to do in real-world data collection.

1.5.4 Synthetic Dataset Description

Utilizing the comprehensive motion-to-event simulator, we have generated a vast synthetic dataset *Yelan-Syn-Dataset*, comprising approximately 4 million data samples (specifically, 3,958,169 frames). Examples of synthetically generated RGB frames are shown in Fig. 1.5. The total dataset size is around 2.6 terabytes. This data was synthesized from 1320 combinations of various variables, including 10 human models, 8 one-minute dance motions, 11 background

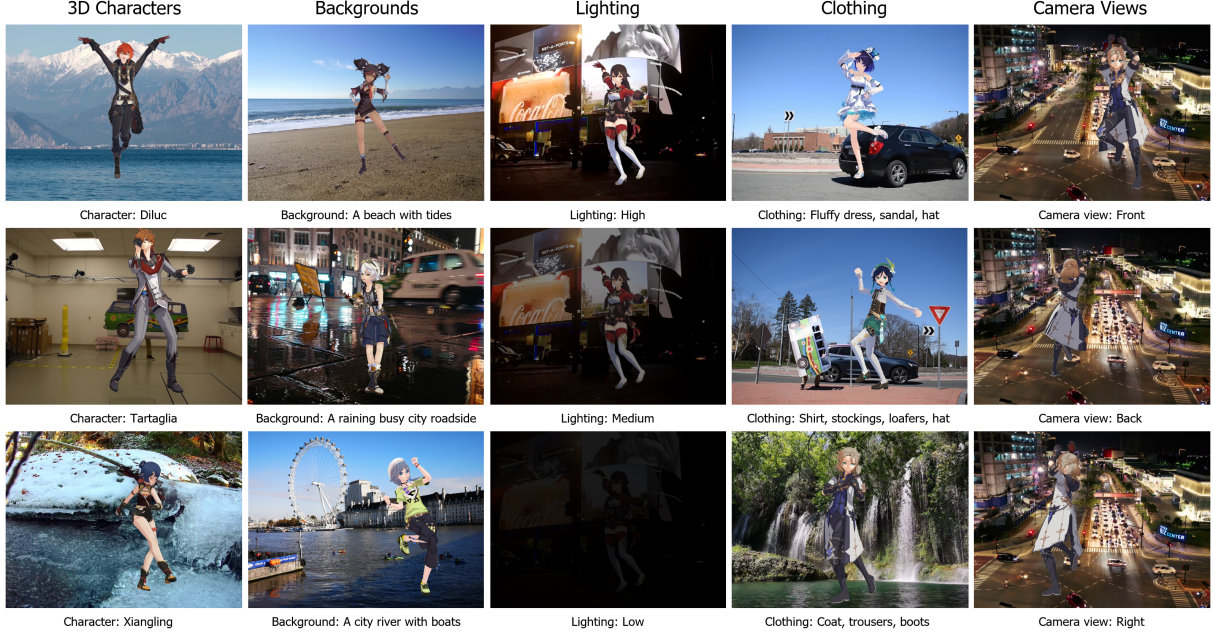


Figure 1.5. RGB frame samples from the synthetic data generated by the comprehensive motion-to-event simulator.

videos, 4 camera views (i.e., front, back, left, right), and 3 different lighting conditions (i.e., high, medium, low). This dataset contains processed TORE volume, paired labels and masks, constant-time frames with identical time steps (20ms), and raw event stream files for generating any other customized representations. Due to computational resource and training time constraints, only 30% of data instances from 330 selected setting combinations are employed for training and validation. They are shuffled and partitioned with an 8:2 ratio. Subsequently, we randomly select 82 setting combinations from the remaining 990 unused combinations as the test set.

1.6 Human Subject Study with Dynamic Lighting and Background

1.6.1 Real-World Dataset in Motion Capture Facility

In addition to the synthetic dataset *Yelan-Syn-Dataset*, we have conducted an Institutional Review Board (IRB)-approved human subject study to collect a real-world human pose dataset, *Yelan-Real-Dataset*, from nine participants. Seven of the participants are male and two are

female. The participants, aged 20-31, were recruited from a university campus using a snowball sampling technique. During this study, our participants were asked to play the Nintendo dance game “Just Dance 2021”. Each participant selected five songs to dance to during the study after a brief training period with the tutorial dances in the Nintendo dance game. A monitor provided further guidance/cues for participants to follow along.

A 9-camera-based 3D motion capture system (Qualisys AB, Göteborg, Sweden) was used to obtain ground truth 3D kinematics data of the human body, providing the 3D absolute coordinates for all selected human joints at a frame rate of 200 frames per second (FPS). A neuromorphic camera (DAVIS 346 [20]) and an RGB camera were simultaneously operated to record the participants’ movements. Other essential equipment included a flicker-free LED light, IR filter, cardboard background, monitor, and Nintendo Switch. During the dataset collection, the lighting condition was strictly controlled to induce low-lighting and high-lighting conditions. All the lights except a dim flicker-free lamp were turned off during the low-lighting conditions session. The IR filter was attached in front of the neuromorphic camera’s lens to filter out the events caused by IR light emitted by the motion capture system. Fig. 1.4 illustrates the data collection settings, hardware, sensor arrangements, and mechanisms to generate background noise.

In the dynamic background condition, a person behind the target participant randomly moved with a large cardboard vehicle. On the other hand, for static background cases, no other movable contents appear in the background during recording. On average, each participant danced for about 20 minutes, and the time was distributed equally to the four conditions mentioned above. Static background cases are only included for result comparison, as mask-based background filtering is not helpful in these cases. We divided *Yelan-Real-Dataset* simply by selecting all the events generated by participants seven (male) and eight (female) as test sets. All other data is shuffled and divided into training and validation sets with the same 8:2 ratio. We prepared a short video to further illustrate the rich synthetic and real-world data, which can be found at bit.ly/yelan-research.

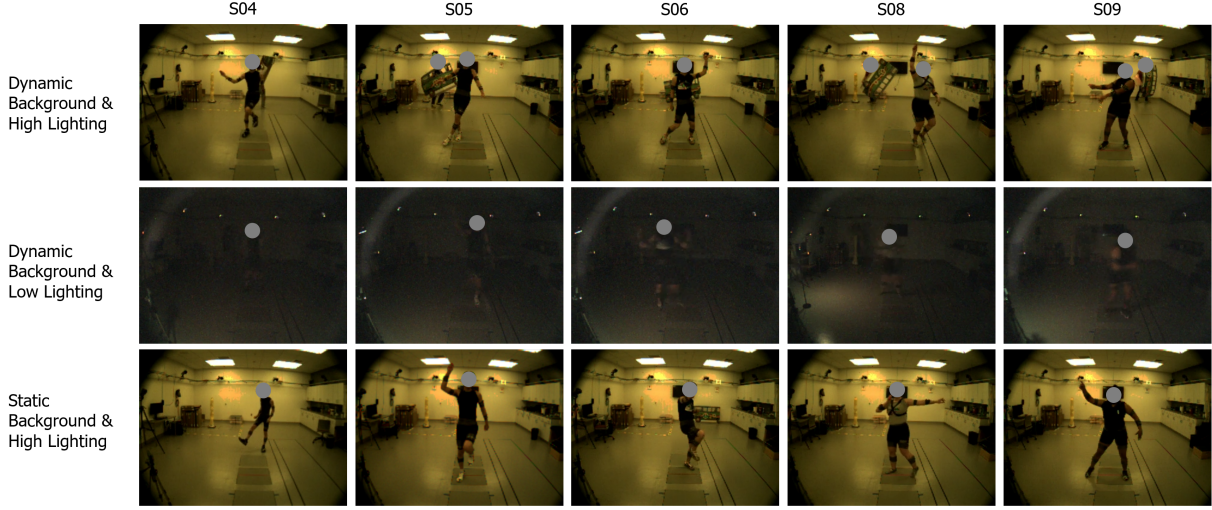


Figure 1.6. RGB frame samples from the *Yelan-Real-Dataset* collected in motion capture laboratory.

1.7 Results

1.7.1 Training Details

In this study, all models are trained using eight 1080ti or 2080ti graphics cards. The batch size remains consistent throughout the training process: stage one employs a batch size of 128, while stage two utilizes a batch size of 16. Training parameters for all stages include a learning rate of 0.001, an Adam optimizer with a weight decay of $1e-5$, and early stopping with a patience of 10 epochs. A learning rate scheduler is implemented to reduce the learning rate by half every N epochs, with N set to 5 in stage one and 10 in stage two to account for differences in training patterns and convergence speeds. This project is executed using Pytorch-Lightning [44]. *YeLan*’s stages one and two are trained separately. Owing to ground truth limitations, the mask prediction network is trained exclusively on synthetic data and applied directly in both *Yelan-Syn-Dataset* and *Yelan-Real-Dataset*. Stage two is initially pretrained on *Yelan-Syn-Dataset* and subsequently fine-tuned on *Yelan-Real-Dataset*.

1.7.2 Evaluation Metrics

In accordance with convention, the primary evaluation metric employed in this study is the Mean Per-Joint Position Error (MPJPE), a widely used metric in HPE. Additionally, two other popular metrics, PCK and AUC, are considered in the main comparison. The MPJPE is calculated by averaging the Euclidean distance between each predicted joint and its corresponding ground-truth joint, typically measured in millimeters. PCK represents the Percentage of Correct Keypoints according to a commonly used threshold value of 150mm. A predicted joint is considered correct and given a value of 1 if it is within a 150mm cube centered on the ground truth joint; otherwise, it is deemed incorrect and assigned a value of 0. AUC refers to the Area Under the Curve for the PCK metric at varying thresholds. Standard threshold sets, consisting of 30 evenly spaced numbers from 0 to 500mm, are used in this study. The target AUC is determined by first calculating the PCK at all threshold values and then computing their mean value.

$$MPJPE = \frac{1}{J} \sum_i^J ||p_{xyz} - \hat{p}_{xyz}|| \quad (1.12)$$

$$PCK_{\alpha} = \frac{1}{J} \sum_i^J \text{sign}(\alpha - ||p_{xyz} - \hat{p}_{xyz}||) \quad (1.13)$$

$$AUC = \frac{1}{N} \sum_{\alpha=0}^A PCK_{\alpha}(p_{xyz}, \hat{p}_{xyz}) \quad (1.14)$$

joint positions, respectively, while J represents the number of skeleton joints. For PCK and AUC calculations, α corresponds to the threshold and A refers to the maximum threshold. N in AUC indicates the number of different thresholds utilized.

In the present study, we selected baseline methods introduced by Scarpellini et al. [133] and Baldwin et al. [12], both initially published in 2021. Scarpellini et al. employed two event-count-based representations—constant event count frames and voxel grid—that exhibit a

variable time step, causing a loss of synchronization with our labels and TORE volumes. As a result, we adapted the original work to accept constant time representations as input, setting the time step for this representation at 20 ms to ensure accurate label matching.

Baldwin et al. presented the TORE volume and evaluated it using the model and dataset proposed by Calabrese et al. [22], employing an alternative representation to demonstrate the superior performance of their novel representation in the multi-view human pose estimation task. With the goal of developing a monocular human pose estimation framework in this paper, we utilized the same monocular HPE model from Scarpellini et al. [133], substituting the representation with TORE to showcase the efficacy of Baldwin et al.’s [12] method in this new task.

Alongside the hourglass-like HPE backbones selected for this study, we also explored other commonly used structures, such as residual block-based and bottleneck block-based backbones [60]. Throughout these experiments, we maintained a consistent number of blocks for all designs and kept all other settings and conditions constant.

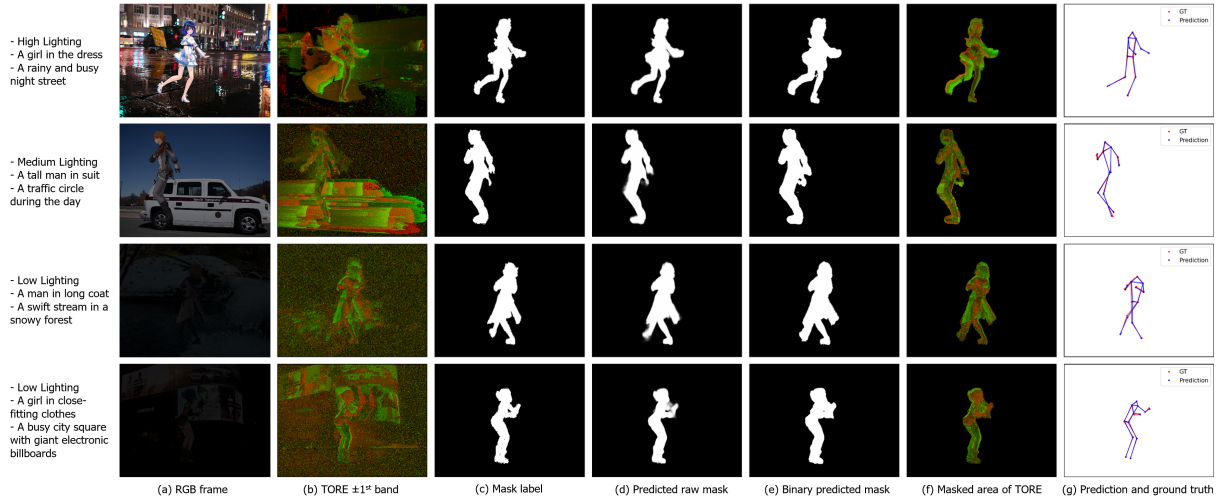


Figure 1.7. Sample results from *Yelan-Syn-Dataset* in different lighting conditions with dynamic backgrounds. The RGB frames, corresponding event representation TORE, generated masks, ground truth, and predicted 3D human pose are shown.

Table 1.2. Test set results comparison on three metrics for *Yelan-Syn-Dataset* and *Yelan-Real-Dataset*. †: P means YeLan with pre-training on Yelan-Syn-Dataset and tuned on Yelan-Real-Dataset. HG: Hourglass-like blocks backbone. RES: Residual backbone. BTN: Bottleneck backbone.

Methods	Representation	Yelan-Syn-Dataset			Yelan-Real-Dataset		
		MPJPE↓	PCK(%)↑	AUC(%)↑	MPJPE	PCK	AUC
Scarpellini et al., 2021	Constant Time	91.88	83.92	80.95	111.57	78.12	77.03
Baldwin et al., 2021	TORÉ	59.34	93.17	86.97	101.97	82.22	78.90
YeLan - BTN	TORÉ	126.50	84.70	81.49	137.21	69.74	72.47
YeLan - RES	TORÉ	58.95	93.51	86.97	112.49	77.76	76.88
YeLan - HG	TORÉ	46.57	96.34	89.37	96.61	81.88	79.78
YeLan - HG - P†	TORÉ	-	-	-	90.94	85.14	80.91

1.7.3 Evaluation

Table 1.2 illustrates the performance comparison between the proposed *YeLan* and baseline models on both *Yelan-Syn-Dataset* and *Yelan-Real-Dataset*. The table shows that the baseline models consistently underperform the proposed YeLan model in both conditions. In *Yelan-Syn-Dataset*, the model proposed by the Scarpellini et al. [133] achieves an MPJPE and PCK of respectively 91.88 and 83.92% while the model proposed by the Baldwin et al. [12] with TORÉ representations have a slightly improved performance of respectively 59.34 MPJPE and 93.17% PCK. The proposed *YeLan* system outperforms the baseline models by a significant margin and achieves the lowest MPJPE of 46.57 and the highest PCK of 96.34% on *Yelan-Syn-Dataset*.

Compared to the synthetic data, the performance of all the models on the real-world data is worse. The real-world data comes with its own challenges and unique characteristics, like more noise and limited transmission bandwidth. Compared to the perfect ground truth labels generated in the synthetic data, some non-negligible spatial and temporal mismatch happens in the real-world data, especially considering the fact that the background activities often block some IR cameras used for motion capture. There is also substantial heterogeneity in the skeletal formations and impedance-matching states in the real-world data. Lastly, in the real-world data, another researcher generating background activities are also in the camera’s view, who is

sometimes recognized and masked as an additional human and confuse the models as a result. Consequently, the performance of the real-world data will be expected to be lower than that of the synthetic data. However, the proposed *YeLan* still achieves the lowest metrics compared to the baseline models. The baseline models are designed to work in ideal environments which do not suffer from dynamic lighting conditions and moving background content problems, which contributes to their poor performance in more realistic environments.

1.7.4 Result Statistics and Analysis

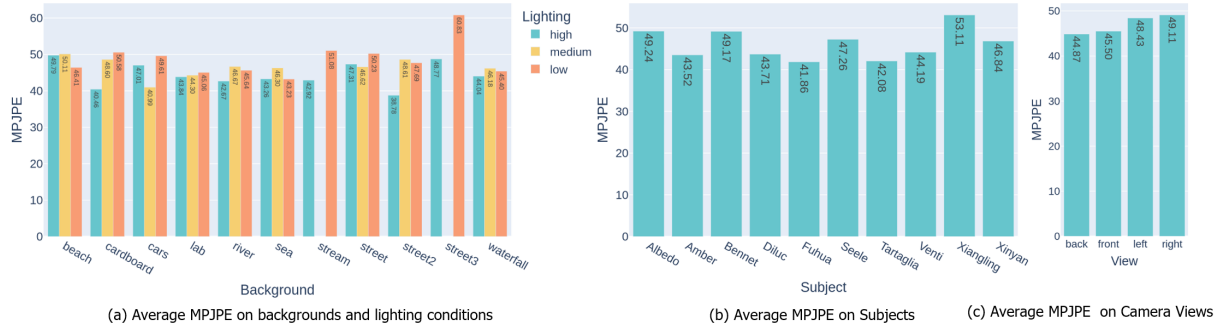


Figure 1.8. The performance across lighting conditions, dynamic backgrounds, subjects, and camera views in our synthetic data.

As Fig. 1.8 shows, *YeLan* has strong stability across the changes in lighting conditions and camera views, while more complicated background contents cast an impact on the results. Different human models also show an impact on performance. From the performance statistics, we observe that the system performs less well for human models “Albedo”, “Seele”, and “Xiangling”. By looking at the original 3D model and test data composition, the reason becomes obvious: Albedo wears a long coat reaching his knee, while Seele wears a fluffy dress with a complex structure. These factors make their joint position harder to estimate. As for Xiangling, though she does not have clothing-related problems, we find some body deformation that happened during the simulation. Also, the randomly selected test cases for Xiangling contain more difficult backgrounds and lighting combinations, such as two busy city street environments in low-lighting conditions, according to Fig. 1.8 (a).

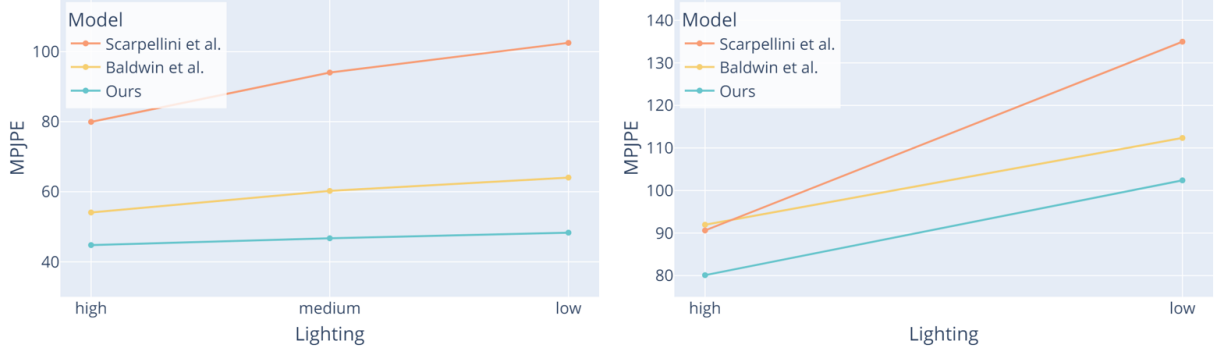


Figure 1.9. Impact of lighting condition on the performance (MPJPE) on our synthetic (left) and real-world (right) dataset.

Moreover, Fig. 1.9 shows that our model behaves better consistently in all lighting conditions compared to baseline models. This feature is significant in real-world applications as the pipeline could generate convincing predictions regardless of the lighting conditions.

The mask prediction network in *YeLan* enables it to reach a similar level of PCK and AUC whether the background is dynamic or static. We compared the real-world test set with dynamic and static background cases using the model pretrained on the *Yelan-Syn-Dataset* and fine-tuned it on *Yelan-Real-Dataset*. Fig. 1.10 depicts that *YeLan* get a comparable result in both scenarios.

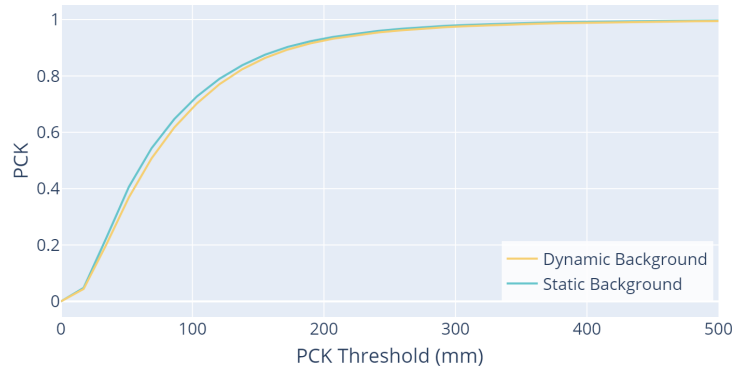


Figure 1.10. AUC curve with dynamic and static background on *Yelan-Real-Dataset*.

1.7.5 Impact of Pretraining on Synthetic Data

Synthetic data from the physics-based simulator is a core contribution of this work which allows us to generate event streams from diverse virtual 3D characters in different simulated settings. They include different clothing, lighting conditions, camera viewing angles, dynamic backgrounds, and movement sequences. The generated synthetic data is faithful to physics and can provide essential data for pretraining *YeLan* since running real-world experiments with a large number of participants is expensive and time-consuming. To this end, we hypothesize that the pretraining on synthetic data will allow *YeLan* to learn better feature representation of physical conditions in diverse simulated settings and achieve superior performance on the *Yelan-Real-Dataset*.

Table 1.2 demonstrates that our proposed model achieves superior performance (comparing MPJPE/PCK/AUC metrics) on the *Yelan-Real-Dataset* if the model is pretrained on *Yelan-Syn-Dataset* from the simulator. The model trained only on the *Yelan-Real-Dataset* achieves 96.61 MPJPE and 81.88% PCK. On the other hand, if the model is pretrained on *Yelan-Syn-Dataset* and then fine-tuned on the *Yelan-Real-Dataset*, it achieves 90.94 MPJPE and 85.14% PCK, which clearly shows the benefit of pretraining on synthetic data from a simulator.

1.7.6 Ablation Study

As introduced in section 1.4, there are two most essential modules in *YeLan*: a mask prediction network and the BiConvLSTM. We do an ablation study by removing one module each time and comparing their performance on *Yelan-Syn-Dataset*. If the mask prediction network is removed, the resulting MPJPE is 63.99; if the BiConvLSTM is removed, the corresponding MPJPE is 49.08. In contrast, the complete *YeLan* pipeline gets an MPJPE of 46.57, which proves the effectiveness of these modules.

1.7.7 Occlusion

In the real world, occlusion is also an inevitable problem. In *Yelan-Syn-Dataset*, different types of clothes and accessories like hats, long hair, fluffy skirts, and long coats already occluded the human body to a relatively large extent. Moreover, the camera’s side views also introduced many self-occlusion. The excellent performance over all these scenarios shows a solid ability to survive occlusion.

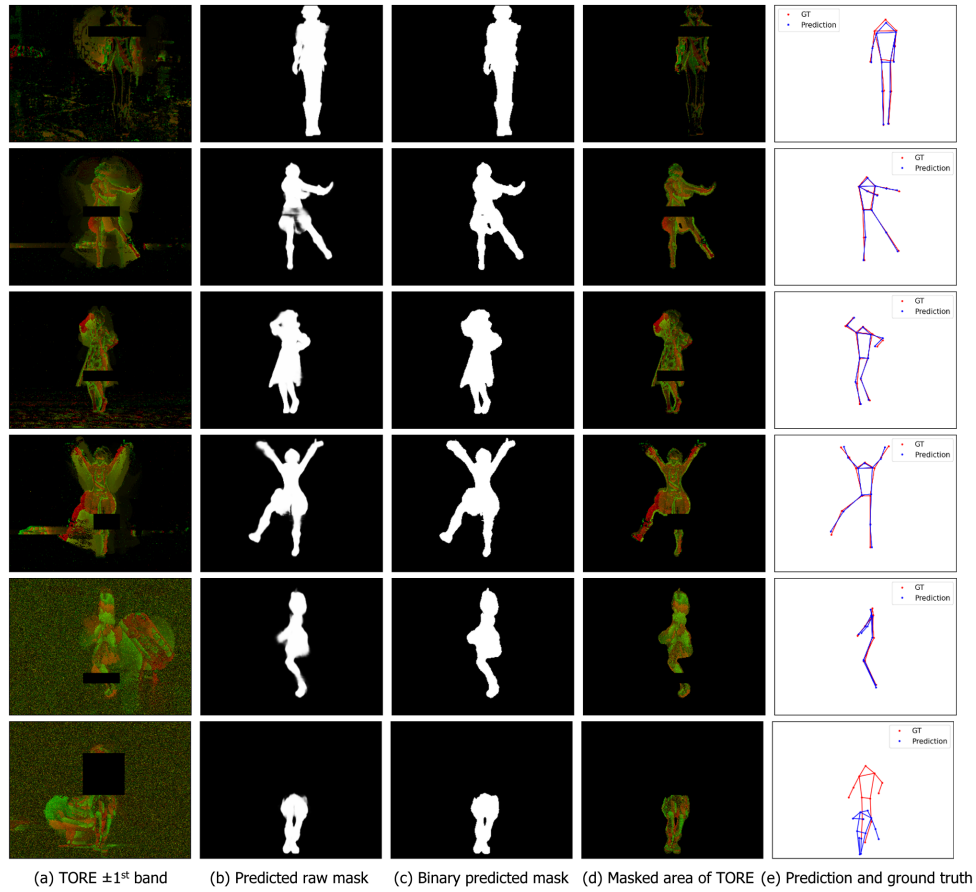


Figure 1.11. Samples of occluded inputs, predicted mask, TORE volume representation, and inferred human poses.

To further prove the occlusion resiliency of the system, we augmented the dataset with more block occlusions. In the eye of the neuromorphic camera, if a static object is placed in front of a human and shadows him, the corresponding area shoots no event due to the object’s immobility. Regarding the input TORE volume, the occluded area becomes pure black, as no

event activity is recorded. To simulate the occlusion like this, we trained a model with the value of random areas set to zero. These rectangular occlusion areas have random sizes and locations, and the occlusion is also applied randomly with a probability of 80% during the training. When testing on the test set with random occlusion enabled, the MPJPE is 96.794. During the test, the occlusion probability is set to 100%. There is an accuracy drop, but considering that the maximum random occlusion area is 80×80 (where the frame size is 260×346), it makes sense as sometimes the majority of the human body could be occluded. Fig. 1.11 shows samples from the test set.

1.7.8 Systems Benchmarking

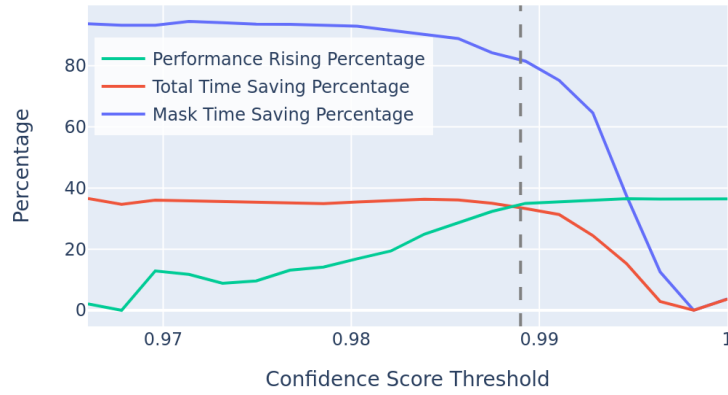


Figure 1.12. Time-saving and performance rising percentage with regard to confidence score threshold.

As is introduced in 1.4.2, the first stage of *YeLan* is an early-exit-style human mask prediction network, where a threshold β is used to decide when to start a new inference. The selection of β is a trade-off. If the threshold is set too high, the mask prediction process will be executed too many times, resulting in a longer inference time. On the contrary, many defective masks will be used if the threshold is too low, which harms the overall accuracy. In order to select the best threshold, we ran an experiment to do human pose estimation on a continuous one-minute event stream with a series of different thresholds. The accuracy and time consumption are recorded and made into Fig. 1.12. From this figure, we can observe that as we increase the

threshold, accuracy goes up and time consumption goes down, while we can achieve a balanced accuracy and time consumption at near 0.99.

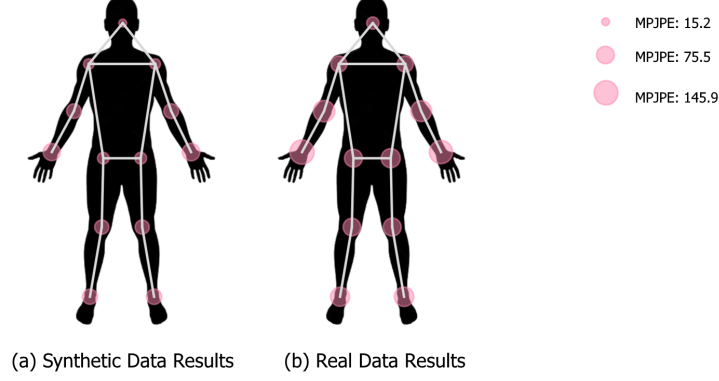


Figure 1.13. Joint-wise average MPJPE analysis on *Yelan-Syn-Dataset*.

Furthermore, to comprehend the error distribution among various joints, we computed the mean MPJPE for all 13 joints. As illustrated in Fig. 1.13, the head joint exhibits the lowest MPJPE, while the shoulder joints possess the second-lowest error. Conversely, the left and right hands exhibit the highest MPJPE, with foot joints following closely. Our observations indicate that this occurs because the head and shoulders maintain the most stable contours across distinct characters and exhibit minimal overlap with other body parts. Nevertheless, as hands and feet are situated farthest from the human center and exhibit the most movement during dancing, they prove more challenging to estimate due to an increased range of potential movements and patterns. Greater pattern variability contributes to increased uncertainty, which further undermines accuracy when occlusion occurs.

The total model size of *YeLan* amounts to 234 MB, with the total parameter count reaching 61.2M, of which the stage one model comprises 40.8M and stage two encompasses 20.4M. The total FLOP stands at 1482G. In terms of time consumption, we conducted a comprehensive test on the MiDHP test set using a batch size of one, resulting in an average time expenditure of approximately 29.1 ms per frame when executed on a 2080ti graphics card.

1.8 Comparison between Different Modalities

1.8.1 RGB Camera

In order to show the superiority of DVS in low-lighting conditions, we also compare our result with RGB frame-based human pose estimation algorithm. We use the DVS camera DAVIS346, where both events and RGB frames are synchronously recorded. As the same device records both modalities via the same lens, there is no difference in the quantity of light captured by RGB and DVS. OpenPose [23] proposed and implemented by Cao et al. is adopted as our RGB baseline. OpenPose is an accurate, fast, and robust 2D human pose estimation algorithm. Although there are differences in joint number and dimension, we calculated the 2D projection of 3D joints generated by *YeLan*. We picked the closest 13 joints from all 25 OpenPose output joints to compare. The comparison is conducted on paired DVS and RGB recordings from the dataset collected in the motion capture lab. Both high and low-lighting condition cases are included, and qualitative comparison is shown in Fig. 1.14. Furthermore, we made a box plot showing the 2D MPJPE of DVS and RGB in two lighting conditions in 1.15. From this figure, it is evident that *YeLan* generates better and more stable predictions in all cases. Although the RGB-based method achieves good performance (marked by the low 2D MPJPE) in high-lighting conditions, the performance deteriorates significantly in the low lighting conditions. OpenPose fails to detect any human from frames due to motion blur and low SNR sensor data when the illumination is lower than a certain threshold. On the other hand, *YeLan* shows strong robustness against lighting conditions changes and constantly generates high-quality predictions (marked by the low MPJPE in both cases).

1.8.2 RGB-Depth Camera

Besides the RGB camera, the depth camera is also widely used in digital dancing games. [170, 79] These cameras are usually paired with RGB cameras, which grant them information from both domains. Compared to RGB cameras, these cameras have a better understanding

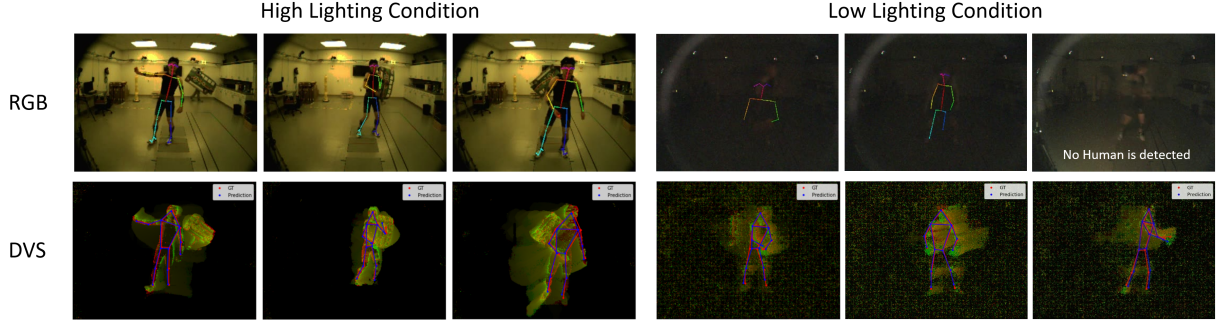


Figure 1.14. HPE comparison between RGB and DVS in both high and low-lighting conditions. RGB and DVS are synchronized.

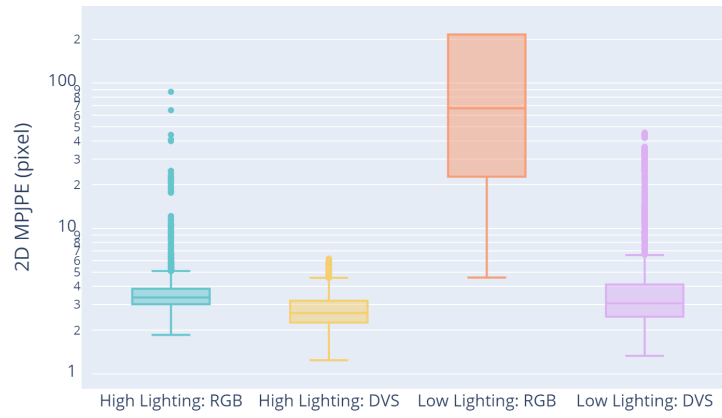


Figure 1.15. 2D MPJPE Comparison between DVS and RGB.

of the 3D space, which results in a more accurate human segmentation and joint location estimation. RGBD camera-based human pose estimation is a well-established problem with many commercial products and pipelines [121, 83, 5], like the Kinect from Microsoft and the RealSense from Intel.

However, the depth camera also has several disadvantages. Depth cameras actively emit and recapture the reflected infrared to build the 3D point cloud. This procedure is significantly more power-hungry (about 200 times higher power consumption than the neuromorphic camera) and suffers from many limitations. Firstly, due to the manufacturing and power consumption consideration, depth cameras' field of view (FOV) is usually fixed and small. Small FOV limits the sensing space coverage and restricts the dancer's movement in a smaller space. Secondly, the distance between the target and the depth camera strongly impacts the detection accuracy. If the

Table 1.3. The comparison between RGB, RGBD, and neuromorphic cameras. For FOV, H, V, and D stand for horizontal, vertical, and diagonal FOV.

Camera Type	RGB (In DAVIS 346)	RGBD (Intel RealSense 435i)	DVS (DAVIS346)
Update Rate	40 Frames / sec	30 Frames / sec	<12 MEvents / sec
FOV	Any large, Dependent on lens. Lens are replaceable.	IR Projector: Depth: H: 90 $\pm$ 3 H: 87 V: 63 $\pm$ 3 V: 58 D: 99 $\pm$ 3 D: 95 FOVs are fixed.	Any large, Dependent on lens. Lens are replaceable.
Dynamic Range	56.7 dB	N/A	120 dB
Power Consumption	140mW	Maximum Power: 2850mW (Measured on Windows 10)	10-30mW (activity dependent)
Working Distance	N/A	0.2~3m, varies with lighting conditions.	N/A

sensor is too close to the target (smaller than 0.5m), the two IR receivers have overlapped IR patterns which saturate the IR camera and result in an estimation failure [77]. Moreover, when the distance is far, the detection rate drops drastically after a certain point, as the IR receiver cannot receive enough reflected IR light. The first two points together limit the functional dancing area and cause a substantial restriction on relative distance.

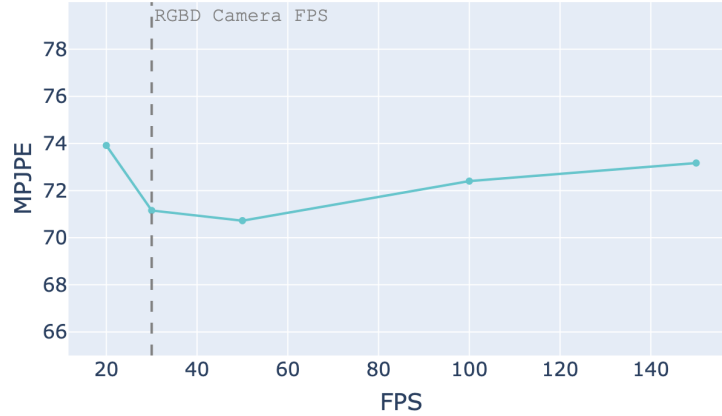


Figure 1.16. Stable human pose estimation of the neuromorphic camera on different FPS settings. The model used is only trained on the 50FPS setting. On the contrary, the depth camera has a fixed 30FPS frame rate.

Thirdly, since the camera emits light itself, it could also be interfered with by other light sources, which is especially true for strong light sources like stage lights and solar lights. This characteristic restricts the application of depth cameras in outdoor and indoor spaces near the

light source.

Fourthly, as we have previously mentioned, our system is able to work at any high FPS, and the only procedure needed to take is to modify the time step size when generating the TORE volumes for an event stream. Compared with the neuromorphic camera, the maximum RGBD camera FPS is usually very low, making it hard to be applied to track fast-paced dances or generate high-fidelity 3D digital dances. The two most commonly used RGBD cameras, i.e., Microsoft Azure Kinect and Intel RealSense, support at most 30 FPS when capturing full RGBD streams. On the contrary, neuromorphic cameras can easily receive ten-millions level events per second, which gives them overwhelmingly huge advantages over RGBD cameras. We tested our model by generating representations from 20FPS to 150FPS (due to the restriction of ground truth label rate) on all the event streams from test subject eight. The result shows that the estimation accuracy is pretty stable on various frame rate settings (Fig. 1.16).

Although it would be ideal to quantitatively compare RGBD and neuromorphic cameras for human pose estimation, these two types of cameras can not work simultaneously. When the depth camera is turned on, its built-in IR projector emits a grid of IR rays 30 times per second. In the eye of the neuromorphic camera, the IR projection of the depth camera is visible and everything in the surrounding is constantly flickering at a high frequency with meshed dots, which negatively impacts the performance. Therefore, we made a qualitative comparison by separately recording two sessions of human dance by Intel RealSense 435i and DAVIS 346. During these two sessions, the participant moves from 2.5m to 3.5m with some dynamic dance patterns, and the results are shown in Fig. 1.17. As the figure shows, as the distance between the camera and the human increases, the RGB-Depth camera gradually fails to capture valid human shapes. The human pose estimation program also stops generating valid predictions after a certain point. For the RGB-Depth camera-based 3D human pose estimation, we chose the NuiTrack [113], a commercial product designed specifically for RGBD camera-based HPE problems, for comparison. Although RGB and depth channels are all used, here, for visualization convenience, only the depth channel is shown, and the estimated human masks are shown in

green.

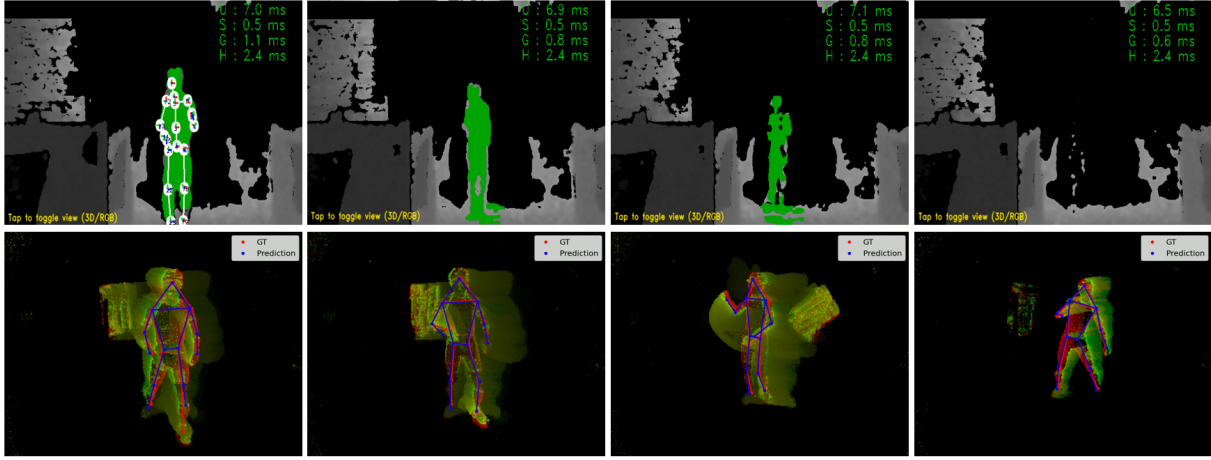


Figure 1.17. Qualitative comparison between the RGB-Depth camera and YeLan in terms of the distance between the camera and the human subject. The first row displays results from the depth camera, while the second row presents outcomes from YeLan. The distance increases from left to right.

1.9 Limitation and Future Works

Overall, YeLan clearly highlights the promise of the 3D human pose estimation in the context of dancing in the presence of different confounding factors. However, the present study exhibits several limitations that we intend to systematically address in future work. In the current work, one assumption was that the neuromorphic camera is stationary and attached to a fixed tripod. Consequently, YeLan is not readily transferable to scenarios where the neuromorphic camera can move and is mounted on a car or a drone. The current version of YeLan has been primarily tested in a single-person scenario and does not support a multiplayer dancing game. By incorporating multi-person segmentation and masking, we plan to extend YeLan for multi-person situations. Additionally, we aim to investigate energy-efficient implementations on neuromorphic computing platforms such as Intel Loihi 2. [101].

1.10 Conclusion

This work discussed existing 3D HPE techniques employed in dance games and evaluated their strengths and limitations. We proposed an innovative neuromorphic camera-based approach to address these shortcomings. We amassed a real-world dance dataset through human subject studies and constructed an extensive motion-to-event simulator to generate a vast amount of fully controllable, customizable, and labeled synthetic dance data to aid in pre-training the model. YeLan surpasses all baseline models in various challenging scenarios on both datasets. Furthermore, we conducted an in-depth analysis and comparison between different modalities, which unequivocally demonstrates YeLan’s superiority in many aspects.

Acknowledgments

Chapter 1, in full, is based on the paper from Zhongyang Zhang, Kaidong Chai, Haowen Yu, Ramzi Majaj, Francesca Walsh, Edward Wang, Upal Mahbub, Hava Siegelmann, Donghyun Kim, and Tauhidur Rahman, “Neuromorphic high-frequency 3D dancing pose estimation in dynamic environment” published in *Neurocomputing*, Volume 547, 2023, 126388, ISSN 0925-2312. The dissertation author is the primary investigator and author of this paper.

Chapter 2

V2CE: Video to Continuous Events Simulator

2.1 Introduction

Neuromorphic cameras, also referred to as Dynamic Vision Sensors (DVS) or event cameras, have recently emerged as a significant area of interest in the field of robotics [132, 178, 90] and computer vision [52, 26, 10]. The exceptionally high optical event capture rate, high dynamic range, low yet adaptive power consumption, sparse output, and a dynamic vision scheme akin to mammalian perception contribute to their success in various computer vision applications [49, 124, 104]. These include feature tracking [136, 41], optical flow estimation [116, 13], as well as gesture and human pose estimation [173, 99]. Characteristically, DVS-based approaches tend to offer superior temporal resolution and quicker inference speeds.

Nevertheless, when compared with the Active Pixel Sensor (APS, standard RGB camera), the DVS emerges as a relatively novel vision sensor. Also, compared to APS frames, labeling DVS data is quite challenging, as events captured are sparse, and inactivative objects trigger few events. Consequently, there is a scarcity of large-scale annotated DVS datasets, which are significantly harder to procure than APS data. Unlike APS data that can be readily generated by mobile devices and obtained from the internet, DVS data collection necessitates specific hardware such as a DVS camera and a laptop. Moreover, dataset collection typically proves to be time-consuming and expensive, thereby posing significant challenges to the acquisition

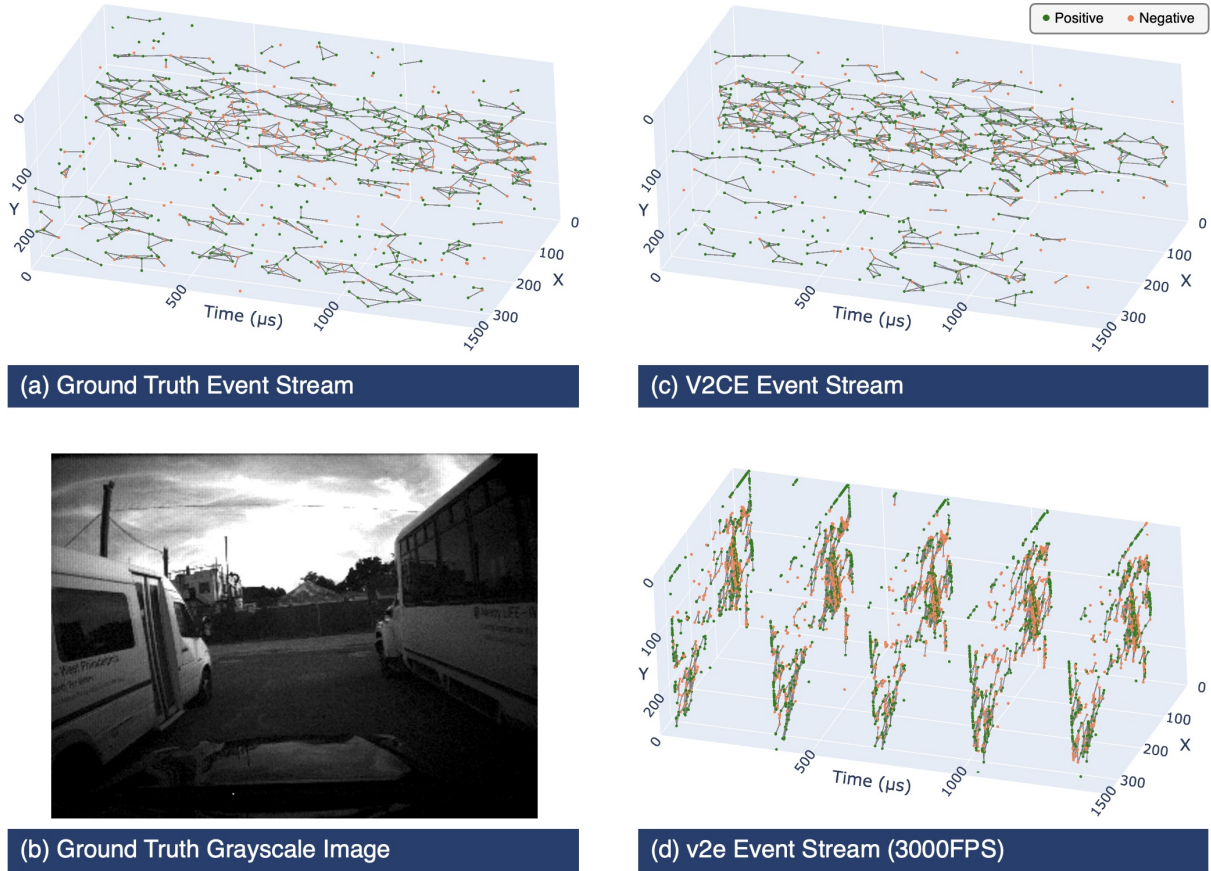


Figure 2.1. K-nearest neighbor graph comparison from events in (x, y, t) space. The v2e event stream is generated with input video upsampled to 3000FPS.

of large-scale DVS datasets. Finally, it is neither practical nor cost-effective to recreate every existing APS dataset for DVS. In this work, we present an optimized video-to-event conversion algorithm that can effectively mimic the nonlinear characteristics of a DVS camera with high fidelity.

There are a few existing works trying to bridge the gap between the APS frames and events, like ESIM [122, 47], v2e [69], and EventGAN [180]. These methods can be roughly divided into two genres: rule-based [69, 122, 47] and learning-based [180]. However, the former abandoned recovering the lost information due to the dynamic range gap between standard APS and DVS [75, 97], while the latter doesn't consider the characteristics difference between these two types of cameras. Lastly, none of these previous works have ever discussed the last mile

problem: how to convert the generated event voxels or the events number into realistic and accurate raw event streams. All these works directly apply either random or even sampling, which is clearly suboptimal.

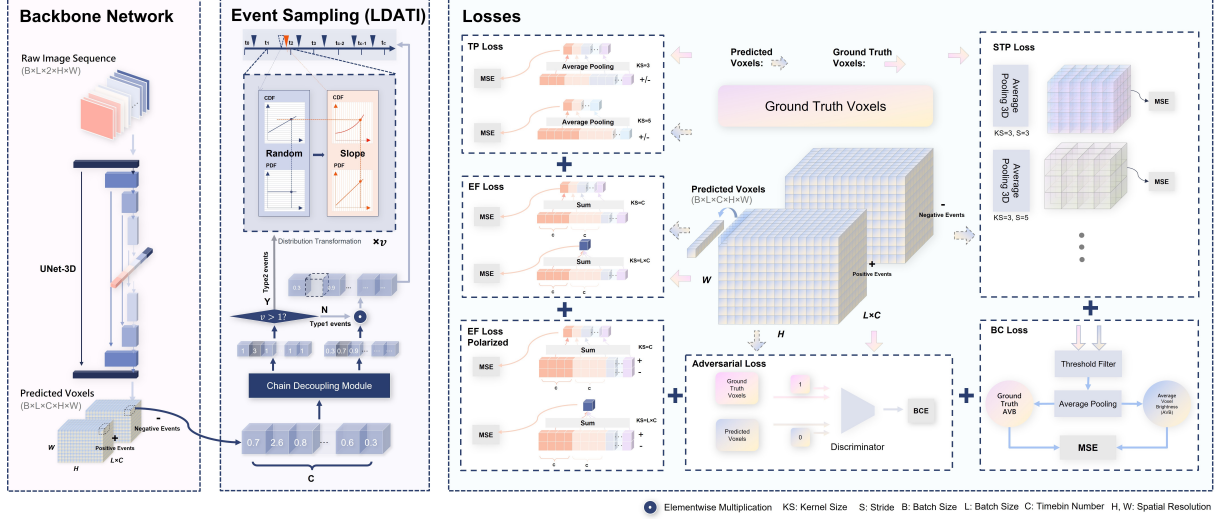


Figure 2.2. Proposed motion-aware event voxel prediction pipeline and hybrid loss structure: Our method consists of two main stages. The Backbone 3D UNet encodes input frame pair sequences and generates event frames. The Event Sampling Module, subdivided into chain decoupling and distribution transformation modules, calculates event counts and in-voxel time, then redistributes events in Type2 voxels. The loss functions for training, displayed on the right, include STP, TP, ADV, BC, and EF Losses, which are elaborated in Section 2.2.1.

Notably, while these methodologies successfully convert videos to events, the resulting events continue to reside in a series of discrete temporal layers. Fig. 2.1 illustrates a 3D visualization for ground truth events and events generated by V2CE and v2e [69]. For v2e, It's apparent that all the generated events share a series of discrete timestamps, instead of spreading across the time axis in a continuous fashion like real DVS recordings. This discrepancy is often negligible when temporal accumulation-based methods are utilized in subsequent task preprocessing, as the temporal information is collapsed anyway. However, for tasks that are sensitive to timestamps distribution, such as Graph Neural Network (GNN) [134, 135, 141, 94] and Spiking Neural Network (SNN) [145, 39, 34, 183], this issue could prohibit using generated synthetic data as pretraining dataset, since these data has a significant domain shift compared to

real events.

In this paper, we introduce a new Video-to-Continuous Events (V2CE) framework that tackles video-to-event conversion challenges via a two-stage pipeline. Our main contributions include:

1. We introduce a specialized suite of loss functions tailored for the video-to-event voxel task, thereby achieving SOTA performance.
2. We develop a novel statistics-based local dynamics-aware timestamp inference algorithm that enables the smooth transition from event voxels to event streams, outperforming existing baseline methods.
3. We establish the first set of metrics grounded in DVS event characteristics, allowing for robust quantitative evaluation in both the video-to-event voxel and the voxel-to-event stream phases.
4. Through rigorous evaluation against established baselines, we demonstrate that V2CE significantly outperforms them across all metrics, and the simulated events' count strictly matches the ground truth. Our comprehensive results analysis further underscores that V2CE is not only the SOTA but also the first option for generating continuous event streams.

2.2 Proposed System

The proposed Video to Continuous Events (V2CE) simulator pipeline (outlined in Fig. 2.2) consists of two stages: motion-aware event voxels prediction, and voxels to continuous events sampling.

2.2.1 Stage1: Motion-Aware Event Voxel Prediction

The Motion-Aware Event Voxel prediction aims to transform the APS video into a 3D voxel grid where the video data is temporally upsampled. The temporal resolution is increased

by a significant margin and the event sequence is represented in a spatio-temporal xyt coordinate system. We use a simple UNet-based architecture to design this module.

However, the main challenge of this task is to preserve the temporal continuity and the microstructure compatibility of event voxels. High-fidelity event voxel reconstruction requires information about nonlinear dynamics of light intensity changes and object movements (e.g., acceleration or higher order moment). While any linear assumption invariably leads to suboptimal video-to-event conversion performance, all prior work in this space only used an adjacent frame pair to infer the events between them. Since no hint is available to infer the nonlinear dynamics, all baseline methods were essentially doing linear interpolation between the input APS frame pair.

Therefore, we advocate using longer frame sequence instead of frame pairs to serve as the input of the model, and help local temporal information flow properly during the inference. Bear these considerations in mind, we modified a 3D UNet model and use a sequence of 16 frame pairs as the input to the model.

Further complicating the task, event and APS cameras differ in dynamic ranges, which affects information compression in overexposed and underexposed areas. Additionally, both camera types have adjustable parameters such as exposure, ISO, and aperture, which can be dynamically tuned to adapt to varying environments. This renders the video-to-event voxel prediction a time-varying task, making a straightforward one-to-one mapping between APS video frames and event voxels especially challenging.

To address this complex task, a key contribution of our work is the development of a hybrid loss function composed of five distinct losses, which we briefly describe below.

Denote the input to the model as $I \in \mathcal{R}^{(B,L,2,H,W)}$, where the five dimensions represent the batch size, sequence length, and spatial resolution. Then the output event voxels satisfy $V \in \mathcal{R}^{(B,L,2 \times C,H,W)}$, where C represents the timebin number between two frames, and the third dimension has a shape of $2 \times C$ since events of different polarities are also separated. All losses take ground truth voxels and predicted voxels as input.

The first loss to introduce is the **Spatial-Temporal-Pyramid Loss (STP Loss, $\mathcal{L}_{STP}$)**. STP loss takes the entire concatenated voxel with a shape of $(B, L \times C, H, W)$ and applies a series of 3D Average Poolings with varying kernel sizes and strides. This produces more compact representations of both ground truth and predicted event voxels. The STP Loss encourages the model to extract multi-scale information from adjacent voxels, enhancing its robustness against noise by applying coarse supra-voxel matching. Formally, the STP Loss is defined as:

$$\mathcal{L}_{STP} = \sum_{k \in \mathcal{K}, s \in \mathcal{S}} w_{k,s} \cdot \left\| \mathcal{P}_{k,s}^{3D}(V_{GT}) - \mathcal{P}_{k,s}^{3D}(V_{pred}) \right\|_2^2 \quad (2.1)$$

Where $\mathcal{P}_{k,s}^{3D}(V)$ denotes 3D average pooling operation applied to voxel v with a kernel size k and stride s , $\mathcal{K}$ represents the set of all kernel sizes used in the pooling operations, $\mathcal{S}$ represents the set of all strides used in the pooling operations, and $w_{k,s}$ denotes the weights for each combination of kernel size k and stride s .

Temporal-Pyramid Loss (TP Loss, $\mathcal{L}_{TP}$) is designed to prioritize neighboring events, which are crucial for voxel-level event reconstruction. We employ 1D average pooling along the time axis using varying kernel sizes and strides on both ground truth and predicted event voxels, followed by an L2 loss calculation. Formally, the TP Loss is defined similarly to STP Loss:

$$\mathcal{L}_{STP} = \sum_{k,s \in \mathcal{K}, \mathcal{S}} w_{k,s} \cdot \left\| \mathcal{P}_{k,s}^T(V_{GT}) - \mathcal{P}_{k,s}^T(V_{pred}) \right\|_2^2 \quad (2.2)$$

Where $\mathcal{P}_{k,s}^T$ denotes 1D average pooling along the time axis.

Similarly, **Event Frame Loss (EF Loss, $\mathcal{L}_{EF}$)** compresses the time axis by summing timebins between adjacent frames or across the entire frame sequence along the time. This addresses the issue of sparsity in voxels and ensures better and aligned information flow between generated event frames and the input frame sequence. Event frames with and without polarity

consideration are separately counted in the loss calculation.

$$\begin{aligned}\mathcal{L}_{EF} = & \left\| \mathcal{S}_C(V_{GT}) - \mathcal{S}_C(V_{pred}) \right\|_2^2 + \\ & \left\| \mathcal{S}_{LC}(V_{GT}) - \mathcal{S}_{LC}(V_{pred}) \right\|_2^2\end{aligned}\quad (2.3)$$

Where $\mathcal{S}_C(\cdot)$ and $\mathcal{S}_{LC}(\cdot)$ denotes the compression operation that sums over timebins C between adjacent frames and the entire frame sequence LC respectively.

Adversarial Loss (ADV Loss, $\mathcal{L}_{ADV}$) aims to enhance the realness of our generated event voxels. Utilizing both ground truth and predicted voxels as real and fake samples respectively, the discriminator is trained for optimal distinction. To prevent $\mathcal{L}_{ADV}$ from becoming unbounded, the generated event voxels strive for high similarity with real voxels to effectively deceive the discriminator.

As previously noted, the relationship between APS frames' pixel brightness and the event number between frame pairs is not static, necessitating a dynamic, semantics-based modeling of intrinsic camera parameters. This complexity arises because APS captures brightness as $\phi(I)$, while DVS records $\log(\phi(I))$, where I is the scene's absolute brightness and $\phi(I)$ represents the effect of camera parameters. Given that multiple intrinsic parameters affect ϕ , a fixed linear mapping is untenable. To address this, we introduce **Brightness-Compensation Loss (BC Loss, $\mathcal{L}_{BC}$)**, which compute the average brightness I_a of voxels exceeding a threshold β , and align this I_a with that of the ground truth voxels. we define the average brightness I_a as:

$$I_a(V) = \frac{\sum_{v \in V, v > \beta} v}{|\{v \in V : v > \beta\}|} \quad (2.4)$$

Where β serves as a threshold to consider voxels that exceed a certain brightness. Given this, the BC Loss between ground truth voxels V_{GT} and predicted voxels V_{pred} is:

$$\mathcal{L}_{BC} = \left\| I_a(V_{GT}) - I_a(V_{pred}) \right\|_2^2 \quad (2.5)$$

At last, all these losses are combined together with a separate weight factor α (which is learned by grid search). The complete loss formula is:

$$\begin{aligned} \mathcal{L} = & \alpha_{STP} \mathcal{L}_{STP} + \alpha_{TP} \mathcal{L}_{TP} + \\ & \alpha_{EF} \mathcal{L}_{EF} + \alpha_{ADV} \mathcal{L}_{ADV} + \alpha_{BC} \mathcal{L}_{BC} \end{aligned} \quad (2.6)$$

Once trained, the motion-aware event voxel prediction generates 10 voxels per pixel between consecutive grayscale video frames at 30 fps. This effectively maps each frame with dimensions $H \times W$ to an event voxel grid of dimensions $(2 \times 10) \times H \times W$. Upon evaluation with the entire MVSEC dataset [179], we found this ten-fold upscaling to be adequate. Specifically, only approximately 2.30% of the voxels are non-zero, and among these, a mere 6.66% exceed one. This validates that partitioning the time range between two frames into 10 timebins is sufficient for capturing the event stream.

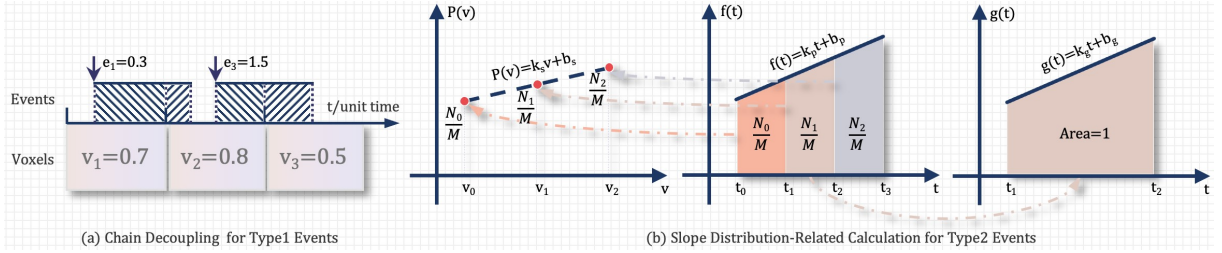


Figure 2.3. Visualization for two stages in LDATI.

2.2.2 Stage2: Voxels to Continuous Events Sampling

This second task recovers precise event timestamps from Stage1’s event voxels. Previous research used basic sampling methods, such as random or uniform sampling. We introduce **Local Dynamics-Aware Timestamp Inference (LDATI)**, an advanced technique considering each voxel’s event firing trends. It achieves a significantly lower error metric, which is only **3.5%** of the best traditional method.

Event voxels aim to discretize temporally contiguous events into a dense tensor, suitable for deep learning inference. Rather than merely counting the event numbers in each voxel (with

the temporal resolution of δ), the generation of event voxels also preserves the relative temporal information of events within the timebin. Each event influences the voxel series for a short and finite duration which can be characterized by a continuous-time unit step signal (with an on-time duration same as δ). The value of each voxel is determined by integrating all the step signals for all events within a voxel's designated time range. The sum of all voxels at the same pixel location equals the total number of events occurring within that time frame. This allows the voxel to summarize the total number of events and their relative times with a single number.

Considering the inverse process of event voxel generation, let v be the value of a voxel and v' be its value after removing the influences of events from the preceding voxel. If only one event is fired in the current voxel, its relative occurrence time within the voxel can be determined by $e = \lceil v' \rceil - v'$. This event will then exert an e -unit influence on subsequent voxels. Starting from the first voxel, where $v = v'$, we can iteratively deduce the event count and their relative positions in each voxel, a process we term as Chain Decoupling (Fig. 2.3.a). This computation is deterministic, as it merely reconstructs the inherent temporal information in the voxels.

Voxels are Type1 if $v' \leq 1$ and Type2 if $v' > 1$. As per Section 2.2.1, event voxels are sparse. Type1 voxels use Chain Decoupling; Type2 requires alternatives. Motion and changes happen in a continuous manner and don't change abruptly under natural condition. Thus, if the preceding voxel has a greater value than the succeeding one, events in the intervening voxel are more likely to be biased towards the former, and vice versa.

To accurately model this phenomenon, we assume that each voxel and its neighboring voxels conform to a slope distribution described by the Probability Density Function (PDF) $f(t) = k_p t + b_p$. Given that the event timestamp distribution within a voxel is primarily influenced by its temporal neighbors, a simpler slope distribution suffices for both accuracy and computational efficiency. This PDF allows us to estimate event timestamps while accounting for local dynamics, outperforming random sampling approaches.

Let the voxel value in three adjacent voxels be denoted as N_0 , N_1 , and N_2 , and their sum as M . Our objective is to derive the expression for their PDF $f(t)$ using these known variables. If

$g(t)$ represents the PDF of the event timestamp distribution conditional on t being in the central voxel v_1 , we have:

$$g(t) = f_{T|V}(t|v) = \frac{f_{T,V}(t, v)}{P(V = v)} = \frac{f(t)}{P(V = v)} \quad (2.7)$$

In this formulation, T and V denote the exact event timestamp and its corresponding timebin, respectively. The relationship among $f(t)$, $g(t)$, and $P(v)$ is visually explained in Fig. 2.3.b. It can be readily shown that $P(v)$ also follows a linear formula with a slope $k_s = \delta k_p = \delta k_g$. Given $k_s = \frac{N_2 - N_0}{2\delta M}$, the expression for $g(t)$ can be derived accordingly.

$$g(t) = k_p t + 1/\delta - \delta k_p / 2 \quad (2.8)$$

where $k_p = \frac{N_2 - N_0}{2\delta^2 M}$. While $g(t)$ -based sampling could be applied individually on each voxel, this would be computationally expensive due to varying distribution formulas and event numbers. However, we can optimize this process through distribution transformation. By initially sampling from a uniform distribution with PDF $\gamma(u) = 1/\delta$, and then converting it to the desired slope distribution via matrix operations, we significantly accelerate the sampling process. This conversion can be achieved using the inverse cumulative distribution function (CDF) method as follows:

$$t = (-b_p + \sqrt{b_p^2 + 2k_p u})/k_p \quad (2.9)$$

2.3 Experiments and Results

2.3.1 Stage1: Motion-Aware Event Voxel Prediction

Existing video to event simulation methodologies primarily rely on qualitative assessments and consequently lack in standardized objective quantitative evaluation metrics. To address these challenges, we introduce the following novel metrics tailored for this specific task:



Figure 2.4. Event frame comparison between V2CE and baseline methods. Event frames are clipped to the maximum value of their corresponding ground truth event frames and then normalized.

Table 2.1. The quantitative comparison between our method and all baseline methods on the testing set.

Method	TPF1 $\uparrow$	TF1	RF1	TPAcc $\uparrow$	TAcc	RAcc	PMSE-2 $\downarrow$	PMSE-4
ESIM [47]	0.3293	0.2670	0.1301	0.6240	0.7200	0.8801	3.03512	1.54268
EventGAN[180]	0.2398	0.1544	0.0520	0.1773	0.1773	0.1530	0.05779	0.03030
V2E[69]	0.3656	0.3197	0.1604	0.7214	0.8207	0.9269	0.04929	0.02618
Ours	0.5323	0.5058	0.3014	0.8891	0.9266	0.9709	0.00267	0.00074

1) **PMSE (Pooling MSE)** aims to compare the predicted voxel cube with the ground truth by employing a 3D average pooling (with kernel and stride of 2 or 4) and then estimating the Mean Square Error (MSE). The 3D average pooling extracts a slightly higher-level summary of the sparse voxel cube. 2) **TAcc** collapses the *Time* axis of the 3D voxel cube to convert to a 2D frame for each polarity, apply a thresholding (with a value of 0.001) to convert it to two binary 2D frames for each polarity, and lastly estimate the accuracy by performing the binary matching. 3) **TPAcc** is similar to TAcc, except it further collapses the event *Polarity* by accumulating the voxel cubes of two polarities. 4) If no temporal or polarity-wise accumulation is performed, the metric is termed **RAcc** (Raw Accuracy). 5) **TPF1, TF1, RF1**: we follow the same procedure as TAcc, TPAcc, or RAcc except F1 scores are calculated instead. Given that event voxels are generally sparse, the F1 score is more representative. TAcc, TPAcc, TF1, and TPF1 aim to evaluate high-frequency spatial patterns, while RAcc and RF1 provide a straightforward way to check the absolute voxel occupancy condition. TPF1 and TF1 are selected as the main metrics.

Comparison with Existing Works

We assess our video-to-events model against baseline methods using metrics that encompass both temporal and spatial aspects, as previously outlined. Consistent testing, training, and validation sets are employed for all methods. For EventGAN [180], we set its timebin number to 10 as well during retraining to align with our framework, while maintaining other settings. ESIM [47] and V2E [69], as non-model-based approaches, were executed with default configurations. As shown in Table 2.1, our model excels across all metrics and achieve SOTA.

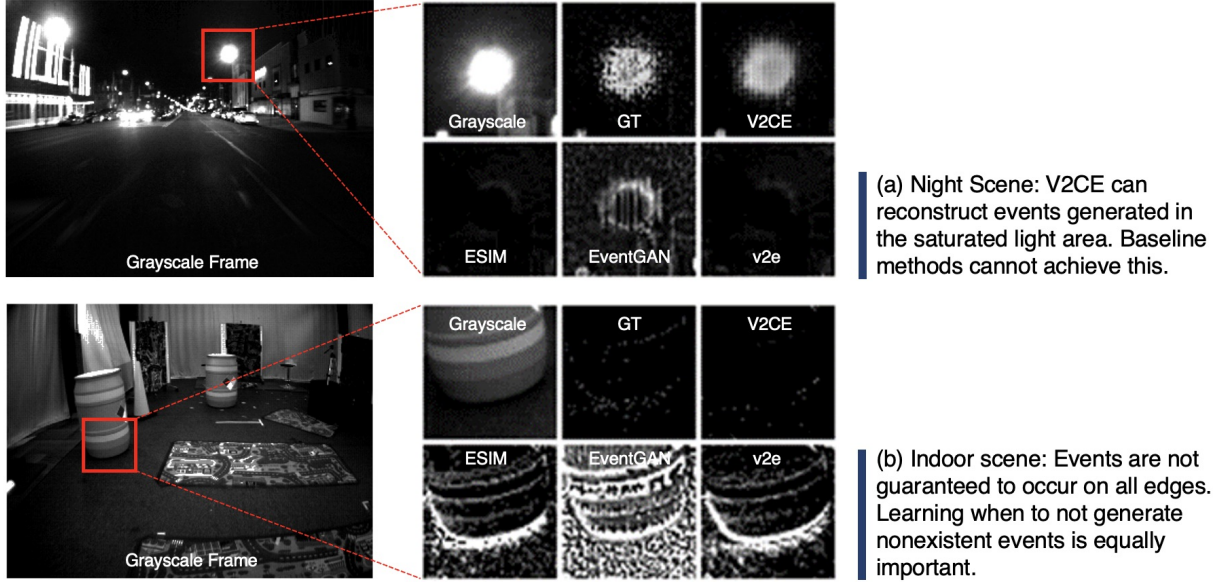


Figure 2.5. Zoom-in comparison on event frame details.

Qualitative comparison can be found in Fig. 2.4, 2.5, and 2.6. Fig. 2.4 shows a 2D event frame-level comparison between all the video-to-event voxel methods, and all event frames are clipped to the maximum value of the ground truth event frame and then normalized. Thanks to our specially designed losses, our predicted event frames are loyal to the actual brightness level, which means the events' number generated is close to the actual events' number. However, all the baseline methods tend to generate much more events in general, and the details don't match well. Due to the unstable and noise-prone nature of the underexposure part of the grayscale images, baseline methods generally behave poorly in night scenes. For the overexposed region of a scene, since the pixel value has saturated and the detailed information in these regions is lost. Without a good semantic understanding, this lost information can never be recovered. Moreover, all baseline methods behave quite aggressively when it comes to any edges, but different from a simple edge detector, a video-to-event converter not only needs to learn when to generate events but also when not to. A zoom-in comparison can be found in Fig. 2.5.

To better visualize the performance of different methods on the time axis, we also generated a 3D plot in the xyt space, as shown in 2.6. If the value of a voxel is greater than 0.001,

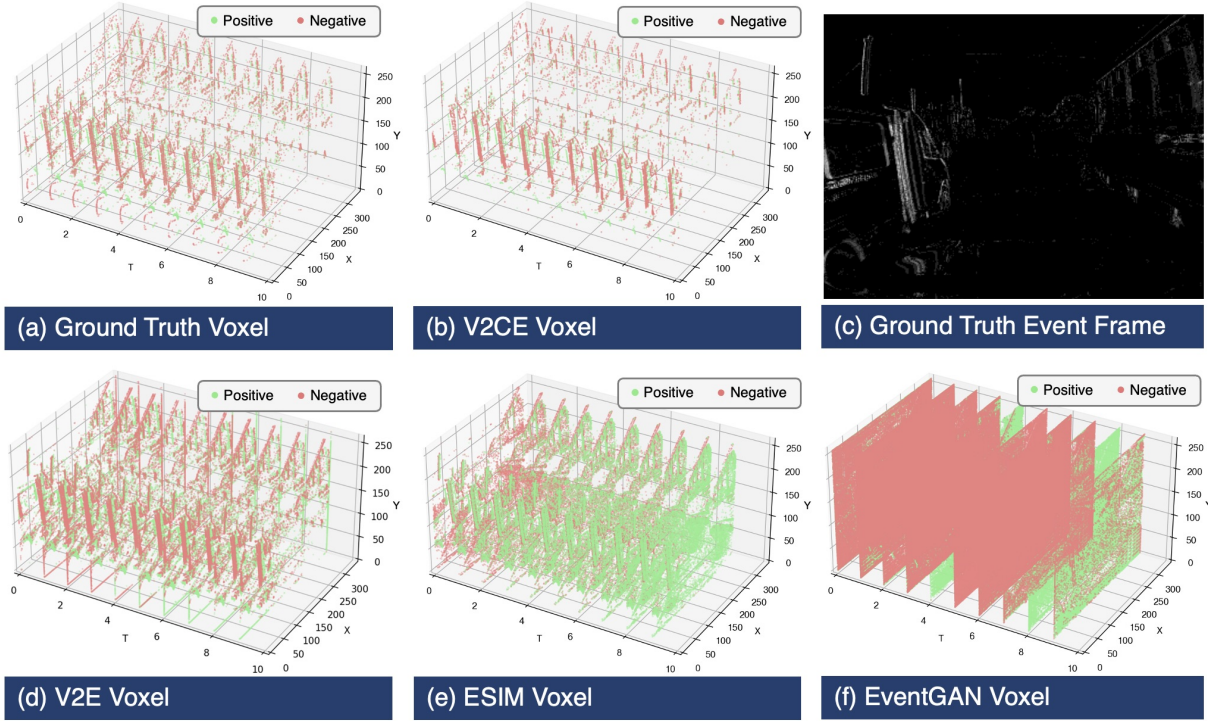


Figure 2.6. Voxel Cube comparison. (Notice: Voxels, not raw event stream.)

one node will be placed in the corresponding location. It’s clear that our prediction also matches the ground truth voxels significantly better both spatially and temporally.

Ablation Study

As indicated in Table 2.2, the removal of any loss functions leads to a decline in our primary metrics, TPF1 and TF1. Eliminating $\mathcal{L}_{\mathcal{I}\mathcal{D}}$ or $\mathcal{L}_{\mathcal{B}\mathcal{C}}$, which focuses on high-level matching and compensation, slightly improves the voxel-wise metric RF1 but negatively impacts

Table 2.2. The ablation study results for all our proposed losses.

Ablation Item	TPF1	TF1	RF1	PMSE-2	PMSE-4
STP Loss	0.5253	0.4999	0.3041	0.00279	0.00078
TP Loss	0.5170	0.4794	0.2528	0.00314	0.00084
EF Loss	0.4638	0.4461	0.2938	0.00248	0.00075
ADV Loss	0.3583	0.2817	0.1662	0.00225	0.00068
BC Loss	0.5253	0.4983	0.3070	0.00277	0.00076
w/o Ablation	0.5323	0.5058	0.3014	0.00267	0.00074

all other metrics. Deactivating the $\mathcal{L}_{\mathcal{AV}}$ results in a significant drop in all F1 metrics, despite a concurrent decrease in PMSE losses. This suggests that while the absence of $\mathcal{L}_{\mathcal{AV}}$ may improve voxel value matching, it significantly deviates the generated voxels from real voxels’ distribution. Hence, each proposed loss function positively influences the final performance.

Training Details

We trained our neural network using the Adam optimizer, setting the learning rate to 0.001 and running the training for 100 epochs on our dataset. This project is built on top of PyTorch Lighting. We employed the Multi Vehicle Stereo Event Camera (MVSEC) dataset [179], which is one of the largest and most commonly used datasets for DVS-based research. The dataset was randomly partitioned into training, validation, and testing subsets, following an 80%/10%/10% distribution. Each data entry in the dataset comprises 17 sequential grayscale frames, along with the associated 16 event packets in between.

The size of our Stage1 model is 52.9MB, and it requires 779.17 GFlops to perform inference on a single image pair sequence. Each sequence has a length of 16, which corresponds to 0.53 seconds of video length when the video’s frame rate is 30Hz. When evaluated on an A10 graphics card, the model’s average inference time for a 16-pairs sequence was 312.83 milliseconds, and the corresponding sampling time in Stage2 was 106.97ms. This allows us to generate events from approximately 39 frames per second, meeting the criteria for real-time performance.

2.3.2 Stage2: Voxel to Continuous Event Sampling

To evaluate event voxel-to-event stream generation, we introduce three metrics. **Mean Event Timestamp Error (METE)** quantifies the average temporal discrepancy, in microseconds, between each ground truth and the nearest predicted event at the same pixel. Ground truth events without a nearby counterpart within 3δ on the time axis are termed “overflow events,” and their error is capped at 3δ . **Number of Overflowed Events (NOE)** quantifies the number of such

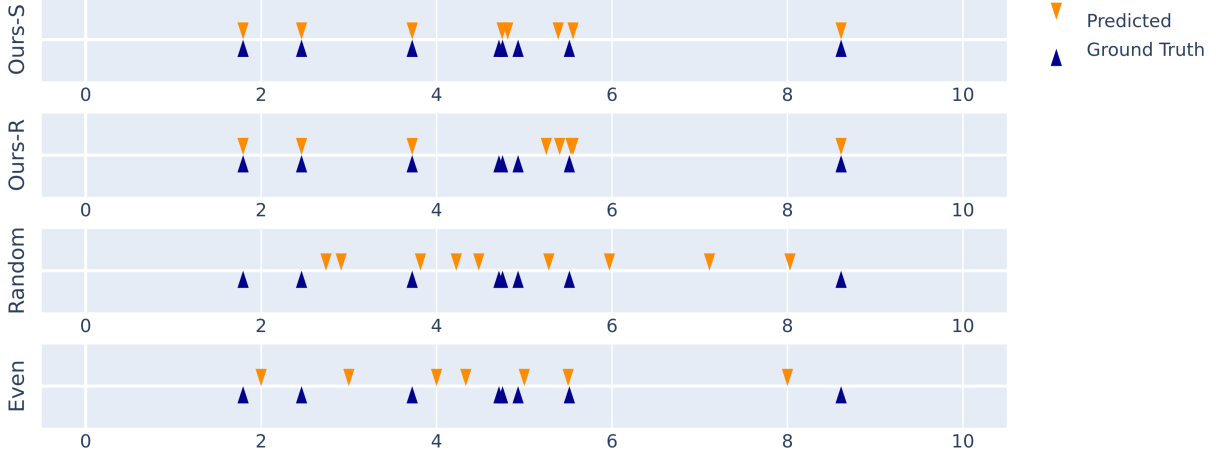


Figure 2.7. Qualitative comparison between all voxel-to-event sampling methods given a randomly generated event stream (blue upward wedges). In this figure, “Ours-S” and “Ours-R” represent chain decoupling with slope distribution and chain decoupling with random distribution for Type2 events.

extreme errors. To mitigate potential bias from oversampling, we introduce **Predicted/Ground Truth Event Number Ratio (PGER)**, aiming for a value as close to 1 as possible. By multiplying the PGER when it is greater than one, we can get the calibrated METE and NOE, which are called **C-METE** and **C-NOE**.

Table 2.3. Testing set event voxel-to-event stream results comparison.

Method	Type2 Events	C-METE	C-NOE	PGER
Even	Even Sampling	4003.528	2362	1.000
Random	Random Sampling	3657.325	2350	1.000
LDATI	Random Sampling	851.679	638	1.000
LDATI	Slope Distribution	142.884	0	1.000

For baselines, we consider two standard techniques: random and even samplings. Let v be the input voxel value. In both methods, the estimated event number $\tilde{v}$ in a voxel is computed as $\tilde{v} = \lfloor v \rfloor + \text{Bernoulli}(v - \lfloor v \rfloor)$. In random sampling, $\tilde{v}$ events are sampled randomly within the timebin. In even sampling, we find the maximum $M = \max(\tilde{v})$ across all voxels, divide the timebin into M equal sub-timebins, and sequentially place $\tilde{v}$ events starting from the leftmost sub-timebin. A qualitative comparison is shown in Fig. 2.7.

To rigorously evaluate our LDATI sampling method, we conducted an experiment on

ground truth event streams from our test set, converting them to event voxels as outlined in Section 2.2.2. We then applied various sampling techniques for event reconstruction and computed the predefined metrics. The results, detailed in Table 2.3, confirm a PGER metric of 1 for all methods, eliminating under- or over-sampling concerns. LDATI significantly outshines the baselines, with a C-METE metric just **3.5%** that of the even sample and a NOE of zero throughout the test set. Substituting our slope distribution sampling with random sampling for Type2 events led to a **6×** higher C-METE error, underscoring the efficacy of slope distribution sampling.

In our study, we implemented a two-stage evaluation and refined the metrics. Although Stage 1 introduces notable voxel-level errors, skewing the overall error relative to ground truth, our V2CE pipeline demonstrates its efficacy as shown in Table 2.4. Unlike prior methods that tend to oversample events substantially (up to **53.3×** more), leading to significant divergence from the true event distribution, V2CE closely matches the ground truth in event count while simultaneously achieving optimal accuracy metrics.

Table 2.4. Testing set video-to-event stream results comparison.

Stage1	Stage2	C-METE	C-NOE	PGER
EventGAN[180]	Random	395057.269	372383	45.280
ESIM[47]	Even	876386.946	1346561	54.299
v2e[69]	Even	63738.935	82238	5.145
V2CE	Even	11131.586	12568	0.720
V2CE	Random	11079.564	12487	0.720
V2CE	LDATI	10039.262	10364	0.997

2.4 Conclusion

In this paper, we present V2CE, a novel pipeline that converts video to high-fidelity event streams for tasks requiring precise events. We introduce quantifiable metrics that elevate this field from qualitative to rigorous scientific analysis. V2CE excels overwhelmingly across all metrics.

Acknowledgments

Chapter 2, in full, is based on the paper from Zhongyang Zhang, Shuyang Cui, Kaidong Chai, Haowen Yu, Subhasis Dasgupta, Upal Mahbub, and Tauhidur Rahman, “V2CE: Video to Continuous Events Simulator” published in 2024 IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 2024, pp. 12455-12461. The dissertation author is the primary investigator and author of this paper.

Chapter 3

Labits: Layered Bidirectional Time Surfaces Representation for Event Camera-based Continuous Dense Trajectory Estimation

3.1 Introduction

As an emerging visual modality, event cameras offer unique and practical advantages. Compared to conventional frame-based cameras, they provide higher temporal resolution, greater dynamic range, higher efficiency, and lower latency ([45]). Furthermore, under stable lighting, event cameras are primarily sensitive to the edges of moving objects, naturally filtering out stationary objects while tracking moving ones. Their ultra-high temporal resolution also enables smoother and more continuous target tracking. In recent years, numerous papers leveraging this feature of event cameras have addressed topics such as feature tracking ([108]), optical flow generation ([153]), and video interpolation ([61]) based on events.

From an event camera’s perspective, each moving point generates a discrete trajectory in the xyt space, with each triggered event representing a sampled point on this trajectory, along with its timestamp. The instantaneous velocity at any point on the trajectory can be calculated from the relative positions and times of these events using straightforward mathematical formula $v = \Delta x / \Delta t$, aligning with intuitive understanding. While real-world factors like rotations, depth

movements, and multiple moving objects complicate pixel-wise speed, the trajectory information persists embedded within the event streams. Our target is to unearth these hidden treasures.

Currently, there are two main approaches to utilizing events. One is to directly construct events into a graph in the xyt space and input it into a GNN (Graph Neural Network) ([94]) or treat each event as a spike to be input into an SNN (Spiking Neural Network) ([85]). The other is to first convert events into a dense representation and then input them into an Artificial Neural Network (ANN). Although GNNs appear efficient, constructing the graph is computationally and memory-intensive with increasing events, and GNNs suffer from over-smoothing as network depth increases ([28]). Additionally, the graph conversion discards fine-grained spatial and temporal information, making position-sensitive scenarios problematic. These limitations restrict GNN applications in event-based vision, especially for low-level tasks. On the other hand, while SNNs offer high energy efficiency, they are more difficult to train and are less robust ([89]). Compared to deep learning, the SNN ecosystem is still in its early stages, and the maturity of neuromorphic hardware is insufficient to support large-scale, high-precision models in practical applications ([114]). These factors limit the widespread use of SNNs.

The mainstream approach in event-based vision remains converting the event stream into a dense representation before inputting it into an ANN for inference ([168], wan2024event). However, existing representations have limitations that prevent event cameras from fully leveraging their strengths for dense trajectory estimation. Common representations like event frames ([123]) and event counts ([106, 181]) entirely remove the temporal dimension by projecting all events onto a 2D plane. Event frames sum the polarities at each pixel, while event counts simply tally the events. Similarly, voxel grid ([182]) applies temporal quantization discards fine-grained temporal information, projecting events onto the nearest temporal grid using a bilinear sampling kernel. Although representations like time surfaces ([86]) and TORE volumes ([11]) preserve some temporal information, each pixel retains only the timestamps of the most recent events, ignoring earlier ones triggered by different objects. This leads to temporal occlusion and limits the ability to extract continuous, dense, and temporally consistent local motion information in

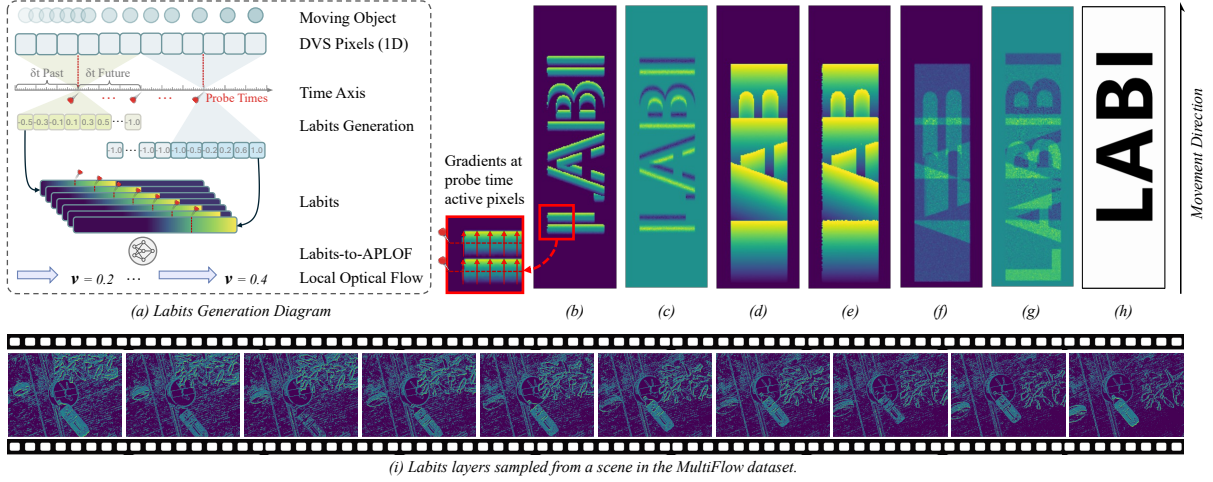


Figure 3.1. (a) Labits generation schematic: For a 1D event camera, at each pixel and probe time pt_i , the algorithm searches for the most recent past event within δt . If none is found, it searches for the next future event within δt . The Labits value is the normalized time difference between probe time pt_i and the found event’s timestamp, or -1 if no event is found. Labits can be converted to APLOF via a small model. (b)-(g): Single channel visualization of the following event representations (more details can be found in related works): (b) Labits (c) Voxel Grid (d) TORE Volume (e) Time Surface (f) Event Count (g) Event Frame (h) RGB frame of the moving target (i) Labits layers samples. Note that the first layer of TORE is exactly the same as time surface.

the spatio-temporal (xyt) domain.

Dense trajectory estimation requires pixel-level movement sensitivity, stable and sharp 2D visual features, and consistent information density throughout the tracking duration. The final requirement is to ensure the representation does not excessively focus on events near the end of the time period while neglecting earlier occurrences. To fully exploit the potential of events for dense trajectory estimation, a representation that preserves all these characteristics is essential. In this paper, we propose Labits: **LA**yered **BI**directional **T**ime **S**urfaces, as a simple and elegant solution to these challenges. The impact of an effective event representation is significant. By simply switching to Labits, it can yield a **13%** improvement on TEPE, while incorporating an additional APLOF extractor can provide an extra **30%** reduction in error for dense trajectory estimation.

Our main contributions can be summarized as follows: 1. We proposed Labits, a novel

Table 3.1. A comparison of event representation methods used in deep learning, where H and W denote the height and width of the representations, respectively. This table extends the one presented in ([12]) with newly included methods.

Event Representation	Dimensions	Description	Characteristics
Event Frame	H, W	Image of event polarities	Discards temporal & polarity information
Event Count	2, H, W	Image of event counts (EC)	Discards temporal information
Time Surface	2, H, W	Image of most recent timestamp	Discards all prior time stamps
Averaged Time Surfaces	2, H, W	Image of average timestamp for window	Discards temporal information
Inceptive Time Surfaces	3, H, W	Image of filtered timestamps & EC	Discards temporal information
Event Spike Tensor	2, B, H, W	4D grid of convolutions	Temporally quantizes information into B bins
TORRE Volumes	2, K, H, W	4D grid of last K timestamps	Discards timestamps prior to last K events
Voxel Grid	B, H, W	Voxel grid summing event polarities	Discards polarity information
Labits	B, H, W	Layered bidirectional time surfaces	Provides multi-layer local speed hints

synchronous event representation that is aware of event camera’s asynchronous characteristic and keeps rich local movement trend information. 2. We trained a corresponding active pixel local optical flow estimator based on Labits layers, which utilizes intermediate motion information. 3. We achieved the SOTA performance on event-based dense trajectory estimation task, and the TEPE of our method decreased **49%** compared to the previous SOTA.

3.2 Related Works

Event Camera: Event cameras contain a bio-inspired dynamic vision sensor, where each pixel unit works asynchronously and triggers an event instantly when it detects a log-intensity change over a predefined threshold. Each event $\mathbf{e} = (t, x, y, p)$ records the spatial coordinate (x, y) of the corresponding pixel position on the image sensor plane, the microsecond-level shooting timestamp t , and a binary polarity value p that indicates the direction of brightness change ([174]). Event cameras are widely used in various computer vision and robotics tasks. They excel in motion-centric tasks like optical flow estimation and object or human pose tracking, as demonstrated in numerous studies ([68, 159, 27, 173]). Their high temporal resolution and event-driven operation also enable innovative video processing techniques, including frame interpolation ([151, 142, 100]) and motion deblurring ([30, 166, 82]). These applications leverage the unique characteristics of event cameras for detailed, dynamic scene analysis without

the high data demands of traditional high-speed video recording.

Event Representations: In event-based vision, mainstream methods convert the event stream into a dense representation, then pass it to ANNs for various tasks ([13]). However, existing representations often fail to fully leverage the unique capabilities of event cameras.

Early representations attempt to convert event streams into intensity frames, highlighting moving edges and mimicking 2D features of traditional cameras. The event count representation ([106, 181]) sums the number of events per pixel within a time window, while the event frame ([123]) sums event polarities. Both discard temporal information, obscuring events over time. The voxel grid representation ([182]) quantizes time and maps events to temporal grids using bilinear sampling. While better than event count, time information retention is still limited, and temporal obscuring persists.

Time surface-style representations form another key branch. The original time surface, or Surface of Active Events (SAE, [15]), encodes only the most recent event’s timestamp per polarity at each pixel, disregarding prior events, no matter the scene’s complexity. This leads to poor 2D pattern capture and temporal occlusion, where newer events overwrite earlier ones. Variants like the averaged time surface ([139]) reduce noise and mitigate occlusion, but reintroduce temporal obscuring and ambiguity. TORE volume ([12]) preserves the most recent K events per pixel, creating multi-layer time surfaces. However, redundancy arises when the same object repeatedly triggers recent events at the same pixel, perpetuating temporal occlusion. This is partly due to the complex textures of real-world objects, which introduce numerous small edges that trigger events. All aforementioned representations, and others, are summarized in Table 3.1.

3.3 Labits: Layered Bidirectional Time Surface

Event cameras have distinct advantages over traditional frame-based cameras due to their asynchronous nature. This characteristic enables event cameras to provide highly precise timestamps at the microsecond level, making them ideal for capturing instantaneous motion

velocity. However, existing event representations fail to fully exploit these advantages. Most approaches convert the sparse and discrete event streams into frame-like structures that highlight 2D visual features such as moving edges and patterns, thus making the event modality compatible with conventional computer vision models. The transformation essentially forces event cameras to conform, fitting themselves into the framework dominated by frame-based cameras, rather than fully leveraging their own strengths and showcasing the unique advantages that conventional cameras cannot replicate.

Building event representations inevitably involves information loss. It’s almost a trilemma to faithfully preserve the original fine-grained timestamp information, compile meaningful 2D visual patterns, and maintain historical movement information at intermediate times simultaneously. However, all three aspects are essential for accurate, dense, continuous-time trajectory estimation. Fine-grained event-level timestamp information is the unique strength of the event modality in movement prediction. 2D visual patterns form the basis for correlation-based tracking mechanisms, while intermediate movement information ensures stable estimation accuracy throughout the entire trajectory.

Therefore, we designed Labits: Layered Bidirectional Time Surfaces to meet these demands. The logic behind this representation is straightforward: a time-surface-like structure is essential for retaining the microsecond-level timestamp features. The accumulation time range of events should be strictly controlled to avoid issues with visual feature blurriness, so splitting the target time range into a series of smaller time bins is essential. Stopping at this stage creates a layered time surface.

However, under this scheme, only the last events’ information is utilized within each time bin at each pixel, which is unpreferable. Moreover, when estimating the local speed at each time bin intersection, only past movement information is considered, so the prediction here is based on backward difference, where the error estimation is known to be $O(\delta t)$. Further reducing this error estimation is nothing difficult: simply consider future event if no event is observed in the past δt search range. This change in representation is simple yet powerful: First, pixels ahead of

the moving edges' direction often don't have triggered events in the near past, by incorporating future events, these originally empty pixels are assigned a meaningful value, thereby increasing the information density of the resulting representation. Second, the first and last event within each time bin at each pixel are both fully utilized. Especially when the time range is small, this sampling strategy is highly representative. Third, considering near past and future events simultaneously changes the basic representation unit from time bin to intermediate probe times. Local speed estimations at these probe times use central difference, reducing the estimation error to $O(\delta t^2)$ (detailed in the Supplementary Material).

Let $E = \{(t_n, x_n, y_n, p_n)\}_{n=1}^N$ represent the event stream, where t_n denotes the timestamp, (x_n, y_n) are the spatial coordinates, and p_n is the polarity of the n -th event. The variables τ_{start} and τ_{end} refer to the first and last event timestamps, respectively, and τ_{total} is the total time duration of the events. The time interval between two adjacent probe times is denoted as τ_{range} , and τ_i denotes the i -th probe timestamp. The output tensor $L \in \mathbb{R}^{B \times H \times W}$ is the Labits representation, where B is the number of probe time points, H and W is the sensor height and width. The Labits generation algorithm is elaborated in Algorithm 1.

As shown in Algorithm 1, the backward and forward temporal search range at each probe time point is t_{range} , which corresponds to the time interval of each time bin. Under this scheme, the timestamp of the earliest and latest events within each time bin are recorded by the Labits layers at the two adjacent probe points surrounding the bin, further minimizing information loss.

Moreover, the design of Labits ensures that it maintains stability and consistency. The values in the each layer of Labits always lie within a fixed range, making Labits inherently robust against outliers, such as hot pixel noise, which can otherwise distort the normalization scale in other representations.

With Labits, dense instantaneous optical flows at active pixels can be generated across evenly-spaced probe times using straightforward neural networks, due to the clear correlation between Labits layer values and their corresponding local speeds. The resulting layered instantaneous optical flows could serve as a basis for future research into tasks requiring high

Algorithm 1: Labits Representation Generation.

Input : $E = \{(t_n, x_n, y_n, p_n)\}_{n=1}^N$, where t_n is timestamp, (x_n, y_n) are spatial coordinates, and p_n is the polarity of the event. Events are ordered by t_n in ascending order.
Output : $L \in \mathbb{R}^{B \times H \times W}$, a tensor representing Labits. B is the number of intermediate probe time points, H and W are the sensor height and width.

- 1 Set $\tau_{start} = t_1$ and $\tau_{end} = t_N$ (first and last event timestamps);
- 2 Compute total duration: $\tau_{total} = \tau_{end} - \tau_{start}$;
- 3 Divide τ_{total} into $B + 1$ equal intervals:

$$\tau_{range} \leftarrow \tau_{total} / (B + 1), \quad \tau_i \leftarrow \tau_{start} + i \cdot \tau_{range}, \quad i \in \{1, \dots, B\}$$

Initialize $L \in \mathbb{R}^{B \times H \times W}$ with -1 ;

```

4 for  $i = 1$  to  $B$  do
5    $E_{prev} \leftarrow \{e_n \mid \tau_i - \tau_{range} \leq t_n \leq \tau_i\}$ ;  $E_{future} \leftarrow \{e_n \mid \tau_i < t_n \leq \tau_i + \tau_{range}\}$ ;
6   Define  $T_{prev} \in \mathbb{R}^{H \times W}$  and  $T_{future} \in \mathbb{R}^{H \times W}$ , initialized with  $-\infty$  and  $\infty$ , respectively;
7   for each  $e_n \in E_{prev} \cup E_{future}$  do
8     Compute normalized time:  $t_{norm} \leftarrow (t_n - \tau_i) / \tau_{range}$ ;
9     if  $e_n \in E_{prev}$  then
10      Update  $T_{prev}(y_n, x_n) \leftarrow t_{norm}$ ;
11    else
12      Update  $T_{future}(y_n, x_n) \leftarrow t_{norm}$ ;
13  for each  $(x, y) \in \{(x, y) \mid x \in [0, W - 1], y \in [0, H - 1]\}$  do
14    Update  $L[i, y, x]$ :
      
$$L[i, y, x] \leftarrow \begin{cases} T_{prev}(y, x), & \text{if } T_{prev}(y, x) \neq -\infty \\ T_{future}(y, x), & \text{if } T_{prev}(y, x) = -\infty \text{ and } T_{future}(y, x) \neq \infty \\ -1, & \text{otherwise} \end{cases}$$

15 return  $L$ ;
```

temporal resolution and advanced motion understanding, such as event-based object or human pose tracking.

Furthermore, the flexibility of Labits enables its use with varying numbers of probe time points and time bin sizes, without the need to retrain the model used to predict instantaneous optical flows. This adaptability stems from the fact that corresponding APLOF are generated separately for each Labits layer, and variations in time bin duration just introduce a scaling factor

to the perceived speed. Subsequent deep learning models can readily adjust to any scaling effects introduced by variations in the generated local optical flows. This combination of adaptability and precise flow prediction enhances Labits’ utility as a versatile and powerful tool for event-based computer vision tasks.

In conclusion, Labits overcomes the limitations of existing event representations by providing a highly accurate, temporally precise, and robust way to capture motion information from event streams. Its ability to handle both near-past and near-future events, coupled with its stability against time normalization, allows it to deliver superior performance in dense trajectory estimation. By fully harnessing the hardware advantages of event cameras, Labits sets a new benchmark for event-based continuous dense trajectory estimation.

3.4 Method

3.4.1 Labits-to-APLOF Net

Event cameras are highly sensitive to motion, particularly instantaneous or local motion. In this context, “local” refers to both spatial and temporal locality, aspects not captured by conventional optical flow techniques. For instance, in the MultiFlow dataset, the optical flow ground truth at each intermediate probe time is relative to the initial reference time point, representing cumulative motion rather than instantaneous speed. In contrast, events triggered by the same moving object within a short time frame are closely tied to the object’s instantaneous speed, based on the variation in their timestamps, rather than cumulative displacement. Therefore, to fully leverage event cameras for tracking, local speed must be taken into account.

Labits retains substantial local motion information at active pixels both before and after the probe time points. Compared to voxel grids, the information encoded in Labits enables the main model to predict trajectories with greater accuracy. However, due to the sparsity of events, noise, and temporal occlusion, it can be challenging for a downstream task model to effectively decode this implicit information without any dedicated guidance.

To fully leverage Labits, we propose a general-purpose model component named the Labits-to-APLOF Net, where **APLOF** stands for “**A**ctive **P**ixel **L**ocal **O**ptical **F**low.” This model component is specifically designed to utilize a single Labits layer as input for three main reasons: First, each Labits layer, generated at an intermediate probe time point, already contains sufficient information to infer the corresponding APLOF. Second, as we mainly focus on spatiotemporal locality, using a single layer and a shallow network naturally preserves that while keeping the model efficient and enhancing generality. Finally, since the number of time bins in Labits can be freely adjusted by the user, maintaining a single-channel input increases flexibility. This design makes the Labits-to-APLOF Net a versatile module compatible with Labits.

The Labits-to-APLOF Net is implemented based on U-Net ([129]). To improve both stability and efficiency, we employ instance normalization within each block, rather than batch normalization. This adaptation arises from shifting the channel dimension of Labits to the batch dimension during inference, enabling the network to more effectively handle Labits while keeping the single input channel. Since Labits contains rich information primarily around active pixels, local speed estimation is most reliable in these areas. Thus, we apply an active pixel mask (APM) to the generated APLOF features, considering only Labits pixels with absolute values below a certain threshold β as active pixels. To ensure alignment between the low-resolution (LR) APLOF features at the bottleneck and the pixel coordinates, we incorporate an auxiliary output head during training. This head is trained to predict an LR version of the local speed flow based on the bottleneck features.

Let $\hat{\mathbf{A}}_h$ and $\hat{\mathbf{A}}_l$ represents the estimated high-resolution (HR) and LR APLOF, while E and D_h denote the encoder and decoder of the U-Net, respectively. D_l means the auxiliary LR decoder, and $\mathbf{M}_h, \mathbf{M}_l$ stand for HR/LR active pixel masks. The following equation formalizes the

process:

$$\mathbf{M}_h = \mathbf{1}_{|\mathbf{L}| < \beta} \quad (3.1)$$

$$\mathbf{M}_l = \mathbf{1}_{\text{AvgPool}_{8 \times 8}(\mathbf{M}_h) < \gamma} \quad (3.2)$$

$$\hat{\mathbf{A}}_h = D_h(E(\mathbf{L})) \cdot \mathbf{M}_h \quad (3.3)$$

$$\hat{\mathbf{A}}_l = D_l(E(\mathbf{L})) \cdot \mathbf{M}_l \quad (3.4)$$

In these equations, the term $\mathbf{1}_{|\mathbf{L}| < \beta}$ is an active pixel mask that is generated based on the condition $|\mathbf{L}| < \beta$, where the mask M_h takes the value 1 when the condition is satisfied and 0 otherwise. In this way, pixels where there are events happened most recently stand out. Similarly, the low-resolution active pixel mask is generated based on a down-sampled version of high-resolution active pixel mask, with a threshold of γ . After fine-tuning, the thresholds β and γ are set to 0.3 and 0.125.

During training, the ground truth APLOF at τ is generated using the ground truth optical flow at τ_- , τ , and τ_+ , where τ_- , τ_+ represents the timestamp of $\tau \pm 10ms$. Denoting a pixel's coordinate at τ_{start} as $\mathbf{x}_{\text{start}}$, and optical flow between τ_{start} and τ as O_τ , the ground truth HR APLOF value is:

$$\mathbf{x}_\tau = \mathbf{x}_{\text{start}} + O_\tau(\mathbf{x}) \quad (3.5)$$

$$\mathbf{A}_\tau(\mathbf{x}_\tau) = (O_{\tau_+}(\mathbf{x}_{\text{start}}) - O_{\tau_-}(\mathbf{x}_{\text{start}})) \cdot \mathbf{M}_h(\mathbf{x}) \quad (3.6)$$

The training loss combines the LR and HR APLOF losses as follows:

$$\mathcal{L}_A = \|\hat{\mathbf{A}}_h - \mathbf{A}_h\|_1 + \|\hat{\mathbf{A}}_l - \mathbf{A}_l\|_1 \quad (3.7)$$

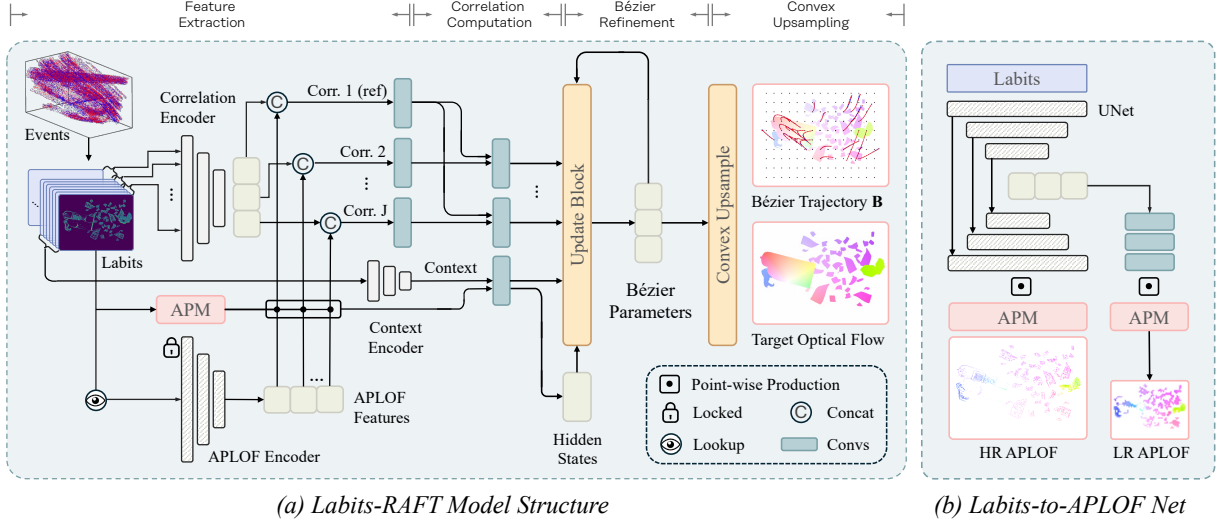


Figure 3.2. (a) Labits-RAFT architecture: Labits are used to generate correlation blocks, content features, APLOF features for intermediate movement integration, and guide Active Pixel Mask (APM) generation. APM layers are point-wise multiplied to their corresponding APLOF feature layers. Features are used to calculate correlation matrix, eventually generate and refine a Bézier curve $\mathbf{B}$ for each pixel at τ_{start} via a ConvGRU. For brevity, we only show the pure event-based pipeline. (b) Labits-to-APLOF Net: HR and LR APLOF are generated based on a customized U-Net and APM.

3.4.2 Pipeline

Accurately estimating pixel trajectories over time goes beyond simply determining their final displacements. Our proposed solution aims to predict each pixel’s shifting at various intermediate probe times within the defined time range. The RAFT-based structure ([146]) excels at extracting correlations between pixels across different spatiotemporal locations. Predicting the trajectory as a Bézier curve, rather than as a series of discrete displacements, allows for a smoother and more continuous representation. Building on top of ([51]), our pipeline consists of **four main components**: feature extraction, correlation computation, Bézier parameter refinement, and upsampling. The task is mathematically defined by the following equations:

$$\mathbf{B} : \mathcal{T} \times \mathbb{N}_0 \times \mathbb{N}_0 \rightarrow \mathbb{R}^2 \quad (3.8)$$

$$(\tau, x, y) \mapsto B(\tau, x, y) \quad (3.9)$$

$\mathcal{T} = \mathbb{R} \cap [0, 1]$ describes the domain of the normalized time $\tau(t) = (t - t_r)/(t_t - t_r)$, with $\tau = 0$ and $\tau = 1$ corresponding to the reference time t_r and target time t_t , respectively, indicating the start and end of the pixel trajectory. This formulation allows the trained model to predict pixel displacement for any initial pixel at subsequent time points.

In part one, the generated event representation and RGB frames (if available) are encoded into feature maps. The time period corresponding to the events is evenly divided into B segments, each lasting milliseconds. Of these, the first $M - 1$ segments provide additional reference information, while the remaining $B - M + 1$ segments constitute the target time period, referred to as the context Labits block. If available, the RGB frame at the reference time is concatenated to the context Labits block to enrich the feature set. Using a sliding window approach, the total B Labits layers are grouped into J correlation Labits blocks, each containing M layers. Each block is also referred to as a “view,” as it represents a distinct timestamp.

In part two, correlation volumes are computed between the first view and subsequent views to facilitate the spatial-temporal correspondence search for initial pixels. Additionally, if RGB frames are provided, a correlation volume between the reference and target frames is also calculated. These volumes are then utilized to iteratively refine the Bézier parameters.

In part three, the goal is to find a set of Bézier control points, $\mathcal{P}$, for each initial pixel at the reference time, t_r . Initially, all control points are set to zero, positioning each pixel’s default trajectory as a straight line along the time axis. This part primarily involves a ConvGRU, which processes the context and correlation information, along with the Bézier parameters, and outputs the residuals of the Bézier parameters. The initial hidden states of the ConvGRU are generated from the context volumes and APLOF features. During each subsequent iteration, the Bézier parameters are progressively refined to achieve optimal estimation. The Bézier curve is defined as follows:

$$\mathbf{B}(\tau, x, y) = \sum_{i=0}^n \binom{n}{i} (1 - \tau)^{n-i} \tau^i \mathbf{P}_i(x, y) \quad (3.10)$$

Here, $\mathcal{P}$ represents the set of Bézier control point parameters, with $\mathcal{P} = \mathbf{P}_1, \dots, \mathbf{P}_n$ when the

Bézier curve has n degrees. After the iterative refinement of the low-resolution Bézier parameters, convex upsampling is applied to upscale the Bézier parameters to the original resolution (**part four**).

The model is supervised with N_k ground truth optical flows along the trajectory. Let $\mathbf{B}_i$ denote the Bézier curve at iteration i , $\tau_k \in [0, 1]$ represent the evaluation timestamps, and $\gamma = 0.8$. The loss function is defined as follows, with more details provided in ([51]).

$$\mathcal{L} = \frac{1}{N_k} \sum_{i=1}^{N_i} \gamma^{N_i-i} \sum_{k=1}^{N_k} \|\mathbf{f}_{gt}(T_k) - \mathbf{B}_i(T_k)\|_1 \quad (3.11)$$

APLOF features introduced earlier are utilized in parts one and three. In part one, APLOF features are generated using the selected Labits layers that correspond to the intermediate trajectory ground truth time points. These features are then merged with the associated correlation block features based on the time range of each view. In part three, the initial hidden states of the ConvGRU, generated from the context block, are enhanced by merging with APLOF features. This integration provides additional local motion information, significantly improving trajectory estimation accuracy. The results section demonstrates the exceptional effectiveness of the APLOF features.

3.5 Results

Datasets and Evaluation Metrics. Our model is trained and evaluated on the MultiFlow dataset ([51]), which comprises 10,100 training and 2,000 test sequences, each including paired RGB images, events, and optical flows relative to a reference time. With optical flow ground truths recorded every 10 ms, we can generate intermediate local optical flows for training the Labits-to-APLOF net, and ensure accurate supervision of the intermediate Bézier parameters. MultiFlow is currently the only available event dataset that supports dense trajectory prediction and active pixel local optical flow estimation simultaneously. The ground truth for pixel trajectories is provided relative to a reference time of 0.4 seconds (t_{ref}), with trajectories

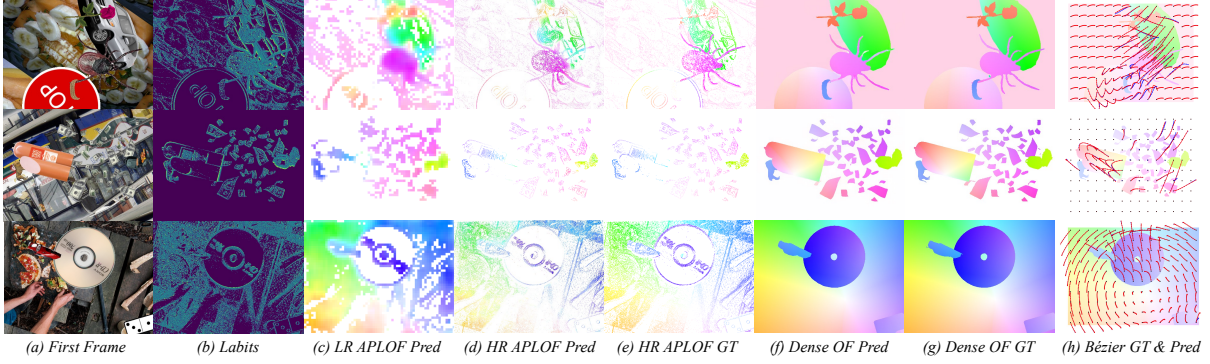


Figure 3.3. Visualization of detailed inputs and outcomes from our model. It predicts instantaneous APLOF at intermediate reference times, end-point optical flow (the end-point optical flow is computed as the displacement of each pixel along its entire trajectory), and pixel-level Bézier trajectories, all closely aligning with the corresponding ground truth data. *OF: Optical Flow.

calculated at 10-millisecond intervals between t_{ref} (0.4 seconds) and t_{tar} (0.9 seconds). Events occurring outside this range provide additional context. To evaluate the trajectory-level estimation accuracy, beyond the standard EPE and AE metrics, we also adopt Trajectory End-Point Error (TEPE) and Trajectory Angular Error (TAE).

Labits-to-APLOF Net Results. The Labits-to-APLOF Net is supervised by high- and low-resolution APLOF ground truth calculated via the Equation 3.6, with a single Labits layer as its basic input unit. This lightweight network (0.524M parameters) efficiently captures local movement information from Labits layers, exhibiting high-quality APLOF estimation. In the main trajectory estimation pipeline, only the encoder part is used, reducing the parameter count to 0.293M. The Labits-to-APLOF net is trained using a StepLR scheduler for 60 epochs and 75k iterations. Training on an NVIDIA RTX 3090 GPU took 35 hours, achieving a low error rate. The model converged with a minimal total loss (Equation 3.7) of 0.084, consisting of LR APLOF L1 loss (0.056) and HR APLOF L1 loss (0.028). Additional qualitative results are detailed in the Supplementary Material.

Main Pipeline Results. Figure 3.3 presents the visualization of our method’s results on the MultiFlow dataset. As shown, both the HR and LR versions of the predicted APLOF closely align with the corresponding ground truth in the early stages, benefiting the main model and

offering additional guidance. For applications that require only the optical flow of active pixels, early exiting here is feasible. Furthermore, the final pixel trajectories and the resulting dense end-point optical flow accurately align with the ground truth, effectively handling challenging scenarios involving small objects and sharp details. More qualitative results are shown in the Supplementary Material.

Table 3.2. Results on MultiFlow dataset. TEPE and TAE represent trajectory-based EPE and AE. Metrics in parentheses are from a linear motion model, indicating these methods aren’t optimized for trajectory prediction. Methods using both events and images are highlighted in grey. Percentage decreases are relative to DCT-RAFT, the top baseline. “w/o” means without.

Method	Input	Trajectory Metrics		2-View Metrics	
		TEPE	TAE	EPE	AE
RAFT	<i>I</i>	(6.89)	(19.31)	7.42	6.71
RAFT + GMA	<i>I</i>	(5.14)	(16.35)	1.47	1.56
E-RAFT	<i>E</i>	(6.70)	(18.44)	7.56	6.19
E-RAFT+Bézier	<i>E</i>	2.62	5.92	4.54	6.06
DCT-RAFT	<i>E</i>	1.85	4.61	3.37	4.80
DCT-RAFT	<i>E+I</i>	1.29	3.35	2.27	3.19
Labits-RAFT (Ours)	<i>E</i>	1.32 ↓ 29%	3.14 ↓ 32%	2.50 ↓ 26%	3.37 ↓ 30%
Labits-RAFT (Ours)	<i>E+I</i>	0.66 ↓ 49%	1.72 ↓ 49%	1.08 ↓ 52%	1.45 ↓ 55%
Ours w/o APLOF Features	<i>E</i>	1.53 ↓ 17%	3.69 ↓ 20%	2.83 ↓ 16%	3.89 ↓ 19%
Ours w/o APLOF Features	<i>E+I</i>	1.01 ↓ 22%	2.63 ↓ 21%	1.64 ↓ 28%	2.25 ↓ 29%
Ours w/o APLOF & Labits	<i>E</i>	1.83	4.57	3.29	4.74
Ours w/o APLOF & Labits	<i>E+I</i>	1.16	3.01	1.99	2.78

To evaluate the effectiveness of our approach, we compare it with four previously published baselines: RAFT ([146]), RAFT-GMA ([74]), E-RAFT ([50]), and DCT-RAFT ([51]). Specifically, RAFT and RAFT-GMA are frame-based approaches utilizing the RAFT architecture, whereas E-RAFT and DCT-RAFT are recent event-based approaches. DCT-RAFT also has a version that takes both frames and events as input. We select these baselines for comparison because they are all derived from the RAFT framework, providing a consistent basis for evaluating our method’s performance.

As presented in Table 3.2, our method, Labits-RAFT, outperforms other methods across all metrics by a remarkably large margin. Our proposed method reduces the TEPE by 49%, from

1.29 to 0.66, compared to the previous state-of-the-art. Similarly, the Trajectory Angular Error (TAE) is reduced by 49%, from 3.35 to 1.72. These results indicate that approaches directly predicting pixel displacements are not suitable for accurate pixel trajectory estimation (e.g., RAFT, RAFT + GMA, E-RAFT). While incorporating Bézier estimation into E-RAFT improves trajectory and two-view metrics, the estimated flow still has room for improvement. DCT-RAFT, which integrates image frames and introduces correlation features, effectively reduces errors in both trajectory and two-view metrics. However, DCT-RAFT’s reliance on voxel grids leads to a significant loss of fine-grained temporal information, representing a critical limitation. Finally, our method also achieves improvements on two-view metrics (52% on EPE and 55% on AE), additionally demonstrating its superiority in accumulative optical flow estimation.

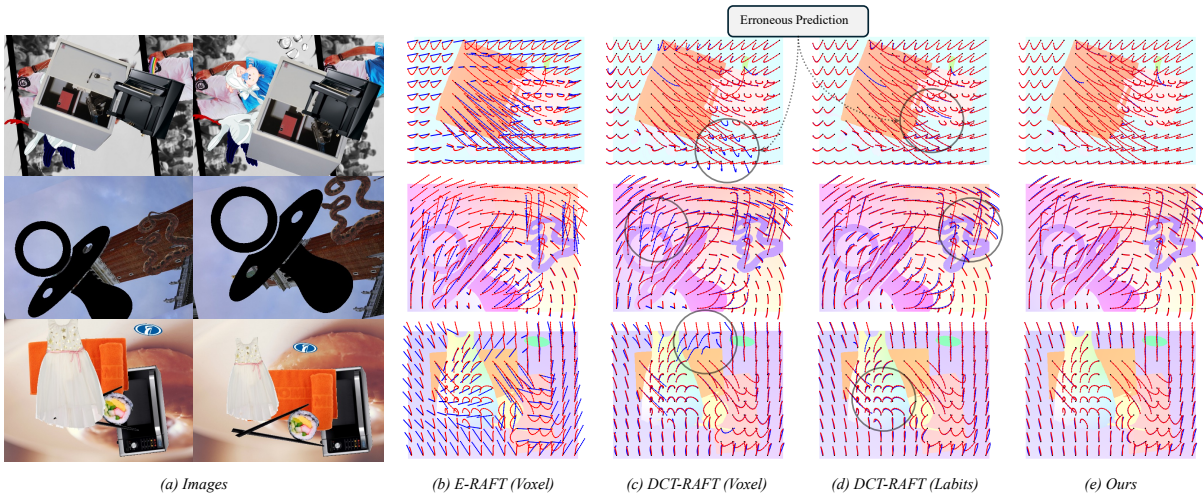


Figure 3.4. Trajectory predictions on the MultiFlow dataset by our proposed model and baseline methods. Ground truth Bézier trajectories are shown in red, while predictions are depicted in blue. The background displays the ground truth optical flow to highlight moving objects. Our model’s predicted trajectories significantly outperform those of all baseline methods.

Ablation Studies. We conduct ablation studies to assess the contributions of our main innovations in dense continuous-time trajectory estimation: the novel Labits event representation and the Labits-to-APLOF net. We carry out two sets of experiments: one excluding the Labits-to-APLOF network and another further replacing Labits with voxel grids. The results, shown in the last four rows of Table 3.2, demonstrate significant performance declines without these modules

in both pure event and event+image scenarios, confirming the effectiveness of both Labits and APLOF features.

Furthermore, Fig. 3.4 provides qualitative comparisons, showcasing two frames and the pixel-wise movement trajectory between them. Our method is compared with the top baseline methods and key ablated results. The superior curve proximity between our predicted trajectory and the ground truth highlights the benefits of incorporating Labits’ implicit local speed cues and APLOF features.

Implementation Details. We implemented our models in PyTorch, training them on the MultiFlow dataset from scratch with AdamW optimizer ([103]), gradient clipping in the range of $[-1, 1]$, and a OneCycle learning rate scheduler. We adopted trajectory loss and two-view loss as detailed in ([51]), supervising the training with 10 flow maps and employing a Bézier curve of degree 10 for precise trajectory modeling. Our model configuration utilizes $J = 6$ correlation Labits blocks across time intervals from 0.4 to 0.9 seconds, quantizing time into $N = 41$ and $M = 25$ bins for context and correlation, respectively. To ensure consistency in evaluation, we adopt the same setup as described in ([51]). Training spanned 100 epochs and 250k iterations, requiring approximately 40 hours on four NVIDIA GeForce RTX 3090 GPUs.

Efficiency Analysis. In our systematic performance evaluation, Labits emerged as an exceptionally efficient representation, adept at capturing dense and detailed motion information with minimal computational delay. We assessed the processing times for various representations, as shown in Figure 3.1, utilizing randomly sampled 500 event packets from the MultiFlow validation set, with each packet focusing on the initial 100 milliseconds of data. Labits achieved an average execution time of 0.220s, demonstrating comparable efficiency to the Voxel Grid (0.225s) and surpassing the more computationally intensive TORE Volume (7.624s) and Time Surface (0.358s). Notably, simplest Event Frame recorded the fastest execution time at 0.062s, followed by Event Count at 0.102s. All the testing is conducted on the same device under the same condition. Although Labits is not the quickest, it offers a superior balance of speed and data richness compared to the others.

3.6 Conclusion

In this work, we introduced Labits, a novel event representation that simultaneously retains fine-grained temporal information, meaningful 2D visual patterns, and local speed cues. Labits is the first to achieve this combination. We also developed the Labits-to-APLOF net, which accurately converts Labits into active pixel local optical flows. Together, these innovations significantly improve performance on event-based continuous-time dense trajectory estimation, with a remarkable 49% reduction in trajectory end-point error compared to the top baseline models. The results highlight that, beyond model architecture, event representations also have a transformative impact on the final outcomes. While Labits’ strengths are most apparent in tasks that utilize intermediate motion information, its adaptability to other event-based vision tasks remains an open area for future research. This versatility positions Labits as a valuable asset for advancing event-based vision, offering numerous possibilities for further exploration and development. Detailed analysis of limitations and future research directions could be found in the Supplementary Material.

3.7 Supplementary Material

3.7.1 Model Specifications

Table 3.3. DCT-RAFT and Labits-RAFT model size and performance metrics

Metric	DCT-RAFT	Labits-RAFT*	Labits-RAFT
APLOF Features	N/A	Yes	Yes
Trainable Parameters	6.8 M	6.8 M	24.8 M
Non-trainable Parameters	0	0	524 K
Total Parameters	6.8 M	6.8 M	25.3 M
Estimated Model Size	27.203 MB	27.203 MB	101.288 MB
Average GFLOPS	4316.287	4432.353	5599.052
Average Inference Time	0.133 s	0.135 s	0.192 s
TEPE	1.29	1.01	0.66
TAE	3.35	2.63	1.72



Figure 3.5. Visual Comparison of Labits and One-way Labits: We visually compare Labits and One-way Labits generated from the same icon sample, as presented in Figure 3.1. For the One-way Labits, we create results using only the past or future time ranges, respectively, to illustrate the visual difference.

Table 3.3 provides detailed performance metrics for the DCT-RAFT and Labits-RAFT models (* means Labits-RAFT w/o APLOF features), based on 100 iterations on an NVIDIA-A10 GPU. The DCT-RAFT model, with 6.8 million parameters, delivers fast inference times but higher error rates, while the more complex Labits-RAFT, with 25.3 million parameters, exhibits a slight increase in inference time but significantly reduces errors by 49%, illustrating the trade-offs between model complexity and performance efficiency.

3.7.2 Instantaneous Speed Prediction Error Estimation Analysis

The motivation behind the development of Labits stems from the necessity of accurately estimating continuous dense optical flow—a process that requires precise understanding of instantaneous pixel-level movements at numerous intermediate moments within a scene. These movements information, which we refer to as local motion anchors, play a crucial role in tasks that demand detailed motion understanding. Based on the formula of the backward (Eq.12) and central difference (Eq.13), combined with Taylor expansion (Eq.14, Eq.15), we know the error estimation for backward difference is $O(\delta t)$, while the one for central difference is $O(\delta t^2)$,

meaning central difference provides higher accuracy.

$$f'(x_n) \approx (f(x_n) - f(x_{n-\delta t})) / \delta t \quad (3.12)$$

$$f'(x_n) \approx (f(x_{n+\delta t}) - f(x_{n-\delta t})) / 2\delta t \quad (3.13)$$

$$f(x_{n+\delta t}) = f(x_n) + \delta t f'(x_n) + 0.5(\delta t)^2 f''(x_n) + O((\delta t)^3) \quad (3.14)$$

$$f(x_{n-\delta t}) = f(x_n) - \delta t f'(x_n) + 0.5(\delta t)^2 f''(x_n) - O((\delta t)^3) \quad (3.15)$$

In each Labits layer, the values represent relative time information, enabling the scaled instantaneous local speed to be calculated as the inverse of the 2D Labits gradient on the image sensor plane. To improve local speed estimation at active pixels, the central difference method, which requires relative time values from both the past and future, is preferred. Therefore, the Labits implementation incorporates both near-past and near-future events at each reference point. In contrast, existing time surface-style representations, which compress information solely from the past, are less accurate for local speed estimation and leave a significant portion of pixels in front of moving objects empty, reducing information density, especially when the time range is short.

3.7.3 Analysis of the Bidirectional Structure in Labits

Since Bidirectional is a key feature of our event representation, this section can provide more concrete evidence to show the effectiveness of this design scheme. We conduct additional ablation studies on whether the representation takes in bidirectional information during the representation generation. For the so-called “One-way Labits”, only events triggered during the past time range is considered, following the traditional time surface generation strategy. We retrain the entire pipeline using the “One-way Labits”, including the Labits-to-APLOF net and the Labits-RAFT model. The results of both model is significantly worse compared to our proposed bidirectional Labits. Details are shown in Table 3.4 and Table 3.5. Additionally, visual comparison of Labits and One-way Labits are detailed in Figure 3.5.

Table 3.4. Final APLOF losses comparison between Labits and one-way Labits.

Model	Total Loss	LR APLOF L1 Loss	HR APLOF L1 Loss
Labits-to-APLOF Net	0.084	0.056	0.028
Labits-to-APLOF Net (One-way)	0.357	0.199	0.158

Table 3.5. Performance comparison of Labits and one-way Labits.

Method	Input	TEPE	TAE	EPE	AE
DCT-RAFT	E+I	1.29	3.35	2.27	3.19
Labits-RAFT (Ours)	E+I	0.66	1.72	1.08	1.45
Labits-RAFT (One-way)	E+I	0.72	1.87	1.22	1.68

3.7.4 Ablation Study on Time Bin Size

Besides, we conduct ablation studies on the number of Probe Times to facilitate comprehensive evaluations. Detailed ablation studies focus on different bin configurations for Labits and Voxel Grid, specifically in the context of trajectory estimation. These studies utilize the same pretrained Labits-to-APLOF network, which was originally trained with Labits featuring 65 bins and a bin size of 0.0125 seconds. We design our study to cover a variety of scenarios by setting bin time spans at 0.0125s, 0.0250s, 0.0500s, and 0.1000s, facilitating a direct comparison with DCT-RAFT [51]. Table 3.6 provides robust results that thoroughly validate the efficacy of Labits, confirming that the pretrained Labits-to-APLOF net can adaptively handle different time bins.

Table 3.6. Performance comparison of DCF-RAFT and Labits-RAFT using different bin setups.

Model	Bin Size	TEPE	TAE	EPE	AE
DCF-RAFT (voxel)	0.1000	1.81	4.53	2.87	3.97
Labits-RAFT (Labits)	0.1000	0.88 ↓ 51%	2.35 ↓ 48%	1.43 ↓ 50%	1.97 ↓ 50%
DCF-RAFT (voxel)	0.0500	1.51	3.82	2.61	3.63
Labits-RAFT (Labits)	0.0500	0.75 ↓ 50%	1.94 ↓ 49%	1.22 ↓ 53%	1.66 ↓ 54%
DCF-RAFT (voxel)	0.0250	1.40	3.56	2.45	3.41
Labits-RAFT (Labits)	0.0250	0.72 ↓ 49%	1.86 ↓ 48%	1.17 ↓ 52%	1.58 ↓ 54%
DCF-RAFT (voxel)	0.0125	1.29	3.35	2.27	3.19
Labits-RAFT (Labits)	0.0125	0.66 ↓ 49%	1.72 ↓ 49%	1.08 ↓ 52%	1.45 ↓ 55%

3.7.5 Additional Comparison Experiments

To further validate the performance of the Labits-RAFT model, we train a Voxel-to-APLOF net under the same conditions as the Labits-to-APLOF net. For active pixels, we employ a zero mask. Additionally, we implemented a complete model pipeline using voxel and the APLOF feature generated by the pretrained Voxel-to-APLOF net. Table 3.7 provides a comparative overview of the minimal total loss (Equation 3.7) between the Labits-to-APLOF net and the Voxel-to-APLOF net. Furthermore, Table 3.8 details the performance metrics of the complete model pipeline that incorporates the Voxel-to-APLOF features and voxel.

Table 3.7. Performance Comparison of Labits-to-APLOF and Voxel-to-APLOF Net.

Model	Total Loss	LR APLOF L1 Loss	HR APLOF L1 Loss
Labits-to-APLOF Net	0.084	0.056	0.028
Voxel-to-APLOF Net	0.217	0.080	0.137

Table 3.8. Ablation study results on Labits and voxel-to-APLOF Net.

Method	Input	TEPE	TAE	EPE	AE
DCT-RAFT	E+I	1.29	3.35	2.27	3.19
Labits-RAFT (Ours)	E+I	0.66	1.72	1.08	1.45
DCT-RAFT + Voxel-to-APLOF features	E+I	0.84	2.25	1.40	1.97

Furthermore, we deploy event representations that are computationally feasible and align well with the layered representation requirements. In this context, we implement and test the Former-Latter Event Groups ([88]) and Gaussian Weighted Polarities ([40]), as shown in 3.9. These methods fit within our computational constraints and ensure fair comparisons across different tasks. This comparison not only facilitates a thorough evaluation of each representation’s efficacy under consistent conditions but also validates the effectiveness of Labits.

Table 3.9. Comparison of more recent layered event representations applied to the RAFT architecture ([146]) for trajectory estimation.

Method	Input	TEPE	TAE	EPE	AE
Former-Latter Event Groups	E+I	3.15	7.36	3.81	5.07
Gaussian Weighted Polarities	E+I	3.47	8.16	4.23	5.46
DCT-RAFT	E+I	1.29	3.35	2.27	3.19
Labits-RAFT (w/o APLOF)	E+I	1.01	2.63	1.64	2.25
Labits-RAFT (Ours)	E+I	0.66	1.72	1.08	1.45

3.7.6 Limitations and Future Directions

Limitations

The proposed solution has specific use cases. Labits demonstrates excellent performance in dense continuous-time trajectory estimation tasks. However, this is not only due to our novel event representation and a well-designed model pipeline, but also because the task definition and the selected dataset inherently leverage Labits’ ability to preserve rich intermediate temporal information. In this task, where continuous motion trajectories of each pixel over the target time span are evaluated, and these intermediate states are included in the metrics, our solution significantly improves results. Conversely, for tasks less sensitive to intermediate states and focused only on final outcomes, such as object detection or optical flow estimation, Labits may not outperform some well-established representations and solutions. In these cases, the additional fine-grained intermediate information may even become irrelevant noise. Different tasks require different types of information, and while our solution is not a one-size-fits-all approach, it can exhibit substantial advantages for suitable tasks.

Labits omits event density information. Labits is designed to maximize the extraction and retention of motion information at intermediate time points. However, this design sacrifices event density information, which might be crucial for certain computer vision tasks. Future work could explore combining multiple representations to create a more generalizable framework that balances these trade-offs.

We did not conduct extensive adaptation experiments across various event camera-based vision tasks. This paper focuses on providing an exceptional solution for the specific task of dense continuous-time trajectory estimation, rather than proposing a universal representation. Applying Labits is not as simple as directly replacing previously used representations in various tasks. Instead, it requires tailored feature fusion designs for Labits, APLOF, and the tasks themselves. Redesigning and conducting training, testing, and ablation studies for diverse tasks is beyond the scope of a single conference paper and was not our intention. This work remains centered on the specific task highlighted in the title.

Future Research Directions

This paper introduces Labits along with a series of models and structures that fully exploit its capabilities. We have demonstrated its unparalleled suitability for dense continuous-time trajectory estimation and validated the effectiveness of Labits and its associated structures through comprehensive ablation studies. Potential future research directions include:

Expanding Labits Applications to Diverse Tasks Adapting Labits and its associated models to other similar tasks, such as human keypoint tracking, object trajectory tracking, or event-based video interpolation. These adaptations would undoubtedly require redesigned model structures tailored to each task’s specific needs.

Targeted Tracking Focused on Active Pixels Leveraging the properties of active pixels proposed in this work to generate representations for selective regions rather than globally. This approach could enable more efficient tracking by focusing on active pixel clusters.

Refinement of Labits Combining Labits with traditional event representations like Voxel Grid to achieve complementary strengths. Labits captures fine-grained temporal information for extracting local motion states at intermediate moments but lacks event density information. Conversely, Voxel Grid provides event density information but discards the fine-grained temporal precision that event cameras excel at. Combining these representations may open new possibilities for many tasks.

Utilizing Polarity Information Labits, for efficiency and information density considerations, does not differentiate between events of different polarities. Exploring more optimized handling of polarity-specific events may further enhance its performance in relevant computer vision tasks.

In conclusion, the innovations proposed in this paper are versatile and can be extended or improved upon. We hope more researchers will contribute to developing better event representations, unlocking the full potential of event cameras.

Acknowledgments

Chapter 3, in full, is based on the paper from Zhongyang Zhang, Jiacheng Qiu, Shuyang Cui, Yijun Luo, and Tauhidur Rahman, “Labits: Layered Bidirectional Time Surfaces Representation for Event Camera-based Continuous Dense Trajectory Estimation” submitted for publication in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2025. The dissertation author is the primary investigator and author of this paper.

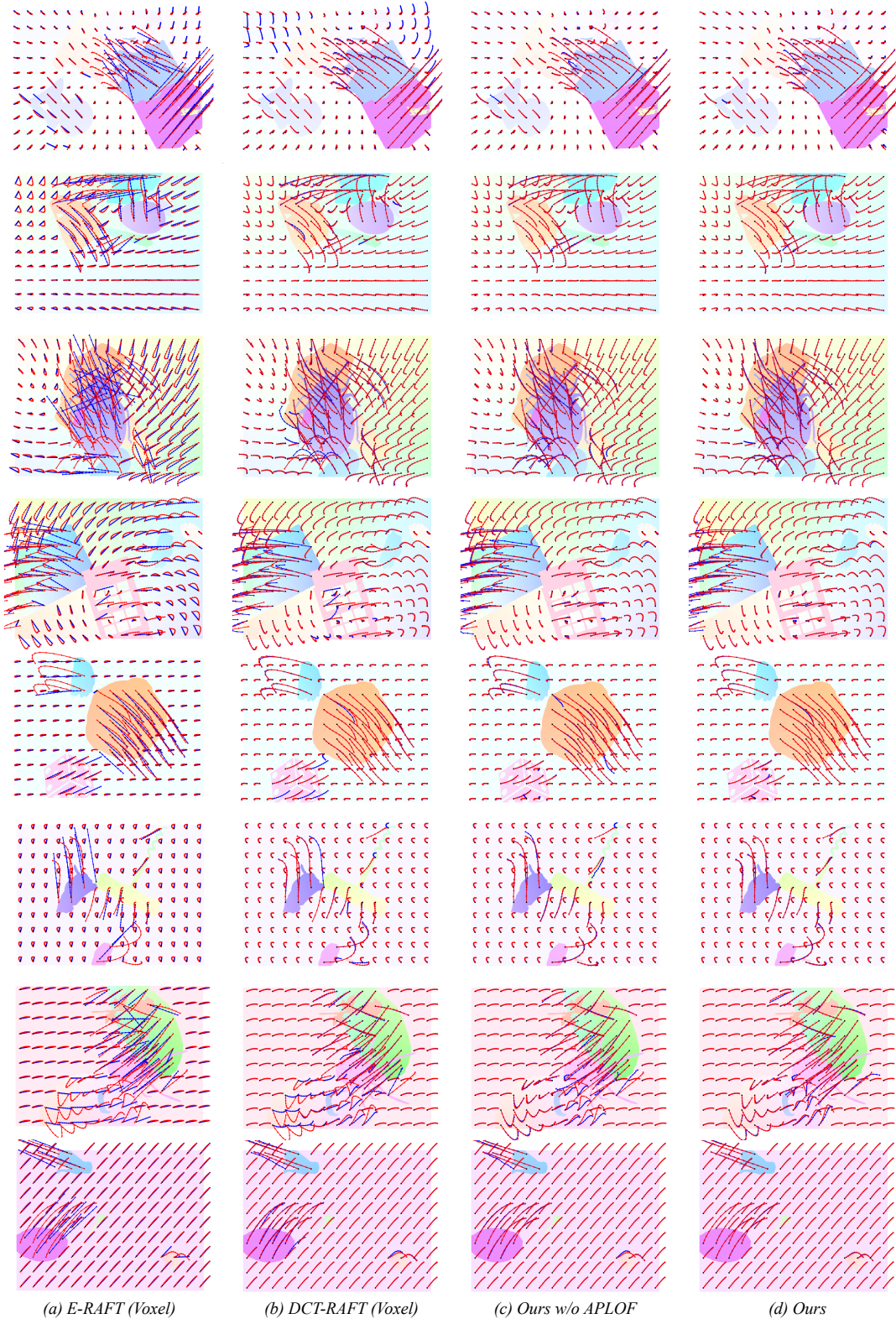


Figure 3.6. Comparison of trajectory predictions: between baseline methods and our approach.

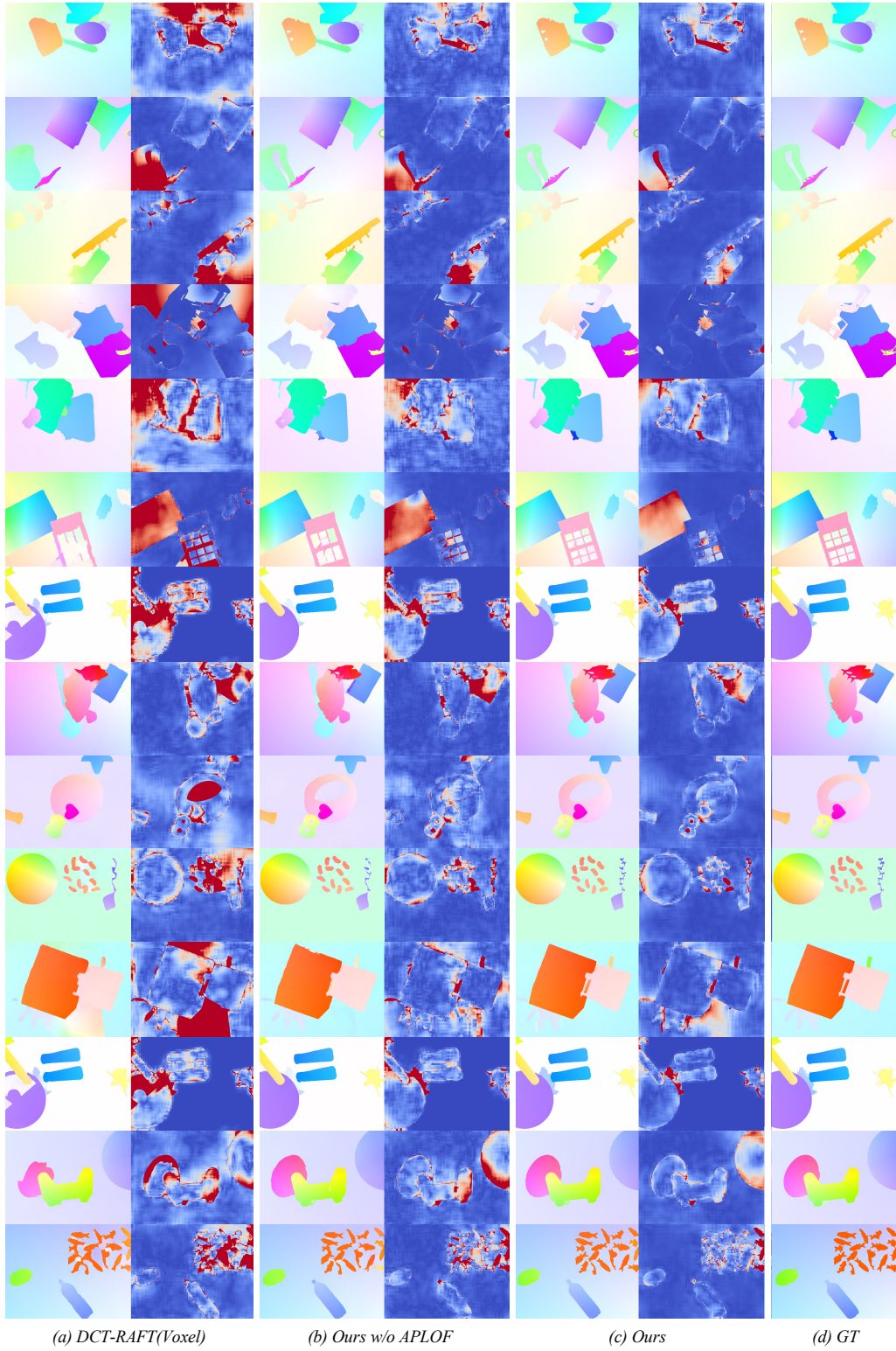


Figure 3.7. Comparison of Optical Flow Estimations: Each method includes a second column displaying the error map generated by comparing the predictions to the ground truth.

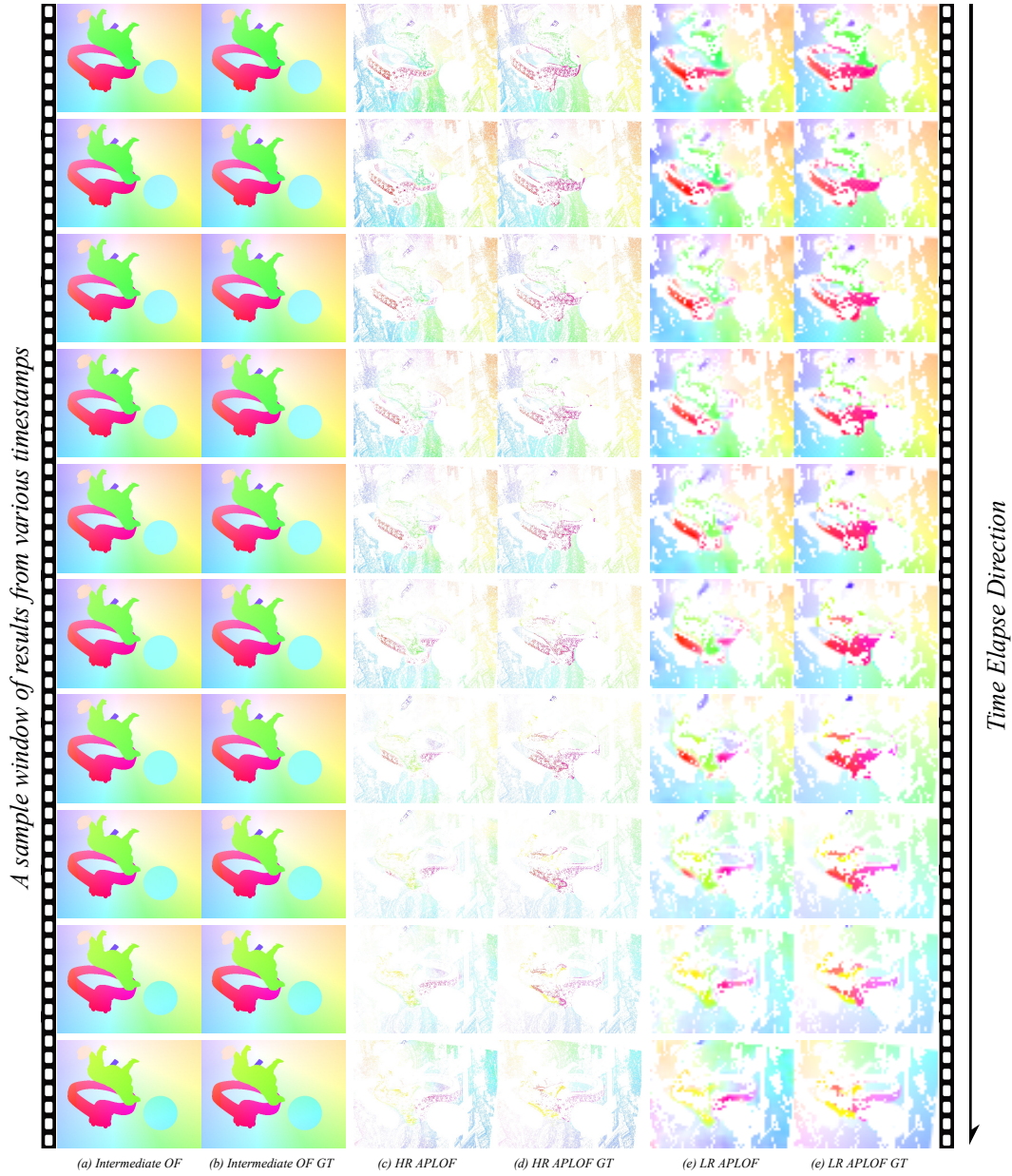


Figure 3.8. Visualization of Intermediate Prediction Results: We display dense, time-continuous prediction outcomes for each trajectory within a sample window from the MultiFlow dataset [51]. The visualization includes ten intermediate instances of optical flow (OF), HR APLOF, LR APLOF, and the corresponding GT, presented in three paired columns. Timestamps for the predictions correspond to those of the GT. The trajectory spans a time range of 0.5 seconds (context). We evenly sampled 10 intermediate timestamps, and show the predicted and ground truth OF between them and the start time. Meanwhile, APLOFs show instantaneous optical flows.

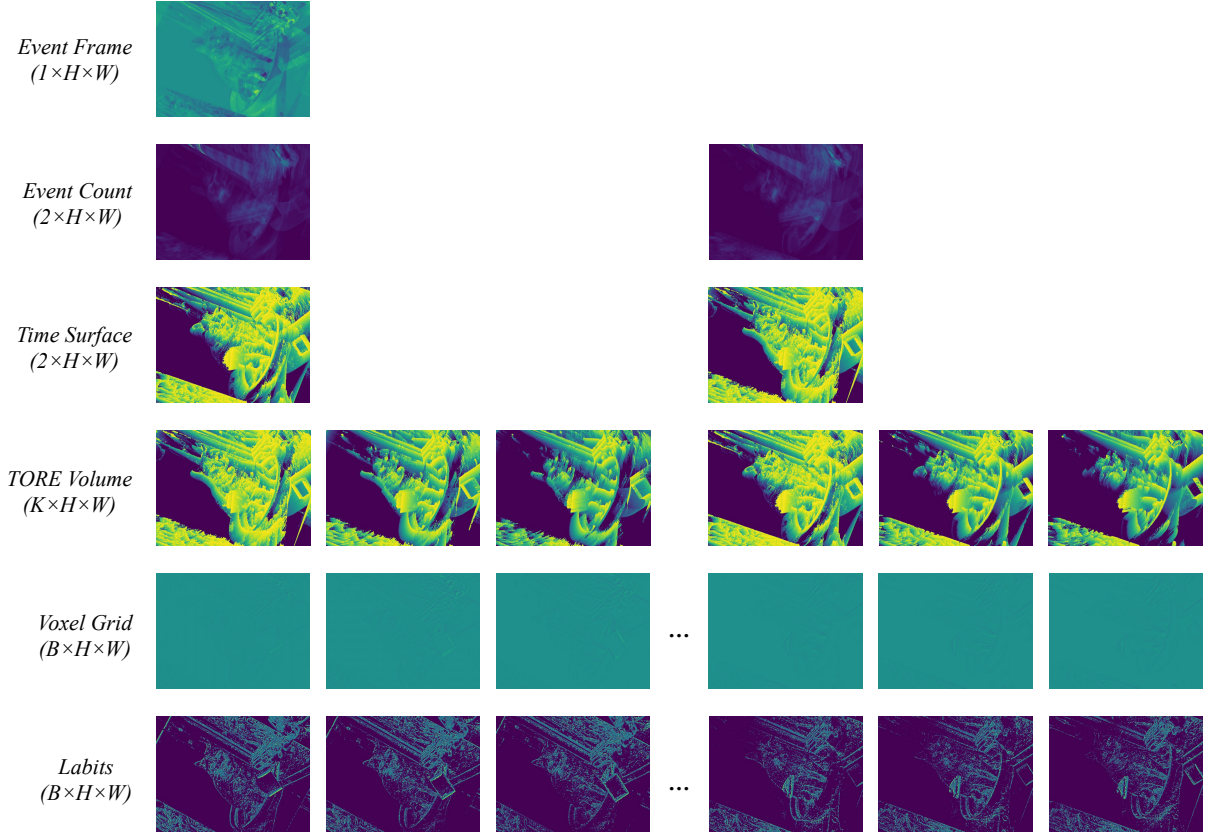


Figure 3.9. Comparative Visualization of Event Representations: This figure presents event data sampled between 100 ms and 900 ms from the MultiFlow dataset, showcasing diverse representations such as Time Surface, Event Frame, TORE Volume ($K=3$), Voxel Grid ($B=65$), and Labits ($B=65$). These hyperparameters are selected following the corresponding paper’s settings. Each method distinctively captures and structures event information. The first three plots of TORE Volume correspond to positive events’ representation layers, while negative for the last three. Voxel Grid and Labits visualized the first three and last three bins, respectively. For time surface and event count, the first and second frames represent the count of negative and positive events.

Chapter 4

Larver: Learning Alignment and Representation for Versatile Event Regeneration

4.1 Introduction

Event cameras, with their unique characteristics, differ significantly from conventional RGB frames or video. Unlike traditional cameras that capture images at fixed intervals, event cameras operate asynchronously at the pixel level, recording only changes in intensity [45, 25]. This enables them to achieve extremely high temporal resolution and dynamic range while minimizing latency. Moreover, event cameras naturally filter out static backgrounds, capturing only the most relevant dynamic information. Despite these advantages, event data is scarce, and its inherently different representation compared to RGB data presents substantial challenges for integration into most computer vision workflows.

Several works have aimed to simulate raw event streams from existing video data, such as v2e [69] and V2CE [174]. These simulators have achieved promising results by converting video frames into event streams, demonstrating the potential of this modality. However, they remain dependent on large amounts of video data for training and conversion, limiting their applicability to real-world scenarios where event data is generated directly from dynamic scenes rather than simulated from conventional video sources.

While recent advancements in multimodal learning and generative models have garnered attention [64, 56, 55, 120], the event modality has received comparatively little focus. There are

several reasons for this. First, aligning event data with conventional RGB frames is non-trivial due to the fundamentally different data representations. Second, the scarcity of large-scale, high-quality event datasets hinders the widespread adoption of event-based learning techniques. Finally, the development of generative models tailored specifically for event data remains in its infancy, posing additional challenges for tasks like event-based video synthesis and cross-modal translation. Given these difficulties, event cameras face significant challenges when integrated into standard vision pipelines. To the best of our knowledge, there is no established or integrated framework that aligns the event modality with other common modalities while also supporting raw event stream generation.

In this work, we present a novel framework that leverages state-of-the-art deep learning models to bridge the gap between event cameras and mainstream modalities. Our approach consists of four key stages: (1) encoding event data into a structured representation, (2) utilizing pretrained image encoders for feature extraction and alignment, (3) employing diffusion models to generate high-quality event representations, and (4) enabling multimodal synthesis, including event-to-image, event-to-video, and event-to-text transformations.

Our contributions can be summarized as follows:

1. **FLER Representation:** We introduce FLER (Frame-Like Event Representation), a novel compressed representation for event data that facilitates knowledge transfer from conventional vision models.
2. **Unified Event Embedding:** We integrate event data into the ImageBind framework by fine-tuning its image encoder, enabling seamless alignment with other modalities.
3. **FLER-Based Event Generation:** We fine-tune Stable Diffusion XL using FLER-labeled images, allowing controllable event synthesis from text and multimodal embeddings.
4. **Event Voxel Reconstruction:** We develop a pipeline to reconstruct event voxel grids from generated FLERs, completing the event generation process.

5. **Multimodal Applications:** We demonstrate event-based cross-modal synthesis, including events-to-image, events-to-text, and event-enhanced video generation.

By leveraging pretrained vision models and generative techniques, our framework opens new possibilities for event camera applications beyond traditional handcrafted methods. We believe this approach will facilitate broader adoption of event-based vision and inspire future research in event-driven deep learning.

4.2 Related Work

Event camera applications. Event cameras have been widely adopted in various computer vision and robotics applications, particularly in motion-centric tasks such as optical flow estimation, object tracking, and human pose estimation [51, 175, 173]. Their high temporal resolution and asynchronous operation enable precise motion analysis, even in challenging conditions. Beyond perception tasks, event-based vision has advanced video processing techniques like frame interpolation, motion deblurring, and HDR reconstruction [162, 142, 188], leveraging sparse yet information-rich event streams. These advancements continue to push the boundaries of high-speed, low-latency vision processing beyond traditional frame-based methods, enabling applications in autonomous systems, augmented reality, and neuromorphic computing [42, 92, 149].

Multimodal representation learning. Multimodal learning has gained significant traction in recent years, driven by the need to integrate heterogeneous data sources such as images, videos, audio, and text into unified, meaningful embeddings. Foundational works such as CLIP [120] and ALIGN [72] have demonstrated the power of contrastive learning for aligning vision and language representations, enabling zero-shot generalization across diverse tasks. More recently, ImageBind [55] has extended this paradigm by learning a joint embedding space across six modalities, including images, audio, depth, thermal, IMU, and text, demonstrating the potential for unified multimodal learning. Despite these advances, event-based vision

remains underexplored in the context of multimodal representation learning. Existing approaches primarily focus on a limited number of modalities, typically text and images, as seen in methods like EventCLIP [160] and EventBind [177]. While these approaches leverage event-based data for specific tasks, their reliance on modality-specific architectures restricts their applicability to broader multimodal learning scenarios, preventing seamless integration with emerging unified frameworks.

Generative models for unconventional modalities. Generative models have achieved remarkable success in handling conventional modalities such as images, videos, and text. A prominent example is Stable Diffusion [127], which leverages latent diffusion processes to generate high-quality images and has become a cornerstone in generative AI. Its ability to model complex data distributions has inspired extensions to diverse tasks, including video synthesis [65]. To adapt these powerful models to specific tasks or domains, fine-tuning techniques like LoRA (Low-Rank Adaptation) [67] have gained popularity. LoRA enables efficient adaptation of large pre-trained models by introducing low-rank updates, significantly reducing the computational cost while preserving the model’s generalization capabilities. Despite these advancements, generative models for unconventional modalities, such as event camera data, remain underexplored. Event cameras output sparse, asynchronous data with high temporal resolution, posing unique challenges that differ fundamentally from conventional data types. Although task-specific solutions, such as event-based video reconstruction or deblurring [184, 156], have been proposed, a generalizable framework for aligning and fusing event data with mainstream modalities like images, videos, and text is still lacking. Developing such a framework is critical to unlocking the full potential of event-based vision and enabling broader applications in multimodal AI.

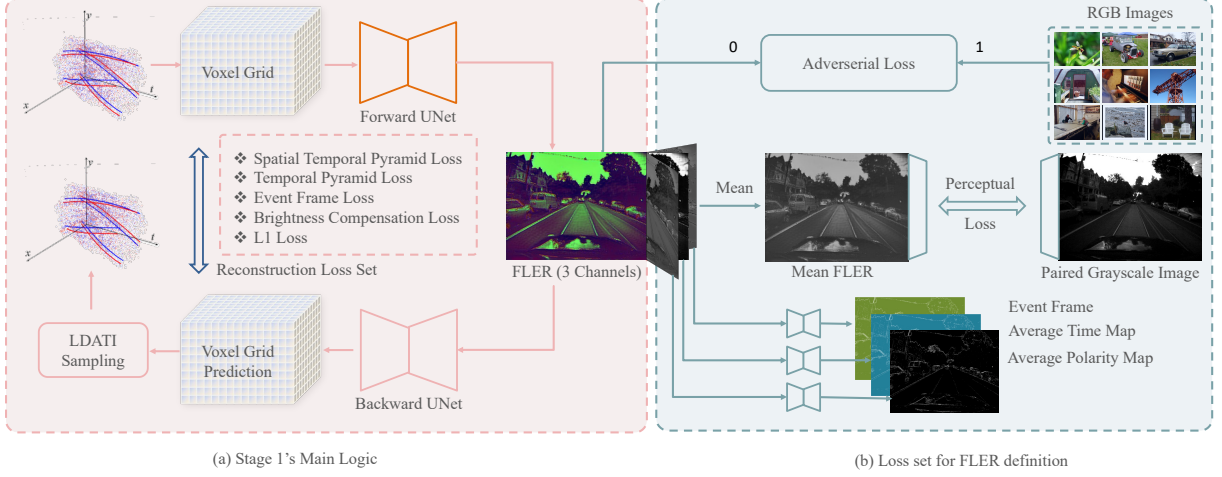


Figure 4.1. Pipeline and the FLER definition loss set of stage 1. Losses are divided into two parts: event voxel reconstruction losses and FLER-definition losses.

4.3 Proposed Method

4.3.1 Overview

In this work, we propose a novel pipeline that enables seamless integration of event camera data with other mainstream modalities such as images, videos, audio, and text. Event cameras offer unique advantages, including high temporal resolution, low latency, and a large dynamic range, making them well-suited for tasks such as frame interpolation, tracking, and deblurring. However, challenges remain in aligning event data with other modalities, the scarcity of high-quality labeled datasets, and the absence of a generative model capable of synthesizing realistic event streams. Existing multimodal approaches involving event cameras are often application-specific, requiring custom models, data collection, and training for each scenario.

To address these challenges, we introduce a four-stage pipeline. First, we perform representation learning to transform event voxel grids into a more compact and interpretable representation called Frame-Like Event Representation (FLER). This representation is learned using an autoencoder and designed to resemble RGB images while preserving essential temporal and polarity information. Second, we integrate FLER into ImageBind, a multimodal alignment framework, by fine-tuning the image encoder to create a dedicated FLER encoder. This enables

FLER embeddings to align with embeddings from other modalities. Third, we leverage this unified embedding space to facilitate event generation via diffusion models. By treating FLER as a unique style in Stable Diffusion XL (SDXL), we fine-tune the model on annotated grayscale images with FLER labels, enabling event generation conditioned on text, images, or other modalities. Finally, we reconstruct event voxel grids from generated FLERs using the trained autoencoder decoder and apply a voxel-to-event sampling method to obtain the final raw event stream.

4.3.2 Representation Learning: FLER

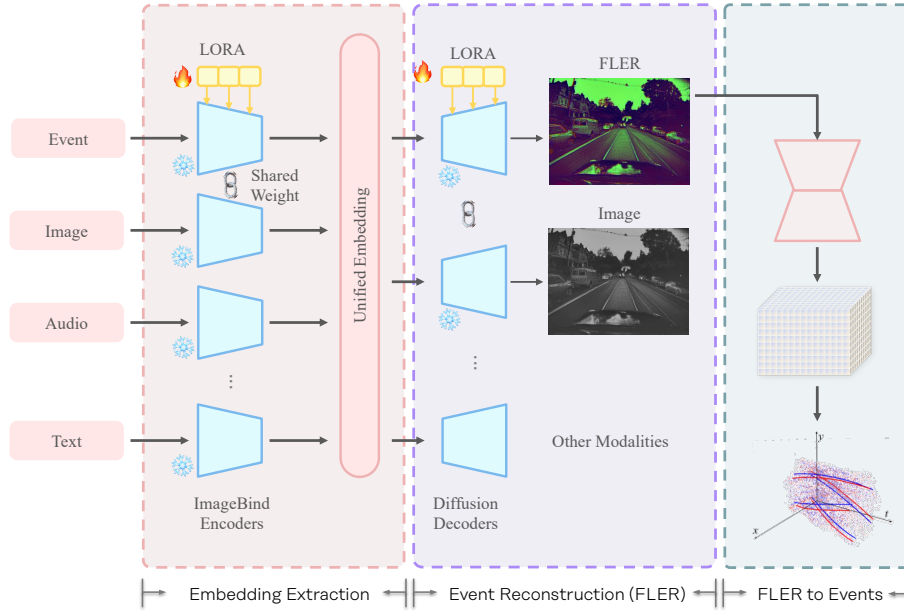


Figure 4.2. Pipeline of the stage 2 and 3, including the event representation FLER-based encoder and diffuser fine-tuning. Both the fine-tuning for encoder and diffuser are via Lora.

Given the lack of large-scale, high-quality labeled datasets for event cameras, directly training generative models for event streams is infeasible. However, event data shares many visual properties with images, suggesting that transferring knowledge from the image domain could be beneficial. To achieve this, we propose FLER—a compact, three-channel event representation that resembles RGB images in structure while encoding essential temporal dynamics of event streams.

FLER is designed with three key objectives: (1) it must maintain a three-channel format similar to RGB images to facilitate compatibility with pretrained models, (2) it should exhibit a clear and structured visual pattern to enhance interpretability, and (3) it must allow accurate reconstruction of the original event voxel grid with minimal information loss. Since no standard exists for defining such a representation, we employ a learning-based approach using an autoencoder.

Our autoencoder consists of a UNet-based encoder-decoder architecture. The input event voxel grid, structured as $[2T, H, W]$ (where T represents temporal bins, and 2 accounts for polarity), is compressed into a $[3, H, W]$ FLER representation. The decoder reconstructs the voxel grid from FLER, ensuring that the transformation preserves critical event information. The three channels of FLER are explicitly designed: the first channel encodes the relative event density at each pixel, the second channel represents the average relative event timing, and the third channel captures the average event polarity.

To ensure a stable and meaningful FLER representation, we define a comprehensive set of loss functions. The representation learning loss consists of five components: (1) a perceptual loss ensuring that FLER visually resembles the grayscale frame corresponding to the event stream, (2) an adversarial loss encouraging FLER to follow the distribution of natural images, (3) an auxiliary task where the first channel of FLER is used to reconstruct an event frame supervised via L1 loss, (4) a loss enforcing the second channel to predict the average event timing, and (5) a loss guiding the third channel to predict the average event polarity. Together, these losses ensure that FLER captures distinct and interpretable aspects of the original event voxel grid.

For voxel grid reconstruction, we adopt the loss set from V2CE, which effectively reconstructs sparse voxel grids. This includes spatial-temporal pyramid (STP) loss, temporal pyramid (TP) loss, event frame loss, and brightness compensation (BC) loss. These losses address the high sparsity of voxel grids and ensure accurate reconstruction without excessive zero predictions.

By transforming event data into FLER, we enable the use of image-based priors and

pretrained models, facilitating knowledge transfer from the image domain to the event modality.

4.3.3 Unified Embedding for Events

To integrate event data into multimodal learning, we incorporate FLER into ImageBind, a contrastive learning framework that aligns various modalities into a shared embedding space. This allows event data to be directly comparable to other modalities such as images, video, audio, and text.

We achieve this integration by fine-tuning the image encoder in ImageBind using LoRA [67] (Low-Rank Adaptation). Since FLER has the same dimensional structure as RGB images, we can efficiently adapt the existing image encoder to process FLER representations. LoRA enables rapid convergence and improved performance by introducing trainable low-rank matrices while keeping most pretrained parameters frozen.

To evaluate the effectiveness of this alignment, we conduct a retrieval task where event stream clips are used to retrieve their corresponding RGB images. Our fine-tuned encoder achieves a retrieval accuracy of 98.3%, demonstrating strong alignment between FLER and other modalities. Additionally, we measure the embedding distance between FLER and corresponding images, showing that their embeddings are closely aligned in the shared space.

4.3.4 Event Generation via Diffusion Models

Training a generative model for event data from scratch is impractical due to the lack of large-scale labeled event-text datasets. Instead, we leverage FLER as an intermediate representation and treat it as a unique style in Stable Diffusion XL (SDXL). This enables us to generate FLERs using text or other modality embeddings, which can then be converted back into event streams.

To enable FLER generation, we collect grayscale images paired with event data from the MVSEC dataset. Since these grayscale images correspond to event streams, they share the same semantic information and can be annotated with descriptive text. We use GPT-4o to generate

detailed scene descriptions for each grayscale image, ensuring that labels cover key objects, lighting conditions, camera angles, and other relevant attributes. Additionally, we append a “FLER” tag to each label to establish FLER as a distinct visual style.

Using these annotated images, we fine-tune SDXL with LoRA [67] to generate FLERs conditioned on text embeddings. Since ImageBind’s unified embeddings are compatible with CLIP embeddings used in SDXL, we can generate FLERs using embeddings derived from any modality.

4.3.5 Reconstruction of Event Voxel Grids

Once FLERs are generated, we reconstruct the original event voxel grids using the trained autoencoder decoder. This step ensures that the generated FLERs can be transformed back into valid event representations.

Generating FLER first and then converting it to voxel grids is advantageous because voxel grids have strict temporal dependencies that are difficult for generative models to capture directly. By decoupling content generation from precise temporal reconstruction, we allow FLER to serve as an intermediate representation that captures spatial-temporal information in a structured manner.

The reconstructed voxel grid, generated by a neural network, may contain voxels with non-zero values, even when they should ideally be zero. Consequently, these voxels can result in the generation of spurious events near the beginning of the voxel. To enhance the quality of the generated event stream and eliminate these false events, we implemented a temporal pooling-based filter. This filter applies a one-dimensional average pooling layer with a kernel size of 3 to each pixel in the voxel grid. Since each event influences both the current and the subsequent time bin at its spatial coordinate, with a total contribution of 1, an average pooled result of less than $1/3$ indicates a likely false event. To ensure a more lenient filtering process, we established a threshold of 0.1. This mask is then multiplied with the voxel grid to produce the filtered voxel grid.

To obtain the final raw event stream, we apply the voxel-to-event sampling method proposed in V2CE. Specifically, we use the Local Dynamics-Aware Timestamp Inference (LDATI) technique, which refines event timestamps by considering temporal relationships between neighboring voxels. This method ensures that the generated event stream maintains accurate and smooth temporal distributions without time-layering artifacts.

4.3.6 Summarization

Our proposed pipeline enables direct and accurate generation of raw event streams by leveraging FLER as an intermediate representation. FLER naturally encodes temporal information in its unique “color style,” improving temporal reliability compared to direct image-to-event conversion. By transferring knowledge from the well-established image domain, our approach mitigates the challenges posed by the lack of large-scale event datasets. Furthermore, our unified embedding framework allows events to guide the generation of other modalities, including images, videos, and text. This opens new possibilities for event-based multimodal applications, bridging the gap between event cameras and mainstream vision models.

4.4 Conclusion & Future Work

As an emerging vision modality, events serve as a valuable complement to traditional image modalities. This work presents a novel and versatile pipeline designed for event encoding, alignment with other modalities, and condition-guided generation. The new event representation, FLER, facilitates the reuse of prior knowledge transferred from pre-trained image-based models while compressing the spatiotemporal information of an event voxel grid in a recoverable manner. The aligned event embedding further enhances the potential for integrating events into a multimodal pipeline as input. The final stage enables direct event generation for general purposes using a generative model guided by any available modalities. This work transforms the event modality from an emerging edge technology into a “truly useful modality” that can seamlessly interact with other mainstream modalities.

Acknowledgments

Chapter 4, in part, is based on the paper from Zhongyang Zhang, Jiacheng Qiu, Gert Cauwenberghs, Upal Mahbub, and Tauhidur Rahman, “Larver: Learning Alignment and Representation for Versatile Event Regeneration” currently being prepared for submission. The dissertation author is the primary investigator and author of this paper.

Chapter 5

Discussion and Future Work

5.1 Summary of Dissertation

This thesis presents two progressive studies on the application of event cameras in motion-related tasks. Initially, it explores the use of event cameras in human pose estimation, highlighting their unique advantages. It then examines their exceptional performance in dense continuous-time trajectory estimation, further demonstrating their sensitivity to motion. These studies emphasize the importance of event representation in extracting motion information and the necessity of leveraging temporal data in such tasks.

To address the scarcity of datasets in the field of event cameras, this thesis also proposes a series of solutions focused on 3D modeling-based simulations, video-to-event conversion, and modality-aligned generative models. These approaches range from high-level to low-level, from requiring extensive manual modeling intervention to being simple and user-friendly, and from highly constrained generation methods to those allowing control via arbitrary modality inputs. This progression lays a solid foundation for alleviating data scarcity and aligning event modalities with mainstream modalities.

Chapter 1 discusses how to fully utilize the characteristics of event cameras for high-frequency and accurate human pose estimation under extremely low-light conditions. It details the human subject study for real-world data collection and a simulation pipeline for generating event data with Blender. Additionally, this chapter introduces a novel model pipeline that employs

an event representation covering long-term historical information and uses BiConvLSTM to enhance temporal information fusion, achieving accurate and smooth 3D human pose estimation.

Chapter 2 highlights breakthroughs in the video-to-event task. In the synthetic data generation pipeline from Chapter 1, existing v2e tools were used but revealed issues such as excessive synthetic event counts and timestamps clustering at discrete time points, causing a time-layering effect. This chapter addresses these problems by designing a custom loss set to accurately convert video data into event voxel grids and developing a rapid and precise event timestamp resampling method using probabilistic approaches, significantly improving the timestamp accuracy of converted events.

Chapter 3 explores the application of event cameras in dense continuous-time trajectory estimation. The abundant temporal and spatial information in events profiles the motion states of different pixels over time, often discarded or obscured in representation generation. We propose an event representation, Labits, that better retains and extracts local instantaneous motion information, accelerating the calculation of instantaneous velocities at intermediate time points for active pixels, achieving efficient and accurate trajectory estimation.

Chapter 4 presents the latest work on aligning the event modality with other mainstream modalities, enabling event stream generation based on various modalities and generating other modalities through events. This chapter introduces a multi-stage pipeline: first, converting events into an intermediate representation with RGB-like appearances to minimize the domain gap. Then, reusing the prior knowledge in the pre-trained image encoders and image diffusers. By fine-tuning these models with LoRA to adapt to the new event representation input, the pipeline facilitates alignment and mutual generation between the event modality and others, even with small datasets.

In summary, this thesis addresses two core questions: how to more effectively generate synthetic event stream data, and how to design improved event representations for downstream tasks.

5.2 Event-base Vision and Real-time Applications

Although event cameras offer ultra-high temporal resolution and are particularly well-suited for tracking fast-moving objects, their speed advantage is often constrained in practical applications due to several objective limitations. Event data is typically converted into a dense representation before being processed by downstream deep learning models. However, this preprocessing step can be time-consuming. Moreover, because each event representation must aggregate events over a fixed time window, it effectively imposes a frame rate on the inherently asynchronous event stream. Events that occur at the beginning of a time window must wait until the full duration elapses before being forwarded for processing. Finally, inference using these dense representations also introduces latency, making the overall delay non-negligible.

Similarly, while converting the event stream into a spatio-temporal graph and processing it with graph neural networks (GNNs) [135, 167, 7] avoids the need for intermediate dense representations, the construction of such graphs—particularly through k-nearest neighbors (KNN)—can be computationally expensive, especially when the temporal window or event density is large. One common strategy is to aggressively downsample the event stream to create sparser graphs or to incrementally update the graph and network structure over time [135]. However, the former inevitably leads to significant information loss, while the latter increases the complexity of graph construction and limits compatibility with more advanced GNN architectures. Additionally, graphs built from events are generally less sensitive to precise pixel-level spatial positioning compared to dense event representations that naturally encode 2D visual patterns. As a result, while graph-based methods may perform well on classification tasks, they often fall short in pixel-level regression tasks such as human keypoint detection.

To fully leverage the hardware potential of event cameras, one promising direction is the use of spiking neural networks (SNNs) [165, 54] in combination with dedicated neuromorphic hardware, such as Intel’s Loihi 2 [101]. Some prior work has explored hybrid architectures that integrate convolutional neural networks (CNNs) with SNN elements—for instance, replacing

ReLU activation layers with leaky integrate-and-fire (LIF) neurons [147] while still operating on dense tensors. Although this approach benefits from the unique properties of LIF neurons as alternative activation functions, it does not offer significant speed improvements over traditional CNNs. True acceleration may only be achieved through fully event-driven SNNs that treat each event as an individual spike occurring at a specific location, and that operate natively on neuromorphic hardware. Only in such settings can the intrinsic “high-speed” advantage of event cameras be fully realized.

In addition, recent studies have explored the use of Liquid State Machines [161, 155] and Reservoir Computing [87, 144] to project raw event spike signals into high-dimensional representations. These approaches have demonstrated promising results that are comparable to those of more complex models, while requiring significantly fewer computational resources and smaller network architectures. This represents another potential direction for future development in event-based processing.

5.3 Event-based Vision and Human Computer Interaction

While most vision-based tasks can be accomplished using conventional frame-based RGB cameras, event cameras offer potentially superior solutions in certain specialized human-computer interaction scenarios. One such domain is autonomous driving. Autonomous vehicles must account for numerous edge cases, and maintaining reliable performance across all of them is critical, as any misinterpretation or misjudgment of the environment could lead to accidents or even severe consequences.

For instance, when driving on unlit roads at night, it is essential to detect and avoid pedestrians or animals crossing the street timely. Similarly, during transitions into and out of tunnels, conventional cameras often require time to adapt parameters such as aperture and ISO to changes in lighting—delays that can jeopardize safe driving. These are inherent challenges for standard RGB cameras. Event cameras, on the other hand, feature an exceptionally high dynamic

range, significantly reducing issues related to overexposure or underexposure and preserving semantic information without the need for frequent parameter adjustments. This makes them a more robust choice for tasks like autonomous driving and other safety-critical applications. In the context of my work, the human pose estimation model presented in Chapter 1—designed to operate under dynamic lighting conditions and strong background interference—has potential applications in specialized scenarios such as pose estimation of coal miners in extremely low-light underground environments, or underwater pose recognition of divers.

Additionally, event cameras are known for their low power consumption compared to frame-based systems. This makes them particularly well-suited for energy-sensitive edge devices such as smart glasses and XR headsets. Event cameras can efficiently capture patterns and edge information of moving objects in the vicinity with minimal power usage. These event streams can be converted into grayscale frames for conventional computer vision models or processed directly by event-specific models, offering flexibility and convenience. Furthermore, event cameras can be embedded inside VR headsets to enable low-power eye-tracking, even in dark environments.

Event cameras can also complement traditional frame-based cameras to enhance applications such as high-precision slow-motion video generation and video stabilization. Frame-based cameras offer high spatial resolution and accurate intensity recordings, while event cameras capture all temporal changes between frames at microsecond precision. Their superior dynamic range also allows for effective fusion with RGB data to generate HDR imagery. This hybrid approach provides smartphone manufacturers with new directions for optimizing imaging quality and user experience.

In summary, rather than serving as a direct replacement for conventional frame-based cameras, event cameras act as a valuable complement. Their low power consumption and adaptability to extreme lighting conditions effectively address several limitations of frame-based systems, opening new possibilities across a wide range of interactive and embedded vision applications.

5.4 Generalizability in Event-based Vision

Unlike conventional frame-based cameras, event cameras produce outputs with inherently variable shapes. Within a fixed time interval, if lighting conditions remain stable and there are no objects moving significantly relative to the camera, the event stream may be nearly empty. In contrast, sudden changes in illumination, complex camera motion, or the presence of numerous moving objects can trigger millions of events. This highly dynamic and context-dependent nature of event streams makes it particularly challenging to design a generalizable processing pipeline.

Targeted data generation or augmentation strategies can help improve the generalizability of such pipelines. If an event data simulator can generate sufficiently diverse training samples, models will be more robust under a wide range of real-world conditions. This diversity should span various factors including lighting conditions, object motion, object speed, and camera motion. Of course, having access to a high-quality and diverse real-world dataset is equally essential. Fine-tuning models on such real-world data helps mitigate domain gaps caused by discrepancies between synthetic and real event distributions. For handling out-of-distribution (OOD) scenarios, generating additional targeted samples or assigning higher importance weights to such cases during training can further enhance performance.

In Chapter 1, we utilized a synthetic data generator to produce a large-scale event dataset featuring variations in illumination, human body models, dance motions, camera viewpoints, and moving background objects. This strategy significantly improved the overall performance and robustness of our model.

Another approach is to leverage models that have been pre-trained on large-scale datasets from other modalities. Although large, high-quality datasets for event cameras remain scarce, the traditional computer vision domain offers an abundance of well-established datasets across a wide range of tasks. By constructing appropriate event representations, it becomes possible to fine-tune models originally pre-trained on RGB data using event data, thereby improving generalizability. This strategy was employed in my most recent work in Chapter 4.

5.5 Event Representations

As a modality that encodes spatio-temporal information in a sparse form, events require carefully designed representations that emphasize the aspects most relevant to the target vision task. For example, in object recognition, where temporal dynamics are less critical, retaining excessive temporal information introduces redundancy. In such cases, compact representations like event frames or event counts—which primarily preserve 2D visual structure with minimal channels—are more suitable. In contrast, tasks sensitive to motion, such as tracking and human pose estimation, benefit from temporally rich representations like time surfaces or TORE volumes. For tasks that demand a balance of spatial and temporal cues, such as dense trajectory estimation, more comprehensive representations like event voxel grids or Labits introduced in Chapter 3 offer superior performance. When the goal is to preserve as many intrinsic characteristics of the event stream as possible—such as density, average trigger time, and polarity—the FLER representation introduced in Chapter 4 is the most appropriate.

Moreover, representations that integrate temporal information effectively can help address challenges that traditional image-based approaches often struggle with. For instance, in the presence of dynamic occlusions—caused either by moving objects or self-occlusion of body parts—event representations provide hints about the object’s recent and upcoming motion, which can partially mitigate occlusion issues. For more severe or persistent occlusions, temporal models such as ConvLSTM [137], transformers [152], or spatio-temporal attention mechanisms [50, 51] can further enhance performance by enabling information flow across time steps.

In summary, there is no one-size-fits-all event representation suitable for all tasks. The optimal choice depends on the specific requirements of the task at hand, and selecting the most task-aligned representation is essential for effective performance.

5.6 Areas of Future Work

The Labits event representation proposed in Chapter 3 can be computed for smaller spatiotemporal cubes, each containing sufficient information to estimate the instantaneous speed of the central point. This feature can be leveraged for efficient feature or human keypoint tracking, rather than just dense trajectory estimation.

Also, considering the system proposed in Chapter 4, developing a slow-motion video diffuser would be highly beneficial. Currently, we rely on pre-trained video diffusers for general purposes. However, event cameras provide rich temporal information and are already used in slow-motion video generation. Training a diffusion model-based slow-motion video generator could enable images and events to collaborate more effectively.

Finally, embeddings from different modalities could be divided into two components: a unified embedding and a modality-specific information embedding. For instance, aligning the event modality with images might lead to a loss of temporal and motion information. While the unified embedding is sufficient for generating quality content of any modality, incorporating the modality-specific embedding would enhance the generation of content specific to each modality.

Bibliography

- [1] Manuela Adcock, Floriana Sonder, Alexandra Schättin, Federico Gennaro, and Eling D de Bruin. A usability study of a multicomponent video game-based training for older adults. *European review of aging and physical activity*, 17(1):1–15, 2020.
- [2] Mehibe Akandere and Banu Demir. The effect of dance over depression. *Collegium antropologicum*, 35(3):651–656, 2011.
- [3] Sarah Fdili Alaoui, Frédéric Bevilacqua, Bertha Bermudez Pascual, and Christian Jacquemin. Dance interaction with physical model visuals based on movement qualities. *Int. J. Arts Technol.*, 6:357–387, 2013.
- [4] Sarah Fdili Alaoui, Baptiste Caramiaux, Marcos Serrano, and Frédéric Bevilacqua. Movement qualities as interaction modality. In *DIS '12*, 2012.
- [5] Dimitrios S Alexiadis, Philip Kelly, Petros Daras, Noel E O’Connor, Tamy Boubekeur, and Maher Ben Moussa. Evaluating a dancer’s performance using kinect-based skeleton tracking. In *Proceedings of the 19th ACM international conference on Multimedia*, pages 659–662, 2011.
- [6] Andrea Alfarano, Luca Maiano, Lorenzo Papa, and Irene Amerini. Estimating optical flow: A comprehensive review of the state of the art. *Computer Vision and Image Understanding*, page 104160, 2024.
- [7] Yusra Alkendi, Rana Azzam, Abdulla Ayyad, Sajid Javed, Lakmal Seneviratne, and Yahya Zweiri. Neuromorphic camera denoising using graph neural network-driven transformers. *IEEE Transactions on Neural Networks and Learning Systems*, 35(3):4110–4124, 2022.
- [8] Inigo Alonso and Ana C Murillo. Ev-segnet: Semantic segmentation for event-based cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.
- [9] Arnon Amir, Brian Taba, David J. Berg, Timothy Melano, Jeffrey L. McKinstry, Carmelo di Nolfo, Tapan Kumar Nayak, Alexander Andreopoulos, Guillaume Garreau, Marcela Mendoza, Jeffrey A. Kusnitz, Michael V. DeBole, Steven K. Esser, Tobi Delbrück, Myron

- Flickner, and Dharmendra S. Modha. A low power, fully event-based gesture recognition system. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7388–7397, 2017.
- [10] Stefanie Anna Baby, Bimal Vinod, Chaitanya Chinni, and Kaushik Mitra. Dynamic vision sensors for human activity recognition. In *2017 4th IAPR Asian Conference on Pattern Recognition (ACPR)*. IEEE, nov 2017.
- [11] R Baldwin, Mohammed Almatrafi, Vijayan Asari, and Keigo Hirakawa. Event probability mask (epm) and event denoising convolutional neural network (edncnn) for neuromorphic cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1701–1710, 2020.
- [12] R Baldwin, Ruixu Liu, Mohammed Almatrafi, Vijayan Asari, and Keigo Hirakawa. Time-ordered recent event (tore) volumes for event cameras. *arXiv preprint arXiv:2103.06108*, 2021.
- [13] Patrick Bardow, Andrew J. Davison, and Stefan Leutenegger. Simultaneous optical flow and intensity estimation from an event camera. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 884–892, 2016.
- [14] *Beat Saber*. https://en.wikipedia.org/w/index.php?title=Beat_Saber&oldid=1120839189.
- [15] Ryad Benosman, Charles Clercq, Xavier Lagorce, Sio-Hoi Ieng, and Chiara Bartolozzi. Event-based visual flow. *IEEE transactions on neural networks and learning systems*, 25(2):407–417, 2013.
- [16] Xavier Berthelon, Guillaume Chenegros, Thomas Finateu, Sio-Hoi Ieng, and Ryad B. Benosman. Effects of cooling on the snr and contrast detection of a low-light event-based camera. *IEEE Transactions on Biomedical Circuits and Systems*, 12:1467–1474, 2018.
- [17] Johan Bertrand, Arda Yiğit, and Sylvain Durand. Embedded event-based visual odometry. In *2020 6th International Conference on Event-Based Control, Communication, and Signal Processing (EBCCSP)*, pages 1–8. IEEE, 2020.
- [18] Blender (software). [https://en.wikipedia.org/w/index.php?title=Blender_\(software\)](https://en.wikipedia.org/w/index.php?title=Blender_(software)).
- [19] Eliane Gomes da Silva Borges, Rodrigo Gomes de Souza Vale, Carlos Soares Pernambuco, Samaria Ali Cader, Selma Pedra Chaves Sá, Francisco Miguel Pinto, Isabel Cristina Ribeiro Regazzi, Virginia Maria de Azevedo Oliveira Knupp, and Estélio Henrique Martin Dantas. Effects of dance on the postural balance, cognition and functional autonomy of older adults. *Revista brasileira de enfermagem*, 71:2302–2309, 2018.
- [20] Christian Brandli, Raphael Berner, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck. A

240 × 180 130 db 3 μs latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits*, 49:2333–2341, 2014.

- [21] Vincent Brebion, Julien Moreau, and Franck Davoine. Real-time optical flow for vehicular perception with low-and high-resolution event cameras. *IEEE Transactions on Intelligent Transportation Systems*, 2021.
- [22] Enrico Calabrese, Gemma Taverni, Christopher Awai Easthope, Sophie Skriabine, Federico Corradi, Luca Longinotti, Kynan Eng, and Tobi Delbrück. Dhp19: Dynamic vision sensor 3d human pose dataset. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1695–1704, 2019.
- [23] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.
- [24] Dario Cazzato and Flavio Bono. An application-driven survey on event-based neuromorphic computer vision. *Information*, 15(8):472, 2024.
- [25] Bharatesh Chakravarthi, Aayush Atul Verma, Kostas Daniilidis, Cornelia Fermuller, and Yezhou Yang. Recent event camera innovations: A survey. *arXiv preprint arXiv:2408.13627*, 2024.
- [26] William Chamorro, Joan Solà, and Juan Andrade-Cetto. Event-imu fusion strategies for faster-than-imu estimation throughput. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3975–3982, 2023.
- [27] William Oswaldo Chamorro Hernández, Juan Andrade-Cetto, and Joan Solà Ortega. High-speed event camera tracking. In *Proceedings of the The 31st British Machine Vision Virtual Conference*, pages 1–12, 2020.
- [28] Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 3438–3445, 2020.
- [29] Guang Chen, Hu Cao, Jorg Conradt, Huajin Tang, Florian Rohrbein, and Alois Knoll. Event-based neuromorphic vision for autonomous driving: a paradigm shift for bio-inspired visual sensing and perception. *IEEE Signal Processing Magazine*, 37(4):34–49, 2020.
- [30] Shiyan Chen, Jiyuan Zhang, Yajing Zheng, Tiejun Huang, and Zhaofer Yu. Enhancing motion deblurring in high-speed scenes with spike streams. *Advances in Neural Information Processing Systems*, 36, 2024.

- [31] Tianlang Chen, Chengjie Fang, Xiaohui Shen, Yiheng Zhu, Zhili Chen, and Jiebo Luo. Anatomy-aware 3d human pose estimation with bone-based pose decomposition. *IEEE Transactions on Circuits and Systems for Video Technology*, 32:198–209, 2022.
- [32] Yucheng Chen, Yingli Tian, and Mingyi He. Monocular human pose estimation: A survey of deep learning-based methods. *Comput. Vis. Image Underst.*, 192:102897, 2020.
- [33] Shu-Li Cheng, Huey-Fang Sun, and Mei-Ling Yeh. Effects of an 8-week aerobic dance program on health-related fitness in patients with schizophrenia. *journal of nursing research*, 25(6):429–435, 2017.
- [34] Loïc Cordone, Benoît Miramond, and Sonia Ferrante. Learning from event cameras with sparse spiking convolutional neural networks, 2021.
- [35] *Dance Central*. https://en.wikipedia.org/w/index.php?title=Dance_Central&oldid=1116869192.
- [36] *Dance Collider*. <https://www.dancecollider.com>.
- [37] *Dance Dance Revolution*. https://en.wikipedia.org/w/index.php?title=Dance_Dance_Revolution&oldid=1118129088.
- [38] María del Mar López-Rodríguez, Adelaida María Castro-Sánchez, Manuel Fernández-Martínez, Guillermo A Matarán-Penarrocha, and María Encarnación Rodríguez-Ferrer. Comparación entre biodanza en medio acuático y stretching en la mejora de la calidad de vida y dolor en los pacientes con fibromialgia. *Atención Primaria*, 44(11):641–649, 2012.
- [39] Shi-Wee Deng, Yuhang Li, Shanghang Zhang, and Shi Gu. Temporal efficient training of spiking neural network via gradient re-weighting. *ArXiv*, abs/2202.11946, 2022.
- [40] Ziluo Ding, Rui Zhao, Jiyuan Zhang, Tianxiao Gao, Ruiqin Xiong, Zhaofei Yu, and Tiejun Huang. Spatio-temporal recurrent networks for event-based optical flow estimation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 525–533, 2022.
- [41] Yan-Tang Dong and Tao Zhang. Standard and event cameras fusion for feature tracking. *Proceedings of the 2021 International Conference on Machine Vision and Applications*, 2021.
- [42] Yuda Dong, Zetao Chen, Xin He, Lijun Li, Zichao Shu, Yinong Cao, Junchi Feng, Shijie Liu, Chunlai Li, and Jianyu Wang. Sevar: a stereo event camera dataset for virtual and augmented reality. *Frontiers of Information Technology & Electronic Engineering*, 25(5):755–762, 2024.

- [43] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.
- [44] William Falcon and The PyTorch Lightning team. Pytorch lightning. <https://www.pytorchlightning.ai>, 3 2019. The lightweight PyTorch wrapper for high-performance AI research. Scale your models, not the boilerplate.
- [45] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(1):154–180, 2020.
- [46] Lihao Ge, Hui Liang, Junsong Yuan, and Daniel Thalmann. Robust 3d hand pose estimation in single depth images: From single-view cnn to multi-view cnns. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3593–3601, 2016.
- [47] Daniel Gehrig, Mathias Gehrig, Javier Hidalgo-Carrió, and Davide Scaramuzza. Video to events: Recycling video datasets for event cameras. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, June 2020.
- [48] Daniel Gehrig, Michelle Rüegg, Mathias Gehrig, Javier Hidalgo-Carrió, and Davide Scaramuzza. Combining events and frames using recurrent asynchronous multimodal networks for monocular depth prediction. *IEEE Robotics and Automation Letters*, 6(2):2822–2829, 2021.
- [49] Mathias Gehrig, Willem Aarents, Daniel Gehrig, and Davide Scaramuzza. Dsec: A stereo event camera dataset for driving scenarios. *IEEE Robotics and Automation Letters*, 6:4947–4954, 2021.
- [50] Mathias Gehrig, Mario Millhäusler, Daniel Gehrig, and Davide Scaramuzza. E-raft: Dense optical flow from event cameras, 2021.
- [51] Mathias Gehrig, Manasi Muglikar, and Davide Scaramuzza. Dense continuous-time optical flow from event cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
- [52] Mathias Gehrig and Davide Scaramuzza. Recurrent vision transformers for object detection with event cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13884–13893, 2023.
- [53] Rohan Ghosh, Anupam Kumar Gupta, Andrei Nakagawa Silva, Alcimar Barbosa Soares, and Nitish V. Thakor. Spatiotemporal filtering for event-based action recognition. *ArXiv*,

abs/1903.07067, 2019.

- [54] Samanwoy Ghosh-Dastidar and Hojjat Adeli. Spiking neural networks. *International journal of neural systems*, 19(04):295–308, 2009.
- [55] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15180–15190, 2023.
- [56] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.
- [57] Priya Goyal, Quentin Duval, Isaac Seessel, Mathilde Caron, Ishan Misra, Levent Sagun, Armand Joulin, and Piotr Bojanowski. Vision models are more robust and fair when pretrained on uncured images without supervision. *ArXiv*, abs/2202.08360, 2022.
- [58] Hiroko Hashimoto, Shinichi Takabatake, Hideki Miyaguchi, Hajime Nakanishi, and Yasuo Naitou. Effects of dance on motor functions, cognitive functions, and mental symptoms of parkinson’s disease: a quasi-randomized pilot trial. *Complementary therapies in medicine*, 23(2):210–219, 2015.
- [59] Mohamed Hassan, Vasileios Choutas, Dimitrios Tzionas, and Michael J. Black. Resolving 3d human pose ambiguities with 3d scene constraints. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2282–2292, 2019.
- [60] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [61] Weihua He, Kaichao You, Zhendong Qiao, Xu Jia, Ziyang Zhang, Wenhui Wang, Huchuan Lu, Yaoyuan Wang, and Jianxing Liao. Timereplayer: Unlocking the potential of event cameras for video interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17804–17813, 2022.
- [62] Yunze He, Baoyuan Deng, Hongjin Wang, Liang Cheng, Ke Zhou, Siyuan Cai, and Francesco Ciampa. Infrared machine vision and infrared thermography with deep learning: A review. *Infrared physics & technology*, 116:103754, 2021.
- [63] Javier Hidalgo-Carri6, Daniel Gehrig, and Davide Scaramuzza. Learning monocular dense depth from events. In *2020 International Conference on 3D Vision (3DV)*, pages 534–542. IEEE, 2020.

- [64] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- [65] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022.
- [66] Stacy Hsueh, Sarah Fdili Alaoui, and Wendy E Mackay. Understanding kinaesthetic creativity in dance. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2019.
- [67] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.
- [68] Liwen Hu, Rui Zhao, Ziluo Ding, Lei Ma, Boxin Shi, Ruiqin Xiong, and Tiejun Huang. Optical flow estimation for spiking camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17844–17853, 2022.
- [69] Y Hu, S C Liu, and T Delbruck. v2e: From video frames to realistic DVS events. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2021.
- [70] Yuhuang Hu, Hongjie Liu, Michael Pfeiffer, and Tobi Delbruck. Dvs benchmark datasets for object tracking, action recognition, and object recognition. *Frontiers in Neuroscience*, 10, 2016.
- [71] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1325–1339, 2013.
- [72] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR, 2021.
- [73] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9000–9008, 2018.
- [74] Shihao Jiang, Dylan Campbell, Yao Lu, Hongdong Li, and Richard Hartley. Learning to estimate hidden motions with global motion aggregation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9772–9781, 2021.

- [75] Yu Jiang, Yuehang Wang, Siqu Li, Yongji Zhang, Minghao Zhao, and Yue Gao. Event-based low-illumination image enhancement. *IEEE Transactions on Multimedia*, pages 1–12, 2023.
- [76] Jianhao Jiao, Huaiyang Huang, Liang Li, Zhijian He, Yilong Zhu, and Ming Liu. Comparing representations in tracking for event camera-based slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1369–1376, 2021.
- [77] Jichao Jiao, Libin Yuan, Weihua Tang, Zhongliang Deng, and Qi Wu. A post-rectification approach of depth images of kinect v2 for 3d reconstruction of indoor scenes. *ISPRS International Journal of Geo-Information*, 6(11), 2017.
- [78] Damien Joubert, Alexandre Marcireau, Nic Ralph, Andrew Jolley, André van Schaik, and Gregory Cohen. Event camera simulator improvements via characterized parameters. *Frontiers in Neuroscience*, page 910, 2021.
- [79] Maged N Kamel Boulos. Xbox 360 kinect exergames for health. *Games for Health: Research, Development, and Clinical Applications*, 1(5):326–330, 2012.
- [80] Muhammad Jaleed Khan, Hamid Saeed Khan, Adeel Yousaf, Khurram Khurshid, and Asad Abbas. Modern trends in hyperspectral image analysis: A review. *Ieee Access*, 6:14118–14129, 2018.
- [81] Junho Kim, Jaehyeok Bae, Gangin Park, Dongsu Zhang, and Young Min Kim. N-imagenet: Towards robust, fine-grained object recognition with event cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2146–2156, 2021.
- [82] Taewoo Kim, Hoonhee Cho, and Kuk-Jin Yoon. Frequency-aware event-based video deblurring for real-world motion blur. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24966–24976, 2024.
- [83] Alexandros Kitsikidis, Kosmas Dimitropoulos, Stella Douka, and Nikos Grammalidis. Dance analysis using multiple kinect sensors. In *2014 international conference on computer vision theory and applications (VISAPP)*, volume 2, pages 789–795. IEEE, 2014.
- [84] Anne D Kloos, Nora E Fritz, Sandra K Kostyk, Gregory S Young, and Deb A Kegelmeyer. Video game play (dance dance revolution) as a potential exercise therapy in huntington’s disease: a controlled clinical trial. *Clinical rehabilitation*, 27(11):972–982, 2013.
- [85] Adarsh Kumar Kosta and Kaushik Roy. Adaptive-spikenet: event-based optical flow estimation using spiking neural networks with learnable neuronal dynamics. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6021–6027.

IEEE, 2023.

- [86] Xavier Lagorce, G. Orchard, Francesco Galluppi, Bertram E. Shi, and Ryad B. Benosman. Hots: A hierarchy of event-based time-surfaces for pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1346–1359, 2017.
- [87] Richard Lau, Anthony Tylan-Tyler, Lihan Yao, Rey de Castro Roberto, Robert Taylor, and Isaiah Jones. Temporal-spatial processing of event camera data via delay-loop reservoir neural network. *arXiv preprint arXiv:2403.17013*, 2024.
- [88] Chankyu Lee, Adarsh Kumar Kosta, Alex Zihao Zhu, Kenneth Chaney, Kostas Daniilidis, and Kaushik Roy. Spike-flownet: event-based optical flow estimation with energy-efficient hybrid neural networks. In *European Conference on Computer Vision*, pages 366–382. Springer, 2020.
- [89] Jun Haeng Lee, Tobi Delbruck, and Michael Pfeiffer. Training deep spiking neural networks using backpropagation. *Frontiers in neuroscience*, 10:508, 2016.
- [90] Haolong Li and Joerg Stueckler. Tracking 6-dof object motion from events and frames. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14171–14177. IEEE, 2021.
- [91] Jianing Li, Siwei Dong, Zhaofei Yu, Yonghong Tian, and Tiejun Huang. Event-based vision enhanced: A joint detection framework in autonomous driving. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1396–1401. IEEE, 2019.
- [92] Jinghang Li, Bangyan Liao, Xiuyuan Lu, Peidong Liu, Shaojie Shen, and Yi Zhou. Event-aided time-to-collision estimation for autonomous driving. In *European Conference on Computer Vision*, pages 57–73. Springer, 2024.
- [93] Wenhao Li, Hong Liu, Hao Tang, Pichao Wang, and Luc Van Gool. Mhformer: Multi-hypothesis transformer for 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13147–13156, June 2022.
- [94] Yijin Li, Han Zhou, Bangbang Yang, Ye Zhang, Zhaopeng Cui, Hujun Bao, and Guofeng Zhang. Graph-based asynchronous event processing for rapid object recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 934–943, 2021.
- [95] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128×128 120 db 15 *mus* latency asynchronous temporal contrast vision sensor. *IEEE journal of solid-state circuits*, 43(2):566–576, 2008.

- [96] Jian Liu, Wei Sun, Hui Yang, Zhiwen Zeng, Chongpei Liu, Jin Zheng, Xingyu Liu, Hossein Rahmani, Nicu Sebe, and Ajmal Mian. Deep learning-based object pose estimation: A comprehensive survey. *arXiv preprint arXiv:2405.07801*, 2024.
- [97] Lin Liu, Junfeng An, Jianzhuang Liu, Shanxin Yuan, Xiangyu Chen, Wengang Zhou, Houqiang Li, Yanfeng Wang, and Qi Tian. Low-light video enhancement with synthetic event guidance, 2022.
- [98] Min Liu and Tobi Delbruck. Edflow: Event driven optical flow camera with keypoint detection and adaptive block matching. *IEEE Transactions on Circuits and Systems for Video Technology*, pages Epub–ahead, 2022.
- [99] Peigen Liu, Guang Chen, Zhijun Li, Daniel Clarke, Zhengfa Liu, Ruiqi Zhang, and Alois Knoll. Neurodfd: Towards efficient driver face detection with neuromorphic vision sensor. In *2022 International Conference on Advanced Robotics and Mechatronics (ICARM)*, pages 268–273, 2022.
- [100] Yuhan Liu, Yongjian Deng, Hao Chen, and Zhen Yang. Video frame interpolation via direct synthesis with the event-based reference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8477–8487, 2024.
- [101] Loihi 2 - Intel - WikiChip. https://en.wikichip.org/wiki/intel/loihi_2.
- [102] María Mar López-Rodríguez, Manuel Fernández-Martínez, Guillermo A Matarán-Peñarrocha, María Encarnación Rodríguez-Ferrer, Genoveva Granados Gámez, and Encarnación Aguilar Ferrándiz. Efectividad de la biodanza acuática sobre la calidad del sueño, la ansiedad y otros síntomas en pacientes con fibromialgia. *Medicina Clínica*, 141(11):471–478, 2013.
- [103] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. 7th int. In *Conf. Learn. Represent. ICLR*, 2019.
- [104] Florian Mahlknecht, Daniel Gehrig, Jeremy Nash, Friedrich M. Rockenbauer, Benjamin Morrell, Jeff Delaune, and Davide Scaramuzza. Exploring event camera-based odometry for planetary robots. *IEEE Robotics and Automation Letters*, 7:8651–8658, 2022.
- [105] Alessandro Manilii, Leonardo Lucarelli, Riccardo Rosati, Luca Romeo, Adriano Mancini, and Emanuele Frontoni. 3d human pose estimation based on multi-input multi-output convolutional neural network and event cameras: A proof of concept on the dhp19 dataset. In *Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part I*, pages 14–25. Springer, 2021.
- [106] Ana I Maqueda, Antonio Loquercio, Guillermo Gallego, Narciso García, and Davide Scaramuzza. Event-based vision meets deep learning on steering prediction for self-

- driving cars. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5419–5427, 2018.
- [107] David X Marquez, Robert Wilson, Susan Aguiñaga, Priscilla Vásquez, Louis Fogg, Zhi Yang, JoEllen Wilbur, Susan Hughes, and Charles Spanbauer. Regular latin dancing and health education may improve cognition of late middle-aged and older latinos. *Journal of aging and physical activity*, 25(3):482–489, 2017.
 - [108] Nico Messikommer, Carter Fang, Mathias Gehrig, and Davide Scaramuzza. Data-driven feature tracking for event cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5642–5651, 2023.
 - [109] Damien Michel, Ammar Qammaz, and Antonis A Argyros. Markerless 3d human pose estimation and tracking based on rgbd cameras: an experimental evaluation. In *Proceedings of the 10th International Conference on Pervasive Technologies Related to Assistive Environments*, pages 115–122, 2017.
 - [110] Sherif AS Mohamed, Jawad N Yasin, Mohammad-hashem Haghbayan, Antonio Miele, Jukka Heikkonen, Hannu Tenhunen, and Juha Plosila. Dynamic resource-aware corner detection for bio-inspired vision sensors. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 10465–10472. IEEE, 2021.
 - [111] Brian B Moser, Arundhati S Shanbhag, Federico Raue, Stanislav Frolov, Sebastian Palacio, and Andreas Dengel. Diffusion models, image super-resolution, and everything: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.
 - [112] Elias Mueggler, Henri Rebecq, Guillermo Gallego, Tobi Delbruck, and Davide Scaramuzza. The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and slam. *The International Journal of Robotics Research*, 36(2):142–149, 2017.
 - [113] NuiTrack. NuiTrack Full Body Skeletal Tracking Software. <https://nuitrack.com/>.
 - [114] Joao D Nunes, Marcelo Carvalho, Diogo Carneiro, and Jaime S Cardoso. Spiking neural networks: A survey. *IEEE Access*, 10:60738–60764, 2022.
 - [115] Mohamed Omran, Christoph Lassner, Gerard Pons-Moll, Peter V. Gehler, and Bernt Schiele. Neural body fitting: Unifying deep learning and model based human pose and shape estimation. *2018 International Conference on 3D Vision (3DV)*, pages 484–494, 2018.
 - [116] Liyuan Pan, Miaomiao Liu, and Richard I. Hartley. Single image optical flow estimation with an event camera. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1669–1678, 2020.

- [117] Kara K Patterson, Jennifer S Wong, Thi-Ut Nguyen, and Dina Brooks. A dance program to improve gait and balance in individuals with chronic stroke: a feasibility study. *Topics in Stroke Rehabilitation*, 25(6):410–416, 2018.
- [118] Rosa Pinniger, Rhonda F Brown, Einar B Thorsteinsson, and Patricia McKinley. Argentine tango dance compared to mindfulness meditation and a waiting-list control: A randomised trial for treating depression. *Complementary therapies in medicine*, 20(6):377–384, 2012.
- [119] Christoph Posch, Daniel Matolin, and Rainer Wohlgenannt. An asynchronous time-based image sensor. *2008 IEEE International Symposium on Circuits and Systems*, pages 2130–2133, 2008.
- [120] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.
- [121] Ioannis Rallis, Apostolos Langis, Ioannis Georgoulas, Athanasios Voulodimos, Nikolaos Doulamis, and Anastasios Doulamis. An embodied learning game using kinect and labanotation for analysis and visualization of dance kinesiology. In *2018 10th international conference on virtual worlds and games for serious applications (VS-Games)*, pages 1–8. IEEE, 2018.
- [122] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza. Esim: an open event camera simulator. In *Conference on robot learning*, pages 969–982. PMLR, 2018.
- [123] Henri Rebecq, Timo Horstschaefer, and Davide Scaramuzza. Real-time visual-inertial odometry for event cameras using keyframe-based nonlinear optimization. In *British Machine Vision Conference*, 2017.
- [124] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. High speed and high dynamic range video with an event camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43:1964–1980, 2019.
- [125] Helge Rhodin, Jörg Spörri, Isinsu Katircioglu, Victor Constantin, Frédéric Meyer, Erich Müller, Mathieu Salzmann, and Pascal V. Fua. Learning monocular 3d human pose estimation from multi-view images. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8437–8446, 2018.
- [126] Beanbonyka Rim, Nak-Jun Sung, Jun Ma, Yoo-Joo Choi, and Min Hong. Real-time human pose estimation using rgb-d images and deep learning. *Journal of Internet Computing and Services*, 21(3):113–121, 2020.
- [127] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer.

High-resolution image synthesis with latent diffusion models, 2021.

- [128] Alejandro Romero-Hernandez, Manuel Gonzalez-Riojo, Meriem El Yamri, and Borja Manero. The effectiveness of a video game as an educational tool in incrementing interest in dance among younger generations. *Educational Technology & Society*, 25(4):56–69, 2022.
- [129] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [130] Marco R  th and Kai Kaspar. Exergames in formal school teaching: A pre-post longitudinal field study on the effects of a dance game on motor learning, physical enjoyment, and learning motivation. *Entertainment Computing*, 35:100372, 2020.
- [131] Mar  a Fernanda Hincapi   S  nchez, Edward David Buritic   Mar  n, and Leidy Tatiana Ordo  ez Mora. Characterization of dance-based protocols used in rehabilitation-a systematic review. *Heliyon*, page e08573, 2021.
- [132] Yulia Sandamirskaya, Mohsen Kaboli, Jorg Conradt, and Tansu Celikel. Neuromorphic computing hardware and neural architectures for robotics. *Science Robotics*, 7(67):eabl8419, 2022.
- [133] Gianluca Scarpellini, Pietro Morerio, and Alessio Del Bue. Lifting monocular events to 3d human poses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1358–1368, 2021.
- [134] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20:61–80, 2009.
- [135] Simon Schaefer, Daniel Gehrig, and Davide Scaramuzza. Aegnn: Asynchronous event-based graph neural networks, 2022.
- [136] Hochang Seok and Jongwoo Lim. Robust feature tracking in dvs event stream using b  zier mapping. *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1647–1656, 2020.
- [137] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015.
- [138] Hui Shuai, Lele Wu, and Qingshan Liu. Adaptive multi-view and temporal fusing transformer for 3d human pose estimation. *IEEE Transactions on Pattern Analysis and*

- [139] Amos Sironi, Manuele Brambilla, Nicolas Bourdis, Xavier Lagorce, and Ryad Benosman. Hats: Histograms of averaged time surfaces for robust event-based object classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1731–1740, 2018.
- [140] Vinkle Srivastav, Thibaut Issenhuth, Abdolrahim Kadkhodamohammadi, Michel de Mathelin, Afshin Gangi, and Nicolas Padoy. Mvor: A multi-view rgb-d operating room dataset for 2d and 3d human pose estimation. *arXiv preprint arXiv:1808.08180*, 2018.
- [141] Daobo Sun and Haibo Ji. Event-based object detection using graph neural networks. In *2023 IEEE 12th Data Driven Control and Learning Systems Conference (DDCLS)*, pages 1895–1900, 2023.
- [142] Lei Sun, Christos Sakaridis, Jingyun Liang, Peng Sun, Jiezhong Cao, Kai Zhang, Qi Jiang, Kaiwei Wang, and Luc Van Gool. Event-based frame interpolation with ad-hoc deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18043–18052, 2023.
- [143] Synthridersvr. Synth Riders - A Freestyle-Dance VR Rhythm Game. <https://synthridersvr.com/>.
- [144] Hongwei Tan and Sebastiaan van Dijken. Dynamic machine vision with retinomorph photomemristor-reservoir computing. *Nature Communications*, 14(1):2169, 2023.
- [145] Amirhossein Tavanaei, Masoud Ghodrati, Saeed Reza Kheradpisheh, Timothée Masquelier, and A. Maida. Deep learning in spiking neural networks. *Neural networks : the official journal of the International Neural Network Society*, 111:47–63, 2018.
- [146] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020.
- [147] Corinne Teeter, Ramakrishnan Iyer, Vilas Menon, Nathan Gouwens, David Feng, Jim Berg, Aaron Szafer, Nicholas Cain, Hongkui Zeng, Michael Hawrylycz, et al. Generalized leaky integrate-and-fire models classify multiple neuron types. *Nature communications*, 9(1):709, 2018.
- [148] Lavinia Teixeira-Machado, Ricardo Mario Arida, and Jair de Jesus Mari. Dance for neuroplasticity: A descriptive systematic review. *Neuroscience & Biobehavioral Reviews*, 96:232–240, 2019.
- [149] Sangay Tenzin, Alexander Rassau, and Douglas Chai. Application of event cameras and

neuromorphic computing to vslam: A survey. *Biomimetics*, 9(7):444, 2024.

- [150] Hanyue Tu, Chunyu Wang, and Wenjun Zeng. Voxelpose: Towards multi-camera 3d human pose estimation in wild environment. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I* 16, pages 197–212. Springer, 2020.
- [151] Stepan Tulyakov, Alfredo Bochicchio, Daniel Gehrig, Stamatios Georgoulis, Yuanyou Li, and Davide Scaramuzza. Time lens++: Event-based frame interpolation with parametric non-linear flow and multi-scale fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17755–17764, 2022.
- [152] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [153] Zengyu Wan, Yang Wang, Zhai Wei, Ganchao Tan, Yang Cao, and Zheng-Jun Zha. Event-based optical flow via transforming into motion-dependent view. *IEEE Transactions on Image Processing*, 2024.
- [154] Bishan Wang, Jingwei He, Lei Yu, Guisong Xia, and Wen Yang. Event enhanced high-quality image recovery. In *European Conference on Computer Vision*, 2020.
- [155] Lei Wang, Sha-Sha Guo, Lian-Hua Qu, Shuo Tian, and Wei-Xia Xu. M-lsm: An improved multi-liquid state machine for event-based vision recognition. *Journal of Computer Science and Technology*, 38(6):1288–1299, 2023.
- [156] Pei Wang, Jiumei He, Qingsen Yan, Yu Zhu, Jinqiu Sun, and Yanning Zhang. Diffevent: Event residual diffusion for image deblurring. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3450–3454. IEEE, 2024.
- [157] Yanxiang Wang, Bowen Du, Yiran Shen, Kai Wu, Guangrong Zhao, Jianguo Sun, and Hongkai Wen. Ev-gait: Event-based robust gait recognition using dynamic vision sensors. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6351–6360, 2019.
- [158] Wikipedia contributors. *Just Dance* (video game series). [https://en.wikipedia.org/w/index.php?title=Just_Dance_\(video_game_series\)&oldid=1121254502](https://en.wikipedia.org/w/index.php?title=Just_Dance_(video_game_series)&oldid=1121254502).
- [159] Yilun Wu, Federico Paredes-Vallés, and Guido CHE de Croon. Lightweight event-based optical flow estimation via iterative deblurring. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14708–14715. IEEE, 2024.

- [160] Ziyi Wu, Xudong Liu, and Igor Gilitschenski. Eventclip: Adapting clip for event-based object recognition. *arXiv preprint arXiv:2306.06354*, 2023.
- [161] Xun Xiao, Lei Wang, Xiaofan Chen, Lianhua Qu, Shasha Guo, Yao Wang, and Ziyang Kang. Dynamic vision sensor based gesture recognition using liquid state machine. In *International Conference on Artificial Neural Networks*, pages 618–629. Springer, 2022.
- [162] Fang Xu, Lei Yu, Bishan Wang, Wen Yang, Gui-Song Xia, Xu Jia, Zhendong Qiao, and Jianzhuang Liu. Motion deblurring with real events. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2583–2592, 2021.
- [163] Hantong Xu, Jiamin Xu, and Weiwei Xu. Survey of 3d modeling using depth cameras. *Virtual Reality & Intelligent Hardware*, 1(5):483–499, 2019.
- [164] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. Vitpose: Simple vision transformer baselines for human pose estimation, 2022.
- [165] Kashu Yamazaki, Viet-Khoa Vo-Ho, Darshan Bulsara, and Ngan Le. Spiking neural networks and their applications: A review. *Brain sciences*, 12(7):863, 2022.
- [166] Yixin Yang, Jinxiu Liang, Bohan Yu, Yan Chen, Jimmy S Ren, and Boxin Shi. Latency correction for event-guided deblurring and frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24977–24986, 2024.
- [167] Yufeng Yang, Adrian Kneip, and Charlotte Frenkel. Evgnn: An event-driven graph neural network accelerator for edge vision. *IEEE Transactions on Circuits and Systems for Artificial Intelligence*, 2024.
- [168] Yaozu Ye, Hao Shi, Kailun Yang, Ze Wang, Xiaoting Yin, Yining Lin, Mao Liu, Yaonan Wang, and Kaiwei Wang. Towards anytime optical flow estimation with event cameras. *arXiv preprint arXiv:2307.05033*, 2023.
- [169] Özgün Yılmaz, Camille Simon-Chane, and Aymeric Histace. Evaluation of event-based corner detectors. *Journal of Imaging*, 7(2):25, 2021.
- [170] Hye-jeong Yun, Kwang-il Kim, Jeong-hun Lee, and Hae-Yeoun Lee. Development of experience dance game using kinect motion capture. *KIPS transactions on software and data engineering*, 3(1):49–56, 2014.
- [171] Jinlu Zhang, Yujin Chen, and Zhigang Tu. Uncertainty-aware 3d human pose estimation from monocular video. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 5102–5113, 2022.

- [172] Jinlu Zhang, Zhigang Tu, Jianyu Yang, Yujin Chen, and Junsong Yuan. Mixste: Seq2seq mixed spatio-temporal encoder for 3d human pose estimation in video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13232–13242, 2022.
- [173] Zhongyang Zhang, Kaidong Chai, Haowen Yu, Ramzi Majaj, Francesca Walsh, Edward Wang, Upal Mahbub, Hava Siegelmann, Donghyun Kim, and Tauhidur Rahman. Neuromorphic high-frequency 3d dancing pose estimation in dynamic environment. *Neurocomputing*, 547:126388, 2023.
- [174] Zhongyang Zhang, Shuyang Cui, Kaidong Chai, Haowen Yu, Subhasis Dasgupta, Upal Mahbub, and Tauhidur Rahman. V2ce: Video to continuous events simulator. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12455–12461. IEEE, 2024.
- [175] Zhongyang Zhang, Jiacheng Qiu, Shuyang Cui, Yijun Luo, and Tauhidur Rahman. Labits: Layered bidirectional time surfaces representation for event camera-based continuous dense trajectory estimation, 2024.
- [176] Xia Zhao, Limin Wang, Yufei Zhang, Xuming Han, Muhammet Deveci, and Milan Parmar. A review of convolutional neural networks in computer vision. *Artificial Intelligence Review*, 57(4):99, 2024.
- [177] Jiazhou Zhou, Xu Zheng, Yuanhuiyi Lyu, and Lin Wang. Eventbind: Learning a unified representation to bind them all for event-based open-world understanding. In *European Conference on Computer Vision*, pages 477–494. Springer, 2024.
- [178] Alex Zihao Zhu, Nikolay Atanasov, and Kostas Daniilidis. Event-based feature tracking with probabilistic data association. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4465–4470. IEEE, 2017.
- [179] Alex Zihao Zhu, Dinesh Thakur, Tolga Özaslan, Bernd Pfrommer, Vijay Kumar, and Kostas Daniilidis. The multivehicle stereo event camera dataset: An event camera dataset for 3d perception. *IEEE Robotics and Automation Letters*, 3(3):2032–2039, 2018.
- [180] Alex Zihao Zhu, Ziyun Wang, Kaung Khant, and Kostas Daniilidis. Eventgan: Leveraging large scale image datasets for event cameras, 2019.
- [181] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Ev-flownet: Self-supervised optical flow estimation for event-based cameras. *arXiv preprint arXiv:1802.06898*, 2018.
- [182] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised event-based learning of optical flow, depth, and egomotion. In *Proceedings of the*

- IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 989–997, 2019.
- [183] Lin Zhu, Xiao Wang, Yi Chang, Jianing Li, Tiejun Huang, and Yonghong Tian. Event-based video reconstruction via potential-assisted spiking neural network, 2022.
 - [184] Lin Zhu, Yunlong Zheng, Yijun Zhang, Xiao Wang, Lizhi Wang, and Hua Huang. Temporal residual guided diffusion framework for event-driven video reconstruction. In *European Conference on Computer Vision*, pages 411–427. Springer, 2024.
 - [185] Yi Zhu, Han Wu, Ming Qi, Sheng Wang, Qin Zhang, Li Zhou, Shiyan Wang, Wei Wang, Ting Wu, Ming Xiao, et al. Effects of a specially designed aerobic dance routine on mild cognitive impairment. *Clinical interventions in aging*, 13:1691, 2018.
 - [186] Christian Zimmermann, Tim Welschhold, Christian Dornhege, Wolfram Burgard, and Thomas Brox. 3d human pose estimation in rgb-d images for robotic task learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1986–1992. IEEE, 2018.
 - [187] Shihao Zou, Chuan Guo, Xinxin Zuo, Sen Wang, Pengyu Wang, Xiaoqin Hu, Shoushun Chen, Minglun Gong, and Li Cheng. Eventhpe: Event-based 3d human pose and shape estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10996–11005, 2021.
 - [188] Yunhao Zou, Ying Fu, Tsuyoshi Takatani, and Yinqiang Zheng. Eventhdr: From event to high-speed hdr videos and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.