Turnitin Originality Report

Guan Weipeng PhD thesis main (2025.05.09).pdf by Weipeng GUAN

From Engineering Thesis 2024-25 (Engineering Thesis 2024-25)

Processed on 10-May-2025 09:48 HKT
ID: 2656419329
Word Count: 76610

Similarity Index	Similarity by Source	
48%	Internet Sources:	41%
	Publications:	32%
	Student Papers:	8%

sources:

1 17% match (Internet from 21-Dec-2023)
http://arxiv.org/pdf/2312.11911.pdf

2 7% match (Weipeng Guan, Peiyu Chen, Yuhan Xie, Peng Lu. "PL-EVIO: Robust Monocular Event-Based Visual Inertial Odometry With Point and Line Features", IEEE Transactions on Automation Science and Engineering, 2024)
Weipeng Guan, Peiyu Chen, Yuhan Xie, Peng Lu. "PL-EVIO: Robust Monocular Event-Based Visual Inertial Odometry With Point and Line Features", IEEE Transactions on Automation Science and Engineering, 2024

3 6% match (Weipeng Guan, Peng Lu. "Monocular Event Visual Inertial Odometry based on Event-corner using Sliding Windows Graph-based Optimization", 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2022)
Weipeng Guan, Peng Lu. "Monocular Event Visual Inertial Odometry based on Event-corner using Sliding Windows Graph-based Optimization", 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2022

4 3% match ()
Guan, Weipeng, Chen, Peiyu, Xie, Yuhan, Lu, Peng. "PL-EVIO: Robust Monocular Event-based Visual Inertial Odometry with Point and Line Features", 2023

5 2% match (Internet from 20-Apr-2023)
http://export.arxiv.org/pdf/2209.12160

6 1% match (Internet from 08-Nov-2023)
https://arxiv.org/pdf/2311.02327.pdf

7 1% match (Internet from 22-Nov-2023)
https://export.arxiv.org/pdf/2212.13184

8 1% match (student papers from 02-Nov-2023)
Submitted to University of Hong Kong on 2023-11-02

9 1% match ()
Rebecq, Henri. "Event cameras: from SLAM to high speed video", 2020

10 1% match (Internet from 16-Apr-2024)
https://papers.nips.cc/paper_files/paper/2023/file/7ac484b0f1a1719ad5be9aa8c8455fbb-Paper-Conference.pdf

11 < 1% match (Internet from 02-Apr-2025)
https://arxiv.org/html/2503.22963v1

12 < 1% match (Internet from 02-Apr-2025)
https://arxiv.org/html/2503.22943v1

13 < 1% match (Internet from 13-Jul-2024)
http://arxiv.org/pdf/2407.07816

14 < 1% match (Internet from 19-Dec-2023)

http://arxiv.org/pdf/2312.09800.pdf

15 < 1% match (Internet from 12-Dec-2024)
https://arxiv.org/html/2412.07080v1

16 < 1% match (Internet from 03-Apr-2025)
https://arxiv.org/html/2504.00139v1

17 < 1% match ()
Chen, Peiyu, Guan, Weipeng, Lu, Peng. "ESVIO: Event-based Stereo Visual Inertial Odometry", 2023

18 < 1% match (Internet from 13-Nov-2022)
https://arxiv.org/pdf/2001.02319.pdf

19 < 1% match ()
Huang, Kunping, Zhang, Sen, Zhang, Jing, Tao, Dacheng. "Event-based Simultaneous Localization and Mapping: A Comprehensive Survey", 2023

20 < 1% match (Internet from 03-May-2023)
http://export.arxiv.org/pdf/2304.09793

21 < 1% match (Internet from 23-Nov-2023)
https://export.arxiv.org/pdf/2302.08890

22 < 1% match (student papers from 09-Jul-2023)
Submitted to University of Hong Kong on 2023-07-09

23 < 1% match (student papers from 08-Jul-2023)
Submitted to University of Hong Kong on 2023-07-08

24 < 1% match (student papers from 25-Apr-2024)
Submitted to University of Hong Kong on 2024-04-25

25 < 1% match (student papers from 20-Aug-2024)
Submitted to University of Hong Kong on 2024-08-20

26 < 1% match (student papers from 19-Jul-2024)
Submitted to University of Hong Kong on 2024-07-19

27 < 1% match (student papers from 13-Nov-2023)
Submitted to University of Hong Kong on 2023-11-13

28 < 1% match (student papers from 20-Jun-2024)
Submitted to University of Hong Kong on 2024-06-20

29 < 1% match (student papers from 29-Aug-2024)
Submitted to University of Hong Kong on 2024-08-29

30 < 1% match (student papers from 24-Aug-2023)
Submitted to University of Hong Kong on 2023-08-24

31 < 1% match ()
Müggler, Elias. "Event-based Vision for High-Speed Robotics", 2017

32 < 1% match (Internet from 26-Dec-2022)
https://www.research-collection.ethz.ch/bitstream/handle/20.500.11850/541700/5/ignacio_alzugaray_doctoral_thesis.pdf

33 < 1% match (Internet from 29-Apr-2023)
https://www.research-collection.ethz.ch/bitstream/handle/20.500.11850/514829/phd_thesis_ftschopp.pdf?isAllowed=y&sequence=1

34 < 1% match (Internet from 16-Mar-2023)
https://deepai.org/publication/pl-evio-robust-monocular-event-based-visual-inertial-odometry-with-point-and-line-features

35 < 1% match (student papers from 15-May-2023)
Submitted to University of Sydney on 2023-05-15

36 < 1% match (student papers from 05-May-2023)
Submitted to University of Sydney on 2023-05-05

37 < 1% match (student papers from 29-Apr-2024)
Submitted to University of Sydney on 2024-04-29

38 < 1% match (Mohsen Shahraki, Ahmed Elamin, Ahmed El-Rabbany. "Event-Based Visual Simultaneous Localization and Mapping (EVSLAM) Techniques: State of the Art and Future Directions", Journal of Sensor and Actuator Networks, 2025)
Mohsen Shahraki, Ahmed Elamin, Ahmed El-Rabbany. "Event-Based Visual Simultaneous Localization and Mapping (EVSLAM) Techniques: State of the Art and Future Directions", Journal of Sensor and Actuator Networks, 2025

39 < 1% match (Internet from 14-Dec-2022)
https://digital.library.adelaide.edu.au/dspace/bitstream/2440/136406/1/Liu2022_PhD.pdf

40 < 1% match (Internet from 03-Apr-2023)
https://ceur-ws.org/Vol-3248/paper12.pdf

41 < 1% match (Wanting Xu, Xin Peng, Laurent Kneip. "Tight Fusion of Events and Inertial Measurements for Direct Velocity Estimation", IEEE Transactions on Robotics, 2023)
Wanting Xu, Xin Peng, Laurent Kneip. "Tight Fusion of Events and Inertial Measurements for Direct Velocity Estimation", IEEE Transactions on Robotics, 2023

42 < 1% match (Lipson, Lahav. "Fast and Robust 3D Reconstruction.", Princeton University)
Lipson, Lahav. "Fast and Robust 3D Reconstruction.", Princeton University

43 < 1% match (Internet from 14-Dec-2024)
https://d-nb.info/1350826243/34

44 < 1% match (Internet from 22-Feb-2025)
https://rpg.ifi.uzh.ch/docs/IROS24_Pellerito.pdf

45 < 1% match (Internet from 19-Nov-2023)
https://pure.rug.nl/ws/files/830806072/Complete_thesis.pdf

46 < 1% match (Peiyu Chen, Weipeng Guan, Peng Lu. "ESVIO: Event-Based Stereo Visual Inertial Odometry", IEEE Robotics and Automation Letters, 2023)
Peiyu Chen, Weipeng Guan, Peng Lu. "ESVIO: Event-Based Stereo Visual Inertial Odometry", IEEE Robotics and Automation Letters, 2023

47 < 1% match (Internet from 21-Jan-2025)
https://upcommons.upc.edu/bitstream/handle/2117/422144/Event-based_egomotion_estimation_Sergi_Sanchez_Orvay_TFG.pdf?isAllowed=y&sequence=3

48 < 1% match (Yi-Fan Zuo, Wanting Xu, Xia Wang, Yifu Wang, Laurent Kneip. "Cross-Modal Semi-Dense 6-DoF Tracking of an Event Camera in Challenging Conditions", IEEE Transactions on Robotics, 2024)

Yi-Fan Zuo, Wanting Xu, Xia Wang, Yifu Wang, Laurent Kneip. "Cross-Modal Semi-Dense 6-DoF Tracking of an Event Camera in Challenging Conditions", IEEE Transactions on Robotics, 2024

49 < 1% match (Yuzhen Wu, Lingxue Wang, Lian Zhang, Mingkun Chen, Wenqu Zhao, Dezhi Zheng, Yi Cai. "Monocular thermal SLAM with neural radiance fields for 3D scene reconstruction", Neurocomputing, 2025)

Yuzhen Wu, Lingxue Wang, Lian Zhang, Mingkun Chen, Wenqu Zhao, Dezhi Zheng, Yi Cai. "Monocular thermal SLAM with neural radiance fields for 3D scene reconstruction", Neurocomputing, 2025

50 < 1% match (student papers from 02-Sep-2024)

Submitted to University of Oklahoma on 2024-09-02

51 < 1% match (Internet from 16-Mar-2024)

https://iris.polito.it/retrieve/1b67f5b0-c083-43fb-ac95-9cc9827596fc/thesis_mirco_planamente.pdf

52 < 1% match (student papers from 07-Feb-2025)

Submitted to Rochester Institute of Technology on 2025-02-07

53 < 1% match (Teed, Zachary. "Optimization Inspired Neural Networks for Multiview 3D Reconstruction", Princeton University, 2022)

Teed, Zachary. "Optimization Inspired Neural Networks for Multiview 3D Reconstruction", Princeton University, 2022

54 < 1% match (Lee, Connor Tinghan. "Learning-Based Perception for Robotics in Suboptimal Data Landscapes", California Institute of Technology, 2024)

Lee, Connor Tinghan. "Learning-Based Perception for Robotics in Suboptimal Data Landscapes", California Institute of Technology, 2024

55 < 1% match (Sarıkamış, Furkan Aykut. "Utilization of Gaussian Splatting in Visual Slam", Middle East Technical University (Turkey))

Sarıkamış, Furkan Aykut. "Utilization of Gaussian Splatting in Visual Slam", Middle East Technical University (Turkey)

56 < 1% match (Queirós Arcanjo, Bruno Rafael. "Efficient Visual Place Recognition in Changing Environments for Resource-Constrained Platforms", University of Essex (United Kingdom))

Queirós Arcanjo, Bruno Rafael. "Efficient Visual Place Recognition in Changing Environments for Resource-Constrained Platforms", University of Essex (United Kingdom)

57 < 1% match (Guillermo Gallego, Tobi Delbruck, Garrick Michael Orchard, Chiara Bartolozzi et al. "Event-based Vision: A Survey", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020)

Guillermo Gallego, Tobi Delbruck, Garrick Michael Orchard, Chiara Bartolozzi et al. "Event-based Vision: A Survey", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020

58 < 1% match (student papers from 22-Sep-2022)

Submitted to University of Birmingham on 2022-09-22

59 < 1% match (Zhu, Shengjie. "Structure and Motion From Depth and Correspondence Models", Michigan State University)

Zhu, Shengjie. "Structure and Motion From Depth and Correspondence Models", Michigan State University

60 < 1% match (Internet from 26-Mar-2024)

https://www.biblio.univ-evry.fr/theses/2023/2023UPAST119.pdf

61 < 1% match (Ninghui Xu, Lihui Wang, Zhiting Yao, Takayuki Okatani. "METS: Motion-Encoded Time-Surface for Event-Based High-Speed Pose Tracking", International Journal of Computer Vision, 2025)

Ninghui Xu, Lihui Wang, Zhiting Yao, Takayuki Okatani. "METS: Motion-Encoded Time-Surface for Event-Based High-Speed Pose Tracking", International Journal of Computer Vision, 2025

< 1% match ("Computer Vision – ECCV 2018", Springer Science and Business Media LLC, 2018)

62 "Computer Vision – ECCV 2018", Springer Science and Business Media LLC, 2018

63 < 1% match (Jiafeng Huang, Shengjie Zhao, Tianjun Zhang, Lin Zhang. "MC-VEO: A Visual-Event Odometry With Accurate 6-DoF Motion Compensation", IEEE Transactions on Intelligent Vehicles, 2024)

Jiafeng Huang, Shengjie Zhao, Tianjun Zhang, Lin Zhang. "MC-VEO: A Visual-Event Odometry With Accurate 6-DoF Motion Compensation", IEEE Transactions on Intelligent Vehicles, 2024

64 < 1% match (student papers from 19-Sep-2024)

Submitted to UCL on 2024-09-19

65 < 1% match (Internet from 18-Feb-2025)

https://repository.essex.ac.uk/31706/1/University_of_Essex_PhD_THESIS_Tuo.pdf

66 < 1% match (Robert Guamán-Rivera, Jose Delpiano, Rodrigo Verschae. "Event-based optical flow: Method categorisation and review of techniques that leverage deep learning", Neurocomputing, 2025)

Robert Guamán-Rivera, Jose Delpiano, Rodrigo Verschae. "Event-based optical flow: Method categorisation and review of techniques that leverage deep learning", Neurocomputing, 2025

67 < 1% match (Zhe Liu, Dianxi Shi, Ruihao Li, Shaowu Yang. "ESVIO: Event-Based Stereo Visual-Inertial Odometry", Sensors, 2023)

Zhe Liu, Dianxi Shi, Ruihao Li, Shaowu Yang. "ESVIO: Event-Based Stereo Visual-Inertial Odometry", Sensors, 2023

68 < 1% match (Internet from 05-Sep-2024)

https://digibug.ugr.es/bitstream/handle/10481/93842/3656469.pdf?isAllowed=y&sequence=1

69 < 1% match (Internet from 05-May-2025)

https://digibug.ugr.es/bitstream/handle/10481/96383/Neuromorphic_Perception_and_Navigation_for_Mobile_Robots__A_Revie isAllowed=y&sequence=1

70 < 1% match (Nir, Jagatpreet Singh. "Low Contrast Visual Sensing and Inertial Navigation in GPS Denied Environments", Northeastern University, 2024)

Nir, Jagatpreet Singh. "Low Contrast Visual Sensing and Inertial Navigation in GPS Denied Environments", Northeastern University, 2024

71 < 1% match (Internet from 09-May-2025)

https://cslinzhang.github.io/home/icmr2025/yang.pdf

72 < 1% match (Internet from 13-Dec-2024)

https://www.roboticsproceedings.org/rss20/p088.pdf

73 < 1% match (Shifan Zhu, Zhipeng Tang, Michael Yang, Erik Learned-Miller, Donghyun Kim. "Event Camera-Based Visual Odometry for Dynamic Motion Tracking of a Legged Robot Using Adaptive Time Surface", 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2023)

Shifan Zhu, Zhipeng Tang, Michael Yang, Erik Learned-Miller, Donghyun Kim. "Event Camera-Based Visual Odometry for Dynamic Motion Tracking of a Legged Robot Using Adaptive Time Surface", 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2023

74 < 1% match (Internet from 16-Feb-2023)

https://www.politesi.polimi.it/bitstream/10589/187047/1/phd-thesis-cannici.pdf

75 < 1% match (Rebello, Jason Joseph. "Unlocking Dynamic Cameras for Visual Navigation.", University of Toronto (Canada), 2023)

Rebello, Jason Joseph. "Unlocking Dynamic Cameras for Visual Navigation.", University of Toronto (Canada), 2023

76 < 1% match (Zhu, Chenqi. "Inertially Constrained Ruled Surfaces for Visual Odometry", University of Maryland, College Park, 2025)

Zhu, Chenqi. "Inertially Constrained Ruled Surfaces for Visual Odometry", University of Maryland, College Park, 2025

77 < 1% match (Jeremías Gaia, Eugenio Orosco, Francisco Rossomando, Carlos Soria. "Mapping the Landscape of SLAM Research: A Review", IEEE Latin America Transactions, 2023)

Jeremías Gaia, Eugenio Orosco, Francisco Rossomando, Carlos Soria. "Mapping the Landscape of SLAM Research: A Review", IEEE Latin America Transactions, 2023

78 < 1% match ("Computer Vision – ECCV 2016", Springer Science and Business Media LLC, 2016)

"Computer Vision – ECCV 2016", Springer Science and Business Media LLC, 2016

79 < 1% match (Jagannatha Sanket, Nitin. "Active Vision Based Embodied-AI Design for Nano-UAV Autonomy", University of Maryland, College Park, 2021)

Jagannatha Sanket, Nitin. "Active Vision Based Embodied-AI Design for Nano-UAV Autonomy", University of Maryland, College Park, 2021

80 < 1% match (Jiang, Zhongyu. "Towards Robust and Effective Human Pose Estimation and Generation", University of Washington)

Jiang, Zhongyu. "Towards Robust and Effective Human Pose Estimation and Generation", University of Washington

81 < 1% match (Kılıç, Onur Selim. "Utilization of Event Based Cameras for Video Frame Interpolation", Middle East Technical University (Turkey), 2024)

Kılıç, Onur Selim. "Utilization of Event Based Cameras for Video Frame Interpolation", Middle East Technical University (Turkey), 2024

82 < 1% match ("Intelligent Robotics and Applications", Springer Science and Business Media LLC, 2023)

"Intelligent Robotics and Applications", Springer Science and Business Media LLC, 2023

83 < 1% match (Internet from 16-Nov-2024)

https://www.preprints.org/manuscript/202405.1094/v1

84 < 1% match (Benny Dai, Cedric Le Gentil, Teresa Vidal-Calleja. "A Tightly-Coupled Event-Inertial Odometry using Exponential Decay and Linear Preintegrated Measurements", 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2022)

Benny Dai, Cedric Le Gentil, Teresa Vidal-Calleja. "A Tightly-Coupled Event-Inertial Odometry using Exponential Decay and Linear Preintegrated Measurements", 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2022

85 < 1% match (Bo Xu, Xin Li, Jingrong Wang, Chau Yuen, Jiancheng Li. "PVI-DSO: Leveraging Planar Regularities for Direct Sparse Visual-Inertial Odometry", IEEE Sensors Journal, 2023)

Bo Xu, Xin Li, Jingrong Wang, Chau Yuen, Jiancheng Li. "PVI-DSO: Leveraging Planar Regularities for Direct Sparse Visual-Inertial Odometry", IEEE Sensors Journal, 2023

86 < 1% match (Xingdong Sheng, Shijie Mao, Yichao Yan, Xiaokang Yang. "Review on SLAM algorithms for Augmented Reality", Displays, 2024)

Xingdong Sheng, Shijie Mao, Yichao Yan, Xiaokang Yang. "Review on SLAM algorithms for Augmented Reality", Displays, 2024

87 < 1% match ("Computer Vision – ECCV 2022", Springer Science and Business Media LLC, 2022)

"Computer Vision – ECCV 2022", Springer Science and Business Media LLC, 2022

88 < 1% match (student papers from 28-Feb-2025)

Submitted to Imperial College of Science, Technology and Medicine on 2025-02-28

89 < 1% match (student papers from 16-Sep-2022)

Submitted to Imperial College of Science, Technology and Medicine on 2022-09-16

90 < 1% match (Cedric Le Gentil, Ignacio Alzugaray, Teresa Vidal-Calleja. "Continuous-Time Gaussian Process Motion-Compensation for Event-Vision Pattern Tracking with Distance Fields", 2023 IEEE International Conference on Robotics and Automation (ICRA), 2023)

Cedric Le Gentil, Ignacio Alzugaray, Teresa Vidal-Calleja. "Continuous-Time Gaussian Process Motion-Compensation for Event-Vision Pattern Tracking with Distance Fields", 2023 IEEE International Conference on Robotics and Automation (ICRA), 2023

---

91 < 1% match (Internet from 15-Feb-2024)
https://khazna.ku.ac.ae/ws/portalfiles/portal/19097820/file

---

92 < 1% match (Weiqiang Zhao, Hang Sun, Xinyu Zhang, Yijin Xiong. "Visual SLAM Combining Lines and Structural Regularities: Towards Robust Localization", IEEE Transactions on Intelligent Vehicles, 2023)

Weiqiang Zhao, Hang Sun, Xinyu Zhang, Yijin Xiong. "Visual SLAM Combining Lines and Structural Regularities: Towards Robust Localization", IEEE Transactions on Intelligent Vehicles, 2023

---

93 < 1% match ("Advances in Guidance, Navigation and Control", Springer Science and Business Media LLC, 2025)

"Advances in Guidance, Navigation and Control", Springer Science and Business Media LLC, 2025

---

94 < 1% match (Dong, Xingshuai. "Visual Guidance for Unmanned Aerial Vehicles With Deep Learning.", University of New South Wales (Australia))

Dong, Xingshuai. "Visual Guidance for Unmanned Aerial Vehicles With Deep Learning.", University of New South Wales (Australia)

---

95 < 1% match (Internet from 07-Jul-2023)
https://www.arxiv-vanity.com/papers/2212.13184/

---

96 < 1% match (student papers from 04-May-2023)
Submitted to Khalifa University of Science Technology and Research on 2023-05-04

---

97 < 1% match (Zhong, Yuanyi. "Sample-Efficient Learning With Self-Supervision", University of Illinois at Urbana-Champaign)

Zhong, Yuanyi. "Sample-Efficient Learning With Self-Supervision", University of Illinois at Urbana-Champaign

---

98 < 1% match ("Artificial Neural Networks and Machine Learning – ICANN 2024", Springer Science and Business Media LLC, 2024)

"Artificial Neural Networks and Machine Learning – ICANN 2024", Springer Science and Business Media LLC, 2024

---

99 < 1% match (Grama Satyanarayana, Srivatsa. "Monocular Event Camera Odometry Using Deep Learning", University of Washington, 2025)

Grama Satyanarayana, Srivatsa. "Monocular Event Camera Odometry Using Deep Learning", University of Washington, 2025

---

100 < 1% match (Yasin Almalioglu, Mehmet Turan, Muhamad Risqi U. Saputra, Pedro P.B. de Gusmão, Andrew Markham, Niki Trigoni. "SelfVIO: Self-supervised deep monocular Visual–Inertial Odometry and depth estimation", Neural Networks, 2022)

Yasin Almalioglu, Mehmet Turan, Muhamad Risqi U. Saputra, Pedro P.B. de Gusmão, Andrew Markham, Niki Trigoni. "SelfVIO: Self-supervised deep monocular Visual–Inertial Odometry and depth estimation", Neural Networks, 2022

---

101 < 1% match (Zhao, Qingqing. "Building Human-Like Embodied Agents from Humans", Stanford University)

Zhao, Qingqing. "Building Human-Like Embodied Agents from Humans", Stanford University

---

102 < 1% match (Feng, Qiaojun. "3D Semantic Scene Understanding and Reconstruction", University of California, San Diego, 2024)

Feng, Qiaojun. "3D Semantic Scene Understanding and Reconstruction", University of California, San Diego, 2024

---

103 < 1% match (Peijing Li, Hexiong Yao, Zhiqiang Dai, Xiangwei Zhu. "Asynchronous Visual-Inertial Odometry for Event Cameras", The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2024)

Peijing Li, Hexiong Yao, Zhiqiang Dai, Xiangwei Zhu. "Asynchronous Visual-Inertial Odometry for Event Cameras", The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2024

104 < 1% match (Internet from 08-Apr-2025)
https://researchrepository.universityofgalway.ie/server/api/core/bitstreams/4108a3ce-5274-4f30-bb87-cf5f6b3471e7/content

105 < 1% match (Internet from 01-Apr-2025)
https://ses.library.usyd.edu.au/bitstream/handle/2123/33750/Event_based_3D_Reconstruction__Innovative_Event_Representat isAllowed=y&sequence=6

106 < 1% match (Internet from 19-Aug-2023)
https://uwspace.uwaterloo.ca/bitstream/handle/10012/18293/Azzi_Charbel.pdf?isAllowed=y&sequence=1

107 < 1% match ("Computer Vision – ECCV 2022 Workshops", Springer Science and Business Media LLC, 2023)
"Computer Vision – ECCV 2022 Workshops", Springer Science and Business Media LLC, 2023

108 < 1% match (Zhang, Yanyu. "Towards AI-Aided Multi-User AR: Cooperative Visual-Inertial Odometry Enhanced by Point-Line Features and Neural Radiance Fields", University of California, Riverside)
Zhang, Yanyu. "Towards AI-Aided Multi-User AR: Cooperative Visual-Inertial Odometry Enhanced by Point-Line Features and Neural Radiance Fields", University of California, Riverside

109 < 1% match (Le Gentil, Cedric. "Gaussian Process Preintegration for Inertial-Aided Navigation Systems", University of Technology Sydney (Australia), 2023)
Le Gentil, Cedric. "Gaussian Process Preintegration for Inertial-Aided Navigation Systems", University of Technology Sydney (Australia), 2023

110 < 1% match (Sangay Tenzin, Alexander Rassau, Douglas Chai. "Application of Event Cameras and Neuromorphic Computing to VSLAM: A Survey", Biomimetics, 2024)
Sangay Tenzin, Alexander Rassau, Douglas Chai. "Application of Event Cameras and Neuromorphic Computing to VSLAM: A Survey", Biomimetics, 2024

111 < 1% match (Zhu, Alex Zihao. "Event-Based Algorithms for Geometric Computer Vision.", University of Pennsylvania, 2020)
Zhu, Alex Zihao. "Event-Based Algorithms for Geometric Computer Vision.", University of Pennsylvania, 2020

112 < 1% match (Internet from 24-Oct-2022)
https://drum.lib.umd.edu/bitstream/handle/1903/29252/Parameshwara_umd_0117E_22654.pdf?isAllowed=y&sequence=2

113 < 1% match (Leighton, Brenton. "Accurate 3D Reconstruction of Underwater Infrastructure Using Stereo Vision", University of Technology Sydney (Australia), 2024)
Leighton, Brenton. "Accurate 3D Reconstruction of Underwater Infrastructure Using Stereo Vision", University of Technology Sydney (Australia), 2024

114 < 1% match (Peiyu Chen, Weipeng Guan, Feng Huang, Yihan Zhong, Weisong Wen, Li-Ta Hsu, Peng Lu. "ECMD: An Event-Centric Multisensory Driving Dataset for SLAM", IEEE Transactions on Intelligent Vehicles, 2023)
Peiyu Chen, Weipeng Guan, Feng Huang, Yihan Zhong, Weisong Wen, Li-Ta Hsu, Peng Lu. "ECMD: An Event-Centric Multisensory Driving Dataset for SLAM", IEEE Transactions on Intelligent Vehicles, 2023

115 < 1% match (Xiaofei Wu, Tao Liu, Caoji Li, Yuexin Ma, Yujiao Shi, Xuming He. "FastGrasp: Efficient Grasp Synthesis with Diffusion", Qeios Ltd, 2024)
Xiaofei Wu, Tao Liu, Caoji Li, Yuexin Ma, Yujiao Shi, Xuming He. "FastGrasp: Efficient Grasp Synthesis with Diffusion", Qeios Ltd, 2024

< 1% match ("Computer Vision – ECCV 2016", Springer Nature, 2016)

116 "Computer Vision – ECCV 2016", Springer Nature, 2016

117 < 1% match (Suraj Bijjahalli, Roberto Sabatini, Alessandro Gardi. "Advances in intelligent and autonomous navigation systems for small UAS", Progress in Aerospace Sciences, 2020)

Suraj Bijjahalli, Roberto Sabatini, Alessandro Gardi. "Advances in intelligent and autonomous navigation systems for small UAS", Progress in Aerospace Sciences, 2020

118 < 1% match ("Computer Vision – ACCV 2020", Springer Science and Business Media LLC, 2021)

"Computer Vision – ACCV 2020", Springer Science and Business Media LLC, 2021

119 < 1% match (Chuanzhi Xu, Haoxian Zhou, Haodong Chen, Vera Chung, Vincent Qu. "A Survey on Event-driven 3D Reconstruction: Development under Different Categories", Qeios Ltd, 2025)

Chuanzhi Xu, Haoxian Zhou, Haodong Chen, Vera Chung, Vincent Qu. "A Survey on Event-driven 3D Reconstruction: Development under Different Categories", Qeios Ltd, 2025

120 < 1% match (Ling Gao, Hang Su, Daniel Gehrig, Marco Cannici, Davide Scaramuzza, Laurent Kneip. "A 5-Point Minimal Solver for Event Camera Relative Motion Estimation", 2023 IEEE/CVF International Conference on Computer Vision (ICCV), 2023)

Ling Gao, Hang Su, Daniel Gehrig, Marco Cannici, Davide Scaramuzza, Laurent Kneip. "A 5-Point Minimal Solver for Event Camera Relative Motion Estimation", 2023 IEEE/CVF International Conference on Computer Vision (ICCV), 2023

121 < 1% match (Yi-Fan Zuo, Jiaqi Yang, Jiaben Chen, Xia Wang, Yifu Wang, Laurent Kneip. "DEVO: Depth-Event Camera Visual Odometry in Challenging Conditions", 2022 International Conference on Robotics and Automation (ICRA), 2022)

Yi-Fan Zuo, Jiaqi Yang, Jiaben Chen, Xia Wang, Yifu Wang, Laurent Kneip. "DEVO: Depth-Event Camera Visual Odometry in Challenging Conditions", 2022 International Conference on Robotics and Automation (ICRA), 2022

122 < 1% match (publications)

Yigang He, Xue Qing. "Automatic Control, Mechatronics and Industrial Engineering", CRC Press, 2019

123 < 1% match ("Computer Vision – ECCV 2024", Springer Science and Business Media LLC, 2025)

"Computer Vision – ECCV 2024", Springer Science and Business Media LLC, 2025

124 < 1% match (Arukgoda, Janindu Sithumini. "Vector Distance Transform Maps for Autonomous Mobile Robot Navigation", University of Technology Sydney (Australia), 2023)

Arukgoda, Janindu Sithumini. "Vector Distance Transform Maps for Autonomous Mobile Robot Navigation", University of Technology Sydney (Australia), 2023

125 < 1% match (publications)

H.L. Gururaj, Francesco Flammini, S. Srividhya, M.L. Chayadevi, Sheba Selvam. "Computer Science Engineering", CRC Press, 2024

126 < 1% match (Liu, Li Yang. "Towards Observable Urban Visual SLAM", University of Technology Sydney (Australia), 2023)

Liu, Li Yang. "Towards Observable Urban Visual SLAM", University of Technology Sydney (Australia), 2023

127 < 1% match (Yuhang Ming, Xingrui Yang, Weihan Wang, Zheng Chen, Jinglun Feng, Yifan Xing, Guofeng Zhang. "Benchmarking neural radiance fields for autonomous robots: An overview", Engineering Applications of Artificial Intelligence, 2025)

Yuhang Ming, Xingrui Yang, Weihan Wang, Zheng Chen, Jinglun Feng, Yifan Xing, Guofeng Zhang. "Benchmarking neural radiance fields for autonomous robots: An overview", Engineering Applications of Artificial Intelligence, 2025

128 < 1% match (Ali Rida Sahili, Saifeldin Hassan, Saber Sakhrieh, Jinane Mounsef, Noel Maalouf, Bilal Arain, Tarek Taha. "A Survey of Visual SLAM Methods", IEEE Access, 2023)

Ali Rida Sahili, Saifeldin Hassan, Saber Sakhrieh, Jinane Mounsef, Noel Maalouf, Bilal Arain, Tarek Taha. "A Survey of Visual SLAM Methods", IEEE Access, 2023

129 < 1% match (Dupeng Cai, Ruoqing Li, Zhuhua Hu, Junlin Lu, Shijiang Li, Yaochi Zhao. "A comprehensive overview of core modules in visual SLAM framework", Neurocomputing, 2024)

Dupeng Cai, Ruoqing Li, Zhuhua Hu, Junlin Lu, Shijiang Li, Yaochi Zhao. "A comprehensive overview of core modules in visual SLAM framework", Neurocomputing, 2024

130 < 1% match (Huai, Zheng. "Robocentric Visual-Inertial Localization and Mapping", University of Delaware, 2023)

Huai, Zheng. "Robocentric Visual-Inertial Localization and Mapping", University of Delaware, 2023

131 < 1% match (publications)

John Billingsley. "Robotics and automation for improving agriculture", Burleigh Dodds Science Publishing, 2019

132 < 1% match (Johnson, Jacob C.. "Continuous-Time Trajectory Estimation and Its Application to Sensor Calibration and Differentially Flat Systems", Brigham Young University, 2023)

Johnson, Jacob C.. "Continuous-Time Trajectory Estimation and Its Application to Sensor Calibration and Differentially Flat Systems", Brigham Young University, 2023

133 < 1% match (Joshi, Bharat. "Robust Underwater State Estimation and Mapping", University of South Carolina, 2024)

Joshi, Bharat. "Robust Underwater State Estimation and Mapping", University of South Carolina, 2024

134 < 1% match (Kang, Peng. "Event-driven Processing and Learning with Spiking Neural Networks", Northwestern University, 2024)

Kang, Peng. "Event-driven Processing and Learning with Spiking Neural Networks", Northwestern University, 2024

135 < 1% match (Queen, Kendall J.. "Event-Based Perception for Ground Vehicle Control", University of Pennsylvania, 2023)

Queen, Kendall J.. "Event-Based Perception for Ground Vehicle Control", University of Pennsylvania, 2023

136 < 1% match (Shaopeng Li, Daqiao Zhang, Yong Xian, Bangjie Li, Tao Zhang, Chengliang Zhong. "Overview of deep learning application on visual SLAM", Displays, 2022)

Shaopeng Li, Daqiao Zhang, Yong Xian, Bangjie Li, Tao Zhang, Chengliang Zhong. "Overview of deep learning application on visual SLAM", Displays, 2022

137 < 1% match (Wang, Zihao. "Synergy of Physics and Learning-Based Models in Computational Imaging and Display.", Northwestern University, 2020)

Wang, Zihao. "Synergy of Physics and Learning-Based Models in Computational Imaging and Display.", Northwestern University, 2020

138 < 1% match (Xin Peng, Ling Gao, Yifu Wang, Laurent Kneip. "Globally-Optimal Contrast Maximisation for Event Cameras", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021)

Xin Peng, Ling Gao, Yifu Wang, Laurent Kneip. "Globally-Optimal Contrast Maximisation for Event Cameras", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021

139 < 1% match (Yalçın, Haktan. "Lie Algebra Based Augmented State EKF Design for Information Fusion in Odometry", Middle East Technical University (Turkey))

Yalçın, Haktan. "Lie Algebra Based Augmented State EKF Design for Information Fusion in Odometry", Middle East Technical University (Turkey)

140 < 1% match (Zhao, Tianxiang. "Deep Learning for Structured Data: Weak Supervision and Interpretability", The Pennsylvania State University)

Zhao, Tianxiang. "Deep Learning for Structured Data: Weak Supervision and Interpretability", The Pennsylvania State University

141 < 1% match ("Computer Vision – ECCV 2020", Springer Science and Business Media LLC, 2020)

"Computer Vision – ECCV 2020", Springer Science and Business Media LLC, 2020

142 < 1% match ("RGB-D Image Analysis and Processing", Springer Science and Business Media LLC, 2019)
"RGB-D Image Analysis and Processing", Springer Science and Business Media LLC, 2019

143 < 1% match (Geneva, Patrick F.. "Efficient, Consistent, and Persistent Visual-Inertial Navigation", University of Delaware)
Geneva, Patrick F.. "Efficient, Consistent, and Persistent Visual-Inertial Navigation", University of Delaware

144 < 1% match ()
Etxeberria Garcia, Mikel. "Computer vision techniques for autonomous vehicles applied to urban underground railway", Mondragon Unibertsitatea. Goi Eskola Politeknikoa, 2022

145 < 1% match (Shi, Yujiao. "Cross-View Image-Based Localization and Synthesis", The Australian National University (Australia), 2023)
Shi, Yujiao. "Cross-View Image-Based Localization and Synthesis", The Australian National University (Australia), 2023

146 < 1% match (Wang, Yifu. "Novel Camera Architectures for Localization and Mapping on Intelligent Mobile Platforms", The Australian National University (Australia), 2021)
Wang, Yifu. "Novel Camera Architectures for Localization and Mapping on Intelligent Mobile Platforms", The Australian National University (Australia), 2021

paper text:

UNIVERSITY OF HONG KONG DOCTORAL THESIS Event-based Vision

for 6-DOF Pose Tracking and 3D Mapping 1

Author: Weipeng GUAN Supervisor: Prof. Peng LU

Co-Supervisor: Prof. James LAM A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy in the Mechanical Engineering Faculty of Engineering May 9, 2025 **Abstract of thesis entitled** 23

Event-based Vision

for 6-DOF Pose Tracking and 3D Mapping 1

Submitted by Weipeng GUAN

for the degree of Doctor of Philosophy at The University of Hong Kong in May, 2025 **Simultaneous Localization and Mapping (SLAM**) serves as **a** 22

foundational technol- ogy for emerging

applications, such as robotics, **autonomous driving**, embodied intel- ligence, **and augmented** / virtual **reality** 12

. However, traditional image-based SLAM sys- tems still struggle with reliable pose estimation and 3D reconstruction under challeng- ing conditions involving high-speed motion and extreme illumination variations.

Event cameras, also known as dynamic vision sensors 39

, have recently emerged as a promising alternative to standard cameras for visual perception. Instead of capturing intensity im- ages

> **at a fixed** frame **rate**, event cameras **asynchronously measure per-pixel brightness changes**, producing **a stream of events that encode the time**, pixel **location, and sign of the brightness changes**. They **offer attractive** 57

advantages,

> **including high temporal resolution** (MHz-**level), high dynamic range (HDR, 140 dB**), low latency (microsecond), **no motion blur** 5

,

> **and low power consumption. However, integrating event cameras into** 47

SLAM systems presents significant challenges due to the fundamentally different char- acteristics of asynchronous event streams compared to conventional intensity images, and new paradigm shifts are required. This dissertation presents innovative solutions and advancements for event-based SLAM. It begins with the development of Mono-EIO, a monocular event-inertial odom- etry framework that tightly integrates event-corner features with IMU preintegration. These event-corner features are temporally and spatially associated using novel event- based representations with a spatial-temporal and exponential decay kernel, and are subsequently incorporated into a keyframe-based sliding window optimization frame- work. Mono-EIO achieves high-accuracy, real-time 6-DoF ego-motion estimation even under aggressive motion and HDR conditions. Building upon this foundation, the thesis introduces PL-EVIO,

> **an event-based visual-inertial odometry framework that** 109

combines event cameras with standard cameras to enhance robustness. The PL-EVIO

> **utilizes line-based event features to provide additional** 2

structural constraints

> **in human- made** environments, **while point-based event and image features** 2

are effectively man- aged to complement each other. This framework has been successfully applied to quadrotor onboard pose feedback control, enabling complex maneuvers such as flip- ping and operation in low-light conditions. Additionally, the thesis includes

> **ESVIO, the first** stereo **event-based visual inertial odometry** framework. **The** 7

thesis also presents DEIO, a learning-optimization-combined framework that tightly coupled fuses the learning-based event data association with the IMU measure- ments within graph-based optimization.

> **To the best of our knowledge**, DEIO is **the first** learning-based **event-inertial odometry** 8

, outperforming over 20 vision-based methods across 10 challenging real-world benchmarks. Finally, the thesis proposes EVI-SAM, a full SLAM system that tackles both

6-DoF pose tracking and 3D dense mapping using a **monocular event camera** 1

. Its tracking module is the first hybrid approach that inte- grates

both direct-based and feature-based methods within an **event-based** 1

framework. The mapping module, on the other hand, is the first

to achieve event-based dense and textured 3D reconstruction without GPU acceleration 1

by employing a non-learning ap- proach. This method not only successfully recovers 3D scenes structure under aggres- sive motions but also demonstrates superior performance compared to image-based NeRF or RGB-D cameras. Through these contributions, this dissertation significantly advances SLAM, offering robust solutions and paving the way for future research and applications in event camera. (485 words) Event-based Vision

for 6-DOF Pose Tracking and 3D Mapping 1

by Weipeng GUAN

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of Doctor of Philosophy at University of Hong Kong May, 2025 **COPYRIGHT** ©2025, **BY** WEIPENG GUAN **ALL RIGHTS RESERVED. i Declaration I**, Weipeng GUAN, **declare that this thesis titled**, “Event-**based** 29

Vision

for 6-DOF Pose Tracking and 3D Mapping 1

”,

which is submitted in fulfillment of the requirements for the Degree of Doctor of Philosophy, represents my own work except where due acknowledgement have been made. I further declared that it has not been previously included in a thesis, dissertation, or report submitted to this University or to any other institution for a degree, diploma or other qualifications. Signed: Date: May **9** 22

, 2025 I dedicate this thesis

to my family, **for their unconditional love and endless support**. ii Acknowledgements **I** 39

embarked on my research journey with profound passion as an undergraduate student in 2014. Over the years, despite encountering numerous challenges and dif- ficulties, my passion for academic and research has remained steadfast. While what has ultimately enabled me to earn my PhD today is not only my perseverance and pas- sion but also the unwavering support and encouragement from my family, supervisor, friends, and colleagues.

First and foremost, I would like to express my deepest gratitude and appreciation **to my** supervisor **Prof**. Lu Peng **for his** enduring **support, guidance and** 59

inspiration.

I am profoundly **grateful for the** academic **freedom and resources** 32

he generously pro- vided, which allowed me to explore and develop my ideas to their fullest potential. His open-mindedness and encouragement to pursue innovative approaches have been piv- otal in the development of my research. Beyond the academic guidance, I am equally grateful for his kindness, patience, and steadfast support during moments of doubt and challenge. His dedication, insightful counsel, and constant encouragement have been indispensable to my growth as a researcher and individual. Prof. Lu is truly the best supervisor who has always been there for me, providing valuable advice and support throughout my research and living journey. I am also deeply indebted to Prof. Huang Guoquan, Prof. Zhang Fu, and Prof. Wu Yuxiang for graciously agreeing to review my thesis and for their constructive feed- back and suggestions. I want to thank Prof. Deng Xiaowei for chairing the examination committee. Additionally,

I would like to extend my heartfelt thanks **to Prof. James Lam, Prof**. Wen Shangsheng, **Prof**. Dong Chao, **Prof** 23

.Li-Ta HSU, and Prof. Wen Weisong for their invaluable guidance and support during various stages of my research journey. My sincere appreciation goes to all members of the ArcLab for their fruitful re- search

discussions and their effort in keeping a friendly, pleasant, and fun 32

study envi- ronment. Special thanks go to Chen Han, Peng Rui, Huang Rui, Lu Minghao, Chen Peiyu, Lin Fuling, Xie Yuhan, Li Mingyang, Tao Yuting, Dong Yinzhao, Shu Zhengjie, Wang Yu, Li Xinqi, and Luo Zeren. Their camaraderie, intellectual discussions, and unwavering support have been indispensable to my progress. The memories we have created to- gether will remain cherished for a lifetime, and I am truly honored to be a part of such incredible team. Finally, and most importantly, I owe my deepest gratitude

to my family, whose un- conditional **love and support have been** 60

the foundation for me. This thesis would not have been possible without their encouragement. I owe everything to my parents, who have always been there for me, providing me with the best education and unwavering support.

I also extend my heartfelt thanks to my 70

lover, Tu Jun, for being my greatest source of motivation and support. Her unwavering belief in me and her encourage- ment

have been a constant source of strength. I am equally grateful **to** 60

her parents who provides me with many trust

and encouragement, which **have meant** so much **to me**. In closing, **I am** 108

deeply humbled and grateful for the efforts and care of everyone who has supported me. The lessons I have learned, the memories I have made, and the inspiration I have drawn from this experience guide me as I step into the next phase of my life. Thank you all for your dedication, kindness, and for being part of this meaningful journey. Weipeng GUAN University of Hong Kong May 9, 2025 v List of Publications JOURNALS: [1] Guan Weipeng, Chen Peiyu, Zhao Huibin, Wang Yu, Lu Peng, “EVI-SAM: Ro- bust,

Real-time, Tightly-coupled Event-Visual-Inertial State Estimation and 3D Dense Mapping", in **Advanced Intelligent Systems** 11

, pp. 1-24, 2024. [IF:7.8] [2] Guan Weipeng*, Chen Peiyu*, Xie Yuhan, Lu Peng, "

PL-EVIO: Robust Monocu- lar **Event-based Visual Inertial Odometry with Point and Line Features**", in **IEEE Transactions on Automation Science and Engineering, pp. 1-17, 2023** 52

. [IF:5.6; *Co-first Author; simultaneously present at ICRA2024] [3] Chen Peiyu*, Guan Weipeng*, Lu Peng, "

ESVIO: Event-based Stereo Visual In- ertial **Odometry", in IEEE Robotics and Automation Letters, pp.3661-3668** 47

, 2023. [IF:5.2; *Co-first Author; simultaneously present at IROS2023] [4] Guan Weipeng*, Lin Fuling*, Chen Peiyu, Lu Peng, "DEIO: Deep Event Inertial Odometry". [*Co-first Author] [5] Chen Peiyu*, Guan Weipeng*, Huang Feng*, Zhong Yihan, Wen Weisong, Hsu Li-Ta, Lu Peng, "

ECMD: An Event-Centric Multisensory Driving Dataset for SLAM", in **IEEE Transactions on Intelligent Vehicles** 1

, pp.1-10, 2023. [IF:8.2; *Co-first Author] [6] Zhao, Huibin*, Guan Weipeng*, Lu Peng, "LVI-GS: Tightly-coupled LiDAR-Visual- Inertial SLAM using 3D Gaussian Splatting",

in IEEE Transactions on Instrumenta- tion **and Measurement, pp.1** 43

-8, 2025. [IF:5.6; *Co-first Author] [7] Chen Peiyu*, Lin Fuling*, Guan Weipeng, Lu Peng, "

SuperEIO: Self-Supervised Event Feature Learning for Event Inertial Odometry 11

". CONFERENCES: [1] Guan Weipeng, Lu Peng, "

Monocular Event Visual Inertial Odometry based on Event-corner using Sliding Windows Graph-based Optimization", in 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2438-2445 1

, 2022. vii

Contents Abstract Declaration Acknowledgements List of Publications List of Figures List of Tables 1 Introduction 1.1 23

Event Camera **for Computer Vision and Robotics 1.1.1** 9
Why We Need **Event**

Camera? . . . . . .

1.1.2 **Working Principle** of **Event Cameras** 31

. . . . . 1.1.3

Advantages and Challenges of Event Cameras 9

. . . . . 1.2 Related Works . . . . . 1.2.1 Visual SLAM . . . . . 1.2.2 Pose Estimation using Event Cameras . . . . . 1.2.3 Depth Estimation using Event Cameras . . . . . 1.

2.3.1 Event-based **Monocular Depth Estimation** . . . . . 1.**2.3.2** Event-based **Stereo Depth Estimation** 1.**2** 65

.4 Datasets for Event-based SLAM Benchmarking . . . . . i i ii v xiii xxv 1 1 1 3 5 8 8 10 10 14 17 18 19 19 21 22 1.3 Thesis Outline . . . . . 2 Mono-EIO: Monocular Event-Inertial Odometry 2.1 Introduction . . . . .

2.2 Related Works . . . . . 2.2.1 Event-**based** SLAM/VO . . . . . **2.2.2** Event-**based** Visual-Inertial Odometry **2** 26

.3 Framework Overview . . . . . 2.4 Event-based Representations . . . . . 2.4.1 Surface of Active Events (SAE) . . . . . 2.4.2 Time Surface (TS) . . . . . 2.4.3 TS with Polarity . . . . . 2.4.4 Normalized TS without Polarity . . . . . 2.5 Event-based Data Association . . . . . . 2.5.1 Event-Corner Feature Detection . . . . . 2.6 Tightly-coupled Event-Inertial State Estimation . . . . . 2.6.1 Formulation of the 6-DoF State Estimation Problem . . . . . 2.6.2 Event-based Visual Constraint . . . . . 2.6.3 IMU Pre-integration Constraint . . . . . 2.6.4 Event-based Re-localization Constraint . . . . . 2.6.5 Initialization and other Implementation Details . . . . . 2.7 Experiments . . . . . 2.8 Conclusion . . . . . 2.9 Related Publications . . . . . 3 EVIO: Image-aided

Event-based Visual-inertial Odometry 3.1 Introduction 111

. . . . . 25 29 30 32 32 33 33 34 35 36 36 37 38 38 40 43 44 44 45 46 48 48 49 50 51 53 54 54 56 57 58 3.2 Related Works 62 3.2.1

Event-based Representation and Feature Extraction 2

. . . . . 62 3.2.2 Event-based Motion Estimation . . . . . 63 3.3 Framework Overview 66 3.4

Motion Compensation for the Event Stream using IMU 5

. . . . . 68 3.5 Point-based Event Measurement . . . . . 69 3.6 Line-based Event Measurement . . . . . 71 3.6.1 Event-based Line Feature Detection and Matching . . . . . 71 3.6.2 Event-based Line Constraint . . . . . 72 3.7 Point-based Image Measurements . . . . . 75 3.8 Formulation of the PL-EIO and PL-EVIO . . . . . 75 3.9 Experiments . . . . . 76 3.9.1 Evaluation in HDR Scenarios . . . . . 77 3.9.2 Evaluation in Aggressive Motions . . . . . 79 3.9.3 Online Quadrotor-flight Evaluation . . . . . 80 3.9.4 Aggressive Quadrotor-flip Evaluation . . . . . 83 3.9.5 Quadrotor-flight Evaluation in Dark Environment . . . . . 85 3.9.6 Outdoor Large-scale Evaluation . . . . . 87 3.9.7 Real-time Analysis 89 3.9.8

Ablation Study on Different Event Representations for Feature Tracking 2

. . . . . 90 3.9.9

Ablation Study on Time Decay Kernels of the TS with Polarity 2

. . 90 3.10 Conclusion . . . . . 93 3.11 Related Publications . . . . . 93 4 Event-based Hybrid Odometry 95

4.1 Introduction . . . . . 95 **4.2 Related Works . . . . . 97 4.2.1** Feature-**based** Event Pose Tracking **97 4.2.2** 91

Direct-based Event Pose Tracking . . . . . 98 4.3 Framework Overview . . . . . 98 4.6 Feature-based Measurements . . . . . 103 4.7 Hybrid Optimization for Feature-based and Direct-based Measurement 104 4.8 Experiments . . . . . 104 4.8.1 Evaluation in Aggressive Motion and HDR Scenarios . . . . . 105 4.8.2 Ablation Study of Direct-based, Feature-based, and the Hybrid Framework . . . . . 108 4.8.3 Ablation Study on Combination Event and Image . . . . . 111 4.9 Conclusion . . . . . 112 4.10 Related Publications . . . . . 112 5 DEIO: Deep Learning-based Event-Inertial

Odometry 113 5.1 Introduction . . . . . . 114 5.2 Related Work 116 5.2.1 60

Non-learning Approaches for Event-based VO . . . . . 116 5.2.2 Learning-based Approaches for Event-based VO . . . . . 117 5.3 Framework Overview . . . . . 119 5.4 Learning-based Event Data Association . . . . . 120 5.5 Learnable Hessian Information Extraction . . . . . 122 5.6 Graph-Based Event-IMU Combined Bundle Adjustment . . . . . 124 5.7 Online EIO System Design . . . . . 126 5.8 Experiments . . . . . 128 5.8.1 Comparisons with SOTA Methods in Challenge Benchmarks . . 128 5.8.2 Qualitative Evaluation on Challenge Benchmarks . . . . . 138 5.8.3 Test on Night Driving Scenarios . . . . . 139 5.8.4 Test on Dark Quadrotor-Flight . . . . . 143 5.8.5 Ablation Study on Real-world Data Fine-tuning . . . . . 145 5.8.6 Ablation Study on Event Representations . . . . . 145 5.8.7 Ablation Study on Number of Event Patches . . . . . 148 5.8.8 Runtime Analysis . . . . . 148 5.8.9 Discussion . . . . . 149 5.9 Integrating the Proximity Loop Closure and Global Bundle Adjustment 150 5.10 Conclusion . . . . . 152 5.11 Related Publications . . . . . 152 6 Event-based 3D Dense Reconstruction 153

6.1 Introduction . . . . . 153 **6.2 Related Works** 155 **6** 137

.2.

1 Monocular Depth Estimation . . . . . 155 6.2.**2 Stereo Depth Estimation** 156 6.**3** 65

Framework Overview . . . . . 156 6.4 Purely Event-based Semi-dense Mapping . . . . . 157 6.5 Image-guided Event-based Dense Mapping . . . . . 159 6.6 TSDF-based Map Fusion . . . . . 163 6.7 Experiments . . . . . 164 6.7.1 Mapping Performance in Diverse Scenarios . . . . . 165 6.7.2 Real-time Onboard Mapping Evaluation . . . . . 167 6.7.2.1 Local Mapping Performance . . . . . 168 6.7.2.2 Global Mapping Performance . . . . . 169 6.7.3 Mapping Performance Comparison with Baselines . . . . . 170 6.7.3.1 Comparison with Traditional Image-based Mapping . . 170 6.7.3.2 Comparison with Event-based Mapping . . . . . 172 6.7.3.3 Comparison with Learning-based Mapping . . . . . 174 6.7.3.4 Comparison with NeRF-based Mapping . . . . . 176 6.7.4 Mapping Performance in Challenge Situations . . . . . 178 6.7.4.1 HDR Scenarios . . . . . 178 6.7.4.2 Aggressive Motions . . . . . 180 6.7.5 Running Time Analysis . . . . . . 180 6.7.6 Limitations and Discussions . . . . . 181 6.8 Conclusion 182 6.9

23

Moving a checkerboard (a) in front of an event camera in different direc- tions [5]. (**b)-(d) shows visualizations of the events obtained by binning events over a short time interval. Pixels that do not change intensity are represented in gray, whereas pixels that** increase **or** decrease **intensity are represented in bright and dark, respectively. Clearly, (b) (only** verti- cal **edges), (c) (only horizontal edges), and (d) can hardly be related to each other in the absence of underlying photometric information (a**

9

An overview of the sensor setup and dataset visualization from ECMD [6]. **Above: The** sensor **suite is mounted on the top of the vehicle. Below: Sensor systems include two sets of stereo event cameras, stereo** indus- trial **cameras, an infrared camera, a top-installed LiDAR with two slanted LiDAR, IMU, and GNSS-RTK/INS systems. Each sensor is indicated with the letter box**

8

the timestamps of the latest **event** triggered **in each pixel**

32

3 (a) The raw event stream (left), the Time Surface with polarity (middle), and the normalized Time Surface without polarity (right); (b) Event-corner tracking on Time Surface with polarity; (c) Loop detection using the event-corner features and the normalized Time Surface without

3

the Arc* algorithm for event-corner detection. (a) For **the** new coming **event**

5

(green) located in a corner region (grey background), triggers the inspection of a circular neighborhood C (blue) in the local SAE. (b) The circular arc can be initial- ized from the most recent element Anew (higher intensity values indicate newer timestamps), with adjacent clockwise ECW (cyan) and counter- clockwise ECCW (red) elements defining arc boundaries. (c) The Anew would be expanded up to a minimum arc length Lmin = 3, and the ECW go to the new position. (d) During the following iterations, the position of ECW is kept updated since its timestamp is bigger than the ECCW. (e) The position of the ECCW would be updated since its timestamp is bigger than the ECW, while the first update step is skipped with minimum arc length Lmin = 3, and

the circle is completed. The event is classified as a corner 32

feature as the complementary continuous arc length is 4, which satisfies the predefined threshold range [Lmin, Lmax] = [3, 6]. 25 34 35 37 39 2.5

Event-corner features generation. The event-corner features are first ex- tracted **from the asynchronous event stream (a), then the TS with polarity (b) is used as a mask to further select the event-corner features, ensuring a uniform distribution** 2

. . . . . 2.6 Examples visualization for the feature tracking

using the event-corner features and the Time Surface with **polarity** 3

. The green arrows

on the event-corner features depict **the** 4

estimated direction and magnitude of the optical flow, while the green lines linking pairs of event-corner features indicate the temporally tracked points across consecutive frames. . . . . 2.7 Examples of visualization for the

loop detection using the event-corner features and the normalized Time Surface without polarity 3

. . . . . 2.8

Comparison of translation and rotation estimates of our proposed Mono- EIO **against ground truth; The relative errors of the translation and yaw angle** 3

. . . . . 2.9

Estimated trajectory in outdoor environment aligned with Google map 3

. 2.10 (

a) Our quadrotor platform and VICON room; (b) The estimated trajec- tory **and its comparison against VICON** 3

. . . . . 3.1 3.2 3.3 41 42 44 52 54 55

Our ESVIO [73] **provides robust and accurate, real-time pose feedback for drones under aggressive motion. Events provide rich and reliable** 7

features, while only a few features are tracked in image frames in high-speed motion. Left bottom: stereo event-based feature tracking. Right bottom: stereo image-based feature tracking

. . . . . 58 (

a) Our PL-EVIO combines events, images, and IMU to provide robust state estimation during aggressive motion. It can provide onboard feedback- control for quadrotors with limited computational resources. (b) Our PL-EVIO in the outdoor environment. Left: event-corner features in the event; Middle: line-based features in the event; Right: point-based fea- tures in the image 2

. 60

The framework of our PL-EIO (Event + IMU) and PL-EVIO (Event + Image + IMU 4

). . . . . 67 3.4 Visualization of the event streams before and after applying our pro- posed motion compensation. 69 3.5

The event-corner feature detection and tracking: (a) Detecting features from raw event streams; (b) Using the TS with polarity as the mask for uniform distribution of the event-corner features; (c)Tracking feature in the TS with polarity 2

. . . . . 70 3.6

Three different kinds of features in our PL-EVIO framework: event-corner features, event-based line features, and image-based point features 2

. The animation for three different types of features

can be viewed on the project website https://kwanwaipang.github.io 101

/PL-EVIO/. 71 3.7

The relative pose error comparison of our PL-EVIO with EIO [44], Ultimate- SLAM [49], and our Mono-EIO 5

[2] . . . . . 81 3.8 Our self-designed quadrotor platform. 82 3.9

The estimated trajectory of our PL-EVIO on the quadrotor flight and its comparison against the ground truth (Taking the Onboard_test_1 as an example 2

). 82 3.10

Onboard quadrotor flight in screw pattern using our PL-EVIO as feed-back control 2

current event mat (white) and the previous 2D event mat (red). (b) Event-based 2D-3D alignment: The current 2D event mat aggregated through a small number of events(b1) is warped to the projected mat recovered from the event-based 3D semi-dense depth in (b2). The result (b3) is a good alignment between the 2D current event mat (white) and the projected 3D event-based map (color

). 102 4.3

Comparing the estimated trajectory in terms of translation and rotation produced by our 1

event-based hybrid pose tracking

with the ground truth trajectory in DAVIS240C [144], **HKU-dataset, and VECtor** [157]. (**a) and (b) show the comparison among our** framework **with a feature-based event pipeline only, a direct-based event pipeline only, and the full event- based hybrid pipeline (feature-based + direct-based, F-D-based); (c) and (d) show the comparison between the PL-EVIO (feature-based) and our** event-based **hybrid** pose tracking. **The red dashed lines highlight the disparity between featured-based and direct-based methods** 1

. . . . . 110 5.1 Overview of the DEIO system. It decouples network training from IMU integration and operates in two phases: offline training and online op- timization.

The main innovations of this work reside in 44

the effective in- tegration of IMU measurements with learning-based methods. During training, a unified event-based optical flow network is trained to pro- vide robust data associations of sparse event patches. At runtime, the Hessian information, derived from the DBA layer in the update opera- tor, is utilized to tightly integrate event patch correspondence with IMU pre-integration through an event patch-based co-visibility factor graph optimization. . . . . 119 5.2 Patch-based co-visibility factor graph for event-IMU combined bundle adjustment. 126 5.3

Comparison of the **estimated position (X, Y, Z)** and orientation (**Roll, Pitch, Yaw** 11

Stereo HKU dataset [2, 73]. The DEIO seam- lessly addresses scale ambiguity and demonstrates precise alignment with the ground truth trajectory. In contrast, the baseline estimates ex- hibit significant scale discrepancies: (a) The baseline trajectory suffers from drift and an overestimated scale. (b) The baseline trajectory shows an underestimated scale. . . . . 131 5.5 The estimated trajectories of our DEIO against the GT in the sequence of ziggy_hdr and rocket_dark from the EDS [67] dataset. The image view (visualization-only) demonstrates the lack of perceptible information un- der low-light conditions, while the event view, though perceptible, re- mains susceptible to interference from the infrared light of the motion capture system. Thanks to our robust learning-based event data associ- ation, the trajectories estimated by DEIO align remarkably closely with the GT. . . . . 135 5.6 The estimated trajectories of our DEIO against the GT in the sequence of indoor_forward_7 from the UZH-FPV [15] dataset. The image view (visualization-only) demonstrates the condition under low texture, HDR, and motion blur. . . . . 136 5.7 Comparison of Estimated trajectories on the DSEC dataset [153]. 138 5.8

The Estimated trajectories of our DEIO **against the ground truth** 4

in the different challenging benchmarks. 140 5.9

The Estimated trajectories of our DEIO **against the ground truth** 4

in the different challenging benchmarks. 141 5.10

The Estimated trajectories of our DEIO **against the ground truth** 4

in the different challenging benchmarks. . . . . 142 5.11 The estimated trajectory of our DEIO in the night driving scenarios [6] and its comparison against the GNSS-INS-RTK as ground truth. The image frame is for visualization only. . . . . 143 5.12 (a) Our quadrotor platform (the image is for visualization-only); (b)

The estimated trajectory of our DEIO in **the quadrotor flight and its** compari- son **against the ground truth** 2

; (c)

The position, orientation, and the corre- sponding **errors of** DEIO **in** quadrotor **flight compared with the ground truth from VICON** 2

. . . . . 144 5.13 Validation results on the training dataset during the fine-tuning process using real-world data from the Vector dataset [157]. . . . . 146 5.14 The training details of the fine-tuning process using real-world data from the Vector dataset [157]. . . . . 147 5.15 Runtime performance (voxels per second) of our DEIO using 48 (P48), 96 (P96), and 120 (P120) event patches per voxel, as well as a 96-patch ver- sion without IMU input (P96 w/o IMU). Values in the brackets indicate the average MPE (%) over all sequences. . . . . 149 5.16 Workflow of the event-based DBA with global bundle adjustment (DEIO+) in the optimization framework. . . . . 150 6.

1 Our EVI-SAM enables the recovery of both the camera pose and dense maps of the scene. The tracking module 1

(detailed in Chapter 4) lever- ages the re-projection

constraint from the feature-based method and the relative pose constraints from the direct-based method within an event- based hybrid tracking framework. The mapping module represents a pioneering event-based dense mapping framework, distinguished as a non-learning method that can conduct real-time dense and texture map- ping **on a standard CPU** 1

. . . . . 154 6.

2 System overview. The EVI-SAM algorithm takes events, images, and IMU as inputs, enabling the recovery of both camera pose and dense map of the scene. The mapping process takes raw event streams as input, using images for guidance, and produces dense and textured 3D map- ping **as output. The tracking thread** 1

(has ben introduced in Chapter 4)

takes event, image, and IMU as input, and constructs the feature-based and direct-based constraints to estimate the 6-DoF pose 1

. . . . . 157 6.3

The model of event-based back-projection and space-sweep **calculations across different depth planes of the DSI** 1

. . . . . 159 6.4

The model of our event-based dense mapping incorporates **edges** de- rived **from the intensity image as guidance. The upper layer represents the event-based semi-dense depth. This layer includes areas where the depth is known (regions successfully recovered through semi-dense** map- ping, **marked in red) and areas with unknown depth (marked in black). The lower layer represents the intensity image with boundary** informa- tion **after segmentation. Since events are triggered in regions with edges, the semi-dense depth and the intensity image edges at the corresponding locations are consistent** 1

. . . . . 160 6.5

The event-based semi-dense and dense mapping of our EVI-SAM. (a) The raw event stream and (g) the intensity image from the event camera; (b) The disparity space image (DSI) of the reference view (RV) point; (c) The purely event-based semi-dense depth generated from our EVI-SAM; (d) The point cloud of the event-based semi-dense depth; (e) The occu- pied **node of the semi-dense mapping after TSDF-fusion; (f) The surface mesh of the semi-dense mapping; (h) The segmentation on the image; (i) The event-based dense depth generated from our EVI-SAM; (j) The point cloud of the event-based dense depth with texture information; (k) The occupied node of the dense mapping after TSDF-fusion; (l) The surface mesh of the dense mapping** 1

; 163 6.6

Qualitative mapping result of our EVI-SAM in various data sequences. Each element includes the image view, dense depth, and the point cloud with texture information. The depth is pseudo-colored, from red (close) to blue (far), in the range of 0.55-6.25 m for the rpg [138] **and davis240c** [144]; **in the range 1.0-6.5 m for MVSEC** [147]; **in the range 0.5-4.0 m for HKU- dataset; in the range** 1

4.0-200 m for DSEC [153], respectively. 166 6.7

The event-based handheld device with the schematics **model for** on- board **evaluation. Please note that the RGB-D camera (Intel D455) is only used for reference, the complete system of our EVI-SAM operates** exclu- sively **with the monocular event camera (DAVIS346** 1

). 167 6.8

Comparing the local texture depth generated by EVI-SAM with the one produced by the RGB-D camera 1

. . . . . 168 6.9

Visualization of the estimated camera trajectory and global 3D recon- struction (**surface mesh) of our EVI-SAM. Sequentially display from right to left includes the event-based dense point** clouds **with texture** informa- tion **and intensity** images, **at selected viewpoints** 1

Mapping result comparison. The first row shows the intensity frames from the event camera. Rows 2 to 4 show semi-depth estimation results from GTS [217, 59], **SGM** [218, 59], **and ESVO** [59], **respectively. The last three rows show the estimated semi-dense depth, dense depth, and texture point cloud from our method. Depth maps are pseudo-colored, from red (close) to blue (far), in the range of 0.55-6.25 m for the rpg** [138] **and in the range 1.0-6.5 m for MVSEC** 1

ground truth depth obtained through 3D LiDAR, the 1

and the ground truth image view. **The** red box highlights **the** difference **of** 55

event-based dense mapping demonstrates performance comparable to 1

Mapping result of EVI-SAM under HDR scenes. The first row shows the intensity frames from the event camera. The second row shows the event-based dense maps, **and the last row is the event-based dense point** clouds **with texture information** 1

of EVI-SAM under aggressive motions. We visualize the results **of our event-based dense mapping along with the** timestamps (**ROS time), and the raw gyroscope and acceleration reading from the IMU. The shaded area in grey represents the phase of aggressive** 1

5 The visualization of various scenarios from ECMD [6] **including event streams, RGB images, and infrared images** 6

. . . . . 196 A.6 The synthetic event stream and the corresponding image frames in the TartanAir dataset. . . . . 198 B.1 Workflow of the eDBA for end-to-end pose tracking. 200 B.2

Architecture of the event-based **feature extractors** network. **D = 128 for the matching-feature extractor and D = 384 for the context-feature** 10

ex- tractor. . . . . 201 B.3 The patch graph. Multiple event-based

patches are extracted from each frame (the **blue** ones) **and are connected to nearby frames** (the **green and** B.4 **purple** ones). . . . . 202 **The** 10

workflow of the correlation operation.

For each edge (k, j) in the patch graph, patch k is re-projected **into a frame j using** Eq. (B.**2**). The **matching features** in frame j **are then cropped using the** re-projected **patch** Pkj. **The correlated matching features (green) and context features** (blue) **form the edge features** 10

. . . . . 204 B.5 Structure of RNN in [26]. Totally Patches K= num of frame in the sliding window × num of extracted patches for each frame. Totally Edges m= num of frame in the sliding window × Totally Patches K. . . . . 206 C.1 The estimated trajectory against GT trajectory (a) with and (b) without trajectory alignment. . . . . 210 xxv

List of Tables 1.**1** Summary **of** the **Event-based** SLAM. **The** 99

notations of modality: E, F, D, L, M,

and I stand for the use of event, frame 1

, depth, LiDAR, Prior-map and IMU, respectively. . . . . 11 1.2 Event-Based datasets for SLAM benchmark across various scenarios and Unmanned aerial vehicle. UGV: Unmanned ground vehicle. 23 2.1

Summary of the data sequences in this **Section** 3

. . . . . 49 2.2

Accuracy Comparison of Our Mono-EIO **with Other** Event+IMU **Works in DAVIS240c Dataset** 1

[144].

Unit:%, 0.39 **means the average error would be 0.21m for 100m motion; Aligning 5 seconds [0-5s] of the estimated trajectory with the ground truth; The notations E, F, and I stand for the use of event, frame, and IMU, respectively** 1

2

. . . . . 91 4.1 Accuracy Comparison of Our Event-based Hybrid Pose Tracking

with Other EIO/EVIO Works in DAVIS240c Dataset 1

[144].

Unit:%, 0.21 means the average error would be 0.21m for 100m motion; Aligning 5 seconds [0-5s] of the estimated trajectory with the ground truth; The notations E, F, and I stand for the use of event, frame, and IMU, respectively 1

. . . . . 105 4.2 Accuracy Comparison of Our Event-based Hybrid Pose Tracking

with Other Image-based or Event-based Methods 1

.

Unit: MPE(%) / MRE(deg/m); Aligning the whole ground truth trajectory with estimated poses; The notations E, F, and I stand for the use of event, frame, and IMU 1

, respec- tively. . . . . 106 4.3 Accuracy Comparison of Our event-based hybrid pose tracking with image-based VIO on Stereo HKU-Dataset.

Unit:%, 0.17 **means the** aver- age **error would be 0**.17m **for 100m motion; Aligning** 1

the whole ground truth trajectory with estimated poses 1

. . . . . 107 4.4 Accuracy comparison of various combinations within our event-based hybrid pose tracking framework, including feature-based, direct-based, and our hybrid (feature-based + direct-based) tracking pipeline.

Unit:%, 0.10 **means the average error would be 0**.10m **for 100m motion; Aligning** 1

the whole ground truth trajectory with estimated poses 1

. . . . . 109 4.5 Accuracy Comparison of various combinations within EVI-SAM frame- work, including Event+IMU, Image+IMU, Event+Image+IMU, direct- based, and our EVI-SAM.

Unit:%, 0.115 **means the average error would be 0**.115m **for 100m motion; Aligning** 1

the whole ground truth trajectory with estimated poses 1

. . . . . 111 5.1

Accuracy comparison [MPE(%)] of our DEIO with other monocular event-based baselines in DAVIS240c dataset 5

[144].

The estimated trajectory is aligned with the ground truth over the 49

first 5 seconds. 129 5.2

Accuracy comparison [MPE(%)] of our dEIO with other image/event-based baselines in Mono-HKU dataset 2

[2].

The estimated trajectory is aligned with the ground truth over the 49

first 5 seconds. 130 5.3

Accuracy comparison [MPE(%)] of our DEIO with other image/event-based baselines in Stereo-HKU dataset 2

[73]. The entire sequence of es- timated poses

is aligned with the ground truth trajectory. The 33

baseline results (DPVO, DEVO) are taken from [84], and the results of Kai et al. are taken from [83]. 131 5.4

Accuracy comparison [MPE(%)] of our DEIO with other image/event-based baselines in VECtor dataset 2

[157]. The entire sequence of esti- mated poses is aligned with the ground truth trajectory. . . . . 132 5.5 Accuracy comparison [ATE/RMSE (cm)] of our DEIO with other event- based baselines in TUM-VIE dataset [156]. The entire sequence of esti- mated poses

is aligned with the ground truth trajectory. The 33

baseline re- sults (EVO, ESVO, and ES-PTAM) are taken from [88], while DH-PTAM and Ultimate-SLAM are sourced from [94], DEVO is from [84], and ES- VIO_AA, ESVO2 are from [101]. . . . . 133 5.6 Accuracy comparison [ATE/RMSE (cm)] of our DEIO with other image/event- based baselines in EDS dataset [67]. The entire sequence of estimated poses

is aligned with the ground truth trajectory. The 33

baseline results (ORB-SLAM3, DPVO, and DEVO) are taken from [84], while RAMP-VO is sourced from [85]. 134 5.7

Accuracy comparison [MPE (%)] of our DEIO with other image/event-based baselines in MVSEC dataset 2

[147]. The entire sequence of esti- mated poses

is aligned with the ground truth trajectory. The 33

DEVO re- sult is taken from [84]. 135 5.8

Accuracy comparison [MPE (%)] **of our** DEIO **with other image/event-based** baselines **in UZH-FPV dataset** 2

[15]. The entire sequence of esti- mated poses

is aligned with the ground truth trajectory. The 33

baseline results (DPVO, DEVO) are taken from [84]. . . . . 137 5.9 Accuracy comparison [ATE/RMSE (cm)] of our DEIO with stereo event- based methods in DSEC dataset [153]. The entire sequence of estimated poses

is aligned with the ground truth trajectory. The 33

baseline results (ESVO and ESVIO_AA) are taken from [100], and ES-PTAM is sourced from [88]. . . . . 137 5.10 Accuracy Comparison [MPE(%)] of Our DEIO Training on Synthetic/Real- world Datasets in Vector [157] . . . . . 145 5.11 Accuracy Comparison [MPE(%)] of DEIO with Different Event Repre- sentations in DAVIS240c Dataset [144] . . . . . 148 5.12 Accuracy comparison [MPE(%)] of DEIO with different numbers of event patches in DAVIS240c dataset [144]. . . . . 148 5.13 Accuracy Comparison of DEIO+ in DAVIS240c Dataset [144]. Unit: MPE(%), 0.06

means the average error would be 0.06m **for 100m motion. Aligning 5 seconds [0-5s] of the estimated trajectory with the ground truth** 1

. . . . . 151 5.14 Accuracy Comparison of DEIO+ in UZH-FPV [15]. Unit: MPE(%). Align- ing

the whole ground truth trajectory with estimated poses 1

. . . . . 152 6.1

Quantitative comparison of mapping performance **in terms of mean depth error (m) and the density percentage of the recovery (%), which is** assessed **against the dense ground truth depth** 1

. . . . . 174 6.2

The average time consumption of different modules in our EVI-SAM 1

. . 181 A.1 Summary of the data sequences of Monocular HKU-Dataset . . . . . 192 1 Chapter

1 Introduction 1.1 Event Camera **for Computer Vision and Robotics** 1.**1.1** 9

Why We Need Event Camera?

The field of robotics has experienced significant advancements **in recent years and has** gradually **become ingrained** 32

globally. Autonomous systems, including self-driving ve- hicles, unmanned aerial vehicles (UAVs), embodied artificial intelligence (EAI), and other autonomous technologies, are increasingly deployed to address

> **a wide range of** complex tasks **in our daily lives** 131

. A key research goal is achieving full autonomy for these systems. To effectively execute real-world tasks, these systems must be able to precisely perceive and interpret uncertainties in the surrounding environment, encom- passing both six-

> **degree-of-freedom (6 DoF) pose and** three-dimensional (**3D** 128

) spatial information, through different sensors.

> **Simultaneous Localization and Mapping (SLAM), which** concurrently tracks the **camera pose** (state estimation) **and** 102

incrementally constructs a map of an unknown environment (depth estimation, or mapping), is the most fundamental and essential component for robotics. Different sensors, such as proprioceptive sensors, e.g., wheel odometry or inertial measurement unit (IMU), and exteroceptive sensors, e.g., cam- eras, depth cameras, radars,

> **light detection and ranging (LiDAR**), have been deeply **integrated into** SLAM **systems to** 98

provide the necessary information for inferring

> **ego- motion and** the **surrounding structure of the environment** 32

. Among these sensors, vi- sion stands as a cornerstone of robotic perception, offering a wealth of information 2 Chapter 1. Introduction about the world. Standard image sensors are the most widely used due to their small size, lightweight design, inexpensive, passive, and ability to provide rich environmen- tal information. Inspired by biological heuristics, where over

> **60 % of human brain is** dedicated to **vision-related tasks** 31

, therefore, visual perception is also regarded as the most critical sensor in the domain of robotics. Recent breakthroughs, driven by the advent of deep learning, have propelled vision-based perception to new heights. (a) Motion blur (b) HDR situation Figure 1.1: The typical challenges in standard image: (a) motion blur caused by rapid movement; (b) Over exposures (HDR) under strong sunlight. Despite decades of advancements in vision-based perception and computer vision, achieving reliable and accurate state estimation and 3D reconstruction under adverse conditions remains challenging, particularly when relying on image-based solutions such as RGB or RGB-D cameras. Image-based visual SLAM systems are often hindered by several inherent limitations: First, they are susceptible to motion blur during rapid brightness changes or fast camera or object movement, as illustrated in Fig. 1.1(a). Sec- ond, the limited dynamic range of standard cameras makes it difficult to accurately capture intensity information in environments with extreme lighting conditions, such as overly bright or dark settings. This limitation frequently causes the visual SLAM sys- tem failures in illumination-challenging environments, as depicted in Fig. 1.1(b). Third, the frame rate of conventional cameras is typically capped at 30 Hz or 60 Hz, which re- stricts the minimum achievable latency in perception. Moreover, higher frame rates generate excessive redundant data, increasing the computational and power demands of the system. More importantly, the fixed frame rate of standard cameras leads to tem- poral discretization of the scene, which makes the system "blind" in the time between frames and potentially leads to tracking loss.

> **In contrast, event cameras**, offering **high temporal** resolutions **and high dynamic** 98

ranges (HDR), provide a novel perspective to address these challenges in visual SLAM,

> **paving the way for more robust and efficient** 1.1. **Event** Camera for Computer **Vision** 12

and Robotics solutions. 1.1.2 Working Principle of Event Cameras Event camera, also

> **known as Dynamic Vision Sensors (DVS**), is **a bio-inspired** sen- sor **that** 51

only captures

> **pixel-wise intensity differences and** outputs **them as** an asyn- chronous **stream rather than** a whole-**scene intensity image** in **frame** 1

format. It consti- tutes a new paradigm shift, operating asynchronously and transmitting only the information related to brightness changes at individual pixels. These pixel-level changes occur due to variations in scene illumination (such as flickering lights) or

> **the relative motion between the camera and the** scene (including dynamic **objects). An** 45

event can be triggered

> **when a large enough intensity change exceeds the pre-defined** contrast **threshold** Tthreshold, **it** 1

> **can be represented as the spatio-temporal coordinates of the** in- tensity **change and its sign: e = {t**, u, v, **p**} ⇔ L(u, v, **t +** △**t**) − L(u, v, **t) = p** · Tthreshold (**1** 5

.1) Figure 1.2: Generative Event Model. The

> **red and blue** circles **represent** the **positive events** 31

for brightness increments and negative events for brightness decrements, re- spectively. where

> **L(u**, v, **t) = log(I(u**, v, **t)) is the logarithmic** mapping of image **brightness** 9

.

> **t is the timestamp that the intensity of** the **pixel (u, v) changes, and p is the polarity that indicates the direction of the intensity change** 1

. As shown in Fig. 1.2, the event camera fires an event when the logarithmic brightness change surpasses the threshold Tthreshold, generating an ON (positive) event for brightness increments or an OFF (negative) event for decrements. An example for the visualization of the data stream from the event camera, in the spatial (x and y) dimension and temporal (t) dimension,

> **is shown in Fig**. 1.3. **Figure** 1.3: **The** visualization **of the** raw **event** 11

stream and two event representations as samples: event accumulated image (event mat) [1] and the time surface with polar- ity [2].

> **The output of event cameras**, referred **to** as **the "event stream** 32

," typically consists of grouped events rather than individual events. Since a single event data typically provides limited information (only the tuple as defined in Eq. (1.1)), the event-based vision community commonly aggregates or groups events for processing, avoiding asynchronous handling of individual events. As illustrated in Fig. 1.4, two fundamen- tal methods for generating event streams: "slice by constant event count" and "slice by constant time duration". In the figure, the green dots represent individual events, while the blue downward arrow delineates the boundaries of the event slices for the current event streams. The former case in Fig. 1.4(a) creates a new event stream once K indi- vidual events are accumulated. In contrast, the latter strategy depicted in Fig. 1.4(b) generates and publishes event streams at fixed time intervals τ.

> **1.1. Event** Camera **for Computer Vision and Robotics** (a) Constant **event** 9

number (K=4) (b) Constant duration (d = τ) Figure 1.4: Event slicing / grouping strategies [3]: (a) constant event count; (b) constant duration; green dots represent individual events, blue arrows are the boundaries of the event slices. 1.1.3 Advantages and Challenges

> **of Event Cameras Event cameras offer** several **advantages over standard cameras** 57

: (i)

> **Low latency. Event cameras** capture **brightness changes with** exceptionally **high temporal** 106

resolution, achiev- ing minimal latency on the order of microseconds. In contrast, the

> **standard cameras typically** operate at **30 HZ**, resulting in **a** 31

temporal discretization of up to 33 millisec- onds. (ii) HDR. Event cameras exhibit an impressive

> **dynamic range of up to** 140 **dB**, far exceeding the **60 dB** 68

range of standard cameras. This is enabled by their indepen- dent pixel operation, where each pixel responds logarithmically to brightness changes, allowing the capture of a wide spectrum of illumination levels. (iii) Non-redundant data / low bandwidth. By reporting only sparse and asynchronous pixel-level bright- ness changes, event cameras eliminate the temporal and spatial redundancy inherent in standard cameras. They produce no output in the absence of relative motion or illu- mination changes, whereas standard cameras keep capturing the entire scene at a fixed frame rate. (iv)

> **Low power** consumption. **Event cameras transmit only** 57

non-redundant data, significantly reducing power consumption during acquisition and transmission.

> **At the** hardware **level, most event cameras consume around 10 mW** 51

, with some pro- totypes achieving even lower power usage. In embedded

> **systems where the sensor is directly connected to a processor** 51

, the total power consumption can be as low as 100 mW or less [4]. While standard cameras typically require several watts to operate. (v) Immunity to motion blur. Standard cameras, including both global shutter and rolling shutter cameras, synchronously integrate light intensity over a fixed exposure time, leading to motion blur during rapid camera or object movement. In contrast, event cameras asynchronously and independently capture brightness changes without global exposure, making them inherently robust to motion blur. These characteristics make event cameras highly promising

> **for robotics and computer vision** applications, particularly **in challenging scenarios** where **traditional cameras** struggle, **such as low- latency, high-speed, and** 57

HDR environments. Despite their promising properties in challenging situations,

> **adopting the event camera into the SLAM** systems **is** still **a** difficult **task. This is caused by the fact that the** output of the **event** 3

camera (event stream) is

> **composed of asynchronous events, which are fundamentally different from the synchronous intensity images** 2

. As a result, con- ventional computer

> **vision algorithms cannot directly process** event **data. Event** stream **is** 105

sparse, irregular, and asynchronous, with each event conveying limited information and containing inherent noise. These unique characteristics demand the development of new paradigms to effectively interpret and utilize their potential, rather than simply adapting existing frameworks designed for image-based systems. Another layer of complexity arises from

> **the motion-dependent nature of event** camera outputs. **The** same scene can **generate** distinct **event** 20

data depending on the di- rection and magnitude of motion, as illustrated in Fig. 1.5. This variability complicates tasks such as data association and correspondence. Additionally, in situations with limited relative movement

> **between the event camera and the scene, such as static** 6

con- ditions, event cameras may generate inadequate usable data or predominantly output noise. To address these limitations,

> **event cameras are often** paired **with** complemen- tary **sensors, such as** IMU **or** standard **cameras** 121

, to ensure robust SLAM performance rather than solely relying on event data. (a) Image

> **frame (b) Left-right motion (c) Up-down motion (d) Diagonal motion Figure 1**.5: **Moving a checkerboard (a) in front of an event camera in different** direc- tions [5]. **(b)-(d) shows visualizations of the events obtained by binning events over a short time interval. Pixels that do not change intensity are represented in gray, whereas pixels that** increase **or** decrease **intensity are represented in bright and dark**, respec- tively. **Clearly, (b) (only vertical edges),** 9

(c) (only horizontal edges), and (d) can hardly be related to each other in the absence of underlying photometric information (a

).

1.1. Event Camera **for Computer Vision and Robotics** Additionally, **event** 9

cameras exhibit exceptionally low latency, capable of capturing scenes at rates of millions of events per second or higher, with the event rate dynam- ically influenced by both camera motion and scene characteristics. While this enables high temporal resolution, it also results in substantial data throughput, imposing sig- nificant computational and temporal burdens on the system. Such demands are par- ticularly pronounced for algorithms that process event streams on an event-by-event basis, often resulting in computational redundancy and inefficiency. Furthermore, event cameras are relatively new sensors, primarily developed for research purposes and commercially available only since 2008. Their optical charac- teristics are not yet fully optimized. They are susceptible to non-idealities

such as shot noise from photons **and transistor circuit noise** 104

.

Compared to standard cameras, current **event cameras** typically **have** lower spatial **resolution** 67

and higher noise levels. They are also prone to interference from infrared light, such as emissions from LiDAR or motion capture systems, often requiring the use of filter lenses to address this issue [2, 6]. Effec- tive use of event cameras requires careful adjustment of parameters such as focal length and aperture, demanding a certain level of user expertise. Incorrectly setting sensitivity parameters, e.g., the bias_sensitivity parameter (also referred to as the contrast thresh- old Tthreshold in Eq. (1.1)), can lead to excessive noise in the event camera outputs. This issue is evident in datasets like M2DGR [7], where improperly configuration results in event streams overwhelmed by noise, as illustrated in Fig. 1.6. (a) Event data from ECMD dataset (b) Event data from M2DGR dataset Figure 1.6: The event data from (a) ECMD dataset [6] (clean / normal conditions) and (b) M2DGR dataset [7] (affected by noise due to improper operation or configuration). Nonetheless,

in recent years, event cameras **have attracted a lot of attention from academia and industry**, giving rise to **event** 112

tracking, optical

flow, etc.) to high-level vision (segmentation, **reconstruction** 57

, recog- nition, etc). While this thesis specifically focuses on using event cameras for SLAM. For those interested in

a broader perspective on event cameras and their applications 32

beyond SLAM,

the survey paper **of Gallego et al**. [4] offers **a** comprehensive **overview** 74

. 1.2 Related Works

In this section, we first provide an overview of the 118

image-based visual SLAM. Af- ter that, we discuss

the state of the art (SOTA) **and related** works **of event cameras for** 31

SLAM (including both pose tracking and mapping), and

discuss their strengths and weaknesses. Finally, we introduce **the** 31

event-based dataset for SLAM, utilized for benchmarking the performance. 1.2.1 Visual SLAM Conventional visual SLAM methods are typically categorized into two primary cate- gories: direct and indirect methods. Indirect methods [8, 9], also referred to as feature- based methods, begin by detecting distinctive feature points in an image and subse- quently matching them using feature descriptors. Pose optimization is achieved by matching these features across frames and minimizing the re-projection / geometric error. On the other hand, direct methods [10, 11] operate directly on raw pixel intensity values, formulating a photometric loss function for pose optimization. The feature-based methods have played a dominant role due to the robustness provided by salient visual features, making them

well-suited for irregular scene vari- ations **and large inter-frame motions** 1

. VINS-Mono [8] is a real-time VIO framework that employs an optimization-based sliding window approach to deliver highly accu- rate and robust odometry. It

features efficient IMU pre-integration with bias correction, automatic 100
estimator **initialization, online extrinsic calibration, failure detection and** re- covery, **loop detection**

, global pose graph optimization, map merge, pose graph reuse, online temporal calibration, and rolling shutter support. Similarly, ORB-SLAM [12, 13, 1.2. Related Works 9] is a keyframe-based SLAM system that leverages

efficient binary ORB features uni- formly **for tracking, mapping, and loop detection** 48

. Openvins [14], a filter-based VIO, fuses IMU data with sparse visual feature tracks to achieve localization and mapping. It demonstrates robustness and high accuracy in challenging environments, such as fast flying drone [15]. In contrast, the direct methods show better resilience in low-textured scenes, as they leverage pixels with minimal intensity variations in their formulation. Sub-pixel alignment in these methods often results in higher accuracy. The pioneering work by Engel, J., et al. [16] introduces the first direct-based methods, incorporating

semi- dense epipolar tracking and incremental depth filtering to 48

enable real-time operation on CPUs. Subsequently, LSD-SLAM [17] extends it by incorporating

loop-closure and large-scale semi-dense mapping 48

. SVO [18] proposes the first semi-direct method, com- bining direct and feature-based techniques to estimate camera pose through direct sparse image alignment and geometric bundle adjustment. DSO [10] further advanced the field by relying on sparse structure and local photometric bundle adjustment opti- mization, significantly improving performance in low-textured scenes. DM-VIO [19], an extension of DSO, proposes a monocular VIO based on delayed marginalization and pose graph photometric bundle adjustment. EnVIO [11] is a photometric VIO frame- work that integrates stochastic gradient descent with uncertainty-aware ensembles, ef- fectively addressing the highly non-convex nature of the photometric error.

Despite significant advances, **current** visual **SLAM systems** still **lack the** 53
robust- ness requirements **for many real-world applications. Failures** can

manifest in numerous ways, such as lost feature tracks, divergence in the optimization algorithm, and the ac- cumulation of drift

. Recently, deep learning methods have gained popularity in visual SLAM. Previous research has explored

replacing hand-crafted features with learned features 53

[20, 21], or using neural radiance field (NeRF) representations [22], Gaussian Splatting [23, 24] as a map representation. Other studies have attempted to develop SLAM or VO systems in an end-to-end fashion [25, 26]. Recently, DUSt3R [27] and its derivatives like MASt3R [28], Mast3R-SLAM [29] have introduced a novel paradigm in visual SLAM, which shifts away from the traditional approach relying on feature matching and geometric optimization. Instead, they propose a versatile framework capable of addressing a wide range of 3D geometric vision tasks. Given a set of un- calibrated image pairs, a feed-forward network predicts a collection of 3D pointmaps in a unified coordinate system. Through subsequent post-processing, this framework enables camera intrinsic calibration, depth estimation, pixel matching, camera pose es- timation, and dense 3D point cloud reconstruction, among other 3D geometric vision challenges. Nonetheless, we still believe that robust visual SLAM will not be possible until further breakthroughs in several key areas, such as response to rapid motion with low latency, and handling extreme lighting conditions. This is also precisely why an increasing number of researchers have turned their attention to event-based SLAM in recent years. 1.2.2 Pose Estimation using Event Cameras Similar to the image-based approaches,

event-based pose tracking can also be divided into two main categories: (i) Feature-based, or indirect methods 1

[49, 72, 73], which

extract a sparse set of repeatable features / keypoints from the event stream, and then estimate the pose and scene geometry by minimizing re- projection errors based on these feature associations. (ii) Direct-based methods 1

[40, 59, 67, 46, 50], which utilize the information from each event, rather than relying only on those that meet a specific feature defini- tion. These approaches directly align event data with depth maps, standard images, or other event representations, without requiring explicit data association. Additionally, some related works in event-based pose tracking utilize learning-based methods [84, 102] or motion-compensation strategies [3, 47].

This section provides an overview of related works in event-based 74

pose tracking (shown in Table 1.1), aiming to offer a com- prehensive perspective on this field. 1.2.

2.1 Feature-based Methods Feature-based manners compress the 110

event data into a few informative primitives, which enables the reduction of computational resources. Point Feature:

The first feature-based 6-DoF event-based pose tracking is intro- duced in 1

[37], which firstly detects the features in the image frame

and then tracks asynchronously using the event stream. The features are subsequently fed 31

into the SVO [18]

framework to estimate the 6-DOF motion. Zhu et al. [43] propose the 67

first Table 1.1: Summary of the Event-based SLAM. The notations of modality: E, F, D, L, M,

and I stand for the use of event, frame 1

, depth, LiDAR, Prior-map and IMU, respectively. Year Methods Modality Type Event Representation Motion Remarks 2008 Kim [30] 2012 Weikersdorfer [31] 2013 Weikersdorfer [32] 2014 Censi [33] 2014 Weikersdorfer [34] 2014 Mueggler [35] 2015 Gallego [36] 2016 Kueng [37] 2016 Kim [38] 2016 Yuan [39] 2016 EVO [40] 2017 Gallego [42] 2017 Zhu [43] 2017 Rebecq [44] 2017 Reinbacher [45] 2017 Gallego [46] 2018 Gallego [47] 2018 Mueggler [48] 2018 Ultimate-SLAM [49] 2019 Bryner [50] 2019 Zhu [51] 2019 Zhu [52] 2020 Ye [53] 2020 IDOL [55] 2020 Liu [56] 2020 Peng [57, 58] 2021 ESVO [59] 2021 Hadviger [60] 2021 Kim [61] 2021 Liu [62] 2022 Liu [63] 2022 Wang [64] 2022 Mono-EIO [2] 2022 Dai [65] 2022 EKLT-VIO [66] 2022 EDS [67] 2022 Zuo [68] 2022 Chamorro [69] 2023 Chamorro [70] 2023 Liu [71] 2023 PL-EVIO [72] 2023 ESVIO [73] 2023 Liu [74] 2023 Wang [75] 2023 MC-VEO [76] 2023 El [77] 2023 Lee [80] 2023 Safa [81] 2023 Huang [82] 2024 Tang [83] 2024 EVI-SAM [1] 2024 CMax-SLAM [3] 2024 DEVO [84] 2024 Rampvo [85] 2024 Guo [86] 2024 Guo [87] 2024 ES-PTAM [88] 2024 Wang [89] 2024 AsynEIO [90] 2024 Li [91] 2024 Zuo [92] 2024 EVIT [93] 2024 DH-PTAM [94] 2024 Wang [95] 2024 FAST-LIEO [96] 2025 Wang [98] 2025 Xu [99] 2025 ESVO2 [100, 101] 2025 DEIO [102] 2025 Choi [103] 2025 EVLoc [104] 2025 SuperEIO [106] 2025 SuperEvent [107] E Direct E Feature E Feature E+F+D Direct E+D Direct E Feature E Direct E+F Feature E Direct E+I+M Feature E Direct E Motion Compensation

E+I Feature E+I Feature E Direct E 9

+M Direct E Motion Compensation

E+I Feature E+F+I Feature E+M Direct E 9

Feature E Learning E Learning E+I Feature E Motion Compensation E Motion Compensation E Direct E Feature E Motion Compensation E Feature E Feature E Motion Compensation

E+I Feature E+I Feature E+F+I Feature E+F Direct E+D Direct E 35

Feature

E+I Feature E+F+I Feature E+F+I Feature E+F+I Feature E+I Direct E Feature E+F Direct E Direct E+F+I 35

Feature E+L Learning E Direct

E+I Feature E+F+I Hybrid E Motion Compensation E Learning E+F Learning E 35

Direct E Direct E Direct E+F Feature

E+I Feature E+I Feature E+D Direct E+I+M Direct E+F 9

Learning E+I Feature L+E+F+I Direct E+I Direct E+D Direct

E+I Direct E+I Learning E+F+I Feature E+L Direct E+I Learning E+I 35

Learning

Individual Event Individual Event Individual Event Event Packet Individual Event Event Packet Individual Event Individual Event 35

Individual Events **Event** Frame **Event Frame Individual** Events **Event** Packet Event Frames **Individual Event** Individual **Event** Individual Events Individual **Event Event Frame Event Frame** Event Frame Event **Voxel** Grids **Event Frame** Individual **Event** Event **Frame Event Frame** TS TS **Individual** Events **Individual** Events **Individual**

events Individual events TS TS Individual events Event Frame TS Individual Event Individual Event Individual events TS TS TS Individual events Event Frame TS Individual events Individual events Individual events TS TS Individual events Event Voxel Grids Event Voxel Grids

Individual Event Individual **Event** Event **Frame Individual events** 38

Individual events Individual events TS TS Event Frame Individual events TS TS TS TS Event Voxel Grids Event Frame Event Frame TS TS Rotation Planar Planar 6DoF 6DoF 6DoF

6DoF 6DoF 6DoF 6DoF 6DoF Rotation **6DoF 6DoF** Rotation **6DoF** Rotation **6DoF** 6DoF **6DoF 6DoF 6DoF 6DoF 6DoF** Rotation Rotation **6DoF 6DoF** Rotation Rotation 6DoF **Planar 6DoF 6DoF 6DoF 6DoF 6DoF 6DoF 6DoF 6DoF 6DoF 6DoF 6DoF 6DoF 6DoF 6DoF** 35

6DoF 6DoF Rotation **6DoF 6DoF** Rotation **6DoF 6DoF** Rotation Rotation **6DoF 6DoF 6DoF** 6DoF **6DoF 6DoF 6DoF 6DoF 6DoF 6DoF 6DoF** 35

6DoF 6DoF 6DoF 6DOF 6DoF 6DoF Bayesian filters and Panoramic First filter-based Panoramic map Filter-based Depth sensor Line feature EGM, prior-map First feature-based Three probabilistic filters Line feature Geometric alignment and EMVS [41] Contrast Maximization First filter-based EIO First Optimization-based EIO Panoramic map Prior-map, optimization-based Contrast Maximization Continuous-time First Optimization-based EVIO Prior-map, filter-based First in-vehicle VO First learning-based event VO Extension of SfMLearner [54] Line feature Contrast Maximization Contrast Maximization First stereo event-based VO Feature-based ESVO Contrast Maximization Spatiotemporal registration Continuous-time Contrast Maximization First EIO with loop closure Optimization-based EIO Event and Image tracker [5] EGM ESVO [59]+depth camera Line feature Line feature Continuous-time Point and line feature First stereo-EIO and stereo-EVIO ESVO [59]+IMU Continuous-time EDS [67]+Contrast Maximization ESVO [59]+stereo EMVS [78, 79] Filter-based EIO Radar, Spiking Neural Network Event-to-TS registration Filter-based EIO First Feature + Direct, dense mappping Contrast Maximization Extension of DPVO [26] Extension of DPVO [26] Panoramic Panoramic EVO [40] + stereo EMVS [78, 79] Continuous-time Continuous-time Continuous-time ESVO [59]+depth camera Prior-map, optimization-based Event Superpoint [20] Continuous-time Extension of FAST-LIVO [97] ESVO [59]+IMU EVO [40]+depth camera ESVO [59]+IMU First learning-based EIO Point and line feature LiDAR+RAFT [105] Event Superpoint [20], SuperGlue [21] Event Superpoint [20], SuperGlue [21] feature-based event-inertial odometry (EIO)

which fuses events with IMU through the Extended Kalman Filter 1

(EKF). However, both of them are more suited to offline appli- cations due to their reliance on complex front-end processes that impose a significant computational burden. The

nonlinear optimization employing feature-based meth- ods, **EIO (event and IMU odometry) and EVIO (event, image, IMU odometry**), are presented **in** [44] **and Ultimate-SLAM** 1

[49]. These approaches utilize image-like event frames, enabling the application of

traditional image-based feature detection [108] **and** optical flow **tracking** [109]. **The tracked features are** subsequently fused **with IMU** 34

pre- integration to perform real-time state estimation.

To some extent, these methods use the edge 4

-like representations of event data to achieve VIO. They often require extensive pa- rameter tuning due to the varying number of events generated across different scenes and are highly sensitive to noise. EKLT-VIO [66] extends

the event-based and image- based feature tracker [5] **as the front end with** an EKF **back end** 1

to perform the pose estimation for Mars-like sequences. Kai et al. [83] introduce an adaptive decay-based time surface to extract features and propose a polarity-aware strategy to enhance the robustness. These tracked features are then fused with IMU data using a filter-based approach, specifically

the Multi-State Constraint Kalman Filter (MSCKF), to 102

achieve camera pose tracking. A monocular feature-based EIO, as presented in Mono-EIO [2], employs

the event-corner features with IMU measurement to deliver **real-time** 5

and ac- curate 6-DoF state estimation. What's more,

this EIO framework can initialize **from unknown states and** maintain **global consistency** through **loop closure** 5

of the event- based representation.

To the best of our knowledge, this work **is the first to** employ the **event-based** 1

loop closure in event-based pose tracking. Dai et al. [65] extract fea- tures

from events-only data and associate **them with a spatio-temporal locality scheme based on exponential decay** 84

. Lee et al. [80] fuses images and events using an 8-DoF warping model for accurate feature tracking [110]. Additionally, it employs a multi- state Kalman filter to consider the uncertainty of front-end features and integrates IMU data for pose estimation. ESVIO [73] proposes

the first stereo EIO and EVIO framework to estimate states through temporally and spatially event-corner feature association 1

. Line Feature: Event cameras are particularly well-suited

for line-based motion es- timation because **they** primarily **respond to edges** 120

, which are formed by strong gradi- ents. In addition, man-made environments often feature regular geometric shapes and straight lines, utilizing line-based features also holds great potential. Thus, the line fea- ture is also a very popular solution for event pose tracking [39]. IDOL [55]

calculates the normal vectors in the spatio-temporal space for each incoming event by utilizing a local neighborhood. Events with similar 2

normal vectors are clustered together to form lines, and an EIO algorithm uses these detected line features **and** IMU **measurements to** esti- mate **camera poses. However, this approach**

relies on the assumption

that lines move at nearly constant speeds over **short** time **intervals** 2

. As a result, their validation exper- iments avoid aggressive motion, which inadvertently sacrifices the key advantages of event cameras. Chamorro et al. [69, 70] employ

the Hough transformation on spatial images generated **from a 3D point-based map to cluster event data into a collection of 3D lines. These lines are subsequently integrated into the Kalman filter to estimate the 3D lines and camera pose. However, their event-to-line matching method suffers from the sudden surge of incoming events** [111] **caused by aggressive motion, scene** com- plexity, **and sudden illumination changes. Additionally, this approach is sensitive to event sparsity and requires at least 6 non-parallel 3D lines, a known-scale predefined marker, or ground-truth pose for** 2

system bootstrapping. Gao et al. [112] introduce

a 5-point minimal solver for estimating **camera relative motion** 38

using event-based line features. However, this solver has notable limitations: it relies on a non-minimal line representation with four degrees of freedom, which can introduce singularities when dealing with parallel lines. Additionally, it uses a polynomial solver that is restricted to handling only five events at a time, making it computationally expensive and unstable. In the subsequent work [113], the authors further improve their approach by utiliz- ing the angle-axis representation for event-based line features.

PL-EVIO [72] **integrates event-based point and line features to perform robust** and accurate **pose** 1

estimation. Its reliability is demonstrated by its ability to provide real-time onboard pose feedback, enabling a quadrotor to perform aggressive flip maneuvers with precision. Choi et al. [103] extract line and point features from motion-compensated event frames and then fuse these features with IMU and image data using an MSCKF backend. Continuous-time: To unleash

the asynchronous nature of event cameras 38

, continuous- time estimation methods have been developed to enable seamless state inference over time. Mueggler et al. [48] introduce a continuous-time framework that addresses re- projection errors from asynchronous events and fuses them with IMU data, using B- spline-based methods. However, this

approach cannot achieve real-time performance **because of the** computationally intensive **optimization required to update the spline parameters upon receiving every event** 5

[44]. Chui et al. [114] use the sliding-window B- spline optimization for longer and more accurate continuous-time event feature track- ing. Liu et al. [71] cubic B-Spline-based VIO to investigate the association among image, events, and IMU. However, this approach still

can not run in real-time and needs **to** avoid **the** aggressive motion **of the** 7

testing sequence. Wang et al. [75] present event-based stereo VO to estimate a continuous-time trajectory using nonparametric Gaussian process (GP) regression. Liu et al. [63] perform the monocular

> **event-only VO by** jointly **optimizing camera poses and 3D feature positions** 1

using the nonpara- metric GP motion model. However, the computational complexity of these GP-based methods is high, making them unsuitable for real-time applications. To address this issue, a lightweight GP-based continuous-time framework is proposed in [89] to esti- mate motion trajectory from asynchronous event-driven data associations with an im- age sensor [5]. This approach includes a sliding-window optimizer for GP regression to manage computational complexity effectively. Nevertheless, this method still cannot accomplish real-time running. AsynEIO [90] and Li et al. [91] propose to employ the GP regression framework to fuse asynchronous event-based features and IMU data for continuous-time pose estimation. While AsynEIO focuses on exploring various iner- tial fusion schemes, the latter work extends [89] by incorporating IMU data into the framework. 1.2.2.2 Direct-based Methods Due to noise and the strong dependence of events on motion, current event-based features or keypoints are generally less accurate and stable compared to those de- rived from traditional image-based methods. Meanwhile, in the realm of image-based SLAM, direct methods have already outperformed feature-based approaches. These observations strongly suggest that direct methods may be more suitable for leveraging the distinctive properties of event cameras, potentially unlocking their full potential. Photometric-based or EGM: The photometric-based methods utilize additional standard images or other event representations

> **to ensure photometric consistency** be- tween **event data and images** or other **event** 20

representations. Censi et al. [33] present the first event-based direct method that utilizes additional images to align the event data with the corresponding brightness pixels for estimating 6-DOF poses. Kim et al. [38]

> **leverage photometric relationships between brightness changes and absolute** bright- ness **intensity to associate events with the corresponding pixels in the reference image** 36

, employing three decoupled probabilistic filters.

> **Based on the event generation** model (EGM), where **each event** signifies a **brightness change from the last event at the same pixel, several** studies align **event data with corresponding** brightness **pixels to estimate camera poses** 50

. Gallego et al. [36] apply

> **the linearized EGM to** derive **camera motion parameters by constructing an implicit measurement function** 36

. EDS [67] proposes

> **an event-image alignment algorithm that minimizes the photometric** errors **between the brightness change from events and the image gradients, enabling 6-DOF** 1

monocular VO. MC-VEO [76] achieves motion estimation

> **by minimizing the brightness increment errors between the** 63

motion-compensated event representation and the image frame. These EGM-based methods explicitly model the event triggering process, requiring careful consideration of the contrast threshold (as described in Eq. (1.1)) and the event generation mechanisms. Thus, they usually struggle with the complexity of accurately modeling the event generation. EVI-SAM [1] introduces

the first event-based hybrid pose tracking framework, which combines the robustness of feature-based event pose tracking with the relatively high accuracy achieved through event-based direct 1

align- ment. It develops

an event-based 2D-2D alignment to construct the photometric con- straint **and tightly integrate it with the event-based** re-projection **constraint** 1

. Geometric Registration: Geometric registration methods compute incremental cam- era poses by aligning edge-like brightness patterns conveyed by events and/or corre- sponding map pixels. EVO [40] presents a direct method

relying on edge-map (2D-3D) model alignment, utilizing the 1

2D image-like event representation **and the** 19

3D map reconstructed from EMVS [41]. It

can perform on small-scale scenes, but it is very parameter-sensitive and requires 14

running in a scene that is planar to the sensor, up to **several seconds, for bootstrapping the system** 17

. ES-PTAM [88] extends the EVO [40] into a stereo setup, estimating depth by maximizing ray density fusion [78, 79], while track- ing the camera pose by optimizing edge-map alignment. ESVO [59]

is the first stereo event-based VO method, which follows a 1

PTAM

scheme to estimate the ego motion through the 2D-3D edge registration on time surface (TS 1

).

However, it barely operates even **in real-time in** 4

low resolution event camera

and also faces limitations due to rig- orous **initialization as well as unreliable pose tracking** 2

. To tackle the degeneracy issue of ESVO, ESVO2 [100, 101] presents a direct method for stereo event camera with IMU, and further adopts the contrast maximization [47] to reduce the redundant operations. El

Moudni et al. [77] replace the mapping module of ESVO with the stereo EMVS us- ing the disparity space image fusion [78, 79]. Building upon direct-based 2D-3D edge registration from ESVO, error-state Kalman filter [98] and sliding window non-linear optimization [74] are separately proposed to integrate the IMU for pose estimation. However, these follow-up works still suffer from the inherent issues of ESVO, such as poor robustness, parameter sensitivity, and the inability to handle challenging situa- tions (e.g., aggressive motion). Their performance remains limited to levels comparable to ESVO, unable to meet the standards of

state-of-the-art event-based methods and 67

per- forming even worse than image-based solutions. Besides,

most of the aforementioned direct-based event pose tracking methods are limited to small-scale environments and small, dedicated movements. They struggle to provide reliable state estimations, **as they heavily depend on successful** 2D-3D **model alignment and the timely updates of the local 3D map** 1

. Depth, LiDAR, or Prior Map-based: In addition, several studies utilize depth cam- eras, LiDAR sensors, or prior map to provide depth information for event and map registration.

Gallego et al. [46] and **Bryner et al** 61

. [50] perform direct pose

tracking of an event camera from a given photometric depth map of **the** 1

3D scene, using a prob- abilistic EGM in Bayesian filtering [46] and a nonlinear optimization [50] framework. Building upon the framework of ESVO, Zuo et al. [68] introduce

a direct VO framework that incorporates a depth camera to supply depth information for the event camera 1

.

The camera pose is updated **through geometric 3D-2D edge alignment** 63

. Zuo et al. [92] leverage

semi-dense 3D point clouds, which are **priors obtained from depth camera- based mapping algorithms** 48

, to register

the camera poses using **semi-dense 3D-2D edge alignment** 48

. This cross-modal tracking approach avoids the limitations of relying solely on event-based maps, which often suffer from inferior quality. EVIT [93] improves the event-based geometric prior semi-dense map tracking paradigm [92] by incorporating IMU data to achieve more robust pose estimation. Xu et al. [99] propose a motion- encoded temporal surface (METS) to represent the event data.

Following a semi-dense 3D-2D alignment pipeline 61

[40, 59], they employ either EMVS [41] or a depth camera to obtain the required 3D scene structure for camera pose estimation. Fast-LEIO [96] extends the SOTA LiDAR-inertial-visual odometry framework

(FAST-LIVO [97]) to in- corporate the event modality. The EIO subsystem is achieved by the alignment between the TS and the edge depth maps derived from LiDAR data. EVLoc [104] projects the LiDAR points into 2D space and employs an optical flow network [105] to align events with these LiDAR points in 2D space. The camera pose is then estimated using a PnP solver. Panoramic-based: Simultaneous mosaicing and tracking methods [30, 45, 86, 87, 115] represent another major branch of event-based direct methods. Reinbacher et al. [45] employ direct alignment from DSO [10] for panoramic tracking and probabilis- tic mapping. In their approach, the panoramic map stores a probability value at each point, reflecting the spatial event rate—the higher the value, the more likely events are to be generated when the camera's pixels traverse that location. Starting from first prin- ciples, Guo et al. [86] develop a photometric bundle adjustment framework for a purely rotating event camera. Their method formulates event-based bundle adjustment as

a non-linear least squares optimization problem 42

, jointly

estimating camera motion and a semi-dense panoramic **map** 42

. While their earlier work [87] extends the filter-based method for mosaicing and tracking. To some extent, these panoramic or mosaicing- based pipelines simply shift the paradigm from 2D-3D geometric registration to the alignment between 2D event representations and panoramic maps. While they may employ different formulations of panoramic maps or varying loss types, they are ulti- mately limited to recovering only rotational or low degrees-of-freedom (DoF) motion. Moreover, the 'map' of these methods is not the concept of depth map in the conven- tional visual SLAM, which prevents the execution of downstream tasks that rely on 3D information. 1.2.2.3 Motion Compensation Methods

Motion compensation methods recover the motion model by warping and aligning the event data into a reference frame based on their spatio-temporal relationships. These **methods** 36

aim to appear in the form of sharp images of warped events (IWE), while the estimated motion is obtained in the form of by-products. Most of them also only focus on low DoF estimation, such as Contrast Maximization (CMax) [47, 116], and other sim- ilar event alignment frameworks (e.g., probabilistic approach [117], Entropy Minimiza- tion [118], Dispersion Minimization (DMin) [119]). Gallego et al. [47] propose a unify- ing CMax framework that aligns

point trajectories on the image plane with event data by maximizing the contrast of **an** 50

IWE. Building upon this framework, several subse- quent works [116, 61, 3, 120] have been proposed for motion estimation based on CMax framework. Kim et al. [61] develop a real-time rotational position and velocity esti- mation method that reduces drift errors by globally aligning events within the CMax framework. Liu et al. [56] and Peng [57, 58] propose globally optimal CMax methods that use

upper and lower bounds of reward functions and the branch-and-bound algo- rithm **for** 20

rotational or fronto-parallel motion estimation. CMax-SLAM [3] combines ro- tational motion bundle adjustment with CMax. These CMax-based methods also face several limitations. First, the selection of the temporal window involves a trade-off between gathering enough events for optimization and maintaining the assumption of constant angular velocity. Additionally, iterative optimization and repeated event warping operations introduce a significant computational overhead. Moreover, CMax- based methods are susceptible to local optima and degenerate solutions, known as the event collapse problem, which compromises estimation reliability. This problem is par- ticularly restrictive in cases of rotational camera motion, further limiting the applicabil- ity. A survey for event-based contrast maximization is available on our website : https: //github.com/KwanWaiPang/Awesome-Event-based-Contrast-Maximization. 1.2.2.4 Learning-based Methods As observed in the image-based counterpart, deep learning methods [26, 25] handle visually degraded conditions more effectively than traditional handcrafted data asso- ciation. Zhu et al. [52] pioneered the first learning-based event odometry framework, utilizing an unsupervised network with a

contrast maximization loss [47]. Ye et al. [53] extend the SfMLearner [54], which employs a depth network and pose network for

event-based optical flow estimation. However, **these methods** 66

show poor generaliza- tion beyond the training scenarios. DH-PTAM [94] proposes a stereo event-image par- allel tracking and mapping system, replacing hand-crafted features with learned ex- tractors and descriptors [20, 121]. DEVO [84] extends the DPVO [26] by incorporat- ing the event modality through voxel-based representation [52], demonstrating strong generalization capabilities from simulation to various real-world event-based bench- marks. This approach outperforms both classical and deep-learning baselines. RAMP- VO [85] introduces an end-to-end VO that also builds upon DPVO, using feature en- coders to fuse event and image data. It achieves robust zero-shot transfer performance

to real data despite being trained only on synthetic 44

dataset. DEIO [102] proposes a learning-optimization-combined framework that tightly-coupled integrate

trainable event-based differentiable bundle adjustment (e-DBA) **with IMU** 11

pre-integration in a patch-based co-visibility factor graph that employs keyframe-based sliding window optimization. Even though it is trained on synthetic data, it still outperforms over 20 state-of-the-art methods across 10 challenging real-world event benchmarks. Building on the image-based keypoint detection and description framework (SuperPoint [20] and SuperGlue [21]), both SuperEIO [106] and SuperEvent [107] replace hand-crafted features with learned features. SuperEIO combines feature-based data association [2] for front-end tracking with learned priors for loop closure detection, while SuperEvent leverages a comprehensive learning framework that encompasses both feature detec- tion and data association within event streams. 1.2.3 Depth Estimation using Event Cameras 1.2.3.1 Event-based Monocular Depth Estimation Non-learning: Depth estimation

with a single event camera is targeted as the **problem** 9

of “temporal stereo”. Kim et al. [38] introduce

the pioneering concept of purely event- based depth estimation **by employing three decoupled probabilistic filters** 1

. However, its computational demands are notable, as it necessitates the recovery of image inten- sity and relies on GPU acceleration to achieve real-time performance. CMax-SLAM [3] and Reinbacher et al. [45] perform panoramic mapping using a monocular event cam- era. However, their mapping concept merely represents the probability of events be- ing generated for each position, rather than depth estimation or recovery. Similarly, EROAM [115] also adopts the concept of panoramic reconstruction from [3, 45]. These conceptual shifts have constrained the applications requiring spatial or depth informa- tion. EMVS [41]

is the first work to achieve semi-dense 3D reconstruction **from a single event camera with a known trajectory without requiring any explicit data association or intensity estimation. The concept of event-based denser mapping was introduced in** [122], **building upon EMVS** [41]. **However, it is still in the conceptual stage and does not successfully reconstruct any event-based dense point cloud. EOMVS** [123] **adopts** an **omnidirectional event camera in EMVS** [41] **to** reconstruct **a wider field-of- view semi-dense** 1

depth. Gallego et al. [47] use

contrast maximization to find the best depth value that fits the event stream. These methods recover semi-dense 3D recon- structions **of** 1

scenes by integrating events from a moving camera over a time interval, and they require knowledge of camera motion

. Chiavazza et al. [124] calculate

semi- dense depth from optical flow using neuromorphic hardware to process asynchronous events. However, its applicability is confined to translational motion only 1

. EVI-SAM [1]

is the first framework that employs a non-learning approach to achieve event-based dense and textured 3D reconstruction without GPU acceleration. It 1

consistently de- livers excellent mapping results in a variety of challenging scenarios, demonstrating performance comparable to learning-based and NeRF-based mapping methods. Learning-based: Recently, deep learning approaches are popular in addressing the monocular event-based depth estimation problem. Zhu et al. [52] construct a convolu- tional neural network (

CNN) with an unsupervised encoder-decoder architecture for semi- dense **depth prediction** 40

. Chaney et al. [125] design a neural network model spe- cially

for environments with a ground plane to learn the ratio between the height of a point from the ground plane and its depth in the event camera frame 40

.

Hidalgo-Carrió et al. [126] are **the first** to estimate **dense depth** 69

maps from a monocular event camera through recurrent CNN architecture. Gehrig et al. [127] apply a recurrent neural net- work (RNN) architecture that

maintained an internal state that was updated through asynchronous **events or** irregular images **and could be** queried for dense **depth** estima- tion **at any timestamp** 40

. Liu et al. [128] utilize a transformer-based architecture to esti- mate monocular depth, incorporating spatial and temporal information from the event stream. In contrast to individual transformers for input modalities, Devulapally et al. [129] introduced a unified transformer that combines both event and RGB modalities to capture inter-modal dependencies and achieve depth prediction. EventNeRF [130], Ev-NeRF [131] E-NeRF [132], EvDNeRF [133], and Mahbub, et al. [134] present Neu- ral Radiance Field (NeRF) built from a single event camera to reconstruct the density 3D volumetric representation of the environment. However, these learning-based ap- proaches still have significant limitations to overcome, including the need for mas- sive event datasets for training and being computationally much heavier than the non- learning approaches. IncEventGS [135] and Event3dgs [136] try to exploit 3D Gaussian Splatting [23] for event-based reconstruction. However, they all failed to meet the chal- lenge requirements, such as real-time performance, handling fast motion, HDR, etc. 1.2.3.2

Event-based Stereo **Depth Estimation** Non-learning: **The majority of works on depth estimation with event cameras target the** problem **of** 62

“instantaneous stereo

” using data from a pair of synchronized and rigidly attached

event cameras. Most **event-based stereo depth estimation methods** use **the** 40

epipolar constraint and the assumption of temporal coincidence of events across reti- nas. Ieng et al. [137] follow

a paradigm of event matching plus triangulation to realize the 3D reconstruction 1

. Zhou et al. [138] and ESVO [59] tackle

a semi-dense reconstruc- tion **problem using a pair of temporally-synchronized event cameras in stereo** config- uration **through energy minimization methods. T-ESVO** [139] **extends the ESVO** [59] **using TSDF to reconstruct 3D environments and re**-estimates **the semi-dense depth** 1

. Ghosh et al. [79, 78]

extend the EMVS [41] **into** a **stereo setup to estimate depth by fusing back-projected ray densities. Learning-based**: Meanwhile, learning-based **methods, such as** 1

[140, 141, 142],

have been applied to stereo event-based depth estimation, where different deep networks are developed to reconstruct event-based dense maps. The first **learning-based** 1

stereo method is present in [140] for event cameras to produce dense depth results. Nam et al. [141] propose to learn an event concentration network with stereo events from both past and future to produce a compact event representation with high details for depth estimation. Ahmed et al. [142] develop a deep event stereo network that recon- structs spatial image features from embedded event data and leverages the event fea- tures using the reconstructed image features to compute dense disparity maps. Chen et al. [143] develop a stereo asymmetric frame-event camera system to estimate dense depth through frame-event stereo matching, monocular frame-based and event-based structure-from-motion (SfM), and a depth estimation network.

Although these learning- based methods can predict dense depth and reveal 3D structures with limited events, they may struggle to handle objects that are **not included in** the **training sets, thus leading to uncertainties in their generalization ability** 1

. What’s more, most of them re- quire high-quality ground truth (GT) depth for supervised training, which is not al- ways available in practice. Furthermore,

these learning-based methods only generate localized depth maps. The production of globally consistent depth maps with rich tex- ture **information, such as surface mesh, remains an** unsolved problem **in** 1

these works. 1.2.4 Datasets for Event-based SLAM Benchmarking The growing interest in event cameras has led

> **to the** creation **of** numerous **event-based** datasets **for SLAM** **110**

benchmarking, with a particular focus on those that are coupled with other sensors. Table 1.2 summarizes the differences among event-based datasets, including variations in sensor types, platforms, and scenarios. Many

> **event-based datasets, combined with various sensors, have been released for SLAM** benchmarking, **utilizing a variety of robotics platforms** **8**

, such as handheld, driv- ing, drone, quadruped, etc. DAVIS240C [144], TUM-VIE [156], VECtor [157], EDS [67], and HKU-dataset [72, 73, 1] are

> **collected by handheld** or **head-mounted devices** in different **environments** **114**

. DAVIS240C [144] is the most widely used dataset for monoc- ular event-based SLAM, which provides event and image data with a resolution of 240×180 pixels, as well as IMU data and 6-DoF ground truth (GT) poses.

> **It contains extremely fast 6-Dof motion and scenes with HDR. The** **2**

HKU-dataset comprises four main parts: monocular setup [72], stereo setup [73], event with RGB-D setup [1], and the synthetic event data (detailed in Appendix A). Most of the sequences are recorded under a broad range of illumination conditions or during aggressive motion, with the GT pose provided by VICON. Additionally, the dataset includes challenging sequences such as large-scale outdoor scenes, drone flighting, and, etc. While the synthetic event data has broad diversity, which can be used for training the learning-based event net- work. M2DGR [7]

> **utilizes ground robots to collect a multi-sensor dataset with an event camera under large-scale scenes, while the event streams exhibit large noises**. Fusion- Portable [158] **proposes multi-sensor campus-scene datasets with stereo event cameras** **8**

Table 1.2: Event-Based datasets for SLAM benchmark across various scenarios and sensor types. MCR: Motion capture system. LT: Laser tracker.

> **UAV: Unmanned aerial vehicle. UGV: Unmanned ground vehicle** **33**

. Dataset Platform Terrain Event Infrared RGB RGBD LiDAR GNSS IMU GT Pose DAVIS240C [144] Handheld Indoor Outdoor 240 × 180 × 1 ✗ ✓ ✗ ✗ ✗ ✓ MCR DDD17 [145] Driving Urban 346 × 260 × 1 ✗ ✓ ✗ ✗ ✓ ✗ ✗ N-CARS [146] Driving Urban 304 × 240 × 1 ✗ ✓ ✗ ✗ ✗ ✗ ✗ MVSEC [147] Driving UAV Suburban 346 × 260 × 2 ✗ ✓ ✓ VLP-16 ✓ ✓ GNSS LiDAR-SLAM UZH-FPV [15] UAV Indoor Outdoor 346 × 260 × 1 ✗ ✓ ✗ ✗ ✗ ✓ LT CED [148] Driving Urban 346 × 260 × 1 ✗ ✓ ✗ ✗ ✗ ✗ ✗ Rebecq, et al. [149] Driving Urban 640 × 480 × 1 ✗ ✗ ✗ ✗ ✗ ✗ ✗ DDD20 [150] Driving Urban 346 × 260 × 1 ✗ ✓ ✗ ✗ ✓ ✗ ✗ Brisbane-Event-VPR [151] Driving Suburban 640 × 480 × 1 ✗ ✓ ✗ ✗ ✗ ✗ ✗ De, et al. [152] Driving Suburban Urban 640 × 480 × 1 ✗ ✓ ✗ ✗ ✗ ✗ ✗ DSEC [153] Driving Suburban 640 × 480 × 2 ✗ ✓ ✗ VLP-16 ✓ ✓ GNSS-RTK GRIFFIN [154] UAV Indoor Outdoor 346 × 260 × 1 ✗ ✓ ✗ ✗ ✗ ✓ LT/MCR AGRI-EBV-AUTUMN [155] UGV Outdoor 240 × 180 × 1 ✗ ✓ ✓ VLP-16 ✗ ✓ LiDAR-SLAM TUM-VIE [156] Handheld Indoor Outdoor 1280 × 720 × 2 ✗ ✓ ✗ ✗ ✗ ✓ MCR M2DGR [7] UGV Indoor Outdoor 640 × 480 × 1 ✓ ✓ ✗ VLP-32 ✓ ✓ MCR, LT RTK/INS VECtor [157] Handheld Indoor 640 × 480 × 2 ✗ ✓ ✓ OS0-128 ✗ ✓ MCR EDS [67] Handheld Indoor 640 × 480 × 1 ✗ ✓ ✗ ✗ ✗ ✓ MCR FusionPortable [158] UGV Handheld Quadruped Indoor MCR, LT Outdoor 346 × 260 × 2 ✗ ✓ ✗ OS1-128 ✓ ✓ GPS-RTK Driving ViViD++ [159] UGV Urban 640 × 480 × 1 ✓ ✓ ✓ OS1-64 ✓ ✓ GNSS-RTK Handheld HKU-dataset [72, 73, 1] UAV Handheld

Indoor Outdoor, Campus 346 × 260 × 2 640 × 480 × 1 ✗ ✓ ✓ ✗ ✗ ✓ MCR M3ED [160] Driving UAV Quadruped Forest Urban 1280 × 720 × 2 ✗ ✓ ✗

OS1-64 ✓ ✓ LiDAR-SLAM GNSS-RTK MA-VIED [161] Driving **Urban** 6

640 × 480 × 1 ✗ ✓ ✗ ✗ ✓ ✓ GNSS-RTK Hadviger, et al. [162] Driving Handheld Urban Indoor 640 × 480 × 2 ✗ ✓ ✗

OS1-128 ✓ ✓ GNSS-RTK/INS ECMD [6] Driving **Suburban Urban, Dense City** 6

EAGLE [163] Quadruped Indoor Outdoor 346 × 260 × 2 640 × 480 × 2 346 × 260 × 1 320 × 240 × 1 ✓ ✓ ✗ ✗ ✓ ✓

VLP-16 Lslidar C16 HDL-32E VLP-16 ✓ ✗ ✓ ✓ **GNSS-RTK/INS** 8

LiDAR-SLAM MCR

on diverse platforms (handheld, quadruped robot, and UGV). Moreover, there exist specialized event-based datasets such as UZH-FPV [15] and GRIFFIN[154], **which are targeted for flying robots** 6

. UZH-FPV [15]

is a high-speed, aggressive VIO **dataset. This dataset includes fast laps around a racetrack with drone racing gates, as well as free- form trajectories around obstacles** 34

.

Moreover, a number of event-based datasets are published under large-scale driv- ing scenarios for computer vision. These autopilot datasets offer more realistic and challenging conditions, including high-speed scenarios, repetitive situations, and HDR scenes compared to datasets collected from handheld devices. The first dataset catering **to driving recordings using an event camera is DDD17** [145], **as well as the follow-up DDD20** [150], **for studying the end-to-end driving application incorporating diverse vehicle control data** 8

. HATS [146], CED [148], Rebecq, et al. [149], and Brisbane-Event- VPR [151] publish

their event-based datasets for the computer vision task of object classification, image reconstruction, and vision place recognition in driving scenarios. MVSEC [147] **is a pioneering cross-modal dataset with stereo event and image cameras, as well as LiDAR. However, a limitation of MVSEC** resides **in the utilization of low- resolution event cameras (346×260) with a compact baseline of 10 cm, coupled with the imprecision of the ground-truth derived from GNSS or LiDAR-SLAM. DSEC** [153] proposes **an event-based dataset whose scenarios are similar to KITTI** [164], provid- ing **higher resolution stereo event (640×480) and image, LiDAR, and IMU under** vari- ous **illumination conditions. M3ED** [160] **encompasses high-resolution event cameras (1280×720) and covers three different robotics platforms:** 6

driving, flight, and legged robot. However, both DSEC and M3ED datasets are primarily utilized for computer vision fields, such as optical **flow estimation, segmentation, and disparity estimation, rather than specifically for localization or mapping problems. Besides, they do not** pro- vide **sufficient challenges for SLAM, as the majority of these datasets** are **collected in rural or suburban areas with relatively low-lying structures, light traffic, and less** dy- namic **objects. ViViD++** [159] **focuses on diverse vision sensors for handheld and** driv- ing **platforms, including event, thermal, and standard cameras. MA-VIED** [161] pro- poses **a comprehensive driving dataset that** encompasses **race track-like loops**, maneu- vers, **and standard driving**

scenarios. However, both of these datasets exclusively offer monocular data for each camera type, thereby precluding the possibility of conducting stereo visual SLAM 6

. Hadviger, et al. [162] introduce a stereo event and standard camera dataset which includes 6-DoF sequences collected indoors with handheld rig, as well as outdoor vehicle driving sequences. For further investigating

the inquiry: Are event cameras ready for autonomous driving 6

? ECMD [6], an event-based dataset for autonomous driving, is proposed. It

provides data from two sets of stereo event cameras with dif- ferent **resolutions** 6

(640x480, 346x260),

stereo industrial cameras, an infrared camera, a top-installed mechanical LiDAR with two slanted LiDARs, two consumer-level GNSS receivers, and an onboard IMU. Meanwhile, the ground-truth of the vehicle was ob- tained **using a centimeter-level high-accuracy GNSS-RTK/INS** navigation **system** 6

. The 1.3. Thesis Outline overview of ECMD dataset can be seen in Fig. 1.7.

LiDAR Infrared IMU Event Standard Image Image GNSS-RTK/INS DAEVvleSn3t46 IMU DVXplorer Event Infrared LiDAR GNSS-INS-RTK Figure 1.7: **An overview of the sensor setup and dataset visualization** from ECMD [6]. **Above: The** sensor **suite is mounted on the top of the vehicle. Below: Sensor systems include two sets of stereo event cameras, stereo industrial cameras, an infrared camera, a top-installed LiDAR with two slanted LiDAR, IMU, and GNSS-RTK/INS systems. Each sensor is indicated with the letter box** 8

. 1.3 Thesis Outline

In this thesis, **we aim to** deeply **investigate the event** 51

camera

for 6-DoF pose tracking and 3D reconstruction. A 105

website 1 is also established to provide the demonstrations and resources for this thesis, including the source codes, related publications, datasets, video demo, and etc.

The thesis is organized as follows: • Chapter 1 provides the necessary background of the event camera and 74

the related works in the field of event-based SLAM 3

. 1https://kwanwaipang.github.io/PhDThesis • Chapter 2 introduces Mono-EIO [2], which is a monocular event-inertial odome- try

based on event-corner feature detection and matching with well-designed fea- ture management 3

, to estimate 6 DoF motion.

Event-corner features are extracted from the asynchronous raw event stream and associated with a 2

spatio-temporal event-based representation using an exponential decay kernel.

To this end, two different kinds of event representations 3

are designed for assisting uniformly dis- tributed feature detection, front-end

feature tracking (for front-end incremental estimation), and feature matching (for loop closure detection). The event tracker and matcher are 3

finally tightly-coupled integrated with the

IMU pre-integration within a keyframe sliding window graph-based optimization framework 2

. • Chapter 3 develops feature-based EVIO [72],

which leverages the complemen- tarity between event cameras and standard cameras 7

. The framework utilizes both hierarchical (point-based and

line-based event features) and heterogeneous (point-based event and image features 4

) visual features, with two variants are in- troduced:

PL-EIO (purely event-based) and PL-EVIO (event with image 4

-aided). These

features are well-designed to provide additional structure or 34

constraint in- formation that

enhances the accuracy and robustness of state estimation 93

. We also

demonstrate the effectiveness of the proposed 93

EVIO methods for quadrotor au- tonomously flying in high-speed and HDR scenarios. • Chapter 4 designs

an event-based hybrid pose tracking, which is the tracking module of 1

EVI-SAM [1]. It

employs a hybrid framework that combines feature- based and direct-based methods to process events, enabling the estimation of 6- DoF pose. A sliding window graph-based optimization framework is designed to tightly fuse the event-based geometric errors (re-projection residuals) and event- based photometric errors (relative pose residuals), along with the image-based geometric errors and the IMU pre-integration. To the 1

best of our knowledge, this is

the first hybrid approach that integrates both direct-based and feature-based methods within an event-based framework 1

. 1.3. Thesis Outline • Chapter 5 proposes DEIO [102], a monocular deep event-inertial odometry frame- work, which combines learning-based methods with traditional nonlinear graph- based optimization. Specifically, we tightly integrate a trainable event-based dif- ferentiable bundle adjustment (e-DBA) with the IMU pre-integration in a patch- based co-visibility factor graph that employs keyframe-based sliding window op- timization. Numerical Experiments on ten public challenge datasets demonstrate

that our method can achieve superior performance compared with the 5

image- based and event-based benchmarks.

To the best of our knowledge, DEIO is the first learning 1

-based EIO framework. • Chapter 6 demonstrates an event-based dense and textured 3D reconstruction system. It is the mapping module of EVI-SAM [1], which

initially reconstructs the event-based semi-dense depth using a space-sweep way through the 6-DoF pose obtained from the tracking module 1

of the EVI-SAM.

Subsequently, it integrates an aligned intensity image as guidance to reconstruct the event-based dense depth and render the texture of the map (i.e., the color of 3D points). Finally, the TSDF- based map fusion is 1

designed to generate a 3D global consistent texture map and surface mesh of the environment

.

To the best of our knowledge, this is the first framework that employs a non-learning approach to achieve event-based dense and textured 3D reconstruction without GPU acceleration 1

. • Chapter 7 concludes this thesis with the

summary of the contributions and the discussion **of** possible **future** 75

research directions. • Appendix A introduces our HKU-dataset for event-based benchmarking. Ap- pendix B details the derivation of the e-DBA network structure. Additionally, Appendix C and Appendix D introduce basic concepts for evaluating pose track- ing and mapping. Please note that although ESVIO [73], ECMD [6], and SuperEIO [106] are also the key publications in event-based vision during my PhD studies, they are not presented as individual chapters in this thesis. Space constraints have limited their inclusion to relevant discussions only, while further details can be obtained from the corresponding publications. Chapter 2 Mono-EIO: Monocular Event-Inertial Odometry

Event cameras are biologically inspired vision sensors **that capture pixel-level** illumi- nation **changes instead of the intensity image at a fixed frame rate. They offer many advantages over the standard cameras, such as high dynamic range, high temporal resolution (low latency), no motion blur, etc. Therefore, developing state estimation** al- gorithms **based on event cameras offers exciting opportunities for autonomous systems and robots. In this** chapter, **we propose monocular** event-**inertial odometry** (Mono- EIO) **for event cameras based on event-corner feature detection and matching with well-designed feature management. More specifically, two different kinds of event** rep- resentations **based on time surface are designed to realize event-corner feature** track- ing (**for front-end** incremental **estimation) and matching (for loop closure detection). Furthermore, the proposed event representations are used to set** a **mask for detecting the event-corner feature based on the raw event stream, which ensures the** uniform distribution **and spatial consistency characteristic of the event-corner feature. Finally, a tightly coupled, graph-based optimization framework is designed to obtain high- accurate state estimation through fusing pre-integrated IMU measurements and event- corner observations. We validate quantitatively the performance of our system on** dif- ferent **resolution event cameras: DAVIS240C (240*180, public dataset, achieve state-of- the-art), DAVIS346 (346*240, real-test), DVXplorer (640*480 real-test**). Furthermore, **we demonstrate qualitatively the accuracy, robustness, loop closure, and re-localization performance of our framework on different large-scale datasets, and an autonomous** 3

Chapter 2. Mono-EIO: Monocular Event-Inertial Odometry quadrotor

flight using our Event-inertial Odometry (EIO) **framework** 3

. This chapter is based on our work [2].

Videos of all the evaluations are presented on the project 3

web- site: https://kwanwaipang.github.io/Mono-EIO. 2.1

> **Introduction State estimation is the most fundamental topic in the field of robotics, such as** Simul- taneous **Localization and Mapping (SLAM) / Visual Odometry (VO), navigation, path planning, drone control, autonomous driving, etc. Recently, the approaches that assist the visual sensor (camera) with** an **inertial sensor (inertial measurement unit, IMU), also known as Visual Inertial Odometry (VIO), have gained significant research** inter- est **and progress** [9, 8]. **However, due to the inherent limitations of the standard** cam- eras, **such as motion blur and low dynamic range, the VIO systems based on standard cameras might easily fail during high-speed motions or in high-dynamic-range (HDR) scenarios. Event cameras, also called Dynamic Vision Sensors (DVS), offer a huge potential to overcome the** aforementioned **issues due to their extremely high temporal resolution and HDR property** [4]. **Event cameras constitute a new paradigm shift that operates asynchronously, transmitting** 3

> **only the information conveyed by brightness changes in individual** pixels. **Event cameras have many advantages over the standard cameras: (i) Extremely high temporal resolution and negligible latency on the order of a few** mil- liseconds; (**ii) HDR (140 dB for the event cameras, while 60 dB for standard cameras). (iii) Since all pixels only capture the brightness change asynchronously and** indepen- dently, **event cameras do not suffer from motion blur** [4], **and also remove the inherent redundancy of standard** images. **These properties allow the event cameras to confer** ro- bustness **to vision-based localization in challenging scenarios. However, adopting the event camera into the SLAM/VO is a very challenging task, this is caused by the fact that the event streams are composed of asynchronous events, which are fundamentally different from the synchronous intensity images. Thus, the traditional computer vision algorithms cannot be directly applied. Some works have addressed this challenge by reconstructing the intensity frame from the event data** [38, 165], **aggregating a group of events within a short period of time into event frame** [44, 49, 40], **or combining event** 2.1. Introduction **cameras with additional sensors (e.g. depth sensors** [34], **standard cameras** [37]). How- ever, **those would introduce bottlenecks and lose the natural advantages of the event camera. In this** 3

chapter, a monocular event-inertial odometry (Mono-EIO) is introduced for the ego-motion state estimation.

> **Different from existing methods that** reconstruct **or aggregate intensity** images **from the event data, we directly use the asynchronous raw events for feature detection (named as event-corner feature** 3

). More specifically, the proposed method extracts event-corner features from raw event stream

> **and associates them with a spatio-temporal** event-**based** representation using an **exponential decay** 84

kernel.

To this end, two different **kinds of event representations based on time surface (TS)** are designed **for assisting uniformly distributed feature** detection, **front-end** fea- ture **tracking** 3

(

for front-end incremental **estimation), and** feature **matching (for loop** clo- sure **detection** 3

).

We investigate the proposed event-corner feature tracker and matcher into monocular event-inertial odometry (EIO) **based on sliding windows graph-based optimization to estimate arbitrary 6 DoF (degree-of-freedom) motion**. The **contributions** of this chapter **are summarized as follows: 1. Instead of extracting the features on the intensity image generated from events, we propose a steady and uniformly distributed event-corner feature detector that directly works on the raw events. 2. We design two different event representations, the TS with the polarity and the normalized TS without** polarity, **to perform robust feature tracking and loop** clo- sure **matching using the previously detected event-corner features. 3. The event-corner features tracker and matcher are** integrated **into a keyframe- based visual-inertial system that tightly fuses the event-corner features with IMU data to update the state. Furthermore, our** Mono-EIO **framework can** 3

bootstrap from unknown initial states and also ensure global consistency thanks to the loop closure. 4. We evaluate the proposed method on different resolution **event cameras: DAVIS240c (240*180, publicly available dataset**, achieves **state-of-the-art), DAVIS346 (346*240, real-test), DVXplorer (640*480, real-test), and DVXplorer-Mini (640*480**, quadro- tor **flighting). It is our knowledge that this is the first** EIO **framework** that **can realize real-time performance in high resolution event cameras with the loop** clo- sure **to reduce the drift** 3

. 2.2 Related Works 2.2.1

Event-based SLAM/VO Event-based ego-motion estimation and perception have gained increasing interest for the task of SLAM/VO in challenging scenarios where the performance of traditional cameras is compromised, such as in HDR scenarios or aggressive motions. Weikers- dorfer **et al**. [32] propose **the first event-based 2D SLAM system, which tracks a ground robot pose while reconstructing the 2D ceiling map with an upward-looking event camera. Ref**. [37] claims **that they** develop **the first event-based VO to track the 6- DoF motion. However, this method is not purely event-based, since the features are first detected in the grayscale frames and then tracked asynchronously using** the **event stream. The first purely event-based 6-DoF VO** is **presented in** [38], **which performed real-time event-based SLAM through three decoupled probabilistic filters that jointly estimate the 6-DoF camera pose, 3D map of the scene, and image intensity. However, it is computationally expensive and requires** a **GPU to achieve real-time performance. EVO** [40] proposes **to solve the SLAM problem without** 3

recovering image intensity, thus reducing computational complexity, and it can run in real-time on a standard CPU. It performs a geometric approach which combines a tracking approach based on image-to-model alignment and semi-dense 3D reconstruction algorithm [41] in par- allel. However, the algorithm is needed to run in the scene that is planar to the sensor, up to several seconds, for bootstrapping the system. ESVO [59] is the first stereo event- based VO method, which follows a parallel tracking-and-mapping scheme to estimate the ego-motion and the semi-dense 3D map of the scene. However, it barely operates real-time in DAVIS346 (346*240) and is limited by rigorous and unreliable initialization. Furthermore, the ESVO needs to perform re-initialization in the case of too few events boosting. Ref. [60] proposes stereo VO for event cameras based on features. The pose estimation is done by re-projection error minimization, while the features are stereo

2.3. Framework Overview

and temporally matched through the consecutive left and right event TS. It solves the problems of ESVO mentioned above. However, it still cannot operate in real-time for high-resolution event cameras (640*480 3

). 2.2.2

Event-based Visual-Inertial Odometry Most of the event cameras, e.g., DAVIS or DVXplorer, have the IMU readily integrated. The first EVIO method is proposed in Ref. [43], which fuses a purely event-based track- ing algorithm with IMU through the Extended Kalman Filter. A similar method is proposed in Ref. [44], which generates motion-compensated edge images aggregated within temporal neighborhood events. The features are detected through a fast cor- ner detector [108] and tracked through pyramidal Lucas Kanade (LK) in the motion- compensated event image, and then combined with IMU measurement using graph- based optimization. The authors also extend this method to leverage the complemen- tary advantages of both standard and event cameras in Ultimate-SLAM [49] to fuse event frames, standard frames, and IMU. Both Ref. [44] and Ref. [49] aggregate the event data into an image frame and then adapt the conventional corner detection algo- rithms, such as FAST corners [108] or Shi-Tomasi [166] score for the feature detection. Ref. [48] proposes to fuse events and IMU measurement into a continuous-time frame- work. However, their approach cannot achieve real-time because of the expensive op- timization required to update the spline parameters upon receiving every event [44]. IDOL [55] investigates line-feature into VIO framework through directly using asyn- chronous raw events without using any frame-like accumulation. However, it also doesn't have real-time capabilities even in low-resolution event cameras (240*180). All of these EVIO works are only evaluated on low-resolution DAVIS240C (240*180). Be- sides, these works lack the loop closure function, which might cause a large amount of drift for long-term motions 3

. 2.3 Framework Overview Fig. 2.

1 gives an overview of our Mono-EIO modules involved and their interactions. Our Mono-EIO framework is composed of two sections: (i) 3

The front-end takes the raw event stream as input and extracts the event-corner features directly using the raw events based on the Surface of Active Events (SAE) [167]. **It establishes feature tracking and recovers the inverse depth of each tracked event-corner for generating 3D event- corner features as landmarks. Furthermore, more event-corner features are extracted and passed to the back-end for loop detection. Two kinds of event representations, the TS with polarity and** the **normalized TS without polarity, are designed for assisting uniformly distributed event-corner feature detection, feature tracking, and descriptor generation. (ii) The back-end tightly fuses the event-corner landmarks and the IMU measurements to estimate the 6 DoF state of the system while the loop closure is used to eliminate the accumulated drifts**

. Figure 2.1: Overview our proposed Mono-EIO pipeline. 2.4 Event-based Representations

An event is triggered only when **the intensity of an individual pixel varies** 4
exceeds a specific **threshold Tthreshold, which can be represented** as the **spatio-temporal** coordi- nates **of the intensity change** and its **sign: e = t**, u, v, **p ⇔ I**(u, v, **t + △t) − I**(u, v, **t) = p · Tthreshold** (2.1) **where t is the timestamp that the intensity of a** pixel (u, v) **changes, and p is the polarity** which **indicates** the direction **of the intensity change**

. Since a single event lacks sufficient

information, it is common to aggregate sets of 1

2.4. Event-based Representations

events within time intervals into synchronous data representations, **rather than** pro- cessing **individual events asynchronously** 1

. Therefore, various event representations have been designed to summarize the event stream, such as the accumulated event im- age [1], motion-compensated event-frame [40, 47],

the surface of active events (SAE) [168], **the** 31

time surface (TS) [169],

the TS with polarity [2], **the normalized TS without polar- ity** 3

[2], Spatiotemporal Voxel Grid [52, 84], the event spike tensor (EST) [170], and etc. In the following subsections, we introduce the event-based representations that are used in this chapter. 2.

4.1 Surface of Active Events (SAE) **The** vast majority **of** 47

event-driven algorithms [168, 171, 172, 167] have made use of vari- ants of the SAE. It is

originally defined as sensor-sized, spatial memories that store **each image pixel location** with **the timestamp of the latest event generated at** that **pixel** 32

loca- tion. Fig. 2.2 illustrates an example of the SAE in a patch (9 × 9) with the latest event at the center. This latest event is represented by the center red bar, which is the highest bar due to its most recent timestamp value. In essence, an SAE consists

> **of per-pixel, fixed- size queues** that store **events in a first-in, first-out** (FIFO) manner. **As a result**, the SAE stores **the** 32

recent events recorded by

> **the event** camera **and can be efficiently updated in an event-driven** manner, with **none or** 32

only minimal tuning parameters [173]. In addi- tion, the SAE also encodes the timestamp

> **information, coupling the scene appearance with the undergone camera motion** 32

. Figure 2.2: SAE representation in the patch (9 × 9) centered about an incoming event [174]. The vertical axis encodes

> **the timestamps of the latest event triggered in** each pixel coordinate. **The** SAE S **can be** 21

mathematically

> **defined as S** = tlast(**x, y), which is a simple 2D map containing the timestamp** tlast(**x, y) of the last event that occurred at each pixel, for positive and negative polarity, respectively. These two SAE record the triggered events in the stream, which are used for event-corner detection** (details **in** 3

Section 2.5.1) in our framework. 2.4.2 Time Surface (TS) The time surface (TS)

> **is a 2D map where each pixel stores the time value. It can** sum- marize **and update the event stream at any given instant, or encode the spatio-temporal constraints of the historical events** 5

.

> **More specifically, it** attempts **to replicate the** inten- sity **value of** 39

an image frame by aggregating the event stream with a temporal decay kernel. Through employing the

> **exponential decay kernel, TS can emphasize recent events over past events** [169]. tlast **is the timestamp of the last event at each pixel** coor- dinate **x = (u, v)T, the TS at time t** ≥ tlast(**x) is defined by: T(x, t) = exp(− t** − tlast(**x)) η** (2.2) **where η is the decay rate parameter** 2

. 2.4.3 TS with Polarity To build the event-corner feature tracker, we propose the TS with polarity. This event representation can be derived from the timestamp value of SAE with an exponential decay kernel, and we further incorporate

polarity into the TS (Tp(x, t)), which can be represented **as follows: Tp(x, t) = p · exp(− t − tlast(x)) η** 4

(2.3) where η

presents a constant decay rate (20-30 ms in our experiments). The polarity p is the sign of the brightness change for each event. **The TS with polarity Tp(x, t** 3

)

represents the recent history of moving edges with direction. We observe that **the polarity p is useful for feature tracking, since it records the direction of the event change** 3

, enabling it to effectively respond to scene edges under 2.4. Event-based Representations varying

relative motions (optical flow). As shown in the middle of Fig. 2.3(**a), the value of Tp(x, t) changes from "white to black" and "black to white" caused by opposite relative motions** 3

. Diverse motion directions induce events with opposite polarities at the same scene positions, resulting in different grayscale values in the TS with polarity (shown in the middle of Fig. 2.3(a)). Thus, the TS with polarity can effectively

cope with the fact that the same event-corner **feature can be represented** as completely **different in the event stream due to the** changes caused by **relative motion between** the **camera and** the **scene, even** 32

when appearance or viewpoint remain unchanged. This polarity-aware TS improves the stability of event-corner feature tracking, particularly when optical flow methods are used for data association, as they can leverage both motion direction and pixel intensity for more accurate tracking. (b) (a) (c) Figure 2.

3: (a) The raw event stream (left), the Time Surface with polarity (middle), and the normalized Time Surface without polarity (right); (b) Event-corner tracking on Time Surface with polarity; (c) Loop detection using the event-corner features and the normalized Time Surface without polarity 3

. 2.4.4 Normalized TS without Polarity To develop the event-corner feature matcher, we design a normalized TS without po- larity as the event representation for the loop closure. The proposed

normalized TS without polarity (Tnp(x, t 4

)) is also generated from the

SAE with the exponential decay kernel 4

, which

can be represented **as follows**: Tnp(**x,t**) = (max(**T′**)−min(**T′**))·(**T** −min(**T** 122

)) 255.0 ′ ′ (2.4) where, T can be obtained from Eq. 2.3. The inclusion of polarity information in the ′ event stream can introduce ambiguity, particularly

when the sensor moves in different directions within **the** 3

same scene. In such cases, the pixel values of identical edges may vary based on polarity, which can significantly impact feature matching.

To address this issue, **we propose an alternative** event **representation** 59

for loop detection,

the TS without polarity, and further refined it to normalized TS without polarity 3

Tnp(x,

t) (as shown in the right side **of Fig**. 2.3(**a**)). This representation captures **the normalized temporal- spatial constraints as** a **scene outline**, ensuring **spatial consistency**. By emphasizing edges, **this event representation** effectively captures **the scene structure**, providing **the** 3

necessary

information for creating **discriminative event-corner descriptors** 3

(details in Section 2.5.3). 2.5 Event-based Data Association Similar to the standard image counterparts, event-based state estimation also aims at solving simultaneously the mapping and tracking sub-problems in a recursive man- ner. The main challenge lies in extracting and maintaining effective data association in the event stream, which allows for inferring depth information and ego motion. For a

new event-stream coming, firstly, the existing event-corner features are tracked by the LK optical flow [109] **on the TS with polarity Tp(x, t** 3

) (details in Section 2.5.2).

The features that are not successfully tracked in the current timestamp would be discarded. After that, new event-corner would be detected from the latest raw event stream when- ever **the number of the tracked features falls below a certain threshold (150-200 in our experiment). Modified from the publicly available implementation of the Arc*** algo- rithm [168] **for event-based corner detection** 3

(detailed in Section 2.5.1),

we extract the event-corners on the raw individual event by leveraging the SAE rather than adopting the conventional image-based **corner detection algorithms** 3

. 2.5.1 Event-Corner Feature Detection

Event-driven feature detection is fundamentally different from the traditional image- **based** corner detection **approaches, due to the notable dissimilarities between** these two **sensing modalities** 32

. In event-driven sensing,

each event captures **only a small**, localized **change at** the **pixel** 32

level. Unlike image-based approaches, it doesn't rely on absolute intensity information, as this data

is not explicitly encoded in the event stream. As a result, **event** 32

-driven feature detection cannot depend solely on the most recent 2.5. Event-based Data Association event in isolation. Instead, it must also take into account the sequence of events that occurred immediately before it. We aim to

use the asynchronous raw events for feature detection 3

without any frame-like accumulation or using the intensity value, the event- corner

features can only be detected by inspecting the **spatiotemporal distribution and polarity** 32

of the events. Therefore,

the SAE is used to summarize and update the event stream at any given instant 3

. Figure 2.4: Figure is borrowed from [168], which illustrates

the Arc* algorithm for event-corner detection. (a) For **the** new coming **event** 5

(green) located in a corner region (grey background), triggers the inspection of a circular neighborhood C (blue) in the lo- cal SAE. (b) The circular arc can be initialized from the most recent element Anew (higher intensity values indicate newer timestamps), with adjacent clockwise ECW (cyan) and counter-clockwise ECCW (red) elements defining arc boundaries. (c) The Anew would be expanded up to a minimum arc length Lmin = 3, and the ECW go to the new posi- tion. (d) During the following iterations, the position of ECW is kept updated since its timestamp is bigger than the ECCW. (e) The position of the ECCW would be updated since its timestamp is bigger than the ECW, while the first update step is skipped with minimum arc length Lmin = 3, and

the circle is completed. The event is classified as a corner 32

feature as the complementary continuous arc length is 4, which satisfies the predefined threshold range [Lmin, Lmax] = [3, 6]. Our event-corner detection scheme leverages the Arc* algorithm [168], which main- tains

two circular sets of events around the new arrival event and detects as corners whenever the continuous arc or its complementary arc in these two circular sets **of SAE is within a certain range** 3

. As illustrated in Fig. 2.4(a), the blue bars represent the neigh- boring pixels surrounding the new incoming event (depicted as the green bar), where timestamp relationships are analyzed to identify corner events. When a new event ar- rives, its location serves

> **as the center of a circular mask** with **a predefined radius (in** 32

our implementation, radius = 3 and 4, corresponding to 16 or 20 pixels, respectively). For enhancing robustness against noise, the detection process employs two distinct cir- cle radii (radius = 3, 4), and an

> **incoming event is classified as a corner** feature only **if** it passes **the** 32

check in both cases. The detection process examines the surrounding pixels within the circular neighborhood in the SAE (denoted as the set C), focusing on the regions (i.e., the SAE) with the same polarity as the incoming event. The newest timestamp location Anew is inspected, and its adjacent clockwise element ECW and counter-clockwise element ECCW are initialized as supporting elements (as shown in Fig. 2.4(b)). In each iteration, the algorithm selects the newer supporting element ECW or ECCW based on the larger timestamp value.

> **If the oldest element in** Anew **is older than the selected** newer **supporting element** 32

, the Anew is expanded to include the selected element in either the clockwise or counter-clockwise direction, and the position of the corresponding supporting element is also updated (as illustrated in Fig. 2.4(c) (e)). Dur- ing initialization, the Anew is always expanded to a minimum arc length Lmin

> **even if the** above **condition is not** satisfied. **The iterative procedure** continues **until** 32

the cir- cle is fully traversed, i.e., until ECW and ECCW point to the same element. Finally,

> **the incoming event is classified as** an event-**corner** feature **if the length of the continuous arc** Anew **or its complementary arc** 32

is within a pre-defined range [Lmin, Lmax] = [4, 6] for radius=3, and [Lmin, Lmax] = [5, 8] for radius=4, respectively. If the arc length does not fall within this range, the event is not considered an event-corner feature. More details implementation and modification of Arc* can be seen from our source code : https://github.com/arclab-hku/ESVIO. The newly

> **detected event-corner** features **would be further selected by setting the TS with polarity Tp(x, t) as the mask** 3

.

> **To enforce the uniform distribution, a minimum distance (10-20 pixels for different resolution event cameras) is set between two** neigh- boring **event-corner features. Meanwhile, we maintain the event-corners, where the pixel value of the TS with polarity** Tp(x, t) **is not equal to 128.0, to emphasize the** de- tected **event-corner features located in the strong edges rather than the too many noisy features in low texture areas** 5

. This process is visualized in Fig. 2.5. 2.5.2 Event-Corner Feature Tracking In traditional image-based SLAM, the data association relies on establishing spatiotem- poral correspondences through identifying and tracking persistent features across mul- tiple frames. This process typically describes features through the patterns of pixel in- tensity values surrounding their locations, allowing them to be matched either across

different parts of an image or with newly detected features in later frames. On the other hand, event cameras capture both scene appearance and motion, which creates challenges in

event-based feature tracking. Specifically, **the** same **event**-corner **feature** 2

2.5. Event-based Data Association (

a) Raw Event Stream (b) TS with 4

Polarity Figure 2.5:

Event-corner features generation. The event-corner features are first extracted **from the asynchronous event stream (a), then the TS with polarity (b) is used as a mask to further select the event-corner features, ensuring a uniform distribution** 2

. may appear

entirely different in the event stream due to the relative motion between the **camera and** the **scene, even** 32

if there are no changes in appearance or viewpoint. To meet the demand of deep reliance on comparing intensity value and the motion- dependence, we propose to use

the TS with polarity Tp(x, t) (details **in** Section **2** 3

.4.3) to track event-based corner features. We apply

the LK optical flow [109] **on the Tp(x, t) to associate the current event-corner with** its **most recent** counterpart within **the kernel operation, assuming** they correspond to **the same event-corner** at **a recent position. Our approach** leverages **the motion variance characteristics of the Tp(x, t) to** maintain **relevant context while** as- sociating **event-based point features into tracks to ensure computational efficiency in the front-end** 2

. This polarity-aware tracking method can enhance the robustness of the event-corner feature tracking (more verification can be shown in Fig. 2.6). Since the mo- tion direction of the event camera influences the polarities of the generated events, the changes of motion direction cause the latest events to exhibit opposite polarities. This can result in fluctuations in the grayscale values of the TS with polarity corresponding to these events, which in turn enables accurate tracking of event-corner features. Figure 2.6: Examples visualization for the feature tracking

using the event-corner fea- tures **and the Time Surface** with **polarity** 3

. The green arrows on the event-corner features depict the estimated direction and magnitude of the optical flow, while the green lines linking pairs of event-corner features indicate the temporally tracked points across con- secutive frames. In addition,

we use a two-way tracking strategy to track event-corner features between **two consecutive timestamps. For any event-corner feature Fe on last timestamp Tp(x, t) is tracked to Fe′ on current timestamp Tp(x, t), we would reverse the** track- ing **process by tracking** Fe′ **on current timestamp** Tp(**x, t) back to Fe″ on last timestamp Tp(x, t). If the distance between** Fe″ **and** 2

Fe′ is smaller than a threshold (1.0 pixel in our implementation), **this event-corner feature would be viewed as being successfully tracked. The event-corner features that are not successfully tracked in the current** times- tamp **would be discarded immediately**. While **the** successful **tracked**

event-corner fea- tures are maintained and labeled as candidate features. 2.5. Event-based Data Association

Furthermore, all the event-corner features are first undistorted **based on the** cam- era **distortion model, and then projected to a normalized camera coordinate system. To remove outliers, we also use the Random Sample Consensus (RANSAC**) with **the** 3

fundamental matrix model [175]

to further filter the outliers. After that, we recover the inverse depth of the features that are successfully **tracked between two consecutive timestamps through triangulation. The landmark whose 3D position has been** success- fully **calculated would be fed to the sliding window for the pose graph optimization. It is worth mentioning that, although we try to maintain the number of event-corners within a certain threshold, the number of event-corners used for pose graph** optimiza- tion **still depends on the relative motion and texture of the scene** 3

. 2.5.3 Event-Corner Feature Matching for Loop Closure

To eliminate the accumulated drifts and ensure global consistency for long-term mo- tion, **in addition to the event-corner features that are used for** the EIO **front-end, extra event-corners are detected, and then described by the BRIEF descriptor** [176], **and** fur- ther **feed to the back-end (as depicted in Fig**. 2.1). **These additional event-corner** fea- tures **are used to achieve a better recall rate on loop detection. Thanks to our designed normalized TS without polarity** 3

, Tnp(x,

t), which would be triggered in the scene that has strong edges, it can help for the place recognition and ensure global consistency. The correspondences are found by the BRIEF descriptor matching by calculating the **Hamming distance. When the number of correspondences of the event-descriptors is greater than a certain threshold (16-25 in our experiments), the loop closure is** de- tected (**as shown in Fig**. 2.3(**c**)). **We adopt the two-step geometric outlier rejection for wrong BRIEF descriptor matching, DBoW2 for loop recognition, and** the **re-localization scheme from** 3

[8] in our implementation. An example visualization for our event-based loop detection can be seen in Fig. 2.7.

After detecting the loop, the connection residual of the previous keyframe and the current keyframe are integrated into the pose graph **optimization as a re-localization residual** 5

(detailed in Section 2.6.4). Figure 2.7: Examples of visualization for the

loop detection using the event-corner features and the normalized Time Surface without polarity	3

. 2.6 Tightly-coupled Event-Inertial State Estimation In this chapter, we consider (·)W, (·)E, and (·)B as the coordinate in world frame, event frame, and body (IMU) frame, respectively. The kth event frame is denoted as Ek. For two consecutive keyframe k and k + 1, TEEkk+1 ∈ SE(3) denotes the homogeneous trans- formation matrix from the Ek to Ek+1. The pose of Ek

in the world frame W can be represented by TWEk. We use the	85

quaternions q to represent the rotation, and the ro- tation matrix R is derived from the quaternion q. qWB and pBW are

the rotation and translation from the world frame to the body frame	117

. (ˆ·) is used to represent the noisy measurements. 2.6.1 Formulation of the 6-DoF State Estimation Problem Following the visual-inertial bundle adjustment (BA), the Mono-EIO can be

optimized by minimizing a joint nonlinear least-squares problem for the system state χ as follows: arg min χ ⎧ ⎨ ⎩ ||rm − Hmχ||2 + K∑−1	1

||rB(ẑbbkk+1, χ)||2Σbk + ∑ LH||rE(ẑElk, χ)||2ΣEk ⎫ ⎬ ⎭ (2.5) k=0 bk+1 (k,l)∈E l ⎧ ⎨ ⎩ ⎫ ⎬ ⎭ where LH(·) is the Huber loss; rm,rB, rE represent the residuals of the marginaliza- tion, IMU pre-integration, and event-corner reprojection measurements, respectively; ẑbbkk+1 stand for the observations of IMU. While ẑElk are

the lth tracked event-corner fea- ture in the	1

kth event frame Ek. k ∈

K (K = 10 in our experiments) is the total number	1

2.6. Tightly-coupled Event-Inertial State Estimation of

keyframes in the sliding window. Hm is the measurement estimation matrix of	85

the marginalization, and denotes the covariance of each term. Σ The full state vector of our Mono-EIO

in the sliding window is defined as: χ = [χb, Teb, λe	4

] (2.6) where Teb = [Rbe, tbe]

is the extrinsic transformation from the event camera frame E to the body (IMU) frame B; λe = [λ0, ..., λm] is the inverse depth of the mth event-corner features in the sliding windows; χb = [X1, ..., XK] is the optimization variables in the sliding windows, which comprises the state of the IMU, with K (K = 10 in our experi- ments), the total number of keyframes in the sliding windows. The system state Xk at kth keyframe is given by the position pwbk, orientation quaternion qwbk, and the velocity vwbk of the	3

IMU in the world frame, and the accelerometer bias bak **and gyroscope bias** bgk **as follows: Xk = [pwbk,qwbk**

,vbwk,bak,bgk] (2.7) 2.6.2 Event-based Visual Constraint The rE(ẑElk , χ) in Eq.(2.5)

is the event-corner measurement residual from the re-projection function. Considering the lth **event-corner feature that is first observed in the** ith **keyframe, the residual for its observation in the** kth **keyframe is defined as** 5

: rE(ẑElk, χ) = ⌈ ulk vlk ⌉ − πe · (Teb)−1·Twbk · Tbwi · Teb · πe−1(1 λ , ⌈ uli (2.8) e ⌉)⌊ ⌊ ⌋⌋ vli⌊ ⌋ where, uli, vli

is the first observation of the lth event-corner feature **in the ith keyframe**. T ulk, vlk [**T is** th]e **observation of the same event-corner feature in the kth keyframe** 1

, Tbwi in- dicates

the movement of the body frame related to the world frame in timestamp i, Twbk [ ] **is the transpose of the pose of the body in the world frame** 1

in the kth keyframe. πe and πe−1 are the projection and back-projection function of the event camera, respec- tively, **which** include **the intrinsic** parameters **for the transform between the 2D pixel coordinates and normalized event camera coordinate** 5

.

fx 0 cx 0 ⌈ πe = 0 **fy cy 0** ⌉ (2.9) 0 **0** 1 **0**⌊ **0** 10

0 0 1 ⌊⌋ ⌊ 2.6.3 IMU Pre-integration Constraint The rB(ẑbbkk+1 , χ) in Eq.(2.5)

is the IMU residual from the IMU pre-integration. The raw measurement of angular velocity ωk and acceleration ak from IMU at time tk are: âk = ak − Rωbk **gω + bak + na** (2.10) **ω^k = ωk + bωk + nω where na, nω are modeled as additive Gaussian noise. bak**, bωk **are modeled as** ran- dom **walks. The Notation (^·) is used to represent noisy measurements. Given the time interval [tk, tk+1] corresponding to keyframe bk and bk+1. pωbk+1**, vbωk+**1**, qbωk+**1 can be** prop- agated **in such time interval by using gyroscope and accelerometer measurements in the world frames as** follows: pωbk+**1** 5

= pbωk +vbωkΔt+ tk+1 ∫∫tk (Rbωk ak)δt2 vbωk+1 = vbωk + tk+1 ∫tk (Rbωkak)δt (2.11) qbωk+1 = tk+1 ∫tk qbωk ⊗ ⌈ 0 12ωk ⌉ δt⌊ ⌊ ⌋⌋ 2.6. Tightly-coupled Event-Inertial State Estimation Based on Eq.(2.10), Eq. (2.11)

can be rewritten as follows: pbωk+**1** − pbωk − **vωbk** Δ**t** − 1 gωΔ**t2** 2 = **tk**+1 ∫∫tk (**Rωbk (âk − bak − na**))δt2 vωbk+**1** − vbωk − **gωΔt tk** 4

+1 (2.12) = ∫tk (Rωbkâkδt − Rωbkbakδt − Rωbknaδt) qbωk+1 = tk+1 qωbk ⊗ ⌈ 0 ∫tk 12 (

ωˆk − bωk − nω) ⎤ δt ⎡ ⎢ ⎥ ⎦ In order to ensure the pre-integration term is only related to the inertial measurements and biases in [tk, tk+1], Rωbk is multiplied on both sides of Eq.(2.12), and we define the pre-integration term αbbkk+1, βbbkk+1, γbbkk+1 as follows: αbbkk+1 = Rωbk tk+1 ∫ ∫tk (Rbωk(âk − bak − na))δt2 βbbkk+1 = Rωbk ∫tktk+1(Rωbkaˆkδt − Rωbkbakδt − Rωbknaδt) γbbkk+1 = qωbk ⊗ qbωk+1 — 4

Discretizing Eq.(2.13) by the zero-order discretization method as follows — 2

: αˆbbki+1 = αˆbbki + βˆbbki δt + 12 R(γˆbbik)(âi − bai)δt2 βˆbbki+1 = βˆbbki + R(γˆbbik)(âi bai)δt − γˆbbki+1 = γˆbbki ⊗ ⎡ 1 12 (

ωˆi − bωi) ⎤ δt ⎢ ⎣ ⎥ ⎦ Eventually, the IMU residual can be derived as follows: ⎡ Rbk (pωbk+1 − pωbk − vωbk Δt − 12 gωΔt2) − αˆbbkk+1 Rbk (vbωk+1 − vωbk − gωΔt) − βˆbbkk+1 ⎤ rB(ẑbbkk+1,χ) = 2 (qωbk)−1 ⊗ qωbk+1 ⊗ (γˆbbkk+1 — 4

)−1 [bak+1 − bak] xyz ⎢ bωk+1 − bωk ⎣ ⎦ ⎥ (2.13) (2.14) (2.15) 2.6.4 Event-based Re-localization Constraint The event-based re-localization

effectively aligns the current sliding window of esti- mated poses with the past poses — 75

stored in the global pose graph, as [8]. This alignment ensures consistency in the pose estimation process and facilitates higher accuracy. The re-localization residual can be easily written as

the re-projection error for the retrieved event-corner features of the — 1

loop closure matching, as Eq. (2.8): rL(ẑElk,χ) = ⎡ ulk ⎤ − πe · (Teb)−1·Twbk · Tbwv · Teb · πe−1(λ1e ,⎡ ulv ⎢ ⎣ vlk ⎥ ⎦) (2.16) ⎦ ⎥ ⎢ ⎣ vlv ⎥ ⎦ where (l, v) means

the lth retrieved event-corner feature observed in the — 4

loop closure frame v. Tbwv is

the pose of the loop-closure frame which is taken from the global pose graph. When the — 65

loop is detected, this re-localization factor would be added into Eq. (2.5). 2.6.5 Initialization and other Implementation Details Initialization: Adopted from [8, 177],

the initialization procedure of our Mono-EIO starts with a vision-only structure from motion (SfM) to build the up-to-scale structure of camera pose and event-corner feature positions. Through loosely aligning the SfM with the pre-integrated IMU Measurements, it can bootstrap the system from unknown initial states, instead of assuming the sensor remains static during the initialization phase [49, 44] or assuming the local scene is planar to the sensor [40]. Keyframe Selection: A new keyframe is selected by two criteria: (i) When the av- erage parallax of the tracked features, between two consecutive timestamps, is beyond a threshold (10 is set in our experiment). (ii) When the number of successfully tracked features from the last — 3

timestamp falls below a certain threshold (30 is set in our experi- ment). **Still State: Since the event cameras output very** few **events (only noise) when the sensor is still. We would restart the** EIO **estimator whenever the number of** events **falls below a threshold (1000 is set in our experiment) to avoid divergence**. 2.7. Experiments **Low Latency: To achieve low latency, we directly forward propagate (loosely- coupled) the latest** EIO **estimation with the IMU measurements to achieve IMU-rate** EIO **outputs which can be up to 1000 Hz**

. 2.7 Experiments

In this section, we assess the accuracy of our Mono-EIO **framework both quantitatively and qualitatively on different challenging sequences with different resolution event** cameras (**Table** 2.1). **We implemented our** Mono-EIO **method with C++ in Ubuntu 20.04 and ROS Noetic. All the sequences are evaluated in real-time using a laptop with Intel Core i7-11800H and are recorded in videos** (see **our project website). In subsection** 3

2.7.1,

we compare the accuracy of our Mono-EIO **framework with other current** event-based **works in a publicly available Event Camera Dataset** [144] **which is acquired by the DAVIS240C (240*180, event-sensor, image-sensor, IMU** sen- sor), **it contains extremely fast 6-Dof motion and scenes with HDR. Table** 2.1: **Summary of the data sequences in** this **Section. rosbag Section** 3

2.7.1 Public Dataset [144] Section 2.7.2 Real-test Section 2.7.3

Real-test Sensor Davis240C DAVIS346 DVXplorer DVXplorer /optitrack/davis /dvs_vicon/gt_pose No Ground Truth Topic /dvs/events /davis346/events /dvxplorer/events /dvs/events /dvs/imu /davis346/imu /dvxplorer/imu /dvs/imu /dvs/image_raw /davis346/image_raw No Image Event Stream Rate 30HZ DAVIS346: 60 HZ DVXplorer: 50 HZ 50HZ Average Duration 59.8s 156.3 s 171.6s Data size 7 × 105 **DAVIS346: 6** × 105 ***DVXplorer: 2** 3

× 106 2 × 106 Resolution 240*180

DAVIS346:346*240 DVXplorer:640*480 640*480 Description **indoor aggressive HDR scenarios under** 3

optitrack

indoor aggressive HDR scenarios under vicon indoor&outdoor HDR scenarios long-term * Datasize : The number of events per stream, e.g. 2 × 106 **indicates the average number of events for the sequences is 2** × 106 × **50HZ × 156.3s = 1.6** × 1010. **To further explore the performance of our** Mono-EIO framework **in high-resolution event cameras, in subsection** 3

2.7.2 and 2.7.3,

we use the DAVIS346 (346*240, event- sensor, image-sensor, IMU sensor) and DVXplorer (640*480, event-sensor, IMU sensor) for data 3

collection. It is worth mentioning that our Mono-EIO only uses the event stream and IMU data. The image-frame output of the DAVIS is only used for illus- tration purposes or image-based VIO/VO comparison. The event camera and IMU calibration (including the intrinsic and extrinsic parameters, and the time offset of the camera-imu) are estimated using Kalibr [178] as well as the DV module 1

.

A motion capture system (VICON) is used to obtain the pose ground truth. Since the active infra- red (IR) emitters on the VICON cameras would influence the event camera greatly, we adopt the IR filter lens to remove the IR influence 3

. These evaluated data sequences are detailed in Appendix A. 2.7.1 Accuracy Evaluation in Aggressive Motions

The estimated and ground-truth trajectories are aligned with a 6-DOF transformation (in SE3), using 5 seconds [0-5s] of the resulting trajectory. We computed the mean po- sition error (Euclidean distance in meters) as a percentage of the total traveled distance of the ground truth, which is calculated by the publicly available RPG Trajectory Eval- uation tool [179]. Due to most of 3

these baselines are not

open source, we directly refer to the raw results from 3

[43, 44, 49, 180, 181, 65] which are

also aligned with SE3 using 5 seconds. Table 2.2 shows the remarkable accuracy of our method compared to the 3

base- lines EIO

works. It's worth mentioning that, although IDOL [55] also provides their results in this Event Camera Dataset [144], they just run 0-40 s of the dataset to avoid the aggressive motion. Fig. 2.8 presents the estimated trajectories against the ground truth for the sequence dynamic_translation and dynamic_6dof, which are generated by the publicly available tool EVO [182], and also visualizes the relative translation and yaw error by averaging the drift over different segments of the trajectory. We find that our Mono-EIO achieves fairly good results 3

. 1https://

gitlab.com/inivation/dv/dv-imu-cam-calibration 3

2.7. Experiments Table 2.2:

Accuracy Comparison of Our Mono-EIO with Other Event+IMU Works in 1

DAVIS240c Dataset

[144].

Unit:%, 0.39 **means the average error would be 0.21m for 100m motion; Aligning 5 seconds [0-5s] of the estimated trajectory with the ground truth; The notations E, F, and I stand for the use of event, frame, and IMU, respectively** 1

. Sequence Ref. [43] Ref. [44] Ref. [49] Ref. [180] Ref. [181] Ref. [65] Our Mono-EIO (

E+I) (E+I) (E+I) (E+I) (E+I) (E+I) (E+I 35

)

boxes_translation 2.69 0.57 0.76 1.50 **2.55** 1.0 **0.34 hdr_boxes 1.23 0.92 0.67** 2.45 **1.75** 1.8 **0.40 boxes_6dof 3.61 0.69 0.44** 2.88 **2.03** 1.5 **0.61 dynamic_translation 1.90 0.47 0.59** 4.92 **1.32 0**.9 **0.26 dynamic_6dof 4.07 0.54 0.38** 6.23 **0.52** 1.5 **0.43 poster_translation 0.94 0.89 0.15** 3.43 **1.34** 1.9 **0.40 hdr_poster 2.63 0.59 0.49** 2.38 **0.57** 2.8 **0.40 poster_6dof 3.56 0.82 0.30** 2.53 **1.50** 1.2 **0.26 Average 2.58 0.69 0.47** 3.92 **1.45** 1.56 **0.39** 5

2.7.2 Accuracy Evaluation

in High-Dynamic-Range Scenarios To **further** demonstrate **the robustness, accuracy, and real-time capability, we also** eval- uate **our** Mono-EIO **using high-resolution event cameras (DAVIS346 (346*240) and DVX- plorer (640*480)) with the ground truth from VICON** 3

. These data sequences are from the monocular HKU-dataset.

The DAVIS346 and DVXplorer are attached together (de- tails **in** Appendix **A) for facilitating comparison. All the sequences are recorded in HDR scenarios with very low** illumination **or strong illumination changes through switching the strobe flash on and off, while the** 11st **sequence** (vicon_**aggressive_hdr) is** character- ized **by aggressive motion. Without loss of generality, we also used the raw image from DAVIS346 to run the VINS-MONO** [8] **and the VO version of ORB-SLAM3** [9] (**the VIO version failed or** could not **initialize in all the sequences), as image-based VIO/VO for comparison. The results are shown in Table** 2.3. **It is worth** 3

mentioning that, for the results of vicon_aggressive_hdr, our Mono-EIO **produces reliable and accurate pose tracking even when the image-based VIO and VO** fail. **Although the VO version of ORB-SLAM3 performs comparably to our** Mono-EIO **in DAVIS346, it would track failures and lose tracking frames during aggressive** mo- tion **or too dark scenarios which would affect the generation of the descriptor seriously. Thanks to the re-localization scheme, the ORB-SLAM3-VO can handle the tracking** fail- ure **issue after re-detecting the ORB descriptor in good illumination** conditions, **but** 3

(

a) dynamic_translation (b) dynamic_6dof Figure 2.8: **Comparison of translation and rotation estimates of our proposed** Mono- EIO **against ground truth; The relative errors of the translation and yaw angle** 3

.

this would cause interruption during the estimation. While our Mono-EIO **can** pro- vide **continuous and smooth state estimation. In addition, one of the main challenging aspects of this dataset is that most of the scenarios are dark (please see the** supplemen- tal **video), this would introduce many noise events. However, our** Mono-EIO **still can obtain satisfactory results. Another challenging aspect is the heavy event load of this dataset compared with the ones in subsection** 3

2.7.1.

We found that the amount of event data from the high- resolution event camera (such as DVXplorer) is as large as the order of 106 (**for each event stream, shown in Table** 2.1). **The public event camera dataset** [144], **which only works on 30 HZ event stream inputs. However, to further evaluate the performance of our** Mono-EIO **in the large throughput event load, we modified the driver code for DAVIS346 and DVXplorer with a higher stream rate and not limited maximum** number of **events for each stream, which can also ensure a steady frequency of the event stream. The large throughput of our well-designed feature management ensures our system can handle 60 HZ data input from DAVIS346 with** 30 HZ **front-end output, and 50 HZ data input from DVXplorer with** 25 HZ **front-end output. It can ensure the real-time ability for heavy event load in the high-resolution event camera rather than slowing** 3

2.7. Experiments Table 2.3:

Accuracy of our Mono-EIO **compared with VINS-MONO and ORB-SLAM3** 3

.

Unit:%/m, 0.54 means the average error would be 0.54m for 100m motion 3

.

Sequence VINS-MONO [8] **ORB-SLAM3** [9] **Our** 3

Mono-EIO Our Mono-EIO DAVIS346 DAVIS346 DAVIS346

DVXplorer vicon_hdr1 0.96 0.32 0.59 0.30 vicon_hdr2 1.60 0.75 0.74 0.37 vicon_hdr3 2.28 0.60 0.72 0.69 vicon_hdr4 1.40 0.70 0.37 0.26 vicon_darktolight1 0.51 0.75 0.81 0.80 vicon_darktolight2 0.98 0.76 0.42 0.57 vicon_lighttodark1 0.55 0.41 0.29 0.81 vicon_lighttodark2 0.55 0.58 0.79 0.75 3

vicon_dark1 0.88 failed 1.02 0.35 vicon_dark2 0.52 0.60 0.49 0.41 vicon_aggressive_hdr failed failed 0.66 0.65 Average 1.02 0.61 0.63 0.54

down the rosbag reproduction, just like [59, 60]. **The running time of each module in our EVIO can be seen in Table** 3

2.4 Table 2.4:

Running Time of our Mono-EIO **in different resolution event cameras (ms) Modules DAVIS240c DAVIS346 DVXplorer Creation of Event Representations** 3

Event-corner Feature Detection

Event-corner Feature Tracking The Whole **Front-end** 3

Process Event-corner Loop Matching 0.37 0.98 0.49 0.40 0.86 0.71 3.98 3.81 27.73 19.56 3.65 1.60 1.16 12.38 67.95 2.7.3 Testing in Outdoor Scenarios

The workspace of the sequences in the previous section **is relatively small, it is** dif- ficult **to distinguish between drift and failure from** the **error value alone. Therefore, in this** 5

section, several sequences are **further recorded outdoors, in** the **HKU campus**, featuring **aggressive motion, long-term movement, strong sunlight, or indoor-outdoor conversion. Additionally, there are several pedestrians in the scene generating outlier events. Since the motion capture system is not available outdoors, we just evaluate the qualitative performance, and we also returned to the same location after a large loop to evaluate the loop closure. As can be seen from Fig**. 2.9, **the estimated trajectory is aligned and almost** coincides **with the Google map. It is worth mentioning that our** Mono-EIO **can effectively detect feature points at a distance of up to 20 meters (shown in Fig**. 2.3(**b**)). **More evaluations for the outdoor environment, robust feature tracking, and the loop closure can be seen in our** project website. Figure **2** 3

.9:

Estimated trajectory in outdoor environment aligned with Google map 3

. 2.7.4

Quadrotor Experiment using DVXplorer-Mini In this section, we demonstrate the quadrotor flight based on our proposed Mono- EIO **using DVXplorer-Mini (640*480). We build our quadrotor (Fig**. 2.10(**a**)) **from** se- lected **off-the-shelf components and custom 3D printed items. Our quadrotor relies on a QAV380 frame with T-MOTOR F60 KV2550. The electronic parts of our quadrotor comprise a Pixracer (FMUv4) autopilot with** an **Up Xtreme i7 8665ue** 3

computer, which runs Ubuntu 20.04 and ROS Noetic. The DVXplorer-Mini is mounted on the front of the quadrotor, looking forward, **which is connected to the Up Xtreme via** a **USB 3.1 A to C cable and transmits events and inertial measurements for our** Mono-EIO **state** estima- tion. **The quadrotor is commanded to hover with slight jitter, our** Mono-EIO **estimated result against the VICON ground truth can be seen in Fig**. 2.10(**b** 3

). 2.8 Conclusion

In this chapter, **we have developed a low-latency, real-time, monocular event-inertial odometry framework to provide accurate metric tracking of the 6 DoF pose. The method** 3

2.8. Conclusion (a) (b) Figure 2.10: (

a) Our quadrotor platform and VICON room; (b) The estimated trajectory and its comparison against VICON 3

.

is based on our designed steady and uniformly distributed event-corner feature detec- tor, **which is done with raw individual events but consecutively tracked in TS with** po- larity, **and spatially matched in normalized TS without polarity. Our** Mono-EIO **can** es- timate **up to 1000 HZ poses while recovering a sparse 3D map of the environment. The performance of the proposed** method **is quantitatively and qualitatively evaluated in different resolution event** cameras: **DAVIS240C (240*180, publicly dataset), DAVIS346 (346*240, real-test), DVXplorer (640*480, real-test), and DVXplorer-Mini (640*480**, quadro- tor **flighting). Our method** achieves **fairly good performance compared with state-of- the-art** VIO **works**, such as **VINS-MONO, and ORB-SLAM3**. However, **the limitations of our work might be that the event cameras tend to only trigger events over edge-like features**, and **low-texture areas generate very few events. Therefore, our** Mono-EIO **might suffer some problems in less texture scenarios. In** the next chapter, **we** further **explore the line-based features for** event-based SLAM. **Besides, multi-sensors** 3

, includ- ing image, would

be fused together to achieve more robust state estimation and exploit the complementary advantage of different sensors with event cameras. It **is** 3

worth noting that SuperEIO [106] utilizes a learning-based event-only detec- tor and descriptor to achieve EIO, while its backbone is built in Mono-EIO. Specifically, the event-based data association employs optical flow tracking in the TS with polar- ity, as described in Eq (2.3). This further validates the effectiveness of the proposed event corner feature tracking method. While its feature management, such as

uniform distribution of the learning-**based event features**, is also **the** 5

same as Mono-EIO. 2.9 Related Publications 1. Guan Weipeng, Lu Peng, “

Monocular Event Visual Inertial Odometry based on Event-corner using Sliding Windows Graph-based Optimization", 2022 **IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2438-2445** 1

, 2022. 2. Chen Peiyu*, Lin Fuling*, Guan Weipeng, Lu Peng, "

SuperEIO: Self-Supervised Event Feature Learning for Event Inertial Odometry 11

". Chapter 3 EVIO: Image-aided Event-based Visual-inertial Odometry

Robust state estimation in challenging **situations is still an unsolved problem**, espe- cially **achieving onboard pose feedback control for aggressive motion** 2

. Building on the previous proposed Mono-EIO in Chapter 2, this chapter proposes

robust and real-time event-based visual-inertial odometry (EVIO) **that incorporates event, image, and** iner- tial **measurements. Our approach utilizes line-based event features to provide** addi- tional **structure and constraint information in human-made scenes, while point-based event and image features complement each other through well-designed feature** man- agement. **To achieve reliable state estimation, we tightly couple the point-based and line-based visual residuals from the event camera, the point-based visual residual from the standard camera, and the residual from IMU pre-integration using a keyframe- based graph optimization framework. Experiments in the public benchmark datasets show that our method can achieve superior performance compared with the state- of-the-art image-based or event-based VIO. Furthermore, we demonstrate the** effec- tiveness **of our pipeline through onboard closed-loop quadrotor aggressive flight and large-scale outdoor experiments** 2

. In addition, we also design the stereo

event-based visual-inertial odometry sys- tem (shown **in** 2

Fig. 3.1), including

both ESIO (purely event-based) and ESVIO (event with image-aided 17

), to exploit the event-based temporal and spatial data association.

To the best of our knowledge, the ESVIO is the **first** stereo **event-based visual inertial** 8

Chapter 3. EVIO: Image-aided Event-based Visual-inertial Odometry odometry framework.

Due to space constraints, we refer **readers to** our **original** pa- per [73] **for** 21

details on the stereo event constraints and the implementation of ESVIO. While this chapter focuses on the introduction of PL-EIO and PL-EVIO. This chap- ter is based on our works [72, 73]. More demonstrations can be seen in the project website PL-EVIO: https://kwanwaipang.github.io/PL-EVIO and ESVIO: https://kwanwaipang.github.io/ESVIO. Figure 3.

> **1: Our ESVIO** [73] **provides robust and accurate, real-time pose feedback for drones under aggressive motion. Events provide rich and reliable features, while only a few features are tracked in image frames in high-speed motion. Left bottom: stereo event-based feature tracking. Right bottom: stereo image-based feature tracking** 7

. 3.1 Introduction

> **Driven by the need for real-time closed-loop control for drones under aggressive** mo- tion **and broad illumination environments, many existing VIO systems fail to meet these requirements due to the inherent limitations of standard cameras. Event** cam- eras **are bio-inspired sensors that capture pixel-level illumination changes instead of the intensity image with a fixed frame rate, which can provide reliable visual** percep- tion **during high-speed motions and in high dynamic range** (HDR) **scenarios**. There- fore, **developing state estimation** algorithms **based on event cameras offers exciting opportunities for robotics. However, adopting event cameras is challenging due to** 3.1. Introduction **the event streams being composed of asynchronous events which are fundamentally different from the synchronous intensity images. Moreover, event cameras** 2

are motion- activated, this means that they might generate very limited

> **information or even noise when the relative motion between the camera and the scene is limited, such as in a still state, while standard cameras can provide rich perception information in most** 5

scenar- ios. On the other hand, relying solely on low-cost IMU, which often has significant bias, usually struggle to handle some special scenarios, such as quadrotor suddenly hover- ing after high-speed flight. Therefore, even though the proposed Mono-EIO pipeline in Chapter 2 has demonstrated good performance in various challenge situations, further exploration of heterogeneous multi-modal visual sensors and isomeric visual features remains essential.

> **Most of the research in both image and event-based SLAM/VO/VIO** relies **on point-based features, while it is** important **to note that human-made structures often exhibit regular geometric shapes, such as lines or planes**. Besides, **point-based** fea- tures are **not** 2

robust in low-texture environments such as corridors [183]. Therefore,

> **point-based features** may **not** always be **the** optimal **representation for visual tracking in all scenarios**. Although **point-based features are more common in natural scenes** 34

,

> **performance degeneracy might occur when only using point-based** 2

features

. In con- trast,

line-based features can provide **more geometric structure information than point- based features** 5

[184, 185, 186], complementing the sparsity of point features and facil- itating the mapping of the surrounding environment structure. Moreover, event cam- eras are particularly

well-suited for line-**based** motion estimation **due to their ability to** 20

respond primarily to edges formed by strong gradients. This inherent edge-centric response highlights the need to reconsider the usage of

line-based features within **the event-based** 4

SLAM systems.

Therefore, for heterogeneous event-based information uti- lization, **we design and extract the line-based feature in the event stream to improve the performance of purely point-based features** 2

. As

can be seen in Fig. 3.2(**b), the** 82

in- tegration

of the line-based feature and point-based feature can further ensure a more uniform distribution of the features and provide additional constraints on scene struc- ture. **In addition, compared to standard cameras, event cameras are capable of** provid- ing **reliable visual perception during high-speed motion and HDR scenarios. However** 2

, (a) (b) Figure 3.2: (

a) Our PL-EVIO combines events, images, and IMU to provide robust state estimation during aggressive motion. It can provide onboard feedback-control for quadrotors with limited computational resources. (b) Our PL-EVIO in the outdoor environment. Left: event-corner features in the event; Middle: line-based features in the event; Right: point-based features in the image 2

. when the event

camera and the scene have restricted relative motion, such as in a static state, event cameras may produce limited information **or even noise. Although the standard** camera encounters **difficulties during high-speed motion or in** HDR scenar- ios, **it can provide rich intensity** values **of the** scenes under **uniform motion or favorable lighting conditions. Observing this** 4

complementarity, combining events and images modality improves accuracy and robustness across diverse scenarios. For example,

> **as illustrated in Fig. 3**.2(**b** 62

), intense sunlight

> **in the** central **field of view (FoV** 62

) creates an HDR challenge that reduces point-based image features, whereas event cameras de- tect rich event-corner features in the same region. Conversely, in low-texture areas (the periphery of the FoV), event cameras struggle to provide information, while stan- dard cameras provide rich point-based features using intensity information. Therefore, events and images serve as complementary sources of visual information. Combining them has been proven useful to improve the accuracy and/or robustness in several applications, such as feature tracking [5, 110], ego-motion estimation [49, 67], depth prediction [127], video reconstruction [187], video frame interpolation [188], etc. Thus, 3.1. Introduction we also seek to get the best of both visual sensors for SLAM tasks. This chapter proposes a monocular event-image-inertial odometry (EVIO)

> **for a sensor setup that includes event, image, and inertial measurement unit (IMU) data, with a well-designed feature management system. Our** EVIO framework **includes the purely event-based VIO (EIO), and the event with image-based VIO (EVIO). More specifically, we first implement a motion compensation algorithm using the IMU data to correct the motion of each event according to its** individual **timestamp, including** ro- tation **and translation motion, into the same timestamp. After that**, we utilize **the event- corner features detection and tracking approach developed in our** previous Mono-**EIO work** [2]. **We conduct an EIO framework, including the line-based event features and the event-corner point** features, **termed PL-EIO (Event+IMU), to perform robust state estimation. Finally, we integrate image measurements into our PL-EIO framework as the PL-EVIO (Event+Image+IMU), in which visual landmarks include event-corner features, line-based event features, and point-based image features. These three kinds of features are well integrated together to leverage additional structure or constraint information for more accurate and robust state estimation** 2

. Through leveraging these hierarchical and heterogeneous visual features,

> **our system can provide robust and** re- liable **state estimation in challenging environments. The efficiency of our system is adequate to achieve real-time operation on platforms with limited resources, such as providing onboard pose feedback for quadrotor flights** 2

.

> **The contributions of this chapter are summarized as follows: 1** 94

.

> **To address the performance degradation when** only using **point-based features in human-made structures, we design the line-based feature and descriptor** in event-**based representation for front-end** incremental estimation. **We** proposed the **PL** 4

-EIO

> **pipeline, which tightly fuses the event-corner features** and **line-based event features** 2

> **together, to** provide **additional structure or constraint information for more accurate and robust state estimation** 2

. 2. By leveraging the complementary advantage between event and image measure- ments, we design

> **the PL-EVIO pipeline, which** integrates **the** 2

complementary ad- vantage between event and image measurements to provide robust and reliable state

> **estimation. 3. We validate that our PL-EVIO can achieve state-of-the-art performance in** differ- ent **challenging datasets. It** can **also be used as onboard pose feedback control for the quadrotor to achieve aggressive motion, e.g., flip** 2

(shown in Fig. 3.2(a)). 3.2 Related Works 3.2.1

> **Event-based Representation and Feature** Extraction Event **cameras are motion-activated sensors that** capture pixel-**level illumination changes instead of** an intensity image **with a fixed frame rate. An event is triggered** only when **the intensity of an individual pixel varies beyond** a specific **threshold Tthreshold, which can be represented** as the **spatio-temporal coordinates of the intensity change** and its **sign: e = {t, x, y, p} ⇔ I(x, y, t + △t) − I(x, y, t) = p · Tthreshold** (3.1) **where t is the timestamp that the intensity of a** pixel I (**x, y) changes, and p is the** po- larity **that indicates** the direction **of the intensity change. The generation model** of the **event stream endowed some good properties, which** also allow **the event camera to confer robustness to vision**-based localization **in challenging scenarios. However, adopting** the event **camera into the SLAM/VO/VIO is a very** challenging task **since the event streams are in** an **asynchronous** format which **is fundamentally different from the synchronous** image **data. Therefore, most methods and concepts** developed for **conventional image-based cameras can not be** directly applied. **To enable the** asyn- chronous **event data** to the synchronous **data representation, different kinds of** event representations **have been proposed: (i) The first method is directly working on the** raw event **stream without any frame-like accumulation. Ref**. [189] proposes **a feature tracker that** employs **the** 4

de- scriptors

> **for event data. Ref**. [190] presents **a feature tracker based on Expectation Maximisation (EM). Ref** 5

. [191, 55] extracts

> **the line feature from the raw asynchronous events. There are several other ways to represent the raw event, such as Voxel Grid or Event** 2

spike tensor [192]. However, these higher-dimensional or learning-based event representations will not be discussed here

. 3.2. Related Works (

ii) The second approach is to combine with the image sensor or generate the inten- sity image from the event through learning-based methods. Ref. [37, 5] firstly detects the features on the grayscale image frames, and then tracks the features asynchronously using event streams. (iii) The third representation is the motion-compensated event image, or edge image, which is generated by aggregating a group of neighboring events within the spatio-temporal window into an edge image. Ref. [44, 49] adopts the conventional corner detection algorithms, such as FAST corners [108] or Shi-Tomasi [166] for feature detection, and the Lucas Kanade (LK) optical flow [109] for feature tracking in the event image. (iv) The last method is the time surface (TS) or Surface of Active Event (SAE), which is a 2D map where each pixel stores the time value. It can summarize and update the event stream at any given instant or encode the spatio-temporal constraints of the historical events. Using an exponential decay kernel, TS can emphasize recent events over past events

5

[169].

tlast is the timestamp of the last event at each pixel coordinate x = (u, v)T, the TS at time t ≥ tlast(x) is defined by: T(x, t) = exp(− t − tlast(x)) η (3.2) where η is the decay rate parameter. Ref. [193, 168] use the SAE or TS to inspect previ- ously triggered events in the stream and the adjacent pixels for classifying a new event as an event-corner

4

. 3.2.2

Event-based Motion Estimation Event-based state estimation has been extensively developed to handle challenging scenes in recent years, particularly in scenarios where traditional cameras struggle to perform well, such as high-speed motion estimation or HDR perception. Ref. [32] proposes the first event-based SLAM system, which is limited to tracking planar mo- tions while reconstructing the 2D ceiling map with an upward-looking event camera

2

. Ref. [33, 37] proposes

the event-based VO to track camera motion. However, these methods still relied on the standard camera, which was still susceptible to motion blur and low dynamic range. The first purely event-based 6-DoF (Degree-of-Freedom) VO is presented in [38], which performs real-time event-based SLAM through three decou- pled probabilistic filters that jointly estimate the 6-DoF camera pose, 3D map of the scene, and image intensity. However, it is computationally expensive and requires a GPU to achieve real-time performance. EVO [40] is proposed to solve the SLAM prob- lem without recovering image intensity, thus reducing computational complexity, and it can run in real-time on a standard CPU. It performs a tracking approach based on image-to-model alignment and adopts the 3D reconstruction

2

method from EMVS [41] to perform the mapping. However, the EVO is needed to run in the scene that is planar to the sensor, for up to several seconds, for bootstrapping the system. ESVO [59] is the first stereo event-based VO method, which follows a parallel tracking-and-mapping scheme to estimate the ego-motion and the semi-dense 3D map of the scene. However, it barely operates in real-time in DAVIS346 (346*260) and also faces limitations due to rigorous initialization as well as unreliable pose tracking. Ref. [60] proposes stereo VO for event cameras based on features. The pose estimation is done by re-projection er- ror minimization, while the features are stereo and temporally matched through the consecutive left and right event TS. It solves the problems of ESVO mentioned above. However, it still cannot operate in real-time in high-resolution event cameras (640*480). The robustness of event-based SLAM/VO systems can be improved by incorpo- rating IMU measurements. The first EIO method is proposed in Ref. [43] which fuses a purely event-based tracking algorithm with pre-integration IMU measurement through the Extended Kalman Filter. Another EIO method is proposed in Ref. [44]. It detects and tracks the features in the edge image, which is generated from motion-compensated event streams, through traditional image-based feature detection and tracking meth- ods. Finally, the tracked features are combined with IMU measurement using keyframe-based nonlinear optimization. The authors extend their method to leverage the com- plementary advantages of both standard and event cameras in Ultimate-SLAM [49] to fuse image-like event representations, standard image frames, and IMU. To some extent, these methods use the edge image to realize VIO, this might introduce bottle- necks since it requires substantial parameter adjustments depending on the varying number of generated events in the scene. EKLT-VIO [66] combines the event-based tracker [5] as the front-end with a filter-based back end to perform the EVIO for Mars- like sequences. However, it is pretty hard to perform in real-time even in the lowest

3.2. Related Works resolution event camera. Ref. [48] proposes

to fuse events and IMU measurements into a continuous-time framework. While their approach cannot achieve real-time since the expensive optimization is required to update the spline parameters upon receiving every event [44]. In our previous work [2], we propose a monocular EIO which the event-corner features with IMU measurements to provide real-time 6-DoF state esti- mation even in high-resolution event cameras. Furthermore, this EIO framework can bootstrap from unknown initial states and can ensure global consistency thanks to the loop closure function. Nonetheless, it still cannot provide onboard pose estimation for closed-loop control of the quadrotor flight since event cameras produce minimal infor- mation or noise when stationary. Recently, there have been several studies focusing on stereo EVIO [73, 74]. There are several works in event-based vision that utilize line features. IDOL [55] calculates the normal vectors in the spatio-temporal space for each incoming event by utilizing a local neighborhood. Events with similar normal vectors are clustered together to form lines, and an EIO algorithm uses these detected lines and inertial measurements to estimate camera poses. However, this approach assumes that lines move at nearly a constant speed in short intervals, leading to the aggressive motion being avoided in their validation experiments and loss of the advantages of event cam- eras. What's more, this method lacks real-time capabilities even with low-resolution event cameras (240*180). Ref. [69] employs the Hough

transformation on spatial im- ages generated **from a 3D point-based map to cluster event data into a collection of 3D lines. These lines are subsequently integrated into the Kalman filter to estimate the 3D lines and camera pose. However, their event-to-line matching method suffers from the sudden surge of incoming events** [111] **caused by aggressive motion, scene** com- plexity, **and sudden illumination changes. Additionally, this approach is sensitive to event sparsity and requires at least 6 non-parallel 3D lines, a known-scale predefined marker, or ground-truth pose readings for**

system bootstrapping. Both spatio-temporal relationship [55] **and Hough transformation** [69] **are** utilized in **event data to cluster events that belong to the same** straight lines. **In contrast, our approach utilizes the line** segment detector (**LSD**) [194] **to extract event-based line features** and strike **a balance between performance and** computational efficiency. **Unlike these two methods that solely rely on** the line **feature and may be susceptible to high levels of** texture in **the scene, our method leverages the complementarity** of point **and line features to enhance its robustness. Moreover, Ref**. [191] utilizes **event cameras for powerline tracking**. Their method involves **detecting planes in the spatio**-temporal signal **to identify lines in the event streams and** subsequently incorporating **events into these lines while tracking** them over **time. However, their approach is restricted to** powerline inspection **tasks and does not involve the data association** of event-**based line features or utilization for incremental**

4

pose estimation. 3.3 **Framework Overview The structure of our proposed method is illustrated in Fig**. 3.3, **which is composed of two sections: (i) The EIO Front-end takes the motion-compensated event stream as input and extracts the event-corner features and the line-based event features. There are two kinds of event representations: the TS with polarity**

2

(Eq. (2.3)) and the normal- ized TS without polarity (Eq. (2.4)),

which are generated from the SAE for point & line feature tracking and loop closure detection, respectively. More detailed discussions of these two kinds of event representations can be seen in Chapter 2 **and the**

2

ablation study of this Chapter (Section 3.9.8 and 3.9.9). (

ii) The EIO Back-end tightly fuses the point landmarks, line landmarks, and the IMU pre-integration to estimate the 6-DoF state, while the loop closure is used to eliminate the accumulated drifts. Finally, to achieve low latency, we also directly forward propagate (loosely-coupled) the latest estimation with the IMU measurements to achieve IMU-rate state outputs which can be up to 1000 Hz. This can ensure the requirement of closed-loop autonomous quadrotor flight. For the keyframe in the sliding window, it is selected by two criteria and only based on the event-corner features: (i) When the average parallax of the tracked event- corner features between two consecutive timestamps exceeds a threshold (10 is set in our experiment). (ii) When the number of successfully tracked event-corner features from the last timestamp falls below a certain threshold (20 is set in our experiment). As for the initialization procedure of our framework, which is adopted from

2

[8, 177],

> **our pipeline commences with a vision-only structure from motion (SfM) to** es- tablish **the up-to-scale structure of camera pose and event-corner feature positions. By loosely aligning the SfM with the pre-integrated IMU measurements, it can bootstrap** 2

3.3. Framework Overview Figure 3.3:

> **The framework of our PL-EIO (Event + IMU) and PL-EVIO (Event + Image + IMU). the** 4

system

> **from unknown initial states rather than using marker** [69] **or assuming the local scene is planar to the sensor** [40]. **It is worth mentioning that if the image is** avail- able **in** the framework, **we only employ the point-based image** visual measurement **for SfM initialization to ensure reliable visual-inertial alignment and up-to-scale camera poses. Regarding loop closure, extra event corners are** detected in **the EIO Front-end, subsequently described by the** BRIEF descriptor, **and fed to the Back-end. These** ad- ditional **event-corner features are used to achieve a better recall** rate on **loop detection. Thanks to our designed normalized** TS without **polarity, which is triggered in scenes with** strong edges, **it can eliminate accumulated drifts and ensure** global consistency. **The correspondences are found through** the BRIEF **descriptor matching by calculating** the **Hamming** distance. **When the number of corresponding event** descriptors is **greater than a certain threshold (16-25 in our experiments), the loop closure is detected. After detecting the loop**, the connection **residual of the previous keyframe and the** current keyframe **are integrated into the nonlinear optimization as** a re-**localization residual. We further extend our PL-EIO framework to** include point-**based image features to provide a more robust** state estimation (**PL-EVIO). Fig. 3**.6(**a) shows the** complemen- tarity of **the image and event information. For the** strong lighting region, **the event can provide reliable event-corner features, while the image can provide rich point-based features** in other **areas. This enables the uniform distribution of** the point-**based event and image features in the scene**. The **line-based event feature can provide more** con- straints (**shown in Fig. 3**.6(**b)), even when the successfully** tracked point-**based event and image features are** fewer **in the scene. Our framework can provide a more robust and accurate** state estimation. **More details of event-based point and line** feature detec- tion **and tracking in our framework can be seen in** 4

the ablation study of this Chapter (Section 3.9.8). 3.4

> **Motion Compensation for the Event Stream using IMU Events can be triggered either by moving objects or by the ego-motion of the camera. Similar to Ref**. [195], **we only rely on the IMU for motion compensation, which** guar- antees **efficiency and** real-time ability. **For the new event stream coming, we use the angular velocity and linear acceleration from the IMU, averaged over the time window where the events are grouped in the same event stream, to estimate the ego-rotation and ego-translation of each event. Using this ego-motion to warp the events into the timestamp of the first event in** 2

the same event stream. The motion (considering both rotation R and translation T) of each event can be calculated through: ΔT(Δt) = ⌈ R(ωIMUΔt) T 0 1 ⌉ = ⌈ R(ωIMUΔt) 12αIMUΔt2 ⌉ (3.3) ⌈ ⌉ 0 1 [ ] [ ] [ ]

where ωIMU and αIMU are the angular velocity and linear acceleration measurements from the IMU in the current event stream timestamp. While R(ωIMUΔt) is the ro- tation matrix generated from the angular velocity ωIMU and the time difference Δt. Each event ei = {eti, exi, eyi, epi} of the event stream is then warped by Δ(Δt) = Δ(eti − tfirst_event), where tfirst_event is the timestamp of the first event of the current event stream and eti is the timestamp of event ei 4

.

Fig. 3.4 compares the raw event streams with our motion-compensated 7

ones. While the former exhibits a certain degree of distortion, the latter displays clear and sharp scene contours. More demonstrations 3.5. Point-based Event Measurement of the comparison between before and after warping events with IMU can be found in https://kwanwaipang.github.io/ESVIO. Figure 3.4: Visualization of the event streams before and after applying our proposed motion compensation. 3.5 Point-based Event Measurement

The event-corner feature detection and tracking 4

process follows the same schedule as described in Mono-EIO [2] (see Chapter 2 for details). Briefly, the SAE is

updated through the motion-compensated event stream, and existing event-corner features are tracked using the LK optical flow on the TS with polarity 5

(detailed in Section 2.4.3)

which is generated from the updated SAE (illustrated in Fig. 3.5(c)). A two-way track- ing strategy is employed to track event-corner features between consecutive 5

times- tamps. When

the number of tracked features falls below a threshold (150-250 in our ex- periments), new event-corner features are detected from the latest motion-compensated event stream (as shown in Fig. 3.5(a 5

)). These new features are detected by leveraging the SAE, as detailed in Section 2.5.1. All

event-corner features in the front-end are undistorted using the camera dis- tortion model and projected into a normalized camera coordinate system. Outliers are filtered using the Random Sample Consensus (RANSAC 5

) algorithm. The inverse depth of

successfully tracked features **between consecutive timestamps** is recovered **through triangulation. The 3D** 103

positions of these features are then fed to construct the event- based point constraint through

the re-projection function. The event 4

-based point con- straint formulation remains consistent with that described in Section 2.6.2. Considering (a) (b) (c) Figure 3.5:

The event-corner feature detection and tracking: (a) Detecting features from raw event streams; (b) Using the TS with polarity as the mask for uniform distribution of the event-corner features; (c)Tracking feature in the TS with polarity. the 2

lth

event-corner feature that is first observed in the ith **keyframe, the residual for its observation in the** kth **keyframe is defined as** 1

: eke,vlent = ⌈ ulk vlk ⌉ − πe · (Teb)−1 · Twbk · Tbwi · Teb · π−e1(1 λ ,⌈ uli (3.4) e ⌉)⌊ ⌊ ⌋⌋⌊ vli ⌋⌋ where, uli, vli

is the first observation of the lth **event-corner feature in the ith** T **keyframe**. u[lk, vlk **T] is the observation of the same event-corner feature in the** kth **keyframe**, πe **and πe−1[are the projection and back-projection function of the event camera**, respec-] tively, **which include the intrinsic parameters for the transform between the 2D pixel coordinates and normalized event camera coordinate**. Tbwi **indicates the movement of the body frame related to the world frame in timestamp i**, Twbk **is the transpose of the pose of the body in the world frame in the** kth **keyframe. 3**.6. **Line-based Event Measurement** 2

3.6 Line-based Event Measurement For heterogeneous event-based information utilization, this chapter proposes to

extract the line-based feature in the event stream to improve the performance of point-based event measurement. **The** event-**based** line **features** 2

serve as a valuable supplement to the event-corner features, especially in low-texture environments. More importantly, because lines consist of multiple points, there is a high probability that the character- istics will be maintained even when an illumination change occurs. As can be seen in Fig. 3.6(

b), the integration of the line-based feature and point-based feature can further ensure a more uniform distribution of the features and provide additional constraints on scene structure 2

. (

a) (b) Figure **3**.6: **Three different kinds of features in our PL-EVIO framework: event-corner** features, **event-based line** features, **and image-** 5

based point features. The

animation for three different types of features

can be viewed on the project website https:// kwanwaipang.github.io 101

/PL-EVIO/. 3.6.1 Event-based Line

Feature Detection and Matching Utilizing the line-based features to improve the performance of point-based VIO is ef- fective as line features can provide additional constraints and structure information in the scene, especially for the human-made environment. Therefore, for incoming new event streams, after using motion compensation, the streams are mapped into the OpenCV-Mat format (event mat). Given that events are typically triggered in scenes with strong edges, generating line features using the event matrix can prevent the gen- eration of invalid line features and enhance efficiency. To efficiently extract line-based features and descriptors from the raw event streams, we have modified the LSD algo- rithm [194] in OpenCV. Utilizing the Sobel filter, we compute the orientation of each event in the event mat and group events with similar angles into a line support region. Additionally, we have studied the hidden parameter tuning and length rejection strat- egy of the LSD algorithm, drawing inspiration from [186] to filter out short line features using a length rejection strategy: Lmin = η · min(W, H) (3.5) where min(W, H) denotes the smaller value between the width and the height of the event camera. η is the ratio factor (0.125 is set in our experiments). After that, we adopt the Line Band Descriptor (LBD) [196] to describe and match line features, respectively. In particular, to ensure good tracking performance and be consistent with point-based event-corner features, we also use the TS with polarity for LBD generation and line- based feature matching. We further execute the line features refinement schemes to identify line features as good matches for successful line tracking: • The Hamming distance between matching line features is less than 30; • The square error of the endpoint between the matching line features is less than 20 * 20 pixel2; • The angle between matching line features is less than 0.1 rad; The successfully tracked line-based event features would be further refined by undistorting the endpoints of the lines and projected onto a unit sphere after pass- ing outlier rejection. The outlier rejection is performed using RANSAC with a funda- mental matrix model. Then, we obtain the line-based landmark by triangulating the correspondences of two line features. The line-based landmark whose 3D position has been successfully calculated would be fed to the sliding windows for the nonlinear optimization 4

. 3.6.2 Event-based Line Constraint

The line re-projection residual is modeled as the distance from the endpoints of the line to the projected line in the normalized image plane. The lth line-based landmarks in 2

3.6. Line-based Event Measurement T

the world frame can be defined using the Plücker Coordinate: Llw = nlw, dlw , nlw denotes the normal vector of the plane determined by Llw 2

and the origin of the world [ ] frame, while dlw denotes the direction vector determined by the two endpoints of Llw. Given the transformation matrix

Twbk = Rwbk, twbk

indicates the movement of the body frame related to the world frame in timestamp k, we can obtain the transformation from [ ] the world frame to the event frame in timestamp k through 2

Twek = Tbe · Twbk, where Tbe = [Reb, tbe]

is the extrinsic transformation from the body (IMU) frame b to the event camera frame e. Then, we can transform the lth line-based event feature Llw in kth **keyframe from world frame to event camera frame by** 2

[197, 198]: Lelk = ⌈ nlek ⌉=⌈ Rewk [twek] × Rwek ⌉⌈ nlw ⌉⌊ delk ⌋⌊ 0 Rwek ⌋⌊ dlw (3.6) ⌋⌊ ⌋⌊ ⌋ where Rwek = Reb · Rbwk, tewk = Reb · twbk + tbe.

The transformation for the Plücker Coordinates of the lth line-**based event feature from ith keyframe to kth** keyframe in **the body frame can be represented as follows: Llbk** 4

= ⌈ nlbk dlbk ⌉ = ⌈ Rbbki 0 [tbbik] × Rbbki ⌉⌈ nlbi ⌉ (3.7) ⌊ Rbbki ⌊ ⌋⌋⌊ ⌊ ⌋⌋ dlbi ⌋⌋⌊ ⌋ where Tbbik = Rbbki , tbbik

indicates the movement of the body frame **related to the world frame in ith keyframe to** kth keyframe. [ ] **The Plücker Coordinates Llw can be represented** using a **four-parameter** orthonor- mal **representation, known for** its superior **convergence performance**[197]. **As a result**, we transfer **the line-based landmark to the four-parameter** orthonormal **representation for the optimization process**. The orthonormal **representation (U, W) ∈ (SO(3), SO(2**)) of the **Plücker Coordinates Llw can be computed using the** 4

QR decomposition [197, 186]: w1 0 [nwl|dwl] = U ⌈ 0 w ⌉, set : W = ⌈ w1 −w2 2 ⌉ (3.8) 0 0 ⌊ w2 w1 ⌊⌋⌋⌊⌋⌋

where U and W denote a three and a two dimensional rotation **matrix, respectively. Let R(θ) = U and R(o) = W be the corresponding rotation transformations**, where U = [**u1, u2, u3]. With this notation, we can now** express the **relationship as follows: R(θ) = U** = nwl dlw **nlw** × dlw [||**nlw|| ||dlw**|| ||nlw × **dlw** 4

||] , , (3.9)

R(o) = W = ⌈ **cos(o) −sin(o)** ⌉⌊ **sin(o) cos(o** 4

) ⌊ ⌋⌋ (3.10) = 1 ||nwl|| −||dwl|| ||nlw||2 + ||dlw||2 ⌈ ||dwl|| ||nwl|| ⌉√ ⌊ ⌋⌋

Up to this point, we have established the connection between the **four-parameter** or- thonormal **representation** ϕmtlihne = [**θT, o] of Eq.(5**.11) **and the Plücker Coordinates Llw. The Plücker Coordinates Llek in the event camera** frame can **be obtained from Llw through Eq**.(3.6), **and then can** be projected **to the line** lelk **in the event imaging plane** by T lelk **l1, l2, l3 = πenlek** = 4

(3.11) [ ] **where πe is the projection function of the event camera**, and the **nlek can be obtained from Eq**.(3.6). **The line re**

-projection error

can be defined as: ekli,nle =⌈ **d(Sllek** , lelk) **d** 4

(Elelk , lelk)⌋⌉(3.

12)⌋⌊⌊ **where** Sllek **and** Elelk **are the homogeneous coordinates of the endpoints of the line feature** lelk **in the image plane, and d(m**, lelk) **denotes the point-to-line distance function from the** endpoints **to the projection line** llek : **d(m** 5

, lelk) = mlelk l21 + l22 (3.13) Slelk = (ukl,S, vkl,S, 1), Elelk = (ukl,E, vkl,E, 1) √ 3.7. Point-based Image Measurements 3.7 Point-based Image Measurements For each incoming image frame, existing point-based image

features are tracked using KLT sparse optical flow 133

[109]. After that, new corner features are detected using the Shi-Tomasi algorithm [166] to ensure the image contains 150–250 features, consistent with

the number of event-corner features 2

.

To enforce a **uniform** feature **distribution, a minimum** 2

pixel separation is maintained between neighboring detected features. While the processes for outlier rejection and triangulation for

the point-based image features follow **the** 2

methods outlined in Chapter 2. Regarding the image-based point constraint, similar to

the event-corner measure- ment **residual from the re-projection function** (Eq. 3.4), **the** 5

lth

point-based image fea- ture **that is first observed in the** ith **keyframe, the residual for its observation in the** kth **keyframe is defined as** 5

: eikm,lage =⌈ ulk ⌉ − πc · (Tcb)−1 · Twbk · Tbwi · Tcb · πc−1(λ1c , ⌈ uli vlk ⌊ vli ⌉) (3.14)⌊⌊ ⌋⌊⌊ ⌋⌊

where, uli, vli **is the first observation of the lth point-based image feature in the ith** T keyfram[e. **ulk**], vlk **T is the observation of the same point-based image feature in the** kth **keyframe. πc and]πc−1 are the projection and back-projection function of the standard [camera, respectively, which** 2

include the intrinsic parameters for the transform between the 2D pixel coordinates and normalized camera coordinate

. 3.8 Formulation of the PL-EIO and PL-EVIO

The full state vector in the sliding windows is defined as: χ = [χb, λe, ϕl, λc 2

, Tcb, Teb] (3.15)

where λe = [λ0,...,λmetvhent], and λc = [λ0,...,λmtimhage] is the inverse depth of the mtehvent event-corner features and mtimhage point-based image features, respectively, while ϕl = [ϕ0,...,ϕmtlhine], ϕmtlihne = [θT,o] is the four-parameter orthonormal representation (as shown in Eq.(3.9) and Eq. (3.10)) of the mtlhine line-based event features, in the sliding windows. Tcb = [Rbc, tbc] or Teb = [Rbe, tbe] is the extrinsic transformation from camera frame (the image c or event e) to the body (IMU) frame b (Tcb = Teb when using the DAVIS which can simultaneously output the image and event data); χb = [X1, ..., XK] is the optimization vector in the sliding windows, which comprises the state of the IMU, with K (K = 10 in our experiments), the total number of keyframes in the sliding windows. The system state Xk at kth keyframe is given by the position pwbk, orientation quaternion qwbk, and the velocity vwbk of the IMU in the world frame, and the accelerom- eter bias bak and gyroscope bias bgk as follows: Xk = [pwbk,qwbk,vbwk,bak,bgk] (3.16) Joint nonlinear optimization is solved for the maximum a posteriori estimation of χ, while the cost function can be written as: J(χ 4

) = ∑ ∑ ||eke,vlent||W2ekvent + ∑ ∑ ||ekli,nle||2Wlkine K−1 K−1

k=0 l∈ζ k=0 l∈ℓ + K∑−1 ||ekimu||2Wikmu + K 5

∑−1 ∑ ||ekim,lage||2Wikmage (3.17) k=0

k=0 l∈ξ + ||em||2Wm + ||er||2Wr Eq. (3.17) contains the point-based event residual ekev,lent with weight 4

Wekvent (de- tailed in Eq. (3.4));

the line-based event residual ekli,nle with weight 4

Wlkine (detailed in Eq. (3.12));

the IMU pre-integration residuals eikmu with weight 4

Wikmu (detailed in Eq. (2.15));

the point-based image residual ekim,lage with weight 2

Wikmage (detailed in Eq. (3.14));

the marginalization residuals em with weight Wm; the re-localization residuals er with weight Wr; while ζ, ℓ, and ξ are the set of event-corner features, line-based event features, and point-based event features, respectively, which have been successfully tracked or matched at least twice in the current sliding window 2

. 3.9 Experiments

In this section, we evaluate the effectiveness of our framework in various challenging sequences using both quantitative and qualitative methods in subsection 2

3.9.1 and 3.9.2. 3.9. Experiments

We implemented our method with C++ in Ubuntu 20.04 and ROS Noetic. All sequences are evaluated in real-time using a laptop with Intel Core i7-11800H and are recorded in videos (shown on our project website). In subsection 34

3.9.3 and 3.9.4,

we demonstrate the quadrotor flight using our method for the closed-loop state estimator and aggres- sive flip. Meanwhile, large-scale experiments are carried out to illustrate the long-time practicability in subsection 2

3.9.6. 3.9.1 Evaluation in HDR Scenarios

For demonstrating the robustness, accuracy, and real-time capability, we initially evalu- ate our PL-EIO using different resolution event cameras (DAVIS346 (346*260) and DVX- plorer (640*480)) with the ground truth from VICON. All sequences 5

of the Mono-HKU dataset

are recorded in broad illumination range conditions or under aggressive mo- tion. Without loss of generality, we use the 4

raw image from DAVIS346 to run the VINS- MONO [8], PL-VINS [186], and ORB-SLAM3 [9], as image-based comparisons. In ad- dition, based on the source code of Ultimate SLAM [49], we also test the EVIO and EIO versions of Ultimate SLAM for event-based comparison. The estimated and ground-truth trajectories are aligned with a 6-DOF transformation (in SE3), using 5 seconds [0- 5s] of the resulting trajectory. We compute the mean position error (Euclidean distance in meters) as a percentage of the total travel distance of the ground truth, which is cal- culated by the publicly available tool [182]. As can be seen from the results in Table 3.1, our PL-EIO has better performances compared with the other methods in different res- olution event cameras. Especially, for the results of vicon_aggressive_hdr, our PL-EIO produces reliable and accurate pose estimation even when the image-based VIO and VO fail. Besides, compared with our previous Mono-EIO [2], the 2

introduction of the line feature, known as PL-EIO, demonstrates significant performance improvements across different resolution event cameras. While the performance **of PL-EVIO, which incorporates image measurements, surpasses our** Mono-**EIO** [2]. **Our experimental** ob- servations **indicate that although the image-aid one (PL-EVIO) exhibits notable** perfor- mance **gains in most sequences, it underperforms in low-light environments such as vicon_dark1 and vicon_dark2), as compared to PL-EIO. This could be attributed to the degradation of point-based image feature tracking in dark environments**

. Table 3.1:

Accuracy Comparison of Our PL-EIO with Other Image-based or Event-based VIO Works 2

.

Unit:%/m, 0.45 means the average error would be 0.45m for 100m motion 5

. Resolution Algorithm hdr1 hdr2 hdr3 hdr4 darkto light1 Sequences (vicon_*) darkto light2 light light todark1 todark2 dark1 dark2 aggressive _hdr Average DAVIS346 (346*260) VINS-MONO [8] ORB-SLAM3 [9] PL-VINS [186] USLAM [49] USLAM [49] Mono-EIO [2]

Our PL-EIO Our PL-EIO+ Our PL-EVIO **VIO VO VIO** 5

EIO EVIO EIO EIO EIO EVIO **0.96 0.32 0.67 1.49 2.44 0.59 0.67 0.57 0.17** 1.60 **0**.75 **0** 5

.90 1.28 1.11 0.74 0.45 0.54 0.12 2.28 0.60 0.69 0.66 0.83 0.72 0.74 0.69 0.19 1.40 0.70 0.66 1.84 1.49 0.37 0.37 0.32 0.11 0.51 0.75 0.84 1.33 1.00 0.81 0.78 0.66 0.14 0.98 0.76 1.50 1.48 0.79 0.42 0.44 0.51 0.12 0.55 0.41 0.64 1.79 0.84 0.29 0.42 0.33 0.13 0.55 0.58 0.93 1.32 1.49 0.79 0.73 0.53 0.16 0.88 failed 0.53 1.75 3.45 1.02 0.64 0.35 0.43 0.52 0.60 failed 1.10 0.63 0.49 0.30 0.38 0.47

failed failed 1.94 failed 2.30 0.66 0.62 0.50 1.97 4

1.02 0.61 0.93 1.40 1.49 0.63 0.56 0.49 0.36 DVXplorer (640*480) USLAM [49] Mono-EIO [2]

Our PL-EIO Our PL-EIO+ EIO EIO EIO EIO 5

1.94 0.30 0.47 0.41 2.38 0.37 0.22 0.21 0.83 0.69 0.47 0.36 2.09 0.26 0.27 0.25 1.96 0.80 0.71 0.71 1.57 0.57 0.56 0.47 2.48 0.81 0.43 0.54 1.37 0.75 0.67 0.60 3.79 0.35 0.51 0.41 2.81 0.41 0.38 0.41 failed 0.65 0.62 0.50 2.12 0.54 0.48 0.45

Regarding the proposed motion compensation algorithm, as evident from the re- sults, **the motion compensation** version (**PL-EIO+) does not exhibit significant** enhance- ments **across various sequences, particularly in scenarios involving aggressive motion. This outcome could potentially stem from biases present in the IMU during such** ag- gressive **motion. On the other hand, in this evaluation, the event stream rate is 60Hz for DAVIS346 and 50Hz for DVXplorer. Such high frequencies reduced time differences within the same event stream, Additionally, we observe that motion compensation for** 2

event streams may not be an optimal choice for high-resolution event cameras due to the trade-off between computational burden and performance improvement. It is worth mentioning that Ultimate-SLAM is provided primarily **for reference**

, as achieving failure-free operation across different sequences is already a significant challenge, even with extensive parameter tuning.

Since the illumination would change greatly in our dataset, it is very difficult for Ultimate-SLAM to choose a certain sta- tionary threshold to integrate the event stream into the edge image. We have tried our best to fine-tune the parameters of Ultimate-SLAM in sequence vicon_hdr3 to achieve good performance and use the same parameters to evaluate other sequences. This also shows that the generalization ability to integrate the event streams into the edge-image for VIO is pretty bad since the number of triggered events depends on many factors, including the resolution of the camera, the texture of the scene, **the illumination, etc** 5

. 3.9.2

Evaluation in Aggressive Motions In this section, we evaluate our PL-EVIO in UZH-FPV dataset [15], **which is a high- speed, aggressive visual-inertial odometry dataset. This dataset includes fast laps around a** racetrack **with drone racing gates, as well as free-form trajectories around obstacles. We compare our PL-EVIO with ORB-SLAM3 (stereo VIO**) [9], **VINS-Fusion (stereo VIO**) [199], **VINS-MONO (monocular VIO**) [8], **and Ultimate SLAM (EVIO**) [49]. **We also computed the mean position error as** a percentage **of the total traveled distance, while the estimated trajectories and ground-truth were aligned in SE3 with all** align- ments. **As can be seen from the results in Table** 3.2, **our proposed PL-EVIO** achieves **better performance even compared with the stereo VIO using a higher resolution** cam- era. **This dataset is so challenging that most of the sequences using Ultimate-SLAM and VINS-Fusion failed, while our PI-EVIO still can provide reliable and satisfying** re-sults. **To achieve optimal performance, deep fine-tuning of parameters is also required for VINS-MONO** 2

.

Table 3.2: **Accuracy Comparison of Our PL-EVIO** with Other **Image/Event-based VIO in UZH-FPV Dataset** 4

[15].

Unit:%/m, 0.70 means the average error would be 0.70m for 100m motion 5

.

Snapdragon (640*480) DAVIS346 (346*260) **Sequence VINS-Fusion** [199] **ORB-SLAM3** [9] **Stereo VIO** Stereo VIO **VINS-MONO** [8] **Ultimate SLAM** [49] **Our PL-EVIO** VIO **EVIO** EVIO **Indoor_forward_3 0.84 0.55 0.65 failed 0.38 Indoor_forward_5 failed 1.19 1.07 failed 0.90 Indoor_forward_6 1.45 failed** 5

0.25 failed 0.30 Indoor_forward_7 0.61 0.36 0.37 failed 0.55 Indoor_forward_9 2.87 0.77 0.51 failed 0.44 Indoor_forward_10 4.48 1.02 0.92 failed 1.06 Indoor_45_degree_2 failed 2.18 0.53 failed 0.55 Indoor_45_degree_4 failed 1.53 1.72 9.79 1.30 Indoor_45_degree_9 failed 0.49 1.25 4.74 0.76 Average 5.26 2.10 0.81 7.26 0.70

Furthermore, we also evaluate our PL-EVIO with the other EIO works in publicly available Event Camera Datasets [144], **which is acquired by the DAVIS240C (240*180, event-sensor, image-sensor, IMU sensor). It contains extremely fast 6-Dof motion and** 2

Table 3.3:

Accuracy Comparison of Our PL-EVIO with Other EIO/EVIO Works in DAVIS240c Dataset 5

[144].

Unit:%/m, 0.24 means the average error would be 0.24m for 100m motion 5

. Sequence Ref.[43] Ref. [44] Ref. [49] Ref. [49] Ref. [181] Ref. [66] Our EIO [2] Our PL-EVIO EIO EIO EIO EVIO EIO EVIO EIO EVIO

boxes_translation hdr_boxes boxes_6dof dynamic_translation dynamic_6dof poster_translation hdr_poster poster_6dof 38

2.69 0.57 0.76 0.27 2.55 0.48 0.34 0.06 1.23 0.92 0.67 0.37 1.75 0.46 0.40 0.10 3.61 0.69 0.44 0.30 2.03 0.84 0.61 0.21 1.90 0.47 0.59 0.18 1.32 0.40 0.26 0.24 4.07 0.54 0.38 0.19 0.52 0.79 0.43 0.48 0.94 0.89 0.15 0.12 1.34 0.35 0.40 0.54 2.63 0.59 0.49 0.31 0.57 0.65 0.40 0.12 3.56 0.82 0.30 0.28 1.50 0.35 0.26 0.14 Average 2.58 0.69 0.47 0.25 1.45 0.54 0.39 0.24

scenes with HDR. We directly report **the raw result in Ref** 5

. [43, 44, 49, 181, 66, 2].

As can be seen from Table 3.3, **our PL-EVIO achieves state-of-the-art performance. Fig**. 3.7 **presents the relative error of our PL-EVIO against other methods, for the** se- quence **box_translation, dynamic_translation, and poster_6dof. It's important to note that, although the Ultimate-SLAM** [49] (**EVIO version) demonstrates performance similar to ours, it relies on different parameters for different** sequences. **While we consider** pa- rameter **tuning to be impractical, we evaluate our methods using fixed parameters for various sequences during the evaluations** 2

. 3.9.3

Online Quadrotor-flight Evaluation To further demonstrate the capabilities of our PL-EVIO, we perform real-world 2

exper- iments

on a self-designed quadrotor platform (shown in Fig. 3.8), carrying a forward- looking IniVation DAVIS346 sensor. An Intel NUC10i7FNH computer running Ubuntu 20.04 is mounted on our quadrotor for onboard computational support. We use Pixracer (FMUv4) autopilot to run the PX4 flight stack. To alleviate disturbance from the mo- tion capture system's infrared light on the event camera, we add an infrared filter on the lens surface of the DAVIS346 camera. Note that the introduction of the infrared filter might cause the degradation of perception for both the event and image camera during the evaluation 34

.

The overall weight of our quadrotor is 1.364kg (GS330 frame with T-Motor F60). In the experiments, the reference trajectories are generated offline. The polyno- mial trajectory generation method [200] is used to ensure the motion feasibility of the 5

(

a) boxes_translation (b) dynamic_translation (c) poster_6dof Figure 3.7: The relative pose error comparison of our PL-EVIO with EIO 34

[44], Ultimate- SLAM [49], and our Mono-EIO [2] quadrotor. To follow

the generated trajectory, a cascaded feed-forward P.I.D. controller is constructed as a high-level position controller running on NUC. Given the position, velocity, and acceleration as inputs, the high-level feed-forward controller computes the desired attitude and throttle sent to the low-level controller running on PX4. We conduct four flight experiments to test the performance of autonomous tra- jectory tracking using our PL-EVIO. The quadrotor is commanded to track different patterns as follows (Offboard and Onboard mean using the VICON and our PL-EVIO as pose feedback control, respectively, while our PL-EVIO runs real-time and online calculations in the onboard computer 2

): Figure 3.8:

Our self-designed quadrotor platform. Figure 3.9: The estimated trajectory of our PL-EVIO on the quadrotor flight and its comparison against the ground truth (Taking the Onboard_test_1 as an example 2

).

Offboard_test_1 and Onboard_test_1 The states estimate from the VICON (Off- board_test_1) and our PL-EVIO (Onboard_test_1 4

)

are used for feedback control of the quadrotor which is commanded to track a figure-eight pattern with each circle being 0.625m in radius and 1.2m in height, shown in Fig.3.9. The yaw angle of the com- manded figure- 5

eight pattern is fixed. The quadrotor follows this trajectory ten times continuously during the experiment. The 1000-HZ online calculation of our PL-EVIO is also recorded for accuracy comparison. Onboard_test_2 The states estimate from our PL-EVIO are used for feedback con- trol of the quadrotor which is commanded to track a screw pattern

shown in Fig.3.10. The quadrotor follows this trajectory ten times continuously during the experiment 5

. Table 3.4:

Accuracy Comparison of Our PL-EVIO with Groundtruth in Quadrotor flight 2

.

Unit: m for translation and deg for rotation 5

. Sequence Translation Error Rotation Error Mean

RMSE Std Mean RMSE Std Offboard_test_1 0.054 0.061 0.028 0.094 0.095 0.015 Onboard_test_1 0.078 0.084 0.030 0.078 0.087 0.039 Onboard_test_2 0.081 0.093 0.046 0.056 0.059 0 4

.019

The 1000-HZ onboard state estimates of our PL-EVIO enable real-time feedback control of the quadrotor. The ground 2

truth is obtained from VICON. The translation and rota- tion error are shown in Table 3.4. Taking the Onboard_test_1 as an example, in Fig. 3.9 and Fig. 3.11, we further illustrate the estimated trajectories (translation and rotation) of our PL-EVIO against the ground truth, as well as their corresponding errors. The total trajectory length is 101.15m. The translation errors in the X, Y, and Z dimensions are all within 0.1m, while the rotation error of the Roll and Pitch dimensions are within 2◦, and the one in the Yaw dimension is within 6 2

◦. Figure 3.10:

Onboard quadrotor flight in screw pattern using our PL-EVIO as feedback control 2

3.9.4

Aggressive Quadrotor-flip Evaluation In this section, we further conduct onboard quadrotor flip experiments to evaluate the performance of our PL-EVIO in aggressive motion. The 2

quadrotor is commanded to perform a flip motion autonomously.

The estimated trajectory of our PL-EVIO com- pared **with the ground truth from VICON during the flip evaluations can be seen in Fig**. 3.12. **The total length of the trajectory is 15m, the mean translation error and the mean angular error are 0.097m and 6.0◦, respectively. Despite the extreme velocity of the motion, our PL-EVIO successfully tracks the quadrotor pose with high accuracy** 2

. (

a) X-axis (b) Y-axis (c) Z-axis (d) Roll-axis (e) Pitch-axis (f) Yaw-axis 5

Figure 3.11: **The position, orientation, and the corresponding errors of our PL-EVIO in onboard** flight **compared with the ground truth from VICON (Taking the** On- board_**test_1 as example**

).

Note that our PL-EVIO is run onboard during the quadrotor flip experiments 2

,

thanks to our well-designed feature management and the complementarity of three kinds of features, our PL-EVIO can provide robust and good performance in multiple quadrotor flip experiments 2

. As shown at the bottom of Fig. 3.12, the image measurement

is barely visible due to motion blur, resulting **the** 9

image-based feature extraction and tracking to be challeng- ing. Despite the standard cameras suffering from serious motion blur, the event camera can still provide useful information. This enables our pipeline to leverage event and IMU measurements to maintain accurate pose feedback control, allowing the quadro- tor to execute flip commands with minimal drift. On the other hand, during quadrotor hovering, a near no-motion condition, standard cameras successfully and reliably track visual features, whereas event cameras often lose features. This occurs because vibra- tions during hovering induce rapid directional changes, causing event-based features to appear inconsistent due to drastic changes in motion direction. By leveraging

the complementary advantages of standard and event cameras 34

, the quadrotor successfully maintains

its position with no noticeable drift 9

. Further details and video demonstra- tions of the quadrotor fliping experiment are available on the PL-EVIO's project web- site. Figure 3.12:

The estimated trajectory of our PL-EVIO on the quadrotor flip, and im- age/**event measurement** 2

. Figure 3.13: The

autonomous flight of a quadrotor in the dark 9

environment using our ESVIO [73] as pose feedback control; The self-designed quadrotor platform is similar to the one in Fig. 3.8 but is equiped with stero cameras. 3.9.5 Quadrotor-flight Evaluation in Dark Environment

In this section, we demonstrate the capability of our 123

ESVIO [73] to enable autonomous quadrotor flight in a dark environment. As can be seen from Fig. 3.14, the experiment begins with the quadrotor taking off in the room with normal lighting. Once the drone takes off,

the room lights are fully **switched off** 9

, leaving only minimal residual illumi- nation from the corridor (very low illumination,

but still enough for the event camera to work). This forces **the** 9

ESVIO pipeline to rely exclusively on events

and IMU mea- surements to maintain autonomous flight. **The** 9

image frames become completely black (middle bottom of Fig. 3.13) while the event frames still carry rich information (left bottom of Fig. 3.13). Figure 3.14: Pixel fire events caused by the abrupt illumination change. We also evaluate

the relative pose error (RPE) of our ESVIO against the 7

ground truth pose from VICON in Fig. 3.15. The quadrotor traveled a total trajectory length of 56.0m, with an average translational error of approximately 0.1 m and an average rotational error of 7◦. Notably, outliers observed between 50 and 60 seconds coincide with rapid yaw changes, causing slight delays in the estimated pose relative to VICON. Additionally,

the root-mean-square error (RMSE) of the **absolute trajectory in** this eval- uation **is 0** 7

.17 m. Abrupt changes in illumination, such as switching the lights on and off, generate significant events across almost all pixels, as shown in Fig. 3.14. However, due to the robust feature management design and the complementary use of IMU mea- surements, our system maintained high accuracy under these challenging conditions in Figure 3.15:

The relative error comparison of our proposed ESVIO with the 7

ground truth pose from VICON. practice. Further details and video demonstrations of the quadrotor flight in the dark environment are available on the ESVIO's project website. 3.9.6

Outdoor Large-scale Evaluation Natural Scenarios In this section, we evaluate our PL-EVIO system in a large-scale environment that encompasses **the HKU campus. This environment includes features such as moving pedestrians, low-texture areas, long-term movement, strong sunlight, and indoor & outdoor transitions. We also return to the same location after a large loop to** 2

evaluate the loop closure. The total evaluation length is approximately 980 m, covering an area of approximately 160 m in length, 100 m in width, and 10 m in height. The estimated trajectory is aligned with the 2

Google map and can be seen in Fig. 3.16(**a). The results show that our PL-EVIO performed almost drift-free in this long- term motion evaluation. The complementarity of three different kinds of features (e.g., line-based event features for human-made environment, point-based event features for HDR scene, and point-based image features for good lighting scene**) ensures **the robust and reliable state estimation. Human-made Scenarios We further conduct additional evaluations specifically** fo- cusing **on building scenarios**. Utilizing **the line features can better represent the** geo- metric **information constraints in human-made structures, as** illustrated **in Fig**. 3.16(**b). The experimental results demonstrate that after a long-distance loop of approximately 260 m within the interior of the building, our PL-EVIO system maintains high accuracy, (a**

) HKU_centennial_garden (b) HKU_main_building Figure 3.16: (

a) The estimated trajectory of our PL-EVIO in the outdoor environment. We also visualize the detection and tracking situation of the event-corner features, line- based event features, and point-based image features, during the experiment. The com- bination **of these features provides more structures and constraints in the scene that ensure robustness. (b) The estimated trajectory of our PL-EVIO as well as the detection and matching performance of the line-based event features** 2

.

forming a complete square shape without significant drifts. To assess this accuracy, we specifically choose the gate of the building as the starting and ending point for quan- titative **evaluation, and the end-to-end distance showed an error of 0.61 m. Owing to the additional geometric structural** information **provided by our proposed event- based line features, our PL-EVIO achieves low drift and reliable performance in this large-scale environment** 2

. 3.9.7 Real-time Analysis We

assessed the real-time performance of our system on quadrotor **flight using an** In- tel **NUC10i7FNH as the** computing **platform. The computational allocations are** pre- sented in **Table** 3.5. **The proposed algorithm sequentially** 4

processes

the event queue, with the event front-end completed within 9 ms, the image front-end completed within 3 ms, and overall optimization completed within 50 ms, without any hardware accel- eration. **In order to achieve low latency, we employ a loosely-coupled approach to directly propagate the latest EVIO estimation along with the IMU measurements. This results in IMU-rate EVIO outputs that can reach up to 1000 Hz. This is critical for achieving onboard quadrotor flight, using our PL-EVIO as pose feedback control** 2

.

Due to the reliable, low-drift, and low-latency characteristics of our PL-EVIO, the flight con- trol **system can quickly obtain accurate pose feedback, thereby ensuring the success of onboard quadrotor flight** 2

. Table 3.5:

Time Consumption of Different Modules in Our **PL**-EVIO Modules **Time-cost (ms) Event Front**-end Point-**based event feature detection** Point-**based event feature tracking** Line-**based event feature detection** Line-**based event feature Matching** 4

0.41 0.62 2.88 3.57 Total 8.68 Image Front-end Point-

based image feature detection Point-**based image feature tracking 0** 4

.70 0.42 Total 2.29 Back-end Construct

point-based event residual Construct **line-based event residual** Construct **point-based image residual** Construct **marginalization residuals** Solve **graph optimization using Ceres** IMU **forward 0** 4

.089 0.0026 0.66 7.50 37.66 0.0056 Total 50.49 3.9.8

Ablation Study on Different Event Representations for Feature Tracking In this section, we focus on conducting an ablation study of the event-based point track- ing **performance using** different **event representations, including our** Tp(x, **t**), Tnp(**x, t), TS in** [59], **and the event accumulated image in** [40] **and** [49]. **It should be noted that we previously only used** Tnp(x, **t) for loop closure detection, while we merely** inves- tigate **its feature tracking performance in** the **front-end for this ablation study**. Dur- ing **the ablation experiments, we employ our PL-EIO framework to control variables by only altering different event representations used for event-based feature tracking, while keeping the event feature points generated from asynchronous event streams unchanged. The qualitative results are presented in Fig**. 3.17, **and the quantitative** eval- uations **are reported in the "Event Representations" section of Table** 3.6. **We utilize** ab- solute **trajectory error (ATE), aligning the estimated** trajectory **with ground truth using 6-DOF transformation (in SE3) to quantitatively evaluate the accuracy. The results** in- dicate **that only our proposed Tp(x, t) is capable of reliably estimating the** 2

state, while other **event representations used for event-based** feature tracking **failed in the** chal- lenge **situation. This could** be due **to the insufficient intensity information available** to satisfy **the requirements of LK optical flow for event**-based feature **tracking. For** exam- ple, **the image generated** from event **streams**[49][40] **is a binary edge image** consisting of **only two possible pixel values (0 or 1), which lacks** the necessary **information for accurately calculating gradients. Consequently, it becomes difficult to determine the** direction and **magnitude of motion of feature points, resulting** in increased **difficulty for optical flow to remove local outliers. In contrast, our** 4

TS with polarity can ensure reliable data association **between event features of adjacent frames**, which effectively **prevents miss-matches**

. 3.9.9

Ablation Study on Time Decay Kernels of the TS with Polarity In order to further investigate the impact of time decay kernels on our event represen- tations (Tp(**x, t)) used for event feature tracking, we conducted ablation experiments on the time decay kernels. The qualitative results on the event-based point and line features are shown in Fig**. 3.18. **From Fig**. 3.18(**a), we can observe that the tracking and** 2

Figure 3.17:

The performance of the event-corner features tracking in different event representations 4

. Table 3.6:

Accuracy Result of the Ablation Study in HKU_**agg**_flip Event **Representations Tp(x, t) Tnp(x, t) TS Event** 4

-image PL-EIO Event+

IMU 0.25 failed failed failed Time **Decay Kernel 10 20 60** 4

100 PL-EIO Event+IMU 0.30 0.25 failed failed Methods PL-EVIO Event+Image+IMU Ref.[49] Event+Image+IMU Ref.[40] Event Ref.[59] Stereo Event 0.12 2.66 failed failed

matching performance of event-based point features and line features are not signif- icantly **different across various exponential decay kernels. This may be attributed to the normal texture conditions and fewer triggered events at a far distance in outdoor environments. However, upon careful observation, we still can notice that larger time decay kernels result in coarser edge contours in the edge regions, which may introduce systematic errors to the visual front-end. In contrast, in indoor environments (as shown in Fig**. 3.18(**b)) with abundant texture, we found that the larger time decay kernels lead to a significant decrease in the successful matching of both event-based line and point features. This might be due to the trailing effect caused by a large time decay kernel, which can create negative effects (i.e. lead to failure during aggressive motion) similar to motion blur. Therefore, we choose a time decay kernel of 20ms to** 2

ensure sufficient information for event-based feature matching **and tracking while minimizing blur and trailing effects. We quantitatively evaluate the influence of the time decay kernel on** 2

(

a) HKU_centennial (b) HKU_agg_flip Figure **3**.18: **The tracking and matching performance of event-based point and line features in various** 4

time decay parameters. We only evaluate the tracking and matching performance in (a) outdoor environments with large-scale and (b) indoor environments **with aggressive motion, respectively**

. 3.10. Conclusion

the performance of pose estimation through the Time 2

Decay Kernel part of Table 3.6. **Furthermore, we also compare the performance of our PL-EVIO using Tp(x, t) as event representation with other event representations from** 2

Ref. [49], [40], and [59], in the Methods part of Table 3.6. 3.10

Conclusion In this chapter, **we propose a robust, highly-accurate, and real-time optimization-based monocular** EVIO **that tightly fuses the event, image, and IMU together, with point and line features. The combination of point-based event-corner features, line-based event features, and point-based image features would provide more geometric constraints on the structure of the environment. Finally, we show superior performance by** compar- ing **against other state-of-the-art open-source image-based or event-based VIO** imple- mentations **in different challenge datasets. Meanwhile, through extensive experiments** including **extremely aggressive motion and large-scale evaluation, we also show that our PL-EVIO pipeline is able to leverage the properties of the standard camera and the event camera with different features to provide robust state estimation. We hope that this work can inspire other researchers and industries to push wide applications for event cameras on robotics and perception** 2

. This chapter also demonstrates the evaluation of our ESVIO [73] through provid- ing pose feedback control for quadrotor flight in dark environments. It is important to note that ESIO is built upon Mono-EIO [2] in Chapter 2 and can be considered as the stereo version of Mono-EIO [2], while ESVIO also can be regarded as the stereo ver- sion of PL-EVIO (without line features). Its robust performance further validates the effectiveness of our proposed EVIO framework and the feature management strategy. 3.11 Related Publications 1. Guan Weipeng*, Chen Peiyu*, Xie Yuhan, Lu Peng, “

PL-EVIO: Robust Monoc- ular **Event-based Visual Inertial Odometry with Point and Line Features”, IEEE Transactions on Automation Science and Engineering, pp. 1-17, 2023** 52

. [simultaneously present at ICRA2024] 2. Chen Peiyu*, Guan Weipeng*, Lu Peng, “

ESVIO: Event-based Stereo Visual Iner- tial Odometry”, IEEE Robotics and Automation Letters, pp.3661-3668 47

, 2023. [simul- taneously present at IROS2023] Chapter 4 Event-based Hybrid Odometry Building on the feature-based event pose tracking introduced in previous Chapter 2 and 3, this chapter presents

a hybrid framework that integrates **feature-based and direct-based methods** 1

for a monocular event camera, facilitating accurate 6-DoF pose estimation.

A novel event-based hybrid tracking framework is designed to estimate the pose, leveraging the robustness of feature matching and the precision of direct align- ment. **Specifically, we develop an event-based 2D-2D alignment to construct the** photo- metric **constraint, and tightly integrate it with the event-based** re-projection **constraint** 1

. This chapter is based on the tracking module of our work [1]. More demonstrations can be seen in the project website: https://kwanwaipang.github.io/EVI-SAM/. 4.1 Introduction

Event-based pose estimation **has also gained significant research interest for** challeng- ing **scenarios where the performance of** standard **cameras may be** compromised, **such as under extreme lighting conditions and very fast motion. Prior studies on event- based pose tracking using direct-based methods (such as EVO** [40], **ESVO** [59], **EDS** [67]) **as well as feature-based methods (such as Ultimate-SLAM** [49], **PL-EVIO** 1

[72], ES- VIO [73]) have made significant progress in state estimation.

Both feature-based and direct-based approaches **have their** advantages **and** limitations **in their** 1

respective as- pects, allowing them to complement each other

to some extent. **Exploring how to better leverage or integrate the respective advantages of feature-based methods and direct-based methods for event-based pose tracking remains an uncharted area. This** 1

motivates us to combine feature-based and direct methods for a more robust and accu- rate **event-based pose tracker** 1

. Events are more compatible with direct methods than standard images, as they in- herently mitigate many challenges commonly linked to image-based direct approaches.

Firstly, events are triggered by intensity changes **and naturally select high-gradient** pix- els, **effectively substituting the gradient selection process in image-based direct** meth- ods [10]. **Secondly, the high** temporal **resolution of** event data ensures **minimal** dis- placement **between** consecutive **event frames** 1

, making it easier to achieve an optimal solution for the photometric loss [201].

Thirdly, the event camera is immune **to** pho- tometric **distortions/variations, gradual** brightness **changes, and highly non-convex or non-linear intensity values. Besides, both feature-based and direct-based methods have their strengths and weaknesses in various aspects, which complement each other to some extent** [18, 11]. **The robustness provided by salient visual features makes feature-based methods well-suited for handling irregular scene variations and large inter-frame motions. Conversely, direct**-based **methods exhibit better resilience and superior accuracy in low-textured scenes by leveraging high-gradient** 1

sub-pixel align- ment. **Therefore**, through integrating **the feature-based and direct-based methods into an event-based hybrid pose tracking**

framework, we can leverage **the** superior robust- ness **of feature-based event pose tracking** while achieving **the relatively high accuracy** provided by **event-based direct alignment** 1

. In this chapter, a novel event-based hybrid tracking framework (the tracking mod- ule of EVI-SAM [1]) is proposed to enable accurate 6-DoF pose tracking,

leveraging the robustness of feature-based methods **and the precision of direct** 1

-based methods. Specif- ically,

a sliding window **graph-based optimization framework is designed to tightly fuse the event-based geometric** errors (**re-projection** residuals) **and event-based** pho- tometric errors (**relative pose** residuals), **along with the image-based geometric** errors **and the IMU pre-integration**. To **the** 1

best of our knowledge, this is

the first hybrid ap- proach **that integrates both photometric and geometric errors within an event-based framework. Our contributions can be summarized as follows: 1. We present a novel direct event-based pose tracking scheme that employs event- based 2D-2D alignment for photometric optimization** 1

. 4.2. Related Works

2. We introduce an innovative event-based hybrid tracking pipeline. This pipeline efficiently constructs a nonlinear graph optimization framework to jointly incor- porate **the** re-projection **constraint from the feature-based method and the relative pose constraints from the direct-based method. 3** 1

.

Extensive experimental evaluations, both qualitative and quantitative, demon- strate **the accuracy, robustness, and outstanding performance of our** 1

event-based hybrid pose tracking approach. 4.2 Related Works

Event-based pose tracking can be classified into two primary categories: (i) feature / indirect-based methods [49, 72, 73], **that extract a sparse set of repeatable event-based features from the event data, and then recover the pose and scene geometry through minimizing the** re-projection errors **based on these feature associations; and (ii) direct- based methods** [40, 59, 67], **that directly** minimize **photometric** errors **in different event representations without explicit data association**. 4.2.1 **Feature-based Event** 1

Pose Tracking The first feature-based event poses **tracking method is proposed in Ref**. [43] **which fuses events with IMU through the Extended Kalman Filter**. After that, **nonlinear** opti- mization **employing feature-based methods, EIO (event and IMU odometry) and EVIO (event, image, IMU odometry**), are **introduced in Ref**. [44] **and Ultimate-SLAM** [49], **where the image-like event frames** are **developed to leverage traditional image-based feature detection and tracking. Ref**. [48] **introduces an EIO approach by addressing the** re-projection errors **from the asynchronous events and directly incorporating IMU within a continuous-time framework. Ref**. [66] combines **the event-based and image- based feature tracker** [5] **as the front end with a filter-based back end. Ref**. [2] **presents a monocular feature-based EIO that employs event-corner** features **to provide real-time 6-DoF state estimation. Ref**. [65] **extracts features from events-only data and** associ- ated **it with a spatio-temporal locality scheme based on exponential decay. Ref**. [63] **performs the event-only VO by optimizing the camera poses and 3D feature positions jointly**

. ESVIO [73] proposes

the first stereo EIO and EVIO framework to estimate states through temporally and spatially event-corner feature association. IDOL [55], Refs. [70] **and** 1

[69]

explore the event-based line feature in state estimation. PL-EVIO [72] inte- grates **event-based point and line features to perform robust pose estimations, which can be used as onboard pose feedback for the quadrotor to achieve aggressive flip** 1

mo- tion. 4.2.

2 Direct-based Event Pose Tracking Ref. [38] **presents an event-based direct method that leverages photometric** relation- ships **between brightness changes and absolute** brightness **intensity to associate events with the corresponding pixels in the reference image. EVO** [40] **performs an event-based tracking approach relying on edge-map (2D-3D) model alignment, utilizing the 3D map reconstructed from EMVS** 1

[41]. Refs. [46] and [50]

present direct-based track- ing **of an event camera from a given photometric depth map. ESVO** [59] **is the first stereo event-based VO method, which follows a parallel tracking and mapping scheme to estimate the ego motion through the 2D-3D edge registration on time surface (TS). Building upon the framework of ESVO**, Ref. [68] introduce **a direct VO framework that incorporates a depth camera to supply depth information for the event camera. EDS** [67] proposes **an event-image alignment algorithm that minimizes the** photomet- ric errors **between the brightness change from events and the image gradients, enabling 6-DOF pose tracking. However, most of the aforementioned direct-based event pose tracking methods are limited to small-scale environments and small, dedicated** move- ments. **They struggle to** 1

provide reliable state estimations, **as they heavily depend on successful direct model alignment and the timely updates of the local 3D map. While the integration of both direct-based and feature-based methods in event-based pose tracking** remains an unexplored **research gap**

. 4.3 Framework

Overview The proposed EVI-SAM utilizes inputs from the **monocular event camera, including events, images, and IMU, to** simultaneously **estimate the 6-DoF pose and reconstruct the 3D dense maps of the environment. The** framework **overview of EVI-SAM is** il- lustrated **in Fig**. 4.1. **Our EVI-SAM consists of tracking and mapping modules, which** 1

4.3. Framework Overview operate in two parallel threads. The mapping module would be introduced in Chap- ter 6, while this chapter only focus on the introduction of the tracking module.

Our proposed event-based hybrid tracking module tightly integrates feature-based EVIO and event-based direct alignment **within a graph-based nonlinear optimization frame- work. The feature-based EVIO (detailed in** 1

Chapter 2 and Chapter 3)

consists of event- based landmarks, image-based landmarks, and IMU pre-integration. The event-based direct alignment (detailed in Section 1

4.5)

takes continuous event streams as input and calculates the photometric errors **at the pixel level to establish relative pose constraints** 1

. Image-based

Event-based Mapping Thread Segmentation **Event** Stream Event-**based** Event-based **TSDF-based** 1

Semi-dense Depth Dense Depth Dense Depth Map Fusion Recovery

Image Frame Direct Event-based Tracking Event-based 2D-2D alignment Relative Pose Residual Event-corner Feature Tracking 1

Event-based Triangulation Re-projection Residual IMU Data Feature Detection and Tracking Image-based Re-projection Residual Sliding Windows IMU Pre-integration Initialized? Y Graph Optimization IMU Residual Event-based Tracking Thread 3D Map 6 DoF Pose Figure 4.1:

System overview. The EVI-SAM algorithm takes events, images, and IMU as inputs, enabling the recovery of both camera pose and dense map of the scene. The mapping process 1

(would be introduced in Chapter 6)

takes raw event streams as input, using images for guidance, and produces dense and textured 3D mapping as output. The tracking thread takes event, image, and IMU as input, and constructs the feature-based and direct-based constraints to estimate the 6-DoF pose 1

. As for the feature-based EVIO,

when a new event stream arrives, the first step is to track the existing event-corner features using optical flow on the TS with polarity denoted as Tp(x, t) (see Eq. (4.2)). Any features that cannot be successfully tracked at the current timestamp are discarded. Subsequently, new event-corners are detected from the incoming event stream whenever the number of tracked features falls below a certain threshold. Meanwhile, the TS with polarity Tp(x, t) is also utilized as a mask to ensure the even distribution of event-corner features. After undistortion, outliers filtering, and triangulation, the event-based landmark, for which the 3D position has been successfully calculated, is integrated to establish the re-projection constraints. Regarding the event-based direct alignment, direct refinement is performed on every event mat Et 1

(detailed in Section 4.4 and Eq. (4.3)) by minimizing the photomet- ric errors

to establish relative pose constraints. After that, a tightly joint optimization scheme (Eq. (4.10)) is performed over the sliding window which contains re-projection constraints and the relative pose constraints. To ensure the co-visibility between the feature-based and direct constraints, the state variables shared by these two constraints are synchronized before optimization 1

. 4.4 Accumulated Event Image

Event cameras are motion-activated sensors that capture pixel-wise intensity differ- ences and report them as a continuous stream, rather than capturing the entire scene as an intensity image frame. An event can be triggered either by moving objects or the ego-motion of the camera, when a large enough intensity change exceeds the pre- defined threshold Tthreshold, it can be represented as follows: $e = \{t, u, v, p\} \Leftrightarrow I(u, v, t + \triangle t) - I(u, v, t) \geq p$ 1

· Tthreshold (4.

1) where t is the timestamp that the intensity of a pixel I (u, v) changes, and p is the po- larity that indicates the direction of the intensity change. Since a single event lacks sufficient 1

information, it is common to aggregate sets of events within time intervals into synchronous data representations, rather than processing individual events asyn- chronously. The 1

proposed event-based hybrid tracking module employs two distinct event representations

as intermediate variables to facilitate the tracking between ad- jacent 1

observations: (i) **the time surface (TS) with polarity (Eq**

. (4.2), detailed in Sec- tion 2.4.3)

for the event-corner feature tracking, and (ii) **the event mat (Eq**. (4.3)) **for event-based 2D-2D alignment. The TS with polarity can encode the spatio-temporal constraints of the historical event stream at any given instant. Using an exponential decay kernel, it can emphasize recent events compared to past events. Assuming** tlast(x) **as the** timestamp **of the last event at each pixel coordinate x = (u, v), the TS with polarity at time t** ≥ tlast(**x) can be defined by: Tp(x, t) = p · exp(− t − tlast(x) η** 1

) (4.2) 4.5. Event-based Direct Measurements

where η is the decay rate kernel. The accumulated **event** 1

image, also known as

event mat, is generated by aggre- gating **a group of events that occur within a temporal neighborhood onto the event- accumulated image. In this process, pixel values are set to 255.0 when** 1

events are trig- gered;

otherwise, they are set to zero, as follows: Et = ∑ {**e**|t0 ≤ **t** ≤ t0 + ∆**t} t**=t0+∆**t** 1

(4.3) **t**=t0 where ∆**t**

is the constant temporal window of the observed events 1

. The temporal win- dow length for event processing remains consistent with the frequency of the input event stream. 4.5

Event-based Direct Measurements **The direct-based measurements of our framework align the observed 2D event mats of the two consecutive** timestamps. **The photometric** errors **can be created by the 2D event mat Et at two timestamps k and i**: e2D−**2D = Ek − Ei** = E(uk, vk) − **E** 1

(∆Tik(ui, vi) (4.4)

The data association of the pixel between these two event mats is completed using the inverse compositional Lucas-Kanade method (see Fig. 4.2(**a)), which iteratively** com- putes **the incremental motion** ∆Tik **from the reference pose of the Ei at timestamp i to the current pose of the Ek at timestamp k** 1

. While the Jacobian and Hessian Matrix of e2D−2D are given by: J = ∑ ∂e2∆D−2kD · e2D−2D u,v ∂ Ti H = ∑ J · JT, H · ∆χ = J (4.5) u,v where ∆χ is the update pose vector, the Lie-algebra, representing the incremental change of the pose within each iteration. While the incremental motion ∆Tik

from the reference **frame to the** current **frame can be** iteratively **calculated** 85

as: ΔTik ← expSE(3)(Δχ)ΔTik (4.

6) (a) Event-based 2D-2D alignment (b) Event-based 2D-3D alignment Figure 4.2: **Direct event-based alignment. (a) Event-based 2D-2D alignment: The 2D event mat in the current timestamp (a1) is warped to the 2D event mat in the previous timestamp (a2). The result (a3) is the alignment between the 2D current event mat (white) and the previous 2D event mat (red). (b) Event-based 2D-3D alignment: The current 2D event mat aggregated through a small number of events(b1) is warped to the projected mat recovered from the event-based 3D semi-dense depth in (b2). The result (b3) is a good alignment between the 2D current event mat (white) and the projected 3D event-based map (color**

).

Instead of directly incorporating the photometric errors **term in the graph** opti- mization, **the pose-pose constraint is leveraged to implicitly contain the relative** trans- forms **derived from the direct method. The relative pose constraint based on the** opti- mized **incremental pose** ΔTik **in Eq**. (4.4) **is given by: r**(zdirect, **χ**

) = ∑ ||ΔTik · (Tbwi)−1 · Tbwk||2 (4.7) i,k∈K where zdirect

represents the direct-based measurements. K (K = 10 in our experiments) is the number of keyframes in the sliding window. Tbwi **and** Tbwk **are the pose of the body frame in the** world **frame in ith and kth keyframe**. An example of **the**

event-based 2D-3D alignment is also illustrated in Fig. 4.2(**b). Current event-based direct pose** tracking **methods**

[40, 59, 67, 68] rely

on the 2D-3D ge- ometric **alignment, resulting in relatively lower localization accuracy and even failures in pose tracking. This** may **be attributed to the slow convergence of the estimated 3D map or the rapid changes in edge patterns between adjacent event packets. We posit that the effectiveness of the event-based 2D-3D alignment method relies heavily on the**

4.6. Feature-based Measurements

accurate construction of the event-based 3D depth map and its successful alignment **with the current 2D event frame. This limitation results in a notable lack of robustness in these event-based direct pose tracking methods, particularly in scenarios involving aggressive motion or HDR. Conversely, our proposed event-based 2D-2D alignment excels in performance without** relying **on the 3D depth maps, surpassing the** perfor- mance **of the conventional event-based 2D-3D alignment**. 4.6 **Feature-based**

Measurements **The feature-based tracking module of our system is the same as the feature-based EVIO in previous**

Chapter 2 and 3,

where the camera poses are optimized by minimizing a joint nonlinear least-squares problem for the system state χ as follows: r(zfeature, **χ) = K∑−1** ||eikmu||2 + **K** 1

∑−1 ∑ ||ekim,lage||2 + K∑−1 ∑ ||eekv,lent||2 (4.8)

k=0 k=0 l∈ξ k=0 l 2

∈ζ Eq. (4.8)

is composed of three components: (i) the IMU pre-integration residuals ekimu; (ii) the feature-based image 1

residuals eikm,lage; (iii) the feature-based event residu- als ekev,lent.

ζ and ξ are the set of event-corner features and image features, respectively. K (K = 10 in our experiments) is the total number of keyframes in the sliding win- dow. **The detailed mathematical explanation** 1

and the keyframe selection scheme can be found in previous Chapter 2 and 3.

Here, we only review the re-projection constraint of the event-corner features. Considering the lth **event-corner feature that is first observed in the** ith **keyframe, the residual for its observation in the** kth **keyframe is defined as** 1

: eke,vlent = ⌈ ukl ⌉ − πe · (Teb)−1 · Twbk · Tbwi · Teb · π−e1(1 λ ,⌈ uil vkl (4.9) l⌈ ⌊ ⌈⌋⌈ ⌊ vil ⌉) ⌈⌋

where λl is the inverse depth of the event-corner features; [uil, vil]T is the first observa- tion **of the lth event-corner** feature **in the ith keyframe. [ukl, vkl]T is the observation of the same event-corner feature in the kth keyframe, πe and πe−1 denote the event camera projection and inverse projection** 1

. Teb

is the extrinsic matrix between the event camera frame and the body frame 1

. Tbwi

indicates the movement of the body frame related to the world frame in timestamp i, Twbk **is the transpose of the pose of the body in the world frame in the** kth **keyframe** 1

. 4.7 Hybrid Optimization for Feature-based and Direct- based Measurement

We define hybrid optimization as the process of maximizing a posterior based on feature- based and direct-based measurements. Assuming that these two measurements are in- dependent of each other and the noise with each measurement is zero-mean Gaussian distributed, the hybrid optimization problem can be formulated as the minimization of a series of costs as follows: χ∗ = arg max p(χ|z) χ n (4.10) = arg min χ { ||rm − Hmχ||2 + i∑=1 ||r(zi, χ

)||2Σi }

where χ is the estimated state of the system, the state

is composed by the position pwb, and the orientation quaternion qwb, and the velocity vbw of the IMU in the world frame

(χ = [pwb, qwb, vwb]).

z stands for the aggregation of the independent feature-based and direct-based measurements (n = 2 in our framework), and {rm, Hm} encapsulates the prior information or the marginalization of the system state. r (·) denotes the resid- ual function of each measurement, Σi denotes the

covariance, and || · ||Σi is the Ma- halanobis

norm. We decompose this optimization problem as two individual factors, the relative pose constraints from direct-based measurements (Eq. (4.7)), and the con- straints from feature-based EVIO (Eq

. (4.8)). 4.8 Experiments In this section, we first

show the effectiveness of our event-based hybrid pose track- ing module by comparing the estimated trajectories against the ground truth pose in challenging situations. Next, in Section

4.8.2, we present an ablation study focusing on the comparison among direct-based, feature-based, and hybrid event pose tracking 4.8. Experiments methods. While Section 4.8.3 delves into the ablation study examining the integration of event and image-based solutions. 4.8.1 Evaluation

in Aggressive Motion and HDR Scenarios To evaluate the tracking performance of our event-based hybrid framework in chal- lenging scenarios, we compare our system with other state-of-the-art image-based or event-based VO/VIO using publicly available datasets: DAVIS240c [144], Stereo HKU- dataset(Appendix A) and VECtor [157]. These datasets offer events, grayscale frames, IMU, and ground truth poses

,

encompassing challenge scenarios such as HDR, ag- gressive motion, large-scale, as well as textureless

environments.

The criterion used to assess tracking accuracy is the mean absolute trajectory error, which aligns the esti- mated **trajectory with the ground-truth pose in the 6-DOF transformation (SE3** 1

).

The best results for each sequence are highlighted in bold. Table 1

4.1: Accuracy Comparison of Our Event-based Hybrid Pose Tracking

with Other EIO/EVIO Works in DAVIS240c Dataset 1

[144].

Unit:%, 0.21 means the average error would be 0.21m for 100m motion; Aligning 5 seconds [0-5s] of the estimated trajectory with the ground truth; The notations E, F, and I stand for the use of event, frame, and IMU, respectively 1

. Sequence Ref. [43] Ref. [44] Ref. [49] Ref. [49] Ref. [180] Ref. [180] Ref. [181] Ref. [66] Ref. [65] Mono-EIO [2] PL-EVIO [72] Ref. [80] Ours (

E+I) (E+I) (E+I) (E+F+I 35

) (

E+I) (E+F+**I) (E+I) (E+F+I) (E+I) (E+I) (E+F+I) (E+F+I**) (E+F+**I** 35

)

boxes_translation 2.69 0.57 0.76 0.27 1

1.50 1.24 2.55 0.48 1.0 0.34 0.06 0.74 0.11 hdr_boxes 1.23 0.92 0.67 0.37 2.45 1.15 1.75 0.46 1.8 0.40 0.10 0.69 0.13 boxes_6dof 3.61 0.69 0.44 0.30 2.88 0.98 2.03 0.84 1.5 0.61 0.21 0.77 0.16 dynamic_translation 1.90 0.47 0.59 0.18 4.92 0.89 1.32 0.40 0.9 0.26 0.24 0.71 0.30 dynamic_6dof 4.07 0.54 0.38 0.19 6.23 0.98 0.52 0.79 1.5 0.43 0.48 0.86 0.27 poster_translation 0.94 0.89 0.15 0.12 3.43 1.83 1.34 0.35 1.9 0.40 0.54 0.28 0.34 hdr_poster 2.63 0.59 0.49 0.31 2.38 0.57 0.57 0.65 2.8 0.40 0.12 0.52 0.15 poster_6dof 3.56 0.82 0.30 0.28 2.53 0.97 1.50 0.35 1.2 0.26 0.14 0.59 0.24 Average 2.58 0.69 0.47 0.25 3.92 1.07 1.45 0.54 1.56 0.39 0.24 0.65 0.21 In Table 4.1,

we directly report raw results from the original papers we compared, as **the same** trajectory **alignment protocol is utilized. Our** 1

event-based hybrid pose tracking

achieves the best performance among event-based VIO, while **with** only a **slight improvement compared to PL-EVIO** 1

. Therefore, we further conduct Table 4.2 to provide a more diverse evaluation.

Compared to pure direct event-based VO meth- ods (**e.g., EVO** [40], **ESVO** 1

[59]) **or pure feature-based EVIO methods (e.g., Ultimate SLAM**

[49]), our event-based hybrid pose tracking

significantly surpasses them in both accuracy and robustness 1

. This underscores the effectiveness of

our proposed hybrid tracking pipeline, which **combines the robustness of the feature-based method with the high accuracy achieved by the direct-based method. Compared** 1

with PL-EVIO [72], **a feature-based method, our** 1

event-based hybrid pose tracking also outperforms it in most sequences.

This highlights the effectiveness of our hybrid pose tracking pipeline, with the improvement stemming from the proposed event-based 2D-2D alignment. Table 1

4.2: Accuracy Comparison of Our Event-based Hybrid Pose Tracking

with Other Image-based or Event-based Methods 1

.

Unit: MPE(%) / MRE(deg/m); Aligning the whole ground truth trajectory with estimated poses; The notations E, F, and I stand for the use of event, frame, and IMU, respectively 1

. Sequence ORB-SLAM3 [9] VINS-Fusion [199] EVO [40] (F+F+I) (F+F+I) (E) ESVO [59] Ultimate SLAM [49] PL-EVIO [72] (E+E) (E+F+I) (E+F+I) Ours (

E+F+I) MPE / MRE MPE / MRE MPE / MRE MPE / MRE MPE / MRE MPE / MRE / MRE MPE / MRE HKU dataset hku_**agg_translation** 1

hku_agg_rotation hku_agg_flip hku_agg_walk hku_hdr_circle hku_hdr_slow hku_hdr_tran_rota hku_hdr_agg hku_dark_normal 0.15 / 0.075 0.35 / 0.11 0.36 / 0.39 failed 0.17 / 0.12 0.16 / 0.058 0.30 / 0.042 0.29 / 0.085 failed 0.11 / 0.019 1.34 / 0.024 1.16 / 2.02 failed 5.03 / 0.60 0.13 / 0.026 0.11 / 0.021 1.21 / 0.27 0.86 / 0.028 failed 0.59 / 0.020 3.14 / 0.026 6.86 / 2.04 2.00 / 0.16 1.32 / 0.54 2.80 / 0.099 2.64 / 0.13 2.47 / 0.27 2.17 / 0.031 0.07 / 0.091 0.23 / 0.12 0.39 / 2.23 0.42 / 0.14 0.14 / 0.62 0.13 / 0.068 0.10 / 0.064 0.14 / 0.30 1.35 / 0.081 0.17 / 0.056 0.24 / 0.056 0.32 / 2.08 0.26 / 0.22 0.13 / 0.56 0.11 / 0.033 0.11 / 0.026 0.10 / 0.26 0.85 / 0.052 VECtor [157] corner-slow robot-normal robot-fast desk-normal desk-fast sofa-normal sofa-fast mountain-normal mountain-fast hdr-normal hdr-fast corridors-dolly corridors-walk school-dolly school-scooter units-dolly units-scooter 1.49 / 14.28 0.73 / 1.18 0.71 / 0.70 0.46 / 0.41 0.31 / 0.41 0.15 / 0.41 0.21 / 0.43 0.35 / 1.00 2.11 / 0.64 0.64 / 1.20 0.22 / 0.45 1.03 / 1.37 1.32 / 1.31 0.73 / 1.02 0.70 / 0.49 7.64 / 0.41 6.22 / 0.22 1.61 / 14.06 0.58 / 1.18 failed 0.47 / 0.36 0.32 / 0.33 0.13 / 0.40 0.57 / 0.34 4.05 / 1.05 failed 1.27 / 1.10 0.30 / 0.34 1.88 / 1.37 0.50 / 1.31 1.42 / 1.06 0.52 / 0.61 4.39 / 0.42 4.92 / 0.24 4.33 / 15.52 3.25 / 2.00 failed failed failed failed failed failed failed failed failed failed failed failed failed failed failed failed 4.83 / 20.98 failed failed failed failed 1.77 / 0.60 failed failed failed failed failed failed failed 10.87 / 1.08 9.21 / 0.63 failed failed 4.83 / 14.42 1.18 / 1.11 1.65 / 0.56 2.24 / 0.56 1.08 / 0.38 5.74 / 0.39 2.54 / 0.36 3.64 / 1.06 4.13 / 0.62 5.69 / 1.65 2.61 / 0.34 failed failed failed 6.40 / 0.61 failed failed 2.10 / 14.21 0.68 / 1.25 0.17 / 0.74 3.66 / 0.45 0.14 / 0.48 0.19 / 0.46 0.17 / 0.47 4.32 / 0.76 0.13 / 0.56 4.02 / 1.52 0.20 / 0.50 1.58 / 1.37 0.92 / 1.31 2.47 / 0.97 1.30 / 0.54 5.84 / 0.44 5.00 / 0.42 2.50 / 14.82 0.67 / 0.85 0.22 / 0.41 1.45 / 0.28 0.18 / 0.38

0.19 / 0.20 0.98 / 0.31 1.39 / 0.65 0.38 / 0.30 5.74 / 0.87 0.67 / 0.26 1.58 / 1.38 1.27 / 1.35 1.53 / 0.89 1.46 / 0.53 0.59 / 0.35 0.83 / 0.38

An interesting observation **regarding the pure direct-based methods**, such as **EVO** [40] **and ESVO** [59], **both** approaches failed **in most of the sequences listed in Table** 4.2, **with only a few sequences** completed. **However, even in cases where they are completed successfully, these methods exhibit significant errors when compared to the ground truth poses. This can be attributed to the challenges presented by** our **evaluation** se- quences, **especially those containing aggressive motion. Consequently, approaches** re- lying **on edge-map (2D-3D) model alignment struggle to provide reliable state** estima- tions, **as they heavily depend on the timely updates of the local 3D map**. In contrast, **our proposed event-based 2D-2D alignment method does not suffer from these** 1

shortcom- ings. By leveraging the strengths of the combination of feature-based and direct-based methods, our method demonstrates improved robustness and reliability across a wider range of scenarios. This hybrid framework effectively balances the accuracy of direct alignment with the resilience of feature-based tracking, enabling more consistent and dependable pose estimation in diverse and challenging environments. We further conduct the comparison

of our event-based hybrid pose tracking with **the** 1

two latest works in the field of image-based VIO [11, 202]. We used the publicly available stereo HKU-dataset (Appendix A) which features rapid motion and HDR sce- narios. Following the trajectory alignment protocol of Table 4.2, we

computed the mean position as percentages of the total traveled distance, while the estimated trajectories and ground-truth were aligned in SE3 with all alignments 2

. Table 4.3: Accuracy Comparison of Our event-based hybrid pose tracking with image- based VIO on Stereo HKU-Dataset.

Unit:%, 0.17 **means the average error would be 0**.17m **for 100m motion; Aligning** 1

the whole ground truth trajectory with estimated poses 1

. Sequence EnVIO [11] MSOC-S-IKF [202] MSOC-S-IKF [202] ORB-SLAM3 [9] Our Hybrid Method

Stereo VIO Stereo VIO Mono VIO **Stereo** VIO **Mono EVIO** 14

hku_agg_translation failed failed 0.51 0.15 0.17 hku_agg_rotation 2.12 1.52 3.50 0.35 0.24 hku_agg_flip 2.94 failed failed

0.36 0.32 **hku_agg_walk** 3.38 **failed** 2.07 **failed** 7

0.26 hku_hdr_circle 0.85 failed failed 0.17 0.13 hku_hdr_slow 0.43 failed failed 0.16 0.11

hku_hdr_tran_rota 0.37 **failed** 5.48 **0**.30 **0** 7

.11 hku_hdr_agg 0.50 failed failed 0.29 0.10 hku_dark_normal failed failed failed failed 0.85 Table 4.3 demonstrates that our event-based hybrid pose tracking consistently sur- passes these image-based VIO methods across all tested sequences. These sequences are particularly challenging due to factors such as rapid motion, low-light environ- ments, and drastic illumination changes, which pose significant difficulties for tradi- tional image-based approaches. For instance, both monocular and stereo versions of the MSOC-S-IKF [202] perform well at the beginning of most of these sequences. How- ever, when confronted with fast motion or sudden illumination changes, these VIO systems are prone to failure or significant drift due to the unreliable perception of im- ages. Certainly, the performance may vary among different image-based VIO methods. For example, although the results of ORB-SLAM3 [9] and the EnVIO [11] also exhibit large errors under such challenging conditions, they tend to avoid serious drift or fail- ure, possibly owing to differences in framework stability. Nonetheless, image-based VIO often struggle to effectively manage high-speed motions and HDR scenarios, high- lighting the superior robustness of our event-based solution. 4.8.2 Ablation Study of Direct-based, Feature-based, and the Hybrid Framework This section provides the analysis of

the performance comparison between feature- based and direct-based EIO/**EVIO** 1

, both in the same framework and across different frameworks. Firstly, we demonstrate performance comparisons among different com- binations within the same framework (our hybrid pose tracking pipeline), including feature-based, direct-based, and hybrid (feature-based+direct-based) configurations. We used the publicly available monocular HKU-Dataset (Appendix A) (346×260, event, image, IMU, GT pose) and the DAVIS240C Dataset [144] (240×180, event, image, IMU, GT pose). Given that the DAVIS240C dataset typically exhibits limited displacement in the final 30 seconds, it is not ideal for effectively evaluating variations in position accu- racy, as the results from our framework tend to reach saturation. Therefore, we edited the rosbag files to include only the [0-30s] segment of the total duration of the 60s. The sequences in Table 4.4 are renamed as "edited" to distinguish them from the complete data

in Table 4.1. **As can be seen from the** result, **the** proposed direct-based **pose** 75

tracking outper- forms the feature-based component in most of the sequences. This result is consis- tent with the theoretical analysis in Section 4.1, as well as the common viewpoint in direct-based methods, i.e., sub-pixel alignment of the direct-based methods normally achieves better accuracy than the feature-based methods. On the other hand, as what has been well evaluated in our previous works [72, 73], our feature-based EVIO can operate robustly in large-scale outdoor environments and flight scenarios. This proves that it is suitable

for handling irregular scene variations and significant **inter-frame** 1

Table 4.4: Accuracy comparison of various combinations within our event-based hy- brid pose tracking framework, including feature-based, direct-based, and our hybrid (feature-based + direct-based) tracking pipeline.

Unit:%, 0.10 **means the average error would be 0**.10m **for 100m motion; Aligning** 1

the whole ground truth trajectory with es- timated poses. Sequence Feature-based Direct-based Hybrid vicon_hdr4 vicon_aggressive_hdr

boxes_translation_edited **hdr_boxes**_edited **boxes_6dof**_edited **dynamic_translation**_edited **dynamic_6dof**_edited **poster_translation**_edited **hdr_poster**_edited **poster_6dof** 38

_edited 0.09 0.08 0.92 0.59 0.15 0.10 0.21 0.12 0.34 0.16 0.40 0.32 0.19 0.18 0.12 0.15 0.55 0.32 0.10 0.08 0.09 0.68 0.14 0.21 0.20 0.34 0.23 0.35 0.44 0.09 motions. Therefore, leveraging the complementary

advantages of direct-based and feature-based methods is significant and 136

promising.

We further analyze the performance comparison between feature-based and direct- based EVIO. To ensure a fair comparison, we individually run the direct-based event tracking pipeline (Direct-based), the feature-based event tracking pipeline (Feature- based), and the entire 1

event-based hybrid pose tracking

system (includes both the direct-based and feature-based pipelines, marked as F-D-based). We visualize the es- timated results for translation along the X-Y-Z axes and rotation around the roll-pitch- yaw axes, along with the ground truth in Fig. 4.3(a) and 4.3(b). It is evident that the introduction of our event-based 2D-2D alignment significantly enhances the perfor- mance of pose tracking. For instance, in 1

the X-axis of Fig. 4.3(b), the Feature-based method deviates significantly from the ground truth between the 45th to 50th seconds, whereas the Direct-based method attains higher accuracy during this interval. Regard- ing the F-D-based, it has some offset influenced by the feature-based pipeline, but its performance remains superior to that of the standalone feature-based event tracking pipeline. On the other hand, it proves effective in handling irregular scene variations and large inter-frame motions, as evidenced by Table 4.2. Our entire 1

hybrid pose track- ing

system not only improves the accuracy compared to the standalone feature-based event tracking pipeline but also retains the advantages of the feature-based method 1

. (

a) boxes_6dof (b) hdr_poster (c) hku_agg_walk (d) mountain_normal Figure 4.3: Comparing the estimated trajectory in terms of translation and rotation produced by our 1

event-based hybrid pose tracking

with the ground truth trajectory in DAVIS240C [144], HKU-dataset, and VECtor [157]. (a) and (b) show the comparison among our framework with a feature-based event pipeline only, a direct-based event pipeline only, and the full event-based hybrid pipeline (feature-based + direct-based, F-D-based); (c) and (d) show the comparison between the PL-EVIO (feature-based) and our event-based hybrid pose tracking. The red dashed lines highlight the disparity between featured-based and direct-based methods 1

.

Furthermore, we use PL-EVIO as a representative of the feature-based method and compare its 6-DoF estimation results **with those of our** 1

event-based hybrid pose tracking,

as illustrated in Figs. 4.3(**c) and** 4.3(**d). It is worth noting that although the numerical improvement in accuracy may not be pronounced in the tables due to** statis- tical **considerations, the effectiveness of our hybrid approach over a pure feature-based method is more evident when examining Fig**. 4.3, **particularly in local state estimation. This validates the improvement in event-based pose tracking and highlights the** signif- icance **of introducing our event-based 2D-2D direct alignment method, particularly in conjunction with the feature-based method** 1

. 4.8.3 Ablation Study on Combination Event and Image As indicated in Section 4.8.1,

compared to standard cameras, event cameras are capable of providing reliable visual perception during high-speed motion and HDR scenarios. However, when event 2

cameras

have restricted relative motion, such as in static states, they **may produce** 4

limited information. Although the image camera encounters diffi- culties under

high-speed motion or HDR scenarios, it can provide rich-intensity tex- tures **of the scenes under uniform motion or favorable lighting conditions**. Therefore, leveraging the **complementary** 2

advantages

of both standard and event cameras 4

is sig- nificant and promising, as emphasized in many previous works [72, 73, 49, 203, 127, 67, 80, 66]. Table 4.5: Accuracy Comparison of various combinations within EVI-SAM framework, including Event+IMU, Image+IMU, Event+Image+IMU, direct-based, and our EVI- SAM.

Unit:%, 0.115 **means the average error would be 0**.115m **for 100m motion** 1

; Align- ing

the whole ground truth trajectory with estimated poses 1

. Sequence EIO Feature-based VIO EVIO Direct-based EVIO Hybrid EVIO vicon_hdr4 0.115 0.116 0.089 0.080 0.092 vicon_aggressive_hdr 0.572 1.266 0.921 0.594 0.683 Although the effectiveness of tightly integrating events, images, and IMU for pose estimation has been thoroughly assessed in various event-based studies, we still con- duct this ablation study of various combinations within our EVI-SAM framework in- cluding event-IMU-based (EIO), image-IMU-based (VIO), and event-image-IMU-based (EVIO) configurations, as shown in Table 4.5. In the first sequence, our feature-based EIO and VIO demonstrate comparable performance, thanks to our well-designed frame- work and the HDR level is not serious. Conversely, in the second sequence, our feature- based EIO significantly outperforms the feature-based

VIO, attributed to the high mo- tion intensity in the testing scene. While our feature-based EVIO performs satisfactory results and surpasses the solely image-based VIO in both testing sequences. 4.9 Conclusion In this chapter, we present the tracking module of

EVI-SAM, a framework designed for 6-DoF pose tracking 1

using the

monocular event camera. To enhance the robustness and accuracy of 6-DoF pose tracking, we propose **the first event-based hybrid pose** track- ing **framework which combines the robustness of feature-based event pose tracking with the relatively high accuracy achieved through event-based direct alignment** 1

.

Our framework balances accuracy and robustness against computational efficiency towards strong tracking performance in challenging scenarios such as HDR or fast motion. No- tably, **EVI-SAM demonstrates computational efficiency for real-time execution, making it suitable for onboard localization and navigation** 1

. 4.10 Related Publications 1. Guan Weipeng, Chen Peiyu, Zhao Huibin, Wang Yu, Lu Peng, “EVI-SAM: Ro- bust,

Real-time, Tightly-coupled Event-Visual-Inertial State Estimation and 3D Dense Mapping”, Advanced Intelligent Systems 11

, pp. 1-24, 2024. Chapter 5 DEIO: Deep Learning-based Event-Inertial Odometry As presented in previous chapters, event cameras show great potential for visual odom- etry (VO)

in handling challenging situations, such as fast **motion and high dynamic range**. Despite **this** 1

promise,

the sparse and motion-dependent **characteristics of event data** continue **to** limit the **performance** 38

of feature-based or direct-based data association methods in practical applications. To address these limitations, we propose Deep Event Inertial Odometry (DEIO),

the first monocular **learning-based event-inertial** framework, **which** combines a learning-**based** 11

method with traditional nonlinear graph-based op- timization. Specifically, an event-based recurrent network is adopted to provide accu- rate and sparse associations of event patches over time. DEIO further integrates it with the IMU to recover up-to-scale pose and provide robust state estimation. The Hessian information derived from the learned differentiable bundle adjustment (DBA) is uti- lized to optimize the co-visibility factor graph, which tightly incorporates event patch correspondences and IMU pre-integration within a keyframe-based sliding window. Comprehensive validations demonstrate that DEIO achieves superior performance on 10 challenging public benchmarks compared with more than 20

state-of-the-art meth- ods. **This** chapter **is based on** 65

our work [102]. More demonstrations can be seen in the project website: https://kwanwaipang.github.io/DEIO. Chapter 5. DEIO: Deep Learning-based Event-Inertial Odometry 5.1 Introduction Achieving reliable and accurate visual odometry (VO) under adverse conditions re- mains challenging, particularly when employing image-based solutions (RGB or RGB- D cameras).

Event cameras are motion-activated sensors that only **capture pixel-wise intensity** 1

changes with microsecond precision and report them as an asynchronous stream instead of the whole scene as an

intensity image with a fixed frame rate 2

. Be- cause of their remarkable properties, such as high temporal resolutions,

high dynamic range (HDR), and no motion blur, event cameras have **the** 5

potential to enable high- quality perception in extreme lighting conditions and high-speed motion scenarios

that are currently not accessible to standard cameras 9

. Despite such promises, integrating event cameras into VO systems presents sig- nificant challenges. This is primarily

due to the sparse, irregular, **and asynchronous nature of event data** 83

, which conveys limited information and contains inherent noise. Besides, event cameras face difficulty in capturing visual information when motion is parallel to edges or in a static state. Moreover, due to the motion-dependent characteris- tic, both feature-based and direct-based methods easily fail in incomplete observation or sudden variation of the event camera. Therefore, current purely event-based VO systems [40, 59] generally lack the robustness requirement for real-world applications. Using additional sensors tends to achieve better performance since different modalities can complement each other, such as image frames [49, 67, 72], depth sensors [92, 68], or even LiDAR [96]. However, these combinations might limit the application of event cameras

in real-world applications due to additional **computational costs and** 80

more complicated sensor calibration requirements [6]. Additionally, this introduces new bot- tlenecks. For instance, relying on image frames would make the system susceptible to motion blur or HDR limitations. Recent studies have introduced learning-based ap- proaches [84, 85] as promising solutions for event-based VO, addressing the previously mentioned limitations by employing neural networks to establish robust associations. These methods, trained exclusively on synthetic event data, demonstrate good gener- alization to real-world benchmarks and can even outperform some traditional systems in accuracy. However, it is worth noting that these visual/event-only systems have 5.1. Introduction inherent limitations, making them vulnerable to low-textured environments or suffer- ing from scale ambiguity. To mitigate visual degradations, a practical and promising strategy is to incorporate the Inertial Measurement Unit (IMU), which is low-cost and readily available in most event cameras. Furthermore, in the minimal configuration of a monocular setup, the IMU can be leveraged to recover the metric scale

and enhance the accuracy and robustness of the 60

VO system. Nevertheless, efficiently integrating learning-based event data association with IMU measurements remains an open prob- lem. In this chapter, we propose Deep Event Inertial Odometry (DEIO), the first deep learning-based event-inertial odometry framework. It is developed based on a

learning- optimization framework that leverages neural networks to predict event correspon- dences and tightly integrates IMU measurements to enhance the robustness of the odometry. More specifically, our framework decouples network training from IMU integration and operates in two phases: offline training and online optimization. Dur- ing the training phase, an event-based recurrent network learns to provide robust data associations of sparse event patches. At runtime, IMU measurements are tightly in- tegrated with the trained network within a factor graph optimization framework to achieve robust 6-DoF pose tracking. This design enables us to train the network us- ing more accessible training datasets without requiring IMU data, while still benefit- ing from IMU measurements during runtime through online optimization. The event- based recurrent network is built on the patch-based structure [26]. The selected event patches are processed using the recurrent optical flow network and the differentiable bundle adjustment (DBA) layer to establish high-confidence data associations for the consecutive event stream. The Hessian information, derived from the event-based DBA layer, is fed into an event patch-based co-visibility factor graph, facilitating its seamless fusion with the

> **IMU pre-integration** via **a** well-designed **keyframe-based** **2**

sliding win- dow optimization framework.

> **Our main contributions are** summarized **as follows**: 1. **We propose a** learning-optimization framework **that** seamlessly **integrates** **45**

the power of deep learning with the efficiency of factor graph optimization.

> **To the best of our knowledge, this is the first** event-inertial odometry **framework that employs** deep **learning** for **event** data association **and** **1**

graph optimization for pose estimation. 2. An event-based co-visibility factor graph optimization is proposed to tightly in- tegrate event patch correspondences by deriving Hessian information from the DBA along with IMU pre-integration. 3. Extensive experiments on 10 challenging event-based real-world benchmarks demon- strate the superior performance

> **of our** DEIO **compared with** over 20 **state-of-the- art methods** **3**

. 4. Codes and the preprocessed event datasets are released to facilitate further re- search in learning-based event pose tracking. Codes are available at: https: //github.com/arclab-hku/DEIO.

> **5.2 Related Work 5.2.1** Non-**learning** **140**

Approaches for Event-based VO

> **The first purely event-based VO** is **presented in** [38], **which** performs **real-time event- based SLAM through three decoupled probabilistic filters that jointly estimate the 6- DoF camera pose, 3D map of the scene, and image intensity** **3**

. EVO [40] presents a geometric 2D-3D model alignment, utilizing

> **the 2D image-like event representation** **19**

and the 3D map reconstructed from EMVS[41]. ESVO [59] is a stereo event-based VO method

> **to estimate the ego-motion through the 2D-3D edge registration on** the **time surface** **1**

. EDS [67] proposes

an event-image alignment algorithm that minimizes pho- tometric errors **between the brightness change from events and the image gradients, enabling 6-DOF** 1

monocular VO. In addition, some approaches rely on a pre-built pho- tometric 3D map [46]

or restrict the motion type to rotation-only 14

[3]. However, most of them are prone to failure

in challenging scenarios (such as fast motion). To enhance **the** 60

robustness of purely event-based VO, existing event-based SLAM methods have demonstrated good performance by incorporating additional sensors. Notably, event-inertial integration is a widely used approach to address the limitations of event-only SLAM, which provides scale awareness and continuity of estimation with minimal setup requirements. Zhu et al. [43] propose the first event-inertial odometry (EIO) method,

which fuses events with IMU through the Extended Kalman Filter 1

. Re- becq et al. [44] propose an optimization-based EIO that

detects and tracks features in the 2

5.2. Related Work edge image,

generated from motion-compensated event streams, through traditional image-based feature detection and tracking. The tracked features are then **combined with IMU** measurements via **keyframe-based nonlinear optimization** 34

. Mono-EIO [2] employs

the event-corner features with IMU measurement to deliver **real-time** 5

and ac- curate 6-DoF state estimation. Chamorro et al. [70] integrate the event-based line fea- ture with the IMU to achieve high ultra-fast pose estimation. Ultimate-SLAM [49] and PL-EVIO [72] investigate the complementary nature of events and images to present an event-image-IMU odometry (EVIO). EKLT-VIO [66] combines

the event-based and image-based feature tracker [5] **as the front end with a filter-based back end** 1

to perform the pose estimation for Mars-like sequences. ESVIO [73] proposes

the first stereo EIO and EVIO framework to estimate states through temporally and spatially event-corner feature association 1

. ESVO2 [100, 101] extends ESVO [59] and presents a direct method for stereo event cameras with an IMU-aided solution. EVI-SAM [1] introduces

the first event-based hybrid pose tracking framework 1

, merging feature-based and direct-based methods. Additionally, it can deliver satisfactory texture and dense mapping results in a variety of challenging scenarios, demonstrating performance comparable to learning-based and NeRF-based dense mapping methods. Despite such remarkable progress, current event-based systems still fall short of the robustness required for certain real- world applications, and their performance has hit a bottleneck. This is caused by

> **the sparse and motion**-dependent **nature of event data** 20

, traditional feature-based or direct- based data association is hardly established in complex scenarios, which motivates the exploration of deep learning approaches. 5.2.2

> **Learning-based Approaches for Event**-based VO Zhu **et al**. [52] employ **a** 38

convolutional neural network (CNN) with an unsupervised encoder-decoder architecture to predict pose, flow, and semi-dense depth. Hidalgo- Carrió et al. [126] estimate dense depth maps from a monocular event camera through recurrent CNN architecture. Gehrig et al. [127] apply

> **a recurrent neural network** (RNN), **which maintains an internal state** that is updated **through** 9

asynchronous events or irreg- ular images and can be queried for dense depth estimation at any timestamp. Ahmed et al. [142] develop a deep event stereo network that reconstructs spatial image features from embedded event data and leverages the event features using the reconstructed image features to compute dense disparity maps. However, most of these methods re- quire knowledge of camera motion and show poor generalization beyond the training scenarios. DH-PTAM [94] proposes a stereo event-image parallel tracking and mapping sys- tem that replaces hand-crafted features with learned features. E-RAFT [204] develops a dense optical flow method for event data association, building on the recurrent network architecture of RAFT [105], using a volumetric voxel grid representation of event data. DEVO [84] extends the DPVO [26] to accommodate the event modality also through the voxel-based representation like E-RAFT [204], demonstrating great generalization from synthetic data to seven real-world event-based benchmarks. RAMP-VO [85] intro- duces an end-to-end VO that also builds upon DPVO [26] using feature encoders to fuse events and image data. It achieves robust zero-shot transfer performance

> **to real data despite being trained only on synthetic** 44

datasets. Both DEVO [84] and RAMP-VO [85] demonstrate the powerful capability of the patch graph and recurrent network archi- tecture from DPVO [26] and RAFT [105] for event-based data association. However, the absolute scale is not observable in these monocular event-only systems without additional information or an IMU. They have inherent limitations, with failures mani- festing in various ways, such as in low-texture environments, optimization algorithm divergence, and scale ambiguity. Visual-IMU integration is the most common solution to address these limitations, which provides scale awareness and continuous estima- tion with minimal setup. Nevertheless, integrating a learning-based event network with IMU remains an unexplored territory due to the challenge of efficiently fusing the learning-based event association with IMU. This work aims to bridge this gap by proposing a combined learning-optimization framework. By leveraging deep learning, it achieves robust event-based data association while simultaneously capitalizing on the complementary strengths of IMU The key advantage

> **lies in its ability to** train **the** network independently **of IMU** 86

data, which not only maintains generalization perfor- mance but also improves adaptability across diverse scenarios. 5.3. Framework Overview Supervision Event Feature Encoder Pose Loss … … Edge Correlation DBA Flow Loss Patch Extraction … Event Voxels Camera Poses Depth Maps Patch Graph Recurrent Update Operator Score Loss Learning-Based Event Data Association DEIO Online System IMU IMU Measurement Pre-Integration Event Voxels Representation Patch Graph Recurrent Update Operator Event-IMU BA Event Stream Key Voxel (Keyframe) Hessian Information (DBA) Figure 5.1: Overview of the DEIO system. It decouples network training from IMU integration and operates in two phases: offline training and online optimization.

The main innovations of this work reside in 44

the effective integration of IMU measurements with learning-based methods. During training, a unified event-based optical flow net- work is trained to provide robust data associations of sparse event patches. At runtime, the Hessian information, derived from the DBA layer in the update operator, is utilized to tightly integrate event patch correspondence with IMU pre-integration through an event patch-based co-visibility factor graph optimization. 5.3 Framework Overview The overall design of the framework aims to tightly fuse the trainable event-based data association with traditional IMU pre-integration. Fig. 5.1 depicts the overview of our system.

The main innovations of this work reside in how 44

learning-based meth- ods and traditional graph optimization are fused together. For the front end, a deep neural network (Section 5.4) is utilized to estimate the sparse patch-based correspon- dence for the optical flow of events. On the back end, Hessian information derived from the learned DBA layer is tightly integrated with the IMU pre-integration (Sec- tion 5.5). Event reprojections are used to formulate an event patch-based co-visibility factor graph (Section 5.5), enabling up-to-scale and robust pose tracking of the entire online system (Section 5.7). This design

leverages the representational power of deep neural networks to 125

achieve robust event-based data association while simultaneously harnessing inertial measurement benefits without requiring IMU training data, thereby preserving the generalization capabilities of our DEIO. 5.4 Learning-based Event Data Association Event Encoding: Event cameras asynchronously capture illumination changes at each pixel location and produce

a stream of events. Each **event is** represented **as a tuple (t**, x, y, **p** 21

), where t denotes the trigger timestamp in microseconds at the pixel (x, y) and pi indicates polarity with p = 1 for an increase in brightness and p = −1 for a decrease. The event streams are divided into segments based on a predefined tempo- ral interval ∆t. We preprocess the event segment within an interval [ti − ∆t, ti) into a tensor Ei ∈ RD×H×W using the voxel representation [52], where D represents the num- ber of discretization steps in time. Therefore, the event-based optical flow estimation from segment i to segment j fundamentally involves establishing data correspondences between Ei and Ej. Patch Structure: For efficiency, instead of constructing dense correspondences like [105, 204], a patch-based architecture [26] is adopted to compute the

flow for a set of sparse event **patches** 42

. A p × p event patch, sampled from the event voxel E, is represented as a set of pixel coordinates P = [x, y] ∈ Rp2×2, where all pixels within the patch are assumed to have a constant inverse depth d ∈ R+. The dynamic event patch graph G is a bipartite graph where each edge is denoted as [(i, n), j], indicating the relationship between event patch n from segment i and the target segment j. The trajectory of the event patch can be obtained by reprojecting it onto all the connected event segments in the patch graph, thereby forming the sparse event data association. Network Structure: The transformation of asynchronous events into a frame-like structure allows compatibility with CNN to model the data association between patches and segments. The unified network for event-based data association inherits the re- current network architecture from [26, 84] and

consists of three primary **components**: (i) **a feature encoder that extracts** patch-based event **feature** 53

representations, includ- ing matching and contextual features; (ii) a

correlation layer that **computes** the **visual similarity between** 53

patches and segments, where patches are selected by a patch selec- tor; (iii) a recurrent update operator that handles event-patch correspondences coupled 5.4. Learning-based Event Data Association with differentiable bundle adjustment (DBA), which estimates the 2D optical flow vec- tor δ and the confidence weights ω for each patch onto the target segment in the patch graph G . More details of the event-based network structure can be seen in Appendix B. Differentiable Bundle Adjustment Layer: The DBA layer is employed to bridge the reprojection error and the predicted optical flow of event patches. This layer jointly refines camera poses and patch depth across the entire patch graph G to match the predicted patch optical flow δ by the following optimization objective: Tj′i}, {di′n = arg min ∑ F(Tji, din) 2 {{ }} T,d G ωinj , F(Tji, din) = π Tji · π−1 Pˆin, din − Pˆin + δinj // // // // (5.1) (( )) ( ) where (δinj, ωinj) is the patch-based optical flow field predicted by the event-based re- current network. The term δinj ∈ R2 denotes

> **a 2D flow vector** that indicates **how the reprojection of the** event-**patch center should be updated** 10

, and ωinj serves as the patch- wise confidence weight of the optical flow. F is the shorthand to denote the residual term on the patch center coordinates, and // · // is the Mahalanobis distance. π and π−

> **1 are the projection and back-projection functions of the event camera**. Tji = Tj−1Ti **represents the transformation from** frame i **to** 8

frame j, where Ti, Tj ∈ SE(3) denote the camera poses in the camera-to-world format. The DBA layer can efficiently backprop- agate gradients through Gauss-Newton iterations, thus enabling the pose supervision during network training. Traininig Details: Our framework decouples network training from IMU integra- tion and enables us to train the network without requiring IMU data. The learning phase is shown in Fig. 5.1. The learning process leverages supervision from poses, flow, and patch selection scores. The pose loss, Lpose, is calculated as follows: Lpose = ∑ logSE(3)(TG−T1,ijTi′j) N (5.2) i=1 // // // where TGT,ij and Tij are

> **the ground truth and** updated **poses** from **the** 53

event-based ′ // // DBA between timestamp i and j. The flow loss, Lflow,

> **measures the average distance between the estimated** and ground truth **optical flow** over edges in **the** 99

patch graph, where the frame distance between the source and target frames of each event-patch is within two frames. Regarding the event-patch selection, given the sparsity of event data in voxel, we follow [84] by adopting a patch selection scheme with the loss on selection scores Lscore.

> **The overall loss is** then **defined as a weighted** 87

combination: L = 10Lpose + 0.1Lflow + 0.05Lscore. The network

> **is implemented in PyTorch and trained** for 240K steps **on two NVIDIA RTX-3090 GPUs** 82

. Each step uses a batch size of 1 with a sequence of 15 frames where

> **the first 8 frames are used for initialization**, and **the** remaining **7 frames are** incrementally **added** 10

for training. During training, each frame contains 80 event patches, each sized 12px×12px. In the first 1000 training steps, the network estimates only the depth of patches with fixed poses, allowing it to stabilize early training. In the subsequent steps, it jointly estimates both pose and patch depth. In the edge state updater, we conduct 18 iterations, supervising the pose and flow at each iteration, while the selection score is supervised only in the final iteration. Given the strong generalization capability of the recurrent optical flow pipeline [105, 25, 26, 84], we utilize the network weights [84] pre-trained solely on the TartanAir [205]

dataset, where events are generated by the ESIM [206] simulator. 5.5 Learnable Hessian Information Extraction The key component of our proposed learning-optimization combined framework is ex- tracting the information from the learning-based event data association and integrating it with the IMU. To achieve this, we first linearize the reprojection errors from Eq. (5.1) as follows: F(ξji ⊕ Tji, din + Δdin) − F(Tji, din) = Jji Jdin ⌈ ξji [ ] Δdin ⌉ (5.3) ⌊ ⌊ ⌋ ⌋ where ξji is the Lie algebras of the updated pose in (3). Δdin denotes the updated state of the inverse depth. The Jacobians Jji, Jdin are

pose Tji and the inverse depth din, respectively. An event-patch Pin can be reprojected from segment i into segment j follows the warping function: Pjn = xjn yjn zjn ′ T 1 = Tji · π−1(P^in, din) (5.4) [ ] 5.5. Learnable Hessian Information Extraction where (xjn, yjn, zjn) is the center of the event patch at segment j in the event camera coordinate. Jji is expressed as: fx 0 fxyjn Jji = ⌈ zjn·din − zf2jnx x·djnin − fxxzj2jnnyjn fx + z2jn − zjn fxx2jn 0 f y zjn ·din − zf2jny y·djnin − fy − ⌊ fyzy2jnj2n fy xzj2jnnyjn fyzxjnjn ⌉ ⌊ (5.5) ⌋ where fx and fy are the given event camera intrinsic parameters. Jdin is given as: Jdin =⌈ fx(ztjnji·[d0]in − tzji2j[n2·]dxijnn) fy(ztjnji·[d1]in − tzji2j[n2·]dyijnn) ⌉ (5.6) ⌊ ⌊ ⌋ ⌋ where tji is the translation vector of the related transform between Ti and Tj. Therefore, the Hessian matrix of Eq. (5.3) can be computed as follows: T Hji = Jji Jdin Winj Jji Jdin (5.7) [ ] [ ] where Winj = diag(ωinj). To improve readability, the notation for [(i, n), j] edges in Hji and Winj will be omitted in subsequent equations unless otherwise specified. By decoupling the

: H ⌈ ξji T ⌉ = − Jji Jdin W F(Tji, din) ⌊ Δdin ⌊ [ ] ⌊ ⌋ (5.8) B E v ⌈ ξji ⌉ = ⌈ ⌉ ⌊ ET C ⌉ ⌈ Δdin u ⌊ ⌊ ⌋ ⌊ ⌋ ⌊ ⌋ Therefore, the following equation can be obtained: ξij = [B − EC−1ET]−1 (v − EC−1u) (5.9) Hg Vg ⏟ ⏟ ⏟ ⏟ where v is a 6K × 1 vector (with 6 DoF pose and totally K = 10 keyframe), and u is an NK × 1 vector (with totally number of N = 96 event patches for each event segment). B is the matrix with size of 60 × 60, E is the residual matrix with size of 60 × 960, and C is a diagonal matrix with size of 960 × 960. A damping factor of 10−4 is also applied to C as [26]. These matrices establish an interframe pose constraint (represented by Hg and Vg) that integrates the DBA information. After updating the camera poses Tj′i = Exp(ξji)Tji, the inverse depth of each event patch can be updated as: di′n = Δdin + din (5.10) Δdin C−1(u − ETξji) = The calculations of Eq. (5.9) and Eq. (5.10) can be efficiently performed in parallel on GPU with CUDA acceleration. All the Hessian information Hg and the correspond- ing Vg, derived from the co-visibility graph, are integrated into the factor graph where they are optimized on the CPU. The DBA contributes extensive geometric informa- tion, incorporating learned uncertainties, to the factor graph. The optimization results (updated poses and depths) are then iteratively fed back to refine the event-based op- tical flow network. This recurrent process

. 5.6 Graph-Based Event-IMU Combined Bundle Adjust- ment Unlike end-to-end approaches that use deep networks to fuse the features from two modalities (visual and IMU) and predict poses directly, our DEIO combines the neural networks with event-inertial bundle adjustment. To this end, we design a learning-optimization combined framework that tightly integrates the Hessian information from DBA and

. The full state vector of the kth

keyframes K = 10 in our implementation), is defined as: χ = [Tbwk,vbwk,bak,bgk],k = 1,2,3,... (5.11) 5.6. Graph-Based Event-IMU Combined Bundle Adjustment where Tbwk = ⌈ Rbwk twbk SE(3) is

tran⌊slation twbk ⌊ and rotation matrix Rwbk. vbwk is

the velocity of the IMU in the world frame. bak **and** bgk are **the accelerometer bias and gyroscope bias** 5

, respectively. We solve the state estimation problem by constructing a factor graph with the GTSAM library and optimizing it with the Levenberg-Marquardt. The cost function, which minimizes residuals from various factors corresponding to different aspects of data constraints,

can be written as: J(χ 3

) = ||rekvent||2Wekvent + k∑=0 ||rkimu||2Wikmu + ||rm||W2m K−1 (5.12) Eq. (5.12) contains the event-based residuals rekvent

with weight Wekvent, **the IMU pre- integration residuals** rkimu **with weight** Wikmu, **and the** 3

marginalization residuals rm with weight Wm, Given the Hessian information Hg and the corresponding Vg, the event residual factor can be written as [207]: ξ we0 rekvent = ξwe0 ξwe1 ⋯ ξwek Hg 1 ⌈ξwe1⌉ 2 [ ] . . − ξew0 ξwe1 ⋯ ξwek Vg (5.13) [ ] ⌊ξwek⌋ ⌊ ⌋ where ξ wek = ξ eb · ξ wbk , and ξ wbk = logSE(3) (Tbwk). ξ wek and ξ wbk are the Lie algebras of the event camera pose and IMU pose in kth keyframe, respectively. ξ eb is the extrinsics between the event camera and IMU.

Eventually, the IMU residual factor **can be derived as follows**: ⌈ Rwbk (twbk+**1** − twbk − vbwk ∆**t** − 12 gw∆**t2** 4

) − αˆbbkk+1 Rwbk (vbwk+1 − vbwk − gw∆t) − βˆbbkk+1 ⌉ rikmu = 2 (qwbk)−1 ⊗ qwbk+1 ⊗ (γˆbbkk+1)−1 (5.14) [bak+1 − bak] xyz ⌊ bgk+1 − bgk ⌊ ⌋ ⌋ where αbbkk+1, βbbkk+1, γbbkk+1 are the IMU pre-integration term [72]; gw is the gravity vec- tor; ∆t

is the time interval **between** keyframe **k and k + 1**; qwbk is **the** 124

quaternion of the corresponding rotation matrix Rwbk with [·]xyz extracts the vector portion. 5.7 Online EIO System Design As illustrated in Fig. 5.2, our DEIO maintains a patch-based co-visibility factor graph that

takes the raw event stream and IMU data **as input and** 3

performs 6 DoF pose esti- mation. Figure 5.2: Patch-based co-visibility factor graph for event-IMU combined bundle ad- justment.

Initialization: We use 8 event segments **for** event-only **initialization** 42

. The new event patches are added until 8 event segments are accumulated, and then run 12 it- erations

of the update operator, followed by two event-only **bundle adjustment** itera- tions. Since **the** 10

IMU provides scale awareness and the gravity direction, it is essential to initialize the states in a metric-scale local frame for better convergence. Similar to 5.7. Online EIO System Design the initialization procedure from [8, 177], where the

vision-only structure from motion (SfM) is replaced by **the** 2

event-only bundle adjustment. After the event-only initial- ization, we employ the IMU pre-integration

to establish the up-to-scale camera pose. Following linear alignment **and** 2

gravity refinement, the scale between the IMU and the event camera is restored, with further correction applied to

the depth of the event- based patch. Subsequently, **the event** 1

-IMU bundle adjustment (Section 5.6) is iteratively performed by updating the co-visibility graph based on the event-inertial initialization results. We also found that with the accurate pose estimation provided by event-based SfM, event-inertial initialization can be easily accomplished along with high-precision gravity recovery. For event-only initialization, sufficient camera motion is essential. To ensure this, we only accumulate event segments that have an average optical flow mag- nitude of at least 8 pixels. This magnitude is estimated after a single update iteration from the preceding segment. In addition, sufficient excitation is also required from the IMU, with the gyroscope's variance exceeding 0.25. Co-visibility Patch Graph Management:

When a new event **patch is added**, the **edges between that** event **patch and the previous r keyframes** 10

(r = 13 in our imple- mentations) are added to the patch graph. After appending these new edges (with the predicted event-based optical flow field (δinj, ωinj)), the update operation of the patch graph is performed, including recurrent update operator and the 2 event-IMU bundle adjustment iterations. After that, the keyframe selection is performed to maintain the keyframe-based sliding window. Keyframe Strategy:

The most recent 3 event segments **are always** considered as **keyframes. After each update** operation of **the** 10

co-visible graph,

the optical flow mag- nitude **between** the (**t − 5**)th **and (t − 3** 10

)th keyframe will be calculated. If this magni- tude is less than 60 pixels, the (t − 4)th keyframe, along with all associated edges and patch-based features, will be removed. Additionally, the event patches and keyframes are discarded

once they fall outside the optimization window. The 10

oldest keyframe that is abandoned will be marginalized, along with its associated IMU pre-integration, to construct the marginalization factor rm in Eq. (5.12). 5.8 Experiments We conduct quantitative and qualitative evaluations of our DEIO across ten challenging real-world datasets with varying camera resolution and diversity scenarios on different platforms. Specifically, in Section 5.8.1, we compare DEIO with baseline methods across multiple challenging event datasets, showcasing its superior performance and excep- tional generalization capabilities. While in Section 5.8.2 provides a detailed qualitative evaluation of DEIO. In Section 5.8.3 and 5.8.4, we assess the performance of our DEIO in night driving scenarios and indoor low-light quadrotor flights, respectively. DEIO is compared against over 20 baseline methods from various literature, including both event-based and image-based approaches. Finally,

we conduct three **ablation studies to** explore **the effects of** 107

fine-tuning on real-world data (Section 5.8.5), the influence of dif- ferent event representations (Section 5.8.6), and the impact of varying number of event patches (Section 5.8.7). 5.8.1 Comparisons with SOTA Methods in Challenge Benchmarks To ensure a fair comparison, consistent trajectory alignment protocol is required. There- fore, we employ different alignment ways and evaluation criteria according to the com- pared baseline methods. For example, regarding the DAVIS240c dataset [144], the es- timated

and ground-truth trajectories are **aligned** using **a 6-DOF transformation (in** SE(3)) over **the** 9

5 seconds, and we calculate

the mean position error (MPE) **as** a per- centage **of the total traveled distance** of **the ground truth** 5

as the evaluation criterion. In contrast, for the stereo HKU dataset [73], although MPE is also used as the metric, we align

the estimated trajectories and ground truth in SE(3) using **all** 5

available data, rather than limiting alignment to the 5-second segment, to maintain consistency with the baseline methods. While for the TUM-VIE dataset [156], the baseline methods use

the root mean squared error (**RMSE**) and **the Absolute Trajectory** Errors (**ATE** 103

) based on the estimated camera pose as the evaluation criterion. To avoid confusion, we an- notate the trajectory alignment protocol in the footnotes of each table, ensuring that all methods within the same table adhere to the same alignment protocol. The notations E, F, and I in each table represent

the use of event, frame, and IMU, respectively 1

. 5.8. Experiments DAVIS240C [144]: This is the most classical monocular event-based SLAM bench- mark, which provides event and image data with a resolution of 240×180 pixels, as well as IMU data and 200 Hz ground truth poses.

It contains extremely fast 6-Dof motion and scenes with HDR. The 2

baseline results (except for DPVO, DBA-Fusion, and DEVO) are directly taken from the original works, which also employed the same trajectory alignment protocol. As shown in Table 5.1, EVI-SAM achieves the best performance among the non-learning methods. In contrast, learning-based methods (such as DEVO) can achieve performance comparable to EVI-SAM (which combines both direct and feature-based methods) using purely event sensors, highlighting the effectiveness and strength of learning-based approaches. Meanwhile, our learning-optimization com- bined method exhibits significantly superior performance compared to other learning- based methods (DPVO, DBA-Fusion, and DEVO). Compared to DEVO, our proposed DEIO reduces the pose tracking error by up to 71%, owing to the effective integration of learning-based and traditional optimization methods. Furthermore, Fig. 5.3 illustrates the estimated trajectory of our DEIO, which shows less drift compared to DEVO, ver- ifying the contribution of IMU integration to maintain low-drifting, metric-scale pose estimation. Table 5.1:

Accuracy comparison [MPE(%)] **of our** DEIO **with other** monocular **event-based** baselines **in** DAVIS240c **dataset** 5

[144].

The estimated trajectory is aligned with the ground truth over **the** 49

first 5 seconds. Methods Modality

boxes_translation hdr_boxes boxes_6dof dynamic_translation 38

dynamic_6dof poster_translation hdr_poster poster_6dof

Average Zhu et al. [43] E+I 2.69 1.23 3.61 1.90 4.07 0.94 2.63 3.56 2.58 Henri et al. [44] E+I 0.57 0.92 0.69 0.47 0.54 0.89 0.59 0.82 0.69 Ultimate-SLAM [49] E+I 0.76 0.67 0.44 0.59 0.38 0.15 0.49 0.30 0.47 Ultimate-SLAM [49] E+F+I 0.27 0.37 0.30 0.18 0.19 0.12 0.31 0.28 0.25

Jung et al. [180] E+I 1.50 2.45 2.88 4.92 6.23 3.43 2.38 2.53 3.92 Jung et al. [180] E+F+I 1.24 1.15 0.98 0.89 0.98 1.83 0.57 0.97 38

1.07 HASTE-VIO [181] E+I 2.55 1.75 2.03 1.32 0.52 1.34 0.57 1.50 1.45 EKLT-VIO [66] E+F+I 0.48 0.46 0.84 0.40 0.79 0.35 0.65 0.35 0.54 Dai et al. [65] E+I 1.0 1.8 1.5 0.9 1.5 1.9 2.8 1.2 1.56 Mono-EIO [2] E+I 0.34 0.40 0.61 0.26 0.43 0.40 0.40 0.26 0.39 Kai et al. [83] E+I 0.36 0.31 0.32 0.59 0.49 0.23 0.18 0.31 0.35 PL-EVIO [72] E+F+I 0.06 0.10 0.21 0.24 0.48 0.54 0.12 0.14 0.24 Lee et al. [80] E+F+I 0.74 0.69 0.77 0.71 0.86 0.28 0.52 0.59 0.65 EVI-SAM [1] E+F+I 0.11 0.13 0.16 0.30 0.27 0.34 0.15 0.24 0.21 DPVO [26] F 0.02 0.71 0.59 0.09 0.05 0.20 0.49 0.44 0.32 DBA-Fusion [207] F+I 0.07 0.27 0.10 0.56 0.11 0.13 0.38 0.19 0.23 DEVO [84] E 0.06 0.06 0.71 0.09 0.08 0.06 0.14 0.44 0.21 DEIO E+I 0.07 0.09 0.05 0.06 0.04 0.04 0.06 0.08 0.06 Monocular HKU-dataset [2]: This dataset is collected using

DAVIS346 (346×260, event, image, and IMU sensor 3

), along with

a motion capture system (VICON) to obtain 3

Figure 5.3:

Comparison of the estimated position (X, Y, Z) and orientation (Roll, Pitch, Yaw 11

) results of our DEIO with DEVO [84] in the sequence of (a) boxes_6dof and (b) poster_6dof from the DAVIA240c dataset [144]. The DEIO efficiently converts scale ambiguity and aligns closely with the ground truth trajectory. Table 5.2:

Accuracy comparison [MPE(%)] of our dEIO with other image/event-based baselines in Mono-HKU dataset 2

[2].

The estimated trajectory is aligned with the ground truth over the 49

first 5 seconds. Resolution Methods Modality vicon _hdr1 vicon _hdr2 vicon _hdr3 vicon _hdr4 vicon vicon vicon vicon _darktolight1 _darktolight2 _lighttodark1 _lighttodark2 vicon _dark1 vicon _dark2 Average DAVIS346 (346×260) ORB-SLAM3 [9] VINS-MONO [8] DBA-Fusion [207] Ultimate-SLAM [49] Mono-EIO [2] PL-EIO [72] PL-EVIO [72] DEVO [84] DEIO F F+I F+

I E+I E+F+I E+I E+I F+E+I E E+I 0.32 0.96 0 9

.32 1.49 2.44 0.59 0.57 0.17 0.11 0.14 0.75 1.60 0.41 1.28 1.11 0.74 0.54 0.12 0.07 0.09 0.60 2.28 failed 0.66 0.83 0.72 0.69 0.19 0.12 0.16 0.70 1.40 failed 1.84 1.49 0.37 0.32 0.11 0.07 0.07 0.75 0.51 0.72 1.33 1.00 0.81 0.66 0.14 0.97 0.11 0.76 0.98 0.55 1.48 0.79 0.42 0.51 0.12 0.12 0.10 0.41 0.55 failed 1.79 0.84 0.29 0.33 0.13 0.15 0.11 0.58 0.55 2.65 1.32 1.49 0.79 0.53 0.16 0.12 0.13 failed 0.88 3.32 1.75 3.45 1.02 0.35 0.43 0.07 0.05 0.60 0.52 failed 1.10 0.63 0.49 0.38 0.47 0.07 0.08 0.61 1.02 1.33 1.40 1.41 0.62 0.49 0.20 0.19 0.10 pose ground truth. To mitigate the significant impact of the active infrared (IR) emit- ters from the VICON cameras on the event camera, an IR filter lens is used to eliminate IR interference. As a result, this dataset shows more pronounced thermal noise com- pared to others. Furthermore, all

sequences are recorded in HDR scenarios under **very low** illumination **or strong illumination changes** by **switching the strobe flash on and off** 3

. Table 5.2 demonstrates that DEIO outperforms all the event-based methods and decreases the average pose tracking error by at least 47%. As illustrated in Fig. 5.4(a), the estimated trajectory of DEVO suffers from significant scale loss because the abso- lute scale cannot be observed in monocular event-only odometry. This is further evi- dent from the projection of the estimated trajectory onto the XY plane, where the blue curve (baseline method) significantly deviates from the ground truth due to the scale loss. In contrast, our DEIO, despite also being based on a monocular setup, effectively overcomes scale ambiguity and aligns closely with

the ground truth trajectory. **This** improvement is **attributed to the** effective compensation provided **by** 1

Stereo HKU dataset [2, 73]. The DEIO seamlessly addresses scale ambiguity and demonstrates precise alignment with the ground truth trajectory. In contrast, the baseline estimates exhibit significant scale discrepancies: (a) The base- line trajectory suffers from drift and an overestimated scale. (b) The baseline trajectory shows an underestimated scale. Table 5.3:

Accuracy comparison [MPE(%)] **of our** DEIO **with other image/event-based** baselines **in** Stereo-HKU **dataset** 2

[73]. The entire sequence of estimated poses

is aligned with the ground truth trajectory. The 33

baseline results (DPVO, DEVO) are taken from [84], and the results of Kai et al. are taken from [83]. Methods Modality agg_translation agg_rotation agg_flip agg_walk hdr_circle hdr_slow hdr_tran_rota hdr_agg dark_normal Average ORB-SLAM3 [9] Stereo F+I 0.15 0.35 0.36 failed 0.17 0.16 0.30 0.29 failed 0.25 VINS-Fusion [199] Stereo F+I 0.11 1.34 1.16 failed 5.03 0.13 0.11 1.21 0.86 1.24 EnVIO [11] Stereo F+I failed 2.12 2.94 3.38 0.85 0.43 0.37 0.50 failed 1.51 MSOC-S-IKF [202] Stereo F+I failed 1.52 failed failed failed failed failed failed failed 1.52 DPVO [26] F 0.07 0.04 0.99 1.17 0.31 0.23 0.67 0.29 failed 0.47 DBA-Fusion [207] F+I 0.13 0.16 0.83 0.37 0.18 failed failed 0.10 0.27 0.29 Kai et al. [83] E+I 0.21 0.28 0.81 0.35 0.71 0.43 0.50 0.27 0.52 0.45 PL-EVIO [72] E+F+I 0.07 0.23 0.39 0.42 0.14 0.13 0.10 0.14 1.35 0.33 EVI-SAM [1] E+F+I 0.17 0.24 0.32 0.26 0.13 0.11 0.11 0.10 0.85 0.25 ESVO [59] Stereo E failed failed failed failed failed failed failed failed failed failed — ESVIO_AA [100] Stereo E+IMU failed failed failed failed failed failed failed failed failed failed — ESVO2 [101] Stereo E+IMU failed failed failed failed failed failed failed failed failed failed — ESIO [73] Stereo E+I 0.55 0.78 3.17 1.30 0.46 0.31 0.91 1.41 0.35 1.03 ESVIO [73] Stereo E+F+I 0.10 0.17 0.36 0.31 0.16 0.11 0.10 0.10 0.42 0.20 EVO [40] E failed failed failed failed failed failed failed failed failed failed — DEVO [84] E 0.06 0.05 0.71 0.90 0.39 0.08 0.08 0.26 0.06 0.29 DEIO E+I 0.06 0.09 0.20 0.48 0.14 0.07 0.09 0.06 0.11 0.15 Stereo HKU-dataset [73]: This

dataset contains stereo event data at 60 Hz **and stereo image frames at** 30 Hz **with** a **resolution** of **346×260, as well as IMU data at** 7

1000 Hz. It

consists of handheld sequences including rapid motion and HDR scenarios 7

. In Table 5.3,

our method outperforms all **previous works in** terms **of** 141

average positioning error. Note that event-only VO

methods, such as EVO [40], **ESVO** [59], **as well as** 1

stereo event and IMU-based methods like ESVO2 [100, 101], fail to perform successfully on any of the sequences in this dataset. Moreover, DEIO beats DEVO and increases the av- erage accuracy of the sequences up to 48%. Especially on agg_walk and agg_flip, DEVO encounters significant errors due to the limitations of relying solely on

a monocular event **camera. As shown in Fig**. 5.4(b), **the** 129

trajectory estimated by DEIO closely aligns with the ground truth, whereas the baseline suffers from significant scale loss.

It is im- portant to note that the 135

alignment

of the estimated trajectories with **the ground truth** is computed **using the** 33

publicly available trajectory evaluation tool [182]. We configure the tool to not align

the scale of the estimated trajectories with the **ground truth** 16

, ensur- ing an unbiased comparison of scale discrepancies. This underscores the importance of leveraging IMU to mitigate the degradation and scale ambiguity inherent in purely event-based VO. Table 5.4:

Accuracy comparison [MPE(%)] **of our** DEIO **with other image/event-based** baselines **in** VECtor **dataset** 2

[157]. The entire sequence of estimated poses is aligned with the ground truth trajectory. Methods Modality corner- slow desk- normal sofa- fast mountain- corridors- corridors- fast dolly walk units- dolly units- scooter average ORB-SLAM3 [9] Stereo F+I 1.49 0.46 0.21 2.11 1.03 1.32 7.64 6.22 2.81 VINS-Fusion [199] Stereo F+I 1.61 0.47 0.57 failed 1.88 0.50 4.39 4.92 2.05 DPVO [26] F 0.30 0.09 0.07 0.11 0.56 0.54 1.52 1.67 0.61 DBA-Fusion [207] F+I 1.72 0.48 0.43 failed 1.37 0.59 1.23 0.48 0.90 EVO [40] E 4.33 failed failed failed failed failed failed failed 4.33 ESVO [59] Stereo E 4.83 failed failed failed failed failed failed failed 4.83 Ultimate-SLAM [49] E+F+I 4.83 2.24 2.54 4.13 failed failed failed failed 3.44 PL-EVIO [72] E+F+I 2.10 3.66 0.17 0.13 1.58 0.92 5.84 5.00 2.92 ESVIO [73] Stereo E+F+I 1.49 0.61 0.17 0.16 1.13 0.43 3.43 2.85 1.41 EVI-SAM [1]

E+F+I 2.50 **1**.45 **0.98 0**.38 1.58 **1**.27 **0**.59 **0** 38

.83 1.32 DEVO [84] E 0.59 0.11 0.38 0.37 0.51 1.04 0.48 0.88 0.55 DEIO E+I 0.50 0.13 0.44 0.24 0.78 0.74 0.35 0.35 0.44 VECtor [157]: This

dataset consists of a hardware-synchronized sensor suite that includes stereo event cameras (640×480), **stereo standard cameras** 7

(1224×1024), and

IMU. It covers the full spectrum of 6 DoF motion dynamics, environment complexi- ties, **and illumination conditions for both small and large-scale scenarios** 95

. As presented in Table 5.4, our proposed DEIO achieves remarkable results on average. It surpasses all image-based baselines with high-quality frames and even outperforms Ultimate- SLAM, PL-EVIO, ESVIO,

and EVI-SAM on over 75% of the sequences, which utilize event, image, and IMU. As for the learning-based image-IMU approach, DBA-Fusion utilizes dense learning-based data association with high-resolution images (1224×1024), but it still performs worse than our DEIO which employs sparse learning-based data association with low-resolution event data (640×480). Similarly, DPVO, which relies on a sparse learning-based image solution, also underperforms compared to our DEIO. These performance gaps may be attributed to image degradation in challenging scenar- ios or the absence of IMU information. Our DEIO also outperforms DEVO on average in large-scale sequences, thanks to the complementary integration of the event and IMU sensors, while other monocular visual-only methods struggle with scale ambiguity and drift. Table 5.5: Accuracy comparison [ATE/RMSE (cm)] of our DEIO with other event-based baselines in TUM-VIE dataset [156]. The entire sequence of estimated poses

is aligned with the ground truth trajectory. The 33

baseline results (EVO, ESVO, and ES-PTAM) are taken from [88], while DH-PTAM and Ultimate-SLAM are sourced from [94], DEVO is from [84], and ESVIO_AA, ESVO2 are from [101]. Methods

Modality 1d-trans 3d-trans mocap- **6dof desk desk2 Average EVO** 16

[40] E 7.5 12.5 85.5 54.1 75.2 47.0 ESVO [59] Stereo E 12.3 17.2 13.0 12.4 4.6 11.9 ESVIO_AA [100] Stereo E+I 3.9 18.9 failed 9.00 9.5 10.3 ESVO2 [101] Stereo E+I 3.3 7.3 3.2 6.2 4.0 4.8

ES-PTAM [88] **Stereo E 1.05 8.53 10.25 2**.5 **7**.2 **5** 16

.9 DH-PTAM [94] Stereo E+F 10.3 0.7 2.4 1.6 1.5 3.9 Ultimate-SLAM [49] E+F+I 3.9 4.7 35.3 19.5 34.1 19.5 DEVO [84] E 0.5 1.1 1.6 1.7 1.0 1.2 DEIO E+I 0.4 1.1 1.4 1.4 0.7 1.0 TUM-VIE [156]: This dataset is recorded with stereo high-resolution event cam- eras (1280×720) mounted on a helmet, capturing footage from an egocentric viewpoint. We evaluate our method on sequences recorded in a motion capture room that featured various 6-DOF camera movements. The results in Table 5.5 demonstrate that DEIO out-performs all other methods

on four out of five sequences, despite **DH-PTAM** [94] utiliz- ing **four cameras of the setup (stereo** 14

events and stereo images). Our DEIO significantly outperforms ESVO2 [101, 73] and improves average accuracy by up to 79%. Notably, ESVO2 relies on stereo event and IMU setup, while our DEIO achieves superior results

using only a monocular event camera and 79

IMU. This highlights that our approach, us- ing

only a monocular event camera and 79

IMU, can recover scale comparable to that of a stereo event setup. Moreover, by leveraging learning-based event data association,

our method significantly **outperforms the baseline**, which relies **on** 62

direct-based event data association.

It is worth noting that, due to the **relatively** 62

low complexity of the selected sequence, the learning-based event data association method is already highly saturated in these scenarios. For instance, the performance of the purely event-based VO (DEVO) achieves an accuracy of 1.2 cm. While the introduction of IMU in DEIO still brings an enhancement by the additional constraints from the IMU measurements. EDS [67]: This dataset provides synchronized

monocular event (640×480), image (640×480), and IMU data from handheld devices with ground truth poses.

> **To the best of our knowledge**, this work presents **the first results for** event-inertial **pose estimation** using the **EDS** dataset. As shown **in** 14

Table 5.6, our DEIO outperforms the image-based baselines, including ORB-SLAM3, DPVO, and DBA-Fusion. Moreover, DEIO achieves an average improvement of 30% over DEVO and demonstrates performance compa- rable to RAMP-VO, a learning-based VO system that leverages both event and image modalities. In the case of DBA-Fusion [207], despite operating within a dense learning- based VIO framework, DEIO demonstrates superior performance, showing the distinct advantages of event cameras in challenging scenarios. For instance, the HDR charac- teristics in the sequence ziggy_hdr cause severe image degradation (as illustrated in Fig. 5.5). In contrast, DEIO leverages the rich perceptual information provided by the event camera, achieving an 80% improvement in accuracy over DBA-Fusion [207] in ziggy_hdr. Table 5.6: Accuracy comparison [ATE/RMSE (cm)] of our DEIO with other image/event-based baselines in EDS dataset [67]. The entire sequence of estimated poses

> **is aligned with the ground truth trajectory. The** 33

baseline results (ORB-SLAM3, DPVO, and DEVO) are taken from [84], while RAMP-VO is sourced from [85]. Methods Modality peanuts_dark peanuts_light peanuts_run rocket_dark rocket_light ziggy ziggy_hdr ziggy_flying all_chars Average ORB-SLAM3 [9] Stereo F+I 6.15 27.26 16.83 10.12 32.53 26.92 81.98 20.57 21.37 27.08 DPVO [26] F 1.26 12.99 25.48 27.41 63.11 14.86 66.17 10.85 95.87 35.33 DBA-Fusion [207] F+I 7.26 149.36 134.92 114.24 117.09 173.50 140.51 11.81 126.36 108.33 DEVO [84] E 4.78 21.07 38.10 8.78 59.83 11.84 22.82 10.92 10.76 21.00 RAMP-VO [85] E+F 1.20 9.03 13.19 7.20 17.53 19.05 28.78 6.35 28.61 14.55 DEIO E+I 1.77 16.27 19.96 8.91 15.41 10.39 23.82 3.84 31.55 14.66 MVSEC [147]: This dataset is collected by stereo DAVIS346. Specifically, we se- lect the indoor_flying sequence, which features arbitrary 6-DOF motion

> **from a drone flying in a** motion capture **room**. The forward **and** 87

backward movements of the drone, along with pauses at the start and end points that produce almost no events, make Figure 5.5: The estimated trajectories of our DEIO against the GT in the sequence of ziggy_hdr and rocket_dark from the EDS [67] dataset. The image view (visualization- only) demonstrates the lack of perceptible information under low-light conditions, while the event view, though perceptible, remains susceptible to interference from the infrared light of the motion capture system. Thanks to our robust learning-based event data association, the trajectories estimated by DEIO align remarkably closely with the GT. these sequences particularly challenging. Additionally, the sparse events generated far from the camera pose an additional challenge for event-based SLAM. In Table 5.7, DEIO surpasses the event-based baseline, especially for the Flying_4, where it attains an RMSE of 40% lower than DEVO. Despite ESVIO [73] utilizing a setup that com- bines stereo images, stereo events, and IMU data, DEIO, relying only on monocular event data and IMU, achieves an average accuracy improvement of over 80%. As for the learning-based VIO method (DBA-Fusion) that relies on dense data association, it failed on three out of the four sequences. This indicates that although deep learning Table 5.7:

> **Accuracy comparison** [MPE (%)] **of our** DEIO **with other image/event-based** baselines **in** MVSEC **dataset** 2

[147]. The entire sequence of estimated poses

> **is aligned with the ground truth trajectory. The** 33

DEVO result is taken from [84]. Methods Modality

> **Flying_1 Flying_2 Flying_3** 38

Flying_4 Average ORB-SLAM3 [9] Stereo F+I 5.31 5.65 2.90 6.99 5.21 VINS-Fusion [199] Stereo F+I 1.50 6.98 0.73 3.62 3.21 EVO [40] E 5.09 failed 2.58 failed 3.84 ESVO [59] Stereo E 4.00 3.66 1.71 failed 3.12

Ultimate-SLAM [49] E+F+I failed failed failed 2.77 2.77 PL-EVIO [72] E+F+I 1.35 1.00 0.64 5.31 2.08 ESVIO [73] Stereo E+F+I 0.94 1.00 0.47 5.55 1.99 DBA-Fusion [207] F+I 2.20 failed failed failed 2.20 DEVO [84] E 0.26 0.32 0.19 1.08 0.46 DEIO E+I 0.24 0.21 0.12 0.78 0.34 methods can provide strong data association capabilities, the degradation of images in challenging environments limits their performance compared to the event modality. Another example of this is illustrated in Fig. 5.6. Figure 5.6: The estimated trajectories of our DEIO against the GT in the sequence of in- door_forward_7 from the UZH-FPV [15] dataset. The image view (visualization-only) demonstrates the condition under low texture, HDR, and motion blur. UZH-FPV [15]: This dataset, captured using the monocular DAVIS346, consists of high-speed trajectories, including

fast laps around a racetrack with drone racing gates and **free-form** paths **around** various **obstacles** 5

. As shown in Table 5.8 and Fig. 5.6, this dataset poses significant challenges for existing methods, with even advanced learning-based VO approaches like DPVO failing to maintain reliable tracking across all sequences. Additionally, incorporating IMU measurements does not resolve these challenges, such as learning-based VIO methods (DBA-Fusion), which also fail to com- plete any sequences. This is due to the motion blur caused by rapid movement, which makes it difficult to effectively establish data association for the image sensor, even if these methods are equipped with a powerful learning network. In contrast, our DEIO achieves higher average performance than all baseline methods across all sequences, demonstrating greater resilience to the fast flight conditions. The modest improvement from DEVO to DEIO can be attributed to significant IMU bias induced by aggressive motion, which limits the potential benefits of IMU-aided optimization. Nevertheless, our DEIO still successfully handles complex flight scenarios that involve challenging maneuvers, such as back-and-forth motion, abrupt directional changes, and loops. Table 5.8:

Accuracy comparison [MPE (%)] **of our** DEIO **with other image/event-based** baselines **in UZH-FPV dataset** 2

[15]. The entire sequence of estimated poses

is aligned with the ground truth trajectory. The 33

baseline results (DPVO, DEVO) are taken from [84]. Methods Modality 3 5 Indoor_forward 6 7 9 10 Average VINS-Fusion [199] Stereo F+I 0.84 failed 1.45 0.61 2.87 4.48 2.45 ORB-SLAM3 [9] Stereo F+I 0.55 1.19 failed 0.36 0.77 1.02 0.78 EVO [40] E failed failed failed failed failed failed — DPVO [26] F failed failed failed failed failed failed —

VINS-MONO [8] **F+I 0**.65 **1**.07 **0**.25 0.37 **0** 71

.51 0.92 0.63 DBA-Fusion [207] F+I failed failed failed failed failed failed — Ultimate SLAM [49] E+F+I failed failed failed failed failed failed — PL-EVIO [72] E+F+I 0.38 0.90 0.30 0.55 0.44 1.06 0.60 DEVO [84] E 0.37 0.40 0.31 0.50 0.61 0.52 0.45 DEIO E+I 0.39 0.36 0.33 0.32 0.59 0.55 0.42 Table 5.9: Accuracy comparison [ATE/RMSE (cm)] of our DEIO with stereo event- based methods in DSEC dataset [153]. The entire sequence of estimated poses

is aligned with the ground truth trajectory. The 33

baseline results (ESVO and ESVIO_AA) are taken from [100], and ES-PTAM is sourced from [88]. Methods Modality

a b dsec_**zurich_city_04 c** 67

d e Average ESVO [59] Stereo E 371.1 116.6 1357.1 2676.6 794.9 1032.9 ESVIO_AA [100] Stereo E+I 105.0 66.7 637.9 699.8 130.3 527.9 ES-PTAM [88] Stereo E 131.62 29.02 1184.37 1053.87 75.9 494.9 ESIO [73] Stereo E+I 543.5 295.1 896.2 2977.0 2326.4 1587.8 ESVIO [73] Stereo E+F+I 371.2 445.8 1892.7 921.7 352.0 596.9 DEIO E+I 80.6 35.4 413.8 207.6 86.1 164.5 DSEC [153]: This dataset is collected with stereo event cameras (640×480) mounted

on a car driving through the streets of Switzerland. Since the 130

DSEC dataset lacks ground truth trajectories, we use the odometry trajectories provided by [88], which are obtained through a LiDAR-IMU-based method, as ground truth. As presented in Table 5.9, our DEIO outperforms the stereo event methods (ESVO, ESVIO_AA, and ESIO)

by large margins on all sequences (at least 66.7% lower 14

RMSE). Fig. 5.7 presents a qualitative comparison of the estimated trajectories of our DEIO and these stereo event-based baselines. The results from our DEIO align more closely with the ground truth, despite using a monocular setup, while the other methods employ a stereo setup. This demonstrates that DEIO can achieve comparable scale estimation to these stereo setups while providing superior state estimation results. Furthermore,

as can be seen from the results, the purely event-based 62

odometry (ESVO) exhibits the most signifi- cant drift, highlighting the importance of complementarity between events and IMU sensors. (a)

dsec_zurich_city_04_a (b) dsec_zurich_city_04_e (c) dsec_zurich_city_04_b (d) dsec_zurich_city_04_c (e) dsec_zurich_city_04 1

_d Figure 5.7: Comparison of Estimated trajectories on the DSEC dataset [153]. 5.8.2 Qualitative Evaluation on Challenge Benchmarks In the previous

section, we presented a quantitative evaluation of our 78

DEIO frame- work, showcasing its superior performance and exceptional generalization capabilities across nine challenging datasets.

To the best of our knowledge, this work constitutes the most comprehensive benchmarking effort in the 41

domain

of event-based SLAM to date. The detailed characteristics of 19

these nine benchmarks, including their inherent challenges and comparative analyses with baseline methods, have been extensively discussed. Building on this foundation, this section delves into a qualitative assess- ment of these demanding scenarios. Fig. 5.5(b) depicts the evaluation of DEIO in an environment characterized by extreme darkness (as evident in the image view) and in- terference from a ceiling-mounted motion capture system. Despite these adverse con- ditions, our framework demonstrated remarkable robustness, with the estimated tra- jectory closely aligning with the ground truth. Similarly, Fig. 5.6 highlights the efficacy of DEIO in a high-speed drone flight scenario. This setting introduced additional chal- lenges, including low-texture environments, intense lighting conditions, and motion blur induced by rapid movement. Notably, neither learning-based sparse VO (DPVO) nor learning-based dense VIO (DBA-Fusion) can process any sequence successfully un- der these conditions. In contrast, DEIO exhibited both robustness and high precision. More video demonstrations, which provide a more intuitive representation of DEIO, are available on the project website. Furthermore, in this section, we also shift our focus to a qualitative analysis by visualizing the estimated trajectories generated by our DEIO within these datasets. As illustrated in the Fig 5.8, 5.9, and 5.10, it is clear that the trajectories estimated by our proposed DEIO align remarkably well with the ground truth across all nine challenging scenarios. These trajectories are highly complex and represent substantial challenges, highlighting the robustness and precision of DEIO in complex environments. 5.8.3 Test on Night Driving Scenarios Driving scenarios pose challenges for event-based state estimation, especially at night- time, where rampant flickering light (e.g., from LED signs) generates an

overwhelming number of noisy events. Additionally, the movement of vehicles, such as sharp turns, sudden stops, and other abrupt movements, can further complicate the event-based es- timator. In this section, we select the Dense_street_night_easy_a sequences of the ECMD dataset [6], which feature numerous flashing lights from vehicles, street signs, build- ings, and moving vehicles, making event-based SLAM more difficult. This dataset is recorded with two pairs of stereo event cameras (640×480 and 346×260) on a car driven through various road conditions such as streets, highways, roads, and tunnels in Hong Kong. Our DEIO runs on the event from the DAVIS346 and the IMU sensor, while

the image frame output of the DAVIS346 **is only used for illustration purposes** 3

. Fig. 5.11 shows a small drift with a 4.7 m error of our estimated trajectory on the 620 m drive.

To the best of our knowledge, we present **the first results on** 7

pose tracking for night (a) boxes_6dof in DAVIS240c [144] (b) poster_translation in DAVIS240c [144] (c) vicon_dark1 in Mono-HKU [2] (d) vicon_hdr4 in Mono-HKU [2] (e) aggressive_translation in Stereo-HKU [73] (f) hdr_agg in Stereo-HKU [73] Figure 5.8:

The Estimated trajectories of our DEIO **against the ground truth** 4

in the dif- ferent challenging benchmarks. (a) corridors_walk1 in Vector [157] (b) units_scooter1 in Vector [157] (c) mocap-6dof in TUM-VIE [156] (d) mocap-desk2 in TUM-VIE [156] (e) indoor_forward_6 in UZH-FPV [15] (f) indoor_forward_7 in UZH-FPV [15] Figure 5.9:

The Estimated trajectories of our DEIO **against the ground truth** 4

in the dif- ferent challenging benchmarks. (a) indoor_flying_1 in MVSEC [147] (b) indoor_flying_3 in MVSEC [147] (c) dsec_zurich_city_04_a in DSEC [153] (e) peanuts_dark in EDS [67] (d) dsec_zurich_city_04_e in DSEC [153] (f) ziggy_flying in EDS [67] Figure 5.10:

The Estimated trajectories of our DEIO **against the ground truth** 4

in the different challenging benchmarks. driving scenarios using event and IMU odometry. The earliest attempt in this area is ESVIO [73, 6], which utilizes a combination of stereo events, stereo images, and IMU data, whereas DEIO operates with a monocular setup. B A A B GT DEIO Figure 5.11: The estimated trajectory of our DEIO in the night driving scenarios [6] and its comparison against the GNSS-INS-RTK as ground truth. The image frame is for visualization only. 5.8.4 Test on Dark Quadrotor-Flight In this section, we evaluate our DEIO in a dark quadrotor flight experiment.

The quadrotor is commanded to track a circle **pattern with** 5

1.5 m

in radius and 1.8 m **in height, shown in Fig**. 5.12(b). **The** illuminance in **the** 4

environment is quite low, re- sulting in minimal visual information captured by the onboard camera (Fig. 5.12(a)).

The total length of the trajectory is 2

60.7 m, with an MPE of 0.15 and an average pose tracking error of 9 cm.

We further illustrate the estimated trajectories (translation and rotation) of our DEIO **against the ground truth, as well as their** 34

corresponding errors

(Fig. 5.12(c)).

The translation errors in the X, Y, and Z dimensions are all within 0.5 m, **while the rotation** errors **of the Roll and Pitch dimensions are within** 6◦, **and the one in the Yaw dimension is within** 3◦. To **the** best **of our** **2**

knowledge, this is also the first implementation of monocular event-inertial odometry for pose tracking in dark flight environments, while the previous works [49, 73] rely on the image-aided event-IMU estimators. (

a) (b) (c) Figure 5.12: (**a** **146**

) Our quadrotor platform (the image is for visualization-only); (b)

The estimated trajectory of our DEIO in **the quadrotor flight and its comparison against the ground truth** **2**

; (c)

The position, orientation, and the corresponding errors of DEIO **in** quadrotor **flight compared with the ground truth from VICON** **2**

. 5.8.5 Ablation Study on Real-world Data Fine-tuning We fine-tune the event-based optical flow network using real data from the Vector dataset [157] to bridge the domain adaptation gap. To avoid conflicts, we additionally use data sequence not listed in Table 5.4 for fine-tuning and evaluate the fine-tuned model on the sequence as Table 5.10. Only small-scale data sequences from Vector, which use a depth camera for dense depth information, are employed for fine-tuning. In contrast, large-scale data sequences of Vector, which rely on LiDAR for sparse depth and lack dense depth information, are not used for the re-training. Fig. 5.14 illustrates the training details of the fine-tuning process, and Fig. 5.13 presents the validation re- sults on the training dataset. Table 5.10 shows the accuracy comparison of our DEIO training on synthetic/real-world datasets, note that since the fine-tuning process only uses small-scale data sequences, so we evaluate only the small-scale sequence. The results demonstrate that the introduction of real data reduces

the domain adaptation **gap between** synthetic **and real-world data** **21**

and improves localization accuracy on both DEVO and our DEIO. Although the performance improvement is not particularly sig- nificant, possibly due to insufficient data volume from real data, it still confirms that the performance gap can be further increasing when trained on real data. Therefore, future work on improving learning-based event SLAM could explore the use of larger and more diverse real-world datasets for training. Table 5.10: Accuracy Comparison [MPE(%)] of Our DEIO Training on Synthetic/Real- world Datasets in Vector [157] Methods Training corner- slow desk- normal sofa- fast mountain- fast average DEVO [84] sim real 0.59 0.94 0.11 0.18 0.38 0.23 0.37 0.07 0.36 0.36 DEIO sim real 0.50 0.54 0.13 0.22 0.44 0.37 0.24 0.16 0.33 0.32 5.8.6 Ablation Study on Event Representations We further investigate different frame-based event representations for the learning- based association, which show superior performance in non-learning event-based SLAM frameworks [40, 72, 73, 59, 100]. The results are presented in our learning-optimization combined framework which operates on the frame-based event representations and Figure 5.13: Validation results on the training dataset during the fine-tuning process using real-world data from the Vector dataset [157]. (a) The training loss, including both flow loss and the poss loss (b) The learning rate and the pixel error (px1) of the event-based optical flow network Figure 5.14: The training details of the fine-tuning process using real-world data from the Vector dataset [157]. IMU with the network weights provided by [26]. It can also be observed from Table 5.11 that these frame-based event representations do not perform as well as the event voxel grid representation. This difference may be due to the fact that these frame-based event representations are more suitable for extracting high-level appearance information but lose the temporal information. However, since the SLAM problems heavily rely on ge- ometric features rather than appearance ones, a representation that more effectively en- codes both spatial and temporal geometry would yield better results. Indeed, straight- forward adaptations of learning-based VIO for event processing often prove inade- quate, specialized event-IMU fusion methods, like our DEIO, can achieve

better gener- alization and higher accuracy in real-world challenge benchmarks. In future work, we may further investigate trainable event representations for self-supervised learning of suitable event representations in learning-based SLAM or for better perception of the geometric features. Table 5.11: Accuracy Comparison [MPE(%)] of DEIO with Different Event Representa- tions in DAVIS240c Dataset [144] Representation Voxel Event-Frame TS TS p boxes_

translation 0.07 **0**.41 **0**.43 **0**.43 **hdr_boxes 0**.09 **0**.80 **0**.80 **0** **9**

.76

boxes_6dof 0.05 **0**.72 **0**.75 **0**.74 dynamic_**translation 0**.06 **0**.42 **0**.10 **0**.66 **9**
dynamic_6dof 0.04 **0**.68 **0**.68 **0**.71 poster_**translation 0**.04 **0**.49 **0**.48 **0**.60
hdr_poster **0**.06 **0**.71 **0**.87 **0**.64 **poster**

_

6dof 0.08 0.54 **0**.54 **0**.52 **Average 0**.06 0.60 **0**.58 **0** **3**

.63 5.8.7 Ablation Study on Number of Event Patches

In this section, we conduct an ablation study to systematically **115**
investigate **the** impact **of the**

number of event patches per voxel on the system's performance. The quan- tity of event patches

plays a critical role in the strength **of learning** **97**

-based event data association, as a larger number of patches introduces more event constraints into the factor graph. Through carefully designed experiments, we aim to determine the op- timal patch configuration that strikes a balance between computational efficiency and system performance. The results are presented in Table 5.12,

on each sequence, **we run five trials and report the median result** **14**

. Table 5.12: Accuracy comparison [MPE(%)] of DEIO with different numbers of event patches in DAVIS240c dataset [144]. Event Patches Per Voxel 48 96 120 boxes_translation 0.07 0.04 0.04 hdr_boxes 0.09 0.08 0.09 boxes_6dof 0.07 0.08 0.08 dynamic_translation 0.08 0.06 0.06 dynamic_6dof 0.06 0.05 0.06 poster_translation 0.04 0.08 0.04 hdr_poster 0.08 0.06 0.06 poster_6dof 0.08 0.07 0.08 Average 0.071 0.065 0.063 5.8.8 Runtime Analysis Fig. 5.15 further illustrates

the real-time performance of our DEIO **on an** **2**

Nvidia RTX 3090 GPU. Our DEIO, configured with 96 patches per event voxel (P96), achieves an average processing speed of 18.4 voxels per second (VPS). Compared to the variant 28 P96 (w/o IMU) [0.219] P96 [0.065] 26 P48 [0.071] P120 [0.063] V oxels per Second 24 22.7 22 22.1 20 18.4 18 16 15.4 14 f 6 d o d o f e s o f x e s i c 6 b o x 6 d b o h d r h d r b o x e s translatioynnam ynamic translation poster poster poster translation d d Figure 5.15: Runtime performance (voxels per second) of our DEIO using 48 (P48), 96 (P96), and 120 (P120) event patches per voxel, as well as a 96-patch version without IMU input (P96 w/o IMU). Values in the brackets indicate the average MPE (%) over all sequences. that excludes IMU data (P96 w/o IMU), the P96 configuration demonstrates a signifi- cant accuracy improvement of 69.0%, with only minimal runtime overhead of 4.3 VPS. The number of event patches is also a key factor influencing the overall system speed. For instance, the P48 variant achieves an average MPE of 0.071 while maintaining a runtime of 22.1 VPS, which is comparable to that of the P96 w/o IMU configuration (22.7 VPS). However, increasing the number of event patches further (P120) leads to diminishing returns in accuracy improvement while significantly increasing computa- tional demands. To achieve an optimal balance between performance and frame rate, we have adopted 96 patches per voxel as the default configuration in our experiments. 5.8.9 Discussion

It is worth noting that this chapter adopts **a learning-based** 91

approach exclusively for event data association, while maintaining graph-based optimization for IMU integration. This design is driven by two primary considerations: (1) learning-based IMU bias estimation methods often exhibit limited generalization across different IMU hardware due to variations in specifications and operating frequencies [208], and (2) Such meth- ods rely on labeled training data, which is challenging to obtain, and their computa- tional requirements can compromise real-time performance. Our proposed learning- optimization combined framework effectively addresses these challenges, demonstrat- ing remarkable generalization capabilities, as evidenced in the previous section. De- spite being trained solely on synthetic data,

our approach outperforms over 20 **state-of- the-art methods** 61

across 10 challenging real-world event benchmarks while maintaining real-time efficiency. 5.9 Integrating the Proximity Loop Closure and Global Bundle Adjustment We further enhance the DEIO system by incorporating a proximity mechanism to mit- igate drift through loop closure, term as DEIO+ in this chapter. The workflow of our DEIO+ is illustrated in Fig. 5.16.

To improve global consistency, we **periodically** insert **long-range edges into the** co-visibility factor **graph**, update the **optical flow** of event patches, **and** perform **global bundle adjustment to** refine **all poses and** 42

event patch depths. This approach

detects loops via assessing **the** camera's **proximity to** previ- ously **visited locations** 42

.

Specifically, we permanently store the event **patch features for all previous time steps** establish **uni-directional** edges linking **these patches to** currently **observed frames**. When **the** 42

nears a previously visited location, we add directed edges from older frames to newer ones still being used by the odometry component. Figure 5.16: Workflow of the event-based DBA with global bundle adjustment (DEIO+) in the optimization framework. 5.9. Integrating the Proximity Loop Closure and Global Bundle Adjustment To enable efficient global bundle adjustment, we simultaneously optimize both event recurrent optical flow factors

and loop-closure factors within **the same** optimiza- tion. **This** requires **a** 42

learnable Hessian information extraction process capable of han- dling global optimization without significantly disrupting other components. By lever- aging the sparse structure of the Hessian, we implement a CUDA-accelerated block- sparse representation for Eq. (5.8) to efficiently extract learnable Hessian information for both event factors and proximity loop closure factors. The technical details of this efficient implementation are provided in the source code. While the proximity loop closure residual for the factor graph can be represented as: rpkroximity = 1 Xe ′ ′ 2 ⌈ (5.15) ⌊ Xp ⌉ Hgg[Xe Xp] − ⌈ Xe Xp ⌉ Vgg ⌊ ⌋ ⌋ ⌊ ⌋ ⌋ where Xp is the Lie algebras

of the event camera pose between **the current** pose **and the** previous one for **the** 1

loop connections. Hgg and Vgg are the hessian information for ′ ′ the pose-only variable with the loop closure. Please note that, beyond the proximity loop closure, our framework also supports traditional visual retrieval methods, such as learning-based event matching [106] or conventional event loop closure techniques [2]. Table 5.13 and 5.14 present comparative experiments

to validate the improvements **brought by the addition of** 67

proximity loop-closure factors. For the DAVIS240C [144] sequence, DEIO has already achieved a "saturated" high accuracy of 0.06, yet there is still a slight improvement when incorporating GBA constraints. As for the UZH- FPV dataset [15], the introduction of GBA leads to a performance boost of over 50% for DEVO (with the mean MPE improving from 0.45 to 0.22). However, while DEIO+ (E+I) shows a greater improvement compared to DEIO and DEVO with more than 69% performance improvement. Table 5.13: Accuracy Comparison of DEIO+ in DAVIS240c Dataset [144]. Unit: MPE(%), 0.06

means the average error would be 0.06m **for 100m motion. Aligning 5 seconds [0-5s] of the estimated trajectory with the ground truth** 1

. Methods Modality boxes hdr

boxes _translation _boxes _6dof _translation **dynamic dynamic _6dof poster** hdr **_translation _poster poster _6dof** 38

Average DEVO [84] E 0.06 0.06 0.71 0.09 0.08 0.06 0.14 0.44 0.21 DEIO E+I 0.07 0.09 0.05 0.06 0.04 0.04 0.06 0.08 0.06 DEIO+ E+I 0.03 0.06 0.05 0.05 0.05 0.04 0.05 0.06 0.05 Table 5.14: Accuracy Comparison of DEIO+ in UZH-FPV [15]. Unit: MPE(%).

Aligning the whole ground truth trajectory with estimated poses 1

. Methods Modality 3 5 Indoor_forward 6 7 9 10 Average DEVO [84] E 0.37 0.40 0.31 0.50 0.61 0.52 0.45 DEIO+ E 0.13 0.30 0.23 0.35 0.21 0.12 0.22 DEIO [102] E+I 0.39 0.36 0.33 0.32 0.59 0.55 0.42 DEIO+ E+I 0.04 0.32 0.14 0.06 0.13 0.08 0.13 5.10 Conclusion

In this chapter, **we propose** DEIO, **a deep learning-based** 107

event-inertial odometry method. An event-based deep neural network is utilized to provide accurate and sparse asso- ciations of event patches over time, and DEIO further tightly integrates it with the IMU during the graph-based optimization process to provide robust 6 DoF pose track- ing. Evaluation on ten challenging event-based benchmarks demonstrates that DEIO outperforms both image-based and event-based baselines. We have shown that the learning-optimization combined framework for SLAM is a promising direction. 5.11 Related Publications 1. Guan Weipeng*, Lin Fuling*, Chen Peiyu, Lu Peng, “DEIO: Deep Event Inertial Odometry“. [*Co-first Author] Chapter 6 Event-based 3D Dense Reconstruction The previous chapters have explored event-based pose tracking, covering feature-based, direct-based, hybrid, and learning-based approaches, all of which effectively address

the localization problem of SLAM. **In** this **chapter, we** 145

further investigate the

3D dense mapping using the **monocular event camera. The** 1

event-based mapping method re- covers

the dense and colorful depth of the scene through the image-guided 1

approach.

Subsequently, the appearance, texture, and surface mesh of the 3D scene can be re- constructed **by fusing the dense depth map from multiple viewpoints using truncated signed distance function (TSDF) fusion.** 1

To the best of our knowledge, this is the first non-learning work to realize event-based dense mapping

. This chapter is based on the mapping module of our work [1]. More demonstrations can be seen in the project website: https://kwanwaipang.github.io/EVI-SAM/. 6.1 Introduction

Event camera captures high-quality data either in extreme lighting scenes or high- speed motion, which paves the way to tackle challenging scenarios in robotics such as Visual Odometry (VO), Visual Inertial Odometry (VIO), Simultaneous Localization and Mapping (SLAM), etc. This novel sensor offers many advantages over standard cam- eras, **including high temporal resolutions (µs-level), high dynamic range (HDR, 140 dB), low power consumption, and immunity to motion blur** [4]. **However, the event** 1

Chapter 6. Event-based 3D Dense Reconstruction y Event Mat Time Surface with Polarity t x

Event Stream Event-based Dense Mapping Event-based Hybrid Tracking Camera Trajectory with Texture Point Cloud Figure 6.1: **Our EVI-SAM enables the recovery of both the camera pose and dense maps of the scene. The tracking module** 1

(detailed in Chapter 4) leverages the re-projection

constraint from the feature-based method and the relative pose constraints from the direct-based method within an event-based hybrid tracking framework. The mapping module represents a pioneering event-based dense mapping framework, distinguished as a non-learning method that can conduct real-time dense and texture mapping on a standard CPU 1

.

camera exclusively responds to moving edges within the scene, inherently resulting in sparse and asynchronous output. This poses a challenge in the estimation of dense depth and the recovery of detailed structures for objects and environments, especially in low-contrast regions where events may not be triggered. Previous works in event-based depth estimation, such as EMVS [41], **Ref**. [138, 78], **have shown impressive performance in sparse or semi-dense mapping, while event- based dense mapping using non-learning methods remains a research gap. Learning-based approaches for event-based dense mapping are present in** 1

[52, 130, 140, 141, 142, 203],

which rely on extensive **training data and provide no guarantees of optimality or generality. Additionally, these approaches only produce local depth maps and are incapable of reconstructing a globally consistent depth map. How to effectively exploit the sparse spatial information and abundant temporal cues from asynchronous events to generate dense depth maps remains an unsolved problem. This motivates us to** in- vestigate **the complementarity between events and images, exploring the event-based dense mapping approach for inferring dense depth from sparse and incomplete event measurements** 1

. In this chapter, we design an event-based dense mapping method (the mapping module of EVI-SAM [1]) that can generate dense and texture depth maps from the monocular event camera. The event-based dense mapping framework initially recon- structs

the event-based semi-dense depth using a space-sweep way through the 6-DoF 1

6.2. Related Works

pose obtained from the tracking module. Subsequently, it integrates an aligned in- tensity **image as guidance to reconstruct the event-based dense depth and render the texture of the map (i.e., the color of 3D points). Finally, the TSDF-based map fusion is designed to generate a 3D global consistent texture map and surface mesh of the** 1

en- vironment.

To the best of our knowledge, this is the first framework that employs a non-learning approach to achieve event-based dense and textured 3D reconstruction without GPU acceleration 1

.

Our contributions can be summarized as follows: 1. **We propose** 93

an event-based dense mapping method. It involves segmenting the event-based semi-dense depth map with image guidance and introduces a novel interpolation method to recover dense depth within each segment. Additionally, we also leverage image measurements to render texture to the map (i.e., the color of 3D points). 2. **We** design **a TSDF-based map fusion method to** integrate **local event-based depth, producing a precise and globally consistent 3D map that includes appearance and surface mesh**. 3. **Extensive experimental evaluations, both qualitative and quantitative**, demon- strate **the accuracy, robustness**, generalization **ability, and outstanding** perfor- mance **of our** event dense mapping **approach** 1

. 6.2 Related Works 6.2.

1 Monocular Depth Estimation Ref. [38] introduces **the pioneering concept of purely event-based depth** estimation **by employing three decoupled probabilistic filters. EMVS** [41] **is the first work to achieve semi-dense 3D** reconstruction **from a single event camera with a known trajectory** with- out **requiring any explicit data association or intensity estimation. The concept of event- based denser mapping** is **introduced in Ref**. [122], **building upon EMVS** [41]. **However, it is still in the conceptual stage and does not successfully reconstruct any event-based dense point cloud. EOMVS** [123] **adopts** an **omnidirectional event camera in EMVS** [41] **to** reconstruct **a wider field-of-view semi-dense** depth. **Ref**. [47] **uses contrast** maxi- mization **to find the best depth value that fits the event stream. These methods recover semi-dense 3D** reconstructions **of scenes by integrating events from a moving** cam- era **over a time interval, and they require knowledge of camera motion. Ref**. 1

[124] **calculates semi-dense depth from optical flow using neuromorphic hardware to** pro- cess **asynchronous events. However, its applicability is confined to translational** mo- tion **only. Besides, various learning-based methods have gained popularity in tackling the monocular event depth estimation problem, such as convolutional neural network (CNN)** [52, 125, 126], **recurrent neural networks (RNN)** [127], **and Neural Radiance Field (NeRF**

) [130, 131, 132, 134].

However, the exploration of non-learning approaches for the **monocular event camera to recover dense and textured 3D structures remains an uncharted research area**. 6.2.2 **Stereo Depth Estimation Ref**. [137] **follows a paradigm of event matching plus triangulation to realize the 3D reconstruction. Ref**. [138] **and ESVO** [59] tackle **a semi-dense reconstruction problem using a pair of temporally-synchronized event cameras in stereo configuration through energy minimization methods. T-ESVO** [139] **extends the ESVO** [59] **using TSDF to** re- construct **3D environments and re**-estimates **the semi-dense depth**

. Ref. [79] and [78]

extend the EMVS [41] **into** a **stereo setup to estimate depth by fusing back-projected ray densities. Meanwhile, learning-based methods, such as Refs**. [140, 141, 142], **have been applied to stereo event-based depth estimation, where different deep networks are developed to reconstruct event-based dense maps. Although these learning-based methods can predict dense depth and reveal 3D structures with limited events, they may struggle to handle objects that** are **not included in** the **training sets, thus leading to uncertainties in their generalization ability**. Furthermore, **these learning-based** meth- ods **only generate localized depth maps. The production of globally consistent depth maps with rich texture information, such as surface mesh, remains an unexplored** ter- ritory **in the field of event-based vision**

. 6.3 Framework

Overview The proposed EVI-SAM utilizes inputs from the **monocular event camera, including events, images, and IMU, to** simultaneously **estimate the 6-DoF pose and reconstruct**

6.4. Purely Event-based Semi-dense Mapping

the 3D dense maps of the environment. The framework **overview of EVI-SAM is** il- lustrated **in Fig**. 6.2. **Our EVI-SAM consists of tracking and mapping modules, which operate in two parallel threads. The tracking module**

has already been introduced in Chapter 4, while this chapter only focuses on the introduction of the mapping module.

The proposed event-based dense mapping module consists of three steps: (i) com- puting **semi-dense depth** maps **from event streams**

(Section

6.4); (

ii) reconstructing dense depth maps **from the event-based semi-dense depth with** images **as guidance (Section** 1

6.5); (

iii) fusing the estimated local depths **into** the **global 3D map using TSDF-based fusion** 1

(Section 6.6). Image-based

Event-based Mapping Thread Segmentation **Event** Stream Event-**based** Event-based **TSDF-based** 1

Semi-dense Depth Dense Depth Dense Depth Map Fusion Recovery

Image Frame Direct Event-based Tracking Event-based 2D-2D alignment Relative Pose Residual Event-corner Feature Tracking 1

Event-based Triangulation Re-projection Residual IMU Data Feature Detection and Tracking Image-based Re-projection Residual Sliding Windows IMU Pre-integration Initialized? Y IMU Residual Graph Optimization Event-based Tracking Thread 3D Map 6 DoF Pose Figure 6.

2: System overview. The EVI-SAM algorithm takes events, images, and IMU as inputs, enabling the recovery of both camera pose and dense map of the scene. The mapping process takes raw event streams as input, using images for guidance, and produces dense and textured 3D mapping as output. The tracking thread 1

(has ben introduced in Chapter 4)

takes event, image, and IMU as input, and constructs the feature-based and direct-based constraints to estimate the 6-DoF pose 1

. 6.4

Purely Event-based Semi-dense Mapping We develop **the event-based space-sweep algorithm used in** [41] **to perform semi-dense depth estimation using** 1

the

monocular event camera, which targets a multi-view-stereo (MVS) problem. It consists of two steps: building a disparity space image (DSI) using a space-sweep **method** [209] **from different reference viewpoints (RV), and then** identi- fying **the local maxima of the DSI to determine the depth value** 1

of the pixel. A new RV would be chosen when the motion of the event camera surpasses a specified threshold. Subsequently, all events between two consecutive RVs are back- projected to the front RV. The creation of an RV triggers the construction of an associ- ated local depth map from that viewpoint. The back-projection rays from these nearby viewpoints are the perspective projection rays of 2D events into the 3D space (DSI) based on the camera model. The DSI, shown in Figs. 6.3 and 6.5(b), is a 3D voxel grid constructed for each RV. It describes the distribution of event back-projected rays and the scores stored in each voxel (i.e., the number of back-projected rays passing through each voxel from nearby viewpoints). The DSI is defined by N depth planes 1

{Zi}Ni=−11 with

each depth plane discretized into w × h × N cuboid voxels, where w and h are the width and height of the event camera, respectively. N is equal to 100 in our imple- mentation. The back-projecting events to the DSI can be discretized to the execution of mapping events to all depth planes 1

. We can warp an event point (

ui, vi) from the cur- rent event camera to the canonical plane Z = Z0 of RV using the planar homography HZ0 : (ui, vi, 1)T → (x(Z0), y(Z0), 1)T 1

: HZ−01 = Rrcuvr + Z0 trcvureT3 1 (6.1) where e3 = (0, 0, 1)T; Rcruvr and tcruvr are

the rotation matrix and translation vector from the RV to the current event camera 1

, respectively: Trcvur = ⎡ Rrcuvr tcrvur 0 1 ⎤ = (Twrv · Tcwur)−1 = ((Trwv) −1 · Tcwur)−1 (6.2) ⎣ ⎦ where Tcwur and Trwv

are the pose of the event camera in the current timestamp and the pose of the RV related to the world frame from the tracking module of EVI-SAM, re- spectively. After that, events in the other plane of DSI are transformed with the depth from Z = Z0 to Z 1

= Zi (shown in Fig. 6.3) using the following equation: HZiHZ−01 = Rcruvr + tcrvureT3 1 −1 1 (6.3) (Z i) (Rcruvr + tcrvureT3

Z 0) The optical center of the event camera in the coordinate frame of the RV can be defined as: X = (Xc, Yc, Zc) = − (Rcruvr)T 1

· tcrvur (6.4) 6.5. Image-guided Event-based Dense Mapping Depth Plane Event Point Reference View Current Event O C Viewpoint 1 Camera Trajectory Viewpoint 2 Viewpoint 3 Figure 6.3:

The model of event-based back-projection and space-sweep calculations across different depth planes of the DSI 1

. Then, we can calculate the warped point (x(Zi), y(Zi)) in the canonical plane Z = Zi from the point (ui, vi) in the current event camera as follows: x(Zi) = ZZ0i((ZZ0i −− ZZ0c)) x(Z0) + (1 − 1 Zi ((ZZ0i −− ZZ0c)))Xc y(Zi) = ZZ0i((ZZ0i −− ZZ0c)) y(Z0) + Zi (1 − ((ZZ0i −− ZZ0c)))Yc 1 (6.5)

After back-projecting events to DSI, we count the number of back-projection rays that pass through each voxel and determine whether or not a 3D point exists in each DSI **voxel. Based on the theory that the regions where multiple back-projection rays nearly intersect are likely to be a 3D point in the scene, the 3D points can be determined when the scores of the DSI voxels are at a local maximum. The** aggregation **of all these 3D points onto the image plane constitutes the semi-dense depth**. 6.5 **Image-guided Event-based Dense Mapping Events are sparse and mostly respond to moving edges, so the event-based semi-dense depth is also incomplete, only capturing depths at edges. Inspired by Ref**. [210], **it is possible to reconstruct the geometry of an unknown environment using sparse and incomplete depth measurements. Therefore, in this section, we leverage the** comple-mentary **strengths of event-based and standard frame-based cameras by** incorporating **additional intensity images as guidance for event-based dense depth completion. It builds on the theories that depth discontinuities are commonly correlated with** inten- sity **boundaries, whereas homogeneous intensity regions** correspond **to homogeneous depth parts** [211, 212]. **Moreover, pixels sharing the same intensity values are likely to belong to the same block in the depth image** [213]. **To this end, we can interpolate or fill the hole of the event-based semi-dense map based on the surrounding sparse depth information and the edges extracted from** in- tensity **images. We first select the intensity image captured at the RV (Fig. 6**.5(**g**)**) to ensure proper alignment with the event-based semi-dense depth. Next, we employ the region-growing approach to segment the image, as depicted in Fig. 6**.5(**h**)**. Other** dif- ferent **segmentation or edge extraction operators** can **also be used, though satisfactory results have already been obtained using such** a **simple selection. After that, all the semi-dense depth points at the current RV are projected onto the segmentation regions. Since events are only generated at pixels where brightness changes occur, they** natu- rally **align with contours. Finally, the vacant depth region can be filled or inpainted by using the weighted sum of the depth-recovered projected event points within the corresponding segmentation region. Depth**

Hole Region $\boldsymbol{\Omega}$ Known-depth Region $\boldsymbol{B\varepsilon\ p}$ with size $\boldsymbol{\varepsilon\ p}$. $\boldsymbol{N}$ Boundary of The Hole $\boldsymbol{\partial\Omega}$ Event-based Semi-dense Depth $\boldsymbol{\psi}$

The Edge in Depth The Edge in Image

Intensity Image $\boldsymbol{I}$ Figure 6.4:

The model of our event-based dense mapping incorporates **edges derived from the intensity image as guidance. The upper layer represents the event-based semi- dense depth. This layer includes areas where the depth is known (regions** success- fully **recovered through semi-dense mapping, marked in red) and areas with unknown depth (marked in black). The lower layer represents the intensity image with** bound- ary **information after segmentation. Since events are triggered in regions with edges, the semi-dense depth and the intensity image edges at the corresponding locations are consistent**

. 6.5. Image-guided Event-based Dense Mapping

We extend the fast marching method [212] to incorporate the intensity image as guidance during the filling or inpainting process for holes in the event-based semi- dense depth, ultimately producing the event-based dense depth. We firstly estimate the depth of a pixel located on the boundary of unknown regions, as illustrated in Fig. 6.4, in which the depth 1

value of point p is unknown situated on the boundary ∂ Ω of the depth-hole region in the event-based semi-dense depth ψ. Considering a Ω small neighborhood region Bε with size ε around p, where the depth values are known 1

.

Assuming the depth values have similarities in the local region, the inpainting value of p can be determined by the values of the known-depth region Bε. If ε is small enough, we can consider the first order approximation d(p, q), q ∈ Bε, the contribution value of known-depth points q to unknown-depth point p can be denoted as: d(p, q) = d(q) + ▽d(q)(p − q) ≈ d(q) (6.6) where the q is the points with known depth values of region Bε; d(q 1

)

is the depth value of point q, and ▽d(q) indicates the depth gradient at the pixel q. Next, the filled value of point p can be calculated by the weighted sum of all the contribution values of points q in the Bε(p), as follows: d(p) = d(p, ω) = ∑q∈Bε(p) ω p,qd(p, q) ∑ q∈Bε(p) ωp,q ∑q∈Bε(p) ωp,qd(q) (6.7) = ∑ q∈Bε(p) ωp,q ωp,q 1

= |ωdir · ωdst · ωlev · ωimg| (6.8) where ωp,

q is the weighting coefficient that measures the similarity of depth value be- tween the point p and q. The term of ωdir ensures that pixels closer to the normal direction N(p) have higher contributions, as defined 1

below: p − q ωdir = ||p − q|| · N(p) (6.9) The weight ωdst in Eq. (6.8)

determines the contribution of pixels based on geomet- ric distances to p. ωdst = ||p − q||2 1 (6.10) This term is directly associated with the continuity of depth, and we have empirically observed that its contribution is larger than that of other terms. The term of ωlev in Eq. (6.8) represents the level set term, assigning a higher weight to pixels closer to the contour through p, as follows: ωlev = 1 + |T(p) − T(q)| 1 (6.11) where T(p) is the distance of point p to the initial boundary of depth-hole. ▽T(p) = N(p) denotes the normal of hole boundary at the point p. The missing data is repaired pixel-wisely and the pixel with smaller T(p) has a higher inpainting priority. Lastly, the weight ωimg in Eq. (6.8) ensures that pixels with similar pixel intensity to I(p) contribute more than others: ωimg = e− ||I(p)−I(q)||2 2 (6.12) I(p) and I(q) indicates the intensity value of the image at point p and q. This term allows the weighting function in Eq. (6.8) to integrate intensity information for dense depth recovery. Pixels with similar intensity values are more likely to be part 1

of the same block in the depth image, resulting in a higher weight. To fill the whole region and generate the event-based dense depth, after recov- Ω ering the depth of pixels on the boundary ∂Ω of the depth hole Ω, we propagate the depth from ∂Ω to through iteratively applying Eq. (6.7). Firstly, we set T(p) = 0 Ω for pixels in known regions and progressively generate the distance map T(p) while marching into Ω, ensuring || ▽ T(p)|| = 1. To remove the outliers and preserve the sharpness of depth discontinuities, we employ the bilateral and non-local means filter on the event-based dense depth (Fig. 6.5(i)). Finally, we render the texture information from the intensity image onto the event-based 3D point cloud to obtain a textured map (Fig. 6.5(j)). 6.6. TSDF-based Map Fusion

(

a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l) Figure 47

6.5:

The event-based semi-dense and dense mapping of our EVI-SAM. (a) The raw event stream and (g) the intensity image from the event camera; (b) The disparity space image (DSI) of the reference view (RV) point; (c) The purely event-based semi- dense depth generated from our EVI-SAM; (d) The point cloud of the event-based semi- dense depth; (e) The occupied node of the semi-dense mapping after TSDF-fusion; (f) The surface mesh of the semi-dense mapping; (h) The segmentation on the image; (i) The event-based dense depth generated from our EVI-SAM; (j) The point cloud of the event-based dense depth with texture information; (k) The occupied node of the dense mapping after TSDF-fusion; (l) The surface mesh of the dense mapping 1

; 6.6 TSDF-based Map Fusion

TSDF can construct the global environmental surface representation based on consec- utive depth maps and associated camera poses. To merge the event-based local depth into a TSDF, we perform the ray-cast from the origin of the event camera to each depth point in the event-based local depth map. Meanwhile, we update the signed distance and weight values of TSDF voxels along this ray. The merging of local depths is based on the current distance Dlast and the weight value Wlast of a TSDF voxel, as well as the new update values from a specific point observation in the event-based local depth map. Given a center position of TSDF voxel (V ∈ R3) is passed by a ray between a 3D depth point (P ∈ R3), and the event camera origin (O ∈ R3). The updated TSDF distance value Dnew and the weight value Wnew for this voxel are described as follows 1

: Dnew = WlasWt·laDstla+st w+ wd , (6.13) Wnew = min(Wlast + w, Wmax),

where d is the distance from the TSDF voxel center V to the new coming 3D depth point P, defined as follows: d = ∕∕P − V ∕∕ sign[(P − V) · (P − O)] (6.14) While for the weight w of this new coming 3D depth point, we follow Ref. [214] to define it with the 1

.

−dt |

where ρ is the depth value of the event-based local depth in the event frame; dt = 4ε is the truncated distance for TSDF, and ε represents the size of voxel (ε = 0.01 in our implementation). Following the creation and update of TSDF voxels, the global depth map will be refined based on Eq. (6.13). **For each depth point in the event-based local depth maps, we project its position onto the TSDF voxel grid and group it with all other depth points** that map **to** the same **voxel. Subsequently, we compute the weighted mean of all points within each TSDF voxel and perform** the ray-cast **only once on this mean position. The surface meshes can also be reconstructed from updated TSDF voxels (shown in Figs. 6.5(f) and 6.5(l)), which allow us to get a better assessment of the perception** 1

envi- ronment. 6.7 Experiments

This section comprehensively evaluates the mapping **performance of our EVI-SAM system** 1

,

employing both quantitative and qualitative assessments through extensive experiments 1

.

We assess the mapping performance of EVI-SAM through four sets of experiments. (i) **The first set** 1

(Section 6.7.1)

evaluates the mapping performance across diverse scenarios to show the generation ability of our event-based dense mapping method 1

. (ii) The second set of experiments (Section 6.7.2)

evaluates the overall system on the 6.7. Experiments **onboard platform, including comparing local mapping with a commercial depth** cam- era **and assessing the global event-based 3D dense reconstruction performance**. (iii) **The** 1

Third set compares our 3D mapping approach with different mapping baselines, including those relying on image-based (Section 6.7.3.1), event-based (Section 6.7.3.2), learning-based (Section 6.7.3.3), and NeRF-based (Section 6.7.3.4) methods. (iv) Ad- ditionally, Section 6.7.4

evaluates our mapping performance **under challenging** situa- tions, **such as HDR** 1

, and the scenarios involving aggressive motion.

Finally, the analysis of the computational performance and the discussion of the limitations are completed in Section 1

6.7.5 and Section 6.7.6, respectively. 6.7.1

Mapping Performance in Diverse Scenarios In this section, we evaluate the mapping performance of our EVI-SAM across diverse platforms and scenarios, **including** handheld **devices (such as** DAVIS240C [144], **rpg** [138], **HKU-dataset), flying drones (MVSEC** [147]), **and driving vehicles (DSEC** [153]). **These datasets provide event streams, grayscale frames, and IMU data. The event stream has a resolution of 240 × 180 pixels for rpg** [138] **and DAVIS240C** [144], **346 × 260** pix- els **for DAVIS346 in MVSEC** [147] **and HKU-dataset, 640 × 480 pixels for DSEC** [153]. **Additionally, these datasets span a wide range of scenarios, from indoor to outdoor** au- tonomous **driving, featuring scenes with varying texture richness, ranging from well-lit to dimly lit** conditions, **and encompassing both dynamic and static objects. We** evalu- ate **the full EVI-SAM system utilizing the event-based hybrid tracking module** 1

(details in Chapter 4)

to provide pose feedback and demonstrate the generalization capability of our event-based dense mapping. As some sequences lack ground truth depth, we only present the **qualitative** map- ping **results in Fig**. 6.6. **The mapping results** are **evident that our method does a** no- table **job of reconstructing 3D dense structures with texture information** in **such** di- verse **scenarios, thus** demonstrating **its excellence and generalization ability in real-world scenes**. This highlights **the** 1

robustness and powerful generalization capability of our algorithm, distinguishing it from learning-based approaches 1

[134, 125, 126, 127, 52, 140, 203, 142]

that typically assess only a limited number of sequences within the same scene 1

. The key distinction lies in their usual requirement for scene-specific pre-training or even re-designing when applied across various scenes. Additionally, as evident in (a) poster_translation (b) indoor_flying_3 (c) HKU_monitor (d) HKU_reader (e) HKU_hdr_box (f) HKU_reader_1 (g) HKU_desk (h)

dsec_zurich_city_04_a (i) **dsec_zurich_city_04_e** (j) **dsec_zurich_city_04_f** Figure 6.6: **Qualitative mapping result of our EVI-SAM in various data sequences. Each element includes the image view, dense depth, and the point cloud with texture** infor- mation. **The depth is pseudo-colored, from red (close) to blue (far), in the range of 0.55- 6.25 m for the rpg** [138] **and davis240c** [144]; **in the range 1.0-6.5 m for MVSEC** [147]; **in the range 0.5-4.0 m for HKU-dataset; in the range** 1

4.0-200 m for DSEC [153], respec- tively. Figs. 6.6(h) - 6.6(j),

the geometry of 3D scenes and the textures of buildings or vehicles have been effectively reconstructed. Notably, in Fig. 6.6(i), **the reconstruction of the crosswalk on the road is particularly well-executed. In the dense depth, the black areas indicate instances where depth recovery was** 1

unsuccessful. For instance, the sky areas in the illustrations correspond to regions that are too far for depth recovery. Addition- ally, **certain textureless regions of the road, lacking sufficient event data, may also pose challenges for recovering dense depth. However, it is essential to recognize that such occurrences are** caused by **inherent characteristics of event cameras. These situations would not occur** in indoor environments. **Our event-based dense mapping effectively recovers the complete dense depth along with the details of the structure. For instance, as illustrated in Fig**. 6.6(a), **our method adeptly reconstructs the texture of the poster, and the** consistency **of the wall depth is excellent. On the flat wall surface, most areas in our event-based dense map display similar colors, representing proximity and uniform distances across those regions** 1

. 6.7.2 Real-time Onboard Mapping

Evaluation Although the data sequence utilized in the previous section is appropriate for assess- ing **the performance of pose tracking and mapping in challenging situations, it does not encompass real-time onboard evaluation and global mapping. Moreover, we lack a dataset for comparing our event-based dense mapping with the depth camera. To fill this gap, we design an event-based handheld device and evaluate the onboard** perfor- mance **of our EVI-SAM. Please note that the depth maps provided by the depth camera only serve as references for qualitative comparison** 1

. (A) (B) Aprilgrid 4s Battery Screen (C) (D) DAVIS 346 Intel D455 NUC Figure 6.7:

The event-based handheld device with the schematics **model for onboard evaluation. Please note that the RGB-D camera (Intel D455) is only used for reference, the complete system of our EVI-SAM operates exclusively with the monocular event camera (DAVIS346** 1

).

Our handheld device is shown in Fig. 6.7, **which includes a power supply unit, an onboard computer NUC (equipped with Intel i7-1260P, 32GB RAM, and Ubuntu 20.04 operation system), a DAVIS346 event camera, and** an **Intel® RealSense™ D455 RGB-D camera. All mechanical modules of this device are designed for 3D printing, and both design schematics and the collected data sequence are available on our GitHub** reposi- tory: **https://github.com/arclab-hku/Event_based_VO-VIO-SLAM** 1

/blob/main/EVI-SAM/ data_srcipt.md (

a) Depth from RGB-D camera (b) Depth from our EVI-SAM Figure 6.8: **Comparing the local texture depth generated by EVI-SAM with the one produced by the RGB-D camera** 1

. 6.7.2.1

Local Mapping Performance We conduct real-time testing of EVI-SAM using the handheld device within the LG01_lab of the Haking Wong Building at The University of Hong Kong. The results, including event-based dense mapping at selected viewpoints and the estimated trajectory, **are** il- lustrated **in Fig**. 6.9. **Three live demonstrations are also available** at **the** project website. **Additionally, we** conduct **qualitative comparisons between the dense mapping results from our EVI-SAM and the depth images** 1

from the RGB-D camera. To ensure a fair com- parison, **we render the texture information from the intensity image of DAVIS346 onto the depth image of the RGB-D camera, maintaining the same resolution as** DAVIS346. **The depth images are directly acquired from the RGB-D camera, with its infrared** rang- ing **sensor obscured. We directly align the texture point** clouds **from the RGB-D camera along with the estimated pose of our tracking module. As observed in the highlighted area within the red circle in Fig** 1

. 6.8, it is evident that

our proposed event-based dense mapping approach demonstrates performance comparable to commercial depth 1

cam- eras. The reconstructed depths from the RGB-D camera are notably noisy and clut- tered, lagging far behind the results obtained from our EVI-SAM.

Moreover, it exhibits superior noise reduction and texture recovery capabilities. More dense **mapping** 1

per- formance comparisons between our EVI-SAM and the RGB-D camera can be found in the Section 6.7.3.1. 6.7.2.2

Global Mapping Performance The global mapping performance of our EVI-SAM is also illustrated in Fig 1

. 6.9, show- ing

the surface mesh generated through TSDF-based map fusion. Our global event- based dense mapping exhibits excellent global consistency 1

. The video demo in Sec- tion 6.7.2.1 also illustrates the incremental

reconstruction of the event-based surface mesh from updated TSDF voxels. This process enables on-demand meshing for visual- ization, **allowing flexibility in generating the mesh at any time. Our TSDF-based map fusion for global mapping is designed to generate surface meshes that** 1

enable humans to assess the 3D reconstructed environment more effectively. This capability supports high-level mission goals, such as collision-free motion planning. During evaluation, an onboard computer NUC is utilized to support real-time pose estimation and local event-based dense mapping. However, the NUC lacks sufficient computational power to support a real-time meshing process. Therefore, we utilize a

personal computer (In- tel

i7-11800H, 32GB RAM) without GPU to output the global mesh of EVI-SAM for **the** global mapping evaluation. **Further details regarding the time-consuming aspects of different modules will be discussed in Section** 1

6.7.5. Figure 6.9:

Visualization of the estimated camera trajectory and global 3D reconstruc- tion (**surface mesh) of our EVI-SAM. Sequentially display from right to left includes the event-based dense point** clouds **with texture information and intensity** images, **at selected viewpoints** 1

. 6.7.3 Mapping Performance Comparison with Baselines 6.7.3.1 Comparison with Traditional Image-based Mapping We conduct qualitative comparisons of the depth estimation results presented as tex- ture point clouds obtained from our EVI-SAM, with the depth estimation results from stereo RGB camera, monocular RGB camera, and RGB-D camera serving as the base- line. We used the data sequence "LG_office", which features fast motion.

To ensure a fair comparison, firstly, **we** only **employ** lightweight **and** 108

real-time methods to estimate depth from standard cameras, as opposed to more complex approaches that rely on GPU acceleration. Since our EVI-SAM operates in real-time solely on the CPU. Sec- ondly, we resize the resolution of the depth generated from the stereo RGB camera, monocular RGB camera, and RGB-D camera to match the resolution of our event cam- era (346 × 260). Thirdly,

we render the texture information from the intensity image of the **DAVIS346 onto the** estimated **depth** to generate **the** 1

textured point cloud. The textured point clouds of these baseline methods are obtained as follows: • RGB-D camera. It is directly obtained from the depth image of Realsense D455 [215], with its infrared ranging sensor obscured. While

the texture information is trans- formed **from the** intensity **image** of **the** 1

DAVIS346. • Stereo RGB cameras. It is generated by the image_undistort tool [216]. • Monocular RGB camera. It is generated by using the monocular dense mapping algorithm in Ref. [213] using GPU acceleration for real-time running. • All of these baseline methods use the monocular image+IMU (VINS-MONO [8]) to provide pose estimation. As can be seen from Fig. 6.10, our EVI-SAM

performs better than these baselines in terms of 1

both the accuracy of depth recovery and the integrity of texture information. The textured point cloud obtained from the RGB-D camera is directly captured from a commercial Realsense camera. However, its infrared ranging sensor is obstructed, as infrared light can interfere with the perception of event cameras. When the infrared ranging sensor is blocked, the depth map produced by the Realsense camera tends to contain a considerable amount of noise, as illustrated in column 4 of Fig. 6.10. This observation matches our expectations, as demonstrated in Section 6.7.2.1, where it is Figure 6.10: Local mapping performance

comparison. The first column shows the in- tensity image of the selected view. Column 2 to 5 show the color point cloud results from our EVI-SAM, stereo RGB camera [216], RGB-D camera [215], and monocular camera [213], respectively.

evident that the depth map generated by the RGB-D camera exhibits more noise com- pared to **that of our EVI-SAM** 1

. Regarding the results from monocular and stereo RGB cameras, as shown in columns 3 and 5 of Fig. 6.10, although their estimated depths exhibit less noise compared to those estimated by RGB-D cameras, the completeness of depth recovery falls far be- hind that of RGB-D and our EVI-SAM. This might be caused by the challenging testing conditions characterized by varying illumination and fast motion. Additionally, the regions with limited texture and the drawbacks, such as the narrow dynamic range of image sensor perception, lead to the poor depth recovery performance of monocular and stereo RGB cameras on this data sequence. While our EVI-SAM can still provide reliable depth perception under such challenging situations. 6.7.3.2 Comparison with Event-based Mapping

In this section, we conduct comparisons with other event-based depth estimation ap- proaches using sequences from MVSEC [147] **and rpg** [138] datasets. **Fig**. 6.11 **shows inverse depth maps produced by our EVI-SAM and baseline methods. The first row displays raw** images **of the selected reference** views. **The second to the fourth row show inverse depth maps generated by GTS** [217, 59], **SGM** [218, 59], **and ESVO** [59], respectively. **The last three rows present the semi-dense, dense, and textured** depths **generated by our EVI-SAM. As anticipated, since event cameras solely respond to the apparent motion of edges, the methods that produce semi-dense depth maps only** de- pict **edges of the 3D scene. In contrast, our EVI-SAM can recover texture and surface information to obtain dense depth maps. The results of GTS** [217, 59], **SGM** [218, 59], **and ESVO** [59] **are borrowed from the respective papers. For comparison, we select the reference views that are similar to** those **used in ESVO** [59]. **It is worth noting that, unlike our method, the baseline methods exclusively rely on known camera poses** pro- vided **by an external motion capture system to run the mapping process. In contrast, our mapping approach relies on the pose feedback provided by our event-based hybrid tracking modules to achieve depth estimation. Additionally, our EVI-SAM utilizes** the **monocular event camera, while the baseline methods employ stereo event cameras. However, it** can **still be observed that our event-based mapping method gives the best results in terms of the overall depth recovery performance and** the **density** 1

percentage.

rpg_reader rpg_box rpg_monitor rpg_bin upenn_flying1 upenn_flying3 1

Scene GTS SGM ESVO Our T exture Our Dense Our Semi- dense Depth Depth Depth Figure 6.11:

Mapping result comparison. The first row shows the intensity frames from the event camera. Rows 2 to 4 show semi-depth estimation results from GTS [217, 59], **SGM** [218, 59], **and ESVO** [59], **respectively. The last three rows show the estimated semi-dense depth, dense depth, and texture point cloud from our method. Depth maps are pseudo-colored, from red (close) to blue (far), in the range of 0.55-6.25 m for the rpg** [138] **and in the range 1.0-6.5 m for MVSEC** 1

[147]. Table 6.1

details the depth errors associated with various event-based depth esti- mation **methods in MVSEC** [147] **flying drone sequences. We utilize mean depth error as our evaluation metric, computed by comparing estimated depth values with the ground truth depth obtained through 3D LiDAR. For the semi-dense methods, we** val- idate **them using their open-source code and known camera poses provided by the dataset. We choose estimated depths from each baseline that closely match the same ground truth depth for assessment. To ensure a fair comparison** 1

with the dense ground truth depth,

we assign zero values to regions where depths fail to recover and calcu- late **the density percentage of the semi-dense depth estimates** 1

. As can be seen from Table 6.1,

our method performs better than these baselines in terms of dense depth re- covery. **To our knowledge, this is the first successful attempt to compute monocular dense depth results for event cameras without using learning-based methods. Note that the reported results of these event-based semi-dense methods are inferior to those presented in** the **original papers. This discrepancy is attributed to the usage of ground truth dense depth rather than ground truth sparse depth. Meanwhile, it should be noted that the ground truth dense depth provided by MVSEC may not be sufficiently accurate. Firstly, the low frequency of the ground truth depth poses a challenge to time alignment, particularly during camera motion. Secondly, errors in the ground truth depth could be introduced during the generation process, stemming from the** odom- etry **procedure. This may cause objects in the ground truth depth to appear inflated compared to their original versions in the image/event space. These factors constrain the ability to achieve accurate validation of mapping performance. Table** 6.1: **Quantitative comparison of mapping** performance **in terms of mean depth error (m) and the density percentage of the recovery (%), which is assessed against the dense ground truth depth. Methods** Types **indoor_flying1 indoor_flying2 indoor_flying3 Mean Error(m) / Density Percentage (%) EMVS** 1

[41] Ref. [78] ESVO [59] Ref. [203] Ours Mono Stereo Stereo Event-only stereo Image-only stereo Event-Image stereo Mono 2.38 / 6.01% 1.82 / 4.90% 2.82 / 4.93% 2.46 / 1.79% 1.85 / 1.35% 2.88 / 1.85% 2.43 / 3.29% 1.80 / 4.00% 2.88 / 2.34% failed failed 0.26 / 100% failed failed 0.24 / 100% failed failed 0.22 / 100% 0.78 / 100% 0.82 / 100% 0.86 / 100% 6.7.3.3 Comparison with Learning-based Mapping Furthermore, Table 6.1 also provides the depth error of a learning-based method [203]. Since we followed the same evaluation protocol as Ref. [203], we can directly report the raw results from the original paper. Despite this method being numerically superior to the other non-learning methods, it requires scene-specific pre-training for each data sequence and high-level hardware resources such as GPU. It is worth noting that when evaluating the indoor_flying3, indoor_flying1 and indoor_flying2 sequences are used to train the network. Additionally, only around 1000 frames from the entire rosbag, containing over 3000 frames, can be used for testing. In contrast, our EVI-SAM can process the entire rosbag of indoor_flying1 to indoor_flying3 without any pre-training. Therefore, we only report the raw result of indoor_flying3 from Ref. [203], and label the results of indoor_flying1 and indoor_flying2 as failed since they only work under specified pre-training conditions. (a) View1 (b) View2 Figure 6.12: Qualitative comparison of our EVI-SAM and the learning-based method [203] in MVSEC [147] dataset. For each element, the first row is the (i) im- age view, and the estimated depth of learning-based method, which utilizes the (ii) event-only, the (iii) image-only, and the (iv) event-image methods; The second row is the (v)

ground truth depth obtained through 3D LiDAR, the 1

estimated depth of our EVI-SAM using (vi) event-only and (vii) event-image, and our estimated depth with texture information. As shown in Fig. 6.12, we further demonstrate the qualitative comparisons with this learning-based approach using the sequences of indoor_flying3 from the MVSEC dataset. Since the source code and pre-training model of Ref. [203] are not released, the estimated depth results are borrowed from the original paper. While

the ground truth depth **and** our **estimated depth** 94

are pseudo-colored, from red (close) to blue (far 1

).

For comparison, we select the reference views that are similar to those **in** Ref. [203]. **It is worth noting that** 1

this learning-based approach employs a complex network with many structures, such as recycling, deformable, and multiscale architecture. Although its numerical results in Table 6.1 may appear superior to our EVI-SAM, a thorough examination of the qualitative results indicates otherwise. There are instances of sig- nificant noise in their estimated results, particularly when employing stereo images or stereo event setups. Conversely, our EVI-SAM not only demonstrates superior map- ping accuracy (both quantitatively and qualitatively) but also can operate without any scene-specific pre-training. Additionally, our EVI-SAM consistently achieves good mapping results across di- verse and challenging scenarios, as evidenced in Sections 6.7.1 and 6.7.4. In particular, thorough the evaluations in Section 6.7.4.2,

our event-based dense mapping reliably **reconstructs the 3D structure of** scenes **even** 1

under fast rotation (up to 290◦/s) and translation (up to 18 m2/s) motions. While most of the event-based mapping works, especially for those relying on learning-based methods, often neglect to assess their performance under aggressive motion. 6.7.3.4 Comparison with NeRF-based Mapping We evaluate the NeRF-based methods using the LG_office sequence of the EVI-SAM dataset in Section 6.7.2. Neural radiance field (NeRF) is a fully-connected neural net- work that can generate novel views of complex 3D scenes, based on a partial set of 2D images. It operates by processing input images representing a scene and interpolating between them to render one complete scene. Therefore, each scene must be retrained and can only be tested in the same scene. Failure to retrain the model when the scene changes would result in entirely incorrect outputs. We employ the Instant-NGP [219] for the novel view synthesis from the image of DAVIS346 and use the COLMAP [220] to estimate the camera parameters, including both intrinsics and extrinsics (pose), for each input image. For comparing the performance

of our event-based dense mapping, we visualize **the** view synthesis **of** 1

similar viewpoints of Fig. 6.9 and Fig. 6.10. As highlighted with the red box in Fig. 6.13, significant blurring effects can be observed in the reconstruction results of the NeRF-based method, which is caused by image blurring in rapid motion. In contrast, our event-based dense mapping method effortlessly overcomes this challenge without encountering motion blur. Additionally, (a) (b) Figure 6.13: Qualitative comparison of our EVI-SAM, the Instant-NGP [219],

and the ground truth image view. **The** red box highlights **the** difference **of** 55

the reconstruction quality and the motion blur caused by image blur in NeRF-based methods. While our

event-based dense mapping demonstrates performance comparable to 1

state-of-the-art NeRF-**based** mapping. **the** 94

memory consumption of Instant-NGP is extremely high, making it impossible for our GPU (NVIDIA GeForce RTX 4070 Ti) to complete training with the entire sequence. Therefore, we had to select 8 corresponding viewpoints for 8 separate training sessions. In other words, the result of the Instan-NGP in Fig. 6.13 stems from 8 entirely distinct training sessions, whereas our EVI-SAM can operate in real-time without the need for segmentation processing or prior training. Figure 6.14: The mapping result of Co-SLAM [221] and the corresponding selected viewpoint. Furthermore, we also utilized the RGB-D data from the sequence to train two

state- of-the-art NeRF-based **RGB-D SLAM** 142

[222, 221]. However, due to significant noise in the depth camera and the lack of an infrared ranging sensor, depth information cannot be captured in low-texture scenes (i.e., the white wall in Fig. 6.13). This resulted in inadequate depth information for training both Nice-SLAM [222] and Co-SLAM [221]. Besides, the rapid and aggressive motion in the testing sequence scenes is too challeng- ing for these methods, resulting in inadequate pose estimation and ultimately leading to mapping failures. The mapping results of Co-SLAM are shown in Fig. 6.14. 6.7.4 Mapping Performance in Challenge Situations

In this section, we specifically evaluate the mapping performance of our EVI-SAM in challenging conditions 1

. 6.7.4.1 HDR Scenarios

In this experiment, we demonstrate the effectiveness of our event-based dense map- ping **in HDR scenarios using an event-based driving dataset (ECMD** [6]). **We** specifi- cally **choose three** sequences (**Dense_urban, Tunnel, and Suburban_road) that exhibit HDR scenes caused by the intense illumination change. Fig**. 6.15 **illustrates the** suc- cessful **recovery of dense depth by our event-based dense mapping under challenging conditions, including intense sunlight and HDR encountered in tunnels** 1

. Since the tex- ture

information is adopted from the image measurements, **the event-based dense point** clouds exhibit **overexposure caused by strong sunlight. While our event-based dense map would not be affected by the illumination** 1

.

Besides, as shown in Fig. 6.6(e), **our event-based dense mapping is capable of recovering dense depth even in low-light** sce- narios. **Despite these achievements**, addressing **nighttime driving scenarios proves to be a significant challenge. This challenge arises from the expansive and dark** character- istics **of nighttime driving environments. Our event-based dense point cloud derives texture information from image measurements, resulting in the point cloud that** ap- pears **pitch-black in such conditions. In conclusion, our event-based dense mapping remains effective in recovering depth under HDR scenarios. Nevertheless, it is crucial to acknowledge that HDR conditions may impact the texture of our dense point cloud. (a) Dense street (b) Tunnel (c) Suburban road** Figure 6.15: **Mapping result of EVI-SAM under HDR scenes. The first row shows the intensity frames from the event** 1

camera. The second row shows the event-based dense maps, and the last row is the event-based dense point clouds with texture information

. 6.7.4.2

Aggressive Motions In this experiment, we examine the robustness of our event-based dense mapping in scenarios involving aggressive motions (see Fig. 6.16), where the maximum angular ve- locity reaches up to 5 rad/s (approximately 290◦/s), and the linear acceleration reaches up to 18 m2/s (see the gyroscope and acceleration readings in the left part of Fig. 6.16). The results of our estimated event-based dense depths and texture point clouds are also presented alongside timestamps, with the aggressive motion phase shaded in grey blocks. It can be 1

observed that our event-based dense mapping consistently and sta- bly reconstructs the 3D structure of the scene even in the presence of intense rotation and translation motions. These results underscore the robustness of our algorithm and demonstrate its capacity to handle scenarios with aggressive motions. For visual demonstrations of our mapping results under aggressive motions, please refer to the video 1

demo. 1 4 1 2 34 6 5 2 5 3 6 Figure 6.16: Mapping results

of EVI-SAM under aggressive motions. We visualize the results of our event-based dense mapping along with the timestamps (ROS time), and the raw gyroscope and acceleration reading from the IMU. The shaded area in grey represents the phase of aggressive motions 1

. 6.7.5

Running Time Analysis In this section, we investigate the average time consumption of our proposed system on the CPU-only NUC. We calculate the average time consumption during the evalua- tion of Section 1

6.7.2,

whose results are shown in Table 6.2. The hybrid tracking module takes an average of 55.24 ms with an additional 3.75 ms for event-corner feature track- ing, and the dense mapping module takes an average of 137.45 ms processing time. The estimated pose frequency can be effortlessly boosted to IMU-rate by directly in- tegrating the latest estimation with IMU measurements in a loosely-coupled manner. Meanwhile, the dense map generation frequency is around 7 Hz, which should be real- time enough for tasks like collision-free motion planning, e.g., obstacle avoidance. Table 6.2: The average time consumption of different modules in our EVI-SAM Sequence Event-corner tracking Mean / Std (ms) Hybrid tracking Mean / Std (ms) Dense mapping Mean / Std (ms) Logo_wall_1 3.50 / 1.17 54.39 / 23.35 138.21 / 79.49 Logo_wall_2 3.75 / 1.03 55.11 / 23.01 123.8 / 49.24 LG_Factory 4.49 / 1.66 67.29 / 54.90 129.63 / 44.54 LG_office 3.89 / 1.37 55.10 / 31.88 112.09 / 40.96 Fountain_1 3.75 / 1.15 49.78 / 15.76 141.46 / 48.59 Fountain_2 1

3.11 / 1.14 49.79 / 8.57 179.49 / 106.68 Average 3.75 / 1.25 55.24 / 26.24 137.45 / 61.57

6.7.6 Limitations and Discussions

Regarding our event-based dense mapping, one limitation lies in the difficulty of attain- ing **comprehensive and precise recovery of dense depth across various environments. This is caused by two primary factors. The first factor is the** inherent **nature of event cameras, which produce sparse pixels and only respond to moving edges. The** sec- ond **factor is attributed to scenes being too distant from the camera, such as the sky in a driving scenario. Conversely, when the event camera operates in structured** envi- ronments **where depth patterns exhibit regularity, such as many planar surfaces with limited edges, or scenarios with sharp object boundaries, our method can leverage this regularity to infer good dense depth from a small number of event measurements**. Be- sides, **our event-based dense mapping is a kind of color-guided depth enhancement method. This kind of approach is under the assumption that the edges of the event- based semi-dense depth and the color edges at the corresponding locations are** con- sistent. **Therefore, the incorrect guidance from the companion color image will lead to texture-copy artifacts and blurring depth edges on the reconstructed dense depth. While various methods** [223, 224] **exist to address this issue, we intend to defer this challenge to future works. Another limitation of our event-based dense mapping is the inevitable occurrence of artifacts. The term** "artifacts" **pertains to specific errors in scale recovery, such as cases where small areas in distant regions are mistakenly reconstructed as closer ones. This** may **be attributed to the implementation of the monocular MVS problem in our approach, resulting in a loss of scale. Consequently, in future work, we may extend our algorithm to incorporate** a **stereo** setup **to address this issue**. Certainly, **image-based dense** 1

reconstruction has been thoroughly studied and they have also achieved promising results. However, the exceptional advantages of event cameras underscore the importance of exploring event-based dense reconstruction, especially considering that non-learning methods for event-based dense reconstruc- tion remain an unexplored territory. Our method generates dense depths from event streams, with images only serving as guidance. Thus, the reconstruction results may not

match the impressive **performance of** some **state-of-the-art** 134

RGB-only or RGB-D methods. Besides, our method does not rely on images to generate dense depth, instead, dense depths are generated

from the event stream with images only **serving as** guidance. Therefore, **the** 61

reconstruction results may not be as impressive as some

state- of-the-art RGB-**based methods**. Nevertheless, **our** 61

focus lies in addressing dense recon- struction using event cameras, which offer a potential solution for challenging scenar- ios not adequately addressed by standard images. This work not only aims to achieve optimal mapping performance in challenging situations but also serves as an inspi- ration for the research community encouraging future research in event-based dense reconstruction. 6.8 Conclusion In this chapter, we have presented the mapping module of EVI-SAM, a framework de- signed for 3D dense mapping using the monocular event camera.

To achieve dense mapping, an image-guided and segmentation-based approach is proposed **to** recon- struct **dense maps from sparse and incomplete event measurements. To create a** com- prehensive **representation of the 3D environment, we** have **designed** the **TSDF-based map fusion to construct both the textured global map and the surface mesh. Our framework balances accuracy and robustness against computational efficiency towards** 1

6.9. Related Publications strong

mapping performance in challenging scenarios such as HDR or fast motion. No- tably, **EVI-SAM demonstrates computational efficiency for real-time execution**, mak- ing **it suitable for onboard mapping and navigation. Meanwhile, the generated event- based dense** maps **can be directly applied to collision-free motion planning, ensuring safe navigation** 1

. 6.9 Related Publications 1. Guan Weipeng, Chen Peiyu, Zhao Huibin, Wang Yu, Lu Peng, “EVI-SAM: Robust,

Real-time, Tightly-coupled Event-Visual-Inertial State Estimation and 3D Dense Mapping”, Advanced Intelligent Systems 11

, pp. 1-24, 2024. Chapter 7 Conclusion and Prospects 7.1 Conclusion This thesis is primarily dedicated to addressing the efficiency challenges associated with using event cameras to solve the SLAM problem, which encompasses both

6-DoF camera **pose tracking and 3D dense mapping** 1

. The methodology followed in each chap- ter

aims to exploit the unique characteristics **of event cameras** 47

. To this end, Chapter 1 provides a comprehensive introduction to event cameras, highlighting their distinctive characteristics, and provides an overview

of state-of-the-art (SOTA) event-based 20

SLAM algorithms. Chapter 2 presents Mono-EIO,

a novel event inertial odometry (EIO) **framework that** 109

avoids relying on traditional image-based corner detection. Instead, it employs an asynchronous, uniformly distributed event-corner detector designed specifically for event-only data. The detected event-corner features are integrated into a sliding win- dow graph-based optimization framework, which tightly fuses these features with IMU measurements to estimate 6-DoF ego-motion. Chapter 3 introduces PL-EVIO,

an event-based visual-inertial odometry (EVIO) framework **that** 9

incorporates both point and line features. Two variants are explored: PL-EIO, which is purely event-based, and PL-EVIO, which incorporates image-aided data. It

combines the strengths of heterogeneous multi-modal visual sensors, including event cameras **and** 60

standard cameras, isomeric visual features, including point-based Chapter 7. Conclusion and Prospects and line-based features. This framework demonstrates sufficient reliability and accu- racy to provide

onboard pose feedback control for quadrotors, enabling them **to** per- form **aggressive** 2

maneuvers such as flipping. Additionally, this chapter also presents

ESVIO, the first stereo **event-based visual inertial odometry** 7

framework, including

ESIO (purely event-based) and ESVIO (event with image-aided 7

). Chapter 5 proposes DEIO, a pioneering learning-optimization-combined frame- work that integrates

trainable event-based differentiable bundle adjustment with IMU 11

pre-integration. This is achieved within a patch-based co-visibility factor graph that employs keyframe-based sliding window optimization.

To the best of our knowledge, DEIO is **the first** learning-based **event-inertial odometry** 8

framework. Remarkably, de- spite being trained on synthetic data, DEIO outperforms over 20 state-of-the-art meth- ods across 10 challenging real-world event benchmarks. Chapter 4 and 6 present EVI-SAM, a comprehensive monocular event-based SLAM system. It employs parallel threads for mapping and tracking, enabling

real-time 3D dense mapping **and 6-DoF** pose **tracking** 9

.

To the best of our knowledge, this is the first framework that employs a non-learning approach to achieve event-based dense and textured 3D reconstruction without GPU acceleration. Additionally, it is also the first hybrid approach that integrates both direct-based **and** feature-based methods **within an event-based framework** 1

. The mapping and tracking module of EVI-SAM are executed in parallel threads that perform full-SLAM, enabling

real-time 3D dense mapping **and 6 DoF** pose estimation **with** monocular **event camera**. Through detailed examination **of** 83

various methodologies tailored specifically for event cameras, including feature-based, direct-based, their combination (hybrid), and learning-based approaches, this thesis provides a comprehensive understanding of the mechanisms underlying event cameras in SLAM. In conclusion, the studies presented in this dissertation underscore

the advantages of leveraging **event cameras for** robust, high-accuracy, **and** 9

real-time

6-DoF pose tracking and 3D dense mapping 1

. We trust that

the insights and findings presented **in this thesis** will **contribute to** advancing **the field of** 55

event-based vision and SLAM, while also serving as a source of inspiration for future research endeavors in these domains. 7.2. Future Works 7.2 Future Works This section further explores potential directions for future research in event-based vi- sion, drawing on the knowledge and perspectives gained throughout my research. • Event-based multi-sensor fusion: The event-driven algorithms remain in the early stages of development, despite several studies showcasing their superior performance compared to traditional image-based solutions. Much of the current research on event cameras remains confined to synthetic data or controlled envi- ronments, which significantly limits their practical applicability in more demand- ing scenarios, such as those involving aggressive motion or HDR conditions. Ac- tive research efforts are underway

to unlock the full potential of event cameras **in** these challenging **scenarios** 104

. Despite these efforts, the field is still in its infancy. Integrating data from multiple

sensors can significantly **enhance the robustness and** adaptability **of event-based** 12

SLAM in complex environments. Moreover, it is essential to view event cameras as complementary tools rather than replacements for standard cameras. Future research may focus on

event-based multi-sensor fu- sion, incorporating **a** broader **range of local perception** systems (e.g., **LiDAR) and global perception** technologies (e.g., **visible light positioning** [225, 226] **for indoor, or GPS for outdoor** 2

), to fully leverage

the complementary strengths **of different sensors** alongside **event cameras**. • Event-based Representation **A** 2

central focus of research lies in developing inno- vative methods to represent event data in ways that leverage its asynchronous and sparse characteristics. Various event-based representations have been pro-posed, including time surfaces (TS) and their variants, event images, motion- compensated event images (e.g., contrast maximization), voxel grids, event ten- sors, and etc. While different representation is suited to various tasks, it is essen- tial to critically evaluate their strengths and limitations to guide future research directions. A crucial point to emphasize is that the defining feature of event cam- eras is their asynchronous nature, which fundamentally distinguishes them from traditional synchronous frame-based imaging systems. To fully unlock the po- tential of

event cameras, the development of new event-**based** representations **is** essential. **In** 83

practice, event-batching methods, e.g., converting asynchronous Chapter 7. Conclusion and Prospects event streams into a pseudo-image format, are often preferred over event-driven (event-by-event) algorithms. This preference arises not only from the ease of inte- grating these methods with conventional computer vision algorithms but also be- cause they provide

access to more comprehensive **information in each processing step** 32

. In contrast, event-by-event algorithms frequently rely on

internally stored states **or memory to compensate for the lack of sufficient information** 32

at each event. Furthermore, trainable event representations, such as those introduced in [170], merit further exploration, particularly in the context of self-supervised learning. These approaches could help identify optimal event representations tai- lored to a wide range of tasks in

event-based vision. • Dynamic SLAM **for Event** Cameras: Event-**based** ego-**motion estimation** 138

and mapping remain challenging tasks, particularly in dynamic environments. On one hand, most existing SLAM systems are designed under the assumption of static environments, which often results in inaccuracies and reduced performance when applied to dynamic scenes. On the other hand, event cameras are inher- ently motion-activated sensors, capturing data generated by relative motion be- tween the scene and the camera, including both ego-motion and the movement of dynamic objects. Unlike standard cameras, dynamic objects tend to contribute a significantly larger proportion of the perceived information in event cameras. Thus, developing a dynamic SLAM framework capable of handling dynamic ob- jects for event cameras represents a promising and impactful direction for future research. •

Event-based Dense Mapping: Regarding **the event-based mapping** 1

, the current works are mainly focused on the 3D sparse or semi-dense mapping, which is in- herently

due to the sparse nature of the event streams 69

. However, the 3D dense mapping is also

essential for many applications, such as augmented reality 106

, vir- tual reality, and obstacle avoidance. Although there are some works on event- based dense mapping, e.g. learning-based [52, 126, 127, 142], NeRF-based [130, 131, 132, 133, 134], 3DGS [135, 136], and EVI-SAM [1], the performance is still far from satisfactory in terms of accuracy and efficiency. • Event-based Re-localization: Robust place recognition (often referred to as loop 7.2. Future Works closure) and reliable re-localization are vital components of SLAM systems. How- ever,

the motion-dependent nature of event data introduces unique **challenges** for these tasks. **Event** 11

streams are triggered by brightness changes resulting from camera motion, the same scene can generate significantly different event patterns depending on the type and direction of the motion. To tackle this challenge, it is crucial to develop motion-invariant event-based matching techniques capable of extracting consistent visual features or structural cues from event data, indepen- dent of motion patterns. Such approaches would not only improve the reliabil- ity of place recognition but also facilitate accurate re-localization through global bundle adjustment, thereby enhancing both the global consistency and long-term stability of event-based SLAM frameworks. • Event Camera for Practical Application:

> **Research on event cameras is still in** its **early stages and has not** yet **reached the same level** of maturity **as conventional computer vision**. However, significant **progress has been made in** recent **years, both in hardware and software** 9

, demonstrating

> **the potential of event cameras to overcome some of the limitations of frame-based cameras** 9

and enabling access to previously inaccessible scenarios. Crucially, the potential application areas for event cameras require deeper exploration. To fully leverage their unique advan- tages, such as exceptional temporal resolution and ultra-low latency, it will be essential to identify and develop scenarios that are inherently suited to event- based technology. In addition, the high cost of event camera hardware continues to pose a significant barrier, with most devices priced in the tens of thousands of Hong Kong dollar, far surpassing the price of conventional cameras, which typically cost just hundreds or even tens of Hong Kong dollar. Moreover, the op- tical performance

> **of event cameras, such as** sensitivity **and spatial resolution** 67

, still needs substantial improvement to effectively compete with traditional cameras. Appendix A HKU Dataset for Event Camera In this appendix, we introduce our dataset for the evaluation of our event-based monoc- ular and stereo odometry as well as event-based mapping. The dataset is available at:

> **https://github.com/arclab-hku/Event_based_VO-VIO-SLAM** 1

. The event-based HKU dataset facilitates benchmarking for event-based vision systems across various domains, including quadrotor flight and autonomous driving, and also serves as a valuable resource for advancing research in event-based vision. It is a comprehen- sive dataset comprising following subsets: (i) Mono-HKU dataset [2] is designed for benchmarking monocular event-based SLAM. It is collected using hardware attached DAVIS346 (346 × 260) and DVXplorer (640 × 480)

> **for facilitating comparison. All the sequences are recorded in HDR scenarios with very low** illumination or strong **illumi- nation changes through switching the strobe flash on and off** 3

. It also provide large-scale indoor and outdoor sequences, as well as the data squences that are collected in the flighting quadrotor platform. (ii) Stereo-HKU dataset [73]

> **contains stereo event data and stereo image frames with resolution in 346 × 260**, IMU data **as well as** 7

> **ground truth poses from motion capture system** 46

.

> **This is a very** challenge **dataset for event-based** SLAM, features **aggressive motion and HDR scenarios** 7

. (iii) The mapping Dataset is the one that we utilized for real-time event-based dense mapping. This dataset also serves as a benchmarking for comparing event-based mapping approach with data captured from depth camera. (iv) The synthetic dataset is the simulated event agent which gener- ates event data from image data through deep learning tool box. (v) ECMD dataset [6] is an event-based dataset for autonomous driving. It

provides data from two sets of 114

Appendix A. HKU Dataset for Event Camera

stereo event cameras with different resolutions (640 × 480, 346 × 260), 6
stereo indus- trial **cameras, an infrared camera, a top-installed mechanical LiDAR with two slanted LiDARs, two consumer-level GNSS receivers, and an onboard IMU. Meanwhile, the ground-truth of the vehicle was obtained using a centimeter-level high-accuracy GNSS- RTK/INS** navigation **system**

. Monocular HKU-Dataset Table A.1: Summary of the data sequences of Monocular HKU-Dataset rosbag Quadrotor Platform Handheld Handheld Outdoor Sensor DAVIS346

DAVIS346 DVXplorer DVXplorer /dvs_VICON/gt_pose 3

/

dvs_VICON/gt_pose No Ground Truth Topic /davis346/**events /davis346/events /dvxplorer/events /dvs/events** /davis346/**imu /davis346/imu /dvxplorer/imu /dvs/imu** /davis346/**image_raw /davis346/image_raw No Image Event Stream Rate** 60HZ **DAVIS346: 60 HZ DVXplorer: 50 HZ 50HZ** 3

Data size 7 × 105 **DAVIS346: 6** × 105 ***DVXplorer: 2** 3

× 106 2 × 106 Resolution 346 × 260

DAVIS346:346 × 260 **DVXplorer:640 × 480** 640 × 480 Description **indoor aggressive** Quadrotor **under VICON** 3

indoor aggressive HDR scenarios under VICON indoor&outdoor HDR scenarios long-term * Datasize : The number of events per stream, e.g. 2 × 106 **indicates the average number of events for the sequences is 2** × 106 × **50HZ × 156.3s = 1.6** 3

× 1010. This data sequence is for evaluating monocular EVIO in

different resolution event cameras. The **DAVIS346 (346** × 260) **and DVXplorer (640 × 480** 34

)

are attached together (shown in Fig. A.1) **for facilitating comparison. All the sequences are recorded in HDR scenarios with very low** illumination **or strong illumination changes through** switch- ing **the strobe flash on and off, while the sequence** (VICON_**aggressive_hdr) is** character- ized **by aggressive motion** 3

.

Another challenging aspect is the heavy event load of this dataset 3

.

We found that the amount of event data from the high-resolution event cam- era (**such as DVXplorer) is as large as the order of** 106 (**for each event stream, shown in Table** A.1). Most **public event camera** datasets [144, 147], **which only** output **on 30 HZ event stream. However, to further evaluate the performance of our EVIO in the large** 3

Appendix A. HKU Dataset for Event Camera

throughput event load, we modified the driver code for DAVIS346 and DVXplorer with a higher stream rate and not limited maximum events for each stream, which can also ensure a steady frequency of the event stream 3

. What's more, we also provide indoor and outdoor large-scale data sequence as well as the data squences that are collected in the flighting quadrotor platform (Fig. A.2) using DAVIS346. Figure A.1: The Handheld Platform for Monocular Data Collection Figure A.2: The Quadrotor Platform for Monocular Data Collection Stereo HKU-Dataset

This is a very challenge **dataset for** stereo **event-based VO/VIO**, features **aggressive motion and HDR scenarios** 7

. This dataset

contains stereo event data at 60HZ and stereo image frames at 30Hz with resolution in 346 × 260, as well as IMU data at 1000Hz. Timestamps between all sensors are synchronized in hardware. We also provide ground truth poses from a motion capture system VICON at 50Hz during the beginning and end of each sequence, which can be used for trajectory evaluation 7

.

The dataset con- sists **of handheld sequences** including **rapid motion and HDR scenarios** 17

. To alleviate disturbance from the infrared light of

the motion capture system on the event camera, we add an infrared filter on the lens surface of the DAVIS346 camera. Note that this might cause the degradation of preception **for both the event and image camera during the evaluation, but it can also further increase the challenge of our dataset for the only image-based method**. The acquisition platform **is** 7

shown in Fig. A.3.

These two DAVIS346 are rigidly attached with a baseline of 6.0 cm and USB 3.0 interfaces are used to transmit sensor measurements to the NUC. However, since the limitation of our hardware and cost, we use 7

DAVIS346-COLOR and DAVIS346-MONO for the data collection. Although this might introduce some artificial inconsistency, we think it is acceptable for the method evaluation. The DAVIS comprises an image camera and event camera on the same pixel array, thus calibration can be done using standard image-based methods, such as Kalibr 1, on the image frames and then are applied to the event camera. We also provide **the**

data squences that are collected in outdoor large-scale environment. The path length of this data sequence is about 1866m, which

covers the place around 310m in length, 170m in width, and 55m in height changes 7

, from Loke Yew Hall to the Eliot Hall and back to the Loke Yew Hall in HKU campus.

We hope that our dataset can help to push the boundary of future research on event-based VO/VIO algorithms, especially the ones that are really useful and can be applied in practice. Figure A.3: **The** 7

Quadrotor Platform for Stereo Data Collection 1https://github.com/ethz-asl/kalibr Appendix A. HKU Dataset for Event Camera HKU-Dataset for Event Mapping To benefit the community, we release the dataset that we utilized for real-time event- based dense mapping. This dataset also serves as a basis

for comparing our event- based dense mapping approach **with** 1

data captured from a depth camera. Our hand- held

device is shown in Fig. A.4, **which includes a power supply unit, an onboard computer NUC (equipped with Intel i7-1260P, 32GB RAM, and Ubuntu 20.04** opera- tion **system), a DAVIS346 event camera, and a Intel® RealSense™ D455 RGB-D camera** 1

. The infrared ranging sensor of D455 is obscured since the infrared light can interfere with the perception of event cameras. We release the all of our schematic files with STL files format, which can be imported and printed directly. Moreover, we also release the CAD source files (with suffix “*.SLDPRT and *.SLDASM”), which can be opened and edited with Solidworks. (A) (B) Aprilgrid 4s Battery Screen (C) (D) DAVIS 346 Intel D455 NUC Figure A.4: The Event-based Handheld Device for Data Collection ECMD Dataset ECMD is an event-based dataset for autonomous driving. It is first event-based SLAM datasets specifically focus on urbanized autonomou driving, which contains 81 se- quences

and covering over 200 km of various challenging driving scenarios including high-speed motion, repetitive scenarios, dynamic objects, etc 6

. We

explore the inquiry: Are event cameras ready for autonomous driving 6

?

ECMD provides data from two sets of stereo event cameras with different resolutions 6

(640x480, 346x260), stereo indus- trial

cameras, an infrared camera, a top-installed mechanical LiDAR with two slanted LiDARs, two consumer-level GNSS receivers, and an onboard IMU. Meanwhile, the ground-truth of the vehicle was obtained using a centimeter-level high-accuracy GNSS- RTK/INS navigation **system** 6

. The project website and download link of the sequence

is available at: https://arclab-hku.github.io/ecmd 6

/. Dense Street Highway Urban road Bridge Tunnel Suburban road Figure A.

5: The visualization of various scenarios from ECMD [6] **including event streams, RGB images, and infrared images** 6

. Appendix A. HKU Dataset for Event Camera Synthetic HKU-Dataset Large amounts of event data are essential for learning-based methods, yet event cam- eras are both scarce and expensive. To address this, we are releasing synthetic event data for the research community. Synthetic HKU-Dataset involves: (i) converting any existing video / image dataset recorded with standard cameras into synthetic event data using the ESIM tool [206, 227]; (ii) simulating event cameras in Gazebo, which al- lows user to obtain event data from self-developed simulated environments. Detailed instructions for generating the synthetic event dataset can be found in our script https: //kwanwaipang.github.io/File/Blogs/Poster/esim.html. Note that, regarding the training DEIO in Chapter 5,

we simulate events for **all** sequences **of the TartanAir dataset** [205] **using ESIM** 14

[206, 227], as shown in Fig. A.6. In each sequence, we ran- domize the contrast thresholds by independently sampling the negative and positive thresholds from a uniform

distribution with a mean of 0.25 **and a variance of** 126

0.09. (

a) (b) (c) (d) (e) (f) (g) (h) Figure **A**.6: **The** 78

synthetic event stream and the corresponding image frames in the TartanAir dataset. Appendix B Architecture of the Event-based Recurrent Network We adopt a network structure similar to DPVO [26], but with modifications tailored for sparse event data. In this appendix,

we provide a detailed description of the net- work architecture **and the** 66

workflow of the event-based differentiable bundle adjust- ment (eDBA),

as illustrated in Fig. B.1. **The** network **is** 9

designed to process asynchronous event data through a series of interconnected components. Firstly, a CNN network extracts deep feature-based event patches, encoding their local context. These event patches are then tracked across time using a recurrent neural network (RNN), which

estimates the 2D motion (optical flow) of each event **patch** relative to **its connected frames in** the **patch graph** 10

. The estimated optical flow is subsequently integrated into

a differentiable bundle adjustment (DBA) **layer, which** alternately updates **the** depth **and** 10

poses of the event patches. This itera- tive process enables end-to-end learning of sparse patch-based correspondences for the asynchronous event data. The architecture for estimating event-patch trajectories comprises three primary components: (i) a CNN network that extracts patch-based event feature representa- tions, capturing local context through matching and contextual features; (ii) a correla- tion layer that computes visual similarity between these event patches; and (iii) a recur- rent update operator that handles event-patch correspondences in conjunction with dif- ferentiable bundle adjustment. Together, these components form the event-based DBA Appendix B. Architecture of the Event-based Recurrent Network Figure B.1: Workflow of the eDBA for end-to-end pose tracking. problem, enabling the

end-to-end learning of reliable **event-based** optical flow. **The** 45

DBA layer iteratively optimizes the pose and depth of each patch by minimizing the re-projection residual, thereby refining the overall estimation process. This integrated approach leverages the strengths of CNN, RNN, and differentiable bundle adjustment to effectively handle the unique challenges posed by sparse and asynchronous event data, providing a robust framework

for event-based visual odometry. CNN for **Event-based Feature** 38

Extraction We utilize the patch-based structure instead of dense event-based optical flow. To es- timate

the optical flow of the event-based **patches, we** begin by extracting **per-pixel features** 10

, which are then used to compute visual similarities for the event-based data association. The architecture of the CNN-based event feature extraction network is depicted in Fig. B.2. Two similar network structures are utilized: one for extracting matching features and the other for context features. The matching feature network operates without normalization layer and produces an output dimension of 384, while the context feature network employs instance normalization layer and outputs a di- mension of 128. The

network starts with a 7 × 7 convolution **layer with** a **stride** 78

of 2. The layer is

followed by two residual blocks at 1/2 resolution (with a **dimension** of 32) **and** two more **residual blocks at 1/4 resolution** (with a **dimension** 10

of 64). This process results in a final feature map that is 1/4 the resolution of the input. A 1 × 1 convolution layer is finally applied to adjust the output dimensions: 128 for the matching features f ∈ RH/4×W/4×128 and 384 for the context features c ∈ RH/4×W/4×384, respectively. Appendix B. Architecture of the Event-based Recurrent Network Using the event-patch Pk which is centered at a specific pixel location and has a size of p × p, we derive the event-patch matching feature g ∈ Rp×p×128 and the event-patch context feature i ∈ Rp×p×384. Figure B.2:

Architecture of the event-based **feature extractors** network. **D = 128 for the matching-feature extractor and D = 384 for the context-feature extractor** 10

. Event-based Patch Selection and Patch Graph Construc- tion Given an input event stream at frame i, we can represent a set of square event-based patches P with a pitch size of p × p as follows: xk ⌈ yk ⌉ Pk = (B.1) 1 ⌊ dk ⌋ ⌈ ⌉ ⌊ ⌋ where dk is inverse depth for the kth event-patch (assuming a constant depth across the entire patch), and (

xk, yk) is the central pixel location of the square event-based patch. k is the 7

index of the patch in the patch graph. Let i denote the timestamp index

from which the patch Pk is extracted. We can then re-project the patch 10

from timestamp i to timestamp j using the following equation: Pkj = πe · Tji · πe−1 · Pki ′ (B.2) where

πe and πe−1 are the projection and back-projection function of the event camera, respectively 5

(as defined in Eq. (2.9)). Tji is

the relative transformation between times- tamps i and j, represented as 55

: Tji = ⎛ Rji tji ⎞ (B.3) ⎝ 0 1 ⎠ The event-based patch

graph is constructed by creating edges between each event- patch and every event-frame within distance r from the source event-frame of the patch. Specifically, the 10

edges (k, j)

in the patch graph connect patches Pk with frames 10

j. As illustrated in Fig. B.3, multiple event-

patches are extracted from each event-frame and connected to nearby event-frames 10

. The trajectory of an event-patch can be con- structed by re-projecting the patch into

all of its connected frames in the patch graph. Figure B.3: The patch 10

graph. Multiple event-based

patches are extracted from each frame (the blue ones) and are connected to nearby frames (the green and purple 10

ones). Event-based Feature Correlation Operation For the SLAM problem, neighboring frames are often

highly correlated. To exploit this correlation, a 42

correlation operation is employed to capture temporal information across event-based patches in different frames. For a given event patch Pk, we first re-project it from frames i to frames j using Eq. (B.2). This creates the edge (k, j) and yields the re- projected event patch Pkj. Given the frame-based matching feature f ∈ RH/4×W/4×128 extracted from the previously mentioned CNN network for frame j,

we construct a two-level feature pyramid by applying average pooling to the frame-based matching feature f with a 4 × 4 filter with stride 4 10

. The resulting feature pyramid consists of two levels: f lvl=1 ∈ RH/4×W/4×128 and f lvl=4 ∈ RH/16×W/16×128. The correlation operation is then performed between the event-based patch match- ing features g ∈ Rp×p×128 and the obtained two-level feature pyramid of frame j.

For each pixel (u, v) in the event-**patch k, we compute its correlation C** ∈ Rp×**p**×128 with a **7 × 7 grid of** 10

128 dimension feature vectors as follows: C = f (Pkj(u, v) + △) ′ (B.4) where △ is a 7 × 7 integer grid, and f (·) represents bilinear sampling. Finally, the cor- relation matching feature is obtained by concatenating the event-based patch matching features g with their corresponding correlations C. The workflow of this event-based feature correlation operation is illustrated in Fig. B.4. It is worth noting that although the size of event-patch is the p × p, the operation is performed on the H/4 × W/4 feature map. This means that the receptive field of each event-based patch effectively extends up to 4p × 4p. RNN for Event-based Feature Correspondence

For each edge k, j in the patch graph, its **features** 10

consist of the correlation matching feature, the context feature i ∈ Rp×p×384 of the event patch Pk, and the hidden state. These features are used by an RNN network to iteratively refine

the optical flow of the event **patches** 42

. After the iterative refinement (number=12 in our implementation), the depth and camera pose are updated

through a Differentiable Bundle Adjustment (DBA) **layer** 42

. Given that event-based

patches are sparsely distributed **and spatially separated, the** 42

RNN incorporates two key components to enable effective information exchange between event patches: (1) 1D Temporal Convolutions and (2) Softmax-Aggregation. Figure B.4: The workflow of the correlation operation.

For each edge (k, j) in the patch graph, patch k is re-projected **into a frame j using** Eq. (B.**2**). The **matching features** in frame j **are then cropped using the** re-projected **patch** Pkj. **The correlated matching features (green) and context features** (blue) **form the edge features** 10

. These components ensure that temporal and spatial information is effectively propa- gated across event patches. After that, the Transition block updates the hidden state, and the Factor Head generates

2D flow revisions and confidence weights. These out- puts **guide the** optimization process in **the** 10

DBA layer, ensuring accurate updates to depth and camera pose. The structure of the RNN, including its components and work- flow, is shown in Fig. B.5. 1D Temporal Convolutions To exploit temporal correlations between neighboring frames, the RNN

apply a 1D- convolution operation along **the temporal dimension to** 10

the

edges in the patch graph. Since **the** 42

edges

vary in length and are dynamically **added** or **removed** 42

within the keyframe-based sliding window, applying standard batched convolution is not straight- forward. Instead, we use a linear projection for each edge and its temporally adjacent neighbors. Specifically, for an edge (k, j), its temporally adjacent neighbors can be in- dexed as (k, j − 1) and (k, j + 1). These two neighbors are projected using two fully connected layers, and the resulting projections are combined with the original edge. Softmax-Aggregation To leverage spatial correlations among the event-based patches, the RNN apply a softmax- aggregation operation to the

edges in the patch graph. This operation is instantiated **in** 10

two ways: (

1) patch aggregation: edges are considered **neighbors if they connect to the same patch** 10

. For example, edges edge (

k, j) and (k, j + 1 42

) are neighbors because patch k is connected to frame j and j + 1. (

2) frame aggregation: edges are considered neigh- bors **if they connect to the same destination frame and** originate from **different patches** in **the same source frame** 10

. For instance, edges (k, j) and (k + 1, j) are neighbors because they connect patches k and k + 1 from frame i to frame j. Transition block After softmax-aggregation, a transition block is used to update

the hidden state for each edge in the patch graph. The **transition block** consists of **two gated residual units** (GRU), each incorporating **normalization** layer **and ReLU non-linearities** 10

. This process yields an updated hidden state for every edge. Figure B.5: Structure of RNN in [26]. Totally Patches K= num of frame in the sliding window × num of extracted patches for each frame. Totally Edges m= num of frame in the sliding window × Totally Patches K. Factor Head The final block in the RNN

is the factor head, which generates **2D** flow **revisions** 42

and

confidence weights for each edge in the patch graph 10

. It comprises two fully connected layers. The first layer predicts trajectory/edge updates, producing a 2D flow vector $\delta_{k,j} \in R2$ for each edge (k, j). This vector indicates how the re-projection of the center

of the event-**patch should be** adjusted **in** the **2D** plane. While **the second** 10

layer outputs two confidence weightsΣk,j ∈ R2 for each edge (k, j). These weights are constrained to the range (0,1) using a Sigmoid function, reflecting the reliability of the 2D flow revision. These outputs guide the DBA layer in refining the depth and camera pose estimates. DBA Layer for the Depth and Camera Poses Updates The DBA layer ultimately updates the depths of each event-patch and camera poses by using the 2D flow revisions δk,j ∈ R2 and confidence weights Σk,j ∈ R2 through optimizing the following: xk xkj ykj δk,j arg min ∑ ||πe · Tji · πe−1 · yk − [+ (B.5) Tji,dk (k,j)∈gragh 1 1 0]||2Σk,j 0 d k dkj where (xkj, ykj) represents the center of the patch Pk in timestamp j. The physical mean- ing of the DBA layer is to minimize the reprojection error of the event-patch, which is conceptually similar to traditional bundle adjustment (as described in Eq. (2.8)). Appendix C Evaluation Metrics for Pose Tracking Quantitative evaluation of pose tracking is crucial for benchmarking the accuracy of different algorithms, a topic that has been widely explored in numerous studies [179, 182]. In this appendix, we offer a concise overview of the evaluation metrics employed to assess our event-based pose tracking methods. The source code for these metrics is accessible in our GitHub Repository https://github.com/KwanWaiPang/Poster_ files/blob/main/trajectory_evaluation/evo_process.ipynb, which is based on pub- licly available trajectory evaluation tool [182]. The accuracy of a pose tracking algo- rithm is determined by comparing the estimated trajectory (i.e., the time history

of the pose) **with the ground truth** (e.g., a **trajectory provided**	132

by VICON or other motion cap- ture systems). This comparison is vital for understanding and benchmarking various pose tracking algorithms. However, quantitatively comparing

the estimated trajectory with **the ground truth**	61

presents two main challenges: Trajectory Alignment

The estimated trajectory and the ground truth	77

are often expressed in different reference frames, making direct comparison impossible. To address this issue, the estimated trajectory must be properly transformed into the same reference frame as the ground truth, this process also known as trajectory alignment. An example of this alignment

between the estimated **trajectory and the ground truth is**	10

illustrated in Fig. C.1. Appendix C. Evaluation Metrics for Pose Tracking (a) (b) Figure C.1: The estimated trajectory against GT trajectory (a) with and (b) without trajectory alignment. There is no universally accepted method for transforming the estimated trajectory into the ground truth reference frame. However, two common approaches are typically used in practice: • using all the estimated states, i.e.

aligning the whole ground truth trajectory with the **estimated poses**. • using only **the**	1

first one or several initial states, i.e.

aligning 5 seconds [0-5s] of the estimated trajectory with the ground truth. The	1

former method generally results in a lower overall error, while the latter tends to produce an error that increases over time. Error Metrics A trajectory comprises states at numerous time points, making it a high-dimensional data. Summarizing the information from the entire trajectory into concise accuracy metrics poses a significant challenge. To tackle this, meaningful error metrics, such as

the absolute trajectory error (ATE) and relative pose error (RPE	20

), must be employed, and their properties thoroughly understood. Appendix C. Evaluation Metrics for Pose Tracking The ATE, also referred to as

the absolute pose error (APE), is a **metric** used **to** eval- uate **the** global consistency **of** 82

a SLAM trajectory. It is calculated based on the absolute relative pose between two poses Pref,i, Pest,i ∈ SE(3) at timestamp i:

Ei = Pest,i ⊖ Pref,i = Pe−st1,iPref,i ∈ SE(3) where ⊖ is the inverse compositional operator, which takes **two poses and** gives the **relative pose. Pest,i** is **the** 64

estimated pose at timestamp i, and Pref,i is the ground truth pose at timestamp i. You can use different pose relations to calculate the ATE: • Using the translation part of Ei: – APEi = ∥trans(Ei)∥ • Using the rotation angle of Ei: – APEi = |angle(logSO(3)(rot(Ei)))|, where logSO(3)(·) is the inverse of expso(3)(·) • Using the rotation part of Ei: – APEi = ∥rot(Ei) − I3×3∥F • Using the full relative pose Ei: – APEi = ∥Ei − I4×4∥F Then, different statistics can be calculated on the these ATE of all timestamps, e.g., the RMSE: N RMSE = 1 N i=1 ∑ ATE2i √ The RPE is a metric used to assess the local consistency of a SLAM trajectory. It evaluates the relative poses along the estimated trajectory against those of the ground truth trajectory. This is calculated based on the difference in delta poses: Ei,j = δesti,j ⊖ δrefi,j = (Pr−e1f,iPref,j)−1(Pe−st1,iPest,j) ∈ SE(3)

where ⊖ is the inverse compositional operator, which takes **two poses and** gives the **relative pose. Pest,i and** Pest,j is **the** 64

estimated pose at timestamp i and j, and Pref,i and Appendix C. Evaluation Metrics for Pose Tracking Pref,j is the ground truth pose at timestamp i and j. You can use different pose relations to calculate the RPE from timestamp i to j: • Using the translation part of Ei,j: – RPEi,j = ∥trans(Ei,j)∥ • Using the absolute angular error of Ei,j: – RPEi,j = |angle(logSO(3)(rot(Ei,j)))|, where logSO(

3)(·) is the inverse of expso(**3)(·) (Rodrigues' formula**) • Using **the** 63

rotation part of Ei,j: – RPEi,j = ∥rot(Ei,j) − I3×3∥F • Using the full delta pose difference Ei,j: – RPEi,j = ∥Ei,j − I4×4∥F Then, different statistics can be calculated on the RPEs of all timestamps, e.g., the RMSE: RMSE = 1 √ ∀ i,j N ∑ RPEi2,j Appendix D Evaluation Metrics for 3D Mapping The performance evaluation of depth estimation methods is conducted using the stan- dardized metrics established by Ming et al. [228] and Eigen et al. [229]. These metrics consist of both error measurements and accuracy thresholds. The error metrics (with lower values indicating better performance) include:

absolute relative error (Abs.**rel), square relative error** (Sq.rel), **root mean square error (RMSE**), and the logarithm **root mean square error** (log RMS). The **accuracy** 37

1.25t, with threshold expo- nents t = 1, 2, 3. The definitions of these metrics are provided below: RMS : √ T i∈T 1 ∑ ||di − digt||2 (D.1)

log RMS : √ T i∈T 1 ∑ || **log** di − **log** dgit||2 (**D** 78

.2) abs.relative : 1 ∑ di − dgit

T i∈T d gt **i (D**.3) sq.**relative : 1** ||di − dgit||2 **T** ∑ **i∈T** 116

thr (D.5) In Section 6.7.3.2, we evaluate the performance of our event-based mapping us- ing the evaluation metrics of mean depth error in Eq. (D.1). We also release our im- plementation of the evaluation metrics on GitHub Repository https://github.com/ arclab-hku/depth_evaluation_tools. Bibliography [1]

W. Guan, P. Chen, H. Zhao, Y. Wang, and P. Lu. “EVI-SAM: Robust, Real-Time, Tightly-Coupled Event–Visual–Inertial State Estimation and 3D Dense Map- ping”. In: Advanced Intelligent Systems (2024), p. 2400243 11

. [2]

W. Guan and P. Lu. “Monocular Event Visual Inertial Odometry based on Event- corner using Sliding Windows Graph-based Optimization”. In: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE. 2022, pp. 2438– 2445 1

. [3] S.

Guo and G. Gallego. “CMax-SLAM: Event-based Rotational-Motion Bundle Adjustment and SLAM System using Contrast Maximization”. In: IEEE Trans- actions on Robotics 12

(2024). [4]

G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leuteneg- ger, A. J. Davison, J. Conradt, K. Daniilidis, et al. “Event-based vision: A sur- vey”. In: IEEE transactions on pattern analysis and machine intelligence 44.1 (2020), pp. 154–180. [5] D. Gehrig, H. Rebecq 23

,

G. Gallego, and D. Scaramuzza. “EKLT: Asynchronous photometric feature tracking using events and frames”. In: International Journal of Computer Vision 128.3 (2020), pp. 601–618 8

. [6]

P. Chen, W. Guan, F. Huang, Y. Zhong, W. Wen, L.-T. Hsu, and P. Lu. “ECMD: An Event-Centric Multisensory Driving Dataset for SLAM”. In: IEEE Transactions on Intelligent Vehicles (2023 1

). [7]

J. Yin, A. Li, T. Li, W. Yu, and D. Zou. “M2dgr: A multi-sensor and multi-scenario slam dataset for ground robots”. In: IEEE Robotics and Automation Letters 7.2 (2021), pp. 2266–2273 8

. Bibliography [8]

T. Qin, P. Li, and S. Shen. “Vins-mono: A robust and versatile monocular visual- inertial state estimator”. In: IEEE Transactions on Robotics 34.4 (2018), pp. 1004– 1020 8

. [9]

C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós. "Orb- slam3: An accurate open-source library for visual, visual–inertial, and multimap slam". In: IEEE Transactions on Robotics 37.6 (2021), pp. 1874–1890. [10] J. Engel, V. Koltun, and D. Cremers. "Direct sparse odometry". In: IEEE transac- tions on pattern analysis and machine intelligence 40.3 (2017), pp. 611–625 22

. [11]

J. H. Jung, Y. Choe, and C. G. Park. "Photometric Visual-Inertial Navigation With Uncertainty-Aware Ensembles". In: IEEE Transactions on Robotics 38.4 (2022), pp. 2039–2052 30

. [12]

R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. "ORB-SLAM: a versatile and accurate monocular SLAM system". In: IEEE transactions on robotics 31.5 (2015), [13] pp. 1147–1163. R. Mur-Artal and J. D. Tardós. "Orb-slam2: An open-source slam system for 22

[14]

P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang. "Openvins: A research platform for visual-inertial estimation". In: 2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE. 2020, pp. 4666–4672 54

. [

15] J. Delmerico, T. Cieslewski, H. Rebecq, M. Faessler, and D. Scaramuzza. "Are we ready for autonomous drone racing? the UZH-FPV drone racing dataset". In: 2019 International Conference on Robotics and Automation (ICRA). IEEE. 2019, pp. 6713–6719. [16 6

]

J. Engel, J. Sturm, and D. Cremers. "Semi-dense visual odometry for a monocu- lar camera". In: Proceedings of the IEEE international conference on computer vision. 2013, pp. 1449–1456. [17] J. Engel, T. Schöps, and D. Cremers. "LSD-SLAM: Large-scale direct monocular SLAM". In: European conference on computer vision. Springer. 2014, pp. 834–849 58

. [18]

C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza. "SVO: Semidirect visual odometry for monocular and multicamera systems". In: IEEE Transactions on Robotics 33.2 (2016), pp. 249–265 9

. Bibliography [19]

L. Von Stumberg and D. Cremers. “DM-VIO: Delayed marginalization visual- inertial odometry”. In: IEEE Robotics and Automation Letters 7.2 (2022), pp. 1408– 1415 43

. [20]

D. DeTone, T. Malisiewicz, and A. Rabinovich. “Superpoint: Self-supervised in- terest point detection and description”. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. 2018, pp. 224–236. [21] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich. “Superglue: Learning feature matching with graph neural networks”. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020, pp. 4938–4947 56

. [22]

B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. “Nerf: Representing scenes as neural radiance fields for view synthesis”. In: Communications of the ACM 65.1 (2021), pp. 99–106. [23] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. “3d gaussian splatting for real-time radiance field rendering”. In: ACM Transactions on Graphics 42.4 (2023 24

), pp. 1–14. [24] H. Zhao, W. Guan, and P. Lu. “LVI-GS: Tightly-coupled LiDAR-Visual-Inertial [25] Z.

Teed and J. Deng. “Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras”. In: Advances in neural information processing systems 34 (2021), pp. 16558–16569 70

. [26]

Z. Teed, L. Lipson, and J. Deng. “Deep patch visual odometry”. In: Advances in Neural Information Processing Systems 36 (2024 86

). [27] S. Wang, V. Leroy, Y. Cabon, B.

Chidlovskii, and J. Revaud. “Dust3r: Geometric 3d vision made easy”. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 59

. 2024, pp. 20697–20709. [28] V. Leroy, Y.

Cabon, and J. Revaud. “Grounding image matching in 3d with mast3r 59

”.

In: European Conference on Computer Vision. Springer. 2024, pp. 71–91. [29] R. Murai, E. Dexheimer, and A 97

. J. Davison. “MASt3R-

SLAM: Real-Time Dense SLAM with 3D Reconstruction Priors". In 144

: (2025). [30]

H. Kim, A. Handa, R. Benosman, S.-H. Ieng, and A. J. Davison. "Simultaneous mosaicing and tracking with an event camera". In: J. Solid State Circ 43 (2008), pp. 566–576 39

. [31]

D. Weikersdorfer and J. Conradt. "Event-based particle filtering for robot self- localization". In: 2012 IEEE International Conference on Robotics and Biomimetics (ROBIO). IEEE. 2012, pp. 866–870. [32] D. Weikersdorfer, R. Hoffmann, and J. Conradt. "Simultaneous localization and mapping for event-based vision systems". In: International Conference on Com- puter Vision Systems. Springer. 2013, pp. 133–142 39

. [33]

A. Censi and D. Scaramuzza. "Low-latency event-based visual odometry". In: 2014 IEEE International Conference on Robotics and Automation (ICRA). IEEE. 2014, pp. 703–710. [34] D. Weikersdorfer, D. B. Adrian, D. Cremers, and J. Conradt. "Event-based 3D SLAM with a depth-augmented dynamic vision sensor". In: 2014 IEEE interna- tional conference on robotics and automation (ICRA). IEEE. 2014, pp. 359–364. [35] E. Mueggler, B 48

.

Huber, and D. Scaramuzza. "Event-based, 6-DOF pose tracking 41

[36]

G. Gallego, C. Forster, E. Mueggler, and D. Scaramuzza. "Event-based camera pose tracking using a generative event model". In: arXiv preprint arXiv:1510.01972 (2015 63

). [37]

B. Kueng, E. Mueggler, G. Gallego, and D. Scaramuzza. "Low-latency visual odometry using event-based feature tracks". In: 2016 IEEE/RSJ International Con- ference on Intelligent Robots and Systems (IROS). IEEE. 2016, pp. 16–23. [38] H. Kim, S. Leutenegger, and A. J. Davison. "Real-time 3D reconstruction and 6- DoF tracking with an event camera". In: European Conference on Computer Vision. Springer. 2016, pp. 349–364 15

. [39]

W. Yuan and S. Ramalingam. "Fast localization and tracking using event sen- sors". In: 2016 IEEE International Conference on Robotics 20

and Automation (ICRA). IEEE. 2016, pp. 4564–4571

. [

40] H. Rebecq, T. Horstschäfer, G. Gallego, and D. Scaramuzza. "Evo: A geometric approach to event-based 6-dof parallel tracking and mapping in real time". In: IEEE Robotics and Automation Letters 2.2 (2016), pp. 593–600. [41] H. Rebecq 8

,

G. Gallego, E. Mueggler, and D. Scaramuzza. "EMVS: Event-based multi-view stereo—3D reconstruction with an event camera in real-time". In: International Journal of Computer Vision 126.12 (2018), pp. 1394–1414 39

. [42]

G. Gallego and D. Scaramuzza. "Accurate angular velocity estimation with an event camera". In: IEEE Robotics and Automation Letters 2.2 (2017), pp. 632–639 39

. [43]

A. Zihao Zhu, N. Atanasov, and K. Daniilidis. "Event-based visual inertial odom- etry". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recog- nition. 2017, pp. 5391–5399. [44] H. Rebecq, T. Horstschaefer, and D. Scaramuzza. "Real-time Visual-Inertial Odom- [45] etry for Event Cameras using Keyframe-based Nonlinear Optimization". In: British Machine Vision Conference (BMVC). 2017. C. Reinbacher, G. Munda, and T. Pock. "Real-time panoramic tracking for event cameras". In: 2017 IEEE International Conference on 46

Computational Photography (ICCP). IEEE. 2017, pp. 1–9 89

. [46]

G. Gallego, J. E. Lund, E. Mueggler, H. Rebecq, T. Delbruck, and D. Scaramuzza. "Event-based, 6-DOF camera tracking from photometric depth maps". In: IEEE transactions on pattern analysis and machine intelligence 40.10 81

(2017), pp. 2402–2412. [47]

G. Gallego, H. Rebecq, and D. Scaramuzza. "A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation". In: Proceedings of the IEEE conference on computer vision and pat- tern recognition. 2018, pp. 3867–3876. [48] E. Mueggler, G. Gallego, H. Rebecq, and D. Scaramuzza 41

. "

Continuous-time visual- inertial odometry for event cameras". In: IEEE Transactions on Robotics 34.6 (2018), pp. 1425–1440 8

. [49]

A. R. Vidal, H. Rebecq, T. Horstschaefer, and D. Scaramuzza. "Ultimate SLAM? Combining events, images, and IMU for robust visual SLAM in HDR and high- speed scenarios". In: IEEE Robotics and Automation Letters 3.2 (2018), pp. 994– 1001 8

. [50] S.

Bryner, G. Gallego, H. Rebecq, and D. Scaramuzza. "Event-based, direct cam- era **tracking from a photometric 3d map** using **nonlinear optimization". In: 2019 International Conference on Robotics and Automation (ICRA**). IEEE. 2019, **pp. 325– 331**. [51] **D. Zhu, Z. Xu, J. Dong, C. Ye, Y. Hu, H. Su, Z. Liu, and G. Chen. "Neuromorphic visual odometry system for intelligent vehicle application with bio-inspired** vi- sion **sensor". In: 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO**). IEEE. 2019, **pp. 2225–2232** 20

. [52]

A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis. "Unsupervised event-based learning of optical flow, depth, and egomotion". In: Proceedings of the **IEEE**/CVF **Conference on Computer Vision and Pattern Recognition. 2019, pp. 989–997**. [53] **C. Ye, A. Mitrokhin, C. Fermüller, J. A. Yorke, and Y. Aloimonos. "Unsupervised learning of dense optical flow, depth and egomotion with event-based sensors". In**: 2020 **IEEE/RSJ International Conference on Intelligent Robots and Systems** (IROS). **IEEE. 2020, pp. 5831–5838** 44

. [54]

T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. "Unsupervised learning of 20

[55]

C. Le Gentil, F. Tschopp, I. Alzugaray, T. Vidal-Calleja, R. Siegwart, and J. Nieto. "IDOL: A framework for IMU-DVS odometry using lines". In: 2020 IEEE/RSJ 1

In- ternational

Conference on Intelligent Robots and Systems (IROS). IEEE. 2020, **pp**. 5863– 5870. [56] **D. Liu, A. Parra, and T.-J. Chin. "Globally optimal contrast** maximisation **for event-based motion estimation". In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020, pp. 6349–6358**. [57] **X. Peng**, Y. Wang, **L. Gao, and L. Kneip. "Globally-optimal event** 41

camera motion

estimation". In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI 16 — 88

.

Springer. 2020, pp. 51–67. [58] X. Peng, L. Gao, Y. Wang, and L. Kneip. "Globally-optimal contrast maximisa- tion for event cameras". In: IEEE Transactions on Pattern Analysis and Machine Intelligence — 40

44.7 (2021), pp. 3479–3495. [59] Y. Zhou, G.

Gallego, and S. Shen. "Event-based stereo visual odometry". In: IEEE Transactions on Robotics 37.5 (2021), pp. 1433–1450 — 47

. [60]

A. Hadviger, I. Cvišić, I. Marković, S. Vražić, and I. Petrović. "Feature-based — 8

[61] H.

Kim and H. J. Kim. "Real-time rotational motion estimation with contrast maximization over globally aligned events". In: IEEE Robotics and Automation Letters 6.3 (2021 — 71

), pp. 6016–6023. [62]

D. Liu, A. Parra, and T.-J. Chin. "Spatiotemporal registration for event-based visual odometry". In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021, pp. 4937–4946 — 32

. [63]

D. Liu, A. Parra, Y. Latif, B. Chen, T.-J. Chin, and I. Reid. "Asynchronous opti- misation for event-based visual odometry". In: 2022 International Conference on Robotics and Automation (ICRA). IEEE. 2022, pp. 9432–9438 — 1

. [64]

Y. Wang, J. Yang, X. Peng, P. Wu, L. Gao, K. Huang, J. Chen, and L. Kneip. "Visual — 20

[65]

B. Dai, C. Le Gentil, and T. Vidal-Calleja. "A tightly-coupled event-inertial odom- etry using exponential decay and linear preintegrated measurements — 1

”.

In: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE. 2022, pp. 9475–9482. [66] F. Mahlknecht, D. Gehrig, J. Nash, F. M. Rockenbauer, B. Morrell, J. Delaune, and D. Scaramuzza. “Exploring Event Camera-based Odometry for Planetary Robots”. In: IEEE Robotics and Automation Letters — 20

(RA-L) (2022). [67]

J. Hidalgo-Carrió, G. Gallego, and D. Scaramuzza. “Event-aided Direct Sparse Odometry”. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022, pp. 5781–5790 — 1

. [68]

Y.-F. Zuo, J. Yang, J. Chen, X. Wang, Y. Wang, and L. Kneip. “DEVO: Depth- Event Camera Visual Odometry in Challenging Conditions”. In: 2022 Interna- tional Conference on Robotics and Automation (ICRA). IEEE. 2022, pp. 2179–2185. [69] W. Chamorro, J — 73

.

Solà, and J. Andrade-Cetto. “Event-based line SLAM in real- time”. In: IEEE Robotics and Automation Letters 7.3 (2022), pp. 8146–8153 — 47

. [70]

W. Chamorro, J. Solà, and J. Andrade-Cetto. “Event-IMU Fusion Strategies for Faster-Than-IMU Estimation Throughput”. In: Proceedings of the IEEE/CVF Con- ference on Computer Vision and Pattern Recognition. 2023, pp. 3975–3982 — 1

. [71] X. Liu, H. Xue, X. Gao, H. Liu, B. Chen, and S. S. Ge. “Cubic B-Spline-

Based Feature Tracking for Visual–Inertial Odometry With Event Camera — 38

”.

In: IEEE Transactions on Instrumentation and Measurement 72 (2023), pp. 1 — 43

–15. [72]

W. Guan, P. Chen, Y. Xie, and P. Lu. “PL-EVIO: Robust Monocular Event-based Visual Inertial Odometry with Point and Line Features”. In: IEEE Transactions on Automation Science and Engineering (2023). [73] P. Chen, W. Guan, and P. Lu. “ESVIO: Event-based Stereo Visual Inertial Odom- etry”. In: IEEE Robotics and Automation Letters — 1

8.6 (2023),

pp. 3661–3668. [74] Z. Liu, D. Shi, R. Li, and S. Yang. “Esvio: Event-based stereo visual-inertial odometry”. In: Sensors 4

23.4 (2023), p. 1998. [75]

J. Wang and J. D. Gammell. “Event-based stereo visual odometry with native temporal resolution via continuous-time gaussian process regression”. In: IEEE Robotics and Automation Letters (2023 72

). [76]

J. Huang, S. Zhao, T. Zhang, and L. Zhang. “MC-VEO: A Visual-Event 6

Odome- [77] A. El Moudni, F. Morbidi, S. Kramm, and R. Boutteau. “An Event-based Stereo 3D Mapping and Tracking Pipeline for Autonomous

Vehicles”. In: 2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC). IEEE. 2023, pp 54

. 5962–5968. [78] S.

Ghosh and G. Gallego. “Multi-Event-Camera Depth Estimation and Outlier Rejection by Refocused Events Fusion”. In: Advanced Intelligent Systems 4.12 (2022 68

), p. 2200221. [79] S.

Ghosh and G. Gallego. “Event-based Stereo Depth Estimation from Ego-motion using Ray Density Fusion”. In: European Conf. on Computer Vision Workshops (EC- CVW) Ego4D (2022 1

). [80]

M. S. Lee, J. H. Jung, Y. J. Kim, and C. G. Park. “Event-and Frame-based Visual- Inertial Odometry with Adaptive Filtering based on 8-DOF Warping Uncer- tainty”. In: IEEE Robotics and Automation Letters (2023 1

). [81]

A. Safa, T. Verbelen, I. Ocket, A. Bourdoux, H. Sahli, F. Catthoor, and G. Gielen. “Fusing event-based camera and radar for slam using spiking neural networks with continual stdp learning”. In: 2023 IEEE International Conference on Robotics and Automation (ICRA). IEEE. 2023, pp. 2782–2788 12

. [82]

X. Huang, Y. Zhang, and Z. Xiong. “Progressive spatio-temporal alignment for efficient event-based motion estimation”. In: 12

Proceedings of the IEEE/CVF Confer- ence **on Computer Vision and Pattern Recognition. 2023, pp. 1537–1546**

. [83] K. Tang, X. Lang, Y. Ma, Y. Huang, L. Li, Y. Liu, and J. Lv. “Monocular Event- Inertial Odometry with Adaptive decay-based Time Surface and Polarity-aware

Tracking”. In: IEEE/RSJ International Conference on Intelligent Robots (IROS 20

). 2024. [84] S.

Klenk, M. Motzet, L. Koestler, and D. Cremers. “Deep event visual odome- try”. **In: 2024 International Conference on 3D Vision (3DV). IEEE. 2024, pp. 739– 749** 11

. [85]

R. Pellerito, M. Cannici, D. Gehrig, J. Belhadj, O. Dubois-Matra, M. Casasco, and D. Scaramuzza. “Deep Visual Odometry with Events and Frames”. In: IEEE/RSJ International Conference on Intelligent Robots (IROS 11

). 2024. [86] S.

Guo and G. Gallego. “Event-based Photometric **Bundle Adjustment”. In** 12

: arXiv preprint arXiv:2412.14111 (2024). [87] S.

Guo and G. Gallego. “Event-based mosaicing bundle adjustment”. In: Euro- pean **Conference on Computer Vision. Springer**. 2024, **pp. 479–496** 12

. [88] S. Ghosh, V.

Cavinato, and G. **Gallego. “ES-PTAM: Event-based Stereo Parallel Tracking and Mapping”. In** 16

: arXiv preprint arXiv:2408.15605 (2024). [89] Z. Wang, X. Li, T. Liu, Y. Zhang, and P. Huang. “Efficient Continuous-Time Ego- Motion

Estimation for Asynchronous **Event-based** Data Associations”. **In: arXiv preprint arXiv** 39

:2402.16398 (2024). [90] Z. Wang, X. Li, Y. Zhang, F. Zhang, et al. “AsynEIO: Asynchronous Monocular Event-Inertial Odometry Using Gaussian Process Regression”. In: arXiv preprint arXiv:2411.12175 (2024). [91] X. Li, Z. Wang, Z. Liu, Y. Zhang, F. Zhang, X. Yao, and P. Huang. “Asynchronous Event-Inertial Odometry using a Unified Gaussian Process Regression Frame- work”. In: arXiv preprint arXiv:2412.03136 (2024). [92]

Y.-F. Zuo, W. Xu, X. Wang, Y. Wang, and L. Kneip. “Cross-modal semi- dense **6- dof tracking of an event camera in challenging conditions**”. In: **IEEE Transactions on Robotics** 12

(2024). [93] R. Yuan, T. Liu, Z. Dai, Y.-F. Zuo, and L. Kneip. “

EVIT: Event-based visual- inertial tracking in semi-dense maps using windowed nonlinear optimization". In: (**2024** 38

), pp. 10656–10663. [94]

A. Soliman, F. Bonardi, D. Sidibé, and S. Bouchafa. "Dh-ptam: a deep hybrid stereo events-frames parallel tracking and mapping system". In: **IEEE** Transac- tions **on Intelligent Vehicles (2024** 11

). [95] Z. Wang, X. Li, Y. Zhang, F. Zhang, and P. Huang. "Continuous Gaussian Pro- cess Pre-Optimization for Asynchronous Event-Inertial Odometry". In: arXiv preprint arXiv:2412.08909 (2024). [96] Z. Wang, Y. Ge, K. Dong, I.-M. Chen, and J. Wu. "

FAST-LIEO: Fast and Real- Time LiDAR-Inertial-Event-Visual Odometry". In: **IEEE** 38

Robotics and Automation Letters (2024). [97]

C. Zheng, Q. Zhu, W. Xu, X. Liu, Q. Guo, and F. Zhang. "Fast-livo: Fast and tightly-coupled sparse-direct lidar-inertial-visual odometry". In: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE. 2022, pp. 4003– 4009 27

. [98] K. Wang, K. Zhao, W. Lu, and Z. You. "Stereo

Event-Based Visual–Inertial Odom- etry". In 111

: Sensors (Basel, Switzerland) 25.3 (2025), p. 887. [99] N. Xu, L. Wang, Z. Yao, and T. Okatani. "

METS: Motion-Encoded Time-Surface for Event-Based High-Speed Pose Tracking 61

".

In: International Journal of Computer Vision 9

(2025), pp. 1–19. [100]

J. Niu, S. Zhong, and Y. Zhou. "Imu-aided event-based stereo visual odometry 72

". In: (2024), pp. 11977–11983. [101]

J. Niu, S. Zhong, X. Lu, S. Shen, G. Gallego, and Y. Zhou. "Esvo2: Direct visual- inertial odometry with stereo event cameras 11

". In: IEEE Transactions on Robotics (2025). [102]

W. Guan, F. Lin, P. Chen, and P. Lu. "DEIO: Deep Event Inertial Odometry". In: **arXiv preprint arXiv:2411.03928 (2024** 11

). [103]

> **B. Choi, H. Lee, and C. G. Park. "Event-Frame-Inertial Odometry Using Point and Line Features Based on Coarse-to-Fine Motion Compensation**". In: **IEEE Robotics and Automation Letters (2025**). [104] K. Chen, **J** 11

. Zhang, and F. Fraundorfer. "EVLoc: Event-based Visual Localization in LiDAR Maps via Event-Depth Registration". In: arXiv preprint arXiv:2503.00167 (2025). [105]

> **Z. Teed and J. 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. Springer. 2020, pp. 402–419** 13

. [106] P. Chen, F. Lin, W. Guan, and P. Lu. "

> **SuperEIO: Self-Supervised Event** Fea- ture **Learning for Event Inertial Odometry** 11

". In: arXiv preprint arXiv:2503.22963 (2025). [107] Y. Burkhardt, S. Schaefer, and S. Leutenegger. "

> **SuperEvent: Cross-Modal** Learn- ing **of Event-based Keypoint Detection** 16

". In: arXiv preprint arXiv:2504.00139 (2025). [108]

> **E. Rosten, R. Porter, and T. Drummond. "Faster and better: A machine** 5

learn- [109]

> **B. D. Lucas, T. Kanade, et al. "An iterative image registration technique with an application to stereo vision". In: Vancouver, British Columbia. 1981** 8

. [110]

> **M. S. Lee, Y**. J. **Kim, J. H. Jung, and C. G. Park. "Fusion of events and frames** 50

[111]

> **W. O. Chamorro Hernández, J. Andrade-Cetto, and J. Solà Ortega. "High-speed event camera tracking". In: Proceedings of the The 31st British Machine Vision** Vir- tual **Conference. 2020, pp. 1–12** 41

. [112]

> **L. Gao, H. Su, D. Gehrig, M. Cannici, D. Scaramuzza, and L. Kneip. "A 5-Point Minimal Solver for Event Camera Relative Motion Estimation".** 76

In: Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023, **pp**

. 8049–8059. [113]

L. Gao, D. Gehrig, H. Su, D. Scaramuzza, and L. Kneip. "An N-Point Linear Solver for Line and Motion Estimation with Event Cameras". In 12

: arXiv preprint arXiv:2404.00842 (2024). [114]

J. Chui, S. Klenk, and D. Cremers. "Event-Based Feature Tracking in Continuous Time with Sliding Window Optimization". In: **arXiv preprint arXiv:2107.04536 (2021** 12

). [115] W. Xing, S. Lin, L. Yang, Z. Zhang, Y. Du, M. Lei,

Y. Pan, and **J**. Pan. "**EROAM: Event-based Camera Rotational Odometry and Mapping in Real-time**". In: **arXiv** 38

preprint arXiv:2411.11004 (2024). [116] S.

Shiba, Y. Klose, Y. Aoki, and G. Gallego. "Secrets of Event-based Optical Flow, Depth and Ego-motion Estimation by Contrast Maximization". In: **IEEE** Trans- actions **on Pattern Analysis and Machine Intelligence** 12

(2024). [117]

C. Gu, E. Learned-Miller, D. Sheldon, G. Gallego, and **P. Bideau. "The spatio- temporal poisson point process: A simple model for the alignment of event camera data". In: Proceedings of the IEEE/CVF International Conference on** Com- puter **Vision. 2021, pp. 13495–13504** 40

. [

118] U. M. Nunes and Y. Demiris. "Entropy minimisation framework for event-based vision model estimation". In: Computer Vision–ECCV 2020 39

: 16th European Confer- [119]

U. M. Nunes and Y. Demiris. "Robust event-based vision model estimation by dispersion minimisation". In: **IEEE Transactions on Pattern Analysis and Machine Intelligence** 44.12 (2021), **pp** 90

. 9561–9573. [120] F. Hamann, Z. Wang, I. Asmanis, K. Chaney, G. Gallego, and K. Daniilidis. "

Motion-prior contrast maximization for dense continuous-time motion 38

estima- tion".

In: European Conference on Computer Vision. Springer. 2024, pp. 18–37. [121 65

]

J. Revaud, C. De Souza, M. Humenberger, and P. Weinzaepfel. “R2d2: Reliable and repeatable detector and descriptor”. In: Advances in neural information pro- cessing systems 32 (2019 54

). [122]

Y. Dong. “Standardand Event Cameras Fusion for Dense Mapping”. In: arXiv preprint arXiv:2102.03567 (2021). [123] H. Cho, J. Jeong, and K.-J. Yoon. “EOMVS: Event-Based Omnidirectional Multi- View Stereo”. In: IEEE Robotics and Automation Letters 1

6.4 (2021), pp. 6709–6716. [124] S.

Chiavazza, S. M. Meyer, and Y. Sandamirskaya. “Low-latency monocular depth estimation using event timing on neuromorphic hardware”. In: Proceed- ings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023, pp. 4070–4079. [125] K. Chaney, A. Zihao Zhu, and K. Daniilidis. “Learning event-based height from plane and parallax”. In: Proceedings of the IEEE/CVF Conference on Computer Vi- sion and Pattern Recognition Workshops. 2019, pp. 1–4. [126] J. Hidalgo-Carrió, D. Gehrig, and D. Scaramuzza. “Learning monocular dense depth from events”. In: 2020 International Conference on 3D Vision (3DV). IEEE. 2020, pp. 534–542. [127] D. Gehrig, M. Rüegg, M. Gehrig, J. Hidalgo-Carrió, and D. Scaramuzza. “Com- bining events and frames using recurrent asynchronous multimodal networks for monocular depth prediction”. In: IEEE Robotics and Automation Letters 1

6.2 (2021), pp. 2822–2829. [128]

X. Liu, J. Li, J. Shi, X. Fan, Y. Tian, and D. Zhao. “Event-based Monocular Depth Estimation with Recurrent Transformers 21

”.

In: IEEE Transactions on Circuits and Systems for Video Technology 74

(2024). [129] A. Devulapally, M. F. F. Khan, S. Advani, and V. Narayanan. “Multi-Modal Fusion of Event and RGB for Monocular Depth Estimation Using a Unified Transformer-based Architecture”.

In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024, pp. 2081–2089. [130] V. Rudnev, M. Elgharib, C. Theobalt, and V. Golyanik. “EventNeRF: Neural ra- diance fields from a single colour event camera”. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023, pp. 4992–5002 1

. [131]

I. Hwang, J. Kim, and Y. M. Kim. “Ev-NeRF: Event based neural radiance field”. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vi- sion. 2023, pp. 837–847. [132] S. Klenk, L. Koestler, D. Scaramuzza, and D. Cremers. “E-nerf: Neural radiance fields from a moving event camera”. In: IEEE Robotics and Automation Letters 1

8.3 (2023), pp. 1587–1594. [133] A. Bhattacharya, R. Madaan, F. Cladera, S. Vemprala, R. Bonatti, K. Daniilidis, A. Kapoor, V. Kumar, N. Matni, and J. K. Gupta. “EvDNeRF: Reconstructing Event Data with Dynamic

Neural Radiance Fields”. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2024, pp 1

. 5846–5855. [134] S.

Mahbub, B. Feng, and C. Metzler. “Multimodal Neural Surface Reconstruc- tion: Recovering the Geometry and Appearance of 3D Scenes from Events and Grayscale Images”. In: NeurIPS 2023 Workshop on Deep Learning and Inverse Prob- lems. 2023 1

. [135] J. Huang, C. Dong, and P. Liu. “

IncEventGS: Pose-Free Gaussian Splatting from a Single Event Camera”. In: arXiv 119

preprint arXiv:2410.08107 (2024). [136]

T. Xiong, J. Wu, B. He, C. Fermuller, Y. Aloimonos, H. Huang, and C. Metzler. “Event3dgs: Event-based 3d gaussian splatting for 50

high-speed robot egomo- tion”. In: 8th Annual Conference on Robot Learning 71

. 2024. [137] S.-

H. Ieng, J. Carneiro, M. Osswald, and R. Benosman. “Neuromorphic event- based generalized time-based stereovision”. In: Frontiers in Neuroscience 12 (2018 40

), [138] p. 442.

Y. Zhou, G. Gallego, H. Rebecq, L. Kneip, H. Li, and D. Scaramuzza. “Semi 1

- [139]

Z. Liu, D. Shi, R. Li, Y. Zhang, and S. Yang. “T-ESVO: Improved Event-Based Stereo Visual Odometry via Adaptive Time-Surface and Truncated Signed Dis- tance Function”. In: Advanced Intelligent Systems 5.9 (2023), p. 2300027 1

. [140] S.

> Tulyakov, F. Fleuret, M. Kiefel, P. Gehler, and M. Hirsch. “Learning an event sequence embedding for dense event-based deep stereo”. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019, pp. 1527–1537. [141] Y. Nam, M. Mostafavi, K.-J. Yoon, and J. Choi. “Stereo Depth from Events Cam- eras: Concentrate and Focus on the Future”. In: Proceedings of the IEEE/CVF Con- ference on Computer Vision and Pattern Recognition. 2022, pp. 6114–6123. [142] S. H. Ahmed, H. W. Jang, S. N. Uddin, and Y. J. Jung. “Deep Event Stereo Lever- aged by Event-to-Image Translation”. In: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 35 1

. 2. 2021, pp. 882–890. [143]

> X. Chen, W. Weng, Y. Zhang, and Z. Xiong. “Depth From Asymmetric Frame- Event Stereo: A Divide-and-Conquer Approach 13

”.

> In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2024, pp. 3045–3054. [144] E. Mueggler, H. Rebecq, G. Gallego, T. Delbruck, and D. Scaramuzza. “The event-camera dataset and simulator: Event-based data for pose estimation, vi- sual odometry, and SLAM”. In: The International Journal of Robotics Research 36.2 (2017 45

), pp. 142–149. [145]

> J. Binas, D. Neil, S.-C. Liu, and T. Delbruck. “DDD17: End-to-end DAVIS driving dataset”. In: arXiv preprint arXiv:1711.01458 (2017 8

). [146]

> A. Sironi, M. Brambilla, N. Bourdis, X. Lagorce, and R. Benosman. “HATS: His- tograms of averaged time surfaces for robust event-based object classification 8

”.

> In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, pp. 1731–1740. [147] A. Z. Zhu, D. Thakur, T. Özaslan, B. Pfrommer, V. Kumar, and K. Daniilidis. “The multivehicle stereo event camera dataset: An event camera dataset for 3D perception”. In: IEEE Robotics and Automation Letters 3 40

.3 (2018), pp. 2032–2039. [148]

> C. Scheerlinck, H. Rebecq, T. Stoffregen, N. Barnes, R. Mahony, and D. Scara- muzza. “CED: Color event camera dataset”. In: Proceedings of the IEEE/CVF Con- ference on Computer Vision and Pattern Recognition Workshops. 2019, pp. 0–0. [149] H. Rebecq, R. Ranftl, V. Koltun, and D. 6

Scaramuzza. “High speed and high dy- namic range video with an event camera”. In: IEEE transactions on pattern analy- sis and machine intelligence

43.6 (2019), pp. 1964–1980. [150]

Y. Hu, J. Binas, D. Neil, S.-C. Liu, and T. Delbruck. “Ddd20 end-to-end event camera driving dataset: Fusing frames and events with deep learning for im- proved steering prediction”. In: 2020 IEEE 23rd International Conference on Intel- ligent Transportation Systems (ITSC). IEEE. 2020, pp. 1–6 8

. [151]

T. Fischer and M. Milford. “Event-based visual place recognition with ensem- bles of temporal windows”. In: IEEE Robotics and Automation Letters 5.4 (2020), pp. 6924–6931. [152] P 8

.

De Tournemire, D. Nitti, E. Perot, D. Migliore, and A. Sironi. “A large scale event-based detection dataset for automotive”. In: arXiv preprint arXiv:2001.08499 (2020). [153] M. Gehrig, W. Aarents, D. Gehrig, and D. Scaramuzza. “Dsec: A stereo event camera dataset for driving scenarios”. In: IEEE Robotics and Automation Letters 6.3 51

(2021), pp. 4947–4954. [154]

J. P. Rodríguez-Gómez, R. Tapia, J. L. Paneque, P. Grau, A. G. Eguíluz, J. R. Martínez-de Dios, and A. Ollero. “The GRIFFIN perception dataset: Bridging the gap between flapping-wing flight and robotic perception”. In: IEEE Robotics and Automation Letters 6.2 (2021), pp. 1066–1073 8

. [155] A. Zujevs, M. Pudzs, V. Osadcuks, A. Ardavs, M. Galauskis, and J. Grund- spenkis. “

An Event-based vision dataset for visual navigation tasks in 38

agricul- tural

environments”. In: 2021 IEEE International Conference on Robotics and Au- tomation (ICRA). IEEE. 2021, pp 12

. 13769–13775. [156] S.

Klenk, J. Chui, N. Demmel, and D. Cremers. “TUM-VIE: The TUM stereo visual-inertial event dataset”. In: 2021 IEEE/RSJ International Conference on Intel- ligent Robots and Systems (IROS). IEEE. 2021, pp. 8601–8608. [157] L. Gao, Y. Liang, J. Yang, S. Wu, C. Wang, J. Chen, and L. Kneip. “VECtor: A Versatile Event-Centric Benchmark for Multi-Sensor SLAM”. In: IEEE Robotics and Automation Letters (2022). [158] J. Jiao, H. Wei, T 6

.

Hu, X. Hu, Y. Zhu, Z. He, J. Wu, J. Yu, X. Xie, H. Huang, et al. "FusionPortable: A Multi-Sensor Campus-Scene Dataset for Evaluation of Localization and Mapping Accuracy on Diverse Platforms". In 6

: arXiv preprint arXiv:2208.11865 (2022). [159]

A. J. Lee, Y. Cho, Y.-s. Shin, A. Kim, and H. Myung. "ViViD++: Vision for Visibil- ity Dataset". In: IEEE Robotics and Automation Letters 7.3 (2022), pp. 6282–6289 8

. [160]

K. Chaney, F. Cladera, Z. Wang, A. Bisulco, M. A. Hsieh, C. Korpela, V. Ku- mar, C. J. Taylor, and K. Daniilidis. "M3ED: Multi-Robot, Multi-Sensor, Multi- Environment Event Dataset". In: Proceedings of the IEEE/CVF Conference on Com- puter Vision and Pattern Recognition. 2023, pp. 4015–4022 8

. [161]

G. Mollica, S. Felicioni, M. Legittimo, L. Meli, G. Costante, and P. Valigi. "MA- VIED: A Multisensor Automotive Visual Inertial Event Dataset". In: IEEE Trans- actions on Intelligent Transportation Systems (2023). [162] A. Hadviger, V.-J. Štironja, I. Cvišić, I. Marković, S. Vražić, and I. Petrović. "Stereo Visual Localization Dataset Featuring Event Cameras". In: 2023 European Con- ference on Mobile Robots (ECMR). IEEE. 2023, pp. 1–6 6

. [163] S. Zhu, Z. Xiong, and D. Kim. "

EAGLE: The First Event Camera Dataset Gath- ered by an Agile Quadruped Robot". In: arXiv 38

preprint arXiv:2404.04698 (2024). [164]

A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. "Vision meets robotics: The kitti dataset". In: The International Journal of Robotics Research 32.11 (2013), pp. 1231– 1237 8

. [165]

H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza. "High speed and high dy- namic range video with an event camera". In: IEEE transactions on pattern analy- sis and machine intelligence 23

(2019). [166]

J. Shi et al. "Good features to track". In: 1994 Proceedings of IEEE conference on computer vision and pattern recognition. IEEE. 1994, pp. 593–600 5

. [167]

E. Mueggler, C. Bartolozzi, and D. Scaramuzza. “Fast event-based corner detec- tion”. In: (2017). [168] I. Alzugaray and M. Chli. “Asynchronous corner detection and tracking for event cameras in real time”. In: IEEE Robotics and Automation Letters 11

3.4 (2018), [169]

X. Lagorce, G. Orchard, F. Galluppi, B. E. Shi, and R. B. Benosman. “Hots: a hier- archy of event-based time-surfaces for pattern recognition”. In: IEEE transactions on pattern analysis and machine intelligence 39.7 (2016), pp. 1346–1359. [170] D. Gehrig, A 26

.

Loquercio, K. G. Derpanis, and D. Scaramuzza. “End-to-end learn 25

- [171]

R. Benosman, C. Clercq, X. Lagorce, S.-H. Ieng, and C. Bartolozzi. “Event-based visual flow”. In: IEEE transactions on neural networks and learning systems 25.2 33

(2013), pp. 407–417. [172]

R. Li, D. Shi, Y. Zhang, K. Li, and R. Li. “Fa-harris: A fast and asynchronous corner detector for event cameras”. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE. 2019, pp. 6223–6229 39

. [173] S.

Lin, F. Xu, X. Wang, W. Yang, and L. Yu. “Efficient spatial-temporal normal- ization of sae representation for event camera”. In: IEEE Robotics and Automation Letters 32

5.3 (2020), pp. 4265–4272. [174] A. Gupta, P. Sharma, D. Ghosh, D. Ghose, and S. K. Muthukumar. “AeVIO: Asynchronous Event based

Visual Inertial Odometry”. In: 2023 IEEE Interna- tional Conference on Electronics, Computing and Communication Technologies (CONECCT). IEEE. 2023, pp 7

. 1–6. [175]

R. Hartley and A. Zisserman. Multiple view geometry in computer vision. Cam- bridge university press, 2003 55

. [176]

M. Calonder, V. Lepetit, C. Strecha, and P. Fua. “Brief: Binary robust 113

indepen

- [177]

T. Qin and S. Shen. “Robust initialization of monocular visual-inertial estimation on aerial robots”. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE. 2017, pp. 4225–4232 5

. [178]

P. Furgale, J. Rehder, and R. Siegwart. “Unified temporal and spatial calibration for multi-sensor systems”. In: 2013 IEEE/RSJ International Conference on Intelli- gent Robots and Systems. IEEE. 2013, pp. 1280–1286. [179] Z. Zhang and D. Scaramuzza. “A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry”. In: 2018 IEEE/RSJ International Conference on In- telligent Robots and Systems (IROS). IEEE. 2018, pp. 7244–7251 3

. [180]

J. H. Jung and C. G. Park. “Constrained filtering-based fusion of images, events 50

, [181]

I. Alzugaray and M. Chli. “Asynchronous multi-hypothesis tracking of features with event cameras”. In: 2019 International Conference on 3D Vision (3DV). IEEE. 2019, pp. 269–278 1

. [182]

M. Grupp. “evo: Python package for the evaluation of odometry and slam”. In: Note: https://github. com/MichaelGrupp/evo Cited by: Table 5

7 (2017). [183]

H. Lim, J. Jeon, and H. Myung. “UV-SLAM: Unconstrained line-based SLAM using vanishing points for structural mapping”. In: IEEE Robotics and Automa- tion Letters 92

7.2 (2022), pp. 1518–1525. [184]

A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer. “PL- SLAM: Real-time monocular visual SLAM with points and lines”. In: 2017 IEEE international conference on robotics and automation (ICRA). IEEE. 2017, pp. 4503– 4508. [185] B. Xu, P. Wang, Y. He, Y. Chen, Y. Chen, and M. Zhou. “Leveraging structural information to improve point line visual-inertial odometry”. In: IEEE Robotics and Automation Letters 4

7.2 (2022),

pp. 3483–3490. [186] Q. Fu, J. Wang, H. Yu, I. Ali, F. Guo, Y. He, and H. Zhang. "PI-vins: Real-time monocular visual-inertial slam with point and line features". In: arXiv preprint arXiv:2009.07462 (2020 4

). [187]

L. Pan, R. Hartley, C. Scheerlinck, M. Liu, X. Yu, and Y. Dai. "High frame rate video reconstruction based on an event camera". In: IEEE Transactions on Pattern Analysis and Machine Intelligence 44.5 (2020), pp. 2519–2533 25

. [188] S.

Tulyakov, D. Gehrig, S. Georgoulis, J. Erbach, M. Gehrig, Y. Li, and D. Scara- muzza. "Time lens: Event-based video frame interpolation". In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021, pp 25

. 16155– 16164. [189]

I. Alzugaray and M. Chli. "ACE: An efficient asynchronous corner tracker for event cameras". In: 2018 International Conference on 3D Vision (3DV). IEEE. 2018, pp. 653–661. [190] A. Z. Zhu, N. Atanasov, and K. Daniilidis. "Event-based feature tracking with probabilistic data association". In: 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE. 2017, pp. 4465–4470. [191] A. Dietsche, G. Cioffi, J. Hidalgo-Carrió, and D. Scaramuzza. "Powerline Track- ing with Event Cameras". In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE. 2021, pp. 6990–6997 5

. [192]

G. Chen, H. Cao, J. Conradt, H. Tang, F. Rohrbein, and A. Knoll. "Event-based neuromorphic vision for autonomous driving: A paradigm shift for bio-inspired visual sensing and perception". In: IEEE Signal Processing Magazine 4

37.4 (2020), pp. 34–49. [193]

V. Vasco, A. Glover, and C. Bartolozzi. "Fast event-based Harris corner detection exploiting the advantages of event-driven cameras". In: 2016 IEEE/RSJ interna- tional conference on intelligent robots and systems (IROS). IEEE. 2016, pp. 4144– 4149 12

. [194]

R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall. "LSD: A fast line segment detector with a false detection control". In: IEEE transactions on pattern analysis and machine intelligence 32.4 (2008), pp. 722–732 43

. [195]

D. Falanga, K. Kleber, and D. Scaramuzza. “Dynamic obstacle avoidance for quadrotors with event cameras”. In: Science Robotics 5.40 (2020), eaaz9712 — 8

. [196]

L. Zhang and R. Koch. “An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency”. In: Journal of Vi- sual Communication and Image Representation 24.7 (2013), pp. 794–805 — 43

. [197]

G. Zhang, J. H. Lee, J. Lim, and I. H. Suh. “Building a 3-D line-based map using stereo SLAM”. In: IEEE Transactions on Robotics — 5

31.6 (2015),

pp. 1364–1377. [198] A. BARTOLI and P. STURM. “The 3D Line Motion Matrix and Alignment of Line Reconstructions”. In: International Journal of Computer Vision 57 — 4

.3 (2004), pp. 159–178. [199]

T. Qin, J. Pan, S. Cao, and S. Shen. “A general optimization-based framework for local odometry estimation with multiple sensors”. In: arXiv preprint arXiv:1901.03638 (2019 — 8

). [200]

D. Mellinger and V. Kumar. “Minimum snap trajectory generation and control for quadrotors”. In: 2011 IEEE international conference on robotics and automation. [201] IEEE. 2011, pp. 2520–2525 — 5

.

K. Huang, S. Zhang, J. Zhang, and D. Tao. “Event-based Simultaneous Localiza- tion and Mapping: A Comprehensive Survey”. In: arXiv preprint arXiv:2304.09793 (2023). [202] Z. Zhang — 1

, Y. Song, S. Huang, R. Xiong, and Y. Wang. “Toward consistent and efficient map-based visual-inertial localization: Theory framework and filter de- sign”.

In: IEEE Transactions on Robotics 39.4 (2023), pp — 139

. 2892–2911. [203]

M. Mostafavi, K.-J. Yoon, and J. Choi. “Event-intensity stereo: Estimating depth by the best of both worlds”. In: Proceedings of the — 1

IEEE/CVF International Confer- ence on Computer Vision. 2021, pp. 4258–4267

. [204]

M. Gehrig, M. Millhäusler, D. Gehrig, and D. Scaramuzza. "E-raft: Dense optical flow from event cameras". In: 2021 International Conference on 3D Vision (3DV). IEEE. 2021, pp. 197–206 66

. [205]

W. Wang, D. Zhu, X. Wang, Y. Hu, Y. Qiu, C. Wang, Y. Hu, A. Kapoor, and S. Scherer. "Tartanair: A dataset to push the limits of visual slam". In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE. 2020, pp. 4909–4916 13

. [206]

H. Rebecq, D. Gehrig, and D. Scaramuzza. "ESIM: an open event camera simu- lator". In: Conference on robot learning. PMLR. 2018, pp. 969–982 26

. [207] Y. Zhou, X. Li, S. Li, X. Wang, S. Feng, and Y. Tan. "DBA-Fusion: Tightly Inte- grating Deep Dense Visual Bundle Adjustment with Multiple Sensors

for Large- Scale Localization and Mapping". In: IEEE Robotics and Automation Letters 127

(2024). [208]

R. Buchanan, V. Agrawal, M. Camurri, F. Dellaert, and M. Fallon. "Deep imu bias inference for robust visual-inertial odometry with factor graphs". In: IEEE Robotics and Automation Letters 77

8.1 (2022), pp. 41–48. [209]

R. T. Collins. "A space-sweep approach to true multi-image matching". In: Pro- ceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Ieee. 1996, pp. 358–363. [210] F. Ma, L. Carlone, U. Ayaz, and S. Karaman. "Sparse depth sensing for resource-constrained robots". In: The International Journal of Robotics Research 1

38.8 (2019), [211]

H.-T. Zhang, J. Yu, and Z.-F. Wang. "Probability contour guided depth map in- painting and superresolution using non-local total generalized variation". In: Multimedia Tools and Applications 1

77 (2018), pp. 9003–9020. [212]

A. Telea. "An image inpainting technique based on the fast marching method 1

". [213]

> **K. Wang, W. Ding, and S. Shen. "Quadtree-accelerated real-time monocular dense mapping". In: 2018 IEEE/RSJ International** Conference **on Intelligent Robots and** Sys- tems (**IROS). IEEE. 2018, pp. 1–9** 1

. [214]

> **H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto. "Voxblox: Incre- mental 3d euclidean signed distance fields for on-board mav planning". In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE**. [215] 2017, **pp. 1366–1373** 28

. intelrealsense.

> **https://www.intelrealsense.com/depth-camera-d455** 143

/. [216] Z. Taylor. A compact package for undistorting images directly from kalibr calibration files. https://github.com/ethz-asl/image_undistort. [217] S.-

> **H. Ieng, J. Carneiro, M. Osswald**, and **R. Benosman. "Neuromorphic event- based generalized time-based stereovision**". In: **Frontiers in neuroscience 12 (2018** 40

), p. 442. [218] H.

> **Hirschmuller. "Stereo processing by semiglobal matching and mutual in- formation**". In: **IEEE Transactions on pattern analysis and machine intelligence 30.2** 79

(2007), pp. 328–341. [219]

> **T. Müller, A. Evans, C. Schied, and A. Keller. "Instant neural graphics primitives with a multiresolution hash encoding". In: ACM transactions on graphics (TOG) 41.4 (2022), pp. 1–15** 24

. [220]

> **J. L. Schonberger and J.-M. Frahm. "Structure-from-motion revisited". In: Pro- ceedings of the IEEE conference on computer vision and pattern recognition. 2016, pp. 4104–4113**. [221] H. Wang, **J** 24

.

> **Wang, and L. Agapito. "Co-slam: Joint coordinate and sparse para- metric encodings for neural real-time slam** 55

".

> **In: Proceedings of the IEEE**/CVF Con- ference **on Computer Vision and Pattern Recognition**. 2023, **pp**. 13293–13302. [222] **Z. Zhu, S. Peng, V.** 49

Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys. “Nice-slam: Neural implicit scalable encoding for slam”. In: Proceed- ings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022

, [223]

O. Choi and S.-W. Jung. “A consensus-driven approach for structure and texture aware depth map upsampling”. In: IEEE transactions on image processing 1

23.8 (2014),

pp. 3321–3335. [224] S. Xiang, L. Yu, and C. W. Chen. “No-reference depth assessment based on edge misalignment errors for T+ D images”. In: IEEE Transactions on Image Processing 1

25.3 (2015), pp. 1479–1494. [225]

Z. Yan, W. Guan, S. Wen, L. Huang, and H. Song. “Multirobot cooperative local- ization based on visible light positioning and odometer”. In: IEEE Transactions on Instrumentation and Measurement 52

70 (2021), pp. 1–8. [226] W. Guan, L. Huang, S. Wen, Z. Yan, W. Liang, C. Yang, and Z. Liu. “Robot lo- calization and navigation using visible light positioning and SLAM fusion”. In: Journal of Lightwave Technology 39.22 (2021), pp. 7040–7051. [227]

D. Gehrig, M. Gehrig, J. Hidalgo-Carrió, and D. Scaramuzza. “Video to events: Recycling video datasets for event cameras”. In: Proceedings of the IEEE/CVF Con- ference on Computer Vision and Pattern Recognition. 2020, pp. 3586–3595. [228] Y. Ming, X. Meng, C. Fan, and 45

H. Yu. “Deep learning for monocular depth esti- mation: A review”. In: Neurocomputing 64

438 (2021), pp. 14–33. [229]

D. Eigen, C. Puhrsch, and R. Fergus. “Depth map prediction from a single image using a multi-scale deep network”. In: Advances in neural information processing systems 18

27 (2014).

3 4 Chapter 1. Introduction 5 6 Chapter 1. Introduction 7 8 Chapter 1. Introduction 9 10 Chapter 1. Introduction 24

1.2. Related Works 11 12 Chapter 1. Introduction 1.2. Related Works 13 14 Chapter 1. Introduction 58

1.2. Related Works 15 16

Chapter 1. Introduction 1.2. Related Works 80

17 18

Chapter 1. Introduction 1.2. Related Works 80

19 20

Chapter 1. Introduction 1.2. Related Works 80

21 22

Chapter 1. Introduction 1.2. Related Works 80

23 24 Chapter 1. Introduction 25 26 Chapter 1. Introduction 27 29 30 31 32 Chapter 2. Mono-EIO: Monocular Event-Inertial Odometry 33 34 Chapter 2. Mono-EIO: Monocular Event-Inertial Odometry 35 36 Chapter 2. Mono-EIO: Monocular Event-Inertial Odometry 37 38 Chapter 2. Mono-EIO: Monocular Event-Inertial Odometry 39 40 Chapter 2. Mono-EIO: Monocular Event-Inertial Odometry 41 42 Chapter 2. Mono-EIO: Monocular Event-Inertial Odometry 43 44 Chapter 2. Mono-EIO: Monocular Event-Inertial Odometry 45 46 Chapter 2. Mono-EIO: Monocular Event-Inertial Odometry 47 48 Chapter 2. Mono-EIO: Monocular Event-Inertial Odometry 49 50 Chapter 2. Mono-EIO: Monocular Event-Inertial Odometry 51 52 Chapter 2. Mono-EIO: Monocular Event-Inertial Odometry 53 54 Chapter 2. Mono-EIO: Monocular Event-Inertial Odometry 55 56 Chapter 2. Mono-EIO: Monocular Event-Inertial Odometry 57 58 59 60 Chapter 3. EVIO: Image-aided Event-based Visual-inertial Odometry 61 62 Chapter 3. EVIO: Image-aided Event-based Visual-inertial Odometry 63 64 Chapter 3. EVIO: Image-aided Event-based Visual-inertial Odometry 65 66 Chapter 3. EVIO: Image-aided Event-based Visual-inertial Odometry 67 68 Chapter 3. EVIO: Image-aided Event-based Visual-inertial Odometry 69 70 Chapter 3. EVIO: Image-aided Event-based Visual-inertial Odometry 71 72 Chapter 3. EVIO: Image-aided Event-based Visual-inertial Odometry 73 74 Chapter 3. EVIO: Image-aided Event-based Visual-inertial Odometry 75 76 Chapter 3. EVIO: Image-aided Event-based Visual-inertial Odometry 77 78 Chapter 3. EVIO: Image-aided

Event-based Visual-inertial Odometry 3 46

.9. Experiments 79 80 Chapter 3. EVIO: Image-aided

Event-based Visual-inertial Odometry 3 46

.9. Experiments 81 82 Chapter 3. EVIO: Image-aided

Event-based Visual-inertial Odometry 3 46

.9. Experiments 83 84 Chapter 3. EVIO: Image-aided

Event-based Visual-inertial Odometry 3 46

.9. Experiments 85 86 Chapter 3. EVIO: Image-aided

Event-based Visual-inertial Odometry 3 46

.9. Experiments 87 88 Chapter 3. EVIO: Image-aided

Event-based Visual-inertial Odometry 3 46

.9. Experiments 89 90 Chapter 3. EVIO: Image-aided

Event-based Visual-inertial Odometry 3 46

.9. Experiments 91 92 Chapter 3. EVIO: Image-aided Event-based Visual-inertial Odometry 93 94 Chapter 3. EVIO: Image-aided Event-based Visual-inertial Odometry 95 96 97

98 Chapter 4. Event-based Hybrid Odometry **99** 100 **Chapter 4. Event-based** 96

Hybrid Odometry 101

102 Chapter 4. Event-based Hybrid Odometry **103** 104 **Chapter 4. Event-based** 96

Hybrid Odometry 105 106 Chapter 4. Event-based Hybrid Odometry 4.8. Experiments 107 108 Chapter 4. Event-based Hybrid Odometry 4.8. Experiments 109 110 Chapter 4. Event-based Hybrid Odometry 4.8. Experiments 111 112 Chapter 4. Event-based Hybrid Odometry 113 114 115 116 Chapter 5. DEIO: Deep Learning-based Event-Inertial Odometry 117 118 Chapter 5. DEIO: Deep Learning-based Event-Inertial Odometry 119 120 Chapter 5. DEIO: Deep Learning-based Event-Inertial Odometry 121 122 Chapter 5. DEIO: Deep Learning-based Event-Inertial Odometry 123 124 Chapter 5. DEIO: Deep Learning-based Event-Inertial Odometry 125 126 Chapter 5. DEIO: Deep Learning-based Event-Inertial Odometry 127 128 Chapter 5. DEIO: Deep Learning-based Event-Inertial Odometry 129 130 Chapter 5. DEIO: Deep Learning-based Event-Inertial Odometry 5.8. Experiments 131 132 Chapter 5. DEIO: Deep Learning-based Event-Inertial Odometry 5.8. Experiments 133 134 Chapter 5. DEIO: Deep Learning-based Event-Inertial Odometry 5.8. Experiments 135 136 Chapter 5. DEIO: Deep Learning-based Event-Inertial Odometry 5.8. Experiments 137 138 Chapter 5. DEIO: Deep Learning-based Event-Inertial Odometry 5.8. Experiments 139 140 Chapter 5. DEIO: Deep Learning-based Event-Inertial Odometry 5.8. Experiments 141 142 Chapter 5. DEIO: Deep Learning-based Event-Inertial Odometry 5.8. Experiments 143 144 Chapter 5. DEIO: Deep Learning-based Event-Inertial Odometry 5.8. Experiments 145 146 Chapter 5. DEIO: Deep Learning-based Event-Inertial Odometry 5.8. Experiments 147 148 Chapter 5. DEIO: Deep Learning-based Event-Inertial Odometry 5.8. Experiments 149 150 Chapter 5. DEIO: Deep Learning-based Event-Inertial Odometry 151 152 Chapter 5. DEIO: Deep Learning-based Event-Inertial Odometry 153 154 155 156 Chapter 6. Event-based 3D Dense Reconstruction 157 158 Chapter 6. Event-based 3D Dense Reconstruction 159 160 Chapter 6. Event-based 3D Dense Reconstruction 161 162 Chapter 6. Event-based 3D Dense Reconstruction 163 164 Chapter 6. Event-based 3D Dense Reconstruction 165 166 Chapter 6. Event-based 3D Dense Reconstruction 6.7. Experiments 167 168 Chapter 6. Event-based 3D Dense Reconstruction 6.7. Experiments 169 170 Chapter 6. Event-based 3D Dense Reconstruction 6.7. Experiments 171 172 Chapter 6. Event-based 3D Dense Reconstruction 6.7. Experiments 173 174 Chapter 6. Event-based 3D Dense Reconstruction 6.7. Experiments 175 176 Chapter 6. Event-based 3D Dense Reconstruction 6.7. Experiments 177 178 Chapter 6. Event-based 3D Dense Reconstruction 6.7. Experiments 179 180 Chapter 6. Event-based 3D Dense Reconstruction 6.7. Experiments 181 182 Chapter 6. Event-based 3D Dense Reconstruction 183 185 186 187 188 189 191 192 193 194 Appendix A. HKU Dataset for Event Camera 195 196 Appendix A. HKU Dataset for Event Camera 197 198 Appendix A. HKU Dataset for Event Camera 199 200 201 202 Appendix B. Architecture of the Event-based Recurrent Network Appendix B. Architecture of the Event-based Recurrent Network 203 204 Appendix B. Architecture of the Event-based Recurrent Network Appendix B. Architecture of the Event-based Recurrent Network 205 206 Appendix B. Architecture of the Event-based Recurrent Network Appendix B. Architecture of the Event-based Recurrent Network 207 209 210 211 212 213 214 215 216 217 218 Bibliography Bibliography 219 220 Bibliography Bibliography 221 222 Bibliography Bibliography 223 224 Bibliography Bibliography 225 226 Bibliography Bibliography 227 228 Bibliography Bibliography 229 230 Bibliography Bibliography 231 232 Bibliography Bibliography 233 234 Bibliography Bibliography 235 236 Bibliography Bibliography 237