ISPRS Journal of Photogrammetry and Remote Sensing 223 (2025) 328–343

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

Hybrid depth-event pose estimation for online dense reconstruction in challenging conditions

Guohua Gou, Xuanhao Wang, Yang Ye, Han Li, Hao Zhang, Weicheng Jiang, Mingting Zhou, Haigang Sui *

State Key Laboratory of Information Engineering in Surveying Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China

ARTICLE INFO

Keywords:
depth-event fusion SLAM
Online dense reconstruction
Hybrid pose estimation
Random optimization
Visual degradation camera tracking

ABSTRACT

In this paper, we present a novel dense SLAM system based on depth-event fusion, aiming to address the challenge of online dense reconstruction in challenging environments. To achieve robust camera tracking, we devise a hybrid depth-event pose estimation framework based on random optimization, which estimates all states jointly. Notably, we introduce an innovative 3D-2D edge alignment method based on particle swarm optimization, specifically tailored for event cameras, to tackle the highly non-linear pose estimation problem. Furthermore, we implement a dynamic update mechanism for both geometric and intensity edges of the 3D reconstruction, enabling efficient and accurate management of edge information. Our method represents the first depth-event dense SLAM system employing a random optimization paradigm, achieving robust performance even with high-speed camera motion, specifically linear velocities exceeding 1 m/s and/or angular velocities exceeding 2 rad/s. The system achieves accurate and globally consistent dense mapping with a maximum spatial resolution of 2 mm, while maintaining real-time performance at approximately 30 FPS for simultaneous localization and 3D reconstruction. Through extensive evaluations on synthetic and real-world datasets, particularly on our newly constructed DEveSet dataset, we demonstrate the superior performance of our proposed method compared to state-of-the-art techniques such as InfiniTAM, ROSEFusion, and DEVO. Contact us for access to the DEveSet download link.

1. Introduction

Real-time 3D mapping in unknown enclosed environments by intelligent mobile systems (e.g., unmanned aerial vehicles, UAV) is crucial for tasks such as emergency response (Boguslawski et al., 2022; Robin R. Murphy, 2021), resource exploration (Biggie et al., 2023; Petrá \vcek et al., 2021), and archaeological surveys (Cao et al., 2023; Xia and Dong, 2023). However, fast-moving mobile platforms often encounter challenges such as weak texture, low illumination, and motion blur in environments like subterranean enclosed spaces, which degrade the robustness of localization systems (Gou et al., 2023; Zuo et al., 2024). Existing research (Agha et al., 2021; Azpúrua et al., 2022; Hudson et al., 2021; Tranzatto et al., 2022a, 2022b) typically addresses these challenges by integrating sensors like cameras, LiDAR, or inertial measurement units (IMU). However, considering portability, cameras are often the preferred sensors for external perception. Therefore, exploring vision-only solutions in challenging conditions remains a significant research direction.

Among vision-only solutions, depth-based methods (Newcombe et al., 2011; Prisacariu et al., 2017; Whelan et al., 2016) are commonly employed for simultaneous localization and mapping (SLAM) in low-light environments. This is because RGB images acquired in such conditions often fail to provide the environmental information necessary for traditional RGB-based SLAM methods, as illustrated in Fig. 1. However, depth-based methods still face challenges concerning robustness and localization accuracy when dealing with fast camera motion (Zhang et al., 2021) and degradation of scene geometry (Gou et al., 2023). Consequently, researchers have begun to explore methods that incorporate other specialized vision sensors. Event cameras (also known as dynamic vision sensors, DVS), with their high dynamic range and temporal resolution, offer a promising alternative. They can robustly capture geometric and intensity edges in challenging conditions, remaining unaffected by motion blur (Gallego et al., 2019). This complements the limitations of depth cameras in perceiving intensity features, as shown

* Corresponding author.
E-mail address: haigang_sui@263.net (H. Sui).

https://doi.org/10.1016/j.isprsjprs.2025.03.013
Received 3 December 2024; Received in revised form 12 March 2025; Accepted 13 March 2025
Available online 25 March 2025
0924-2716/© 2025 International Society for Photogrammetry and Remote Sensing, Inc. (ISPRS). Published by Elsevier B.V. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

in Fig. 1, further supporting the feasibility of depth-event fusion for addressing localization and mapping in challenging scenarios. However, since event cameras only output intensity change information, photo-consistency-based fusion methods (Bryner et al., 2019) cannot be directly applied, and feature extraction (Messikommer et al., 2022) remains unstable. Furthermore, the lack of rich intensity features and photometric information in depth images prevents some event-based fusion methods (Bryner et al., 2019; Gehrig et al., 2018) designed for conventional RGB-event data from being directly applied to depth-event fusion. Additionally, the limited spatial resolution and low signal-to-noise ratio of event cameras also pose challenges for constructing high-density, high-quality 3D maps. A recent study (Zuo et al., 2024, 2022) achieved reliable pose estimation based on semi-dense map geometry through a 2D-3D registration by directly optimizing the edges generated from both event and depth images. However, fast camera motion can lead to large rotations, making camera pose optimization highly non-linear and prone to trapping gradient descent methods in local optima (Schmidt and Niemann, 2001). This makes robust pose estimation in challenging conditions difficult, which is crucial for dense reconstruction.

This paper presents **DEveFusion**, a novel **De**pth-**Eve**nt **Fusion** approach for robust cross-modal pose estimation, enabling online dense 3D reconstruction in challenging conditions. We leverage depth images to construct a global dense volumetric map and dynamically maintain a local active map within a view frustum. Upon acquiring a new depth-event data pair, we perform a weighted update of the active map, ensuring the global consistency of the reconstructed map. Camera pose is optimized by jointly minimizing the distances between depth image projections and reconstructed voxels in the active map, and between 3D edges in the active map and their 2D projections onto the event image. Specifically, our contributions are as follows:

- We develop a unified hybrid depth-event pose estimation framework based on random optimization, where all states are estimated jointly without initialization.
- We propose a novel 3D-2D edge alignment method based on random particle swarm optimization to address the highly non-linear pose estimation problem, achieving accurate pose estimation while maintaining high computational efficiency.
- We introduce a surface edge update and extraction strategy that dynamically updates both geometric and intensity edges of the 3D reconstructed surface. This strategy enables fast and accurate edge extraction from the dense reconstruction without traversing the entire map.

The remainder of this paper is structured as follows: Section 2 reviews related work. Section 3 introduces preliminaries, including the overall framework and depth-event data pre-processing. Section 4 elaborates on the scene representation and reconstruction process. Section 5 details the random optimization-based pose estimation method. Section 6 presents evaluation experiments and results. Finally, Section 7 concludes this work. This paper represents a significant extension of our previous work (Gou et al., 2023).

2. Related work

In this section, we review related work in the field of visual SLAM (vSLAM) that utilizes the fusion of RGB-D cameras and event cameras. We will provide separate overviews of RGB-D camera-based SLAM, event camera-based SLAM, and their fusion methods, followed by a discussion of the current issues and challenges within this field.

2.1. RGB-D camera-based SLAM

Methods fusing depth and RGB images typically employ dense

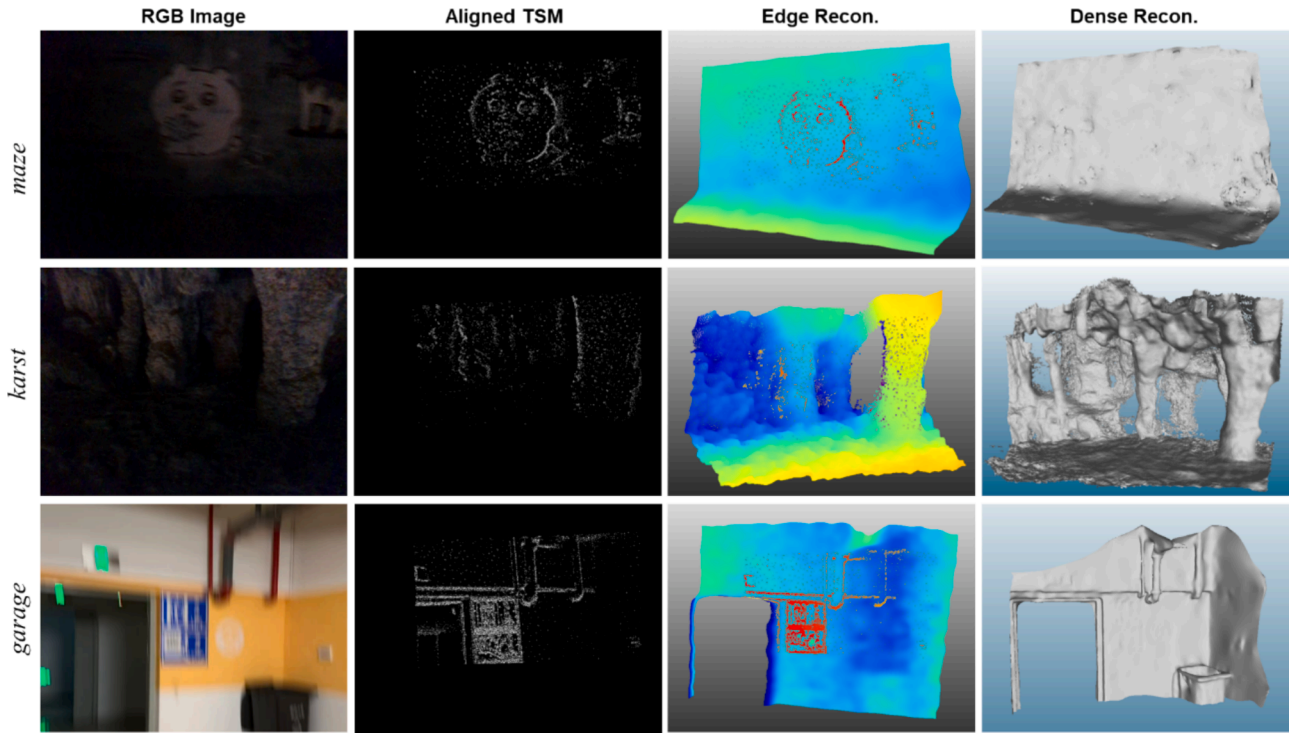

Fig. 1. Challenging scenarios and results. The first column demonstrates challenging illumination conditions in the maze and karst sequences, with slight motion blur also visible in the maze sequence. The garage sequence exhibits pronounced motion blur due to fast camera motion under indoor illumination. The second column shows the event camera's time surface map (TSM), aligned with the RGB-D images after extrinsic calibration, which highlights the event camera's ability to capture edges even in low-light environments. The third column illustrates the edge reconstruction of the scene, encompassing both geometric edges and intensity edges on the 3D support plane. The fourth column presents the dense reconstruction within the current view frustum.

photometric alignment (Endres et al., 2014; Kerl et al., 2013; Steinbrücker et al., 2011), a process that relies on extensive parallel computation. Moreover, such methods often suffer performance degradation in challenging visual conditions, such as low-light environments and/or fast camera motion, which can lead to motion blur or imaging failures in RGB images (Dai et al., 2016; Mur-Artal et al., 2015; Mur-Artal and Tardós, 2016). Due to the active imaging advantages of depth images, depth-only methods (Newcombe et al., 2011; Whelan et al., 2016) demonstrate potential in dark environments. These methods generally employ gradient descent to directly solve for local optima in pose estimation (Bylow et al., 2013; Dai et al., 2016), making them prone to tracking failures when fast camera motion leads to highly non-linear pose optimization. Zhang et al. (2021) exploited the fact that depth images are less affected by fast camera motion (they do not exhibit motion blur like RGB images) and achieved tracking of fast-moving cameras in lightless environments through random particle swarm optimization for pose estimation. However, this method requires processing high frame-rate dense depth images, and despite the authors' design of particle swarm templates (PST) to accelerate particle sampling, it still demands substantial computational resources and energy consumption. Building upon this, Gou et al. (2023) devised a planar-constrained particle swarm optimization method for pose estimation, achieving tracking of high-speed camera motion on resource-constrained UAV onboard computing devices and extending the scope of scene reconstruction. Their surface reconstruction module is similar to the one presented in this paper. Despite the initial success of depth-based methods in tracking and mapping applications within challenging conditions, some limitations remain. Firstly, as existing depth-only methods rely entirely on geometric information, they inevitably fail when consecutive frames lack distinctive geometric features (Gou et al., 2023; Zhang et al., 2021). Secondly, depth-based methods become ineffective when depth information is severely missing due to light reflection or absorption. Lastly, tracking failures can also occur under extremely high-speed motion (e.g., ~5m/s in (Zhang et al., 2021)).

2.2. Event-based SLAM

Event cameras, with their potential to overcome the inherent limitations of frame-based cameras in challenging conditions such as low-light and fast camera motion, have garnered significant research interest in the field of vSLAM (Huang et al., 2023). Most event-based vSLAM systems follow the basic workflow of PTAM (Klein and Murray, 2007), consisting of two main components: camera tracking and mapping, which are responsible for estimating camera pose and reconstructing the 3D scene map, respectively. However, achieving 6 degrees of freedom (DoF) pose estimation using event cameras remains a challenging problem. Before tackling general 6-DoF pose estimation, researchers extensively investigated constrained pose estimation scenarios, such as pure rotation (Cook et al., 2011; Gallego and Scaramuzza, 2017; Kim et al., 2014; Reinbacher et al., 2017) and planar motion (Liu et al., 2020; Peng et al., 2021; Weikersdorfer et al., 2013; Weikersdorfer and Conradt, 2012) (up to 3-DoF). Kim et al. (2016) proposed a sophisticated framework comprising three decoupled probabilistic filters to estimate intensity, depth, and pose, respectively, representing the first complete 6-DoF solution. Subsequently, a series of geometry-based solutions (Rebecq et al., 2018, 2017a; Wang et al., 2021) emerged, leveraging geometric features in the scene for pose estimation without relying on intensity estimation. To fully exploit the temporal resolution advantage of event cameras, probability filter-based pose optimization methods (Gallego et al., 2016; Kim et al., 2014; Weikersdorfer and Conradt, 2012) have achieved minimal latency, albeit with complex event representations. In contrast, frame-based techniques (Bryner et al., 2019; Gallego and Scaramuzza, 2017) typically achieve more accurate results through non-linear optimization at the expense of latency and susceptibility to local optima. Our proposed method employs a frame-based filtering approach (particle filtering), enabling globally optimal pose estimation while balancing time efficiency and simplicity of event representation. The aforementioned methods are all based on single event camera setups. Zhou et al. (2020) released the first stereo visual odometry algorithm for event cameras, ESVO, which provided inspiration for the design of our tracking module.

The mapping module in vSLAM aims to reconstruct a sparse, semi-dense, or dense 3D map. Generally, event-based vSLAM methods consider the 3D map generated from all event data as a semi-dense scene map (Rebecq et al., 2017a; Zuo et al., 2022). However, some methods (Guan et al., 2022; Kueng et al., 2016) opt to construct efficient sparse maps using a subset of events to improve time efficiency and accuracy. Since the 2D perspective of an event camera inherently lacks complete 3D information about the world scene, event-based vSLAM algorithms typically cannot generate dense 3D maps (Huang et al., 2023). Overall, the map information obtained solely from events is usually of poor quality, motivating our proposed hybrid depth-event dense reconstruction method.

2.3. Hybrid camera-event solutions

Due to the complexity of event data, event cameras are often used in conjunction with other sensors in vSLAM, such as IMUs (Gentil et al., 2020; Mueggler et al., 2017b; Zhu et al., 2017), standard cameras, or depth cameras. This paper focuses solely on the combination of event cameras and depth cameras. Since intensity images (i.e., RGB images) provided by standard cameras contain rich static features, while event cameras offer high temporal resolution, an intuitive approach is to extract features from intensity images and track them using event data (Gehrig et al., 2018; Kueng et al., 2016; Tedaldi et al., 2016). Commonly used features include point features (Alzugaray and Chli, 2018; Li et al., 2019; Mueggler et al., 2017a; Vasco et al., 2016) and line features (Chamorro et al., 2022; Everding and Conradt, 2018; Guan et al., 2022; Valeiras et al., 2019). However, methods relying on intensity images as the core do not fully leverage event data and may fail due to motion blur. To address the challenge of insufficient effective feature extraction caused by image blur, several methods that process images and events separately have been proposed (Mahlknecht et al., 2022; Rebecq et al., 2017b; Vidal et al., 2017). However, these methods often lead to inaccurate local event maps and severe robustness issues if adverse visual conditions persist for extended periods or in environments where intensity images are unavailable (e.g., dark environments). Weikersdorfer et al. (2014) extended their previous work (Weikersdorfer et al., 2013) by incorporating depth information to adapt to extreme conditions. However, their method is based on an outdated low-resolution event camera model, operates within limited scene sizes, and cannot generate dense 3D maps. A recent work introduced DEVO (Zuo et al., 2024, 2022), a visual odometry algorithm combining a high-resolution event camera and a depth camera to accurately construct semi-dense 3D local maps. Due to the highly non-linear nature of the problem caused by large rotations resulting from fast camera motion, the gradient descent method used by DEVO to directly optimize the equations struggles to perform well in such conditions. Furthermore, direct optimization methods can only achieve local optima (Zhou et al., 2020). Additionally, DEVO estimates pose by merging multiple local maps into a global map, which can accumulate errors over time. To the best of our knowledge, our work represents the first depth-event dense SLAM method based on a random optimization framework.

3. Preliminaries

This section provides an overview of our proposed hybrid depth-event dense SLAM framework based on random optimization, followed by a detailed description of the pre-processing steps and methods required for both depth and event data.

3.1. Framework overview

The proposed **DEveFusion** comprises three main components: data preparation, surface reconstruction, and pose estimation. Its workflow is illustrated in Fig. 2. During the data preparation stage, the input event stream is represented as a time surface map (TSM) for efficient and accurate edge extraction and alignment. Concurrently, surface measurements are performed on the input depth image to obtain spatial positions and structural information for 2D pixels, which are then aligned with the TSM edges, denoted as $\mathscr{E}$. Details of data preparation are presented in Sections 3.2 and 3.3, respectively. In the surface reconstruction stage, a hierarchical voxel map, $\mathscr{S}$, representing the scene is reconstructed using the vertices map with edge information (VME), denoted as $\mathscr{V}_e$, obtained from the surface measurements. Scene edge information is stored within voxels and dynamically updated during subsequent processes, as detailed in Section 4. Surface reconstruction relies on a stable and accurate camera pose, $\mathbf{s}^*$, thus pose estimation runs in parallel with surface reconstruction. The pose estimation module takes the partition normal map (PNM), $\mathscr{P}$, and $\mathscr{V}_e$ from surface measurements, as well as the reconstructed scene map, $\mathscr{S}$, and its 3D edges, $\mathscr{S}_e$, as inputs. Tracking is performed by constructing a truncated signed distance field (TSDF) (cf. Section 5.2) and a time surface distance field (tsDF) (cf. Section 5.3) within the current visible region. Based on a pre-sampled particle swarm, we intelligently sample and evaluate particle energy values within TSDF and tsDF. During the optimization process, we minimize the particle energy within the joint TSDF and tsDF field (cf. Section 5.4) to progressively converge towards the optimal solution, thereby achieving pose optimization. Finally, we extract the dense scene reconstruction result from the global voxel map.

3.2. Time surface maps

We represent an event mathematically as $e_i = [\mathbf{x}_i, \quad t_i, \quad p_i]^T, i \in \mathbb{N}$, where e_i denotes an event with polarity t_i occurring at location $\mathbf{x}_i$ at time p_i. Since individual event data carries limited information and is susceptible to noise, events are often transformed into alternative representations to aggregate meaningful information for vSLAM, thereby improving the signal-to-noise ratio (Huang et al., 2023). Within regular time intervals, events are commonly aggregated into three potential representations: voxel grids (Mostafavi et al., 2021; Zhu et al., 2018b), event frames (Rebecq et al., 2017a; Ye et al., 2020), and time surface maps (TSM) (Lagorce et al., 2017). In this paper, we adopt the TSM representation for events to efficiently and accurately perceive edge information within the scene.

A TSM, $\mathscr{T}(\mathbf{x}, t)$, is a 2D image, I_e, where the value at each pixel location, $\mathbf{x}$, is determined by an exponential decay function:

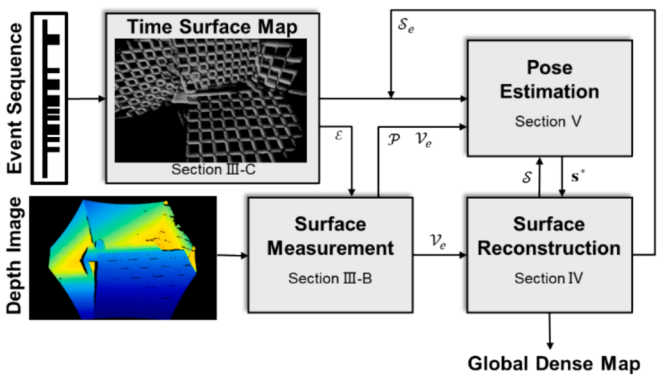

Fig. 2. Overview of our proposed simultaneous localization and dense mapping (Dense SLAM) pipeline.

$$T(\mathbf{x}, t) = \exp\left(-\frac{t - t_{last}(\mathbf{x})}{\tau} \right), \# \tag{1}$$

where t represents the timestamp at any given moment, and $t_{last}(\mathbf{x}) \le t$ represents the timestamp of the last triggered event at location $\mathbf{x}$. A larger $\mathscr{T}(\mathbf{x}, t)$ value indicates a more recent event, represented by brighter pixels during visualization (as shown in Fig. 1), thereby emphasizing locations with recent motion. τ is a decay rate constant determined empirically in experiments.

3.3. Depth surface measurement

Similar to our previous work, the purpose of surface measurement is to pre-process the input depth image, I_d, and generate the vertices map, $\mathscr{V}$, normal map, $\mathscr{N}$, and partition normal map, $\mathscr{P}$, required for subsequent steps (cf. (Gou et al., 2023) Section 3.1 for a detailed explanation). In this paper, we introduce a crucial addition by incorporating edge information to guide map reconstruction and camera tracking. Therefore, we propose the construction of a vertices map with edge information (VME).

We build the VME for the current depth frame in two steps. First, let $\mathscr{T}_{dep}(\bullet) = \mathscr{T}(\bullet, t_{dep})$ be the TSM generated from the event stream at the exposure time of the depth image, I_d. We select $\mathscr{E}_{dep} = \{\mathbf{x} s.t. \ \mathscr{T}_{dep}(\mathbf{x}) > \delta\}$ as edge pixels, based on the assumption that high-gradient edges in the TSM are more likely to trigger events. δ is a threshold constant, and balancing τ and δ helps prevent the extracted edges from being too thin and ensures close alignment with the true edges. Then, we warp the pixels in the depth map, I_d, to $\mathscr{E}_{dep}$ using a simple reprojection formula(Gou et al., 2025; Zuo et al., 2024), as follows:

$$\mathbf{x}^e = \mathbf{K}_e \mathbf{T} \bullet \mathrm{I}_d(\mathbf{x}^d) \bullet \mathbf{K}_d^{-1} \mathbf{x}^d, \# \tag{2}$$

where $\mathbf{K}_{e/d}$ are the pre-calibrated intrinsic matrices of the event and depth cameras, respectively. The transformation matrix, $\mathbf{T}$, between the two sensors is obtained through extrinsic calibration of the event and depth cameras. $\mathbf{x}^d$ represents an arbitrary point in the depth map, $\mathrm{I}_d(\mathbf{x}^d)$ is its corresponding depth value, and $\mathbf{x}^e$ is its corresponding point on the TSM. We determine whether $\mathbf{x}^e$ belongs to the pixel set $\mathscr{E}_{dep}$. If so, the corresponding depth pixel, $\mathbf{x}^d$, is marked as an edge point on the vertices map; otherwise, no action is taken. Finally, we obtain a vertices map with edge information, $\mathscr{V}_e$.

4. Surface reconstruction

Unlike other event-based mapping methods that acquire sparse or semi-dense scene point clouds, our hybrid depth-event module reconstructs a dense 3D map. To improve algorithm efficiency and conserve graphics processing unit (GPU) memory, we construct a hierarchical sparse representation for the scene surface and maintain a dynamic submap within the active space (c.f. Section 4.1). Furthermore, to ensure consistent edge reconstruction, we propose an edge-weighted reconstruction and fusion method (c.f. Section 4.2).

4.1. Surface representation

We construct a multi-resolution level voxel representation for the scene based on surface roughness, utilizing a hash table for efficient access to voxels at each level. Initially, we divide the voxel size into three resolution levels, L_0, L_1, and L_2, from coarse to fine, assigning different levels of voxels within the volume for depth measurements from the depth image. Notably, if edge measurements from the VME are assigned to the coarsest level, L_0, it indicates that these edge details cannot be captured by the depth camera (e.g., graffiti on a wall). Prioritizing these points for pose estimation enhances system robustness, as discussed in

detail in Section 5.4. If a voxel corresponding to a depth measurement in the current frame has already been allocated, its data is globally fused with historical data, as detailed in Section 4.2. After voxel memory allocation, we quantitatively assess the roughness of all surface voxels to determine whether they need to be subdivided to a finer level or merged into a coarser level.

We employ an efficient bi-directional GPU-CPU data stream scheme (Kähler et al., 2015; Nießner et al., 2013) to manage the reconstructed volume. We introduce the concept of "active space," which refers to the map space within the field of view of the current camera. The reconstruction results within the active space are referred to as the view frustum dense map (VFDM), as illustrated in Fig. 3. During system operation, we dynamically transfer voxels in the inactive space to CPU memory. When these voxels reappear within the active space, they are reloaded into GPU memory.

4.2. Surface fusion

For voxels observed multiple times by different frames, we fuse and update their internally stored values to maintain global consistency in the reconstruction. In previous voxel-based dense reconstruction systems, voxel internal parameters typically only include the TSDF value, F, and its weight, W_f (Newcombe et al., 2011). Unlike previous methods, we introduces a novel approach by incorporating edge attributes, E, and their weights, W_e, as voxel parameters, enabling them to participate throughout the scene surface reconstruction and update process to effectively leverage edge information from events. If a spatial point, $\mathbf{p}$, is observed as an edge point in the $\boldsymbol{k}$-th frame, $E_k(\mathbf{p})$ is set to "true" and remains unchanged in subsequent updates. For the edge point weight, $W_e^k(\mathbf{p})$, we define an efficient update method:

$$W_e^k(\mathbf{p}) = W_e^{k-1}(\mathbf{p}) + W_e^{R_k}(\mathbf{p}), \# \tag{3}$$

where we simply set $W_e^{R_k}(\mathbf{p}) = 1$, and $W_e^{k-1}(\mathbf{p})$ represents the weight from the last time $\mathbf{p}$ appeared in the VFDM. In practice, we observe that due to minor errors in edge extraction, the same 3D point might be identified as an edge point in some frames' VFDMs but not in others. Our proposed edge weight update method effectively handles this situation. During particle adaptability evaluation (c.f. Section 5.4), we prioritize edge points with larger weights, minimizing the adverse effects of edge extraction errors without incurring additional computational costs. We utilize the method from our previous work (Gou et al., 2023) to update F and W_f.

5. Pose estimation

This section defines the pose estimation problem based on particle swarm optimization (c.f. Section 5.1) and describes the construction of the TSDF (c.f. Section 5.2) and tsDF (c.f. Section 5.3) for particle filtering. In Section 5.4, we further elaborate on how to combine the energy functions of the TSDF and tsDF to guide the particle swarm covers the optimal solution, thus achieving pose optimization.

5.1. Problem statement

Building upon the VFDM generated by the surface reconstruction module, we can now elaborate on the 6-DoF pose estimation module. We employ particle swarm optimization (PSO) for pose estimation. PSO is a random optimization method that transforms the objective function into a target probability density function (PDF). It then simulates the PDF by randomly generating a set of particles through sequential importance sampling and utilizes swarm intelligence to guide their movement, ultimately aiming for the sampled particles to cover the optimal value of the objective function. Since particle swarm sampling and updating are computationally expensive, we utilize a PST for sampling and updating. Notably, the core of PSO lies in designing an effective system dynamic function to drive the particle swarm towards favorable local optima.

Given an aligned depth image, I_d^t, and a time surface map, I_e^t, at time t, as well as the current VFDM, $\mathscr{S}^t$, our task is to compute the 6-DoF camera pose, $[\mathbf{R}^t|\mathbf{t}^t] \in SE(3)$, of I_d^t and I_e^t in the global coordinate system, where $\mathbf{R} \in SO(3)$ and $\mathbf{t} \in \mathbb{R}^3$ represent the 3D rotation and translation in the global coordinate system, respectively. In PSO, we define the state as the camera pose, $\mathbf{s} = (\mathbf{R}, \mathbf{t}) = \left(q_x, q_y, q_z, x, y, z\right)$, where $\left(q_x, q_y, q_z\right)$ are the imaginary parts of the rotation quaternion, and $\mathbf{t} = (x, y, z)^T$. We pre-sample a fixed number of particles uniformly in the 6D state space, referred to as the particle swarm template (PST) (Zhang et al., 2021), denoted by Ω. Our objective is to find the optimal camera pose, $(\mathbf{R}^*, \mathbf{t}^*) \in \Omega$, that maximizes the likelihood of observing I_d^t and I_e^t:

$$(\mathbf{R}^*, \mathbf{t}^*) = \underset{\mathbf{R},\mathbf{t}}{\operatorname{argmax}} (I_d^t, I_e^t|\mathbf{R}, \mathbf{t}). \# \tag{4}$$

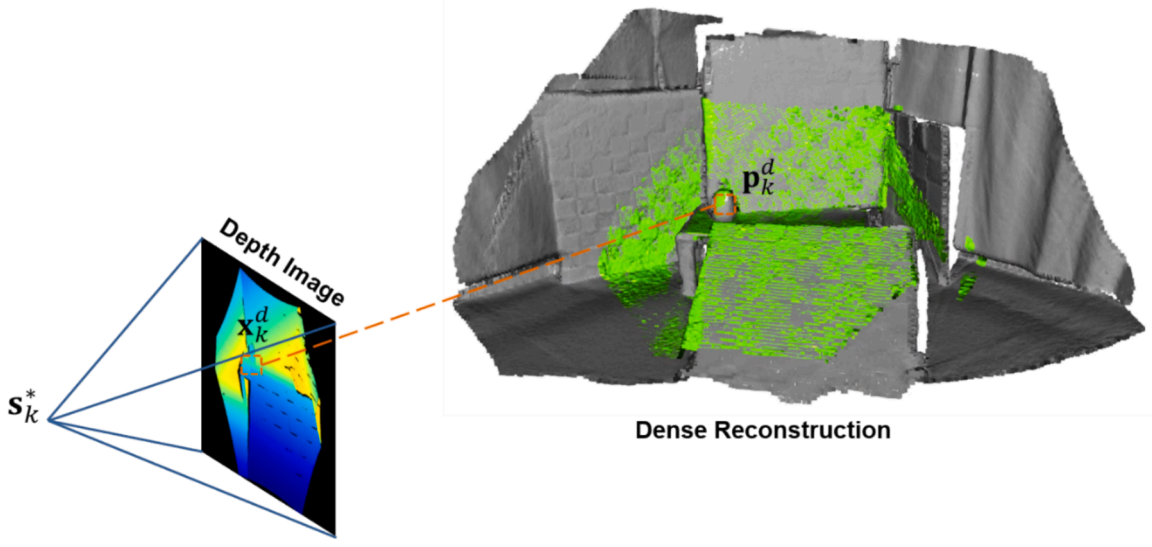

Fig. 3. Illustration of the view frustum dense map (VFDM). The green area is the VFDM of the $\boldsymbol{k}$-th depth frame, which represents the dense reconstruction visible within the field of view of the current depth image. The orange box represents the dense reconstruction of the corresponding pixel area on the depth image. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

To achieve this, we define a new particle fitness evaluation function, $\mathscr{F}(\mathbf{s}_t^i)$, also known as the system dynamic function, which assesses the fitness of I_d^t and I_e^t to $\mathscr{S}^t$ under the i-th particle state in Ω. This function drives the PST's movement and scaling, ultimately leading to the progressive coverage of the optimal solution. $\mathscr{F}(\mathbf{s}_t^i)$ is defined as:

$$F(\mathbf{s}_t^i) = \lambda_1 \mathscr{G}_{TSDF} + \lambda_2 \mathscr{G}_{tsDF}, \# \tag{5}$$

where the two energy terms, $\mathscr{G}_{TSDF}$ and $\mathscr{G}_{tsDF}$, represent the TSDF and tsDF, respectively. The parameters λ_1 and λ_2 are system adaptive parameters that can be dynamically adjusted according to changes in the scene without manual intervention, as detailed in Section 5.4.

5.2. Particle fitness by TSDF

Our TSDF energy function for particle fitness evaluation is based on a depth frame-to-model error metric. This method assesses the quality of a particle, i, represented by its pose state, $[\mathbf{R}^i|\mathbf{t}^i]$, by measuring the alignment between I_d and the TSDF within the VFDM. For each pixel, $\mathbf{x}_{uv}^d$, in I_d with a depth value of $I_d(\mathbf{x}_{uv}^d)$, we can obtain its corresponding 3D point, $\mathbf{P}_{uv}^d$, in the current frame's camera coordinate system:

$$\mathbf{P}_{uv}^d = I_d(\mathbf{x}_{uv}^d) \bullet \mathbf{K}_d^{-1} \mathbf{x}_{uv}^d. \# \tag{6}$$

In the surface measurement module, we store the value of $\mathbf{P}_{uv}^d$ in the corresponding pixel of the vertices map, $\mathscr{V}$. Then, we transform the points in $\mathscr{V}$ to the global coordinate system based on $[\mathbf{R}^i|\mathbf{t}^i]$:

$$\mathbf{P}_{uv}^g = \mathbf{R}^i \mathbf{P}_{uv}^d + \mathbf{t}^i. \# \tag{7}$$

Next, we use these projected points to query the TSDF values, $\psi(\mathbf{P}_{uv}^g)$, in the VFDM. If the camera pose is correct, the sum of all projected points' TSDF values should be close to zero, as the TSDF represents the distance from a spatial point to the scene surface (Newcombe et al., 2011). Therefore, our objective can be transformed into finding the camera pose, $[\mathbf{R}^*|\mathbf{t}^*]$, that minimizes the distance between each 3D point in $\mathscr{V}$ and the zero-crossing surface of the TSDF:

$$(\mathbf{R}^*, \mathbf{t}^*) = \underset{\mathbf{R},\mathbf{t}}{\operatorname{argmin}} \sum\nolimits_{(u,v)\in\mathscr{V}} \psi(\mathbf{P}_{uv}^g)^2. \# \tag{8}$$

In implementation, unlike previous methods, we select edge pixels from the projected vertices map with edge information, $\mathscr{V}_e$, to reduce the computational burden of fitness evaluation, which is typically the most time-consuming part of random optimization. To avoid erroneous low overall TSDF values caused by an insufficient number of valid projected points, we normalize the sum of TSDF values:

$$(\mathbf{R}^*, \mathbf{t}^*) = \underset{\mathbf{R},\mathbf{t}}{\operatorname{argmin}} \frac{\sum_{(u,v)\in\mathscr{V}_e} \psi(\mathbf{P}_{uv}^g)^2}{|O_d^e|}, \# \tag{9}$$

where O_d^e is the set of valid edge 3D points in $\mathscr{V}_e$ that participate in fitness evaluation, i.e., the edge pixels that overlap between the current and previous depth frames. Our optimization objective is to minimize Eq. (9). Therefore, we define the objective function, $\mathscr{G}_{TSDF}$, within the TSDF as:

$$\mathscr{G}_{TSDF} = \exp\left(- \frac{\sum_{(u,v)\in O_d^e} \psi(\mathbf{R}^i \mathbf{P}_{uv}^d + \mathbf{t}^i)^2}{|O_d^e|} \right). \# \tag{10}$$

5.3. Particle fitness by tsDF

Building on the geometric 3D-2D registration method for maps and events (Zhou et al., 2020), we define a novel model-to-frame error metric based on the tsDF. This error metric is utilized in random optimization to evaluate the fitness of particle states, representing one of the most significant contributions of this work. This metric assesses the alignment between edges in I_e and model edges under the state $[\mathbf{R}^i|\mathbf{t}^i]$. Unlike previous methods that rely on aligning semi-dense maps generated from edge depths with events (Zuo et al., 2024), our approach can accurately and efficiently extract 3D edge points from an arbitrary voxel map to establish correspondences with events, making it suitable for dense reconstruction.

The TSM encodes the motion history of scene edges, with higher values at current edge locations compared to previous ones. Conversely, the inverse TSM exhibits smaller values at edges in the current frame. Therefore, we define the tsDF at time t_{cur} as:

$$\overline{\mathscr{T}}(\bullet) = 1 - T(\bullet, t_{cur}). \# \tag{11}$$

For each reconstructed edge point, $\mathbf{P}_m^e$, in the VFDM, we can transform its coordinates to the camera coordinate system using $[\mathbf{R}^i|\mathbf{t}^i]$ and then project it onto the TSM to obtain the 2D projection point, $\mathbf{x}_{uv}^e$, of the reconstructed edge point:

$$\mathbf{x}_{uv}^e = \mathbf{K}_e \left(\mathbf{R}^{i^{-1}} (\mathbf{P}_m^e - \mathbf{t}^i) \right), \# \tag{12}$$

where $\mathbf{K}_e$ is the intrinsic matrix of the event camera. Consequently, we define the objective function, $\mathscr{G}_{tsDF}$, within the tsDF as:

$$\mathscr{G}_{tsDF} = exp\left(- \frac{\sum_{m\in\mathscr{S}_e} \overline{\mathscr{T}} \left(\mathbf{K}_e \left(\mathbf{R}^{i^{-1}} (\mathbf{P}_m^e - \mathbf{t}^i) \right) \right)^2}{|\mathscr{S}_e|} \right), \# \tag{13}$$

where $\mathscr{S}_e$ is the set of valid 3D edge points in the current frame's VFDM that are visible in the previous frame's VFDM. In implementation, we determine this visibility relationship by checking whether the voxels in the current VFDM have undergone GPU-CPU data exchange. Notably, if a projected point has no corresponding pixel value, $\tilde{\mathbf{x}}$, on the TSM, i.e., $\mathscr{T}(\tilde{\mathbf{x}}, t) = 0$, then $\overline{\mathscr{T}}(\tilde{\mathbf{x}}, t) = 1$. This indicates that the model edge point has no corresponding edge on the TSM, significantly reducing the fitness of $\mathscr{G}_{tsDF}$.

5.4. Joint fitness evaluation

We innovatively propose jointly evaluating particle fitness using both the $\mathscr{G}_{TSDF}$ and $\mathscr{G}_{TSDF}$ energy terms. To achieve this joint evaluation, we define adaptive parameters, λ_1 and λ_2, in Eq. (5). The main idea is to weight the contribution of each energy term based on the voxel edge attributes (geometric or intensity edges). We first select a certain number, N, of 3D points based on the edge weights, $W_e^{cur}(\mathbf{p}_j), j \in N$, of each voxel in the current VFDM, in descending order, for projection and fitness calculation. This approach reduces the influence of edge uncertainty on fitness calculation while lowering computational costs. Then, we calculate the proportion, ω, of voxels at level L_0 within the selected set of edge voxels:

$$\omega = \frac{N_p}{N}, \# \tag{14}$$

where N_p is the number of voxels at level L_0 in the voxel set. The magnitude of the parameter ω reflects the planar intensity edge information that the depth camera cannot effectively capture. This information helps to improve the robustness of visual odometry in scenes with degraded geometric features. Therefore, under the same conditions, the voxel set should prioritize including such 3D points. It is important to note that we limit the range of ω to $0.5 \le \omega < 1$. This is because when the ω decreases, it indicates that there are rich geometric features in the scene, and edge voxels contain both intensity edges and geometric edges. Therefore, in this case, we adopt equal weights to evaluate the merits of particles, that is, the evaluation results of particles

are determined by two energy terms with the same weights, to ensure the accuracy of pose estimation. Another possibility for a decrease in the ω is that both geometric features and intensity features in the scene are degraded. At this time, we also use equal weights to handle the particle optimization problem to avoid over-reliance on any one feature, thereby ensuring the stability of the algorithm. Thus, we define:

$$\lambda_1 = 1 - \omega, \lambda_2 = 1 - \lambda_1 .\# \tag{15}$$

It is evident that the adaptive parameters λ_1 and λ_2 are continuously adjusted; this dynamic adjustment of parameters will be maintained throughout the entire operational cycle of the system without requiring manual intervention. It is worth noting that the TSM is defined as an exponential decay function of time in Eq. (1), limiting the tsDF values to the range of 0 to 1. In contrast, the TSDF directly records the distance from a point to the zero-crossing surface. Therefore, we also need to normalize $\mathcal{G}_{TSDF}$:

$$\mathcal{G}_{TSDF} = \exp\left(- \frac{\sum_{(u,v)\in O_d^e} \psi\left(\mathbf{R}^i \mathbf{P}_{uv}^d + \mathbf{t}^i\right)^2}{\mu^2 \left| O_d^e \right|} \right), \# \tag{16}$$

where μ is the truncation distance of the TSDF (Newcombe et al., 2011). In implementation, both energy terms, $\mathcal{G}_{TSDF}$ and $\mathcal{G}_{TSDF}$, are calculated concurrently. Fig. 4 illustrates our pose estimation process. Before particle fitness evaluation, we filter the original particle swarm template, Ω, to obtain a candidate particle set (CPS) and utilize the PNM for fast coarse localization (Gou et al., 2023) to improve the efficiency of random optimization. Similar to (Zhang et al., 2021), driven by our designed system dynamic function, the PST is iteratively scaled and moved until the optimal pose, $[\mathbf{R}^*|\mathbf{t}^*]$, is found:

$$[\mathbf{R}^*|\mathbf{t}^*] = \left(\mathbf{s}_t^i | \max\left(\mathcal{F}\left(\mathbf{s}_t^i\right)\right)\right) = \max(\lambda_1 \mathcal{G}_{TSDF} + \lambda_2 \mathcal{G}_{tsDF}).\# \tag{17}$$

6. Experimental evaluation

All our experiments were conducted on an embedded computing device, the NVIDIA Jetson Orin NX 16 GB. This device features an octa-core ARM Cortex-A78AE CPU, is equipped with 16 GB LPDDR5 RAM, and has an NVIDIA Ampere architecture GPU with 1024 CUDA cores, providing a computational power of 100 TOPS. Its compact size and lightweight design make it particularly suitable for use in small unmanned aerial vehicles. We set the voxel resolution to three levels: $L_0 = 8mm, L_1 = 4mm, L_2 = 2mm$. L_2 offers a very high level of detail, enabling dense 3D reconstruction with a maximum spatial resolution of $2mm$. We found that setting the truncation range of the TSDF to three times the voxel size of level L_0, i.e., $\mu = 24mm$, yields favorable reconstruction results. These parameters can be adjusted flexibly according to specific task requirements.

To comprehensively evaluate the performance of our proposed method, we designed a series of experiments, comparing it with existing methods both quantitatively and qualitatively. To enhance the credibility of the experimental results, we did not apply any additional code acceleration optimizations to the proposed method and ensured that the same input data was used for comparison with other methods. To ensure the best performance of competing methods, we strictly adhered to the default parameter settings provided by their authors. Additionally, we conducted ablation studies to validate the core design of our method, analyzing the contributions and roles of each module.

6.1. Benchmark

To validate the effectiveness of our proposed method more broadly, we investigated relevant datasets. Publicly available datasets like *MVSEC* (Zhu et al., 2018a), *VECtor* (Gao et al., 2022), and *ViViD++* (Lee et al., 2022) provide synchronized event streams and depth camera sequences (or alternatives to depth cameras), along with ground-truth trajectories. Specifically, *MVSEC* offers sequences with varying motion speeds under the same scene and lighting conditions. *ViViD++* includes sequences with different motion speeds and illumination conditions within the same scene. *VECtor* provides diverse sequences with varying motion speeds, texture features, and illumination conditions, as summarized in Table 1. We tested our proposed method against state-of-the-art (SOTA) approaches on these sequences and extended the experimental scope to several large-scale, self-collected sequences to comprehensively validate its performance.

We introduce a self-collected dataset called *DEveSet*, which comprises sequences with diverse texture, motion, and lighting conditions. The scenes are named *karst*、*maze*, and *garage*, respectively. The *karst* scene is located in an underground karst cave, *maze* features outdoor labyrinth walls with graffiti, and *garage* is situated in an urban underground parking lot. Data acquisition was performed in each scene under varying motion speeds, with illumination conditions categorized as light, dim, and dark. The *karst* cave has dim lighting, the *garage* has ample lighting, and the *maze* scene was captured during both daytime and nighttime. These sequences were collected using a custom-designed multi-sensor system consisting of a high-resolution event camera

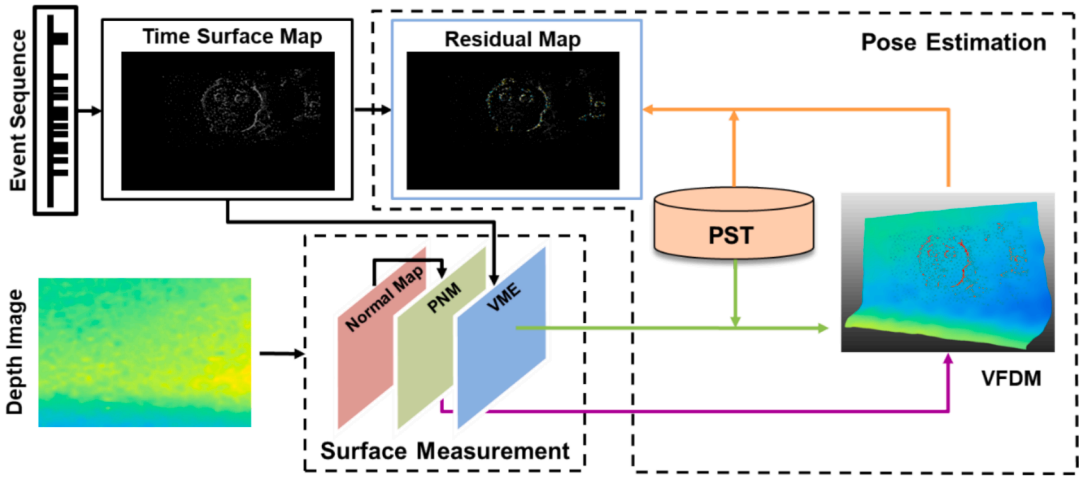

Fig. 4. Overview of our proposed pose estimation pipeline. Black arrows represent data flow within the data pre-processing module. Purple arrows indicate the data flow for filtering the Candidate Particle Set (CPS), which is a preparatory step for particle swarm optimization. Green and orange arrows represent the data flow for particle fitness evaluation using the TSDF and tsDF, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Statistics on scene characteristics (maximum linear velocity v_{max}, maximum angular velocity ω_{max}, illumination situations ***Illum.***, geometric ***Geo.Feat.*** and texture ***Tex.Feat.*** features) for different benchmark datasets.

Sequences	v_{max}(m/s)	ω_{max}(rad/s)	*Illum.*	*Geo.Feat.*	*Tex.Feat.*
MVSEC_indoor_flying	2.00	1.12	light	poor	rich
ViViD++_aggressive	1.62	2.11	light/dim/dark	mid	mid
VECtor_robot_fast	1.62	3.77	light	mid	mid
VECtor_desk_fast	1.73	2.55	light	rich	rich
VECtor_sofa_fast	1.73	2.51	light	rich	rich
VECtor_hdr_fast	0.91	2.26	dim	poor	rich
VECtor_mountain_fast	1.45	2.57	light	poor	rich
DEveSet_garage_full	2.43	4.58	light	rich	mid
DEveSet_garage_empty	2.61	4.37	light	poor	poor
DEveSet_maze_day	4.29	5.73	light	poor	rich
DEveSet_maze_night	3.58	5.02	dark	poor	rich
DEveSet_karst_flat	3.14	4.85	dim	mid	poor
DEveSet_karst_uneven	2.38	4.29	dim	rich	poor

(Prophesee EVK4) and an RGB-D sensor (Intel Realsense D435i), with detailed specifications listed in Table 2. The multi-sensor system was deployed on both handheld and UAV platforms (as shown in Fig. 5) and underwent rigorous intrinsic and extrinsic calibration using calibration tools (Gao et al., 2022; Oth et al., 2013). Given the difficulty of capturing camera trajectories under limited illumination and high-speed motion using visual motion capture systems, we adopted a similar approach to previous studies (Zhang et al., 2021) and acquired high-precision 3D laser-scanned dense reconstruction models of the scenes as ground-truth. The accuracy of pose estimation is indirectly evaluated by calculating the error between the dense reconstruction model and the laser-scanned model.

6.2. Comparisons with SOTA in challenge condition

We compared our proposed method with the current SOTA methods **InfiniTAM** (Prisacariu et al., 2017), **ROSEFusion** (Zhang et al., 2021), **Canny-EVT** (Zuo et al., 2024), and **DEVO** (Zuo et al., 2022). **InfiniTAM** and **ROSEFusion** utilize depth-only data as input; **Canny-EVT** incorporates both event and RGB data; while **DEVO**, most similar to our method in terms of input, uses depth maps and event data, as shown in Table 3. All methods utilize event camera and RGB-D camera data as input. We first conducted a qualitative comparison of the methods' performance, followed by a quantitative evaluation of our proposed method against the SOTA in terms of density, accuracy, and efficiency.

A qualitative comparison with the SOTA methods provides an intuitive assessment of our proposed method's performance. We selected two challenging illumination sequences, *ViViD++_aggressive_dim* and *VECtor_hdr_fast*, from public datasets with ground-truth trajectories to compare the trajectory tracking performance of different methods, as shown in Fig. 6. The results demonstrate that **DEveFusion** outperforms other competing methods across all sequences.

Although **Canny-EVT** utilizes event inputs with high dynamic range and high temporal resolution, its camera tracking module relies on a semi-dense edge 3D reconstruction created by monocular **ORB-SLAM** (Mur-Artal et al., 2015). However, due to insufficient scene illumination and the highly dynamic nature of platform motion, the tracking performance of monocular ORB-SLAM is significantly affected by motion blur and a lack of edge features. As a result, it fails to generate the semi-dense map required for effective **Canny-EVT** tracking, thereby hindering accurate camera localization. This is because stable and reliable texture features are difficult to capture under such conditions. In contrast, methods relying on depth as input are more adaptable to light-limited environments as depth sensors actively capture the scene's geometric features. However, methods solely dependent on geometric feature information for pose estimation experience tracking failures when encountering geometrically degraded scenes (e.g., *VECtor_hdr_fast*), even with abundant intensity features present.

Table 2
Specifications of sensors used in the custom sensor setup.

Sensor	Resolution	Frame Rate
Intel Realsense D435i	640 × 480	30 FPS
Prophesee EVK4	1280 × 720	–

Our **DEveFusion** method fuses both depth and event data for pose estimation. When geometric features are degraded, the intensity edges captured by the event sensor significantly enhance the system's robustness. **DEVO**, which also utilizes depth and event data as input, demonstrates good camera tracking performance in geometrically degraded scenes as long as stable intensity features exist and supports faster linear camera motion speeds (>1 m/s). Unfortunately, the creators of **DEVO** have not publicly released their code for testing. However, their report (Zuo et al., 2022) indicates that the **DEVO** system becomes vulnerable when the camera moves with high angular velocities (approximately > 2 rad/s), struggling to handle large-magnitude rotational motion. This is because the highly non-linearity introduced by large rotations hinders the convergence of gradient descent methods, leading to tracking failures. Large-magnitude rotational motion is prevalent in agile UAV movements, especially in indoor scenarios. Our proposed **DEveFusion** addresses the pose estimation problem with depth and event data fusion through PST filtering within a unified random optimization framework, significantly improving the system's pose estimation capabilities in complex environments.

Canny-EVT and **DEVO** ultimately generate semi-dense point clouds, lacking detailed scene information, while **InfiniTAM**, **ROSEFusion**, and our proposed **DEveFusion** achieve dense reconstruction, as indicated in Table 3. We focus our comparison on the dense reconstruction results and conduct a qualitative evaluation of these three dense reconstruction methods on our self-collected dataset, *DEveSet*, as shown in Fig. 7. We select three sequences: *karst_uneven_mid* with mid-speed motion, *garage_empty_fast* with fast motion, and *maze_night_fast*, covering diverse motion speeds, environmental features, and illumination conditions. The results demonstrate that our **DEveFusion** successfully reconstructs these three challenging scenes, while the other two methods fail.

ROSEFusion can track fast camera motion within a specified range, but reconstruction is terminated when the camera trajectory exceeds the preset cube, limiting its applicability in large-scale scenes (e.g., *karst_uneven_mid*). The *maze_night_fast* sequence is particularly challenging as the scene consists almost entirely of flat graffiti walls, lacking structural variations. This leads to severe degradation of depth sensor measurements, rendering **InfiniTAM** and **ROSEFusion**, which rely solely on structural information, unable to achieve successful reconstruction. Despite this, our method still achieves good reconstruction quality, thanks to our hybrid algorithm and the graffiti features on the walls. When the depth camera input deteriorates, the scene intensity features captured by the event camera maintain the robustness of the system. The *garage_empty_fast* sequence is also challenging as it contains only a few

Fig. 5. Custom sensor system with event camera and RGB-D sensor mounted on handheld and drone platforms for the self-collected datasets.

Table 3
Statistics on input data (color image *C*, depth image *D* and Event data *E*), pose optimization methods (non-linear optimization *n-L* and particle filter *PF*) and result map density for different methods.

	Canny-EVT	DEVO	InfiniTAM	ROSEFusion	DEveFusion
Input data	*C* + *E*	*D* + *E*	*D*	*D*	*D* + *E*
Pose opt.	*n-L*	*n-L*	*n-L*	*PF*	*PF*
Map density	Semi-dense	Semi-dense	Dense	Dense	Dense

repetitive structures and lacks sufficient intensity features, causing some drift in our method as well. Since our current system does not include loop closure detection and global optimization modules, it is unable to correct pose drift, resulting in localized distorted deformations in the reconstruction, as shown in Fig. 7. In the quantitative assessment of pose estimation accuracy, the errors are also significantly higher than those of other sequences, as indicated in Table 5. The intense and abrupt motion leads to significant pose differences between consecutive frames, causing non-linear optimization methods that rely on initial values to fall into local optima or even fail to converge. This is one of the reasons why **InfiniTAM** cannot provide competitive results on fast camera motion sequences. Although **ROSEFusion** exhibits significant advantages in tracking fast camera motion, its success heavily relies on rich geometric features in the scene. Large flat walls and ground are the primary causes of its system crashes. The highly intertwined nature of its tracking and mapping modules contributes to the failure of reconstruction in such challenging scenarios.

Our quantitative comparison experiments evaluate the performance of **DEveFusion** against SOTA methods in terms of accuracy and speed. We utilize the absolute trajectory error (ATE) metric to assess the accuracy of different methods on publicly available datasets with ground-truth trajectories. The results are summarized in Table 4. Overall, our method demonstrates the best performance. Except for **ROSEFusion**, all other compared methods failed to complete tests on all sequences in the public datasets. The primary reasons for tracking failures are challenging lighting conditions, motion speeds, and scene textures.

In Table 4, the fast motion in the *MVSEC* sequences is the main cause of failure for **Canny-EVT** and **InfiniTAM**. Moreover, the degradation of geometric features in the scene also has a detrimental effect on **InfiniTAM**, which relies entirely on geometric features for tracking. Thanks to its random optimization framework, **ROSEFusion** successfully tracks all sequences, but its tracking accuracy is significantly compromised by geometric feature degradation. Our method maintains system stability and accuracy by relying on abundant intensity features in the scene when encountering geometric information degradation. The *ViViD++_aggress.* sequence is extremely challenging due to the fast camera motion in a feature-poor environment, leading to tracking failures for **Canny-EVT** and **InfiniTAM**. The tracking accuracy of our method is also affected but remains superior among the compared methods. Unlike *MVSEC* and *ViViD++*, the small-scale sequences in the *VECtor* dataset reduce the difficulty of tracking. The favorable illumination conditions also increase the likelihood of success for methods relying on photometric information, enabling *Canny-EVT* to perform well on these sequences. Our method, utilizing both depth and event information as input, maintains high robustness and competitive tracking accuracy. This is attributed to our proposed hybrid depth-event random optimization framework.

The plots in Fig. 8 provide a detailed analysis of per-frame pose accuracy for six representative sequences. Specifically, we use the translation error (TE) (rotation error manifests as translation error during camera movement) to measure the pose error for each frame and plot the percentage of frames with pose TE below different thresholds. The results demonstrate that our method achieves more accurate per-frame pose estimation compared to the three competing methods. In the highly challenging sequence (*ViViD++_aggress.*), the per-frame pose accuracy of all methods declines. However, within an error threshold of less than 20*cm*, our method still achieves a 50 % success rate for tracked frames. Under the same conditions, **ROSEFusion**'s success rate is less than 9 %, while **InfiniTAM** fails to complete all tracking tasks.

In Table 5, we further compare the performance on 12 self-collected sequences. Since these sequences only provide LiDAR-based ground-truth reconstructions, we focus on evaluating the reconstruction quality, specifically completeness and accuracy relative to the ground-truth reconstruction. The reconstruction quality is most intuitively reflected in the completeness of the scene model. The experimental results demonstrate that our method consistently outperforms the other two alternative methods in challenging environments. The visual results of the dataset reconstruction are presented in Fig. 7.

The completeness of the model reflects the robustness of the corresponding system in the scene. Its value is closely related to the scene's size and the extent of the ground-truth reconstruction. For example, even with the same reconstructed area, a larger scene (such as *maze*) will result in a lower completeness value. Under the same scene, varying illumination conditions and motion speeds can lead to tracking failures, which are the primary cause of reduced reconstruction completeness. Additionally, another possibility for incomplete reconstruction is that the camera motion exceeds the system's preset reconstruction range. This situation only occurs in methods that require a predefined reconstruction area, such as **ROSEFusion**'s performance in the two long *karst_uneven* sequences. It should be noted that since the 3D laser scanning range is generally larger than the visual sensor's capture range, the scene model completeness values in Table 5 cannot reach 100 %.

We assess the model's accuracy by calculating the average distance between the reconstructed model and the LiDAR-based ground-truth

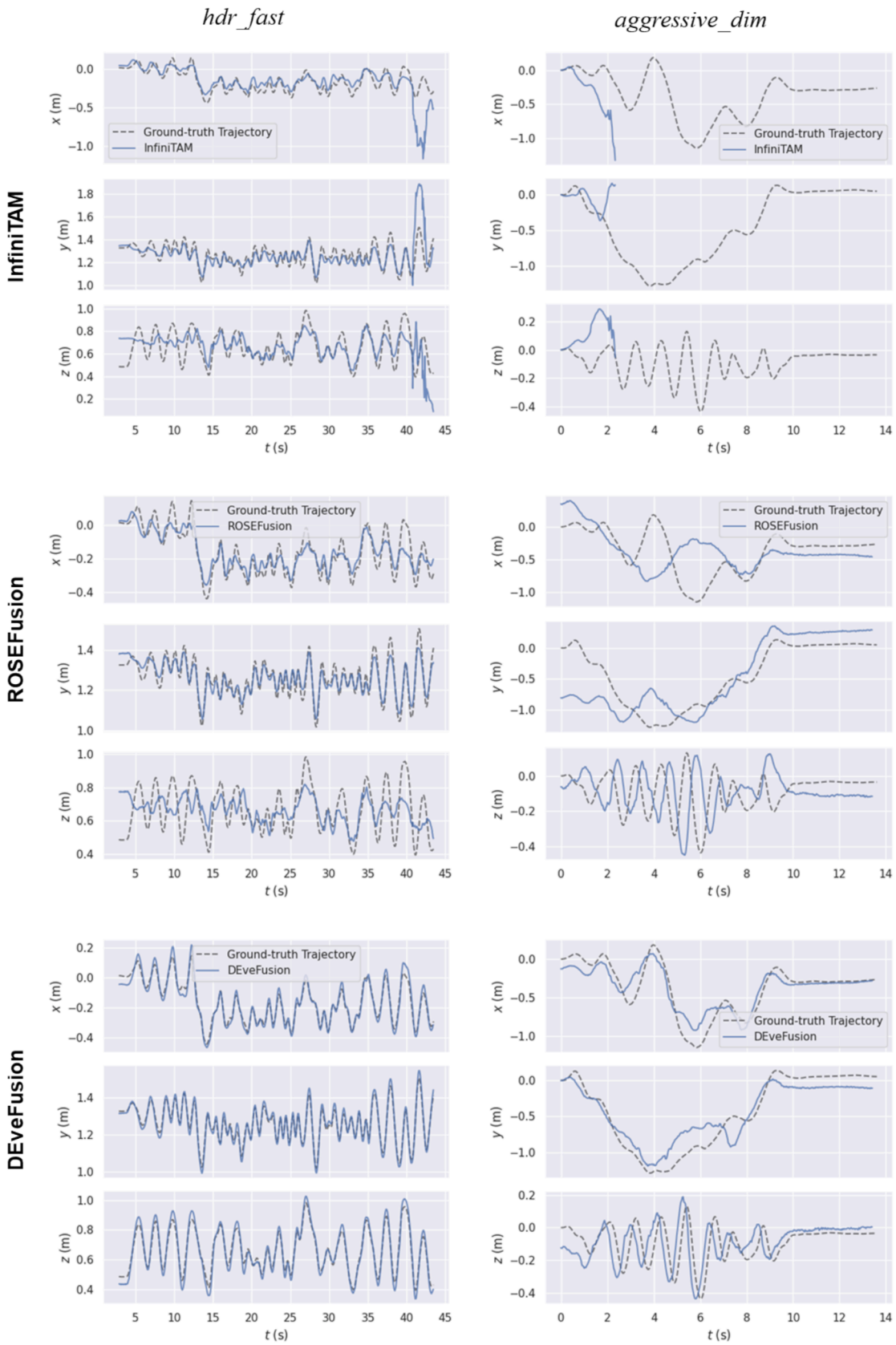

Fig. 6. Tracked camera trajectories (dotted curves) for two public dataset sequences and the ground-truth trajectories (solid curves) are overlaid for reference purpose. Compared to the competing methods, our method provides robust and reliable pose estimation performance.

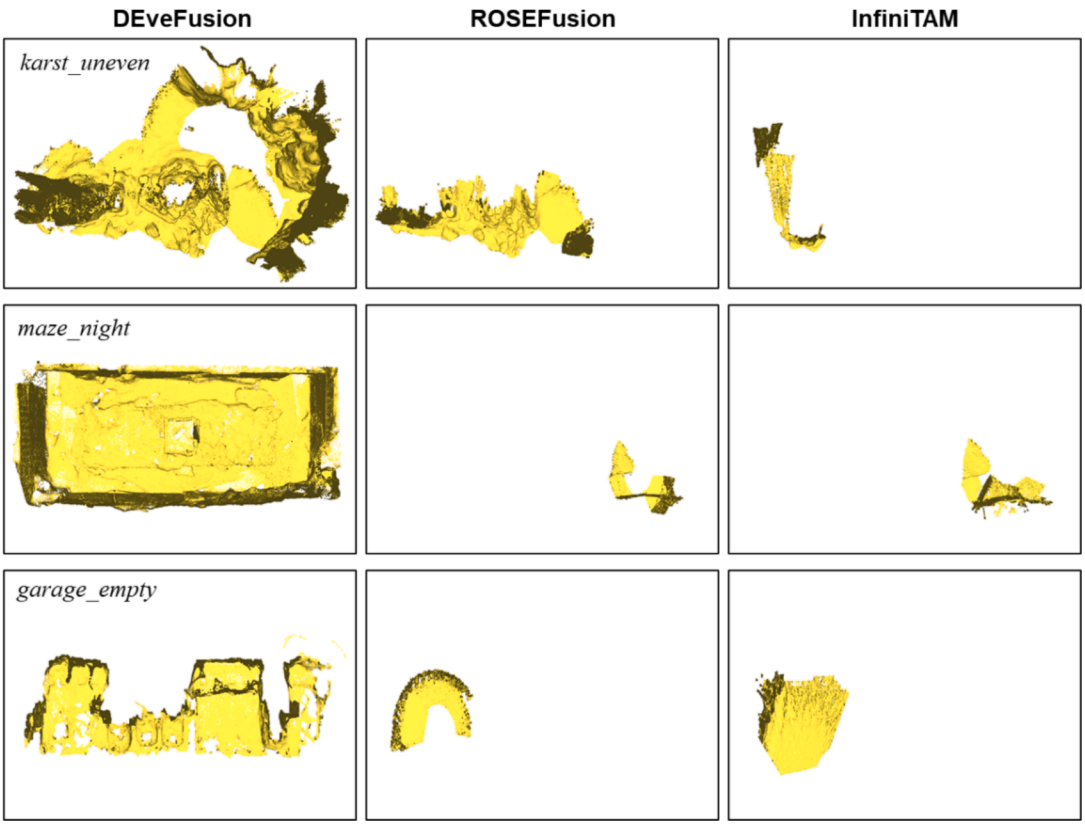

Fig. 7. Dense 3D reconstruction results for three challenge condition scenarios sequences of the self-collected datasets. For each sequence, our method demonstrates higher completeness and consistency in the reconstruction results.

Table 4
Comparing the accuracy (ATE cm) of camera tracking on the challenge sequences of the public datasets. The best and the second-best results are highlighted in **bold** and *italic*, respectively. '-' indicates that the tracking was failed or no test data.

Sequences	Canny-EVT	DEVO	InfiniTAM	ROSEFusion	DEveFusion
MVSEC_indoor_flying1	45.6	20.58	–	**8.37**	*9.50*
MVSEC_indoor_flying2	79.80	*11.33*	–	*42.37*	**5.40**
MVSEC_indoor_flying3	–	*10.60*	–	20.53	***5.03***
MVSEC_indoor_flying4	–	**13.16**	–	31.05	*13.29*
ViViD++_aggress._light	–	–	–	*35.84*	**34.85**
ViViD++_aggress._dim	–	–	–	*53.68*	**28.24**
ViViD++_aggress._dark	–	–	–	**30.39**	*30.84*
VECtor_robot_fast	9.07	–	5.37	*5.26*	**4.09**
VECtor_desk_fast	**5.80**	–	10.24	9.98	*6.49*
VECtor_sofa_fast	**1.29**	–	22.10	20.54	*4.46*
VECtor_hdr_fast	–	–	21.78	*14.04*	**4.62**
VECtor_mountain_fast	**2.14**	–	11.72	11.91	*5.65*

reconstruction. In Table 5, our method exhibits superior performance in reconstruction surface accuracy compared to the two competing methods, indicating that the proposed **DEveFusion** achieves higher pose estimation accuracy in challenging scenarios. It is important to clarify that since the reconstruction accuracy only measures the root mean squared error (RMSE) of the overlapping (inlier) regions between the reconstructed surface and the ground-truth surface, the values for the three methods are relatively close. In our evaluation, the inlier threshold is set to 15*cm*.

Our method demonstrates excellent competitiveness under varying motion speeds and in scenes with feature degradation. This is attributed to the proposed fusion framework, which enables the system to adapt to scenes with either geometric or intensity feature degradation, or both. For instance, in the *garage_empty* sequence, where both geometric and intensity features exhibit some degree of degradation, the accuracy of all compared methods decreases significantly. However, our method still completes the reconstruction of the entire sequence. In contrast, the other two methods terminate the mapping process shortly after the sequence begins due to tracking loss, preventing the accumulation of large errors and resulting in accuracy values numerically close to our results. This further highlights the advantage of our method in terms of global consistency. It is important to note that our system does not yet incorporate loop closure detection and global optimization modules. Our fusion method also demonstrates impressive performance in handling sequences with extremely fast camera motion. For example, in the *maze_day_fast* sequence from the *DEveSet* dataset, which exhibits the highest motion speed, our method shows greater model completeness and pose estimation accuracy. This is attributed to the exceptionally high temporal resolution of event data, which we incorporate into the random optimization pose estimation framework. Even when fast camera motion causes significant loss of depth information, the tsDF constructed based on stable event edges continues to provide effective

Table 5
Comparing the reconstruction completeness (Compl. %) and accuracy (Acc. cm) of alternative methods and our method over the DEveSet dataset. The best and the second-best results are highlighted in **bold** and *italic*, respectively.

Sequences	InfiniTAM		ROSEFusion		DEveFusion	
	Compl.	Acc.	Compl.	Acc.	Compl.	Acc.
garage_full_mid	23.59	8.94	**91.32**	*5.07*	*90.68*	**4.28**
garage_full_fast	14.52	9.90	*90.27*	*5.89*	**91.10**	**4.73**
garage_empty_mid	*15.92*	10.24	14.96	*9.57*	**84.33**	**9.01**
garage_empty_fast	15.21	11.97	*15.33*	*9.88*	**85.48**	**9.66**
maze_day_mid	9.47	*9.52*	*11.52*	10.11	**94.39**	**5.19**
maze_day_fast	8.59	*9.63*	*10.90*	9.97	**93.71**	**5.07**
maze_night_mid	*11.29*	8.23	9.81	*7.14*	**95.25**	**4.89**
maze_night_fast	9.29	8.03	*11.07*	*7.35*	**93.22**	**5.50**
karst_flat_mid	28.97	7.18	*73.62*	*6.22*	**89.31**	**4.95**
karst_flat_fast	12.73	7.37	*74.88*	*6.84*	**88.63**	**5.32**
karst_uneven_mid	36.16	*6.61*	*44.08*	6.92	**90.38**	**4.13**
karst_uneven_fast	11.91	8.41	*43.37*	*7.39*	**91.22**	**4.77**

particle evaluation performance, thereby ensuring the stability of the pose estimation system.

Our system achieves real-time 3D reconstruction at approximately 30*FPS* on an embedded device, comparable to the performance of many real-time 3D reconstruction systems running on graphics workstations, as shown in Table 6. The SOTA high frame rate 3D reconstruction system, **InfiniTAM**, can operate at an ultra-high frame rate exceeding 1000*FPS* on advanced graphics workstations and maintains a speed of around 60*FPS* even on resource-constrained embedded devices. **ROSEFusion** achieves a real-time reconstruction efficiency of 30*FPS* on the same workstation, but its frame rate drops below 8*FPS* in our embedded test environment, failing to meet the requirements for real-time reconstruction. This is mainly due to the significant computational overhead incurred by particle swarm fitness evaluation during its random optimization process. Thanks to our proposed key design elements, our system maintains a real-time reconstruction frame rate of approximately 30*FPS* even when employing a random optimization method. We will elaborate on and analyze this in the ablation experiments (c.f. Section 6.3).

It is important to note that the frame rate of the system is not constant but fluctuates with scene changes. For scenes rich in geometric features, the point cloud registration-based **InfiniTAM** algorithm converges more easily, resulting in higher efficiency. Conversely, the frame rate of **ROSEFusion** decreases in geometrically rich scenes. This may be because, under the particle fitness evaluation mechanism, scenes with simple geometric features are more likely to guide the particle swarm to converge rapidly to a local optimum, quickly generating a "less rigorous" solution. This is also one of the reasons why such methods exhibit decreased accuracy in scenes with simple geometric features. The opposite is true for scenes with rich geometric features. The efficiency trend of our method across different feature scenes aligns more closely with the latter. In summary, our method not only significantly improves the accuracy and robustness of pose estimation but also maintains high computational efficiency.

Table 6
Comparing the running efficiency (frame per seconds, FPS) of alternative methods and our method over the mid-speed sequences of the DEveSet dataset. The best and the second-best results are highlighted in **bold** and *italic*, respectively.

Sequences	InfiniTAM	ROSEFusion	DEveFusion
garage_full_mid	**67.82**	5.25	*31.86*
garage_empty_mid	**57.76**	7.81	*33.23*
maze_day_mid	**60.79**	5.40	*34.95*
maze_night_mid	**61.34**	5.32	*34.28*
karst_flat_mid	**62.84**	5.08	*32.32*
karst_uneven_mid	**74.07**	4.57	*29.79*

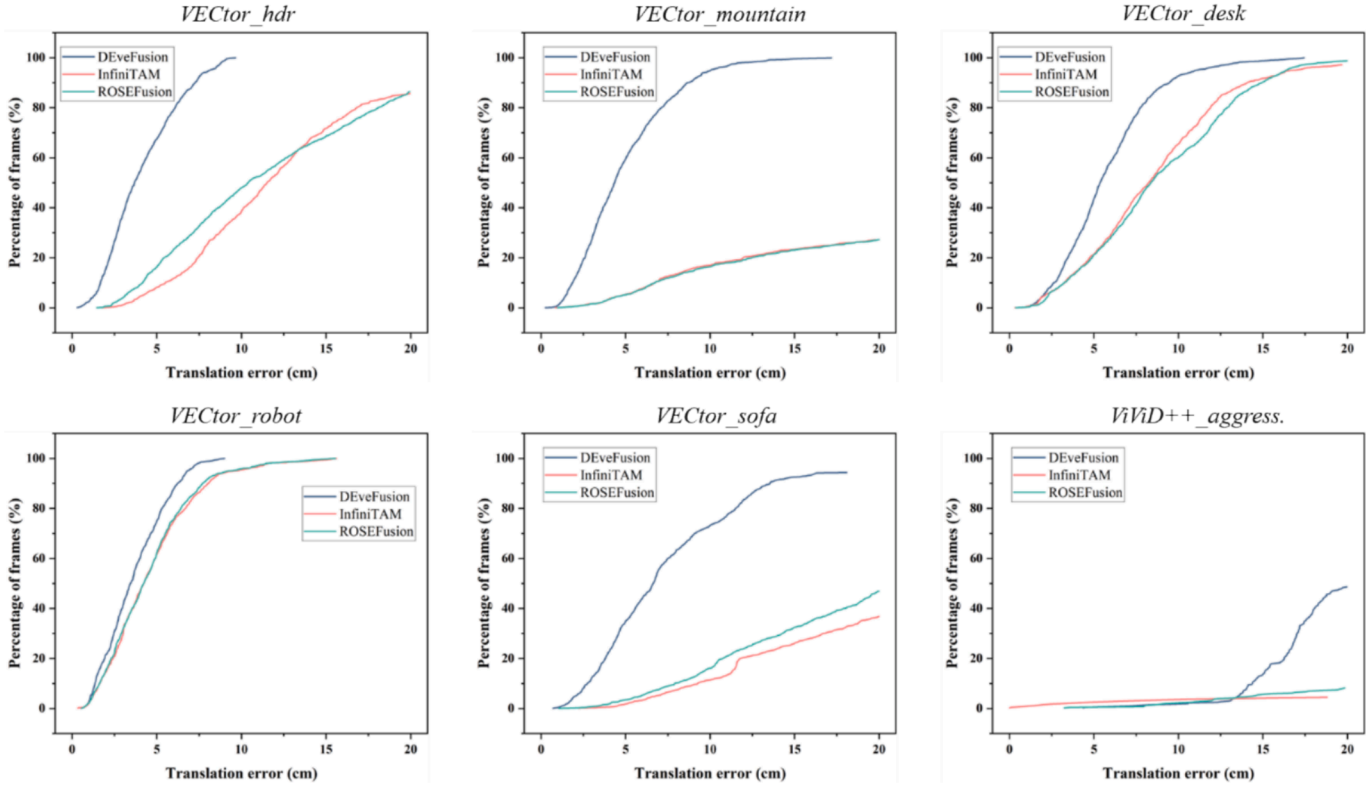

Fig. 8. Comparing percentage of frames under increasing tolerance of translation error of per-frame pose on six representative sequences of the public dataset. Our method achieves significantly higher success rate of per-frame pose tracking than alternative methods.

6.3. Ablation studies

We conducted a series of ablation studies to validate the necessity of several key design choices in our method. These include the hybrid depth-event pose estimation framework, the random optimization-based 3D-2D edge alignment mechanism, the surface edge update and extraction scheme, and the surface edge point selection strategy.

6.3.1. Pose estimation framework

The key to the success of our method in handling motion tracking in scenes with degraded geometric features lies in the meticulously designed hybrid depth-event pose estimation framework. The core algorithm is shown in Eq. (5), through which we incorporate both event data and depth information into a unified pose estimation framework. To validate the effectiveness of this core algorithm design, we compared the complete method with an algorithm that does not include the event term (representing our previous work (Gou et al., 2023)). To ensure a fair comparison, all methods utilize the same number and resolution of PSTs.

To compare with the results in Table 6, we evaluated the performance of both methods on six mid-speed sequences, including four sequences with degraded geometric features. Table 7 summarizes the reconstruction accuracy and efficiency of the two methods across different sequences. The statistics demonstrate that the hybrid method achieves a significant improvement in accuracy compared to the depth-only method, especially in scenes with degraded geometric features. Additionally, the reflection or absorption of light in the environment can lead to significant loss of depth information, which may result in a decline in system performance or even failure. This adverse effect is often present in artificial environments, such as the reflective mirrors commonly found in *garage* scenes, as well as the transparent glass and black bodies of vehicles in the *garage_full* scenario. In this work, we categorize both geometric feature degradation and the conditions of depth information loss under the term "geometric information degrades". In challenging conditions, our method exhibits satisfactory robustness, reflected in its near 100 % tracking success rate (TSR). In contrast, the depth-only method suffers from a lower TSR due to tracking loss. This improvement is attributed to the design of the hybrid framework: when geometric information degrades, the event tracking mechanism maintains the accuracy and robustness of the system. In terms of efficiency, the efficiency of our system decreases slightly compared to the depth-only method due to the introduction of more observation information, but it still maintains a real-time frame rate of 30*FPS*. Overall, our hybrid framework design strikes a good balance between performance and efficiency.

6.3.2. Random optimization-based 3D-2D edge alignment

In our core algorithm design, we achieve pose optimization by aligning the depth image with the reconstructed model and the reconstructed 3D edges with the 2D edges in the TSM. This problem is formulated as a mathematical optimization problem in Eq. (17). The alignment method between the depth image and the reconstructed model has been discussed in our previous work (Gou et al., 2023), so this paper focuses only on the 3D-2D edge alignment between reconstructed edges and TSM edges.

We propose to optimize the single-frame camera pose by maximizing the observation likelihood instead of using non-linear optimization methods commonly used in current event-based SLAM systems (Zuo et al., 2024). To validate the effectiveness of our design, we compare the accuracy and robustness of the two optimization methods within our hybrid framework, as shown in Table 8. In fast-motion sequences, the non-linear optimization method exhibits lower reconstruction accuracy, as the highly non-linearity makes pose estimation prone to falling into local optima. The robustness of the non-linear optimization method also faces severe challenges. Its TSR is below 50 % in the *maze* and *karst* sequences, which contain significant rotational motion. The *karst* sequence is particularly challenging due to its complex natural terrain. Our UAV data acquisition platform's autonomous obstacle avoidance and exploration in this scene involve substantial rotational motion, leading to the failure of the non-linear optimization method.

6.3.3. Surface edge update and extraction

A prerequisite for performing 3D-2D edge alignment is extracting sufficient edges from the dense reconstruction. To improve the accuracy and efficiency of edge extraction, a key design in this work is the surface edge update and extraction scheme based on edge reconstruction, as shown in Eq. (3). We demonstrate the superiority of our method by comparing it with a method that directly extracts point cloud edges by traversing the dense reconstruction results.

Table 9 presents the comparison results of the two methods in terms of tracking accuracy and time consumption. We select representative sequences from four test datasets for comparison and implement the traversal-based point cloud edge extraction method using functions from the Open3D open-source library (Zhou et al., 2018). The point cloud-based edge traversal method can only extract sharp geometric edges from the dense reconstruction results, while our edge reconstruction scheme can dynamically update both geometric and intensity edges of the 3D reconstructed surface, as shown in Fig. 9. In sequences with degraded geometric features (e.g., *MVSEC* and *VECtor_hdr*), the edge traversal method leads to sensor tracking failure due to a lack of sufficient features. On the other hand, in sequences with relatively rich

Table 7
Comparing the accuracy (Acc. cm), running efficiency (frame per seconds, FPS) and tracking success rate (TSR %) of the depth-only method and the hybrid depth-event method over the mid-speed sequences of the *DEveSet* dataset. The best results are highlighted in **bold**.

Sequences	Depth-only			Hybrid depth-event		
	Acc.	FPS	TSR	Acc.	FPS	TSR
garage_full_mid	5.49	**50.79**	94.3	**4.28**	31.86	**100.0**
garage_empty_mid	10.78	**58.41**	15.3	**9.01**	33.23	**97.6**
maze_day_mid	9.20	**54.29**	23.8	**5.19**	34.95	**100.0**
maze_night_mid	6.79	**52.78**	19.4	**4.89**	34.28	**100.0**
karst_flat_mid	6.87	**49.33**	88.6	**4.95**	32.32	**100.0**
karst_uneven_mid	7.51	**42.97**	57.2	**4.13**	29.79	**100.0**

Table 8
Comparing the accuracy (Acc. cm) and tracking success rate (TSR %) of the non-linear optimization method and the random optimization method over the fast sequences of the *DEveSet* dataset. The best results are highlighted in **bold**.

Sequences	Non-lin. Opt.		Ran. Opt.	
	Acc.	TSR	Acc.	TSR
garage_full_fast	10.03	75.3	**4.73**	**100.0**
garage_empty_fast	13.71	62.9	**9.66**	**91.7**
maze_day_fast	11.64	47.9	**5.07**	**100.0**
maze_night_fast	10.39	39.8	**5.50**	**95.0**
karst_flat_fast	9.42	45.2	**5.32**	**96.3**
karst_uneven_fast	9.38	27.1	**4.77**	**100.0**

Table 9
Comparing the accuracy (Acc. cm) and time consumption of the edge traversal method and the edge reconstruction method over the representative sequences of the test dataset. The best results are highlighted in **bold**. '-' indicates that the tracking was failed.

Sequences	Edge traversal		Edge recon.	
	Acc.	Time	Acc.	Time
MVSEC	–	6.9 s	**5.03**	**1.9 ms**
ViViD++	56.89	8.1 s	**30.84**	**1.8 ms**
VECtor	–	6.5 s	**4.62**	**2.1 ms**
DEveSet	6.02	7.2 s	**4.13**	**2.6 ms**

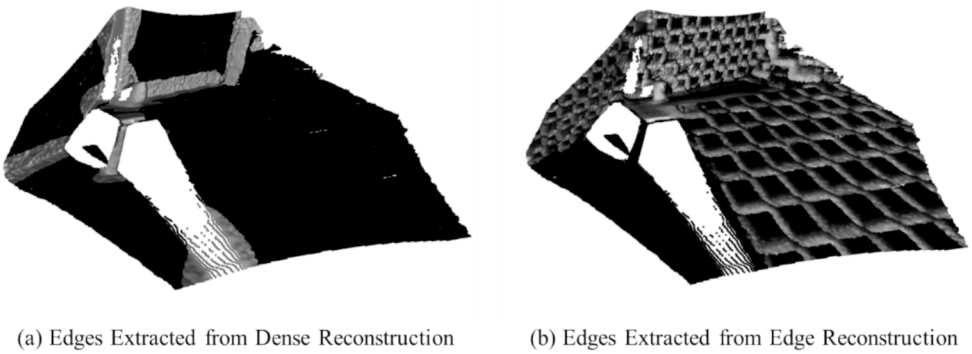

Fig. 9. Edges extracted results: (a) edge traversal method; (b) edge reconstruction method. The latter extracts more intensity edges located on the support plane in addition to geometric edges.

geometric features (e.g., *ViViD++* and *DEveSet_karst_uneven*), although the comparison method can also extract sufficient geometric edges, the edge extraction accuracy limits the improvement of the overall system accuracy. Notably, our edge reconstruction scheme can extract edges directly from the dense reconstruction results without traversing the entire reconstruction, achieving extremely fast edge extraction speed. This makes real-time applications of the system feasible.

6.3.4. Surface edge point selection

Complex textures and highly dynamic platform motion have been shown to have significant negative impacts on event streams (Zuo et al., 2024). Furthermore, we also observed that during close-range scanning, especially in low-light environments (e.g., *DEveSet_maze_night*), changes in the infrared light emitted by the RGB-D camera are easily captured by the event camera, producing noise points in the event stream that are difficult to filter out, as shown in Fig. 10(b). Fortunately, our proposed weighted edge reconstruction method exhibits strong robustness against such random noise points. In our edge point selection mechanism, these sporadic random noise points are not selected to participate in the particle fitness evaluation calculations because their edge weights are unlikely to exceed the weights of true edges. Random edge point selection methods are sensitive to noise and easily susceptible to noise interference, causing the optimization process to fall into local optima or

Table 10
Comparing the accuracy (Acc. cm) of the randomly selected edge points strategy and the weighted selected edge points strategy over the representative sequences of the test dataset. The best results are highlighted in **bold**.

Sequences	Rand. Sel.	Wgt. Sel.
MVSEC	9.33	**5.40**
ViViD++	37.82	**28.24**
VECtor	5.17	**4.09**
DEveSet	10.69	**4.89**

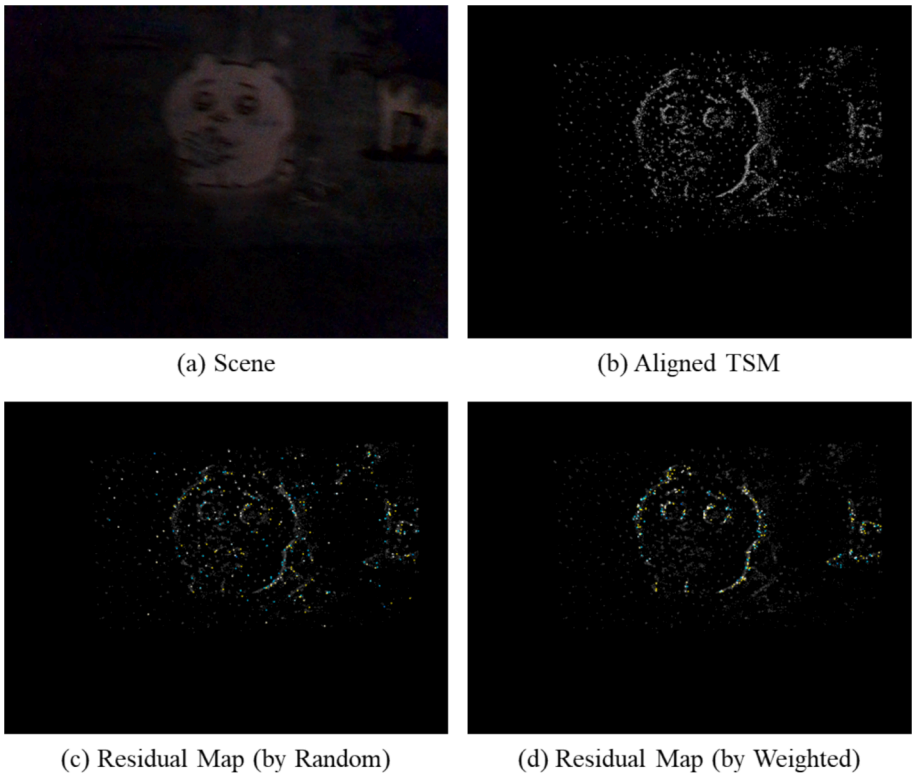

Fig. 10. Visualization of (a) an scene image and (b) the aligned TSM after extrinsic calibration, along with corresponding residual maps for randomly selected edge points and weighted selected edge points. Gray represent points on the aligned TSM, while colored represent the reprojected points.

even fail to converge because the alignment might not correspond to actual scene edges. Table 10 demonstrates the impact of weighted and random edge point selection strategies on system tracking accuracy. In general, when the noise level in the event stream is high, the edge point selection strategy has a greater impact on system accuracy; conversely, the impact is smaller when the noise level is low.

7. Conclusion and discussion

This paper presents a dense visual SLAM solution suitable for scenarios where photometric information is unavailable. Event cameras, with their high dynamic range, high temporal resolution, and low power consumption, are considered a promising alternative vision sensor. Recent research has demonstrated the potential advantages of event-based visual SLAM in challenging conditions. However, due to their reliance on traditional visual SLAM frameworks, these methods have limitations in tracking fast motion and adapting to diverse scenes. Furthermore, the limited spatial resolution of event data restricts the reconstruction of high-quality 3D maps. This paper demonstrates how the proposed method effectively utilizes event cameras to overcome these limitations.

We have developed a unified hybrid depth-event pose estimation framework based on random optimization, capable of effectively handling those problems. The complementary combination of depth and event information fully leverages both geometric and intensity information within the scene. The proposed fusion mechanism enhances the system's adaptability to degraded environments. Using depth data, we achieve dense 3D reconstruction of the scene and, through the proposed edge reconstruction strategy, ingeniously establish a tight coupling between dense reconstruction and pose estimation. In summary, we have built a hybrid depth-event dense SLAM system and conducted extensive tests in challenging scenarios. The method achieves a good balance between performance and efficiency.

We hope that this work will enhance the capabilities of versatile intelligent mobile systems in autonomous exploration and mapping across various challenging conditions. In future work, we plan to incorporate IMU signals to further enhance the system's robustness and reliability in extremely harsh environments where both geometric and intensity features are degraded. Integrating IMU signals with visual signals within a unified random optimization framework presents an exciting research direction. Currently, we are exploring the development of a loop closure detection and global pose optimization module that does not rely on photometric information to eliminate pose drift accumulated during prolonged system operation. Another pressing issue is how to integrate this method with online 6-DoF motion planning to achieve autonomous reconstruction for intelligent mobile systems (e.g., UAVs).

CRediT authorship contribution statement

Guohua Gou: Writing – original draft, Visualization, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Xuanhao Wang:** Writing – original draft, Visualization, Validation, Software. **Yang Ye:** Software, Formal analysis. **Han Li:** Formal analysis, Data curation. **Hao Zhang:** Visualization, Validation. **Weicheng Jiang:** Investigation, Data curation. **Mingting Zhou:** Writing – review & editing, Project administration, Funding acquisition. **Haigang Sui:** Formal analysis, Data curation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was funded by National Natural Science Foundation of China (No. 42401451), Key Research and Development Program of the Department of Science and Technology of Hubei Province (No. 2024BCB103) and Guangxi Science and Technology Major Project (AA22068072).

References

Agha, A., Otsu, K., Morrell, B., Fan, D.D., Thakker, R., Santamaria-Navarro, A., Kim, S.-K., Bouman, A., Lei, X., Edlund, J.A., Ginting, M.F., Ebadi, K., Anderson, M.O., Pailevanian, T., Terry, E., Wolf, M.T., Tagliabue, A., Vaquero, T.S., Palieri, M., Tepsuporn, S., Chang, Y., Kalantari, A., Chavez, F., Lopez, B.T., Funabiki, N., Miles, G., Touma, T., Buscicchio, A., Tordesillas, J., Alatur, N., Nash, J., Walsh, W., Jung, S., Lee, H., Kanellakis, C., Mayo, J., Harper, S., Kaufmann, M., Dixit, A., Correa, G., Lee, C.-A., Gao, J.L., Merewether, G.B., Maldonado-Contreras, J., Salhotra, G., da Silva, M.S., Ramtoula, B., Fakoorian, S.A., Hatteland, A., Kim, T., Bartlett, T., Stephens, A., Kim, L., Bergh, C.F., Heiden, E., Lew, T., Cauligi, A., Heywood, T., Kramer, A., Leopold, H.A., Choi, C.S., Daftry, S., Toupet, O., Wee, I., Thakur, A., Feras, M., Beltrame, G., Nikolakopoulos, G., Shim, D.H., Carlone, L., Burdick, J.W., 2021. NeBula: Quest for Robotic Autonomy in Challenging Environments; TEAM CoSTAR at the DARPA Subterranean Challenge. ArXiv abs/2103.1.

Alzugaray, I., Chli, M., 2018. Asynchronous Corner Detection and Tracking for Event Cameras in Real Time. IEEE Robot Autom Lett 3, 3177–3184.

Azpúrua, H., Saboia, M., Freitas, G.M., Clark, L., Agha-mohammadi, A., Pessin, G., Campos, M.F.M., Macharet, D.G., 2022. A Survey on the autonomous exploration of confined subterranean spaces: Perspectives from real-word and industrial robotic deployments. RoboticsAuton. Syst. 160, 104304.

Biggie, H., Rush, E.R., Riley, D.G., Ahmad, S., Ohradzansky, M.T., Harlow, K., Miles, M. J., Torres, D., McGuire, S., Frew, E.W., Heckman, C., Humbert, J.S., 2023. Flexible Supervised Autonomy for Exploration in Subterranean Environments. Field Robotics 3, 125–189.

Boguslawski, P., Zlatanova, S., Gotlib, D., Wyszomirski, M., Gnat, M., Grzempowski, P., 2022. 3D building interior modelling for navigation in emergency response applications. Int. J. Appl. Earth Obs. Geoinf. 114, 103066 https://doi.org/https://doi.org/10.1016/j.jag.2022.103066.

Bryner, S., Gallego, G., Rebecq, H., Scaramuzza, D., 2019. Event-based, Direct Camera Tracking from a Photometric 3D Map using Nonlinear Optimization. International Conference on Robotics and Automation (ICRA) 2019, 325–331.

Bylow, E., Sturm, J., Kerl, C., Kahl, F., Cremers, D., 2013. Real-Time Camera Tracking and 3D Reconstruction Using Signed Distance Functions, in: Robotics: Science and Systems.

Cao, C., Nogueira, L., Zhu, H., Keller, J., Best, G., Garg, R., Kohanbash, D., Maier, J., Zhao, S., Yang, F., Cujic, K., Darnley, R., DeBortoli, R., Drozd, B., Sun, P., Higgins, I., Willits, S.M., Armstrong, G., Zhang, J., Hollinger, G.A., Travers, M.J., Scherer, S.A., 2023. Exploring the Most Sectors at the DARPA Subterranean Challenge Finals. Field Robotics 3, 801–836.

Chamorro, W., Solà, J., Andrade-Cetto, J., 2022. Event-Based Line SLAM in Real-Time. IEEE Robot Autom Lett 7, 8146–8153.

Cook, M., Gugelmann, L., Jug, F., Krautz, C., Steger, A., 2011. Interacting maps for fast visual interpretation. The 2011 International Joint Conference on Neural Networks 770–776.

Dai, A., Nießner, M., Zollhöfer, M., Izadi, S., Theobalt, C., 2016. BundleFusion: real-time globally consistent 3D reconstruction using on-the-fly surface re-integration. ACM Trans, Graph, p. 36.

Endres, F., Hess, J., Sturm, J., Cremers, D., Burgard, W., 2014. 3-D Mapping With an RGB-D Camera. IEEE Trans. Rob. 30, 177–187.

Everding, L., Conradt, J., 2018. Low-Latency Line Tracking Using Event-Based Dynamic Vision Sensors. Front Neurorobot 12.

Gallego, G., Delbrück, T., Orchard, G., Bartolozzi, C., Taba, B., Censi, A., Leutenegger, S., Davison, A.J., Conradt, J., Daniilidis, K., Scaramuzza, D., 2019. Event-Based Vision: A Survey. IEEE Trans Pattern Anal Mach Intell 44, 154–180.

Gallego, G., Lund, J.E.A., Mueggler, E., Rebecq, H., Delbruck, T., Scaramuzza, D., 2016. Event-Based, 6-DOF Camera Tracking from Photometric Depth Maps. IEEE Trans Pattern Anal Mach Intell 40, 2402–2412.

Gallego, G., Scaramuzza, D., 2017. Accurate Angular Velocity Estimation With an Event Camera. IEEE Robot Autom Lett 2, 632–639.

Gao, L., Liang, Y., Yang, J., Wu, S., Wang, C., Chen, J., Kneip, L., 2022. VECtor: A Versatile Event-Centric Benchmark for Multi-Sensor SLAM. IEEE Robot Autom Lett 7, 8217–8224.

Gehrig, D., Rebecq, H., Gallego, G., Scaramuzza, D., 2018. EKLT: Asynchronous Photometric Feature Tracking Using Events and Frames. Int J Comput vis 128, 601–618.

Gentil, C.L., Tschopp, F., Alzugaray, I., Vidal-Calleja, T., Siegwart, R.Y., Nieto, J.I., 2020. IDOL: A Framework for IMU-DVS Odometry using Lines. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2020, 5863–5870.

Gou, G., Li, H., Wang, X., Zhang, H., Yang, W., Sui, H., 2025. Unsupervised deep depth completion with heterogeneous LiDAR and RGB-D camera depth information. International Journal of Applied Earth Observation and Geoinformation 136, 104327 https://doi.org/https://doi.org/10.1016/j.jag.2024.104327.

Gou, G., Wang, X., Sui, H., Wang, S., Zhang, H., Li, J., 2023. OwlFusion: Depth-Only Onboard Real-Time 3D Reconstruction of Scalable Scenes for Fast-Moving MAV. Drones.
Guan, W., Chen, P.-Y., Xie, Y., Lu, P., 2022. PL-EVIO: Robust Monocular Event-based Visual Inertial Odometry with Point and Line Features. ArXiv abs/2209.1.
Huang, K., Zhang, S., Zhang, J., Tao, D., 2023. Event-based Simultaneous Localization and Mapping: A Comprehensive Survey. ArXiv abs/2304.0.
Hudson, N., Talbot, F., Cox, M., Williams, J.L., Hines, T., Pitt, A., Wood, B., Frousheger, D., Surdo, K. Lo, Molnar, T., Steindl, R., Wildie, M., Sa, I., Kottege, N., Stepanas, K., Hernández, E., Catt, G., Docherty, W., Tidd, B., Tam, B., Murrell, S., Bessell, M.S., Hanson, L., Tychsen-Smith, L., Suzuki, H., Overs, L., Kendoul, F., Wagner, G., Palmer, D., Milani, P., O'Brien, M.J., Jiang, S., Chen, S., Arkin, R.C., 2021. Heterogeneous Ground and Air Platforms, Homogeneous Sensing: Team CSIRO Data61's Approach to the DARPA Subterranean Challenge. ArXiv abs/2104.0.
Kähler, O., Prisacariu, V.A., Ren, C.Y., Sun, X., Torr, P.H.S., Murray, D.W., 2015. Very High Frame Rate Volumetric Integration of Depth Images on Mobile Devices. IEEE Trans vis Comput Graph 21, 1241–1250.
Kerl, C., Sturm, J., Cremers, D., 2013. Robust odometry estimation for RGB-D cameras. IEEE International Conference on Robotics and Automation 2013, 3748–3754.
Kim, H., Handa, A., Benosman, R.B., Ieng, S.-H., Davison, A.J., 2014. Simultaneous Mosaicing and Tracking with an Event Camera, in: British Machine Vision Conference.
Kim, H., Leutenegger, S., Davison, A.J., 2016. Real-Time 3D Reconstruction and 6-DoF Tracking with an Event Camera, in: European Conference on Computer Vision.
Klein, G., Murray, D., 2007. Parallel Tracking and Mapping for Small AR Workspaces. In: IEEE and ACM International Symposium on Mixed and Augmented Reality, pp. 225–234. https://doi.org/10.1109/ISMAR.2007.4538852.
Kueng, B., Mueggler, E., Gallego, G., Scaramuzza, D., 2016. Low-latency visual odometry using event-based feature tracks. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2016, 16–23.
Lagorce, X., Orchard, G., Galluppi, F., Shi, B.E., Benosman, R.B., 2017. HOTS: A Hierarchy of Event-Based Time-Surfaces for Pattern Recognition. IEEE Trans Pattern Anal Mach Intell 39, 1346–1359.
Lee, A.J., Cho, Y., Shin, Y., Kim, A., Myung, H., 2022. ViViD++ : Vision for Visibility Dataset. IEEE Robot Autom Lett 7, 6282–6289.
Li, R., Shi, D., Zhang, Y., Li, K., Li, R., 2019. FA-Harris: A Fast and Asynchronous Corner Detector for Event Cameras. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2019, 6223–6229.
Liu, D., Bustos, Á.P., Chin, T.-J., 2020. Globally Optimal Contrast Maximisation for Event-Based Motion Estimation. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2020, 6348–6357.
Mahlknecht, F., Gehrig, D., Nash, J., Rockenbauer, F.M., Morrell, B., Delaune, J., Scaramuzza, D., 2022. Exploring Event Camera-Based Odometry for Planetary Robots. IEEE Robot Autom Lett 7, 8651–8658.
Messikommer, N., Fang, C., Gehrig, M., Scaramuzza, D., 2022. Data-Driven Feature Tracking for Event Cameras. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2023, 5642–5651.
Mostafavi, M., Wang, L., Yoon, K.-J., 2021. Learning to Reconstruct HDR Images from Events, with Applications to Depth and Flow Prediction. Int J Comput vis 129, 900–920.
Mueggler, E., Bartolozzi, C., Scaramuzza, D., 2017a. Fast Event-based Corner Detection, in: British Machine Vision Conference.
Mueggler, E., Gallego, G., Rebecq, H., Scaramuzza, D., 2017b. Continuous-Time Visual-Inertial Odometry for Event Cameras. IEEE Trans. Rob. 34, 1425–1440.
Mur-Artal, R., Montiel, J.M.M., Tardós, J.D., 2015. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. IEEE Trans. Rob. 31, 1147–1163.
Mur-Artal, R., Tardós, J.D., 2016. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. IEEE Trans. Rob. 33, 1255–1262.
Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohli, P., Shotton, J., Hodges, S., Fitzgibbon, A.W., 2011. KinectFusion: Real-time dense surface mapping and tracking. In: 2011 10th IEEE International Symposium on Mixed and Augmented Reality, pp. 127–136.
Nießner, M., Zollhöfer, M., Izadi, S., Stamminger, M., 2013. Real-time 3D reconstruction at scale using voxel hashing. ACM Transactions on Graphics (TOG) 32, 1–11.
Oth, L., Furgale, P.T., Kneip, L., Siegwart, R.Y., 2013. Rolling Shutter Camera Calibration. IEEE Conference on Computer Vision and Pattern Recognition 2013, 1360–1367.
Peng, X.-Z., Gao, L., Wang, Y., Kneip, L., 2021. Globally-Optimal Contrast Maximisation for Event Cameras. IEEE Trans Pattern Anal Mach Intell 44, 3479–3495.
Petrá\vcek, P., Krátký, V., Petrlík, M., Bá\vca, T., Kratochvíl, R., Saska, M., 2021. Large-Scale Exploration of Cave Environments by Unmanned Aerial Vehicles. IEEE Robot Autom Lett 6, 7596–7603.
Prisacariu, V.A., Kähler, O., Golodetz, S., Sapienza, M., Cavallari, T., Torr, P.H.S., Murray, D.W., 2017. InfiniTAM v3: A Framework for Large-Scale 3D Reconstruction with Loop Closure. ArXiv abs/1708.0.
Rebecq, H., Gallego, G., Mueggler, E., Scaramuzza, D., 2018. EMVS: Event-Based Multi-View Stereo—3D Reconstruction with an Event Camera in Real-Time. Int J Comput vis 126, 1394–1414.
Rebecq, H., Horstschaefer, T., Scaramuzza, D., 2017b. Real-time Visual-Inertial Odometry for Event Cameras using Keyframe-based Nonlinear Optimization, in: British Machine Vision Conference.
Rebecq, H., Horstschaefer, T., Gallego, G., Scaramuzza, D., 2017a. EVO: A Geometric Approach to Event-Based 6-DOF Parallel Tracking and Mapping in Real Time. IEEE Robot Autom Lett 2, 593–600.
Reinbacher, C., Munda, G., Pock, T., 2017. Real-time panoramic tracking for event cameras. IEEE International Conference on Computational Photography (ICCP) 2017, 1–9.
Murphy, R.R., 2021. How robots helped out after the surfside condo collapse. IEEE Spectr.
Schmidt, J., Niemann, H., 2001. Using Quaternions for Parametrizing 3-D Rotations in Unconstrained Nonlinear Optimization, in: International Symposium on Vision, Modeling, and Visualization.
Steinbrücker, F., Sturm, J., Cremers, D., 2011. Real-time visual odometry from dense RGB-D images. IEEE International Conference on Computer Vision Workshops (ICCV Workshops) 2011, 719–722.
Tedaldi, D., Gallego, G., Mueggler, E., Scaramuzza, D., 2016. Feature detection and tracking with the dynamic and active-pixel vision sensor (DAVIS). In: 2016 Second International Conference on Event-Based Control, Communication, and Signal Processing (EBCCSP), pp. 1–7.
Tranzatto, M., Mascarich, F., Bernreiter, L., Godinho, C., Camurri, M., Khattak, S., Dang, T., Reijgwart, V., Loeje, J., Wisth, D., Zimmermann, S., Nguyen, H., Fehr, M., Solanka, L., Buchanan, R., Bjelonic, M., Khedekar, N., Valceschini, M., Jenelten, F., Dharmadhikari, M., Homberger, T., Petris, P. De, Wellhausen, L., Kulkarni, M., Miki, T., Hirsch, S., Montenegro, M., Papachristos, C., Tresoldi, F., Carius, J., Valsecchi, G., Lee, J., Meyer, K., Wu, X., Nieto, J.I., Smith, A.P., Hutter, M., Siegwart, R.Y., Mueller, M.W., Fallon, M.F., Alexis, K., 2022a. CERBERUS: Autonomous Legged and Aerial Robotic Exploration in the Tunnel and Urban Circuits of the DARPA Subterranean Challenge. ArXiv abs/2201.0.
Tranzatto, M., Miki, T., Dharmadhikari, M., Bernreiter, L., Kulkarni, M., Mascarich, F., Andersson, O., Khattak, S., Hutter, M., Siegwart, R.Y., Alexis, K., 2022b. CERBERUS in the DARPA Subterranean Challenge. Sci Robot 7.
Valeiras, D.R., Clady, X., Ieng, S.-H., Benosman, R.B., 2019. Event-Based Line Fitting and Segment Detection Using a Neuromorphic Visual Sensor. IEEE Trans Neural Netw Learn Syst 30, 1218–1230.
Vasco, V., Glover, A.J., Bartolozzi, C., 2016. Fast event-based Harris corner detection exploiting the advantages of event-driven cameras. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2016, 4144–4149.
Vidal, A.R., Rebecq, H., Horstschaefer, T., Scaramuzza, D., 2017. Ultimate SLAM? Combining Events, Images, and IMU for Robust Visual SLAM in HDR and High-Speed Scenarios. IEEE Robot Autom Lett 3, 994–1001.
Wang, Y., Yang, J., Peng, X.-Z., Wu, P., Gao, L., Huang, K., Chen, J., Kneip, L., 2021. Visual Odometry with an Event Camera Using Continuous Ray Warping and Volumetric Contrast Maximization. Sensors (Basel) 22.
Weikersdorfer, D., Adrian, D.B., Cremers, D., Conradt, J., 2014. Event-based 3D SLAM with a depth-augmented dynamic vision sensor. In: In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 359–364. https://doi.org/10.1109/ICRA.2014.6906882.
Weikersdorfer, D., Conradt, J., 2012. Event-based particle filtering for robot self-localization. IEEE International Conference on Robotics and Biomimetics (ROBIO) 2012, 866–870.
Weikersdorfer, D., Hoffmann, R., Conradt, J., 2013. Simultaneous Localization and Mapping for Event-Based Vision Systems. International Conference on Virtual Storytelling.
Whelan, T., Salas-Moreno, R.F., Glocker, B., Davison, A.J., Leutenegger, S., 2016. ElasticFusion: Real-time dense SLAM and light source estimation. Int J Rob Res 35, 1697–1716.
Xia, Q., Dong, W., 2023. Design and Application of a Multifunctional Exploration Platform for Robotic Archaeology. In: 2023 IEEE 18th Conference on Industrial Electronics and Applications (ICIEA), pp. 1394–1400.
Ye, C., Mitrokhin, A., Fermüller, C., Yorke, J.A., Aloimonos, Y., 2020. Unsupervised Learning of Dense Optical Flow, Depth and Egomotion with Event-Based Sensors. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2020, 5831–5838.
Zhang, J., Zhu, C., Zheng, L., Xu, K., 2021. ROSEFusion: Random Optimization for Online Dense Reconstruction under Fast Camera Motion. ACM Trans. Graph. 40, 56:1-56:17.
Zhou, Q.-Y., Park, J., Koltun, V., 2018. Open3D: A Modern Library for 3D Data Processing. ArXiv abs/1801.09847.
Zhou, Y., Gallego, G., Shen, S., 2020. Event-Based Stereo Visual Odometry. IEEE Trans. Rob. 37, 1433–1450.
Zhu, A.Z., Atanasov, N.A., Daniilidis, K., 2017. Event-Based Visual Inertial Odometry. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017, 5816–5824.
Zhu, A.Z., Thakur, D., Özaslan, T., Pfrommer, B., Kumar, V.R., Daniilidis, K., 2018a. The Multivehicle Stereo Event Camera Dataset: An Event Camera Dataset for 3D Perception. IEEE Robot Autom Lett 3, 2032–2039.
Zhu, A.Z., Yuan, L., Chaney, K., Daniilidis, K., 2018b. Unsupervised Event-Based Learning of Optical Flow, Depth, and Egomotion. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2019, 989–997.
Zuo, Y., Xu, W., Wang, X., Wang, Y., Kneip, L., 2024. Cross-Modal Semidense 6-DOF Tracking of an Event Camera in Challenging Conditions. IEEE Trans. Rob. 40, 1600–1616.
Zuo, Y., Yang, J., Chen, J., Wang, X., Wang, Y., Kneip, L., 2022. DEVO: Depth-Event Camera Visual Odometry in Challenging Conditions. International Conference on Robotics and Automation (ICRA) 2022, 2179–2185.