Advanced Robotics Research

RESEARCH ARTICLE OPEN ACCESS

Improving the Robustness of Visual Teach-and-Repeat Navigation Using Drift Error Correction and Event-Based Vision for Low-Light Environments

Fuhai Ling[1] | Ze Huang[2] | Tony J. Prescott[1]

[1]Department of Computer Science and Sheffield Robotics, Sheffield University, Sheffield, UK | [2]School of Data Science, Fudan University, Shanghai, China

Correspondence: Tony J. Prescott (t.j.prescott@sheffield.ac.uk)

Received: 28 May 2025 | **Revised:** 2 November 2025 | **Accepted:** 5 November 2025

Keywords: deep learning for image processing | drift correction | event-based camera | visual teach and repeat navigation

ABSTRACT

We present a framework for visual teach-and-repeat (VTR) navigation designed to operate robustly in environments characterized by variable or low light levels. First, we show that navigation accuracy for VTR can be improved by integrating a topological map with a decision-making strategy designed to reduce latencies and trajectory error. Specifically, a local scene descriptor, acquired through deep learning, is coupled with stereo camera imaging and a proportional-integral controller to compensate for inaccuracies in visual matching. This approach facilitates accurate teach-and-repeat navigation with correction for odometry drift with respect to both orientation and along-route error accumulation using only monocular images during route following. Next, we adapt this general approach to operate with an off-the-shelf event-based camera and an event-based local descriptor model. Experiments in a night-time urban environment demonstrate that this event-based system provides improved and robust navigation accuracy in low-light environments when compared with a conventional camera paired with a state-of-the-art RGB-based descriptor model. Overall, high trajectory accuracy is demonstrated for VTR navigation in both indoor and outdoor environments using deep-learned descriptors, whilst the extension to event-based vision extends the capability of VTR navigation to a wider range of challenging environments.

1 | Introduction

Visual navigation has gained significant traction in both academic and industrial settings, offering compelling advantages such as cost-effective imaging technology and good energy-efficiency, especially with the integration of state-of-the-art deep learning algorithms. Nevertheless, lack of robustness is limiting its broader adoption [1, 2]. As a development of route-following approaches in mobile robotics, teach-and-repeat navigation creates a topological map during a teleoperated phase which the robot later uses for autonomous navigation [3–9]. A key advantage compared to full-blown simultaneous localization and mapping (SLAM) is that teach-and-repeat methods do not rely on acquiring a consistent metric map of the environment, thus reducing computational complexity and eliminating the need for explicit, global localization [7, 8]. Whilst often deployed in highly structured environments such as factories, teach-and-repeat navigation systems can provide autonomous navigation with emergency return capability in GPS-denied environments, underwater, or in densely constructed outdoor environments [8, 10–12]. Indeed, research on teach-and-repeat navigation systems has recently focused on robust operation across more challenging scenarios, including environments characterized by poor lighting or variable illumination [2, 10, 13].

https://doi.org/10.1002/adrr.202500105

This article introduces a novel framework for visual teach-and-repeat (VTR) which operates by constructing a topological map using deep-learned local descriptors and applies this to address two significant limitations of state-of-the-art systems.

First, since VTR employs a topological map rather than a metric one, it is more reliant on odometry for its navigation strategy. One of the main challenges facing current VTR systems, particularly when using a monocular camera, is therefore the accumulation of trajectory error due to odometric drift. In part I of this contribution, we couple a local scene descriptor, acquired through deep learning, with stereo camera imaging for 3D-2D feature mapping and in a manner that improves 2D feature matching when route-following with just monocular images. We show that this approach, combined with a proportional-integral controller, can provide accurate teach-and-repeat navigation in both indoor and outdoor settings and with correction for drift with respect to both orientation and along-route error accumulation.

Second, despite notable advancements in leveraging deep-learned descriptors for tasks such as day-to-night navigation, existing RGB-camera-based systems still face substantial challenges in poor lighting conditions. This limitation derives largely from the sensorial constraints of conventional cameras equipped with standard CMOS sensors. In contrast, bio-inspired event-based cameras offer substantial advantages, including microsecond-level temporal resolution, minimal power consumption, and high dynamic range [14]. Event-based cameras operate on a radically different principle to conventional devices, where each pixel functions independently and asynchronously, capable of detecting luminance changes over a dynamic range between $60dB$ to $140dB$. These characteristics could allow for effective navigation in poor lighting scenarios where traditional visual imaging technologies fail [15]. Current methods for handling the spatio-temporal structure of event streams in vision-based tasks also have limitations. For instance, hand-crafted event-based local features often lack the capability for effective noise filtration, semantic understanding, and dynamic adaptation to sensory data [16–19]. To overcome these challenges, part II of this contribution shows how our general approach can be adapted to use an off-the-shelf event-based camera, paired with a deep-learned descriptor that we recently developed specifically for use with event data [17]. Real-world experiments on a mobile robot platform show that this approach to VTR navigation provides improved and robust performance under night-time conditions compared to recent state-of-the-art benchmark systems.

We conclude by discussing potential future extensions of these methods including the calculation of visual odometry using a stereo pair of event-based cameras to reduce reliance on wheel-based odometry.

2 | Background

As demonstrated by Zhang and Kleeman [20], teach-and-repeat navigation can be formulated as a provably convergent control rule and Krajnik et al. [7] have shown its stability for closed polygonal routes, even for arbitary paths. For example, a set of images captured during human-guided route following can be used to build a visual path with visual servoing used to reconstruct the trajectory [7, 9, 21, 22]. Alternatively, the navigation path can be described as series of nodes each containing a collection of distinctive visual features and feature tracking used to guide the robot's motion between nodes [4, 23].

Successive nodes, in a VTR path, can be targeted either using direct image-based methods or feature-based methods [10]. Image-based methods typically use low resolution, normalized images that can be compared between the teach and repeat phases. Although this can be computationally efficient, this approach can suffer from low robustness due to variation in the visual scene, especially over the longer term. In feature-based methods, a set of salient features is extracted from each image and compared between teach and repeat phases. Image features can include edges, points, or corners, which can be used to calculate geometric shifts across the image-matched pairs.

The feature-based approach can be extended using methods from the field of deep neural networks for visual place recognition, where detailed image descriptors are processed to achieve unique identification of images in the environment [24]. For example, Camara et al. [3] used a convolutional neural network to create image descriptors for a teach-and-repeat task. This approach has the added benefit of enhanced robustness with respect to changes in appearance or viewpoint. Dall'Osto et al. [10] demonstrated a low-cost, bio-inspired teach and repeat technique that utilizes wheel odometry and a low-resolution monocular camera. They compared the similarity between images captured from the teach-and-repeat runs and estimated horizontal displacements from the cross-correction of a camera image and a map image for the robot's route transversal. Siegwart et al. [25] investigated the robustness of an image-processing approach when faced with varying degrees of wheel odometry corruption and without exploiting the precise kinematic equations of the robot. Their solution exhibited local stability but with significant reliance on hyperparameter selection and linear and angular velocity adjustments for each maneuver.

To address the challenge of environmental variance in long-term navigation, Sun et al. [2] previously proposed a monocular teach-and-repeat navigation system that utilizes deep-learned descriptors. In this work, a tailored self-supervised descriptor was introduced to deal with high variance illumination for day-to-night navigation. However, since a monocular camera was used to eliminate visual errors between teach-and-repeat runs the inability to resolve depth resulted in a lack of scale information. The resulting system was therefore susceptible to odometry drift leading to some deviations from the desired trajectory in both longitudinal and lateral directions. In the current paper we extend this deep-learned image descriptor approach to rectify odometry drifts and to correct the cumulative errors of the proprioceptive sensors. To this end, and as described in Part 1 below, 3D features obtained using a stereo camera are introduced to generate geometric pose information for control adjustments. This approach provides greater robustness to appearance changes. We then further address the specific challenge of variance of illumination by deploying an event-based camera and descriptor model in part II.

3 | Part I: Drift Correction in VTR

We will first show that navigation accuracy for VTR can be improved by integrating a topological map with a decision-making strategy designed to reduce latencies and trajectory error.

We begin by providing a general introduction to our VTR methodology, broken down into **teach** and **repeat** phases, including our approach to drift correction using 3D features obtained from a stereo RGB camera. Figure 1 provides an overall summary of the VTR navigation procedure described in more detail below. The figure also illustrates the software architecture as implemented for a DrRobot Jaguar 4 x 4 robot.

3.1 | Methodology

3.1.1 | Teach Phase

During the teaching phase a sequence of associated odometry positions and stereo camera images are recorded while the robot is teleoperated along the desired path. We use wheel speed and IMU-derived orientation data to estimate dead-reckoning odometry. Whenever the distance traveled by the robot from its last recorded position, as determined by odometry, exceeds a threshold τ, a pair of simultaneously generated images (left and right cameras) is captured and processed to obtain the 3D feature set, D, which is then combined with the left camera image, I, to construct the next element of the topological map R. We next explain the deep image processing and topological map construction in more detail. To aid understanding, Figure 2 shows an example test environment used in this study and some exemplar stereo camera images.

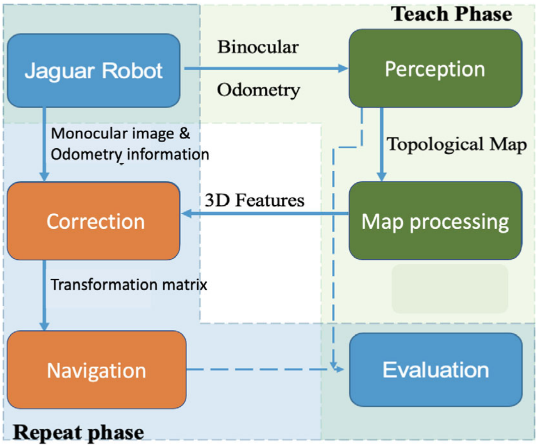


FIGURE 1 | Processing architecture for VTR navigation. Teach phase: The perception node receives stereo image pairs and odometry (wheels speeds and IMU data) from the robot and records the teleoperation commands. The map-processing node obtains the stereo-matched DarkPoint features and their 3D coordinates and constructs the topological map. Repeat phase: The correction node matches the DarkPoint features in the current (monocular) camera image with those from the stored image retrieved from the topological map and calculates the 3D-2D error correction. The navigation node receives the transformation matrix for proportional-integral control of the robot platform. The evaluation node performs trajectory comparisons using the Lidar point cloud and the LIO-SAM algorithm.

3.1.1.1 | Image Processing with DarkPoint. As noted above, image-processing methods based on deep learning are increasingly deployed in VTR navigation to provide robustness to appearance changes, including to variations in illumination, rotation, and scale. Here we build on our previous work [2] in which we developed and deployed the deep descriptor DarkPoint a self-supervised learning approach designed to enable robust visual localization for robots operating across day–night illumination changes. DarkPoint builds upon a VGG-style (Visual Geometry Group) architecture similar to SuperPoint [26], optimizing for both runtime efficiency and invariance to lighting, scale, and rotation. A key innovation in DarkPoint was its illumination adaptation strategy, which applied nonlinear Gamma correction specifically to the value channel in HSV (hue, saturation, and value) space to simulate realistic variations in lighting while preserving color integrity. This was combined with standard photometric augmentations (noise, contrast, shading) and homographic warping to generate diverse training pairs (for full details see [2]). Trained on modified COCO datasets (https://cocodataset.org/) with γ values spanning 0.2–4 to cover a wide variation in lighting conditions, DarkPoint successfully targeted the challenge of long-term VTR navigation where maintaining feature matching reliability under varying illumination is critical. More generally, this approach served to demonstrate how targeted synthetic augmentation combined with contrastive learning can bridge the gap between day and night visual environments. We next explain the process of constructing a topological map using DarkPoint features.

3.1.1.2 | Topological Map Construction. After teleoperating the robot to follow a specified trajectory we extract DarkPoint features from the image pairs and then match features between left and right images to calculate depth using triangulation. Specifically, a set of features F is generated from the image pairs

$$\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_n] \in \mathbb{R}^{d \times n} \tag{1}$$

together with depth information P for each feature

$$\mathbf{P} = [(\mathbf{x_1}, \mathbf{y_1}, \mathbf{z_1}), (\mathbf{x_2}, \mathbf{y_2}, \mathbf{z_2}), \ldots, (\mathbf{x_n}, \mathbf{y_n}, \mathbf{z_n})] \in \mathbb{R}^{3 \times n} \tag{2}$$

to create the set of feature/depth descriptors, D where

$$\mathbf{D_i} = \begin{bmatrix} \mathbf{F_i} \\ \mathbf{P_i} \end{bmatrix} \in \mathbb{R}^{(3+d) \times n} \tag{3}$$

The depth information we calculate is the 3D coordinates relative to the current robot position. As noted above, we construct the topological map, R, as an ordered list of cardinality M containing successive samples of 3D features, D, and monocular images, I, i.e.

$$R = \{(D_1, I_1), (D_2, I_2), \cdot;\cdot\, (D_m, I_m)\} \tag{4}$$

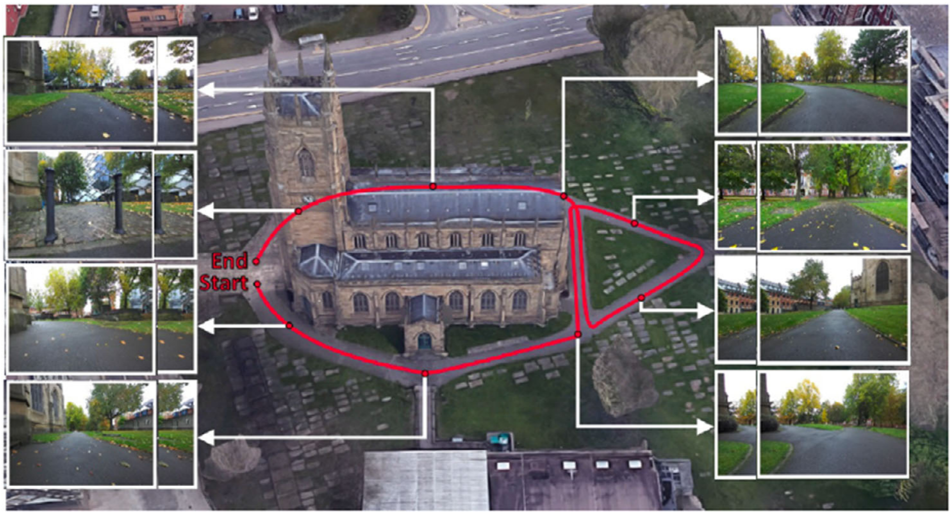


FIGURE 2 | Teleoperation in the grounds of a former church for map generation. In the teach phase a sequence of image pairs from a stereo camera are captured, some exemplar image pairs are shown superimposed on a background aerial image of the church and its surroundings.

The sequence of teleoperation action events is also saved during the teach phase as a form of coarse control for robot navigation as explained further below.

3.1.2 | Repeat Phase

In the navigation (repeat) phase, the system replays the recorded action events while simultaneously performing visual localization at each traversed distance τ by loading the image I_τ^o from the topological map and matching it with the current online image I_τ^p. This matching process supports two forms of drift correction: orientation correction and along-route correction. Note that whilst the teach phase uses a stereo camera for map construction, in the repeat phase a monocular camera is sufficient.

3.1.2.1 | Orientation Correction by Image Matching. Given a teach-and-repeat navigation model such as that described in [8], the translational error t_y perpendicular to the stored path can be approximately described as

$$t_y \sim z_t \, \mathrm{e}_\mathrm{t}^\mathrm{d}(o_m^d, o_t^d) \tag{5}$$

where $\mathrm{e}_\mathrm{t}(o_m^d, o_t^d)$ denotes the visual offset (image disparity) between the loaded map image and current on-board camera frame and z_t is the distance to the camera viewpoint extracted from the image. After the visual offset is estimated, the integral of the latency errors can be accumulated to adapt the steering control to different scales z. In contrast to [8], our method not only utilizes histogram voting to deduce visual offset but also accumulates this offset over time to refine the robot's trajectory. Specifically, we employ a brute-force nearest neighbor approach to match DarkPoint features and subsequently apply histogram voting to the x offsets, to obtain the most frequent inliers.

To perform orientation correction we follow the heading control strategy developed by Krajnik et al. [8]. Specifically, we introduce velocity adjustments based on the calculated visual offset

$$vel_a^{d'} = vel_a^d + vel_\mathrm{gain} \tag{6}$$

where vel_gain is the angular velocity compensation, determined through a combination of error integration and vision-based adjustments. We use a classic proportional-integral (PI) controller to estimate vel_gain, governed by manually tuned hyperparameters (see [2]). The robot's movements thus integrate stored actions and real-time vision-based corrections to converge on the pretrained route more efficiently.

3.1.2.2 | 3D Feature-based Along-Route Error Correction. Robot odometry information is degraded by wheel drift and can generate large along-route errors if allowed to accumulate over time [10]. Since the heading correction strategy described above is based only on the visual offset between the monocular map image and the current camera image this scale-drift problem cannot be directly mitigated by that approach. To overcome this challenge, 3D features from the topological map can be introduced. As the 3D features contain the position information (in meters) relative to the robot at the time of image capture, after matching with the current on-board monocular image, a 3D-2D feature pair will be obtained

$$\mathbf{D_m^\tau} = \begin{bmatrix} \mathbf{F_i} \\ \mathbf{P_i} \end{bmatrix} \in \mathbb{R}^{(3+d)\times n} \xrightarrow{\text{Matching}} \left(\mathbf{D_m^\tau}, \mathbf{F_t^\tau} \in \mathbb{R}^{\mathbf{d}\times\mathbf{n}}\right) \tag{7}$$

Here, D_m^τ indicates the 3D feature from the map frame and F_t^τ denotes the corresponding matched 2D feature at distance τ. We can apply a perspective-n-point (PnP) model to solve the movement between the map frame and the current camera frame using a set of matched 3D-2D features. In this article, we use Bundle Adjustment (BA) [27] to solve the PnP problem and obtain the appropriate translation. Specifically, let ${}^mT_t$ represent the pose of the current navigation position, as determined by the camera frame, relative to map frame. Next, calculate

$$\Delta \boldsymbol{d} = \mathrm{argmin}_\mathbf{T}(\|\phi({}^T T_N^o) - N * \tau * \phi(\mathbf{{}^m T_t})\|^2) \tag{8}$$

where ${}^T T_N^o$ indicates the odometry pose when map frame N is loaded and ${}^mT_t$ denotes the translation between the current camera image and map frame N (τ is the fixed translational interval as previously described). Since the translation T is a homogeneous transformation matrix, the function ϕ is used to obtain the 3D spatial coordinates from the homogeneous coordinates through projective transformation and then solve for the vertical distance. After calculating the difference Δd, we use this to update the odometry transversal distance correcting for drift and cumulative errors.

3.2 | Experiments on 3D-2D Error Correction

3.2.1 | Hardware and Software

In all experiments we use a DrRobot Jaguar 4x4 robot with dimensions of 62cm × 57cm × 90cm and a weight of 35kg. A ZED2 stereo camera was mounted on the robot to support 3D-2D error correction. We also mounted an Alienware X15 laptop on the robot paired with an NVIDIA Geforce RTX2070 Super-GPU to operate the image processing and real-time control. Additionally, to generate the 'ground truth' for evaluation of VTR navigation performance we used a 3D Ouster OS1−64 LiDAR and an Xsens MTi-G710-GNSS IMU sensor. This allowed us to use the LIO-SAM algorithm [28]) which uses LiDAR-inertial-GPS SLAM and ICP (Iterative Closest Point) matching for tracking and association of multisession trajectories. LIO-SAM was used to create all trajectory plots in this article. The integrated hardware and software systems are shown in Figure 3.

3.2.2 | Metrics and Baselines

In each experiment we evaluate the navigation accuracy by comparing the difference between the teach (mapping) and repeat (navigation) trajectories using two measures: the absolute trajectory error (*ATE*) and the relative pose error (*RPE*). In our earlier article [2] we compared our Darkpoint-based solution with STROLL [7] which was the only state-of-the-art solution for VTR navigation available at that time that was open-source. Our literature survey has not identified any more recent open-source VTR algorithms to compare against therefore we benchmark here against the baseline as described in [2] (since this out-performed STROLL). Note that we do not use a Visual SLAM algorithm as a benchmark since that approach is not directly comparable to VTR navigation (instead we use a LiDAR-based SLAM algorithm as 'ground truth').

3.2.3 | Forced Odometry Error Indoor Test

In order to verify our system's ability to compensate for large odometry errors, such as those caused by slippage and skidding, our initial test strategy was to manually raise the mobile robot slightly above the ground, allowing the wheels to run in the air for a several seconds, during the repeat trajectory (see Figure 4). As the robot replays the sequence of control events recorded from the teach phase, the odometry should continue to accumulate leading to a large accumulated error, whilst the camera remains in its current position along the path. The navigation trajectory, in an example experiment in an indoor laboratory environment, is illustrated in Figure 5 and the quantitative results shown in Table 1. We see that the repeat trajectory with the uplift test fits well with the teach trajectory in Figure 4 and achieves small ATE and RPE (see Table 1). For comparison we used the VTR model described in our earlier work [2] as a baseline, however, this baseline system failed to navigate in an appropriate direction following the uplift operation and therefore was unable to complete the navigation task.

3.2.4 | Error Correction in an Outdoor Environment

In a second experiment, we investigated the performance of regular teach-and-repeat (i.e. no uplift) in the outdoor environment shown in Figure 2 in daylight. The starting points of the teach and repeat phases were first arranged to be the same, we use $\Delta d = 0$ to indicate the difference between these two starting positions. Both trajectories (teach and repeat) were recorded and are illustrated in Figure 6. Good ATE and RPE are achieved of 0.0654 m and 1.345°respectively. Next, to test performance with respect to along-path error correction, the starting points for the teach and repeat phases were arranged to be in different positions, with $\Delta d = 1.5$m being transversal distance. The system described in [2] was again used as a baseline. As shown in Figure 7 left and Table 1, the robot is able to repeat the teach trajectory with high accuracy and smoothly correct for the difference is starting position Δd with only 0.097m ATE. However, the baseline navigation trajectory Figure 7 right deviates significantly from t he teaching path and with large route errors, 0.228 m ATE and 4.20° RPE.

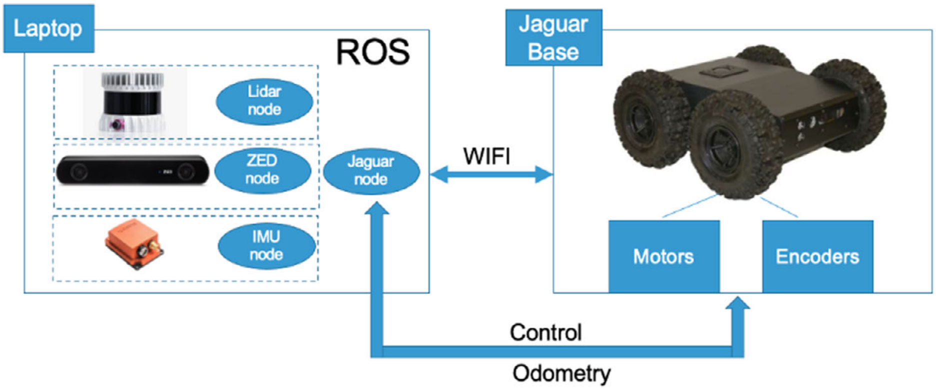


FIGURE 3 | Robot control systems, communications and platform. See text for further details.

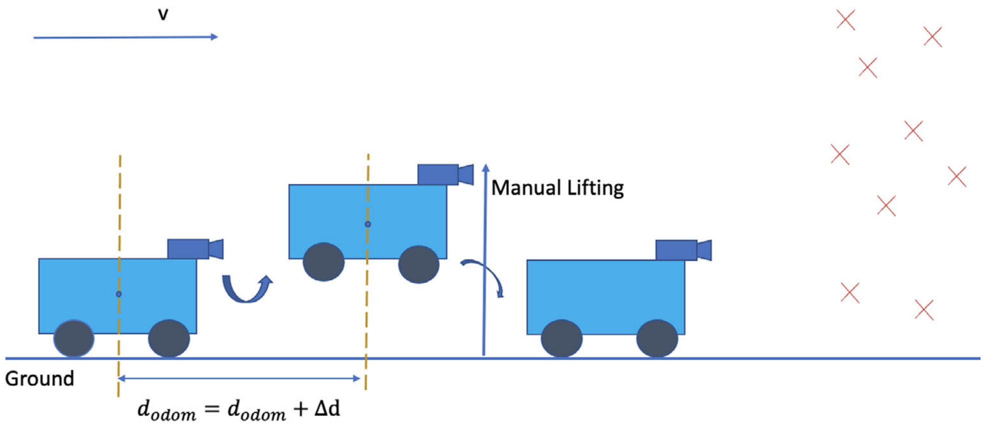


FIGURE 4 | Uplift test. To provide a strong test of our system's capacity to accommodate odometry errors we twice raised the robot above the ground for a short period during the repeat phase.

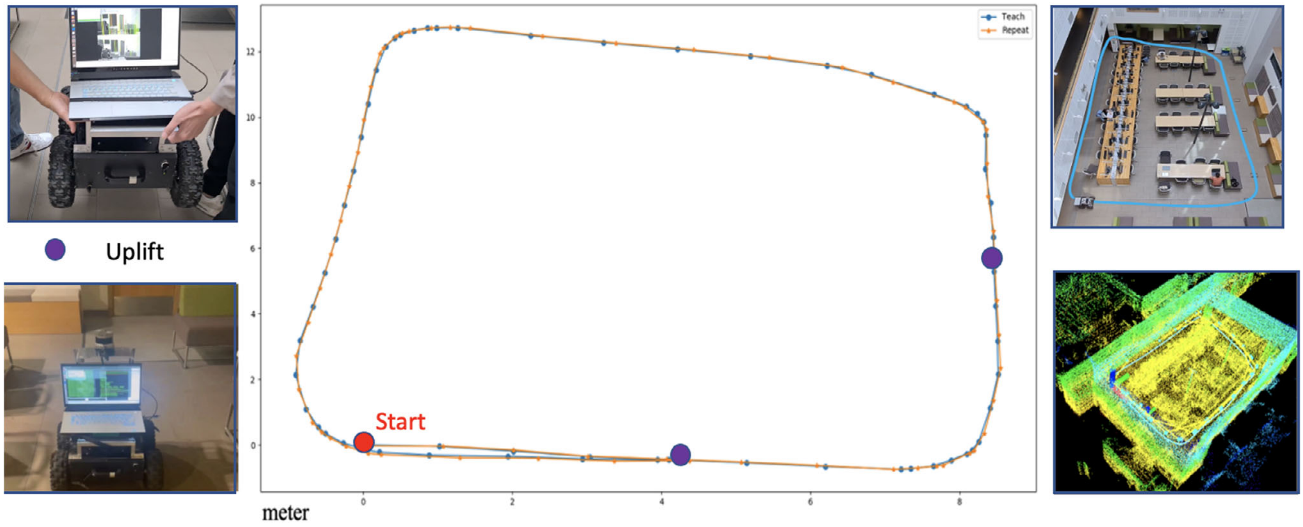


FIGURE 5 | Uplift test results. Center: The teach trajectory is shown in blue and the repeat trajectory in orange. Note that when the two trajectories are closely aligned the teach trajectory is largely obscured by the repeat trajectory. The Red dot indicates the starting point for both the teach and repeat runs. The purple dots indicate where the robot was lifted up (twice) during the repeat phase. Left: These images show the robot locomoting (bottom) and during the uplift intervention (top). Right: These images show the indoor environment (a university robotics laboratory) where the test was conducted (top) and the LIO-SAM point cloud and trajectory (bottom) used to calculate the ground truth for the 2D-projected trajectories in the central figure.

TABLE 1 | Quantitative evaluation of trajectory accuracy for 3D-2D error correction. Absolute trajectory error (ATE) is measured in meters and relative pose error in degrees. "Corrected" is the system with 3D correction described, baseline is our earlier monocular VTR system [2]. Mean and median values are calculated over all nodes of the topological map. RMSE = Root Mean Square Error.

Exp	Method	ATE (RMSE)	ATE (mean)	ATE (median)	RPE (RMSE)	RPE (mean)	RPE (median)	Result
Indoor Uplift ($\Delta d = 0$)	Corrected	0.0654	0.531	0.042	1.345	1.003	0.807	success
Indoor Uplift ($\Delta d = 0$)	Baseline	NA	NA	NA	NA	NA	NA	fail
Church ($\Delta d = 0$)	Corrected	0.0689	0.055	0.051	1.445	1.253	0.887	success
Church ($\Delta d = 1.5m$)	Corrected	0.097	0.073	0.065	1.971	1.553	1.168	success
Church ($\Delta d = 1.5m$)	Baseline	0.228	0.187	0.171	4.202	3.332	1.819	success

In summary, these experiments show the successful extension of VTR navigation using a stereo camera and deep learned descriptor model to allow 3D-2D error correction. This approach facilitates accurate teach-and-repeat navigation with correction for odometry drift with respect to both orientation and along-route error accumulation when retracing the route using just monocular camera images.

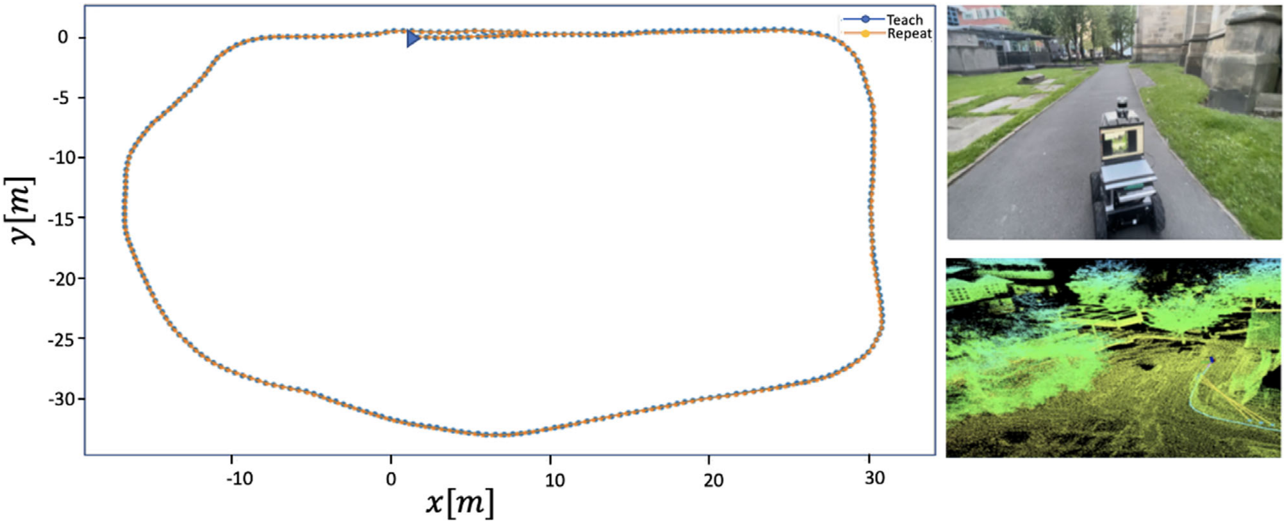


FIGURE 6 | Outdoor test of 3D-2D error correction. In this test, the teach and repeat runs were set to start at the same position, shown as the triangular icon. The right two images indicate the robot path-following at the church (top) and its corresponding LIO-SAM point cloud and trajectory (bottom).

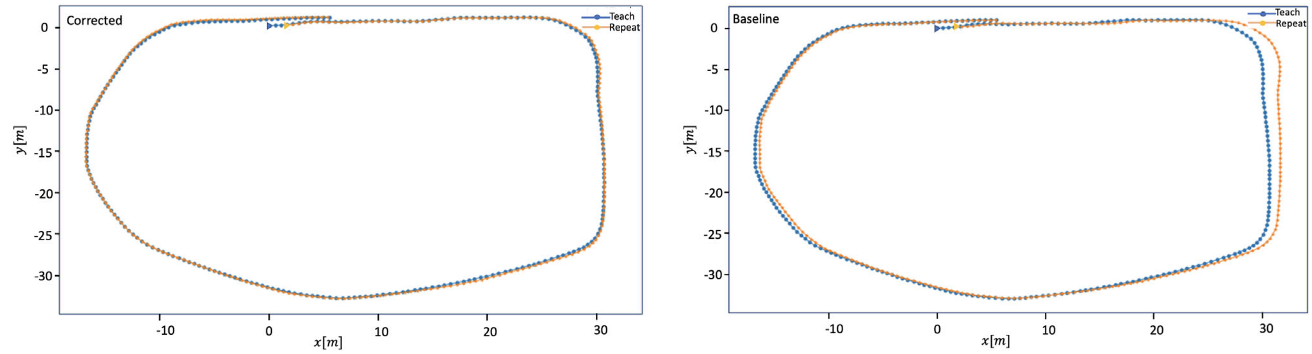


FIGURE 7 | Trajectories for 3D-2D corrective and baseline systems in the outdoor test. In this test, the teach and repeat runs were set to start at different positions with a 1.5m offset shown by the blue and orange triangular icons. The corrective system (left) shows a much closer match between teach and repeat phases than the baseline system (right), the latter is significantly offset to the right due to the change in starting position (although it subsequently recovers).

4 | Part II: Event-based Visual Teach and Repeat (VTR) Navigation in Variable Illumination Environments

In Part 2 we extend our general approach in a different direction by using an off-the-shelf event-based camera system in place of more conventional imaging systems whilst also addressing the core challenge of improving performance of long-term VTR navigation in variable environmental conditions.

Event-based cameras [14] have received significant attention in recent years due to their unique sensing characteristics and applications. They operate in a fundamentally different manner to RGB cameras, responding asynchronously to changes in local illumination and outputting a high bandwidth stream of labeled per-pixel brightness changes rather than generating whole camera frames at a fixed rate. The pixel-level response depends on the change in illumination rather than on the absolute level of illumination, consequently these cameras typically have better dynamic range and low-light sensitivity compared to RGB cameras. Within the domain of computer vision event-based vision has been heavily investigated for fundamental tasks such as object tracking and recognition [29–31] and optical flow estimation [32–34]. Being a pivotal area in robotics, there has also been significant interest from the field of SLAM navigation (e.g [35–37]). While previous research has demonstrated success in many fundamental or advanced tasks, there is still a relative lack of studies conducted in real-world scenarios with much of the aforementioned research carried out using synthetic data or in highly structured environments [38–40]. Within the field or robotics there is significant potential to harness the capacity of event-based cameras for stable imaging in variable or low light conditions [41]. Our work contributes to this challenge by integrating the capabilities of event-based cameras into our VTR navigation system.

Whereas in Part 1 we explored the use of the deep-learned descriptor DarkPoint, which was developed for robustness to variable illumination when using RGB cameras, here we use *EventPoint* a model optimized for event-based cameras developed by Huang et al. [17] which we describe next.

4.1 | Methodology

4.1.1 | Event-based Local Descriptor Model

The feature descriptor model used in our event-based VTR navigation system follows our previously published work, namely EventPoint [17], which is a deep-learned local feature network designed specifically for use with event-based cameras. As shown in Figure 8, EventPoint deploys a SuperPoint-like [26] fully convolutional neural network for local feature detection and description using event frames. The EventPoint model is pretrained using spatiotemporal-based self-supervised learning [17] on a publicly available real-world dataset DSEC [42] which was developed for use with event-based cameras in the context of driving scenarios. We deploy the pre-trained model directly in our VTR navigation system without fine-tuning to our specific scenes, challenging the model's ability to generalize to novel real-world situations.

4.1.1.1 | Event Frame Generation. To process the 3D event stream it is first transferred into a frame-like 2D representation as follows. We initialize a blank frame $F \in \mathbb{R}^{H \times W \times 3}$, where H and W are the height and the width of the event camera's resolution. Then, given a fixed temporal resolution Δt and a start time t_s all events that fall within the time window $[t_s, t_s + \Delta t]$ are projected into F

$$\begin{aligned} F[x,y] &= (255,0,0) \leftarrow (x,y,t,+1),\\ F[x,y] &= (0,0,255) \leftarrow (x,y,t,-1) \end{aligned} \tag{9}$$

Here (x,y,t,p) is the standard format of a single event point, where x and y refer to the 2D-position, t refers to the timestamp, and p is the polarity.

4.1.1.2 | Local Feature Generation with EventPoint. EventPoint consists of a shared encoder and two heads as shown in Figure 8. Given event frame F as input, the VGG-style encoder transforms the gray-scaled input $F \in \mathbb{R}^{H \times W}$ into a low-resolution but high-dimensional feature map $f \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times 128}$. The feature map f is then fed to two heads for local feature generation. One head, entitled 'detector', outputs a heatmap $h \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times 65}$ that gives the probability of each pixel falling within an $8 \times 8 + 1$ sized bin via a Softmax function. The last channel value represents

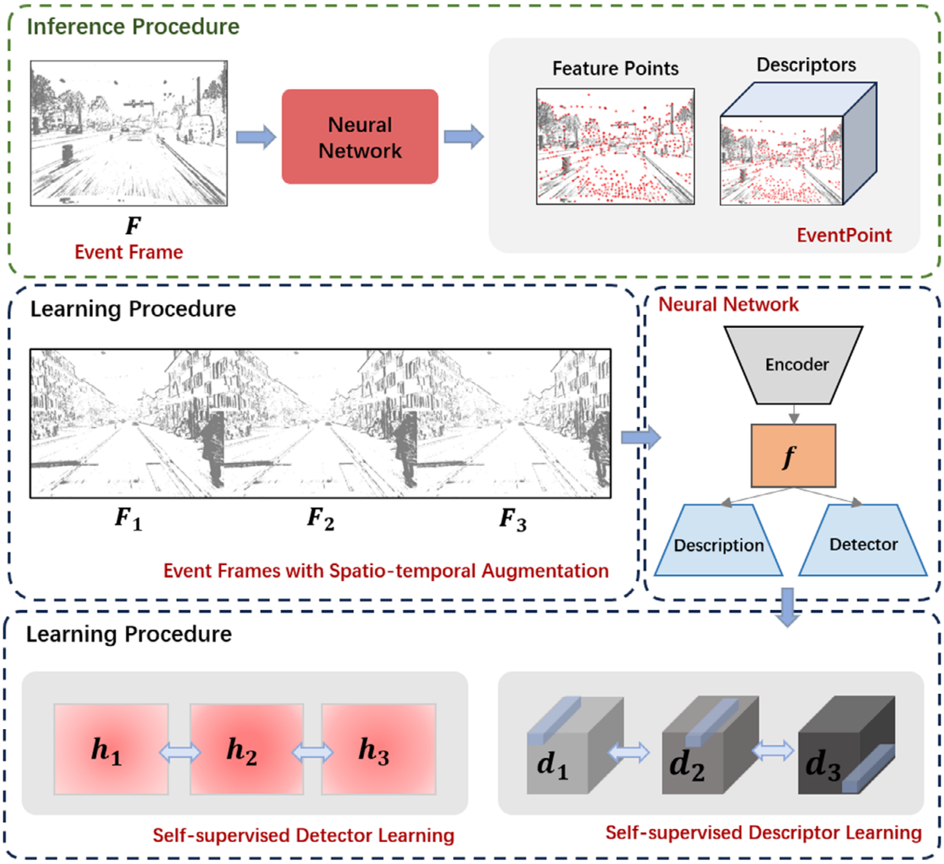


FIGURE 8 | The EventPoint descriptor model. EventPoint is an encoder-decoder-based neural network employed to process event frames as input that serves as the foundation for event-based local feature extraction, feature point detection and description. As explained further in the text and in [17], EventPoint is trained with a focus on spatiotemporal consistency ensuring that the location and descriptor of a feature point remains consistent across event frames generated with small temporal resolution perturbations.

whether the bin holds a local feature or not. The heatmap h can be further restored to the original frame size through the Reshape operation

$$h \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times 65} \xrightarrow[65]{\text{Softmax}} \xrightarrow{\text{Reshape}} h_{out} \in \mathbb{R}^{H \times W} \tag{10}$$

All points with a probability higher than a threshold τ are regarded as local feature points. The second head, entitled “description”, first outputs a dense grid of descriptors $d \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times 128}$, and then obtains a dense descriptor of the same size as the original event frame through bi-cubic interpolation

$$d \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times 128} \xrightarrow{\text{bi}-\text{cubic}} d_{out} \in \mathbb{R}^{H \times W \times 128} \tag{11}$$

The descriptors are then furthered normalized, and the detected feature points in h_{out} paired with their corresponding descriptors in d_{out} based on pixel coordinates. For further details see [17].

4.1.2 | VTR Navigation Method for Event-based Vision

During the teaching stage, and as described in Part I, we bypass metric mapping and localization in favor of an image-centric topological map. In this phase, the robot is teleoperated as before, odometry data are again employed to measure traveled distances, and images are captured at predefined intervals, this time using the event-based camera. In contrast to Part I, we construct the topological map, R, as an ordered list of cardinality M containing just the event frames

$$R = \{I_1, I_2, \ldots, I_m\} \tag{12}$$

In other words, in this system we do not currently make use of the conventional cameras and we defer calculating image features until the repeat navigation phase.

During the navigation phase, the robot again replays prerecorded actions while adjusting its pose based on the real-time and saved images at each distance d. For orientation correction we used the method described in part I for calculating the visual offset, this time comparing the current and stored event frames after processing with EventPoint.

4.2 | Low Illumination Experiments with the Event-based Camera

4.2.1 | System Configuration and Baselines

We used the robot system described in part 1 augmented with an *Inivation* DVXplorer Lite event camera sensor (https://inivation.com/). We wished to test the system in variable illumination conditions including under low-light levels. In VTR research, state-of-the-art methods include STROLL [7, 8], multiexperience map [43–45], and adaptive feature [46, 47]. However, DarkPoint [2], described in Part 1, which was specifically developed for VTR navigation in day–night conditions, achieved superior performance to these other methods in low light environments. Therefore, DarkPoint was selected as our baseline for experimental comparison with EventPoint. Specifically, the following visual descriptor models are integrated on our system for night-time VTR navigation experiments.

4.2.1.1 | EventPoint with Nearest Neighbor Matching.

To ensure the real time operation and save computational resources, a maximum number of 300 feature points is set. The event visual frame is represented by a Tencode [17] event accumulator with an accumulation time Δt of 50 ms and a detection threshold τ to 0.015. Figure 9 shows examples of feature discovery (red points) and matching (green lines) for the EventPoint model in indoors and outdoors environments.

4.2.1.2 | DarkPoint with Nearest Neighbor Matching.

We use the same system settings and RGB camera (ZED2) as [2] as a baseline. Parameters includes a detection threshold of 0.005 and a nonmaximum suppression radius of 4 pixels. Figure 10 shows examples of matches obtained in the same environments as Figure 9, and under similar low-light conditions, for an RGB camera using DarkPoint features. Note the much lower number of identified and successfully matched features compared to the event-based camera system.

4.2.2 | Procedure for Outdoor Night-Time Experiments

We conducted tests at the darkest time of the night (00:30-03:00am) when natural illumination was minimal and most street and building lights were turned off. Experimental sites included a former church and a building yard, both in Sheffield and situated near the University. The navigation path at the former church (see Figure 11) spanned ≈ 120 m, while the trajectory distance in the yard was around 50 m. Five test runs were performed in each environment.

As in [2], during the teach phase, camera images and odometry were used to generate a topological map while the robot was

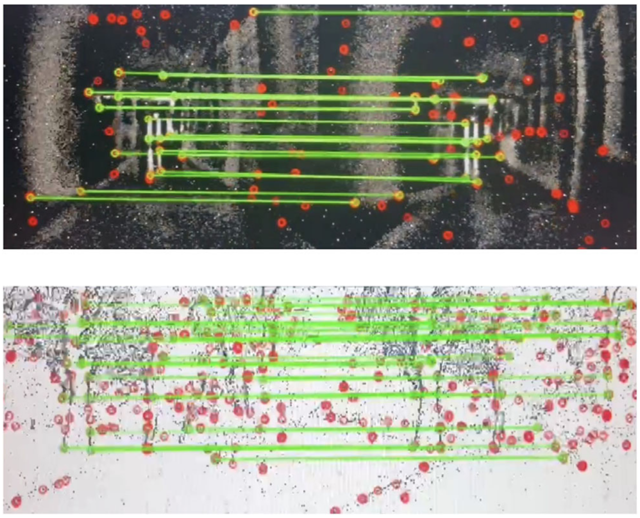


FIGURE 9 | Feature discovery and matching with EventPoint and the event-based camera. Example feature matching between the map event frame (left) and the current observation event frame (right) using EventPoint for indoors (top) and outdoors (bottom) environments, both with low light levels.

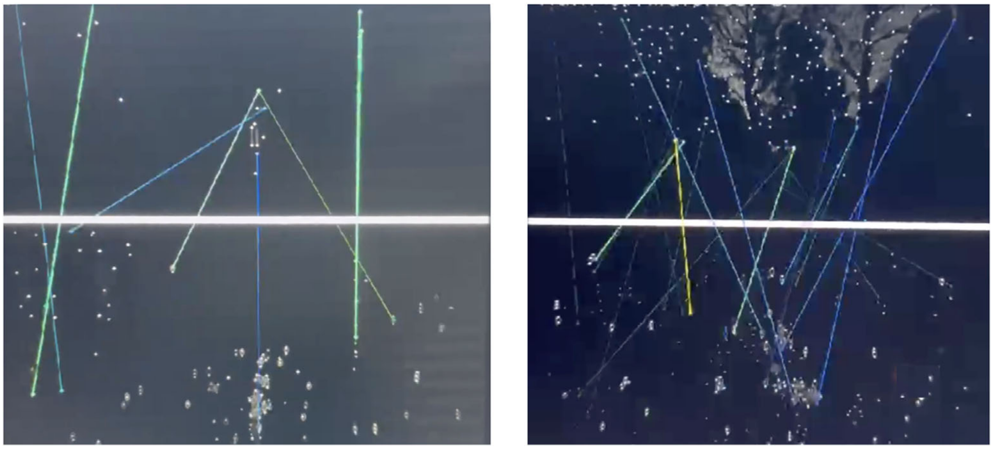


FIGURE 10 | Feature discovery and matching with DarkPoint and the RGB camera. Example DarkPoint feature matching using the RGB camera between the map frame (top) and the current observation (bottom). These were captured in the same indoor (left) and outdoor (right) environments as in Figure 9 with similar low light levels.

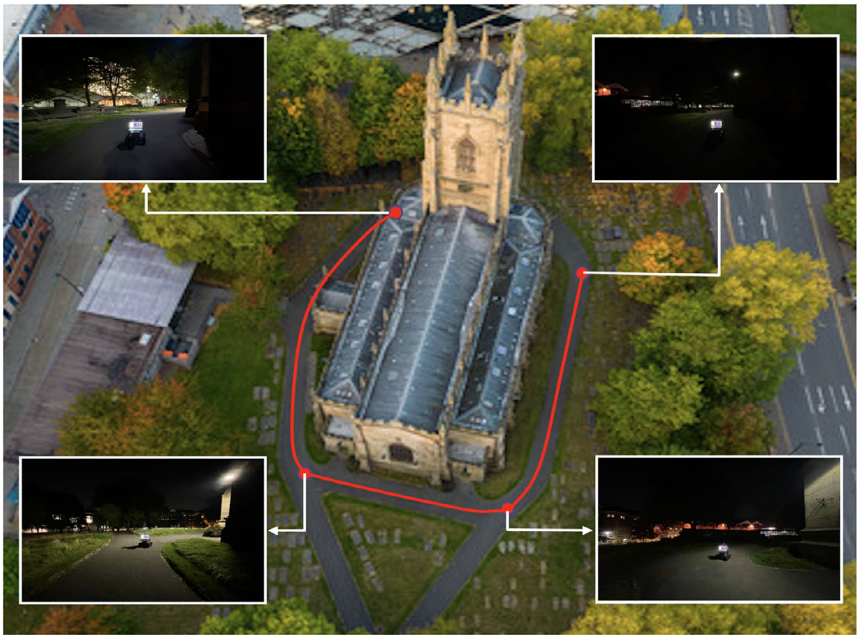


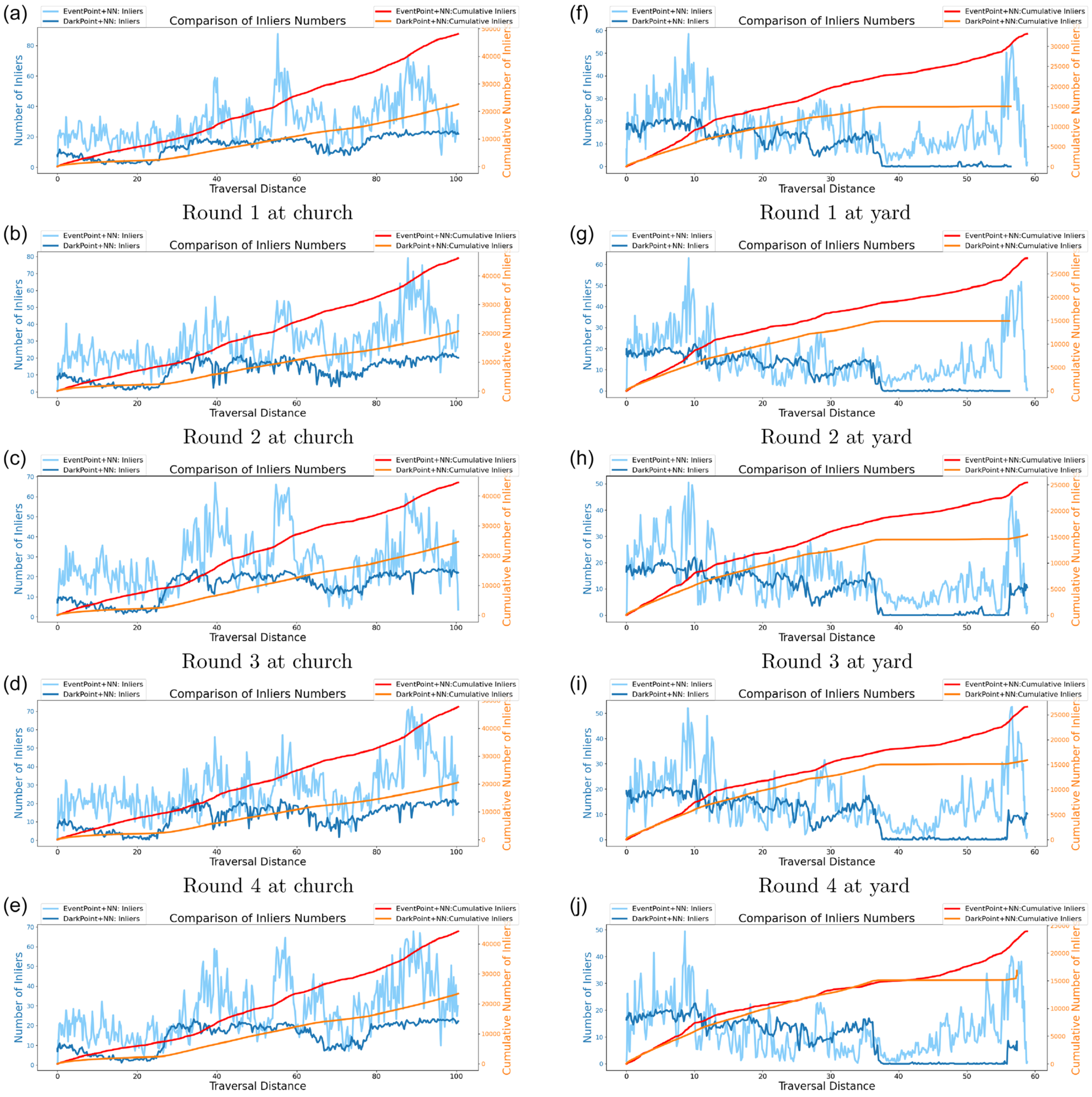
FIGURE 11 | Night-time robot operation around the church. Images taken with a conventional camera (cell phone) showing the physical robot, ambient light levels and physical features on the robot's path during a night-time experiment. The background aerial photo was taken in day-time.

teleoperated, and during the repeat phase, the system performed feature extraction and matching between the corresponding map image and current image to correct localization offset for navigation (as illustrated in Figure 9). To ensure a fair comparison of performance, both methods followed the same teaching trajectory for mapping. The number of matching inliers and trajectories from each repeat operation were recorded to evaluate navigation robustness and accuracy alongside the ATE and RPE metrics described in Part I.

4.2.3 | Results

4.2.3.1 | Feature Matching. First, navigation robustness is assessed by examining the effectiveness of the feature discovery and matching processes. Figure 12 shows a comparison of feature matching performance of the EventPoint and DarkPoint systems in the two test environments. We plot both the instantaneous number of inliers over each of the five test runs and their cumulative totals. A larger number of inliers indicates greater accuracy of localization. Compared to the RGB-based DarkPoint method, our event-based approach achieved approximately 50%–150% more inliers during repeat navigation. Moreover, EventPoint consistently maintained a relatively high number of inliers throughout the entire navigation period across all trajectories, particularly at the church site. Conversely, DarkPoint typically had many fewer inliers, and, at the yard site where the latter part of the path was notably darker, the number of inliers fell-off catastrophically to zero or near-zero levels.

4.2.3.2 | Navigation Accuracy. We evaluated navigation accuracy by comparing the repeat trajectory to the teach path, as illustrated in Figures 13 (church) and 14 (yard). In the church experiment, it is evident that the DarkPoint system exhibited larger deviations from the teach path, particularly noticeable in the first three



Round 1 at church

Round 2 at church

Round 3 at church

Round 4 at church

Round 5 at church

Round 1 at yard

Round 2 at yard

Round 3 at yard

Round 4 at yard

Round 5 at yard

FIGURE 12 | Quantitative inlier comparisons. Left: (a)–(e) results of experiments at church between 2:00 am-3:00 am. Right: (f)–(j) results of experiments at building yard between 00:30 am-2:00 am. EventPoint achieved 113%, 124%, 76%, 134%, and 83% more accumulated inliers than DarkPoint in a, b, c, d, and e, and 147%, 80%, 68%, 69%, and 55% more in f, g, h, i, and j.).

rounds, and only managed to follow approximately two-thirds of the path in the fourth round before veering off-path. Similarly, in the building yard, DarkPoint displayed significant trajectory deviations, with the last two rounds deviating off-path entirely, resulting in navigation failure. In contrast, the EventPoint system successfully repeated the trajectory with good precision in all ten test runs, closely matching the teach path in each case.

As shown in Table 2, our method outperformed DarkPoint with less than half the mean ATE and significantly lower mean RPE at the church site. Importantly, our method achieved superior performance with approximately one-tenth the mean ATE and roughly half the mean RPE at the yard, indicating lower drift during low-light navigation.

5 | Discussion

5.1 | Key Findings

In part I, we showed that navigation accuracy for VTR can be improved by integrating a topological map with a decision-making strategy designed to reduce latencies and trajectory

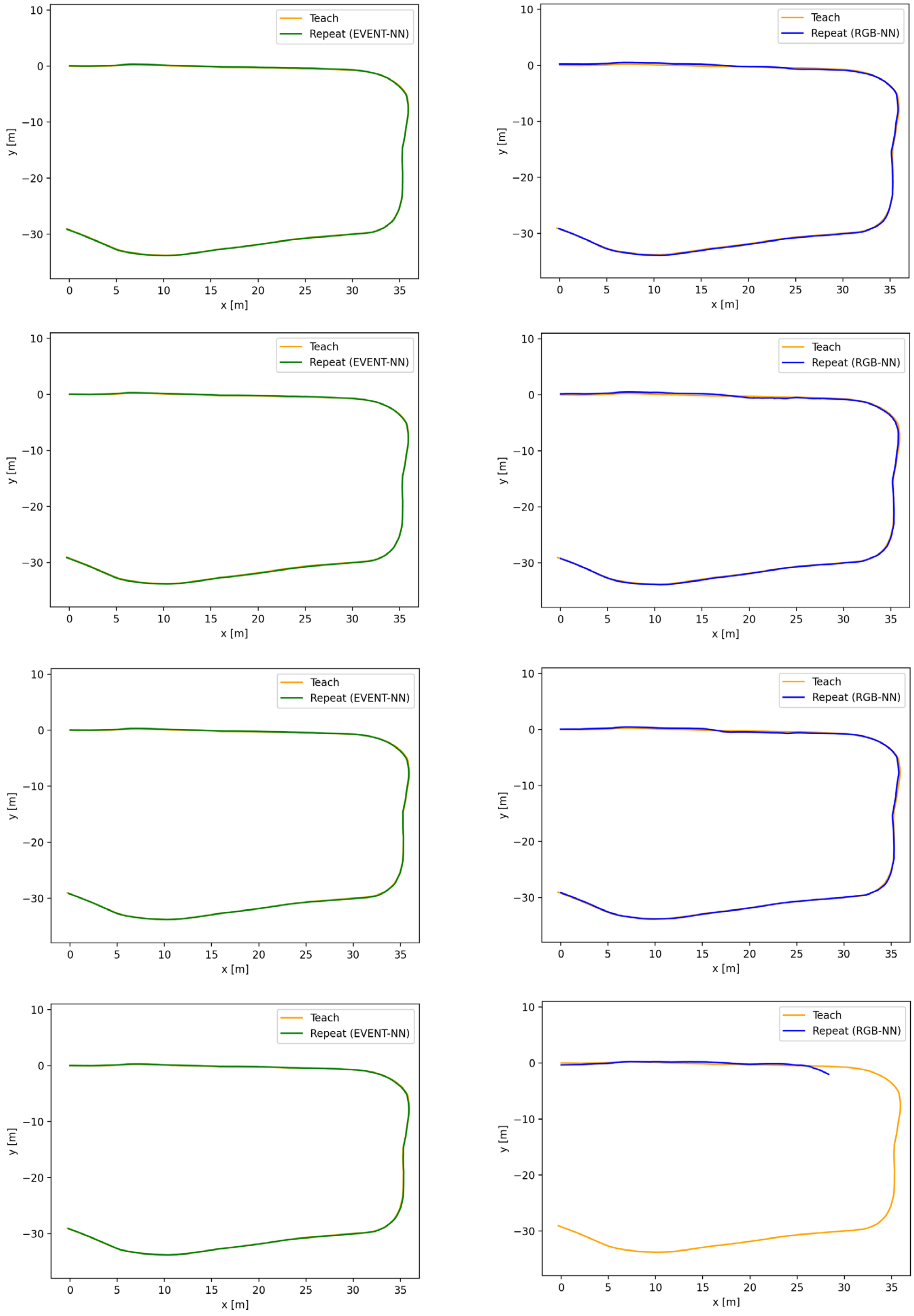


FIGURE 13 | Teach and repeat trajectories at the church site. Left column: Results obtained with the event-based camera and EventPoint. Right column: Results obtained with the RGB camera and DarkPoint.

error. Specifically, during the teaching phase, a local scene descriptor, acquired through deep learning, was coupled with stereo camera imaging and a proportional-integral controller to compensate for inaccuracies in visual matching. This approach facilitates accurate repeat navigation with correction for odometry drift with respect to both orientation and along-route error accumulation using only monocular images.

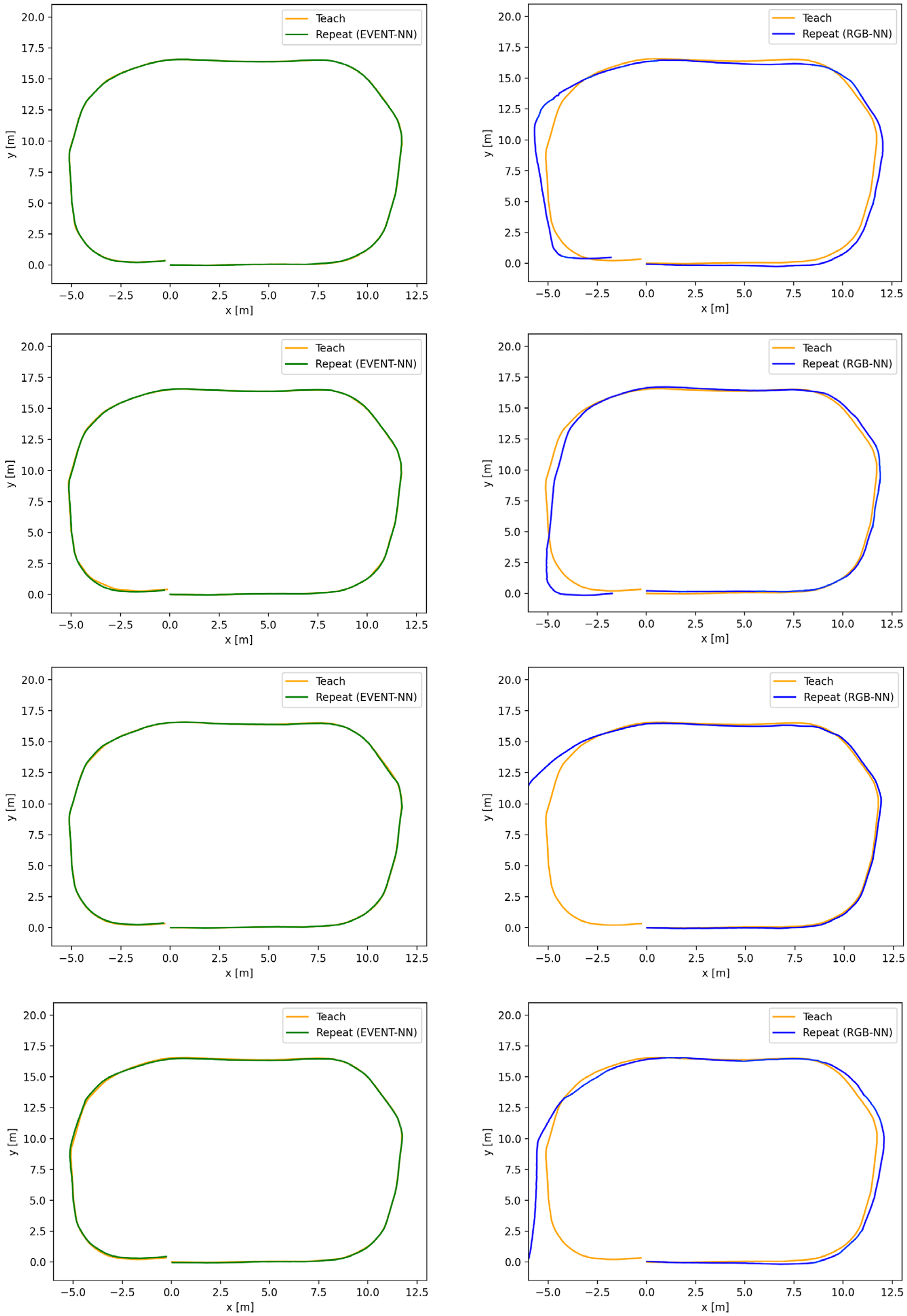


FIGURE 14 | Teach and repeat trajectories at the building yard site. Left column: Results obtained with the event-based camera and EventPoint. Right column: Results obtained with the RGB camera and DarkPoint.

In part II, we adapted this general approach to operate with an event-based camera and an event-based local descriptor model. Experiments in a night-time urban environment demonstrated that this system can provide improved and robust navigation accuracy in low-light environments when compared with a conventional camera paired with a state-of-the-art RGB-based descriptor model.

TABLE 2 | Quantitative evaluation of trajectory accuracy for Event-based VTR navigation. ATE is measured in meters and relative pose error (RPE) in degrees.

Exp	Method	ATE (RMSE)	ATE (mean)	ATE (median)	RPE (RMSE)	RPE (mean)	RPE (median)	result
Church	EventPoint	0.069	0.056	0.047	1.385	1.074	0.820	success
Church	EventPoint	0.075	0.067	0.066	1.519	1.102	0.772	success
Church	EventPoint	0.071	0.059	0.054	1.542	1.126	0.843	success
Church	EventPoint	0.066	0.059	0.054	1.776	1.242	0.816	success
Church	EventPoint	0.063	0.054	0.048	1.434	1.011	0.605	success
Church	DarkPoint	0.084	0.071	0.065	2.738	1.976	1.245	success
Church	DarkPoint	0.141	0.122	0.111	1.434	1.011	0.606	success
Church	DarkPoint	0.173	0.152	0.128	3.027	2.435	1.968	success
Church	DarkPoint	0.168	0.146	0.127	4.669	3.268	2.218	success
Church	DarkPoint	0.322	0.236	0.198	5.222	3.672	2.669	failed
Yard	EventPoint	0.041	0.037	0.035	1.97	1.55	1.16	success
Yard	EventPoint	0.061	0.047	0.040	3.202	2.082	1.319	success
Yard	EventPoint	0.047	0.044	0.045	2.671	1.886	1.232	success
Yard	EventPoint	0.042	0.038	0.037	2.944	2.059	1.318	success
Yard	EventPoint	0.067	0.057	0.057	2.303	1.686	1.104	success
Yard	DarkPoint	0.495	0.435	0.375	6.243	3.593	1.961	success
Yard	DarkPoint	0.460	0.332	0.262	5.486	3.254	1.58	success
Yard	DarkPoint	0.654	0.450	0.294	6.579	3.714	1.818	success
Yard	DarkPoint	0.419	0.296	0.225	3.303	2.686	2.103	failed
Yard	DarkPoint	0.598	0.461	0.369	5.076	2.994	1.602	failed

Overall, high trajectory accuracy is demonstrated for VTR navigation in both indoor and outdoor environments using deep-learned descriptors, whilst the extension to event-based vision extended the capability of VTR navigation to a wider range of challenging environments.

5.2 | Relation to Previous Work

Table 3 summarizes research on VTR navigation since 2010 (for discussion of earlier work see reviews in [5, 60]). Note that much of the prior work on VTR navigation, including our own earlier research [2], has been based on monocular cameras and is faced with the odometry drift problem. The general approach to drift correction has been to align features or images with stored sequences, sometimes using particle filtering or scale propagation to improve matching. As explained in [2], using scale propagation to estimate the depth of features in images has had limited success in terms of performance, whilst increasing the complexity of the decoder architecture (leading to higher computational cost). Previous work using stereo cameras for VTR navigation has predominantly used stereo visual odometry in place of wheel odometry in both the teach and repeat phases. In part 1 of this contribution, we achieved good error drift elimination and fast run-time performance by applying 3D-2D feature for depth estimation. This approach used a stereo camera in the teach phase but only required monocular images for repeat navigation, reducing computational costs for route-following. We are not aware of any other work that has explored this solution. Even where stereo visual odometry is available for both teach and repeat phases, the addition of drift-corrected wheel-based odometry can make for a more robust overall solution to the navigation challenge.

We believe part II of this contribution to be the first demonstration of the use an event-based camera to support VTR navigation in low-light environments.

In previous work [2], we used RGB cameras with deep neural networks for feature extraction in a low-light environment, which can achieve good performance. However, in extremely low-light scenarios, such as during night-time operations, RGB cameras perform poorly, images become grainy due to statistical fluctuations in the number of photons arriving at nearby pixels, and the signal-to-noise ratio is low. Consequently, this approach becomes increasingly error-prone as light-levels fall. By contrast, event cameras detect changes in light intensity rather than relying on absolute illumination levels. They also do so asynchronously on a per-pixel basis. This property allows them to extract richer information in challenging conditions, though of-course they do still require a minimal amount of illumination to function. Although event cameras remain relatively expensive at present, continued research and wider adoption in promising domains such as night-time autonomous driving are expected

TABLE 3 | Approaches to visual teach and repeat navigation.

Authors/Citation	Camera type	Feature detector type[a]	Odometry/drift correction
Krajník et al. (2010, 2018) [7]	Monocular	Classical	Heading correction by feature alignment
Furgale & Barfoot (2010), Churchill & Newman (2013) [5, 43]	Stereo	Classical	Stereo visual odometry
Nitsche et al. (2014) [48]	Monocular	Classical	Particle filter
Paton et al. (2015, 2016, 2017, 2018), Mactavish et al. (2017) [1, 44, 45, 49, 50]	Stereo	Classical	Stereo visual odometry
Dequaire et al. (2016) [51]	Stereo	Classical	Prediction of localization envelope
Rozsypálek et al. (2022) [52]	Monocular	Classical	Image alignment
Siegwart et al. (2011) [25]	Monocular	Image-based	Image alignment
Dall'Osto et al. (2021), Nourizadeh et al. (2024) [10, 53]	Monocular	Image-based	Image alignment
Broughton et al. (2021), Rozsypálek et al. (2022), Rousek et al. (2022), Rousek et al. (2024) [13, 54–56]	Monocular	Image-based	Image alignment by CNN
Rozsypálek et al. (2023) [57]	Monocular	Image-based	Multidimensional particle filter
Gridseth & Barfoot (2022) [58]	Stereo	Neural network	Image alignment with scale prop.
Sun et al. (2021) [2]	Monocular	Neural network	Feature alignment with scale prop.
Zhao et al. (2021) [59]	Monocular	Neural network	Feature alignment with scale prop.
Camara et al. (2020) [3]	Monocular	Neural network	Image alignment by CNN + particle filter

[a] Classical feature detectors include SURF, SIFT, BRIEF and GRIEF (see, e.g [5, 7, 8]); neural network methods include superpoint [2, 59], darkpoint [2], deep keypoint [58]; image-based methods typically use down-scaling and normalization.

to enable mass production. This could substantially reduce their cost in the future as has happened historically with RGB and depth camera sensors.

5.3 | Limitations and Future Work

A clear limitation of the current work is that the 3D-2D error correction method, described in part 1, has still to be combined with the event-based VTR navigation system described in part 2. Whilst this could be achieved by employing a stereo RGB camera alongside the event-based camera, a more natural and appropriate extension would be to use data from the event-camera itself to compute depth information. This could be achieved, for instance, by reconstruction of depth from optic flow via contrast maximization [61]. The high temporal resolution and wide dynamic range of event-based cameras offer the potential to provide depth measures that may be more accurate than those obtained from RGB cameras. Recent research has explored the reconstruction of stereo depth estimates from event-based cameras as we discuss next. This could allow for the use of vision-based odometry and reduce reliance on wheel-based odometry.

To enable event-based depth estimation, two main approaches are commonly employed for multicamera setups (typically two event cameras): *instantaneous stereo* methods and *temporal baseline* methods [62].

Instantaneous stereo methods rely only on the most recent stereo events without requiring explicit knowledge of camera motion. They are primarily used to estimate the depth of independently moving objects as well as static parts of the scene. Within this category, two main families of algorithms exist: model-based and learning-based. Model-based instantaneous stereo matching methods have progressed from early approaches based on simple event accumulation into frames [63–65] to more advanced frameworks that fully exploit the asynchronous nature and high temporal resolution of event data. A central concept in these methods is the use of time surfaces to encode motion information, in combination with geometric and spatio-temporal constraints, to achieve reliable depth estimation (initially sparse, later dense) in both static and dynamic environments [66–69]. In contrast, learning-based algorithms mark a paradigm shift: rather than relying on handcrafted, model-driven strategies, they employ data-driven deep learning approaches to learn optimal representations and processing pipelines for event data in stereo matching [70–73].

Unlike instantaneous approaches, temporal baseline methods aim not to produce a single instantaneous depth ‘snapshot’ of a possibly dynamic scene but to integrate information over extended periods. These methods operate under the assumptions of a static world and known camera motion, fuzing events over time to yield more accurate and consistent depth maps. The model-based algorithms in this category rely on motion priors to improve accuracy. Two representative examples are MC-EMVS [74], which assumes known camera poses, and TSES [75], which assumes constant camera velocity. Both methods exploit temporal integration: MC-EMVS builds and fuzes a 3D evidence volume using precise poses, while TSES warps and evaluates event consistency under a constant velocity assumption. For long-term stereo, ESVO [76] represents a leading framework. It estimates depth by matching time surfaces and fuzes measurements over time using Student-t filters to construct consistent 3D maps.

In the future, we hope to apply one of above event-based stereo solutions to create a fully event-based solution that is robust to drift error in low-light conditions.

The methods described here could also be tested in a wider range of environment and confronted with other forms of environmental change to further evaluate their robustness and utility, particularly for longer-term VTR applications or for challenging environments. Finally, VTR navigation would benefit from richer environmental models than can be provided by feature descriptors alone. For instance, reconstruction and classification of objects and surfaces could aid local and global navigation. Event-based neural radiance fields (EV-NeRF) offer one promising path towards building useful 3D volumetric representations from visual event data [77].

6 | Conclusion

In this article, we have described advances to current methods for VTR navigation, including 3D-2D error correction using a stereo camera and improved low light (night-time) navigation with a monocular event-based camera. Both systems make use of deep-learned descriptor models for visual feature detection and matching, with the event-based system utilizing our recent EventPoint local descriptor model to address low illumination challenges. Furthermore, the event-based system achieved minimal trajectory error (0.047m, 1.385°) and demonstrated robustness in navigating paths even in near-darkness. Overall, high trajectory accuracy was demonstrated for VTR navigation in both indoor and outdoor environments using deep-learned descriptors, whilst the extension to event-based vision extended the capability of VTR navigation to a wider range of challenging environments. More broadly, this VTR approach is flexible and calibration-free, eliminating the need for precise metric mapping and explicit localization, and therefore allowing for robust and computationally-efficient navigation.

Acknowledgments

The authors are grateful to Li Sun for his advice and guidance in developing the VTR navigation system described in part 1 of the paper. The authors are also grateful to Alex Lucas and Alejandro Jimenez-Rodriguez for their help with robot experiments. TJP’s contribution was part-supported by InnovateUK (award no. 10039052) through the UK government’s Horizon Europe guarantee scheme for the EIC Pathfinder project CAVAA (project no. 101071178).

Funding

This work was supported by the InnovateUK (10039052), CAVAA (101071178).

Conflicts of Interest

TJP is a director and shareholder of Consequential Robotics ltd, the company does not stand to benefit from publication of this article. The other authors declare no conflict of interest.

Data Availability Statement

The data that support the findings of this study are openly available in Figshare at https://figshare.com/s/1c6c2ab7eb9cbdc33338, reference number 0.

References

1. M. Paton, K. MacTavish, L.-P. Berczi, S. K. van Es, and T. D. Barfoot, "*I can see for miles and miles: An Extended Field Test of Visual Teach and Repeat 2.0," in Field and Service Robotics: Results of the 11th International Conference*, (Springer, 2018), 415–431.

2. L. Sun, M. Taher, C. Wild, et al., "*Robust and Long-term Monocular Teach and Repeat Navigation using a Single-experience Map," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (IEEE, 2021), 2635–2642.

3. L. G. Camara, M. J. T. Pivonka, C. Gäbert, K. Košnar, and L. Přeučil, "*Accurate and Robust Teach and Repeat Navigation by Visual Place Recognition: A CNN Approach," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (IEEE, 2020), 6018–6024.

4. Z. Chen and S. T. Birchfield, "Qualitative Vision-Based Path following," *IEEE Transactions on Robotics and Automation* 25, no. 3 (2009): 749–754.

5. P. Furgale and T. D. Barfoot, "Visual Teach and Repeat for Long-Range Rover Autonomy," *Journal of Field Robotics* 27, no. 5 (2010): 534–560.

6. K. Kidono, J. Miura, and Y. Shirai, "Autonomous Visual Navigation of a Mobile Robot Using a Human-Guided Experience," *Robotics and Autonomous Systems* 40, no. 2-3 (2002): 121–130.

7. T. Krajník, J. Faigl, V. Vonásek, K. Košnar, M. Kulich, and L. Přeučil, "Simple yet Stable Bearing-Only Navigation," *Journal of Field Robotics* 27, no. 5 (2010): 511–533.

8. T. Krajník, F. Majer, L. Halodová, and T. Vintr, "*Navigation Without Localisation: Reliable Teach and Repeat Based on the Convergence Theorem," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (IEEE, 2018), 1657–1664.

9. E. Royer, M. Lhuillier, M. Dhome, and J.-M. Lavest, "Monocular Vision for Mobile Robot Localization and Autonomous Navigation," *International Journal of Computer Vision* 74, no. 3 (2007): 237–260.

10. D. Dall'Osto, T. Fischer, and M. Milford, "*Fast and Robust Bio-inspired Teach and Repeat Navigation," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (IEEE, 2021), 500–507.

11. P. King, A. Vardy, and A. L. Forrest, "Teach-and-Repeat Path following for an Autonomous Underwater Vehicle," *Journal of Field Robotics* 35, no. 5 (2018): 748–763.

12. M. Warren, M. Greeff, B. Patel, J. Collier, A. P. Schoellig, and T. D. Barfoot, "There's No Place like Home: Visual Teach and Repeat for Emergency Return of Multirotor UAVs during GPS Failure," *IEEE Robotics and Automation Letters* 4, no. 1 (2019): 161–168.

13. Z. Rozsypálek, G. Broughton, P. Linder, et al., "Contrastive Learning for Image Registration in Visual Teach and Repeat Navigation," *Sensors* 22, no. 8 (2022): 2975.

14. G. Gallego, T. Delbruck, G. Orchard, et al., "Jorg Conradt, Kostas Daniilidis, and Davide Scaramuzza. Event-Based Vision: A Survey," *IEEE Transactions on Pattern Analysis & Machine Intelligence* 44, no. 01 (2022): 154–180.

15. P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128 × 128 120db 15 μs Latency Asynchronous Temporal Contrast Vision Sensor," *IEEE Journal of Solid-State Circuits* 43, no. 2 (2008): 566–576.

16. X. Clady, J.-M. Maro, S. Barre, and R. B. Benosman, "A Motion-Based Feature for Event-Based Pattern Recognition," *Frontiers in Neuroscience* 594 (2017): 10:594.

17. Z. Huang, L. Sun, C. Zhao, S. Li, and S. Su, "*Eventpoint: Self-supervised Interest Point Detection and Description for Event-based Camera," in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, (IEEE, 2023), 5396–5405.

18. B. Ramesh, H. Yang, G. Orchard, N. A. L. Thi, S. Zhang, and C. Xiang, "DART: Distribution Aware Retinal Transform for Event-Based Cameras," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20 (2020): 2767–2780.

19. V. Vasco, A. Glover, and C. Bartolozzi, "*Fast Event-based Harris Corner Detection Exploiting The Advantages of Event-driven Cameras," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (IEEE, 2016), 4144–4149.

20. A. M. Zhang and L. Kleeman, "Robust Appearance based Visual Route Following for Navigation in Large-scale Outdoor Environments," *The International Journal of Robotics Research* 28, no. 3 (2009): 331–356.

21. G. Blanc, Y. Mezouar, and P. Martinet, "*Indoor Navigation of a Wheeled Mobile Robot Along Visual Routes," in IEEE International Conference on Robotics and Automation (ICRA)*, (IEEE, 2005), 3354–3359.

22. Y. Matsumoto, M. Inaba, and H. Inoue, "*Visual Navigation using View-sequenced Route Representation," in IEEE International Conference on Robotics and Automation (ICRA)*, (IEEE, 1996), 83–88.

23. S. Segvic, A. Remazeilles, A. Diosi, and F. Chaumette, "*Large Scale Vision-based Navigation without an Accurate Global Reconstruction," in IEEE Conference on Computer Vision and Pattern Recognition (ICCVPR)*, (IEEE, 2007), 1–8.

24. S. Lowry, N. Sünderhauf, P. Newman, et al., "Visual Place Recognition: A Survey," *IEEE Transactions on Robotics* 32, no. 1 (2016): 1–19.

25. R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, Introduction to Autonomous Mobile Robots, 2nd ed. (MIT Press, 2011).

26. D. DeTone, T. Malisiewicz, and A. Rabinovich, "*Superpoint: Self-supervised Interest Point Detection and Description," in IEEE Conference on Computer Vision and Pattern Recognition Workshops*, (IEEE, 2018), 224–236.

27. B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle Adjustment- A Modern Synthesis," in *Vision Algorithms: Theory and Practice,* LNCS 1883, (Springer, 2000): 298–372.

28. T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, LIO-SAM: Tightly-coupled lidar inertial odometry via smoothing and mapping. arXiv 2007.00258,2020.

29. A. Amir, B. Taba, D. Berg, et al., "*A Low Power, Fully Event-based Gesture Recognition System," in IEEE Conference on Computer Vision and Pattern Recognition (ICCVPR)*, (IEEE, 2017), 7243–7252.

30. T. Delbruck and M. Lang, Robotic Goalie with 3 ms Reaction Time at 4% CPU Load Using Event-Based Dynamic Vision Sensor, *Frontiers in Neuroscience* 7 (2013): 223.

31. A. Glover and C. Bartolozzi, "*Event-driven Ball Detection and Gaze Fixation in Clutter," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (IEEE, 2016), 2203–2208.

32. R. Benosman, C. Clercq, X. Lagorce, S.-H. Ieng, and C. Bartolozzi, "Event-Based Visual Flow," *IEEE Transactions on Neural Networks and Learning Systems* 25, no. 2 (2013): 407–417.

33. J. Hagenaars, F. Paredes-Vallés, and G. De Croon, Self-Supervised Learning of Event-Based Optical Flow with Spiking Neural Networks, *Advances in Neural Information Processing Systems* 34 (2021): 7167–7179.

34. A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, EV-FlowNet: Self-Supervised Optical Flow Estimation for Event-Based Cameras. arXiv1802.06898, 2018.

35. W. Guan and P. Lu, "*Monocular Event Visual Inertial Odometry Based on Event-corner using Sliding Windows Graph-based Optimization," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (IEEE, 2022), 2438–2445.

36. Z. Liu, D. Shi, R. Li, and S. Yang, "ESVIO: Event-Based Stereo Visual-Inertial Odometry," *Sensors* 23, no. 4 (1998): 2023.

37. A. R. Vidal, H. Rebecq, T. Horstschaefer, and D. Scaramuzza, "Ultimate SLAM? Combining Events, Images, and IMU for Robust Visual SLAM in HDR and High-Speed Scenarios," *IEEE Robotics and Automation Letters* 3, no. 2 (2018): 994–1001.

38. W. Guan, P. Chen, Y. Xie, and P. Lu, Pl-Evio: Robust monocular event-based vi sual inertial odometry with point and line features. arXiv preprint arXiv:2209.12160,2022.

39. E. Mueggler, H. Rebecq, G. Gallego, T. Delbruck, and D. Scaramuzza, "The Event-Camera Dataset and Simulator: Event-Based Data for Pose Estimation, Visual Odometry, and Slam," *The International Journal of Robotics Research* 36, no. 2 (2017): 142–149.

40. Y. Zhou, G. Gallego, H. Rebecq, L. Kneip, H. Li, and D. Scaramuzza, "*Semi-dense 3d Reconstruction with a Stereo Event Camera," in European conference on computer vision (ECCV)*, (CVF, 2018), 235–251.

41. J. Zhang, X. Yu, H. Sier, H. Zhang, and T. Westerlund, "*Event-based Sensor Fusion and Application on Odometry: A Survey," in IEEE 6th International Conference on Image Processing, Applications and Systems (IPAS)*, (IEEE, 2025), 1–6.

42. M. Gehrig, W. Aarents, D. Gehrig, and D. Scaramuzza, "DSEC: A Stereo Event Camera Dataset for Driving Scenarios," *IEEE Robotics and Automation Letters* 6, no. 3 (2021): 4947–4954.

43. W. Churchill and P. Newman, "Experience-Based Navigation for Long-Term Localisation," *The International Journal of Robotics Research* 32, no. 14 (2013): 1645–1661.

44. K. MacTavish, M. Paton, and T. D. Barfoot, "*Visual Triage: A Bag-of-words Experience Selector for Long-term Visual Route Following," in IEEE International Conference on Robotics and Automation (ICRA)*, (IEEE, 2017), 2065–2072.

45. M. Paton, K. MacTavish, M. Warren, and T. D. Barfoot, Bridging the Appearance Gap: Multi-Experience Localization for Long-Term Visual Teach and Repeat, *IROS*, (IEEE, 2016), 1918–1925.

46. L. Halodová, E. Dvořráková, F. Majer, et al., "*Predictive and Adaptive Maps for Long-term Visual Navigation in Changing Environments," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (IEEE, 2019), 7033–7039.

47. N. Zhang, M. Warren, and T. D. Barfoot, "*Learning Place-and-time-dependent Binary Descriptors for Long-term Visual Localization," in IEEE International Conference on Robotics and Automation (ICRA)*, (IEEE, 2018), 828–835.

48. M. Nitsche, T. Pire, T. Krajník, M. Kulich, and M. Mejail, Monte Carlo Localization for Teach-and-Repeat Feature-Based Navigation, *Advances in Autonomous Robotics Systems*, eds. M. Mistry, A. Leonardis, M. Witkowski and C. Melhuish, (Springer International Publishing, 2014), 13–24.

49. M. Paton, K. MacTavish, C. J. Ostafew, and T. D. Barfoot, "*It's Not Easy Seeing Green: Lighting-resistant Stereo Visual Teach & Repeat using Color-constant Images," in IEEE International Conference on Robotics and Automation (ICRA)*, (IEEE, 2015), 1519–1526.

50. M. Paton, F. Pomerleau, K. MacTavish, C. J. Ostafew, and T. D. Barfoot, "Expanding the Limits of Vision-Based Localization for Long-Term Route-Following Autonomy," *Journal of Field Robotics* 34, no. 1 (2017): 98–122.

51. J. Dequaire, C. H. Tong, W. Churchill, and I. Posner, "*Off the Beaten Track: Predicting Localisation Performance In Visual Teach and Repeat," in IEEE International Conference on Robotics and Automation (ICRA)*, (IEEE, 2016), 795–800.

52. Z. Rozsypálek, G. Broughton, and P. Linder, *Tomáš Rouček, Keerthy Kusumam, and Tomáš Krajník. Semi-Supervised Learning for Image Alignment in Teach and Repeat Navigation*, in 37th ACM/SIGAP Symposium on Applied Computing, (ACM, 2022), 731–738.

53. P. Nourizadeh, M. Milford, and T. Fischer, Teach and repeat navigation: A robust control approach. arXiv: 2309.15405,2024.

54. G. Broughton, P. Linder, T. Rouček, T. Vintr, and K. Tomáš, "*Robust Image Alignment for Outdoor Teach-and-repeat Navigation," in European Conference on Mobile Robots (ECMR)*, (IEEE, 2021), 1–6.

55. T. Rouček, A. S. Amjadi, Z. Rozsypálek, et al., "Self-Supervised Robust Feature Matching Pipeline for Teach and Repeat Navigation," *Sensors* 22, no. 8 (2022): 2836.

56. T. Rouček, Z. Rozsypálek, J. U. Jan Blaha, and T. Krajník, "Predictive Data Acquisition for Lifelong Visual Teach, Repeat and Learn," *IEEE Robotics and Automation Letters* 9, no. 11 (2024): 10042–10049.

57. Z. Rozsypálek, T. Rouček, T. Vintr, and T. Krajník, "Multidimensional Particle Filter for Long-Term Visual Teach and Repeat in Changing Environments," *IEEE Robotics and Automation Letters* 8, no. 4 (2023): 1951–1958.

58. M. Gridseth and T. D. Barfoot, "Keeping an Eye on Things: Deep Learned Features for Long-Term Visual Localization," *IEEE Robotics and Automation Letters* 7, no. 2 (2022): 1016–1023.

59. C. Zhao, L. Sun, T. D. Tomáš Krajník, and Z. Yan, "*Monocular Teach-and-repeat Navigation using A Deep Steering Network With Scale Estimation," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (IEEE,2021), 2613–2619.

60. A. Krawciw and T. D. Barfoot, Local Maps Are All You Need: A Review of Topometric Teach and Repeat Navigation., *Annual Review of Control, Robotics, and Autonomous Systems* 9 (2025):

61. S. Shiba, Y. Klose, Y. Aoki, and G. Gallego, "Secrets of Event-Based Optical Flow, Depth and Ego-Motion Estimation by Contrast Maximization," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46, no. 12 (2024): 7742–7759.

62. S. Ghosh and G. Gallego, "Event-Based Stereo Depth Estimation: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 47, no. 10 (2025): 9130–9149.

63. A. N. Belbachir, M. Litzenberger, S. Schraml, et al., "*Care: A Dynamic Stereo Vision Sensor System for Fall Detection," in IEEE international Symposium on Circuits and Systems (ISCAS)*, (IEEE, 2012), 731–734.

64. M. Dominguez-Morales, E. Cerezuela-Escudero, A. Jimenez-Fernandez, et al., "*Image Matching Algorithms in Stereo Vision using Address-event-representation: A Theoretical Study and Evaluation of the Different Algorithms," in International Conference on Signal Processing and Multimedia Applications*, (IEEE, 2011), 1–6.

65. S. Schraml, A. N. Belbachir, N. Milosevic, and P. Schön, "*Dynamic Stereo Vision System for Real-time Tracking," in IEEE International Symposium on Circuits and Systems (ISCAS)*, (IEEE, 2010), 1409–1412.

66. S.-H. Ieng, J. Carneiro, M. Osswald, and R. Benosman, "Neuromorphic Event-Based Generalized Time-Based Stereovision," *Frontiers in Neuroscience* 12 (2018): 442.

67. X. Lagorce, G. Orchard, F. Galluppi, B. E. Shi, and R. B. Benosman, "HOTS: A Hierarchy of Event-Based Time-Surfaces for Pattern Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, no. 7 (2016): 1346–1359.

68. M. Muglikar, G. Gallego, and D. Scaramuzza, "*ESL: Event-based Structured Light," in International Conference on 3D Vision (3DV)*, (IEEE, 2021), 1165–1174.

69. Z. Xie, S. Chen, and G. Orchard, "Event-Based Stereo Depth Estimation Using Belief Propagation.", *Frontiers in Neuroscience* 11 (2017): 535.

70. H. Cho, J. Cho, and K.-J. Yoon, "*Learning Adaptive Dense Event Stereo from the Image Domain," in IEEE/CVF Conference on Computer Vision and Pattern Recognition (ICCVPR)*, (IEEE, 2023), 17797–17807.

71. J. Gu, J. Zhou, R. S. W. Chu, et al., "Self-Supervised Intensity-Event Stereo Matching arXiv preprint arXiv : 211.00509,2022.

72. M. Lin, C. Zhang, C. He, and L. Yu, "Learning Parallax for Stereo Event-Based Motion Deblurring," *IEEE Transactions on Circuits and Systems for Video Technology* 35, no. 10 (2025): 10032–10046.

73. P. Liu, G. Chen, Z. Li, H. Tang, and A. Knoll, "*Learning Local Event-based Descriptor for Patch-based Stereo Matching," in International Conference on Robotics and Automation (ICRA)*, (IEEE, 2022), 412–418.

74. S. Ghosh and G. Gallego, "Multi-Event-Camera Depth Estimation and Outlier Rejection by Refocused Events Fusion," *Advanced Intelligent Systems* 4, no. 12 (2022): 2200221.

75. A. Z. Zhu, Y. Chen, and K. Daniilidis, "*Realtime Time Synchronized Event-based Stereo," in European Conference on Computer Vision (ECCV)*, (CVF, 2018), 433–447.

76. J. Niu, S. Zhong, X. Lu, S. Shen, G. Gallego, and Y. Zhou, "ESVO2: Direct Visual-Inertial Odometry with Stereo Event Cameras," *IEEE Transactions on Robotics* 41 (2025): 2164–2183.

77. I. Hwang, J. Kim, and Y. M. Kim, "*Ev-NeRF: Event based Neural Radiance Field," in 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, (IEEE Computer Society, 2023), 837–847.