From Motion to Localization: Cross-view Optimization with Stationary Event and RGB Cameras for Enhanced Pose Estimation

YUKUN ZHAO, The Hong Kong University of Science and Technology, Hong Kong
XINYUAN SONG, Peking University, China
HUAJIAN HUANG, The Hong Kong University of Science and Technology, Hong Kong
TRISTAN BRAUD, The Hong Kong University of Science and Technology, Hong Kong

Applications such as Augmented Reality (AR) require accurate device positioning to minimize alignment errors. While visual positioning techniques offer high accuracy, their performance can degrade due to environmental changes like lighting variations and object movements. This paper introduces a new approach to visual positioning, relying on a stationary joint event/RGB sensing platform to track scene dynamics in real-time. This platform is at the core of a localization pipeline to predict the pose of user devices. First, a cross-modal object tracker matches dynamic objects between RGB and event images captured by the platform. These objects contribute to building a dynamic map, combined with the initial static 3D Structure from Motion (SfM) model to form a global feature map. Finally, a cross-view pose optimizer estimates pose uncertainties between modalities to refine and improve localization accuracy. To validate our approach, we collect a large-scale dataset over three scenes to account for typical AR scenarios where dynamics can affect the quality of visual positioning. We contribute this dataset to the community for future research on scene dynamics. Our approach shows significant improvement over existing methods, reducing translation and rotation errors by 12.9% and 13.4%, respectively, for weekly data over 4 weeks, and by 38.5% and 16.2% for monthly data over 4 months, compared to HLoc (SP+SG). It also reduces performance degradation by up to 50% after only 4 weeks.

CCS Concepts: • **Computing methodologies → Computer vision problems**; **Mixed / augmented reality**; • **Human-centered computing → Ubiquitous and mobile computing systems and tools**.

Additional Key Words and Phrases: Augmented Reality, Event Camera, Visual Positioning

ACM Reference Format:
Yukun Zhao, Xinyuan Song, Huajian Huang, and Tristan Braud. 2025. From Motion to Localization: Cross-view Optimization with Stationary Event and RGB Cameras for Enhanced Pose Estimation. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 9, 4, Article 243 (December 2025), 25 pages. https://doi.org/10.1145/3770631

1 Introduction

Accurate device positioning is a critical challenge in various applications, ranging from gesture detection [52] to robotics and augmented reality (AR) [3]. As these applications become increasingly pervasive, they require consistent and precise positioning over time across extensive areas. Visual positioning techniques can achieve high accuracy, which is essential in applications such as AR, where even minor misalignments can significantly

Authors' Contact Information: Yukun Zhao, The Hong Kong University of Science and Technology, Hong Kong, yzhaoeg@connect.ust.hk; Xinyuan Song, Peking University, Beijing, China, songxinyuan@stu.pku.edu.cn; Huajian Huang, The Hong Kong University of Science and Technology, Hong Kong, hhuangbg@connect.ust.hk; Tristan Braud, The Hong Kong University of Science and Technology, Hong Kong, braudt@ust.hk.

ACM 2474-9567/2025/12-ART243
https://doi.org/10.1145/3770631

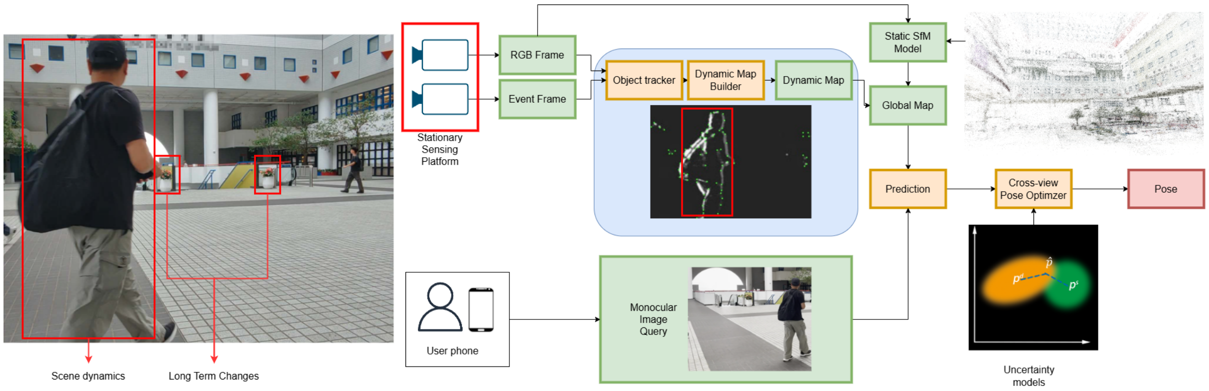


Fig. 1. Cross-view pose optimization over large-scale scenes. The scene presents both scene dynamics (e.g., people moving around), and long-term changes (e.g., objects moving, plants growing). A stationary sensing platform embedding both event and RGB images monitors the changes in the space in real-time. The data from both cameras is matched through a cross-modal object tracker, and integrated into a dynamic map that reflect the scene dynamics at the given timestamp. The dynamic map is combined with the static SfM model to create a global map as the reference model onto which the user device's pose is predicted. Finally, a cross-view pose optimizer estimates the pose uncertainty on the static and dynamic maps to further refine the user pose.

impact the user experience. However, the performance of these techniques is heavily influenced by changes in the visual features of the environment. Factors such as variations in illumination, the movement of humans, objects, and furniture, and seasonal changes can alter the visual appearance of a scene, leading to substantial localization errors[56]. As such, most applications either operate over short timespans, rely on frequent data capture with continual learning [51], utilize additional mobile sensors [49], or focus on immutable points in the environment [27]. These constraints significantly limit their applicability to real-world scenarios.

Event cameras are bio-inspired vision sensors that function fundamentally differently from conventional cameras. The raw output of event cameras consists of a sequence of asynchronous events. These events represent discrete pixel-wise brightness changes corresponding to scene illumination variations. Event cameras offer several advantages over conventional cameras, including high temporal resolution, wide dynamic range, the absence of motion blur, and low energy consumption. However, only a few studies have employed event cameras for camera relocalization, with limited success [26, 30, 42]. Although event cameras have limited efficiency for direct camera pose estimation, they can efficiently record a scene's dynamics while filtering out static data, such as the geometric information of architectural structures. The changes in brightness recorded by an event camera may thus complement the image data captured by conventional cameras to improve the accuracy of camera relocalization. However, due to nonlinearity and uncertainties in the event generation model, model-based fusion approaches are fragile and sensitive to hyperparameter tuning [38, 55].

This paper introduces a cross-view pose optimization method that combines stationary event and RGB cameras to achieve stable and dynamic-aware localization. Figure 1 summarizes the primary steps of the method. We develop a stationary event/RGB sensing platform that continuously updates the dynamics of a given space in real-time. This capture platform is integrated with a localization pipeline to relocalize the queries of mobile users in the space. The sensing platform consists of an event camera and an RGB camera sensor, with data capture hardware-synchronized. The cameras are equipped with wide-angle lenses and positioned adequately to monitor the space. The localization pipeline comprises three modules: a cross-modal object tracker, a dynamic map constructor, and a cross-view pose optimizer. The cross-modal object tracker identifies and matches dynamic objects between the RGB and event images. The objects' bounding boxes are then provided to the dynamic map constructor. This module uses recurrent asynchronous multimodal networks to estimate the dynamic objects' depth and incorporates them into a 3D feature map. The dynamic feature map is combined with a static map, which consists of a 3D Structure from Motion (SfM) model generated offline using SuperPoint [5] and HF-Net [34], and database images. Together, these form the global feature map, onto which monocular RGB image queries are registered for localization. However, discrepancies between the static SfM model and the dynamic feature map can still result in inaccuracies. The cross-view optimizer estimates the pose uncertainty for the prediction on the static map and the dynamic map. As the static and dynamic maps are constructed differently, their pose uncertainties exhibit distinct characteristics, enabling a more accurate global pose estimation.

To validate the effectiveness of our proposed method, we collected a large-scale dataset across three scenes of various sizes, ranging from a small-scale open-space office to a large semi-open atrium at a local university. The dataset includes initial images for building the static SfM model, RGB/event camera pairs from the sensing platform, and mobile phone images and sensor data used as user queries. To simulate realistic scenarios for AR use cases, the mobile phone data is captured through user-defined traces, rather than specific routes in the space. We contribute the resulting event and frame dataset along with user VIO (ARKit[1]) and other sensor data (gyroscope, accelerometer, magnetometer). The dataset presents both the initial training data and the testing data, collected weekly over 4 weeks, and monthly over 3 months afterwards. Our evaluation of the dataset indicates that, on average, EVS-CVPO achieves a 12.9% reduction in translation error and a 13.4% reduction in rotation error compared to HLoc (SP+SG) [34] for data captured weekly over 4 weeks. Furthermore, for data captured monthly over 4 months, EVS-CVPO shows an average improvement of 38.5% in translation error and 16.2% in rotation error when compared to HLoc (SP+SG). Furthermore, our system maintains resilience to long-term changes by integrating scene dynamics, resulting in up to 50% less degradation in performance compared to HLoc after only 4 weeks in very dynamic environments.

Overall, the main contributions of this paper are as follows:

(1) We propose a new localization framework which leverages the respective advantages the structure-based visual positioning and event-based dynamic map feature matching method. This framework includes:
  (a) A stationary event/RGB sensing platform to monitor the dynamics of a space in real-time.
  (b) A visual-based cross-view localization pipeline leveraging the data from the sensing platform to account for the short- and long-term changes in a space in visual positioning.
  (c) A novel cross-view pose refinement method that builds pose uncertainty models to optimize for a more accurate pose, improving localization accuracy.
(2) We evaluate the proposed approach through real-life evaluation on three datasets collected over various periods of time to reflect typical scenarios in augmented reality with high scene dynamics. Our proposed approach achieves higher accuracy than SOTA techniques and improves robustness to scene changes over time.
(3) We contribute the evaluation dataset as the first event camera dataset with complete AR VIO data, and open source the implementation code of this paper, for the benefit of the broader research community.

2 Related Works

This section first discusses the most prominent works on visual localization. We then extend to the fusion of event camera with traditional RGB sensors, and review applications to object tracking and depth estimation.

[1] https://developer.apple.com/augmented-reality/arkit/

2.1 Visual Localization

Visual localization can take many different forms depending on the conditions and sensors available. In this work, we focus on pose estimation using a single monocular RGB image as input. Simultaneous Localization and Mapping (SLAM) is a popular solution for estimating the camera pose relative to prior localization. Several works explore using event cameras for this purpose [9, 23, 48]. However, most SLAM solutions track the camera pose with respect to a prior pose. Although some solutions integrate techniques to estimate the camera pose without prior knowledge, these techniques usually focus on loop closing and restoring the experience in case of lost tracking. In this paper, we focus on predicting the camera pose in absolute coordinates. Two primary approaches have been applied in the literature:

Structure-based. Structure-based approaches establish correspondences between 2D features in query images and 3D scene coordinates. This can be done indirectly through feature extraction and matching [5, 35, 36], or directly via scene coordinate regression [1]. Hierarchical Localization (HLoc) [34] incorporates image retrieval to reduce the search space, achieving high accuracy on visual localization benchmarks. However, it requires static pre-built models and mapping image databases as reference. As such, dynamic occlusion and long-term changes in the scene significantly affect the localization performance.

Learning-based. Absolute Pose Regressors (APR) are end-to-end learning-based methods that directly estimate the absolute camera pose from input image. The seminal work is introduced by PoseNet (PN) [14], while further modifications include network architectures [40, 44, 45], and training strategies [13]. MS-Transformer (MS-T) [40] extends the single-scene APRs to multiple-scene APRs. APRs provide faster inference than structure-based methods without the knowledge of 3D models and 2D-3D correspondences, on the other hand, with lower accuracy and robustness. Sattler et al. [37] shows that APR methods cannot generalize well beyond their training set via image retrieval baseline. To address the problem, Arthur Moreau et al. [22] and Tayyab Naseer and Wolfram Burgard [24] expands the training set with synthetic dataset for better generalization. DFNet [4] applies unlabeled data with NeRF synthesis in a semi-supervised manner. However, APRs still suffer from limited generalizability [25], and synthetic dataset are unable to interpret dynamics in real-life scenarios. Using unlabeled data from the test set to finetune the APR network is also impractical.

To the best of our knowledge, all existing visual localization methods face significant challenges related to dynamic occlusion and long-term changes in the scene, with typical solution being focusing on established static parts of the scene [27], relying on additional sensor data [49], or using continual learning with periodic data collection [51]. This paper adopts a cross-view monocular localization pipeline that leverages a fixed event infrastructure that preserves visual contextual information to account for dynamic changes.

2.2 Event-image Fusion

The integration of event streams and RGB images encounters numerous challenges related to data formats, spatial dimensions, temporal aspects, and information characteristics. Given the complementarity of event data and frames, various algorithms have been developed to leverage the advantages of both modalities through fusion. Current fusion approaches can be broadly classified into two primary categories: (1) pixel-level and (2) feature-level methods. Pixel-level approaches [21, 41, 43, 50] align events and images at the pixel level, utilizing the imaging constraints inherent to event cameras for fusion; these methods are predominantly employed in low-level vision tasks. In contrast, feature-level methods [20, 47] align events and images within the feature space, capitalizing on spatial-temporal relationships for fusion, and are frequently applied in middle-level and high-level vision tasks. Since visual localization aims to estimate the user pose in 3D space, it is essential to consider the complementary use of both modalities in imaging, the spatial alignment of data from different viewpoints, and the spatial-temporal consistency of dynamic elements. This introduces challenges that surpass those encountered in previous tasks. In this case, we review fusion works that relate to different modules of our pipeline, namely object tracking and monucular depth estimation.

Object tracking. Owing to the event's innate characteristics and superiority for object tracking, event-based tracking has been a progressively prevalent subject for research in recent years. Event-driven tracking approaches [17, 53, 57] facilitate the individual and asynchronous updating of features with each event. However, tracking with probabilistic data association [53] suffers from limitations in accuracy and computational speed, hindering real-time implementation. The event pattern tracking utilizing Gaussian Processes [17] exhibits enhanced accuracy, but also remains computationally intensive. Zhiyu Zhu et al. [57] proposes to utilize inherent motion information of event data to achieve effective object tracking, but is limited in finding complementary information from RGB data. Due to the extraordinary correlation modeling ability of vision Transformer (ViT), our cross-modal tracker module employs orthogonal high-rank augmentation [58] to adaptively process different modalities in realtime, outputting attention objects for further dynamic map construction with depth estimation.

Depth estimation. Many contemporary event-based depth estimation algorithms depend significantly on ideal sensor models [7], which degrade under non-ideal conditions. Therefore, state-of-the-art methods often adopt a data-driven approach. Among these, many [11, 31] utilize recurrent architectures to enhance prediction accuracy. Yuhuang Hu et al. [12] and Stefano Pini et al. [28] fuse both modalities by synchronizing and concatenating the inputs before passing them to a feed-forward network. Although this strategy offers improvements over processing each input individually, it limits the leveraged temporal context of events. The RAM network we employ addresses these limitations by leveraging mature data-driven models for robust estimation, maximizing the temporal context through the use of recurrency, and introducing individual state update rules for each input modality for high temporal resolution of events, which is perfectly suitable for our maintenance on the dynamic feature map in 3D space.

3 Motivation and Use Case

Markerless AR on mobile devices has the potential to enhance user experiences in entertainment venues such as exhibitions and shows, as well as in other AR-assisted services, by utilizing visual feature-based registration techniques. However, existing platforms enabling markerless AR often rely on a static 3D model of the environment established prior to the experience. Therefore, pure vision-based methods encounter difficulties in these contexts because of the presence of dynamic objects and long-term changes that affect the visual appearance of events. Although some frameworks such as Niantic Lightship require users to capture the environment in different lighting conditions [2], most solutions do not accommodate changes in the scanned areas, resulting in a decline in both performance and accuracy over time. Large-scale environments pose an additional challenge for depth mapping. Depth maps generated by mobile devices, even with additional sensors like the iPhone's LiDAR, tend to be insufficient and often only accurate at short ranges. Additionally, these large-scale settings introduce a greater likelihood of variability. Consequently, the trained models may lack robustness in environments that undergo significant alterations over time.

Let us consider the case of AR navigation at the atrium of a local university (see Figure 2). Located right at the entrance of the campus and serving as its logical center, the atrium experiences significant activity, resulting in short-, medium-, and long-term changes. At lunchtime, most students move from the classrooms to the canteens through the atrium. The large, rapidly moving crowds can greatly obstruct the device's camera view for a user at ground level, potentially reducing localization accuracy. Additionally, the atrium has other dynamic elements, such as trees, that can affect feature-based methods. The atrium is primarily lit by sunlight, which causes significant changes in exposure depending on the weather and time of day. These variations can affect localization accuracy if not accounted for during data capture. Finally, many events occur in this location, including student

[2] https://lightship.dev/docs/ardk/how-to/vps/tooling/create_vps_activated_location/

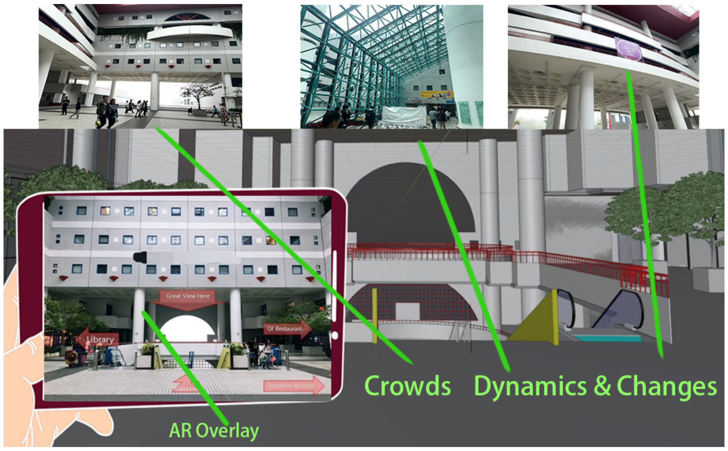


Fig. 2. Concrete usage scenario concept. Providing accurate localization for mobile AR. The large-scale scene Atrium is characterized by high traffic and significant environmental variations over time, where numerous events are generated.

society activities and career fairs. These events often bring about considerable medium- to long-term changes. For instance, a stage may be set up at the center for performances by bands or dance societies, while the career fair will require the setup of many booths and signs. With its central location on campus, the atrium is a key area for deploying AR navigation. However, the magnitude and frequency of changes significantly affect the applicability of such applications.

To enable reliable device pose estimation in these scenarios, several critical challenges must be addressed, particularly concerning lighting, changes in the environment, and occlusion. First, a localization technique must demonstrate resilience to variations in lighting and remain unaffected by scene dynamics. For instance, a street scene should be recognized both at day time and night time. Second, the system must effectively operate in an environment with significant short-to-long term occlusion. Short-term occlusion include passing pedestrians and cars, who, in large amounts, can not only significantly obstruct the scene, but also cause motion blur due to their rapid movement. such as parked vehicles, which can obstruct visual information. Medium-to-long term occlusion, such as parked cars in a city or furniture moved in indoor environment, also obstructs visual information. These two aspects negatively impact our use case for the delivery of virtual content to a wide audience. Finally, the system should deliver accurate camera pose estimates without requiring substantial device movement or cumbersome platform setup, compared to traditional techniques that require either substantial re-scanning of the environment, movement to initialize visual odometry systems, or ultra-accurate initial pose calibration.

Event-based Vision Sensor (EVS) captures movements by luminance changes, where differential luminance data is processed and output as events. EVS has high dynamic range and is not affected by motion blur or under/overexposure, which allows it to track objects that are not pretrained in models and interfere with the inference. Therefore, rather than merely addressing these challenges, we can leverage these dynamic elements to our advantage. EVS shows potential for adaptability to dynamic changes over extended periods and broadcasting these out-of-model information to users for improving performance; however, there is a notable scarcity of such systems in highly dynamic mobile AR use cases. Our framework, EVS-CVPO, uses a single fixed event/RGB sensing platform, which can be easily established within the environment. By using event data, the platform is immune to lighting variations over the course of the day. Furthermore, the event data can be processed to identify changes in the environment and transmitted to users' devices to improve the localization accuracy of monocular RGB queries without requiring additional equipment.

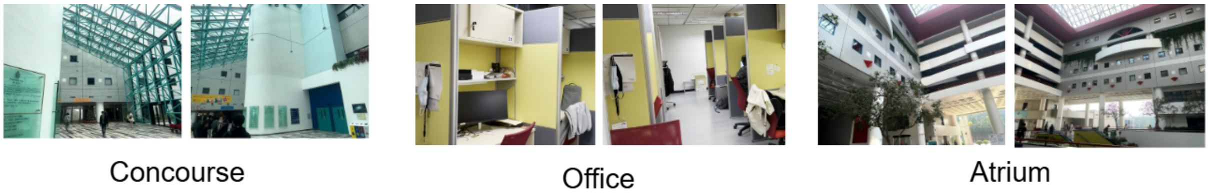


Fig. 3. Example frames from the representative datasets at different scales with pure vision challenges including reflection, featureless walls, repetitive patterns.

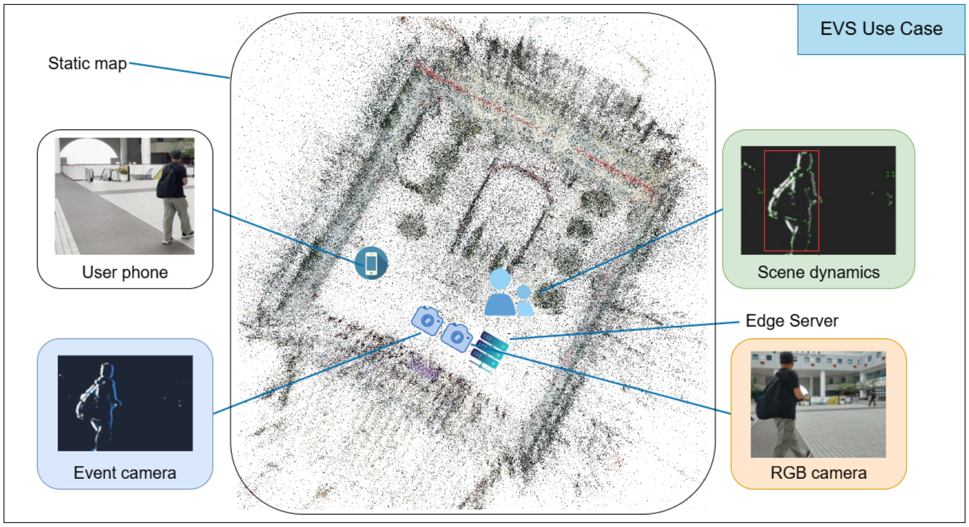


Fig. 4. Event-based Vision Sensor (EVS) use case concept. Fixed infrastructure event sensor and RGB cameras are deployed to build feature map on scene dynamics. Camera frames from the user device are transmitted to a server, which combines the dynamic map and static map with the aid of 3D geometry information to provide accurate localization for mobile AR.

To demonstrate the various challenges posed by the environment and to validate the capabilities of our method, we selected three representative scenes at a local university, illustrated in Fig. 3. All three scenes present significant challenges for visual positioning systems. Concourse is a large indoor area characterized by glass reflections and featureless white walls. Traditional visual positioning systems exhibit low accuracy in such environments. Furthermore, as the main pathway between the university entrance and classrooms, significant dynamic occlusion happens due to human traffic. Office is a smaller-scale indoor open-space office. As such, it features repetitive patterns, and frequent changes, such as the arrangement of objects on desks and shelves, as well as dynamic occlusion by the office users. Finally, Atrium is a large semi-open area, which features highly repetitive patterns and textures. Events regularly take place in this space, resulting in significant changes in its geometry. All three spaces represent typical areas where markerless AR applications could be deployed. Navigation applications could be deployed in the Concourse scene to help visitors and new students find their destination. As the main public space on the campus, the Atrium could showcase tech demos and artworks, as well as event-related content. Finally, the office scene could be used for work-related AR applications and spatial computing. However, they present significant challenges for establishing accurate and reliable positioning.

To address this issue, we propose a cross-view pose optimization pipeline supported by event and RGB camera hardware. The modules in our pipeline combine the advantages of RGB and event cameras, leveraging features in frame-based cameras while benefiting from the high dynamic range, the absence of motion blur, and the focus on motion in event-based cameras. We consider the following implementation scenario:

- For each space, a series of images are captured to build an initial 3D model using structure from motion (SfM) techniques, similar to popular markerless AR and visual positioning frameworks.
- A stationary event/RGB sensing platform composed of hardware-synchronized event and RGB cameras is placed in each space to continually monitors changes in the environment.
- Users traverse the environment, looking at the AR content using their mobile phones.

Camera frames from the user device are transmitted to a server which performs visual positioning with respect to the SfM model and the data coming from the event/RGB sensing platform to achieve greater accuracy and resilience to dynamic changes in the scene as shown in Fig. 4.

4 Event/RGB Sensing Platform

The localization pipeline relies on an event/RGB sensing platform that is placed in a stationary location in the space to monitor dynamic changes. We utilized a multimodal data acquisition device for data collection, which consists of several components, including an event camera, a pinhole camera, a Livox Avia LiDAR, a NUC mini-computer (Ryzen R9-8945HS, 64G RAM), and a display screen. Unlike normal frame-based cameras, which capture a whole image at a predetermined time interval, event cameras capture events based solely on changes in brightness at individual pixels. The event/RGB sensing platform utilizes an event camera to complement a standard camera, focusing on dynamic objects within the scene. Users capture the scene with their regular monocular device cameras (smartphones in the case of markerless AR). It is important to note that the 3D point cloud generated by the LiDAR is not used or integrated into the operational pipeline and is solely utilized for ground truth reference within the dataset.

4.1 Capture Platform Architecture

The capture platform is composed of two main image sensors and a LiDAR which are hardware-synchronized.

4.1.1 Hardware. The event camera input is captured from our SE1-S4-USB[3]. The camera uses SONY IMX646 [6] EVS image sensor with M12 mount lens and outputs USB3.0 interface, and is fully compatible with MetaVision[4] software. The IMX646 has 1280 x 720 pixels with a latency of 800 us @ 1000lux. The RGB camera used is JHEM304UC[5] at the resolution of 2048 x 1536, with standard USB3 Vision protocol and integrated with vision software platform Halcon[6]. The event and RGB cameras are placed stationarily in the scene, capturing at 30 Hz. In the event camera, events are timestamped with a 1MHz clock (1µs period). This clock is internally generated when used in standalone mode. We use hardware trigger to connect and synchronize the two cameras to produce synchronous timestamping. The LiDAR is a high-performance device featuring a field of view (FoV) of 70.4° × 77.2°, and it is equipped with an integrated Inertial Measurement Unit (IMU). To calibrate the extrinsic parameters between the LiDAR and the cameras, we employed a calibration toolbox [16].

4.1.2 Camera Synchronization. To facilitate the integration of frame and event data within the cross-modal object tracker, the cameras must be hardware-synchronized to ensure accurate data alignment. As the event camera IMX646 does not provide a Trigger Out facility, the RGB camera is set as the master camera, which generates a

[3] https://www.sony-semicon.com/en/products/is/industry/evs.html
[4] https://docs.prophesee.ai/stable/index.html
[5] https://jinghangtech.com/product/u3v/306.html
[6] https://www.mvtec.com/products/halcon

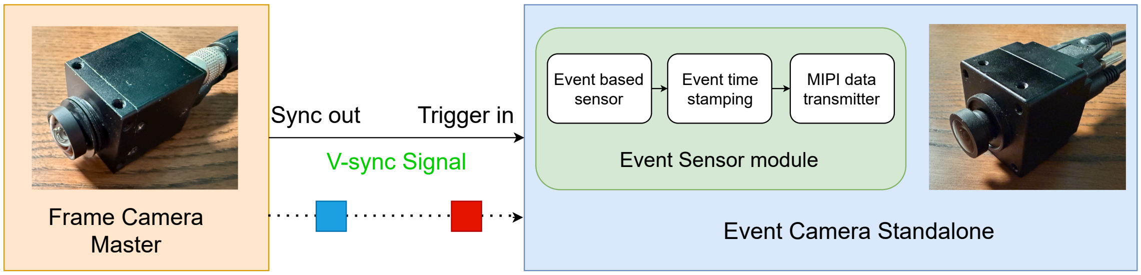


Fig. 5. RGB camera is set as master for synchronization and generate a VSync signal which is fed to trigger in interface of the Event camera. The VSync signal generates External Trigger Event in the event stream where ON event (blue) at each rising edge of VSync and OFF event (red) at each falling edge.

VSync signal which is fed to Trigger In interface of the Event camera as shown in Fig. 5. The event camera is used in standalone mode, which is the default mode of the camera. The markers that Trigger In allows to inject into the stream of data is a specific type of events called External Trigger Events. Those events are timestamped along the other events. When standalone event cameras is started, the data stream starts immediately. After that, once the master RGB camera is started, VSync is fed to event camera and External Trigger events are generated in event camera.

The VSync signal provided by the RGB camera is generating External Trigger Event in the event stream. When a complete transition cycle occurs, a specific rising-edge ON event is generated first (blue) followed by a specific falling edge OFF event (red). The two events are generated while the sensor is streaming its data. The 2 data flows are merged together and provided to the sensor output transmission interface. Hence the data streamed from the camera can be a mixed of Contrast Detector (CD) events, which will be described in detail in Section 4.3, and External Triggers events. Those events are then used to synchronize the frame and prepare event streams for data preprocessing.

4.1.3 Camera Calibration. The intrinsic parameters of the RGB camera are calibrated using a checkerboard, while extrinsic parameters are calibrated utilizing an AprilTag [46].

As event cameras do not record stationary images, we follow the Prophesee standard procedure [7] of intrinsic calibration. It detects a calibration pattern from different viewpoints and estimates the intrinsics from those detections, which can be summarized in three steps: 1) **Event-based Pattern Detection**. This step involves detecting a calibration pattern within the event stream. During the detection, it displays the events generated by the camera and the output with the current detected pattern, the total number of successfully detected patterns and their overlay indicating the coverage of the field of view as shown in Fig. 6. The camera is mounted on a tripod and moved together. Once the pattern is captured from one viewing angle, the camera is moved to a new position to capture the pattern from a different viewing angle. 2) **Intrinsics Estimation**. In this phase, the camera intrinsic parameters are estimated utilizing the detections obtained from the previous step, along with a specified camera model. 3) **Intrinsics Validation**. This final step entails visually validating the results of the intrinsic estimation by employing both the detections and the estimated camera geometry from the preceding steps.

4.1.4 Capture Platform Setup. The LiDAR data and RGB frames are integrated into a LiDAR-Inertial-Visual SLAM system [18], which employs both LiDAR-Inertial Odometry (LIO) and Visual-Inertial Odometry (VIO).

[7] https://docs.prophesee.ai/stable/api/cpp/calibration/algorithms.html?highlight=calibration

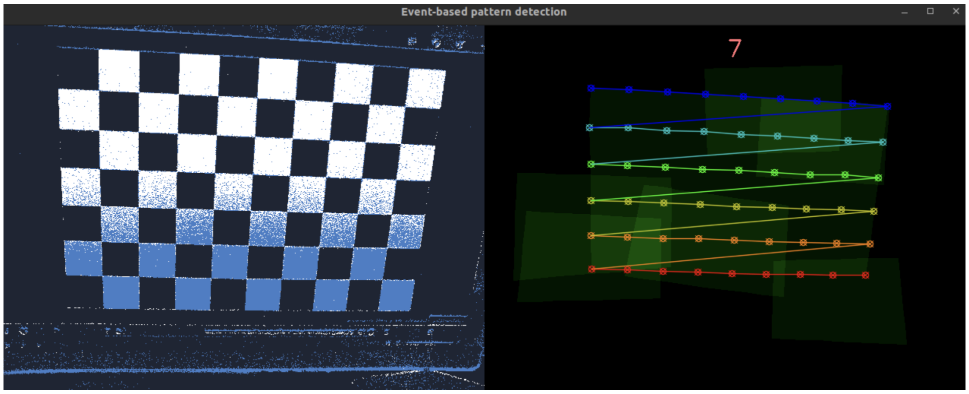


Fig. 6. The event-base pattern detection displays the events generated by the camera (left) and the output (right) with the current detected pattern, the total number of successfully detected patterns and their overlay indicating the coverage of the field of view.

During the data collection process, the LiDAR was configured to operate at a frequency of 10 Hz. We established a synchronized trigger between the LiDAR and the RGB camera, along with hardware synchronization between the event camera and the RGB camera (as detailed in subsection 4.1.2), to facilitate pseudo-registration between different modalities.

In practical applications, the straightforward setup of the stationary platform allows for arbitrary placement to capture the desired angles by aligning coordinates and registering the platform through an online process. The event camera and RGB camera can be adjusted to operate at a suitable distance and over a wide viewing angle to cover most dynamics in each scene. Although we utilize Bundle Adjustment (BA) optimization to refine the registered pose and LiDAR point map of ground truth contributed for the dataset, the LiDAR data is not included or used in the system pipeline, which leads to a light-weighted and convenient setup in practice called event/RGB sensing platform.

4.2 Data Capture Process

The event/RGB sensing platform is meant to be used online. Data from the platform is used together with the initial SfM model of the environment to process user requests on a remote or edge server. To facilitate evaluation, we collect a dataset over the three scenes presented in Section 3.

During the dataset collection, data from the event/RGB sensing platform and user traces are collected concurrently. The event/RGB sensing platform is placed at a stationary location, while users are instructed to navigate through the scene freely, without following specific routes to effectively simulate an AR scenario. Users are equipped with Iphone 16 Pro device. We develop an AR data capture application [19] to capture the devices' sensor data using ARKit 5.1.5. The AR data capture application records images, VIO data and sensor data—including magnetometer, gyroscope, and accelerometer reading, all at a fixed frequency, ensuring alignment with each corresponding image. Users capture data at a frequency of 2 Hz in indoor scenes (Concourse and Office) and 1 Hz in the semi-open scene (Atrium).

The phone images from the user device are split into two sets: the training set and the test set at a ratio of around 4:1 at the first capture session. The training set is used to build the initial SfM model of the environment. The RGB and event images processed from event data collected online are used as input to the cross-modal tracker and dynamic map constructor to maintain the dynamic feature map, using the dynamics in the scene to

Table 1. Dataset summary statistics

Scene	Surface	Sequences		Scene Specifics
		Train	Test	
Concourse	$950m^2$	8	2 + 6 + 6	Large scale, high dynamic occlusion, low feature and reflective surfaces.
Office	$45m^2$	8	2 + 6 +6	Daily object movement and dynamic occlusion by users.
Atrium	$1500m^2$	8	2 + 6 + 6	Large-scale, repetitive textures, high traffic, variations last for long time.

register the pose of query images. The test set images are timestamped and queried against the dynamic feature map corresponding to the timestamp to predict the pose of the user devices at a given time.

Data collection occurred once per week for the first month to capture medium-term changes, followed by once per month for the subsequent three months to assess long-term changes. The straightforward setup allows for arbitrary placement to cover the angles of interest. The RGB images captured from the stationary platform are registered to the original SfM model, and the subsequently constructed dynamic maps are computed in 3d space, with coordinate transformations applied to align them with the same coordinate system as the static map. As such, we systematically evaluate the results across three distinct scenes to ensure that both short-term and long-term changes are effectively addressed in dynamic environments with various characteristics. The dataset also provides comprehensive coverage of diverse lighting conditions in both daytime and nighttime settings. The statistics for each scene in the dataset is illustrated in detail in Tab. 1.

- The Concourse dataset has many repetitive features, reflective glass and featureless walls, which adds difficulty to pure visual positioning methods.
- Office is a semi-synthetic dataset where we manually change the scenes by moving chairs or books on the shelf during collection to simulate extended use over time.
- The Atrium dataset aims to present large changes over time to demonstrate the viability of our method towards long-term changes.

The same conditions apply to all three scenes during the four months of capture. The first capture for each scene each lasted approximately four hours and comprised 10 sequences. 5 sequences were taken at daytime and 5 at night. The subsequent captures over the following three weeks each took about 0.5 to 1 hour, with two sequences collected per week for each scene, one during the daytime and the other at night. The varying durations are due to the differing capture rates between indoor scenes (2Hz) and open scenes (1Hz) for a fixed number of images (200) to accurately represent the environments. Finally, the remaining monthly captures collected over the next three months similarly feature two sequences for each time for each scene, 1 daytime and the other nighttime. Each sequence contains over 200 groups of images, with each group composed of the event image, RGB image and 3d point cloud from the Lidar captured at a given timestamp.

The collected data is classified as follows: For each scene, 8 sequences from the first capture are used as the training set or database for different networks, as detailed in Sec. 6.2, while 2 sequences are designated as test set, of which 1 at daytime and the other at nighttime. All sequences collected afterward were used as a test set to highlight the differences accumulated over time. The weekly sequences focus on short-term changes while the monthly sequences address longer-term changes.

Table 1 presents the summary statistics[8].

[8] To promote more research on the topic of cross-modal visual positioning, the dataset can be found online at https://github.com/XRIM-Lab/EVS-CVPO.

4.3 Data Preprocessing

The event sensor uses a Contrast Detector (CD) which emits events each time there is a variation in the light level with asynchronous signal processing capabilities. It generates an ON or OFF event, to signify an increase or decrease in intensity that exceeds specified thresholds [29]. To bridge the vast gap between the two modalities, we first convert the event streams into event images. Subsequently, we apply a denoising process before inputting the data into the cross-modal tracker module, where the connection between the two modalities is established.

The CD data is read out using Metavision SDK in the form of (x, y, p, t), where (x, y) is the position of the pixel in the sensor array, t is the timestamp of the light change, expressed in microseconds, p is the polarity. A polarity value of 1 indicates a CD ON event, which corresponds to the detection of a positive contrast, signifying a transition in light from a darker to a lighter state; and inversely.

Taking these tuples accumulated over the period, we first transform it to an event image $I \in R^{h*w}$, where h and w are the dimensions of the event image, namely 1280 and 720. The second step is to mute the noise through Gaussian blur with the 5×5 kernel as the primary focus is on the dynamic objects within the scene. The preprocessing plays an important role since it affects the quality of the event images, which are used to track the object and build the dynamic map. The user images captured by phone are registered using the SfM tool COLMAP [39] to generate the ground truth pose. They are matched with the pairs of event and frame images by timestamp to be processed with our pipeline modules.

5 Camera Pose Estimation with Event-RGB Sensor Fusion

The proposed RGB/event capture platform is at the core of a novel EVS-based visual localization framework that optimizes the pose predictions by leveraging the combined advantages of the RGB and event frames.

5.1 System Overview

The overall pipeline of the proposed visual localization method with the event camera system is shown in Fig. 7. The proposed method consists of three modules: cross-modal tracker, dynamic map constructor and cross-view pose optimization. Among them, the cross-modal tracker module addresses the problem of object tracking between RGB and event data. It detects and matches objects in the RGB and event frame and provides their accurate tracking boxes, thus bridging the vast distribution gap between the two modalities. Then, the dynamic map constructor uses depth estimation combining events and RGB frames to maintain a dynamic map at each sensing timestamp. Finally, the cross-view pose optimization method is used to fuse the current dynamic map and the prebuilt static map, achieving robust visual localization even in environments with high scene dynamics or presenting significant long-term changes.

5.2 Cross-modal Tracker

In the cross-modal tracker module, the combination of RGB information and event data leverages the respective advantages and characteristics of both modalities, addressing the limitations of single-modality approaches in complex scenarios. RGB data provides static semantic features of the scene, such as color, texture, and contextual appearance, while event data, with its extremely high temporal resolution, captures rapidly changing dynamic information in real-time. By integrating these two modalities, a more comprehensive and accurate model of target objects can be constructed.

However, traditional methods for combining RGB and event data have shown limited effectiveness. For instance, using SuperPoint [5] to extract features from event images and then matching them with features from RGB images performs poorly due to the excessive sparsity of features in event data and the significant domain gap between modalities. To address this issue, we adopt a cross-modal tracker, which fully leverages the strengths of

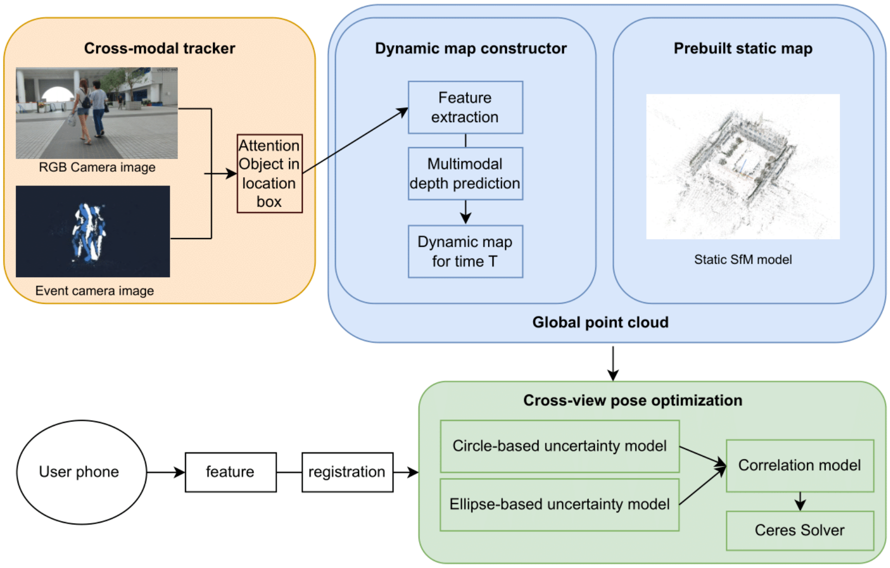


Fig. 7. Overview of the proposed method. The cross-modal tracker detects and matches objects between the stationary RGB and event frames. These objects are used to build a dynamic map, which is then combined with the prebuilt static SfM map to form the global map. This global map is then used by the cross-view pose optimization module to refine the predicted pose for the user query.

both RGB and event data and achieves more precise object tracking through deep interaction between the two modalities.

The cross-modal tracker takes information from both the RGB and the event modality as input. The input data for both modalities is divided into template images and search images, which are used to initialize the target position and locate the target. The RGB and event data are partitioned into several patches and encoded into feature vectors, referred to as tokens, which represent local image or event features. These tokens serve as the fundamental units for cross-modal tracking and are uniformly processed within the Transformer model.

The Transformer model uses a multi-head attention mechanism to achieve feature fusion between the RGB and event modalities and to facilitate deep interaction between the modalities. In the Transformer, the tokens for the template and search regions are mapped to the Query(Q), Key(K), and Value(V) feature spaces. d_k is the dimension of the feature vector. For each pair of template and search tokens, the attention weights are computed as follows:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}}) * V \tag{1}$$

The fused features are enhanced through the layer-by-layer Transformer modules, which can strengthen the cross-modal collaboration.

Next, we match the template and search regions by calculating the correlation matrix between them. Here, M_{ij} represents the correlation strength between template token i and search token j.

$$M_{ij} = softmax(\frac{Q_i K_j^T}{\sqrt{d_k}}) \tag{2}$$

We use the classification branch of the cross-modal tracker to predict whether each token is a target, and we use the regression branch to predict the position and size offset of the bounding box. The target box offset is given by $\Delta b_i = W_{reg} h_i$, where h_i is the feature of the i^{th} token, and W_{cls} is the weight matrix of the classification layer. Based on the initial bounding box b_T from the template region and the offset Δb_i, we can calculate the target object's bounding box in the search region as follows:

$$b_S = b_T + \Delta b_i \tag{3}$$

Finally, by combining the results of the classification branch, which predicts the presence probability of the target in the search region, and the regression branch, which predicts the position and size offset of the bounding box, we can obtain the precise bounding box of the target object in the search region.

5.3 Dynamic Map Constructor

Monocular depth estimation estimates pixel-wise scene depth from a monocular input. Event cameras offer significant advantages compared to standard cameras due to their high temporal resolution, high dynamic range and lack of motion blur. However, events only measure the varying components of the visual signal, which constrains their capacity to encode contextual information about the scene. In contrast, standard cameras capture absolute intensity frames, offering a more comprehensive representation of the visual environment. Consequently, these two types of sensors are complementary, especially in estimating the depth of dynamic objects.

However, due to the asynchronous nature of events, combining them with synchronous images remains challenging, especially for learning-based methods. Traditional recurrent neural networks (RNNs) are not designed for asynchronous and irregular data from additional sensors. To establish the real-time depth prediction over the attention objects, we employ RAM Net [8], a Recurrent Asynchronous Multimodal (RAM) network that extends the capabilities of standard RNNs to accommodate asynchronous and irregular data inputs from multiple sensors. Inspired by traditional RNNs, RAM networks maintain a hidden state that is updated asynchronously and can be queried at any time to generate a prediction.

The architecture of RAM Net is a fully convolutional encoder-decoder architecture based on U-Net [32]. We use the event and frame image pairs synchronized by hardware trigger from Section 4.1.2 as inputs. RAM Net outputs predictions for the events and image inputs, and the predictions are dependent on all previous inputs due to the recursive calculation of the state. The network is implemented in PyTorch and optimized using the ADAM optimizer [15] with a batch size of 8. We use the checkpoint fine-tuned on MVSEC [54] as the pretrained model in the experiment.

Dynamic map. Utilizing the tracking boxes of moving objects from Section 5.2, we extract features with Superpoint [5] from the objects within these boxes on the frame images. The 3d coordinates (X_w, Y_w, Z_w) of these features are computed by transforming from image coordinate to camera coordinate as follows:

$$\begin{cases} X_c = \dfrac{i - u_0}{f_x} * D(i,j) \\ Y_c = \dfrac{j - v_0}{f_y} * D(i,j) \\ Z_c = D(i,j) \end{cases} \tag{4}$$

Then the feature points are transformed from camera coordinates to world coordinates:

$$\begin{cases} X_w = R[0,0] * X_c + R[0,1] * Y_c + R[0,2] * Z_c + T[0] \\ Y_w = R[1,0] * X_c + R[1,1] * Y_c + R[1,2] * Z_c + T[1] \\ X_z = R[2,0] * X_c + R[2,1] * Y_c + R[2,2] * Z_c + T[2] \end{cases} \tag{5}$$

where the depth estimation value on pixel (i, j) is $D(i, j)$, f_x, f_y, u_0, v_0 are the camera intrinsics, R is the rotation matrix and T is the translation vector.

Static map. The static map is the point cloud that contains the static information of the scene. SfM models may struggle with complex camera trajectories or highly dynamic scenes, but are suitable for the static map where we generate the Ground Truth (GT) pose of user images in the experiment.

However, the original point cloud generated by COLMAP is not suitable for localization when combining different feature detectors. Consequently, we construct new 3D models utilizing the keypoints identified by SuperPoint and HF-Net. The process entails the following steps: i) performing 2D-2D matching between reference frames using our features in conjunction with an initial filtering ratio test; ii) further refining the matches within COLMAP through the application of two-view geometry; iii) triangulating 3D points with the provided ground truth reference poses. These steps result in the creation of a 3D model that maintains the same scale and reference frame as the original model.

The global point cloud is composed of the dynamic map and the static map. In this manner, a real-time global map for the dynamic scene appears as a point cloud, where the dynamic map represents dynamic objects within the scene at that timestamp, and the static map remains unchanged.

5.4 Cross-view Pose Optimizer

5.4.1 Direct estimation with global point cloud. Our objective is to estimate the camera pose of the user phone within simulated AR scenarios. We first leverage the global point cloud to directly estimate the user pose. Triggered by timestamp, the global point cloud for that frame is used to align the variables for visual positioning. It allows us to define losses directly on the camera parameters. Similarly, we extract features using Superpoint [5] for the images from the user's phone. The idea is to minimize the loss of feature discrepancies when reprojecting the global point cloud onto the user's phone to optimize for the resulting pose. The global point cloud x^t can be re-parameterized with user camera parameters K^t, $P^t = [T^t|R^t]$ at timestamp t:

$$x^t_{(i,j)} = P^{t^{-1}} h(K^{t^{-1}}[iD^t_{i,j}, jD^t_{i,j}, D^t_{i,j}]) \tag{6}$$

where D^t for depth map , (i, j) for pixel coordinate and $h()$ for homogeneous mapping. The optimization of the pose can be formed as

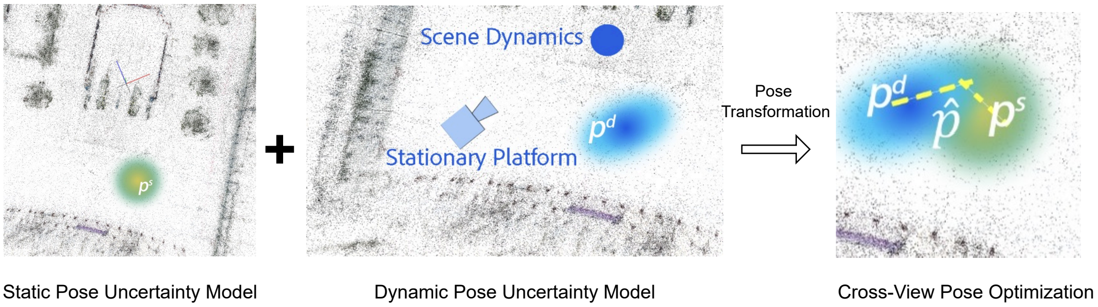
$$argmin\ L_{align}(x^t, P^t_w) \tag{7}$$

where P^t_w is the global point cloud from section 5.3 in world coordinate. We denote the pose output by this method as DirectEVS. However, the localization accuracy is low and can be directly affected by depth estimation errors of dynamic objects, which is insufficient to meet the actual localization requirements of AR applications in real scenes. To address the above challenges, we propose a cross-view pose optimization method to achieve locally higher accuracy and globally better adaptation to changes in the environment.

5.4.2 Cross-view pose optimization. To make use of dynamic objects and solve the long-term changes problem in visual localization, we present a new localization framework and introduce a novel cross-view pose optimization method to build pose uncertainty models, which can handle the differences between the two complementary types of data.

The static map with SfM information is locally stable but unknown to changes, inspired by this, we propose a circle-based pose uncertainty model to characterize the uncertainty of SfM pose. Based on Section 5.3, the estimated position by the static map in the ground coordinate is $p^s = (p^s_x, p^s_y)$, and we consider the position uncertainty and construct the following equation:

$$\begin{cases} \hat{p}p^s \times \hat{p}p^d = 0 \\ (\hat{p}_x - p^s_x)^2 + (\hat{p}_y - p^s_y)^2 = r^2 \end{cases} \tag{8}$$



Static Pose Uncertainty Model

Dynamic Pose Uncertainty Model

Cross-View Pose Optimization

Fig. 8. Cross-view pose optimization. We combine a static pose uncertainty model (from the original user query) and with a dynamic pose uncertainty model (from the stationary event/RGB sensing platform) to refine the predicted pose.

where $\hat{p} = (\hat{p}_x, \hat{p}_y)$ is the optimized pose, $p^d = (p_x^d, p_y^d)$ is the estimated position by the dynamic map, and r is the radius of a circle representing the position uncertainty.

The cross-modal tracks the dynamic object accurately in the 2d pixel space of x and y, but the exact z value is relatively vague in the depth direction, with possible out-of-line errors. Therefore, we propose an ellipse-based uncertainty model for the pose obtained from the dynamic point cloud. After the dynamic feature map has been initialized, the uncertainty model can be formulated as:

$$\begin{cases} (\hat{p}^T - {p^d}^T) R_d \sigma R_d^T (\hat{p} - p^d) < 1 \\ R_d = \begin{bmatrix} cos(\theta) & sin(\theta) \\ -sin(\theta) & cos(\theta) \end{bmatrix} \\ \sigma = \begin{bmatrix} 1/{p_x^d}^2 & 0 \\ 0 & {p_y^d}^2 \end{bmatrix} \end{cases} \tag{9}$$

where σ represents the pixel uncertainty of the depth estimation. θ is the rotation of the camera along the z-axis direction. R_d is the transformation matrix about the current camera pose. By combining and jointly solving the set of circle (8) and ellipse (9) equations, the larger error is discarded to reduce its impact on localization.

To fully utilize the advantages of the two models and achieve complementary advantages, we introduce a pose optimization method to realize localization as shown in Fig. 8. The principle of pose optimization is a maximum likelihood estimation problem. we use the Ceres Solver [33] to iteratively solve this nonlinear problem to achieve stably higher accuracy and changes-aware EVS correction localization, as follows:

$$min(\|\hat{P}^t - P_s^t\|_2 + \alpha\|\hat{P}^t - P_d^t\|_2) \tag{10}$$

where $P^t = [T^t|R^t]$ is the pose at timestamp t, α is the hyperparameter between the two models. The resulting pose output by the cross-view optimizer is denoted as EVS-CVPO, where CVPO stands for cross-view pose optimization.

6 Evaluation

In this section, we conduct accuracy comparison experiments, data analysis and an ablation study to validate the performance of the proposed method. Since our method relies on our data collection platform of the fixed RGB camera and event camera, we set up and capture data on a local campus to verify the localization performance.

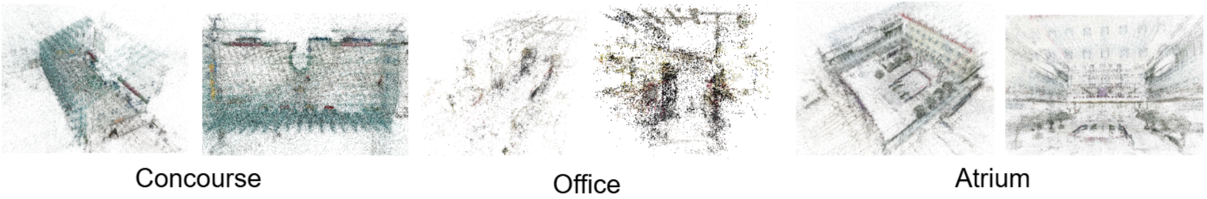


Fig. 9. Sparse SfM model for each scene. It is built with the set of reference images taken at the first data capture session. The images in later capture sessions are registered into the model for pose GT.

To demonstrate the scalability and effectiveness of our proposed method, we collect three scenes of various scales by three different means according to their characteristics.

6.1 Datasets

Our dataset consists of three scenes, namely Concourse, Office, and Atrium. Each scene is composed of a sparse SfM model built with a set of reference images taken at the first data capture session as shown in Fig. 9. Concourse ($950m^2$) is a large indoor scene with many moving people. Office ($45m^2$) is a corner of an office representing the daily use of the office. Atrium ($1500m^2$) is a large semi-open scene with many moving people.

The data collection processes in these scenes are discussed in detail for both platform online data and user query data in Sec. 4.2, as well as the partitioning of training and test sets on user images. For the queried images from user phone, we estimate their absolute poses with the feature map built by the data collection platform in an online manner. To establish GT pose, we feed all the user images into the SfM framework using COLMAP. All subsequent data is registered into the SfM coordinate system that was initially built for evaluation.

6.2 Implementation Details

To understand the impact of multimodality and the cross-view module, we implement and test on three baselines termed (i) purely learning-based (PN and MS-T) or structure-based (HLoc), (ii) event and frame-based (DirectEVS) (ours), (iii) event and frame with our cross-view pose optimizer modules (EVS-CVPO) (ours) and pseudo event from RGB inputs (see Sec. 6.2 only and frame with our cross-view pose optimizer modules (RGB-CVPO) (ours). The user handheld images from phone at the first capture are split into training set and test set in at 4:1 ratio. The sequences from later capture sessions are also used as test set, with a similar amount to the first session's test set, to demonstrate the effects of changes in the environment. The training set of PN and MS-T is also used as the database for HLoc. The test set is the same for all the baseline methods and our proposed methods.

PN. PoseNet [14] serves as the baseline method in this study. We adopt the approach proposed in [2] and employ ResNet34 [10] as the underlying network architecture.

MS-T. MS-Transformer [40] expands upon the single scene paradigm of APR to encompass simultaneous learning of multiple scenes. It encodes aggregate activation maps with self-attention to focus on general features that are informative for localization while embedding multiple scenes in parallel. Consequently, we train one instance of MS-T for all indoor scenes (Concourse, Office) and another for the semi-open scene (Atrium).

HLoc. Hierarchical-Localization [34] leverages image retrieval and feature matching for accurate and stable visual localization. It builds the SfM model and run with SuperPoint [5]+SuperGlue [35] to localize with the database. Therefore, we denote it as HLoc (SP+SG), where SP and SG stands for SuperPoint and SuperGlue respectively. All experiments are conducted using an NVIDIA GeForce RTX 3090 GPUs.

Table 2. Mean absolute translation/orientation errors in m/°. for three scenes for weekly captures (4 weeks). The best result is highlighted in bold.

	Week	PN	MS-T	HLoc(SP+SG)	DirectEVS(ours)	RGB-CVPO(ours)	EVS-CVPO(ours)
Concourse	1	1.59/7.50	1.71/5.49	0.27/0.64	0.38/1.07	0.31/1.02	**0.21/0.58**
	2	1.71/6.86	1.79/5.81	0.29/0.62	0.42/1.16	0.37/1.25	**0.28/0.59**
	3	1.83/7.89	1.88/5.78	0.38/0.82	0.46/1.29	0.44/1.33	**0.30/0.71**
	4	1.85/8.22	1.97/6.32	0.51/0.95	0.53/1.34	0.45/1.18	**0.35/0.87**
Office	1	0.42/7.28	0.18/5.66	**0.07**/0.80	0.14/2.49	0.12/1.37	0.09/**0.72**
	2	0.93/9.57	0.35/6.32	0.13/1.43	0.19/2.80	0.16/2.32	**0.12/1.25**
	3	0.86/9.10	0.38/6.55	0.18/1.86	0.22/2.61	0.25/1.96	**0.15/1.40**
	4	0.97/9.46	0.47/7.31	0.19/2.05	0.20/2.59	0.27/2.48	**0.16/1.33**
Atrium	1	1.73/6.24	1.97/5.28	**0.38/1.75**	0.51/2.10	0.48/2.12	0.41/1.84
	2	1.99/6.73	2.12/5.39	0.53/**1.83**	0.64/2.43	0.55/2.05	**0.49**/1.88
	3	2.25/7.63	2.41/5.17	0.67/2.19	0.70/2.11	0.85/2.61	**0.61/1.92**
	4	2.68/8.21	2.45/5.28	0.63/2.24	0.66/2.20	0.78/2.50	**0.60/1.97**
Avg	-	1.57/7.89	1.47/5.86	0.35/1.43	0.42/2.02	0.41/1.85	**0.31/1.26**

6.3 EVS-CVPO Evaluation on Pose Estimation

Given a stream of user images I_i, the cross-view pose optimizer $\mathcal{R}$ outputs estimated global position $\hat{\mathbf{x}}_i$, $\hat{\mathbf{x}}_i \in \mathbb{R}^3$ and quaternion $\hat{\mathbf{q}}_i$, $\hat{\mathbf{q}}_i \in \mathbb{R}^4$ which encodes the rotation so that $\mathcal{R}(I_i) = \hat{p}_i =< \hat{\mathbf{x}}_\mathbf{i}, \hat{\mathbf{q}}_i >$ is the 6DoF camera pose for each image. To calculate the absolute pose error between the result and the GT, the absolute position error (APE) and absolute orientation error (AOE) are computed as:

$$\mathrm{APE} = ||\mathbf{x}_i - \hat{\mathbf{x}}_\mathbf{i}||_2 \tag{11}$$

$$\mathrm{AOE} = 2 \arccos |\mathbf{q}_i^{-1} \hat{\mathbf{q}}_\mathbf{i}| \frac{180}{\pi} \tag{12}$$

where $\mathbf{x}_i$, $\mathbf{q}_i$ are the position and quaternion of GT, $\mathbf{q}^{-1}$ denotes the conjugate of $\mathbf{q}$, and we assume all quaternions are normalized: $\hat{\Delta}_{rot}(i+1, i) = 2 \arccos |\hat{\mathbf{q}}_{i+1}^{-1} \hat{\mathbf{q}}_i| \frac{180}{\pi}$. Tab. 2 and Tab. 3 show the mean absolute translation/rotation errors in m/° for each scene for short-term (weekly) and medium-term (monthly) captures, respectively.

Our proposed method EVS-CVPO achieves robust and stable localization results in different sequences at scenes of different scales, achieving superior positional accuracy and rotational accuracy over APR methods. Compared to HLoc in general, the result of EVS-CVPO reduces the average absolute tranlation and rotation errors by 12.9% and 13.4% in the interval of week and by 38.5% and 16.2% in the interval of month in Tab. 2 and 3. Delving into each scene by week, the optimized result in the indoor scene is better than the semi-open scene, namely 24.1% and 10.1% for Concourse and 3.8% and 5% for Atrium in APE and AOE. This is because our method additionally considers the dynamic elements, meanwhile integrates the static models. The same applies to Office where EVS performance degrades after the first capture session but remain stable afterwards in Tab. 3, as benefited by dynamics, forming the suitable solutions for EVS-CVPO in all three scenes. On the contrary, DirectEVS doesn't outperform HLoc as the depth estimation of dynamic objects enlarges the error of registration when global point cloud is processed as a whole. EVS-CVPO builds the uncertainty models to fuse the two modalities complementarily in the use case for AR.

Tab. 2 also shows that the precision of EVS-CVPO in Atrium is slightly lower than that of HLoc in the initial data capture session. This discrepancy primarily arises from the extensive scale of the Atrium in 3D space, coupled with HLoc emphasis on the visual geometry of the scene. And it results in diminished precision for dynamic

Table 3. Mean absolute translation/orientation errors in m/°. for three scenes with monthly captures (4 months). The result of the first month is the average of first 4 weeks. The result of the following months is represented by the capture session for each month. The best result is highlighted in bold.

	Month	PN	MS-T	HLoc(SP+SG)	DirectEVS(ours)	RGB-CVPO(ours)	EVS-CVPO(ours)
Concourse	1	1.75/7.62	1.84/5.85	0.36/0.76	0.45/1.21	0.39/1.12	**0.29/0.69**
	2	2.42/8.91	2.39/7.08	0.58/0.92	0.71/1.45	0.65/1.09	**0.37/0.83**
	3	2.51/8.30	2.12/6.59	0.60/1.24	0.63/1.38	0.88/1.55	**0.36/1.02**
	4	2.93/9.71	2.60/7.93	0.75/1.37	0.81/1.57	1.12/1.76	**0.39/1.07**
Office	1	0.79/8.85	0.35/6.46	0.14/1.54	0.19/2.62	0.20/2.03	**0.13/1.18**
	2	0.92/7.91	0.57/6.35	0.20/2.14	**0.17**/2.29	0.28/2.31	0.18/**1.43**
	3	1.12/8.63	0.66/7.12	0.24/2.69	0.27/2.58	0.31/2.51	**0.20/1.39**
	4	1.05/8.21	0.71/6.83	0.23/3.02	0.24/2.86	0.30/2.62	**0.18/1.50**
Atrium	1	2.16/7.20	2.24/5.28	0.55/2.00	0.63/2.21	0.67/2.32	**0.53/1.90**
	2	2.75/8.31	2.91/6.85	0.87/2.73	1.15/3.87	0.98/3.70	**0.62/2.36**
	3	2.87/7.93	2.74/6.32	0.96/2.69	1.03/3.32	1.10/3.45	**0.68/2.49**
	4	3.21/8.83	2.88/7.17	1.01/2.93	1.32/3.85	1.22/3.37	**0.71/2.61**
Avg	-	2.04/8.37	1.83/6.65	0.54/1.79	0.63/2.43	0.67/2.32	**0.39/1.54**

estimates and subsequently impacts the fused results. In contrast, as shown in Tab. 3 accounting for long-term changes, HLoc exhibits a 85.4% and 73.9% increase in APE and AOE from the first month to the fourth month. While EVS-CVPO displays a 35.6% and 39.8% increase of APE and AOE errors in the same way. Notably, this contrasts with the fact that the first capture session on Atrium performs better using HLoc, wherein the image database for HLoc was taken and established immediately prior to testing, which is impractical for real use cases. It shows that our system can maintain resilience to long-term changes and integrate dynamics for robust visual localization.

Additionally, by constructing the dynamic map with pseudo dynamic images where the input is RGB images only, we make the baseline together with our pipeline called RGB-CVPO. The cross-view optimizer integrates the two modalities of pseudo dynamics and RGB images, while remains the same in other modules. Next, we will explain the implementation details and discuss the results compared to EVS-CVPO to show the limitations and advantages.

6.4 Pseudo Dynamics from RGB Images

To illustrate the efficacy of event cameras in detecting short-term changes and mitigating long-term performance degradation, we consider the scenario where the stationary sensing platform only comprises an RGB camera without event camera. We thus extract the pseudo-dynamics from the RGB frames captured by the platform for scene updates and user phone localization. To calculate these pseudo-dynamics, we first perform pixel-level frame subtraction to monitor dynamic objects and alterations between two consecutive RGB frames, given that the RGB camera remains fixed on the platform. Subsequently, we employ feature extraction and monocular depth estimation on a single RGB image, thereby generating a comparable 3D dynamic map of the dynamic objects derived from frame subtraction.

The resulting pseudo-dynamic map undergoes further processing through a cross-view pose optimizer to enhance user pose optimization in conjunction with the static map. The dynamics identified from the subtraction of two consecutive RGB frames are classified as short-term changes. If these short-term changes stabilize subsequently, they may be removed from the dynamic map due to frame subtraction. If they correspond with features from subsequent RGB images, they are then classified as long-term changes. These long-term changes are utilized for scene updates within the static map.

As demonstrated in Tab. 2 and Tab. 3, the errors associated with the pseudo-dynamic map generated from RGB images are 24.4% and 31.9% by week and 41.8% and 33.6% by month greater than those produced by EVS within the same pipeline, in terms of distance and orientation. The predominant source of error arises from inaccuracies in the dynamic map concerning both dynamic detection and monocular depth estimation. The EVS exhibits superior efficiency in recognizing dynamic elements and integrates stereo depth capabilities through the combination of the event camera and RGB camera on the platform. Furthermore, the computational burden of the pure RGB solution, encompassing frame subtraction, feature extraction, and monocular depth estimation, totals 175 ms, which is nearly four times the computational cost of the EVS solution, measured at 46.3 ms (14.8 ms for the cross-modal object tracker and 31.5 ms for the dynamic map reconstructor). The results also show that the dynamic map constructed by EVS-CVPO is more accurate than the one by RGB inputs due to high dynamic range, absense of motion blur, and immunity to various light conditions.

Lighting. Our proposed method can sensitively capture moving objects without being affected by different light conditions in the scene. However, the RGB camera has large errors even if adequate images of both daytime and nighttime have been trained because APR methods training on traditional images cannot generalize well beyond the training set via image retrieval baseline [37]. As shown in Tab. 2, EVS-CVPO has higher accuracy as for those without the participation of EVS, the circle-based uncertainty model p^s is more susceptible to the influence of ambient light. On the other hand, the effect of various light conditions also make the feature extraction and matching less accurate for RGB inputs only, resulting in lower-quality dynamic maps, especially in nighttime.

Occlusion. The results of methods under different light conditions confirm our cross-view pose optimization module is effective and fits the design outcomes in Section 6.3. Similarly, the accuracy of the dynamic map constructed is much higher for multimodal depth prediction from EVS and RGB cameras than the one of monucular depth estimation from only RGB images. It shows that EVS-CVPO are robust to different lighting, weather, moving objects, and long-term changes where RGB inputs fail to capture dynamics effectively and suffer in terms of depth accuracy.

6.5 Ablation Study

To better validate the effectiveness of our proposed pipeline, we carry out the ablation study with different hyperparameters on the same test. In this experiment, we compare and analyze the localization results when changing the value of hyperparameters in the preprocessing and the cross-view pose optimizer.

Preprocess. After transmitting per-pixel brightness changes as a continuous stream of asynchronous events and generating event images during preprocessing, we apply Gaussian blur to eliminate noise points. This denoising process is crucial prior to inputting the image pairs into the cross-modal object tracker, as it affects the result of attention dynamics. We change the size of Gaussian filter kernels which reduce image noise and enhance image structures at different scales.

As shown in Tab. 4, there are minor differences in performance among the kernels of sizes 3x3, 5x5, and 7x7, with the 5x5 kernel generally yielding the best results. In contrast, the 9x9 kernel demonstrates a comparatively larger difference, as it reduces excessive detail in the event inputs to our modules. Furthermore, the original images, which contain significant noise without the application of a Gaussian filter kernel, exhibit relatively high errors due to a decrease in the accuracy of the multimodal network for depth estimation.

Cross-view pose optimizer. The hyperparameter α in Equ. 10 represents the uncertainty factor between the RGB and event modalities in the cross-view pose optimizer and is set to 1 by default. We adjust its value to evaluate its influence on the EVS-CVPO pose estimation results. The findings indicate that our method achieves relatively stable visual localization across various hyperparameter values.

Table 4. Mean absolute translation/orientation errors in m/° for EVS-CVPO with different size of Gaussian filter kernels in preprocessing. "None" means the original generated event images are used without the kernel. The best result is highlighted in bold.

Scene	Week	None	3*3	5*5	7*7	9*9
Concourse	1	0.31/0.72	0.24/0.59	**0.21**/0.58	0.23/**0.56**	0.27/0.68
	2	0.33/0.75	0.25/0.63	0.24/**0.59**	**0.24**/0.61	0.28/0.70
	3	0.47/0.88	0.34/0.76	**0.30/0.71**	0.31/0.77	0.36/0.84
	4	0.53/1.06	0.42/0.90	**0.35**/0.87	0.38/**0.85**	0.40/0.89
Office	1	0.13/0.82	**0.08**/0.74	0.09/**0.72**	0.11/0.74	0.13/0.79
	2	0.19/1.45	0.15/1.28	**0.12**/1.25	0.13/**1.21**	0.14/1.27
	3	0.23/1.57	0.16/1.43	**0.15/1.40**	0.18/1.51	0.21/1.54
	4	0.27/1.51	0.18/1.35	**0.16/1.33**	0.19/1.38	0.24/1.44
Atrium	1	0.54/2.19	**0.40**/1.91	0.41/**1.84**	0.44/1.86	0.52/1.99
	2	0.59/2.15	0.53/1.94	**0.49/1.88**	0.55/1.91	0.61/2.31
	3	0.63/2.19	0.65/2.28	0.61/**1.92**	**0.58**/1.96	0.69/2.35
	4	0.83/2.45	0.77/2.34	**0.60/1.97**	0.65/2.13	0.81/2.42

Table 5. Mean absolute translation/orientation errors in m/° for HLoc and EVS-CVPO with different values of α, which ranges from 0.5 to 2. The best result is highlighted in bold.

Scene	Week	HLoc(SP+SG)	0.5	1	1.5	2
Concourse	1	0.27/0.64	0.23/**0.56**	**0.21**/0.58	0.29/0.71	0.32/0.86
	2	0.29/0.62	**0.28**/0.61	**0.28/0.59**	0.31/0.65	0.35/0.78
	3	0.38/0.82	0.35/0.76	**0.30/0.71**	0.41/0.87	0.37/0.79
	4	0.51/0.95	0.45/0.92	**0.35**/0.87	0.44/0.90	0.40/**0.85**
Office	1	**0.07**/0.8	0.09/0.74	0.09/**0.72**	0.11/1.24	0.12/1.49
	2	0.13/1.43	**0.12**/1.29	**0.12/1.25**	0.15/1.38	0.17/1.60
	3	0.18/1.86	0.19/1.71	0.15/**1.40**	0.17/1.65	**0.14**/1.47
	4	0.19/2.05	0.18/1.87	0.16/1.33	0.17/1.28	**0.15/1.25**
Atrium	1	**0.38/1.75**	0.45/1.79	0.41/1.84	0.43/1.88	0.51/1.83
	2	0.53/1.83	0.53/1.96	**0.49**/1.88	0.58/**1.77**	0.63/1.85
	3	0.67/2.19	0.68/2.53	**0.61/1.92**	0.73/1.97	0.92/2.03
	4	0.63/2.24	0.64/2.83	**0.60/1.97**	0.69/2.42	0.73/2.34

However, the localization accuracy of purely visual methods, such as HLoc, is impacted by temporal factors and long-term changes. The static map, constructed based on the SfM model, accumulates errors over time, resulting in a decline in localization accuracy. As illustrated in Table 5, the results improve when greater weight is assigned to the dynamic map model as time progresses. This highlights the effectiveness of our dynamic attention pipeline, supported by event infrastructure, and offers insights for the implementation of real-world use cases.

Respective effectiveness on short-term and long-term changes. By capturing instantaneous dynamics and recording long-term changes, the Event-Based Vision System with Cross-View Pose Optimization (EVS-CVPO) enhances pose estimation by incorporating additional dynamic information and updated scenes to accommodate

Table 6. Mean absolute translation/orientation errors in m/° for HLoc, EVS-CVPO, and EVS-CVPO with solely short-term and long-term changes applied respectively. The best result is highlighted in bold.

Scene	Week	HLoc(SP+SG)	EVS-CVPO	EVS-CVPO (short-term)	EVS-CVPO (long-term)
Concourse	1	0.27/0.64	**0.21**/0.58	0.25/**0.56**	0.30/0.72
	2	0.29/0.62	**0.28/0.59**	0.37/0.71	0.31/0.65
	3	0.38/0.82	**0.30/0.71**	0.36/0.90	0.34/0.78
	4	0.51/0.95	**0.35**/0.87	0.47/0.93	0.39/**0.82**
Office	1	**0.07**/0.8	0.09/**0.72**	**0.09**/0.75	0.10/0.86
	2	0.13/1.43	**0.12/1.25**	0.15/1.48	0.14/1.54
	3	0.18/1.86	0.15/**1.40**	**0.14**/1.52	0.16/1.75
	4	0.19/2.05	**0.16/1.33**	0.22/2.16	0.17/1.63
Atrium	1	**0.38/1.75**	0.41/1.84	0.43/1.91	0.44/1.96
	2	0.53/1.83	**0.49**/1.88	0.50/**1.81**	0.52/2.02
	3	0.67/2.19	**0.61/1.92**	0.69/2.13	0.65/1.98
	4	0.63/2.24	**0.60/1.97**	0.71/2.24	0.62/2.02
Avg	-	0.35/1.43	**0.31/1.26**	0.36/1.42	0.34/1.39

a changing environment. To thoroughly evaluate the effectiveness of these modifications, we analyze results in which short-term and long-term changes are distinctly processed within the pipeline.

Tab. 6 presents the errors associated with the depth map constructor when solely short-term changes and long-term changes are applied, respectively. Short-term changes show priority sometimes at the outset, while scene updates demonstrate maintained accuracy relative to benchmark methods after extended periods. Although short-term dynamics are efficiently captured by EVS, they alone do not achieve the same level of stability of accuracy in 3d space as static map. Meanwhile, long-term changes help mitigate performance degradation over time by integrating scene changes into the static model. However, the accuracy of the generated feature cannot match that of the architecture from SfM construction, where they alone do not demonstrate priority until sufficient scene variation has accumulated. Overall, neither instantaneous dynamics nor scene updates alone perform as effectively as when both are integrated within the pipeline. We conclude that both short-term and long-term changes are beneficial and complement each other within the processing framework.

6.6 System Efficiency

We evaluate the processing time of the processed modules on a PC equipped with an NVIDIA GeForce GTX 3090 GPU. Based on the implementation in Sec. 5, we repeat each measurement 1000 times. The cross-modal object tracker takes 14.8*ms*. The dynamic map constructor takes 31.5*ms*. The cross-view optimizer takes about 142.6*ms*. This means our pipeline can get the estimation that is applicable in real-time use cases. Through its high modularity, our pipeline provides a new paradigm for visual localization with event infrastructure support.

7 Conclusion

This paper presents a novel localization framework that integrates stationary event and RGB cameras to enhance device positioning in dynamic environments. Our method leverages an external sensing platform that monitors the dynamics of the space, which are then combined with the initial static SfM model to improve the accuracy of visual positioning. To address the challenges of matching the dynamic map with the static SfM model, we introduce a cross-view pose optimizer that estimates the pose uncertainty to refine the localization. To evaluate our method, we capture a dataset combining stationary RGB/event camera data and user mobile traces. We provide this dataset to the community for future research in augmented reality and dynamic scene analysis. Our proposed method achieves high accuracy in a diversity of scenarios, and can noticeably improve performance as the space changes over extended periods of time.

References

[1] Eric Brachmann and Carsten Rother. 2021. Visual camera re-localization from RGB and RGB-D images using DSAC. *IEEE transactions on pattern analysis and machine intelligence* 44, 9 (2021), 5847–5865.

[2] Samarth Brahmbhatt, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz. 2018. Geometry-aware learning of maps for camera localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2616–2625.

[3] Guanyu Cai and Jiliang Wang. 2024. ATP: Acoustic Tracking and Positioning under Multipath and Doppler Effect. In *IEEE INFOCOM 2024 - IEEE Conference on Computer Communications*. 1841–1850. doi:10.1109/INFOCOM52122.2024.10621165

[4] Shuai Chen, Xinghui Li, Zirui Wang, and Victor A Prisacariu. 2022. Dfnet: Enhance absolute pose regression with direct feature matching. In *European Conference on Computer Vision*. Springer, 1–17.

[5] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. 2018. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 224–236.

[6] Thomas Finateu, Atsumi Niwa, Daniel Matolin, Koya Tsuchimoto, Andrea Mascheroni, Etienne Reynaud, Pooria Mostafalu, Frederick Brady, Ludovic Chotard, Florian LeGoff, Hirotsugu Takahashi, Hayato Wakabayashi, Yusuke Oike, and Christoph Posch. 2020. 5.10 A 1280×720 Back-Illuminated Stacked Temporal Contrast Event-Based Vision Sensor with 4.86µm Pixels, 1.066GEPS Readout, Programmable Event-Rate Controller and Compressive Data-Formatting Pipeline. In *2020 IEEE International Solid-State Circuits Conference - (ISSCC)*. 112–114. doi:10.1109/ISSCC19947.2020.9063149

[7] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. 2020. Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence* 44, 1 (2020), 154–180.

[8] Daniel Gehrig, Michelle Rüegg, Mathias Gehrig, Javier Hidalgo-Carrió, and Davide Scaramuzza. 2021. Combining events and frames using recurrent asynchronous multimodal networks for monocular depth prediction. *IEEE Robotics and Automation Letters* 6, 2 (2021), 2822–2829.

[9] Shuang Guo and Guillermo Gallego. 2024. CMax-SLAM: Event-based Rotational-Motion Bundle Adjustment and SLAM System using Contrast Maximization. *IEEE Transactions on Robotics* (2024).

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[11] Javier Hidalgo-Carrió, Daniel Gehrig, and Davide Scaramuzza. 2020. Learning monocular dense depth from events. In *2020 International Conference on 3D Vision (3DV)*. IEEE, 534–542.

[12] Yuhuang Hu, Jonathan Binas, Daniel Neil, Shih-Chii Liu, and Tobi Delbruck. 2020. Ddd20 end-to-end event camera driving dataset: Fusing frames and events with deep learning for improved steering prediction. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 1–6.

[13] Alex Kendall and Roberto Cipolla. 2017. Geometric loss functions for camera pose regression with deep learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5974–5983.

[14] Alex Kendall, Matthew Grimes, and Roberto Cipolla. 2015. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *IEEE international conference on computer vision*.

[15] Diederik P Kingma. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[16] Kenji Koide, Shuji Oishi, Masashi Yokozuka, and Atsuhiko Banno. 2023. General, Single-shot, Target-less, and Automatic LiDAR-Camera Extrinsic Calibration Toolbox. arXiv:2302.05094 [cs.RO] https://arxiv.org/abs/2302.05094

[17] Cedric Le Gentil, Ignacio Alzugaray, and Teresa Vidal-Calleja. 2023. Continuous-Time Gaussian Process Motion-Compensation for Event-Vision Pattern Tracking with Distance Fields. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 804–812.

[18] Jiarong Lin and Fu Zhang. 2021. R3LIVE: A Robust, Real-time, RGB-colored, LiDAR-Inertial-Visual tightly-coupled state Estimation and mapping package. arXiv:2109.07982 [cs.RO] https://arxiv.org/abs/2109.07982

[19] Changkun Liu, Yukun Zhao, and Tristan Braud. 2024. MARViN: Mobile AR Dataset with Visual-Inertial Data. In *2024 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE, 532–538.

[20] Nico Messikommer, Carter Fang, Mathias Gehrig, and Davide Scaramuzza. 2023. Data-driven feature tracking for event cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5642–5651.

[21] Nico Messikommer, Stamatios Georgoulis, Daniel Gehrig, Stepan Tulyakov, Julius Erbach, Alfredo Bochicchio, Yuanyou Li, and Davide Scaramuzza. 2022. Multi-bracket high dynamic range imaging with event cameras. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 547–557.

[22] Arthur Moreau, Nathan Piasco, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. 2022. Lens: Localization enhanced by nerf synthesis. In *Conference on Robot Learning*. PMLR, 1347–1356.

[23] Elias Mueggler, Henri Rebecq, Guillermo Gallego, Tobi Delbruck, and Davide Scaramuzza. 2017. The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM. *The International Journal of Robotics Research* 36, 2 (2017), 142–149.

[24] Tayyab Naseer and Wolfram Burgard. 2017. Deep regression for monocular camera-based 6-dof global localization in outdoor environments. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 1525–1530.

[25] Tony Ng, Adrian Lopez-Rodriguez, Vassileios Balntas, and Krystian Mikolajczyk. 2021. Reassessing the limitations of CNN methods for camera pose regression. *arXiv preprint arXiv:2108.07260* (2021).

[26] Anh Nguyen, Thanh-Toan Do, Darwin G Caldwell, and Nikos G Tsagarakis. 2019. Real-time 6dof pose relocalization for event cameras with stacked spatial lstm networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 0–0.

[27] Chun Ho Park, Ahmad Alhilal, Tristan Braud, and Pan Hui. 2024. AnchorLoc: Large-Scale, Real-Time Visual Localisation Through Anchor Extraction and Detection. In *2024 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 125–134.

[28] Stefano Pini, Guido Borghi, Roberto Vezzani, and Rita Cucchiara. 2019. Video synthesis from intensity and event frames. In *Image Analysis and Processing–ICIAP 2019: 20th International Conference, Trento, Italy, September 9–13, 2019, Proceedings, Part I 20*. Springer, 313–323.

[29] Christoph Posch and Daniel Matolin. 2011. Sensitivity and uniformity of a 0.18 μm CMOS temporal contrast pixel array. In *2011 IEEE International Symposium of Circuits and Systems (ISCAS)*. IEEE, 1572–1575.

[30] Henri Rebecq, Timo Horstschaefer, and Davide Scaramuzza. 2017. Real-time visual-inertial odometry for event cameras using keyframe-based nonlinear optimization. (2017).

[31] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. 2019. High speed and high dynamic range video with an event camera. *IEEE transactions on pattern analysis and machine intelligence* 43, 6 (2019), 1964–1980.

[32] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer, 234–241.

[33] K. Mierle S. Agarwal and T. C. S. Team. [n. d.]. GitHub - ceres-solver/ceres-solver: A large scale non-linear optimization library — github.com. https://github.com/ceres-solver/ceres-solver. [Accessed Nov, 2024].

[34] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. 2019. From Coarse to Fine: Robust Hierarchical Localization at Large Scale. In *CVPR*.

[35] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. 2020. SuperGlue: Learning Feature Matching with Graph Neural Networks. In *CVPR*.

[36] Paul-Edouard Sarlin, Mihai Dusmanu, Johannes L Schönberger, Pablo Speciale, Lukas Gruber, Viktor Larsson, Ondrej Miksik, and Marc Pollefeys. 2022. Lamar: Benchmarking localization and mapping for augmented reality. In *European Conference on Computer Vision*. Springer, 686–704.

[37] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixe. 2019. Understanding the limitations of cnn-based absolute camera pose regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 3302–3312.

[38] Cedric Scheerlinck, Henri Rebecq, Daniel Gehrig, Nick Barnes, Robert Mahony, and Davide Scaramuzza. 2020. Fast image reconstruction with an event camera. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 156–163.

[39] Johannes Lutz Schönberger and Jan-Michael Frahm. 2016. Structure-from-Motion Revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

[40] Yoli Shavit, Ron Ferens, and Yosi Keller. 2021. Learning multi-scene absolute pose regression with transformers. In *IEEE/CVF International Conference on Computer Vision*. 2733–2742.

[41] Lei Sun, Christos Sakaridis, Jingyun Liang, Qi Jiang, Kailun Yang, Peng Sun, Yaozu Ye, Kaiwei Wang, and Luc Van Gool. 2022. Event-based fusion for motion deblurring with cross-modal attention. In *European conference on computer vision*. Springer, 412–428.

[42] Ahmed Tabia, Fabien Bonardi, and Samia Bouchafa-Bruneau. 2023. Fully Convolutional Neural Network for Event Camera Pose Estimation.. In *VISIGRAPP (4: VISAPP)*. 594–599.

[43] Stepan Tulyakov, Alfredo Bochicchio, Daniel Gehrig, Stamatios Georgoulis, Yuanyou Li, and Davide Scaramuzza. 2022. Time lens++: Event-based frame interpolation with parametric non-linear flow and multi-scale fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 17755–17764.

[44] Florian Walch, Caner Hazirbas, Laura Leal-Taixe, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers. 2017. Image-based localization using lstms for structured feature correlation. In *Proceedings of the IEEE international conference on computer vision*. 627–637.

[45] Bing Wang, Changhao Chen, Chris Xiaoxuan Lu, Peijun Zhao, Niki Trigoni, and Andrew Markham. 2020. Atloc: Attention guided camera localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 10393–10401.

[46] John Wang and Edwin Olson. 2016. AprilTag 2: Efficient and robust fiducial detection. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 4193–4198. doi:10.1109/IROS.2016.7759617

[47] Junyi Wang and Yue Qi. 2023. Deep 6-DoF camera relocalization in variable and dynamic scenes by multitask learning. *Machine Vision and Applications* 34, 3 (2023), 37.

[48] David Weikersdorfer, David B Adrian, Daniel Cremers, and Jörg Conradt. 2014. Event-based 3D SLAM with a depth-augmented dynamic vision sensor. In *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 359–364.

[49] Shen Yan, Yu Liu, Long Wang, Zehong Shen, Zhen Peng, Haomin Liu, Maojun Zhang, Guofeng Zhang, and Xiaowei Zhou. 2023. Long-term visual localization with mobile sensors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 17245–17255.

[50] Jiqing Zhang, Yuanchen Wang, Wenxi Liu, Meng Li, Jinpeng Bai, Baocai Yin, and Xin Yang. 2023. Frame-event alignment and fusion network for high frame rate tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9781–9790.

[51] Zichao Zhang, Torsten Sattler, and Davide Scaramuzza. 2021. Reference pose generation for long-term visual localization via learned features and view synthesis. *International Journal of Computer Vision* 129, 4 (2021), 821–844.

[52] Han Zhou, Yi Gao, Xinyi Song, Wenxin Liu, and Wei Dong. 2020. LimbMotion: Decimeter-level Limb Tracking for Wearable-based Human-Computer Interaction. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 4, Article 161 (Sept. 2020), 24 pages. doi:10.1145/3369836

[53] Alex Zihao Zhu, Nikolay Atanasov, and Kostas Daniilidis. 2017. Event-based feature tracking with probabilistic data association. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 4465–4470.

[54] Alex Zihao Zhu, Dinesh Thakur, Tolga Özaslan, Bernd Pfrommer, Vijay Kumar, and Kostas Daniilidis. 2018. The multivehicle stereo event camera dataset: An event camera dataset for 3D perception. *IEEE Robotics and Automation Letters* 3, 3 (2018), 2032–2039.

[55] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. 2019. Unsupervised event-based learning of optical flow, depth, and egomotion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 989–997.

[56] Chen Zhu, Michael Meurer, and Christoph Günther. 2022. Integrity of visual navigation—developments, challenges, and prospects. *NAVIGATION: Journal of the Institute of Navigation* 69, 2 (2022).

[57] Zhiyu Zhu, Junhui Hou, and Xianqiang Lyu. 2022. Learning graph-embedded key-event back-tracing for object tracking in event clouds. *Advances in Neural Information Processing Systems* 35 (2022), 7462–7476.

[58] Zhiyu Zhu, Junhui Hou, and Dapeng Oliver Wu. 2023. Cross-modal orthogonal high-rank augmentation for rgb-event transformer-trackers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 22045–22055.