# EventBoost: Event-based Acceleration Platform for Real-time Drone Localization and Tracking

Hao Cao[1], Jingao Xu[1]✉, Danyang Li[1], Zheng Yang[1], Yunhao Liu[2]
[1]School of Software and BNRist, Tsinghua University [2]Global Innovation Exchange, Tsinghua University
{ihaocao, xujingao13, lidanyang1919, hmilyyz, yunhaoliu}@gmail.com

IEEE INFOCOM 2024 - IEEE Conference on Computer Communications | 979-8-3503-8350-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/INFOCOM52122.2024.10621101

***Abstract*—Drones have demonstrated their pivotal role in various applications such as search-and-rescue, smart logistics, and industrial inspection, with accurate localization playing an indispensable part. However, in high dynamic range and rapid motion scenarios, traditional visual sensors often face challenges in pose estimation. Event cameras, with their high temporal resolution, present a fresh opportunity for perception in such challenging environments. Current efforts resort to event-visual fusion to enhance the drone's sensing capability. Yet, the lack of efficient event-visual fusion algorithms and corresponding acceleration hardware causes the potential of event cameras to remain underutilized. In this paper, we introduce *EventBoost*, an acceleration platform designed for drone-based applications with event-image fusion. We propose a suit of novel algorithms through software-hardware co-design on Zynq SoC, aimed at enhancing real-time localization precision and speed. *EventBoost* achieves enhanced visual fusion precision and markedly elevated processing efficiency. The performance comparison with two state-of-the-art systems shows *EventBoost* achieves 24.33% improvement in accuracy with 30 ms latency on resource-constrained platforms.**

***Index Terms*—Sensor fusion, event camera, hardware acceleration, mobile sensing**

## I. Introduction

Given their flexibility and versatility, Unmanned Aerial Vehicles (UAVs) have become instrumental in various productive and life-saving operations [1]–[3]. Nowadays, UAVs are particularly crucial in challenging missions such as disaster recovery site surveys [4], mine explorations [5], and expansive surveillance of major transportation arteries [6], playing a critical role in protecting life and property.

However, the unique conditions where these missions take place pose distinct challenges to the conventional vision sensors on UAVs, including those equipped with higher framerate cameras. Specifically, they frequently encounter difficulties like overexposure, underexposure, handling high dynamic range (HDR [7]) environments, and dealing with motion blur [8]. These drawbacks hinder the accuracy and real-time performance of several fundamental UAV modules, such as self-localization, mapping, and obstacle avoidance [9]–[13].

Event cameras, new visual sensors that asynchronously record changes in pixel-level brightness, have attracted interest in academia and industry [14]. With a high sampling rate, low power use, and high dynamic range, event cameras have complementary advantages and show promise for complex tasks, *e.g.*, high-speed SLAM [15], motion tracking with HDR lighting [16], and fast obstacle avoidance [17].

By fully embracing the new advantages brought by event cameras, current innovations explore the fusion of event cameras and frame-based cameras for enhancing drone-based applications in challenging scenarios [18]. Existing event-frame fusion frameworks can be categorized into two aspects: $(i)$ *Frame-interpolation-based solutions* [19] work by accumulating event data, say every $5ms$ time-window, into virtual frames. These frames are then added to the original sequence of images taken by the camera. This process aims to increase the frame rate, which helps handle rapidly changing scenes. $(ii)$ *Optimization-based solutions* [20], [21] typically combine accumulated event data with video frames through motion and image models for joint optimization.

Despite the promising results of event-visual fusion algorithms, we observe several challenges when deploying them to more complex and dynamic real-world environments:

• **Image Over-reliance Hinders Precision**. Current event-visual fusion methods rely heavily on image data. That is, although mainstream state-of-the-art (SOTA) systems [19], [21] each have distinct processing strategies, they commonly emphasize image frames over events. This underutilizes the innate advantages of events, restricting algorithms within the standard dynamic range (SDR), exposure time, and low frame rate of conventional sensor images. Consequently, the natural merits of event cameras are sacrificed.

• **Hardware-Inefficiency Exacerbates Processing Latency**. Present-day hardware demonstrates considerable inefficiency in handling tasks related to event-image fusion. When these systems operate on CPUs, the simultaneous processing of events and images is impeded by context-switching interference, resulting in a substantial decline in event throughput. While GPUs are highly competent in managing conventional vision tasks [22], [23], they lack proficiency in dealing with the asynchronous, high-frequency stream of event data. On the other hand, FPGAs offer improvements in processing either images [24] or events [25] in isolation; however, they fall short in providing a comprehensive optimization across the entire fusion pipeline.

**Remark:** In summary, the lack of software algorithm and hardware platform support for event-image fusion results in a compromise on the overall system efficiency and accuracy, posing significant drawbacks for real-time drone-based applications in challenging scenarios.

**Our work.** To tackle the above challenges, we propose *EventBoost*, an acceleration platform designed for event-visual-fusion-based tasks. The design and implementation of *Event-*

✉ Jingao Xu is the corresponding author.

*Boost* follow a software-hardware co-design paradigm, making *EventBoost* achieve better system accuracy and efficiency. On the one hand, we design specialized algorithms aimed at augmenting the fusion of event and image; On the other hand, we develop dedicated hardware based on commercial Zynq systems-on-chip (SoCs) [26], [27] to accelerate the proposed whole software stack. Benefiting from *EventBoost*, UAV tasks like real-time localization, detection, and perception in challenging scenarios can be efficiently accomplished. Overall, our design excels in the following two aspects, spanning both software and hardware layer:

• On the algorithm side, we introduce a bimodal enhancement strategy, which not only efficiently fuses the complementary strengths of events and images, but also compensates for their limitations, significantly improving dual-modal feature utilization. Building upon it, the Hierarchical Pose Estimation strategy further reduces reliance on traditional images and better unleashes the potential of event cameras. Through the hierarchical solving structure, it estimates the pose from coarse to fine and effectively reuses prior computational results. Together, the two strategies break away from the over-dependency on images while ensuring in-depth integration between the modalities, achieving markedly improved overall accuracy.

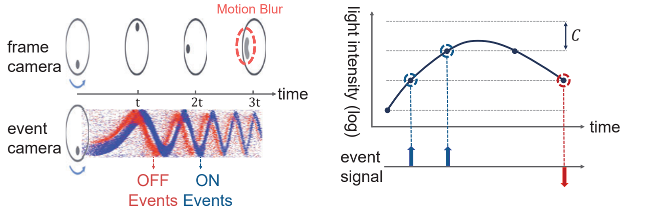
• On the hardware side, we propose a hardware-accelerated Image-aided Event Pose Tracker, which combines both modalities to optimize event feature extraction and noise filtering while enhancing feature stability. Besides, our proposed Fusion Optimization Processor introduces a modularized processing approach for event-visual fusion, which considerably minimizes hardware resource demands, elevates processing parallelism, and trims data transfer latencies. This method partitions the fusion problem into manageable segments, optimizing each independently while maintaining a global context. Finally, through the software-hardware co-design paradigm, we ensure the algorithms can run in real-time with low power on resource-constrained mobile platforms.

We implement *EventBoost* and deploy it on a UAV testbed for evaluation. Extensive experiments were conducted, including public datasets with extreme scene datasets as well as our self-collected dynamic and HDR datasets. We benchmark the localization and mapping accuracy and latency of our system against two state-of-the-art systems. The results demonstrate that *EventBoost* surpasses the accuracy of state-of-the-art system 24.33% while achieving 30 $ms$ latency, reaching the goal of real-time on resource-constrained platforms. To further showcase the real-world application effects of our accelerated platform, we integrated our system with ArduPilot, an open-source flight control system, and executed application case studies, encompassing detection, and obstacle avoidance. In all real-world use cases, *EventBoost* exhibited excellent performance, proving its efficacy and reliability.

In summary, the main contributions of this work are:

(1) We perform an in-depth analysis of current event-visual fusion algorithms and systematically identified their primary limitations, tracing the root causes, which lie in flawed algorithm design and insufficient hardware efficiency.

(2) We propose a tailored acceleration platform for event-visual fusion, comprising both accurate and robust fusion algorithms, as well as real-time, low-power hardware design. To our best knowledge, this is the first hardware acceleration platform for optimization-based event-visual fusion.



(a) (b)

Fig. 1. (a) Illustration of event stream. (b) Illustration of event generation.

(3) We fully implement the *EventBoost* acceleration platform through a software-hardware co-design strategy and evaluate its capabilities with extensive experiments and case studies. Comparisons with SOTA systems and real-world case studies demonstrate the significant advantages and immense application potential of our proposed platform.

## II. Background and Motivation

### A. Principle of Event Cameras

Event cameras are bio-inspired sensors that work differently from traditional frame-based counterparts. As shown in Fig. 1a, unlike conventional cameras that capture images at fixed time intervals, event cameras record asynchronous changes in pixel brightness, resulting in a stream of events at microsecond resolution [14]. The minimum sensing unit of event cameras is still pixels, but each pixel perceives independently. Once a pixel detects a predefined-magnitude change of intensity (*i.e.*, log intensity) in the scene, it will instantly output an event $e_k = (\boldsymbol{x}, t_k, p_k)$, encoding the occurrence time $t_k$ (at microsecond resolution), pixel location $\boldsymbol{x} = (u, v)^T$, and polarity $p_k$ (+1 for brighter and -1 for darker) of the intensity changes.

Different from traditional cameras that capture the entire scene at a fixed rate (typically 30Hz), event cameras only respond to pixel-level intensity changes. When the change of the log intensity of a pixel exceeds a predefined threshold $\pm C$ (illustrated in Fig. 1b), an event will be generated which contains the polarity of the change, indicating whether the pixel becomes brighter (ON event) or darker (OFF event).

Besides, owing to the event cameras' intrinsic ability to capture the intensity changes on a log scale, they exhibit an extensive dynamic range (up to 120 dB). Consequently, they can discern details spanning from extremely bright to deep dark.

### B. Event-Visual Fusing Pose Tracking

Event cameras and frame cameras provide complementary information. While event cameras capture instantaneous changes with high contrast, traditional cameras record the overall scene information. By combining data from these two sources, researchers have found that the accuracy and robustness of various visual tasks can be significantly enhanced [19], [20], [28]. Exciting fusion frameworks can be categorized into two aspects: $(i)$ *Frame-interpolation-based solutions* work by predicting or supplementing frame data using event data. By adding this event information, they try to create more detailed or frequent frames. However, these methods are limited in accuracy because they discard the event information and rely on the quality of the interpolated frames. $(ii)$ *Optimization-based solutions* work by minimizing the photometric errors between events and their frame predictions [20], [21]. This approach is more accurate than the frame-interpolation-based solutions, but it is computationally expensive and not suitable for real-time applications. Additionally, since it relies on image frames as references to minimize, the quality of images could significantly impact the precision of motion estimation.

### *C. Hardware Acceleration for Event-Visual Fusing*

Recently, event cameras have gained significant attention for their potential in low-power applications [29], [30]. While numerous studies have aimed at speeding up the processing of either event data or visual data [31]–[33], there remains a gap in the joint processing of both. Some research has successfully accelerated the processing of event or visual data individually, achieving certain advancements in their respective domains. However, when considering the combined processing of event and visual data, the performance of these acceleration techniques seems limited, failing to leverage the complementary nature of both. A more pressing challenge is the current lack of studies specifically targeting the acceleration of optimized event-visual fusion strategies. Although FPGA acceleration techniques based on least squares optimization have been applied in visual data processing [34], [35], the vast amount of data involved in event-visual fusion, compared to the sparse feature points of vision-based solutions, presents new challenges for these methods.

### *D. Limitations of Existing Approaches*

Event-based cameras offer exciting potential for drone localization and tracking tasks. Yet, many techniques, while effective in normal settings, face challenges in real-world conditions. Through our analysis of existing methods, we identify the following limitations:

• **Image Over-reliance Hinders Precision.** Existing event-visual fusion methods often lean heavily on traditional image data, which tends to overlook the unique benefits of event cameras. Algorithms tailored to the standard dynamic range, exposure time, and frame rate of conventional sensors miss the opportunity to fully tap into the distinct advantages of event cameras, especially in challenging environments dominated by rapid intensity changes and subtle motions. This bias has overshadowed the genuine prowess of events. Event-driven data, when used properly, can be very effective. However, these event-only strategies suffer from accumulated errors over an extended period. How to effectively fuse event and visual data to achieve the best of both worlds remains a challenge.

• **Hardware-Inefficiency Exacerbates Processing Latency.** Within the optimization-based solutions, the most significant computational load stems from minimizing the photometric errors between event predictions and actual frames. Regrettably, there are no dedicated hardware acceleration schemes currently in place explicitly designed for this core task. As a result, the efficiency of the overarching fusion strategy remains suboptimal, failing to truly elevate the system's capabilities to new heights. The lack of holistic hardware solutions tailored to the specific computational requirements of the event-visual method currently impedes seamless and efficient integration of event and visual data, resulting in performance bottlenecks and suboptimal system efficiency.

In summary, although event cameras hold great potential for tasks such as drone localization and tracking in challenging areas, there still lack effective algorithms and hardware support to fully unleash their potential. In this work, we aim to overcome the above two challenges by proposing an algorithm pipeline and a software-hardware co-design accelerator.

## III. System Overview

To solve the challenges highlighted in the previous sections, we introduce *EventBoost*: a cutting-edge event-based acceleration platform for real-time drone localization and tracking. As illustrated in Fig. 2, we detail its primary modules:

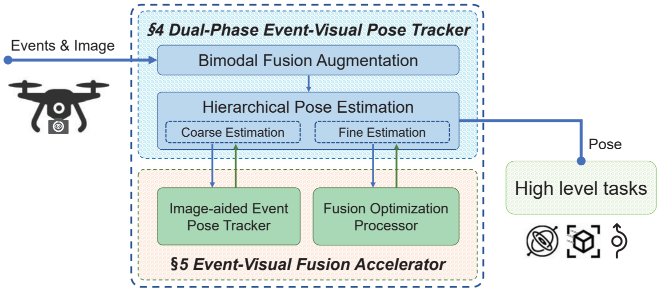

Events & Image
§4 Dual-Phase Event-Visual Pose Tracker
Bimodal Fusion Augmentation
Hierarchical Pose Estimation
Coarse Estimation
Fine Estimation
Pose
High level tasks
Image-aided Event Pose Tracker
Fusion Optimization Processor
§5 Event-Visual Fusion Accelerator

Fig. 2. *EventBoost* architecture.

• **Dual-Phase Event-Visual Pose Tracker**: as the primary tracking module of *EventBoost*, it not only precisely fuses event data and image data but is also designed for the computational-constrained embedded platforms. This module includes several hardware-friendly strategies to conserve computational resources, ensuring real-time and efficient pose estimation across complex environments.

• **Event-Visual Fusion Accelerator**: as a cornerstone of the *EventBoost* platform, this module handles the processing of event and image data. By using the mid-stage results from both event and image data to speed up their processing. Further, it is also designed for accelerating the optimization-based fusion framework, facilitating the processing of the most computationally demanding tasks.

The synergy between *EventBoost*'s two core modules creates a comprehensive solution that fully utilizes the capabilities of event cameras. The goal of *EventBoost* is to provide accurate drone localization and tracking to serve various upper-layer applications.

## IV. Dual-Phase Event-Visual Pose Tracker

We propose a novel algorithm named Dual-Phase Event-Visual Pose Tracker that fully leverages the high temporal resolution of event data and high spatial resolution visual data. The algorithm shares a similar pipeline with traditional optimization-based fusion methods. Our system embodies the concept of bimodal complementary augmentation, where each modality augments the other's shortcomings. In the following sections, we elaborate on the workflow of our algorithm, as well as the two core modules that drive its performance.

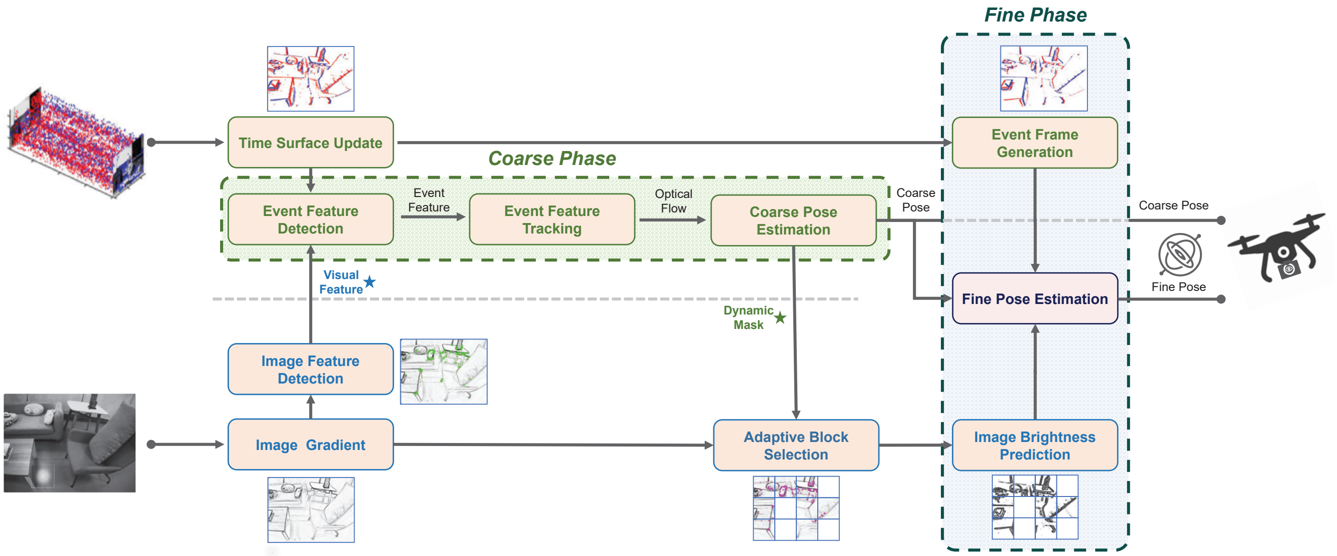


Fig. 3. Workflow of Dual-Phase Event-Visual Pose Tracker

### A. Algorithm Workflow

Fig. 3 illustrates the workflow of Dual-Phase Event-Visual Pose Tracker. Our algorithm starts by receiving both event and image data as inputs. These two types of data are preprocessed separately to augment their quality and accuracy. And then fused to get the pose. The processing workflow is as follows:

- **Image**: We first extract the FAST features [36] of the image, which will be preparing for subsequent mapping and assisting in event filtering, as detailed in Section IV-B. After undergoing processes such as gradient computation, the image is partitioned into $N \times N$ blocks ($N = 8$ in our settings). Among these, blocks with no dynamic objects will be adaptively selected for the next processing. Following this, an image brightness increment model is employed to predict intensity changes in these chosen blocks. Once these steps are completed, the selected blocks are fused with accumulated event data.
- **Event**: The event data is initially directed to a time surface [14] update for feature detection and tracking phase, from which we extract condensed representations from events and capitalize on them for coarse pose estimation. Concurrently, based on the estimated pose, dynamic areas within the scene are masked. This mask aids in the adaptive block selection for images, as elaborated in Section IV-B. Subsequently, events are accumulated to generate an event frame that will later be integrated with images.
- **Event-Image Fusion**: Once we have acquired the accumulated event frame and the selected image blocks, we integrate the bimodal data and precisely estimate the pose(as detailed in Section IV-C).

Throughout the algorithm workflow, two core modules play a pivotal role in determining the performance of the approach, which are: (i) Bimodal Fusion Augmentation, and (ii) Hierarchical Pose Estimation. The following parts delve deeper into these three modules.

### B. Bimodal Fusion Augmentation

Event and image data synergistically enhance pose estimation: event data provides high temporal resolution, filling in image data's temporal gaps, while image data offers detailed spatial resolution and brightness, balancing event data's limitations. Our fusion approach leverages both for improved accuracy and robustness.

**Image-Augmented Event Feature Detection.** Event-driven data inherently struggles with noise interference, complicating the extraction of precise feature points, particularly in dynamic scenarios. On the other hand, visual data, with its clear spatial detail and absolute intensity, facilitates the extraction of high-quality, stable feature points, which are crucial for accurately tracking the more variable event data.

Motivated by this observation, we propose a strategy wherein feature points extracted from the visual data are delineated as "trust regions", providing a robust reference for the event data. Specifically, events located within a distance $k$ of these trust regions are more likely to represent genuine scene dynamics than mere noise or false-positive events, due to their close association with the visual feature points. To harness this advantage, we assign higher weights to these events during the event feature point extraction phase. Our weight assignment strategy can be mathematically articulated as:

$$W(e_k) = \begin{cases} 1 + \frac{\|\boldsymbol{x}-\boldsymbol{c}\|_1}{k}, & \text{if } \|\boldsymbol{x}-\boldsymbol{c}\|_1 \le k \\ 1, & \text{if } \|\boldsymbol{x}-\boldsymbol{c}\|_1 > k \end{cases}$$

where $e_k = (\boldsymbol{x}, t_k, p_k)$ and $W(e_k)$ denotes the weight of event $e_k$, $\boldsymbol{c}$ represents the position of the proximate visual feature point, and $\|\cdot\|_1$ signifies the L1 norm. The weight is maximized when the event's distance to the feature point is zero and minimized when the distance equals $k$. This paradigm is designed for hardware implementations like FPGA.

**Events-Augmented Adaptive Block Selection.** Traditional image data, with its discrete temporal nature, finds it difficult to capture continuous changes in dynamic scenes, particularly for fast-moving objects. Event cameras, however, excel in detecting minute dynamic changes, effectively differentiating moving objects from static backgrounds.

To address these issues, we present a method that utilizes event data to enhance dynamic region masking in images. As detailed in Algorithm 1, we first extract feature points from event data and compute the optical flow. Using the RANSAC algorithm [37], we then deduce scene pose changes. During this, optical flows inconsistent with the pose estimation are detected, often indicating dynamic objects. These regions are then masked in the image to reduce prediction errors, effectively integrating both event and image data and improving accuracy in dynamic settings.

**Algorithm 1** Events-Augmented Adaptive Block Selection

```
Require: 2D Image $I$, Event data $E$, 3D map $M$
Extract feature points from $E$
Match event features to 3D map points for 2D-3D correspondences
Compute optical flow of $E$
for iteration $i$ in $N$ do
    Select a subset of matched 2D-3D points
    Estimate pose $P_i$ using PnP
    Count inliers by projecting 3D points with $P_i$
end for
Select pose $P$ with most inliers
Find mismatched regions in optical flow against pose $P$
Segment $I$ into blocks
Mask mismatched regions in $I$ as dynamic
return Masked image $I$
```

### C. Hierarchical Pose Estimation.

Optical flow techniques, though widely used in traditional visual processing, haven't been maximally exploited in current event-image fusion systems. We believe optical flow provides a precise portrayal of short-term pose changes using event data and is a direct, efficient way to leverage the high temporal resolution unique to event data. Building on this, we present a two-tiered hierarchical pose estimation framework:

- **Coarse Phase**: During the blind-time between two image frames, the pose is estimated continuously by extracting feature points and computing optical flow from event data. This approach facilitates accurate real-time pose estimations. Accumulating the short-term pose variations between two frames provides a relatively precise preliminary estimation, especially beneficial when the quality or availability of image data is compromised.
- **Fine Phase**: When a new image frame is received, we utilize the event data accumulated since the last image capture, employing both the event generation model and a brightness increment prediction method. These models collaboratively optimize the pose change between frames. While this step is computationally intensive, leveraging the pose estimated from the prior phase (Coarse Phase) enhances efficiency and accuracy.

In conclusion, our two-phase strategy combines the rapid feedback from event data with the detail-rich nature of traditional visual cues. The Coarse Phase provides quick estimations, which the Fine Phase then refines, making use of the initial data to cut down on computing needs. This structured approach bolsters our system's efficiency and reliability, making it robust and accurate for pose estimation in dynamic environments.

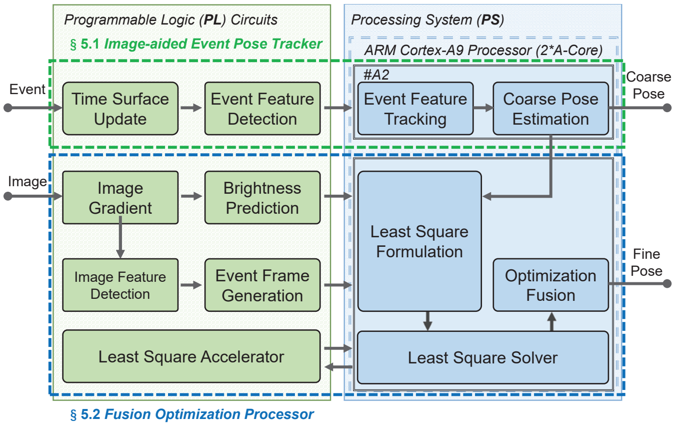


Fig. 4. Architecture of Event-Visual Fusion Accelerator.

## V. Event-Visual Fusion Accelerator

Considering the previously discussed challenges, low-power SoCs like Zynq, which combine both FPGA and CPU, emerge as ideal candidates for handling stream processing and demanding computational tasks. This becomes even more pertinent when focusing on real-time processing and parallel multitasking. Building on this realization, we adopt a collaborative hardware-software design approach and design an event-visual fusion accelerator: Event-Visual Fusion Accelerator (EVFA). This accelerator prioritizes real-time, efficient processing in mobile environments. At its heart, the design taps into the combined strengths of the FPGA and CPU in Zynq, as depicted in Fig. 4. This arrangement ensures that intensive computations and stream processing are assigned to the best-suited components. Central to the EVFA are two modules: *Image-aided Event Pose Tracker (IEPT)* and *Fusion Optimization Processor (FOP)*. Together, they enable real-time fusion of event and visual data on mobile devices, effectively capitalizing on the advantages of a unified hardware-software approach. The following parts will detail how these modules interact and delve into their technical features.

### A. Image-aided Event Pose Tracker

**Image-aided Event Feature Extraction.** Time Surface (TS) is a lightweight representation of event streams, which can well adapt to the dynamic nature of event streams. TS is a 2D map where every pixel value displays the timestamp of the last event that occurred at that pixel. We leverage Polarity Time-Surface (P-TS) [38], whose pixels also display the polarity of the last event.

Our algorithm operates as follows:

- Image-aided Initialization: We utilize visual assistance, *i.e.* FAST corner detection, to establish a set of visual feature points. This step not only assigns an initial location to the feature points but also effectively reduces noise within the event stream by filtering out events distant from the visual feature points.
- Time Surface Update: We define a P-TS matrix of the same size as the image, initialized to zero. With each incoming event, its position in the P-TS matrix gets updated, capturing the event's timestamp and polarity in real-time. The timestamps within the P-TS can be employed to gauge the relative activity level of an event.
- Event Feature Detection: If an event's timestamp is considerably more recent than the average timestamp of its vicinity and it is close to other visual feature points, it is identified as a new feature point. In the absence of visual feature point guidance, we extract feature points directly based on the recency of timestamps.
- Feature Aging: To maintain the timeliness and accuracy of the feature points, we introduce an aging mechanism. If a feature point's timestamp in the P-TS matrix hasn't been updated for an extended period, indicating no new events in its area, it's considered 'aged' and is subsequently removed from the collection.

**Optical Flow and Pose Estimation.** Given the intricate nature of the process, yet the manageable computational complexity, both optical flow estimation and pose estimation are implemented on the PS.

Our algorithm's workflow is structured as follows:

- Event Feature Tracking: Upon receiving a new set of event feature point data from the PL, which has been preliminarily verified for its validity, we embark on the subsequent steps. The initial operation entails searching for the best match within our currently maintained active feature point list, leveraging this data. Through this, we calculate the motion trajectory for each feature point, thereby deriving the optical flow information.
- Coarse Pose Estimation: Subsequently, with our pre-existing 3D map, where each feature point is endowed with depth information from the real world, we harness the optical flow data and apply the PnP algorithm to deduce the camera's pose. This estimated pose serves a dual purpose. While it offers a coarse pose estimation in scenarios with suboptimal visual images, it is also accumulated as an initial value for the fine pose optimization when fused with image data. The cycles of optical flow estimation and pose determination recur until a subsequent video frame is received. The pose estimations between these frames are aggregated, contributing to the fused pose optimization.

### B. Fusion Optimization Processor

Event-image fusion, at its core, revolves around solving a least squares problem. While current technical solutions have sought FPGA-based optimization for least squares computations [35], this is fraught with challenges in the event-image fusion domain, notably constraints in resources, limited parallelism, and significant transmission latency.

In response to these challenges, we present the software-hardware co-design *Fusion Optimization Processor (FOP)*. Central to our approach is the optimal subproblem division strategy, which is achieved by our unique algorithmic design. Specifically, by utilizing event-based optical flow and pose estimation, we eliminate dynamic objects in the scene (illustrated in Section IV-B). This ensures that each data block arises solely from camera motion, and hence, shares the same motion pattern. This strategy offers three main benefits:

1) **Resource Efficiency**: By processing smaller blocks of data, the demand on FPGA hardware resources is reduced.
2) **Enhanced Parallelism**: With a more judicious partitioning of data blocks, parallel processing becomes more efficient.
3) **Latency Reduction**: Reduced data transfer between the Processing System (PS) and the Programmable Logic (PL) leads to quicker data transmission times.

Detailing our solution further, the following critical steps are involved:

- Block Selection: As outlined in Section IV-B, we incorporate event-based optical flow and pose estimation to efficiently filter out dynamic blocks.
- Block Pre-processing: Leveraging the PL, we accumulate event frames and pre-process images, generating blocks of data that include pre-processed images and event frames.
- Least Square Formulation: After pre-processing, the blocks are transmitted to the PS, where a least squares optimization problem is constructed for each block. Subsequently, these problems are relayed back to the corresponding solver modules in the PL.
- Least Square Solving: Established works implemented on the PL perform the computation to solve the least squares problems, yielding the pose for each block.
- Optimization Fusion: Initial steps include consistency checks for each block result. This is achieved by evaluating the pose estimation error against a preset threshold. Blocks exceeding this error threshold are deemed outliers and excluded from subsequent fusion. The poses of the remaining blocks undergo a weighted fusion, providing the final camera pose.

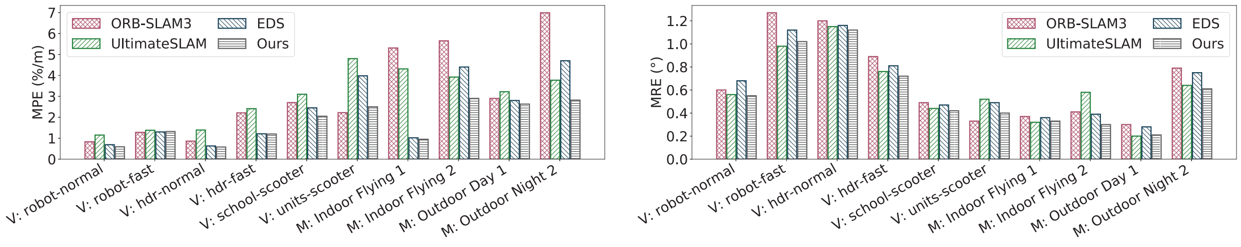
Aiming to further reduce algorithmic latency, we've meticulously architected a dual DMA system. By facilitating unidirectional data flow and fostering efficient hardware-software co-design, we've established a potent pipelined processing mechanism. This design substantially ramps up processing speed, ensuring real-time requirements are met.

In conclusion, our adoption of the optimal subproblem division strategy, coupled with a thoughtfully designed data pipelining mechanism, effectively addresses the seminal challenges in event-image fusion resource constraints, parallelism issues, and data transmission delays. This infuses our approach with marked improvements in performance and real-time processing.

## VI. Evaluation

### A. Experiment Methodology

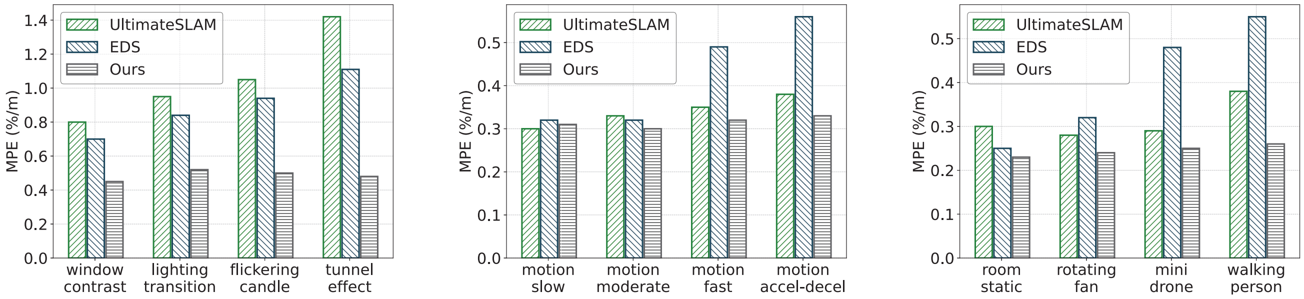
**Experiment Data Description.** We conduct extensive experiments on both public datasets and real-world scenarios to evaluate the performance of our proposed platform. Specifically, we first evaluate our system on two public datasets:



(a) Mean Position Error.

(b) Mean Rotation Error.

Fig. 5. Overall Localization Performance.



(a) Impact of HDR.

(b) Impact of Moving Speed.

(c) Impact of Scene Dynamics.

Fig. 6. Robustness Evaluation.

VECtor [39] (marked as V) and MVSEC [40] (marked as M) to demonstrate its overall performance in different scenarios. We then evaluate the robustness and real-time performance of our system on our self-collected dataset, which contains fast 6-Dof motion, HDR scenes and scenes with various dynamics. The dataset was collected using stereo DAVIS 346 event cameras carried by an AMOV-P450 drone, providing visual, event and IMU data. In our case, only the left camera data will be used. We deploy an OptiTrack motion capture system with four cameras to collect ground truth.

**Experiment Platform.** The experiments platform is Zynq-7020 SoC, including a Xilinx Artix-7 FPGA and a Dual-Core ARM Cortex-A9. The frequencies of the ARM core and FPGA are 667MHz and 150MHz, respectively. The system distribution version is Linux Debian 10.

**Evaluation Metrics.** To evaluate the accuracy of our system, we conducted a quantitative analysis using two metrics: mean position error (MPE, %) and mean rotation error (MRE, °/m). When using monocular methods, a scale transformation is applied to estimate the absolute trajectory, and data from the left camera will be used. We aligned the estimated trajectory with the ground truth using a 6-DOF transformation in SE3, which was calculated using the tool [41].

**Baseline Methods.** We compare *EventBoost* with visual-only and event-visual VO methods. Three baselines are selected:

- ORB-SLAM3 [42]: A state-of-the-art versatile visual SLAM system.
- UltimateSLAM [19]: An integrated system that harnesses events, images, and IMU measurements. UltimateSLAM is the representation system of frame-interpolation-based solutions.
- EDS [21]: Event-camera based direct sparse odometry. EDS is the representation system of optimization-based solutions.

### B. Overall Performance

TABLE I
PLATFORM OVERHEAD

| Platform | PC | Ours | Comparision |
|---|---|---|---|
| Feature Detection / ms | 0.6-30 | 0.1-0.2 | > **6x** faster |
| Optical Flow / ms | 8.5 | 1.5 | > **5x** faster |
| Coarse Pose Estimation / ms | 15 | 5 | > **3x** faster |
| Fine Pose Estimation / ms | 126 | 20.1 | > **6x** faster |
| Total Power / W | 17.5 | 4.8 | **3x** more efficient |

**Localization Accuracy.** Fig. 5 depicts the overall localization performance of our SLAM system in comparison to the other three competitive methods. As shown in Fig. 5a, *EventBoost* consistently achieves superior localization accuracy across different scenarios in most cases. The average MPE error of *EventBoost* is 1.754%/m, outperforming ORB-SLAM3, UltimateSLAM and EDS by 43.33%, 40.44%, 24.33%, respectively. *EventBoost* performed especially well in challenging environments marked by strong dynamic changes, fast motions, and HDR conditions.

As depicted in Fig. 5b, the MRE error of *EventBoost* is 0.56°, outperforming ORB-SLAM3, UltimateSLAM and EDS by 14.00%, 7.64%, 12.75%, respectively. Notably, even without integrating an Inertial Measurement Unit (IMU), which is often used to boost the accuracy of SLAM systems, *EventBoost* delivered results that were comparable to those of IMU-enhanced systems like UltimateSLAM.

In summary, by combining frames and events, our method not only proves resilient across diverse conditions but also consistently posts lower localization errors. This data strongly suggests that our system offers a robust and reliable solution for real-world SLAM applications.

**End-to-end Latency.** We further evaluate the end-to-end (E2E) latency (defined as the time taken from receiving an image frame to getting the fine pose). Our tests spanned three scenarios: simple, moderate, and complex. In all these tests, not only did we ensure consistent latency, but we also met the real-time requirement (30fps). This achievement underscores our system's ability to run in real-time on mobile platforms.

**Platform Overhead.** As shown in Table I, our platform consistently outpaced a PC setup (Intel i5-8259U), achieving over 6x speed in feature detection and fine pose estimation, over 5x in optical flow, 3x in coarse pose estimation, and proving 3 times more energy-efficient in overall power consumption.

### *C. Robustness Evaluation*

Due to the inferior performance of traditional vision-based systems in the overall localization experiments, especially under conditions with HDR, rapid movement, and dynamic scenes, our robustness experiments only focused on comparisons with the other two event-based systems.

**Impact of HDR.** We investigate the influence of HDR on various systems. To thoroughly evaluate this, various HDR phenomena were artificially triggered in our test environment: significant luminance disparities between the insides and outsides of windows(window contrast), transitions from lights-on to lights-off (lighting transition), the flickering light of a burning candle (flickering candle), and the abrupt brightening experienced when exiting a tunnel which simulated using curtain pulling (tunnel effect). The results, as shown in Fig. 6a, reveal some clear disparities in performance:

UltimateSLAM, which heavily relies on visual feature points, performed poorly under HDR conditions. This can be attributed to the frequent occurrences of overexposure and underexposure in HDR, which can be challenging for such systems to handle. Similarly, EDS, which is based on a frame prediction model, also encountered difficulties, resulting in suboptimal outcomes in HDR settings.

Contrastingly, *EventBoost* displays commendable stability and proves its efficacy under HDR conditions. This underscores its potential to reliably function in real-world environments with varying lighting conditions.

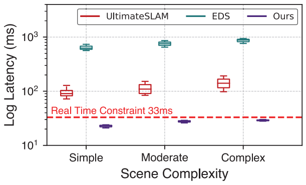


Fig. 7. Overall End-to-end Latency.

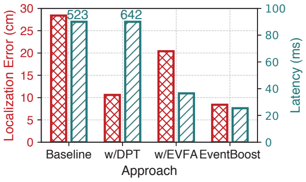


Fig. 8. Impact of Modules.

**Impact of Moving Speed.** Next, we explore the impact of moving speed on the performance of the systems. As illustrated in Fig. 6b, *EventBoost* continues to outshine the others. UltimateSLAM manages to deliver fairly consistent results even under rapid motion with assistance from IMU. However, EDS falters in such scenarios. The motion blur introduced by swift movements adversely affects EDS's image gradient calculations, leading to inaccurate pose estimations.

In contrast, our system capitalizes on Coarse Pose Estimation with the event camera. This inherent feature empowers *EventBoost* to maintain stable and precise localization results, even when confronted with high-speed motion. This further cements the robustness and versatility of our proposed system across varying operational conditions.

**Impact of Scene Dynamics.** We finally assess the impact of scene dynamics on the performance. In our experimental setup, we introduced varying degrees of dynamics: a completely static room (room static), a working desk fan (rotating fan), a flying mini-drone (mini drone), and a person walking across the test area (walking person).

As depicted in Fig. 6c, the performance of EDS deteriorates sharply as the dynamism of the scene increases, primarily due to the breakdown of its frame prediction model. UltimateSLAM, leveraging event-frame technology, manages to filter out some of the dynamic entities in the scene. However, the need for accumulating event frames over time means the system still experiences a slight performance hit in dynamic environments.

In contrast, *EventBoost* stands resilient against these challenges. By harnessing event optical flow and Coarse Pose Estimation, it effectively isolates and discards the influence of dynamic objects. This capability ensures that our system delivers both accurate and consistent results, regardless of the scene's dynamics. This experiment further solidifies our claim that *EventBoost* is adept at handling a wide array of environmental challenges

### *D. Ablation Study*

**Contributions of Each Module.** We examine how Dual-Phase Event-Visual Pose Tracker (DPT) and Event-Visual Fusion Accelerator (EVFA) contribute to *EventBoost*. To break this down, we incrementally incorporated DPT and EVFA into our baseline system (when EVFA is omitted, DPT runs exclusively on the CPU). Following each integration, we reassessed the localization accuracy and end-to-end latency. In Fig. 8, the Baseline without the two modules has a localization error of 28.4 $cm$ and latency of $523ms$. Integrating the DPT module reduces the error to 10.6 $cm$, though latency rises to $642ms$ due to increased CPU demands. Adding the EVFA further reduces the error to 25.4 $cm$, but again, latency increases, a side effect of our DPT's design. Combining both DPT and EVFA, *EventBoost* optimally balances accuracy and speed, marking a significant advancement in SLAM system performance.

In summation, our ablation study reaffirms the pivotal role played by DPT and EVFA in shaping the performance contours of *EventBoost*, each contributing its unique strength to make the system robust and efficient.

## VII. Conclusion

In this study, we highlighted the challenges of fusing event-driven and image data for UAVs in challenging conditions. While event cameras promise solutions, their full potential is curtailed by algorithm and hardware shortcomings. Our *EventBoost* platform, developed through a software-hardware co-design approach, adeptly addresses these issues. The experiment results indicate not only improved accuracy and latency but also its practical value in UAV tasks. In essence, our work offers a blueprint for enhancing the future efficiency and reliability of UAV systems.

## Acknowledgment

We sincerely thank the MobiSense group for their invaluable support. This work is supported in part by the National Key Research Plan under grant No. 2021YFB2900100, the NSFC under grant No. 62372265, No. 62302254. No. 62332016, No. 62202262, and No. 62272462.

## References

[1] B. Vergouw, H. Nagel, G. Bondt, and B. Custers, “Drone technology: Types, payloads, applications, frequency spectrum issues and future developments,” *The Future of Drone Use: Opportunities and Threats from Ethical and Legal Perspectives*, pp. 21–45, 2016.

[2] DJI Industrial Solutions. DJI Official. [Online]. Available: https://www.dji.com/products/industrial

[3] Amazon Prime Air prepares for drone deliveries. [Online]. Available: https://www.aboutamazon.com/news/transportation/amazon-prime-air-prepares-for-drone-deliveries

[4] A. Restas *et al.*, “Drone applications for supporting disaster management,” *World Journal of Engineering and Technology*, vol. 3, no. 03, p. 316, 2015.

[5] J. Shahmoradi, E. Talebi, P. Roghanchi, and M. Hassanalian, “A comprehensive review of applications of drone technology in the mining industry,” *Drones*, vol. 4, no. 3, p. 34, 2020.

[6] A. Loder, T. Otte, and K. Bogenberger, “Using large-scale drone data to monitor and assess the behavior of freight vehicles on urban level,” *Transportation Research Record*, vol. 2676, no. 11, pp. 496–507, 2022. [Online]. Available: https://doi.org/10.1177/03611981221093620

[7] “High dynamic range,” Jul. 2023, page Version ID: 1164769087. [Online]. Available: https://en.wikipedia.org/w/index.php?title=High_dynamic_range&oldid=1164769087

[8] “Motion blur,” Jun. 2023, page Version ID: 1161456536. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Motion_blur&oldid=1161456536

[9] D. Falanga, K. Kleber, and D. Scaramuzza, “Dynamic obstacle avoidance for quadrotors with event cameras,” *Science Robotics*, vol. 5, no. 40, p. eaaz9712, 2020.

[10] J. Xu, H. Cao, D. Li, K. Huang, C. Qian, L. Shangguan, and Z. Yang, “Edge Assisted Mobile Semantic Visual SLAM,” in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, p. 10.

[11] H. Cao, J. Xu, D. Li, L. Shangguan, Y. Liu, and Z. Yang, “Edge assisted mobile semantic visual slam,” *IEEE Transactions on Mobile Computing*, vol. 22, no. 12, pp. 6985–6999, 2023.

[12] S. Bu, Y. Zhao, G. Wan, and Z. Liu, “Map2DFusion: Real-time incremental UAV image mosaicing based on monocular SLAM,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 4564–4571. [Online]. Available: http://ieeexplore.ieee.org/document/7759672/

[13] J. Xu, H. Cao, Z. Yang, L. Shangguan, J. Zhang, X. He, and Y. Liu, “{SwarmMap}: Scaling Up Real-time Collaborative Visual {SLAM} at the Edge,” pp. 977–993. [Online]. Available: https://www.usenix.org/conference/nsdi22/presentation/xu

[14] G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, “Event-based vision: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 1, pp. 154–180, 2022.

[15] K. Ok, K. Liu, K. Frey, J. P. How, and N. Roy, “Robust object-based slam for high-speed autonomous navigation,” in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 669–675.

[16] T.-H. Wu, C. Gong, D. Kong, S. Xu, and Q. Liu, “A novel visual object detection and distance estimation method for hdr scenes based on event camera,” in *2021 7th International Conference on Computer and Communications (ICCC)*. IEEE, 2021, pp. 636–640.

[17] D. Falanga, K. Kleber, and D. Scaramuzza, “Dynamic obstacle avoidance for quadrotors with event cameras,” vol. 5, no. 40, p. eaaz9712. [Online]. Available: https://www.science.org/doi/10.1126/scirobotics.aaz9712

[18] A. Tomy, A. Paigwar, K. S. Mann, A. Renzaglia, and C. Laugier, “Fusing event-based and rgb camera for robust object detection in adverse conditions,” in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 933–939.

[19] A. R. Vidal, H. Rebecq, T. Horstschaefer, and D. Scaramuzza, “Ultimate SLAM? Combining Events, Images, and IMU for Robust Visual SLAM in HDR and High Speed Scenarios,” vol. 3, no. 2, pp. 994–1001. [Online]. Available: http://arxiv.org/abs/1709.06310

[20] D. Gehrig, H. Rebecq, G. Gallego, and D. Scaramuzza, “EKLT: Asynchronous Photometric Feature Tracking Using Events and Frames,” vol. 128, no. 3, pp. 601–618. [Online]. Available: http://link.springer.com/10.1007/s11263-019-01209-w

[21] J. Hidalgo-Carrió, G. Gallego, and D. Scaramuzza, “Event-aided direct sparse odometry,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[22] Kwang-Ting Cheng and Yi-Chu Wang, “Using mobile GPU for general-purpose computing - a case study of face recognition on smartphones,” in *Proceedings of 2011 International Symposium on VLSI Design, Automation and Test*. IEEE, pp. 1–4. [Online]. Available: http://ieeexplore.ieee.org/document/5783575/

[23] B. Nagy, P. Foehn, and D. Scaramuzza, “Faster than FAST: GPU-Accelerated Frontend for High-Speed VIO.” [Online]. Available: http://arxiv.org/abs/2003.13493

[24] D. G. Bailey, *Design for embedded image processing on FPGAs*. John Wiley & Sons, 2011.

[25] M. Liu, W.-T. Kao, and T. Delbruck, “Live demonstration: A real-time event-based fast corner detection demo based on fpga,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[26] “System on a chip,” Jul. 2023, page Version ID: 1163919838. [Online]. Available: https://en.wikipedia.org/w/index.php?title=System_on_a_chip&oldid=1163919838

[27] Xilinx, “Zynq-7000 soc: A comprehensive processor and fpga system,” 2023, accessed: 2024-01-11. [Online]. Available: https://www.xilinx.com/products/silicon-devices/soc/zynq-7000.html

[28] S. Bryner, G. Gallego, H. Rebecq, and D. Scaramuzza, “Event-based, direct camera tracking from a photometric 3d map using nonlinear optimization,” in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 325–331.

[29] A. Amir, B. Taba, D. Berg, T. Melano, J. McKinstry, C. Di Nolfo, T. Nayak, A. Andreopoulos, G. Garreau, M. Mendoza *et al.*, “A low power, fully event-based gesture recognition system,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7243–7252.

[30] B. Ramesh, A. Ussa, L. Della Vedova, H. Yang, and G. Orchard, “Low-power dynamic object detection and classification with freely moving event cameras,” *Frontiers in neuroscience*, vol. 14, p. 135, 2020.

[31] I. Alzugaray and M. Chli, “Asynchronous corner detection and tracking for event cameras in real time,” *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3177–3184, Oct 2018.

[32] E. Mueggler, C. Bartolozzi, and D. Scaramuzza, “Fast event-based corner detection,” in *British Machine Vision Conference (BMVC)*, 2017.

[33] V. Vasco, A. Glover, and C. Bartolozzi, “Fast event-based harris corner detection exploiting the advantages of event-driven cameras,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 4144–4149.

[34] Q. Liu, S. Qin, B. Yu, J. Tang, and S. Liu, “$\pi$-BA: Bundle Adjustment Hardware Accelerator Based on Distribution of 3D-Point Observations,” vol. 69, no. 7, pp. 1083–1095.

[35] S. Qin, Q. Liu, B. Yu, and S. Liu, “$\pi$-BA: Bundle Adjustment Acceleration on Embedded FPGAs with Co-observation Optimization,” in *2019 IEEE 27th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, pp. 100–108.

[36] E. Rosten and T. Drummond, “Machine learning for high-speed corner detection,” in *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9*. Springer, 2006, pp. 430–443.

[37] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[38] J. Xu, D. Li, Z. Yang, Y. Zhao, H. Cao, Y. Liu, and L. Shangguan, *Taming Event Cameras with Bio-Inspired Architecture and Algorithm: A Case for Drone Obstacle Avoidance*. New York, NY, USA: Association for Computing Machinery, 2023. [Online]. Available: https://doi.org/10.1145/3570361.3613269

[39] L. Gao, Y. Liang, J. Yang, S. Wu, C. Wang, J. Chen, and L. Kneip, “VECtor: A Versatile Event-Centric Benchmark for Multi-Sensor SLAM,” vol. 7, no. 3, pp. 8217–8224. [Online]. Available: http://arxiv.org/abs/2207.01404

[40] A. Z. Zhu, D. Thakur, T. Özaslan, B. Pfrommer, V. Kumar, and K. Daniilidis, “The Multivehicle Stereo Event Camera Dataset: An Event Camera Dataset for 3D Perception,” *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2032–2039, Jul. 2018, conference Name: IEEE Robotics and Automation Letters.

[41] M. Grupp, “evo: Python package for the evaluation of odometry and slam.” https://github.com/MichaelGrupp/evo, 2017.

[42] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, “ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM.” [Online]. Available: http://arxiv.org/abs/2007.11898