EV-Tach: A Handheld Rotational Speed Estimation System with Event Camera

Guangrong Zhao, Yiran Shen*, *Senior Member, IEEE*, Ning Chen, Pengfei Hu, *Member, IEEE*, Lei Liu, *Member, IEEE*, and Hongkai Wen, *Member, IEEE*

Abstract—Rotational speed is one of the important metrics to be measured for calibrating electric motors in manufacturing, monitoring engines during car repairs, detecting faults in electrical appliance and more. However, existing measurement techniques either require prohibitive hardware (e.g., high-speed camera) or are inconvenient to use in real-world application scenarios. In this paper, we propose, *EV-Tach*, a novel handheld rotational speed estimation system that utilizes emerging imaging sensors known as event cameras or dynamic vision sensors (DVS). The pixels of DVS work independently and trigger an event as soon as a per-pixel intensity change is detected, without global synchronization like conventional RGB cameras. Thus, its unique design features high temporal resolution and generates sparse events, which benefits the high-speed rotation estimation. To achieve accurate and efficient rotational speed estimation, a series of signal processing algorithms are specifically designed for the event streams generated by event cameras on an embedded platform. First, a new cluster-centroids initialization module is proposed to initialize the centroids of the clusters to address the issue that common clustering approaches are easy to fall into a local optimal solution without proper initial centroids. Second, an outlier removal module is designed to suppress the background noise caused by subtle hand movements and host devices vibrations. Third, a coarse-to-fine alignment strategy is proposed with an event stream alignment method to obtain angle of rotation and achieve accurate estimation for rotational speed in a large range. With these bespoke components, *EV-Tach* is able to extract the rotational speed accurately from the event stream produced by an event camera recording rotary targets. According to our extensive evaluations under controlled and practical experiment settings, the Relative Mean Absolute Error (RMAE) of *EV-Tach* is as low as $0.3‰$, which is comparable to the state-of-the-art laser tachometer under fixed measurement mode. Moreover, *EV-Tach* is robust to subtle movement of user's hand and dazzling light outdoor, therefore, can be used as a handheld device under challenging lighting condition, where the laser tachometer fails to produce reasonable results. To speed up the processing of *EV-Tach* and reduce its resource consumption on embedded devices, event stream is significantly downsampled by merging neighboring events while preserving its formation in spatial-temporal domain. At last, we implement *EV-Tach* on Raspberry Pi and the evaluation results show that the downsampling process preserves the high measurement accuracy while saving the computation speed and energy consumption by approximately 8 times and 30 times in average.

Index Terms—mobile sensing, rotational speed measurement, dynamic vision sensing

1 Introduction

Machines and devices with rotary components are pervasive in our daily life and play significant roles in various industrial fields, such as energy, aviation, automobile and home appliance. In manufacturing, rotational speed is one of the key indicators to reflect the current working state of the machines, therefore, there is a huge demand for measuring the rotational speed with an accurate and convenient tool. In the field of appliance repairing, repairmen normally use tachometer (an instrument to measure the rotational speed) to measure the rotational speed of the electrical motors of the appliance, such as the condensing unit of air-conditioners and washing machines, to infer possible faults from the irregular rotational speed. In the automotive maintenance, the checking of rotational speed of the wheels has become a standard item in the annual vehicle inspection manuals [1]. Real-time measurement of rotational speed is also useful for predicting the actions of flying drones because they change their flying direction and speed by adjusting the rotational speed of one or multiple propellers; the actions should be always after the change of rotational speed due to inertial effect. At last, rotational speed calibration of some devices and equipment, such as drones, watermeters and car engines, is another typical scenario in which the rotational speed needs to be calibrated precisely to ensure the devices are functioning as expected.

A number of different measurement approaches have been proposed to obtain the rotational speed of different targets under different circumstances. They are different from the requirement of physical contact (contact or non-contact) or sensing modalities (electromagnetic, laser or vision). Mechanical tachometers [2] are a type of traditional devices to measure rotational speed of large machines via physical connection to the shaft of the targets. Electrostatic [3], hall-effect [4] and optical encoder tachometers [5] are non-contact but they must be placed in proximity to measure the rotation of the extra hardware mounted on the shafts of the targets. All the above approaches are invasive as the

*Corresponding author

- *G. Zhao, Y. Shen, N. Chen and L. Liu are with School of Software, Shandong University, Jinan, 250100, China. E-mail:{guangrong.zhao, yiran.shen, ningchen, l.liu}@sdu.edu.cn*
- *P. Hu is with School of Computer Science and Technology, Shandong University, Qingdao, 266237, China. E-mail:phu@sdu.edu.cn*
- *H. Wen is with Department of Computer Science, University of Warwick, UK. E-mail:hongkai.wen@dcs.warwick.ac.uk*

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2023.3335221

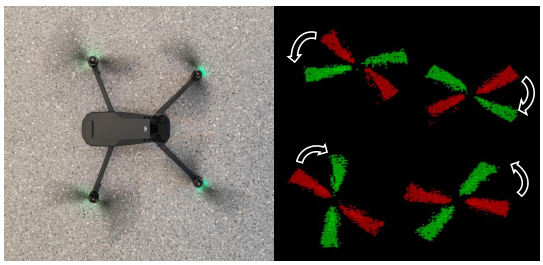


Fig. 1. Shooting high-speed rotating wings of drone with a conventional RGB camera (left) and an event camera(right) respectively.

physical contact or extra hardware may place significant influence on the natural rotating. Laser tachometers [6], [7] make a step forward to more accurate and convenient measurement on rotational speed. The laser tachometer enables highly accurate (the error rate is below $0.4‰$) and low invasive measurement and can be used in reasonable working distance. Therefore, the laser tachometer has become the mainstream golden standard instrument for rotational speed measurement. However, the requirement of retro-reflective labels on the targets still limits the application of laser tachometers under some circumstances as attaching labels may not be convenient or even impossible for some devices. Then, most importantly, although the laser tachometers are built as portable devices, it is difficult for the users to point to the extremely small label on the rotating target with the laser tachometer in hands especially when the object is vibrating or in long measurement distance and the accuracy in handheld mode degrades significantly according to our evaluation in Section 4.2. Vision-based approaches [8], [9], [10], [11], [12], [13], [14] require no extra hardware on the rotating targets and can further extend the working distance with zoom lens. They also show strong environmental adaptability and robustness. However, for vision-based approaches with CCD/CMOS [8], [10], [11], [12], [13], the range of rotational speed to be measured is limited by the frame rate, which is normally between 30-50 fps (frames per second) and the accuracy is not good enough (error rate is over $10‰$) for high-precision measurement. As shown on the left of Figure. 1, high-speed rotation of drone's propellers will place significant motion blur in the recorded video of normal RGB cameras and causes failure of measurement. High-speed cameras with frame rate of few hundreds and even thousands can cover larger range of rotational speed. However, high-speed cameras are highly resource-consuming which is too prohibitive for processing on portable devices with embedded CPUs.

Introducing Event Camera for Rotational Speed Estimation: to solve the above issues, we adopt the event cameras or dynamic vision sensors (DVS), a new sensing modality [15], [16], to capture high speed rotation without motion blur. DVS is bio-inspired and its pixels work independently to detect the change of intensity. Unlike the frame-based RGB cameras, the output of DVS consists of nonstructural and discrete event points in spatial-temporal domain and is termed as event streams [17], [18]. In recent years, DVS has been widely applied in a variety of computer vision tasks, such as super resolution, image deblurring, gesture recognition, etc. Unlike traditional RGB cameras, DVS do not produce synchronous video frames at fixed rate, but asynchronous event streams [17], [18]. Specifically, pixels of the DVS work independently, to detect the change of the intensity of the scene as,

$$|log\ I(x, y, t_{now}) - log\ I(x, y, t_{previous})| < C \qquad (1)$$

where $I(x, y, t)$ is the intensity value of pixel (x, y) at time t. When the change of intensity at the pixel is over the threshold C, an event will be released immediately. An event stream is a collection of events overtime and is represented as a stream of quadruplet $\{x, y, t, p\}$. When the event corresponds to a positive change, the polarity p is $+1$ otherwise it is -1. Compared with traditional RGB cameras, DVS possess a number of unique characteristics. As an event is launched as soon as a change is detected without global synchronization, the event streams are high in temporal resolution and low in response latency (in the order of microseconds) [19], [20], [21]. DVS save sensing energy and bandwidth as they produce events only when changes are detected. The high dynamic range (140 dB vs. 60 dB of traditional RGB cameras) enables them work greatly under challenging lighting conditions. These characteristics make DVS have great potential for high-speed motion capture and working on resource-constrained devices. Especially, DVS has microsecond temporal resolution [20], [21], which makes it more appealing than frame-based RGB cameras for capturing high-speed motion.

Figure. 1 presents the rotating propellers of a drone landing on the floor. The left figure is an RGB frame from a video recording in a frame rate of 60 fps by an off-the-shelf smartphone camera and the right figure shows accumulative outputs of two 2 ms *event streams* from DVS in red and green respectively. From the figures we can observe the rotating propellers are severely blurred in the RGB frame while the shape of the propellers are well preserved and the rotation between the two slices can be easily identified in DVS outputs.

Motivation and Challenges: though DVS is able to capture the high speed motion naturally due to its high temporal resolution, how to infer the rotational speed from the non-structural and noisy spatial-temporal event streams consisting of thousands sparsely distributed events are not straightforward as most of the traditional image processing techniques cannot be applied. There is a first attempt for measuring rotational speed with DVS in [22], its main purpose is to estimate visual optical flow. However, its setting is far from the practical usage: to estimate the rotational speed of a white plate, it calculates the optical flow of a straight line (drawn from center to edge) to infer the rotational speed of a rotating plate. Therefore, it cannot be used to estimate most of the rotating objects in real-world as most of them are rather than a simple straight line. Moreover, as reported in [22], as the noisy nature of event streams, the accuracy of optical flow cannot meet the requirement of precise rotational speed measurement: it can only provide reasonable results when rotational speed is less than 500rpm though the setting of experiment in [22] has been far simpler than real-world scenarios. To fully explore the potential of DVS on rotational speed measurement task under real-world scenarios, a number of challenges need to be solved. First, multiple rotating targets can be recorded

simultaneously and how to separate the event streams generated by different targets correctly is fundamental for estimating the rotational speed of each target accurately. Then, to be used handheld under practical scenarios, the movement of user's hand and host devices will cause events irrelevant to the rotation. The noisy event streams will bring difficulty on obtaining accurate measurement for handheld tachometers. Finally, processing the event streams with tens of thousands events poses a huge computation burden for embedded CPU and prohibitive for resource-constrained handheld platforms. To deal with the challenges above, we propose *EV-Tach*, an event-based rotational speed estimation system facilitating a number of bespoke modules for event streams processing. The contributions of this paper can be summarized as follows:

- To the best of our knowledge, this is the first sophisticated approach and comprehensive work on rotary motion sensing with DVS and demonstrate its privilege in capturing high speed rotation under practical and challenging scenarios as a handheld device.
- An event-based rotational speed estimation system, *EV-Tach*, is proposed via efficient high-fidelity rotary motion sensing with DVS to estimate high-speed rotating targets. *EV-Tach* consists of a series of bespoke modules for accurate and robust rotational speed estimation, including heatmap-based stream-centroids initialization to separate multiple rotating objects, outlier removal to deal with noisy events caused by subtle movement and a coarse-to-fine strategy to improve the alignment of consecutive slices of event streams.
- We conduct extensive evaluations on the accuracy of *EV-Tach* and robustness to different rotational speeds, working distances, view angles, ratios of occlusion and vibration. According to the results, *EV-Tach* achieves comparable accuracy to laser tachometer for fixed setting. The robustness of *EV-Tach* to subtle movement shows it can be used in **handheld** mode and measure the unstable rotating targets where the laser tachometer does not work.
- The experiments under practical scenarios are also conducted to measure the rotational speeds of the four propellers of drone, condensing unit of air-conditioner deployed outdoor and car wheels under maintenance. The results show that *EV-Tach* can provide accurate and robust measurement on multiple rotating targets simultaneously or under challenging conditions.
- At last, *EV-Tach* is implemented on Raspberry Pi and VoxelGrid filtering is applied to significantly reduce the number of events to be processed so that the processing time and energy consumption is saved by up to 10.5 and 40 times respectively.

The rest of the paper is organized as follows. Section 2 provides related work on rotary motion measurement and DVS. Then we overview the system design in Section 3 including the algorithms of event stream processing and rotational speed estimation. Extensive evaluations are conducted and results are presented in Section 4. Finally, we discuss the advantages and limitations of *EV-Tach* in Section 5, and conclude the whole paper in Section 6.

2 Related Work

In this section, we will review the work related to rotary motion sensing and dynamic vision sensor.

Traditional Rotary Motion Sensing. Mechanical tachometers [2] are physically connected to and rotate with the shaft of the target to measure the rotational speed. Its error rate was about $5‰$. However, the physical contact constrains the working distance and causes inaccurate measurement due to the mass and friction of the mechanical tachometers. Electrostatic [3] and hall-effect [4] sensors detect the change of electromagnetic field caused by shaft-bearing fixed on the target and the frequency of the change was estimated as the rotational speed. Optical encoder tachometers [5] relied on a photoelectric sensor to detect light through the disc of encoder placed between the LED light source and photoelectric sensor. An encoder is a disc mounted on the shaft of the rotating target with opaque and transparent segments so that rotational speed can be estimated based on the pattern of the light. The electrostatic and optical encoder tachometers can be regarded as non-contact and their error rate were around $0.2‰$ to $5‰$, but they must be placed in proximity to measure the rotation of extra hardware attached on the shaft of target. RFTacho [23] proposed an RF-based approach for contactless rotational speed estimation and was able to estimate multiple targets simultaneously. However, compared with *EV-Tach*, the robustness of RF-Tacho to the subtle movement was not addressed, therefore, could not be used handheld; moreover, its error rate was about $5‰$, which was at least 10 times worse than our approach. Laser tachometer [6], [7] measures the rotational speed by detecting the small and lightweight retro-reflective labels attached on the surface of the target, its error rate was about $0.5‰$. However, the use of labels may cause inconvenience during measurement and the laser tachometer cannot provide accurate measurement when used as a handheld device.

Vision-based Rotational Speed Estimation. There have been a number of non-contact approaches being proposed for rotational speed estimation. Wang at el. [8] calculated the structural similarity and two-dimensional correlation between the consecutive frames, and then the similarity-related parameters were used to reconstruct a continuous and periodic signal of time-series. Fast Fourier transformation was applied to calculate the period of the signal which was used to infer the average speed of rotation. Other approach [12] was also proposed to utilize the periodical change of similarity between frames and the difference was Chirp-Z transform and the parabolic interpolation based auto-correlation were applied to estimate the period in other domain. To improve the accuracy and range of measurement, Natali at el. [11] obtained the coefficients sequence of correlation between the reference and each of the following frames. Then the rotational speed could be calculated through the short-time Fourier transform (STFT), which enabled more accurate measurement of the rotational speed of non-stationary and disturbing systems. Instead of directly calculating the complete period of the rotation, there are some works to obtain the rotational speed by calculating the instantaneous angular speed (IAS). Zhu at el. [10] extracted two adjacent frames from the video of

a rotating objects, then the Hough transform was applied to detect straight lines, and the angular changes of these lines could be calculated. Since the interval time of the two frames was known, the angular velocity of the object could be easily obtained. However, these methods above were limited by the frame rate of the conventional RGB cameras and can only accommodate the rotational speed less than 900rpm and the accuracy is far from our approach: the error rate is over 10‰ which is about 25 times worse than our proposed *EV-Tach*. In order to obtain a larger measurement range of rotational speed, some researchers used high-speed cameras [14], [9] to measure the instantaneous angular speed of rotating object. However, the cost of the high-speed cameras is prohibitive for embedded platforms and both of these methods required special-style markers attached on the rotating targets.

Dynamic Vision Sensors. Processing of event streams is a new topic to study. To facilitate existing methods, event streams were converted to other familiar formats, including images, graphs and 3D pointclouds. Image-like representations of event streams were introduced by accumulating the event points for each pixel overtime and corresponding methods were proposed for gesture recognition [24], gait recognition [25] and estimating optical flow of event streams [26]. However, the image-like representation ignored the temporal information of event stream. Graph-based representations were proposed to preserve the spatial-temporal information of event streams. 2D-Graphs [27] or 3D-Graphs [28] were built by selecting and connecting event points via nearest neighbor search, then graph-based convolutions were applied to extract higher-level information. The spatial-temporal event streams could also be processed as 3D pointclouds then the PointNet [29] and PointNet++ [30] were applied, e.g., for gesture recognition [31].

Most relevant work. In [19], the authors pointed out that DVS has the potential to capture high-speed rotational motion, but they did not give a specific rotational speed measurement method, nor evaluated its performance under different challenging practical settings. Gallego at el. [32] proposed an approach to estimate the rotation of event camera by processing the event streams. They applied contrast-maximizing edge alignment algorithm to estimate the angular velocity which was relevant to our work. However, it measured the rotation of the event camera itself, which was fundamentally different from the goal of our work, meanwhile it produced significantly lower measurement accuracy (the error rate is around 20‰) and sensing range (less than 167rpm) than our approach. In [22], the authors proposed a method to calculate the optical flow of moving object in event stream and showed its application on estimating rotational speed of a plate. Though it also utilized the DVS as sensing modality for rotational speed estimation, the design of algorithm was not sophisticated enough to deal with the non-structural and noisy event stream to obtain accurate measurement on high-speed rotation nor showed any evidence for working under practical setting: the method in the paper only estimated the rotational speed of a white plate with a black straight line and the method could only provide reasonable measurement for rotational speed less than 500rpm (the accuracy was also far from our method even within the sensing range, around 20‰). Moreover, the approach above only considers single rotating object while it becomes challenging when multiple targets are recorded simultaneously, e.g, a drone with multiple blades. To adopt DVS for rotational speed estimation in practical scenarios, a number of challenges still remain unexplored: how to separate complex event streams caused by multiple objects, how to obtain accurate estimation from noisy event streams caused by vibration from subtle movement of handheld and vibrating object, and how to process thousands of events efficiently on resource-constrained devices. In this paper, *EV-Tach* is proposed to bridge the gap above.

3 System Design

In this paper, we propose an event-based rotational speed measurement system, *EV-Tach*. The overall system design of *EV-Tach* is presented in Fig. 2 and can be vastly divided into three components, i.e., rotating object extraction, event stream downsampling and ICP-based registration.

Rotating objects extraction: it starts with recording a short-period of event stream. To accommodate multiple rotating objects scenarios, the event stream is separated into multiple clusters corresponding to different rotating targets through a series of event stream processing components. Specifically, the centroid-initialization component initializes the locations of centroids for clusters based on the heatmap of events to avoid the clustering algorithm falls into local optimal. Then the K-means-based approach is used to cluster thousands of events and automatically determining the number of rotating targets without the prior knowledge. At last, to suppress the interference of events not belonging to the rotating object, the outlier events are removed by comparing the distance of the events to the cluster centroid to find out noisy events caused by subtle movement.

Event stream downsampling: when the event stream is separated into multiple clusters according to different rotating objects. The event stream downsampling component is triggered to save the computation and energy consumption of the whole system. It takes a cluster with thousands of events as input and applies VoxelGrid filtering [33] to reduce the number of events by merging multiple events within the same voxelgrid. As the VoxelGrid filtering can preserve the spatial-temporal structure of the cluster, the accuracy will not experience noticeable drop.

ICP-based registration:At last, the downsampled clusters of events are fed to the ICP-based registration to obtain the rotational speed of each object. The event stream in spatial-temporal domain is similar to 3D pointcloud, we propose an ICP-based registration method to estimate the transformation (i.e., how to align two slices of event streams) between the two slices of event streams to obtain the angle of rotation. Then the rotational speed can be calculated by dividing the angle of rotation with the time difference. A two stage coarse-to-fine alignment strategy is also applied to improve the accuracy in which the initial alignment provides important hints for the second alignment to refine the estimation.

3.1 Event Stream Processing

In this section, we will describe the event stream processing algorithms in details. The algorithms aim to extract high

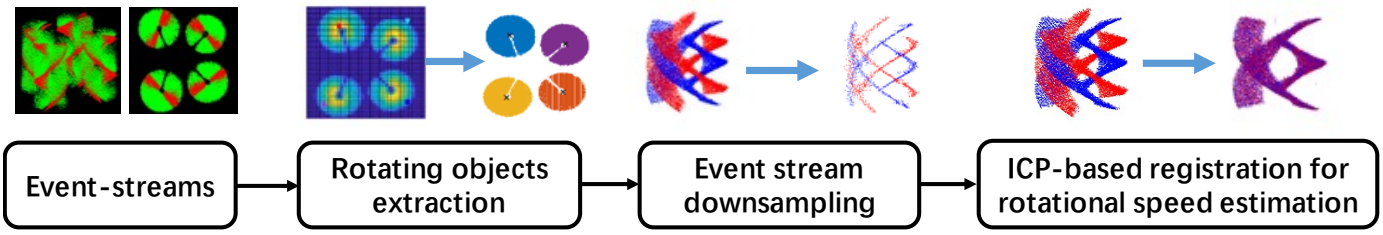


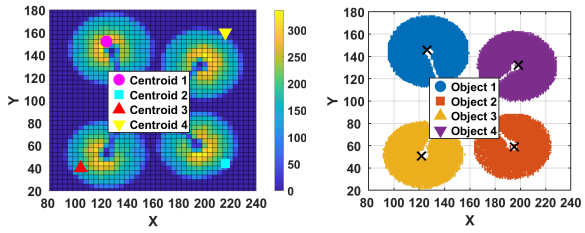
Fig. 2. System overview of *EV-Tach*

quality and low-dimensional event stream with single rotating target for the rotational speed estimation in next section.

3.1.1 Rotating Objects Extraction

One of the important merits of *EV-Tach* over electromagnetic and laser tachometer is its capability on sensing multiple rotating targets simultaneously. For example, in drones manufacturing, the four independent electric-motors need to be calibrated. If electromagnetics and laser tachometers are exploited, the number of sensors needs to be the same as the electric-motors. The manpower and material consumption during the settings of the measurement will be a challenge. On the contrary, with *EV-Tach*, only one DVS is needed without changing the settings of the measurement. To separate different rotating targets, we propose a K-means-based rotating objects extraction algorithm to isolate the events belonging to different rotating targets for further processing.

Heatmap-based Stream-centroids Initialization: K-means is widely used clustering algorithm in euclidean space. It iteratively merges the points to the nearest clusters and update the centroids accordingly until it converges. For rotary motion sensing, the resultant centroids are the locations of the rotation axes of all rotating targets. The computational complexity of K-means is low and can run in-situ on resource-constrained platforms. However, it suffers from instability and sensitivity to the initial location of centroids [34]: the poor choice of initial centroids may lead the algorithm fall into local optimal and result in incorrect clusters.



(a) initial centroids on heatmap (b) clustering result

Fig. 3. The initial locations of the centroids on the heatmap and the clustering results of four-propellers drone.

Considering the characteristics of event streams produced by rotating targets, we propose a lightweight stream-centroids initialization method based on the heatmap of accumulated events to enable reliable rotating objects extraction. After caching a fixed-length of event stream, e.g., 150ms, the number of events on each pixel are accumulated into a heatmap. The size of the grid in our setting is 4×4 pixels. Figure 3(a) presents the locations of the initial centroids on the heatmap of an event stream collected from a four-motor drone. From the heatmap, we can observe, more events are generated near the center of the rotating target. We find the grid with the highest value (denoted as h) in the heatmap as the centroid of the first cluster i.e., c_1. Then the remaining centroids are chosen from the grids whose values larger than ϵh ($\epsilon = 0.3$ in our evaluations and experiments). Specifically, the second initial centroid c_2 is the most distant grid from the set $S = \{c_1\}$, the $(i+1)$-th initial centroid c_{i+1} is chosen as the most distant grid from the set $S = \{c_1, c_2, ..., c_i\}$ and so on, where the distance between a grid g and a set S is defined as:

$$\mathcal{D}(g, S) = \min_{\mathrm{c_i} \in \mathrm{S}} d\left(g, \mathrm{c_i}\right), \tag{2}$$

$d\left(g, \mathrm{c_i}\right)$ is the euclidean distance between grid g and i-th initial centroid c_i from set S.

Figure 3(a) shows an example of the four initial centroids chosen by our method. The principle behind this strategy is to maximize the inter-cluster distance to avoid the local optimal of K-means algorithm. Figure 3(b) demonstrates the clustering results of the event stream generated by a four-propeller drone, where the four rotating objects are separated correctly.

Spatial Clustering on Event Streams: After the centroids initialization, the event stream should be segmented into multiple clusters corresponding to each individual rotating target. In each iteration, K-means-based clustering is applied to associate each event e_i to one of the clusters with the nearest centroid in euclidean space and then update the centroid of each cluster by:

$$\hat{c}_i = \frac{1}{|Q_i|} \sum_{e_j \in Q_i} l(e_j) \tag{3}$$

where $\hat{c}_i$ is the updated centroid location, $|Q_i|$ is the total number of events and $l(e_j)$ is the location of the event e_j. The events clustering and centroids updating procedures are executed alternatively until the location of the centroids remain (almost) the same, which indicates a stable clustering result has been achieved.

Choice of k: As the number of rotating objects in an event stream can be various, the choice of k is of importance for the K-means clustering. However, it is inconvenient to manually determine the number of rotating targets. To address this challenge, we apply Davies-Bouldin Index (DBI) [35] to evaluate the quality of the clustering. Then k can be determined without the prior knowledge. Specifically, we assume

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2023.3335221

the candidate values of k are $\{k_1, k_2, k_3.., k_i, ...\}$. When $k = k_i$, a collection of k_i clusters $\mathcal{Q}_{k_i} = \{Q_1, Q_2, ..., Q_i, ..., Q_{k_i}\}$ are obtained from K-means clustering. Then the DBIs of the clustering results with different value of k are calculated and the minimal DBI is desired to maximize the inter-cluster distance and minimize the intra-cluster distance. To calculate DBI when $k = k_i$, we need first estimate the dispersion of each cluster and the separation between any of the two clusters. The dispersion of cluster Q_i when $k = k_i$ is

$$Disp(Q_i) = \frac{1}{|Q_i|} \sum_{e_j \in Q_i} d\left(e_j, c_i\right) \tag{4}$$

where $d(e^j, c_i)$ is the euclidean distance between any event and the centroid of the cluster. Then the separation between cluster Q_i and Q_j in $\mathcal{Q}$ is:

$$Sep(Q_i, Q_j) = d\left(c_i, c_j\right). \tag{5}$$

With the dispersion and separation, we can obtain the similarity between the two clusters Q_i and cluster Q_j,

$$Sim(Q_i, Q_j) = \frac{Disp(Q_i) + Disp(Q_j)}{Sep(Q_i, Q_j)}. \tag{6}$$

Then the similarity between Q_i and the whole collection $\mathcal{Q}$ is defined as maximum similarity between Q_i and any other cluster from the collection:

$$SIM(Q_i, \mathcal{Q}) = \max_{j=1..k_i, j \neq i} Sim(Q_i, Q_j) \tag{7}$$

Then DBI of the clustering result when $k = k_i$ can be expressed as,

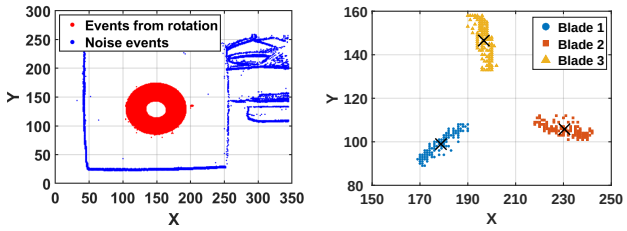
$$DBI_{k_i} = \frac{1}{k_i} \sum_{i=1}^{k_i} SIM(Q_i, \mathcal{Q}). \tag{8}$$

Finally, the value of k bringing the smallest DBI is chosen and the corresponding clustering extracts multiple rotating targets from the event stream.

Post-processing for Outliers Removal: After the rotating objects extraction, a post-processing module is triggered to remove the outliers of each event stream caused by the movement irrelevant to the rotating objects. Except for the rotating targets, the subtle movement of users (in handheld measurement) and vibration of the host devices will also cause noticeable events in DVS and these events are regarded as outliers. Figure 4(a) presents the accumulated event stream in pixel domain recorded by a user holding an event-camera in front of a rotating target (the ring in red). Due to the subtle movement of user's hand, the outliers of the hosting device and edges in background (in blue) are also detected in DVS. As the rotating target causes significantly larger density of events than the subtle movement and the outliers are normally far from the centroid. Most of the valid events should concentrate around the center of rotation. To remove the outliers, we first estimate the median distance of the events to the centroid,

$$D_m = median\{d\left(e_1, c_i\right), d\left(e_2, c_i\right), ..., d\left(e_j, c_i\right), ...\}, e^j \in Q_i \tag{9}$$

Then we set distance over three times of the D_m as the threshold and the events with distance over the threshold are marked as outliers.



(a) Outliers removal (b) Blades detection

Fig. 4. Example of outliers removal (a) and number of blade detection (b).

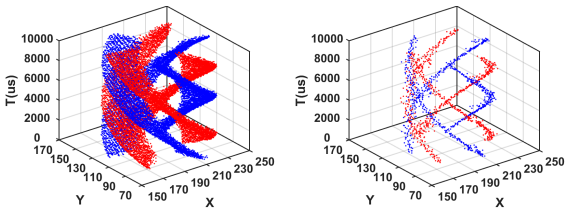
Angle of Rotational Symmetry: For each identified rotating object, we need to track the amount of rotation precisely within appropriate period to avoid events generated by different blades entangled spatially causing ambiguity in the following steps. In many real-world applications, many rotating objects are of centrosymmetric shapes, such as propellers of drone, fans of condensing unit and wheels of automobile etc. To estimate rotational motion of these objects, we identify certain features on the rotating objects, e.g. the blades of the propellers/fans, spokes of the wheels, and track the motion of those features (i.e. angle of rotation) as a proxy for the rotational motion of the target objects. Note that for objects without those intuitive features, such as a plain rotating disk, in practice we can easily annotate them, e.g. with stickers or patterns, to create such trackable features. In addition, without loss of generality in this work we assume the rotating objects are rigid bodies, i.e., there is no significant deformation during their motion.

In this context, to accurately track the rotational motion of these features, e.g. the blades of the propellers, an important parameter to determine is the *angle of rotational symmetry*, i.e. the smallest angle for which a feature can be rotated to coincide with itself or the other features. In our case, this is used to determine the appropriate length of event streams for the later ICP-based registration to avoid ambiguity. For example, Figure 4(b) shows an event stream caused by a rotating object with three separated blades. K-means ++ [36] with Davies-Bouldin Index (DBI) evaluation is applied to separate different blades in event stream with only a small number of events (e.g., 300). Because when the number of events become large, the events generated by different blades will be entangled spatially due to the rotation. Then the angle of symmetry can be determined by the number of repeated parts, e.g., blades.

3.1.2 Event Stream Downsampling

EV-Tach is designed as a rotational speed estimation system on embedded platforms so that it can be used handheld. However, processing the event streams with tens of thousands events is a huge burden for embedded CPU and prohibitive for resource-constrained platforms, e.g., Raspberry Pi. To reduce the number of events to be processed, we downsample the event streams through VoxelGrid filtering [33]. The VoxelGrid filtering starts with dividing the whole spatial-temporal space of event stream into 3D grids, termed as voxels, with size (V_X, V_Y, V_T) in each dimension. V_X and V_Y are sizes of the voxel in pixels along the X and Y axes respectively while V_T is the size of the voxel in

milliseconds along the T axis. *EV-Tach* applies cubic voxels so that the sizes of the three dimensions are equal and denoted as V_s. In each voxel, the events are merged into one representative point at the spatial-temporal centroid of them so that the following ICP-based registration can work on the representative points directly. Figure 5 presents an example of VoxelGrid filtering on downsampling an event stream. Though the number of events are significantly reduced, the traces of the rotation are still clear and complete.



(a) Before downsampling (b) After downsampling

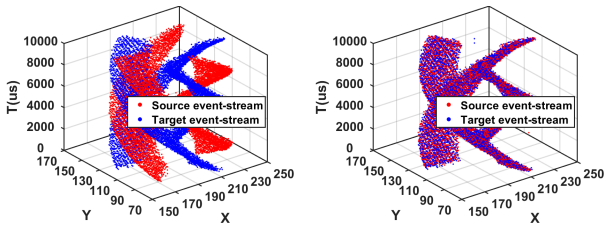
Fig. 5. VoxelGrid filtering for event stream downsampling.

3.2 Rotational Speed Estimation

After the event streams corresponding to different rotating targets are extracted, an Iterative Closest Point (ICP)-based registration approach is proposed to estimate the rotational speed according to the transformation of event stream overtime. ICP is a commonly used technique in computer vision and computer graphics [37], [38], [39] for aligning two or more pointclouds in 3D space, and has wide applications in various fields such as object recognition [40] robot navigation [41] and 3D registration [42]. It aims to find the optimal transformation (rotation and translation) from one pointcloud to the other pointcloud by minimizing the ℓ_2-norm error (mean square error, MSE) between the points from the two pointclouds. The typical steps of the ICP algorithm [43] starts with a nearest neighbor search to find the correspondence between points from different pointclouds. Then the euclidean distance between each pair of points are calculated to derive the MSE of the alignment. The rotation and translation is obtained by solving the singular value decomposition (SVD) [44], [45] problem and applied to align the two pointclouds. The steps are repeated until a desirable MSE is achieved. To accommodate larger range of rotational speed, the ICP-based alignment approach applies a two-stage **coarse-to-fine strategy**: initial alignment provides a coarse estimation as a feedback to the refinement stage to obtain accurate rotational speed.

3.2.1 Initial Estimation

In *EV-Tach*, the rotational speed is calculated by estimating the angle that a propeller has rotated around its axis in a specific time. For example, Figure 6(a) presents two consecutive 10ms-slices of event stream generated by a rotating propeller with three blades. The two slices share 7ms overlap and the step between the two slices are 3ms. As the two slices of event stream are generated by the same propeller, the angle of rotation between the two slices can be obtained through aligning the two slices of event stream. In this paper, we propose an even-stream registration algorithm based on iterative closest point (ICP).



(a) Before registration (b) After registration

Fig. 6. Event stream Registration

Rotations in Spatial-temporal Domain: Before touching the details of the ICP-based registration algorithm, we first briefly introduce the rotation matrix, which is the core output we need from the event stream registration to calculate the rotational speed. The output of ICP normally consists of a translation matrix and a rotation matrix. Rotation matrix $\mathbf{R}$ describes the rotation in 3D space and can be decomposed into *roll*, *pitch* and *yaw*, the three independent rotations around each axis according to Euler's rotation theorem [46]. As shown in Figure 7, the spatial-temporal event stream can be regarded as in three-dimension (X, Y, T). The overall rotation matrix R can be decomposed into three independent rotation matrices in the spatial-temporal domain:

$$\mathbf{R} = \mathbf{R}_X(\alpha)\mathbf{R}_Y(\beta)\mathbf{R}_T(\gamma) \tag{10}$$

where R_X, R_Y and R_T are the rotation matrices to the three axes and α, β and γ are the angles in *roll*, *pitch* and *yaw* respectively. From Figure 7, we can easily identify, the *Yaw* rotation around T-axis is directly related to the rotation of the propeller. Therefore, we only need to focus on the rotation matrix R_T, which can be expressed as,

$$\mathbf{R}_T(\gamma) = \begin{bmatrix} \cos(\gamma) & \sin(\gamma) & 0 \\ -\sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{11}$$

With ICP-based registration, we can obtain the *yaw* rotation angle γ from R_T.

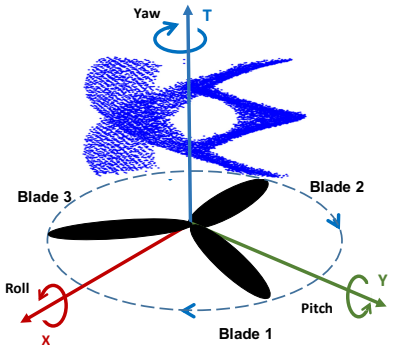


Fig. 7. Demonstration of rotation in spatial-temporal space

ICP-based Rotational Speed Estimation: The ICP-based event stream registration works on two consecutive slices of event stream $\mathbf{P}$ and $\mathbf{Q}$ with length t_l, overlap t_p and step length t_f ($t_l = t_p + t_f$), where $\mathbf{P}$ is termed as source event stream and $\mathbf{Q}$ is the target event stream. In this paper, we view source and target event streams as event pointclouds

in the spatial-temporal domain. The ICP-based registration aligns the two slices of event streams by rotating the source event stream to match the target event stream in spatial-temporal domain such that the rotating angle between the two slices can be obtained. The objective of our ICP-based approach is to optimize the ℓ_2-norm error E between the rotated source event stream and target event stream by solving:

$$\min E(R, T_r) = \min \frac{1}{N} \sum_{i=1}^{N} \|q_i - (R \cdot p_i + T_r)\|^2, \tag{12}$$

where R and T_r are the optimal rotation and translation, respectively; q_i and p_i are samples in Q and P, respectively, and N denotes the number of event points in $\mathbf{P}$. We solve the optimal R and T_r by a singular value decomposition (SVD)-based method [44]. It starts with the nearest neighbor search, i.e., for each event in $\mathbf{P}$, the closest event from $\mathbf{Q}$ is found in spatial-temporal domain. After the nearest neighbor search, we can obtain a subset $\mathbf{Q}'$ consisting events from $\mathbf{Q}$ which are nearest neighbors of events in $\mathbf{P}$. Then the co-variance between the $\mathbf{P}$ and $\mathbf{Q}'$ is,

$$cov = \sum_{i=1}^{N} (p_i - \bar{p}) (q_i' - \bar{q}')^T \tag{13}$$

where p_i and q_i' are the spatial-temporal positions of the i_{th} event from $\mathbf{P}$ and $\mathbf{Q}'$ respectively. $\bar{p}$ and $\bar{q}'$ are the spatial-temporal positions of the centroids. Then SVD is applied to factorizing the co-variance matrix, i.e.,

$$cov = U \Sigma V^T. \tag{14}$$

The rotation matrix is $\mathbf{R} = VU^T$ (according to [44]) and translation matrix is $\mathbf{T}_r = \bar{q}' - \mathbf{R}\bar{p}$. Along with Eq (10) and Eq (11), the yaw rotation angle γ can be estimated and the source event stream is transformed according to the rotation and translation matrices. Then the operations above are repeated and γ from each iteration is accumulated:

$$\gamma_{acc} = \gamma_{acc} + \gamma \tag{15}$$

The iteration terminates when yaw rotation diminishes, i.e., $\gamma / \gamma_{acc} < 0.001$.

It is worth noting that, DVS is noisy and non-structural, the source and target event streams normally cannot perfectly aligned. To reduce the influence of misalignment on the estimation of rotation, we apply bi-directional registration by simply switching the source and target event streams. The average yaw rotation $\bar{\gamma}$ is adopted to calculate the rotational speed (in rpm) from initial alignment:

$$r_{init} = \frac{\bar{\gamma}}{2\pi t_s} \times 60. \tag{16}$$

3.2.2 Estimation Refinement

For the initial alignment stage, there is no prior knowledge on how fast the rotational speed is. Therefore, to accommodate high rotational speed and avoid events from different blades being entangled spatially, we choose a small step length, i.e., $t_s = 1ms$, in case the event streams from different blades are overlapped and lead to ambiguity in rotation estimation. However, when the rotational speed is slow, e.g., less than 1000rpm, only very few events are generated by the rotating object within the super short period. Considering the noisy nature of DVS, the estimation from initial alignment can be unreliable. To improve the accuracy of estimation, we propose a simple but effective refinement approach based on the coarse result from initial alignment. According to the rotational speed r_{init} from initial alignment, we can extend t_s to include more events meanwhile avoid the ambiguity caused by central symmetry: t_s cannot lead to rotation over the angle of symmetry mentioned above, otherwise, the ICP-based registration will align two different parts of the object together and causes incorrect estimation on rotation. According to above constraints, the new step length can be inferred as,

$$t_s' = \eta \frac{60}{2 r_{init}} \cdot \frac{\theta_c}{2\pi} \tag{17}$$

where θ_c is angle of central symmetry determined above, η is a scale factor (<1) to accommodate the inaccuracy of initial alignment. Then the new step length t_s' is adopted to run the ICP-based event stream registration again to obtain a refined estimation on rotational speed.

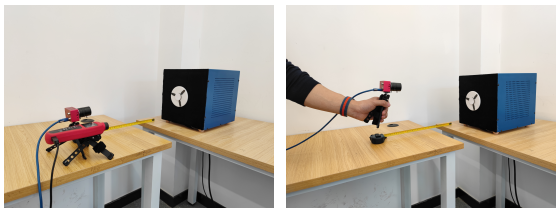
4 EVALUATION

In this section, we evaluate our proposed *EV-Tach* on datasets collected from monitoring the rotation of a customized device and compare it with laser tachometer on accuracy, robustness, convenience, etc. Then a number of experiments on practical devices including drones, air-conditioner and automobile are conducted under indoor and outdoor environments. At last, by implementing it on Raspberry Pi, the trade-off between accuracy and efficiency, and the resources consumption will be demonstrated.

4.1 Evaluation Setup

Data Collection: As shown in Figure 8, during data collection, an event-camera is used to collect raw event streams of a rotating target on the customized device and laser tachometer is also deployed as benchmark. The customized device is equipped with a servo motor whose rotational speed can be precisely controlled through an interface on a laptop and the highest rotational speed is 6000rpm. A white plate is connected to the motor shaft as the rotating target and "propellers" can be printed and attached on the plate as requirement. The event-camera is DAVIS346 [20] whose spatial resolution is 346×260 and temporal resolution is $20\mu s$. DAVIS346 comes with a vari-focal CS-mount lens which can be used to extend the measurement distance. The laser tachometer (UNI-T UT372) provides high precision measurement with relative error of $\pm 0.4‰$ and the results can be easily streamed to a computer via cable. Moreover, the measurement distance ranges from $20 - 50cm$. The datasets are collected by changing the rotational speed of the servo motor, distance to the target, different number of blades, and etc.

Implementation Details: As shown in Figure 9, we implement *EV-Tach* on an embedded platform, Raspberry Pi 4B [47], as prototype to evaluate the resource consumption of *EV-Tach*. Raspberry Pi 4B features a Broadcom BCM2711 embedded CPU with $4\times$ 1.5GHz Cortex-A72 processors and 4GB LPDDR4-3200 SRRAM and the operating system is Ubuntu 20.04. The DAVIS346 is connected to Raspberry Pi



(a) DAVIS346 and UT372 (b) DAVIS346 handheld

Fig. 8. Experiment setup for data collection

via USB 3.0 so that the *EV-Tach* running on the embedded CPU can control the event-camera to conduct dynamic vision sensing and compute the rotational speed accordingly. To fully exploit the computational capability of multi-core processors of Raspberry Pi, multi-threads implementation is adopted to speed up the processing of *EV-Tach*. For examples, when downsampling the event streams by VoxelGrid filtering, the source and target event streams can be processed in parallel. During nearest neighbor search, which is most time-consuming component of ICP-based registration, can also be easily paralleled in multiple threads.

Fig. 9. Prototype of *EV-Tach* on Raspberry Pi 4B

Evaluation Metrics: The evaluations in this section can be vastly divided into two aspects: the first is to compare the accuracy of *EV-Tach* to laser tachometer and the second is to evaluate its system overhead on embedded system. Relative mean absolute error (RMAE) is adopted to present the accuracy of measurement and defined as,

$$RMAE = \frac{1}{M}\sum_{i=1}^{M}\frac{|r_i - r_{gt}|}{r_{gt}} \tag{18}$$

where M is the number of tests and M = 30 in the following evaluation; r_i is the i_{th} measured rotational speed and r_{gt} is the ground-truth. Low RMAE means high measurement accuracy. Then the time and energy consumption of *EV-Tach* on Raspberry Pi is profiled by estimating the current and voltage of the prototype.

Competing Methods: In the evaluations on accuracy, we consider four different measurement methods to compare the accuracy of *EV-Tach* to the state-of-the-art laser tachometer, which are:

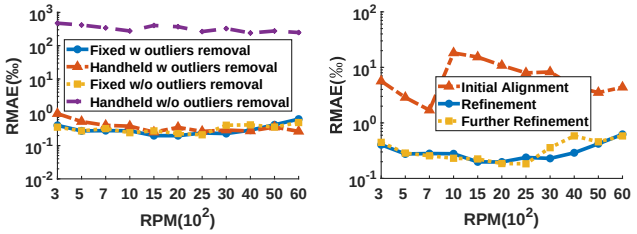
- **DVS-Fixed**: DAVIS346 is fixed on a tripod and placed on the table while recording event streams via DVS.
- **DVS-Handheld**:DAVIS346 is held in hand of a user while recording the event streams. To mitigate subjectivity, we let multiple i.e., ten users perform trials independently and report the mean RMAEs.
- **Laser-Fixed**: laser tachometer is fixed on a tripod and placed on the table. It points to a $1cm \times 1cm$ reflective label on one of the blades.
- **Laser-Handheld**: laser tachometer is held in hand of a user. The user tries to point to the reflective label while measuring the rotation. Similar to DVS-Handheld, ten users are asked to perform the trials independently similar to **DVS-Handheld**, and the mean RMAEs are reported.

4.2 Evaluation on Accuracy of Rotational Speed Estimation

In this section, we will provide extensive evaluations on the accuracy of *EV-Tach* against different parameter settings, including rotational speed, measurement distance, number of blades, host vibration and occlusion ratios. Laser tachometer, as the state-of-the-art rotational speed measurement tool, is chosen as the benchmark to compare with *EV-Tach*.

4.2.1 Ablation study

We first evaluate the effectiveness of **Outliers Removal** and **Estimation Refinement** through an ablation study.



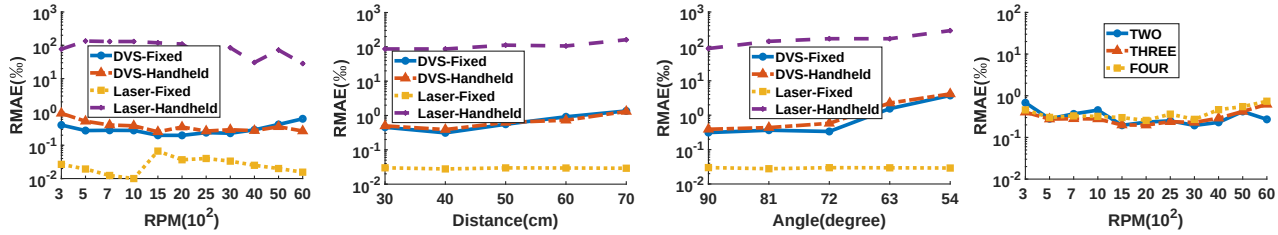
(a) Different methods (b) Different components

Fig. 10. Evaluation of the impact of Outliers Removal on *EV-Tach*(a). RMAE from different components of event stream registration(b).

Evaluation on Outliers Removal: We first evaluate the impact of outliers removal module on our system by comparing the accuracy of *EV-Tach* with or without outlier removal under fixed or handheld measurement. We demonstrate the RMAEs against different rotational speeds, from 300rpm to 6000rpm, in Figure. 10(a). From the result, we can observe, three methods, except for **Handheld w/o outlier removal** can produce accurate measurement with RMAEs less than 1‰, and the average RMAEs of **Fixed w outlier removal)** and **Fixed w/o outlier removal)** are 0.31‰, 0.33‰, respectively. However, in the case of handheld measurement, the average of RMAEs of **DVS-Handheld** rockets from 0.38‰ (**Handheld w outlier removal**) to 330.72‰ (**Handheld w/o outlier removal**). Therefore, outliers removal module is essential for *EV-Tach* to accommodate the subtle movement irrelevant to rotating objects, as it can remove the background noise interference of the rotating object and guarantee the measurement accuracy of our system in the case of handheld.

Evaluation on Estimation Refinement:*EV-Tach* applies a coarse-to-fine strategy to refine the estimation obtained from initial alignment. To show the refinement stage really works and no further refinement is needed, we compute

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2023.3335221



(a) Different rotational speed (b) Different measurement distances (c) Difference view angles (d) Different number of blades

Fig. 11. Evaluation of different measurement methods with changing parameter settings

the RMAE of the outputs of the three different components of *EV-Tach* (**DVS-Fixed**) including initial alignment, refinement and further refinement. The further refinement means to use the rotational speed obtained by refinement to recalculate t'_s by equation 16, then a new refined rotational speed estimation can be obtained by ICP-based event stream registration. The RMAEs against different rotational speeds, from 300rpm to 6000rpm, are shown in Figure 10(b). First, we can observe the RMAEs of initial alignment fluctuates along RPM, which is due to a small step length, i.e., $t_s = 1ms$ cannot provide enough events for stable ICP-based registration results. However, by comparing different stages, we can observe, the first refinement effectively reduces the RMAE for all rotational speeds compared with initial alignment. For example, via refinement, the average of RMAEs across all rotational speeds drops significantly from $7.6‰$ to $0.31‰$, which means approximately 25 times improvement on accuracy. Moreover, further refinement cannot guarantee noticeable improvement and consumes extra resources, therefore, a two-stage strategy with initial alignment and one-time refinement is sufficient to obtain an accurate measurement.

4.2.2 Evaluation on Different Rotational Speeds

From this section, we will compare *EV-Tach* with the laser tachometer under different circumstances. First, the four different measurement methods **DVS-Fixed**, **DVS-Handheld**, **Laser-Fixed** and **Laser-Handheld** are evaluated against different rotational speeds. The measurement distance is set as 40cm. By changing the rotational speed of the servo motor from 300rpm to 6000rpm, we can obtain the corresponding RMAEs of the four different methods as shown in Figure 11(a). As mentioned before, each RMAE is obtained from averaging the results of 30 independent tests and five users are recruited for the handheld measurement. From the figure, we can observe, three methods, except for **Laser-Handheld**, can produce accurate measurement with RMAEs less than $1‰$. Especially, **Laser-Fixed** achieves the lowest RMAE ($< 0.1‰$). However, when the laser tachometer is held in hand, the average of RMAEs of **Laser-Handheld** rockets to $87‰$ due to the subtle movement of user's hand. When it comes to *EV-Tach*, the **DVS-Fixed** and **DVS-Handheld** methods produce similar RMAEs and the average of RMAEs of **DVS-Handheld** is below $0.4‰$ which is over 210 times better than **Laser-Handheld**. Therefore, we can claim that, *EV-Tach* is robust to the subtle movement. While the laser tachometer, though designed as a portable device, is not suitable for handheld measurement. It is worth noting that, it is very hard to point to the small reflective label attached on the blade when the target is fast rotating. It normally takes at least tens of seconds for user to point to the correct spot then it deviates easily from the label due to subtle movement of hand. Comparatively, *EV-Tach* is significantly more convenient. Users only need to make the camera approximately face to the front of the rotating target and the procedure is in no time. By comparing **DVS-fixed** and **DVS-handheld** in Figure 11(a), we find different trends with the increase of rotational speed. During experiment, we observe slight vibration of the servo motor with the desk when it is fast rotating, which may cause the trivial performance decrease of **DVS-fixed** with the growth of the rotational speed. While for handheld scenario, the subtle movement from hand and host may cancel out by chance. Therefore, there is no clear trend on the performance change with the increase of rotational speed for **DVS-Handheld**.

4.2.3 Evaluation on Measurement Distance

The distance to the rotating target during measurement can be various in use. In this section, we evaluate the accuracy of the four methods against different measurement distances. Again, RMAEs of the four methods are computed by gradually increasing the measurement distance from 30cm to 70cm and the results are shown in Figure 11(b). Each RMAE in the figure is obtained from averaging the RMAEs obtained under different rotational speeds. From the results we can observe, **Laser-Fixed** is not affected by the measurement distance: as far as the reflection of laser can reach, it will produce stable and accuracy measurement. The RMAEs of the remaining methods all increases with the growth of measurement distance due to different reasons. For **Laser-Handheld**, with the increase of distance, it becomes more difficult for users to point the laser to the small reflective label and keep not deviate from the correct spot during the measurement. The accuracy of *EV-Tach* approaches, **DVS-Fixed** and **DVS-Handheld**, declines as the occupied area of the event stream shrinks with the growth of distance. However, it can be solved by using a zoom lens on event-camera. For example, by changing the focal-length of DAVIS346 from 8mm to 120mm, we can zoom-in on the rotating target. According to our evaluation, when the measurement distance is 7m, the average of RMAEs of **DVS-Handheld** with 120mm focal-length is below $0.69‰$ which is similar to that of **DVS-Handheld** with 8mm focal-length in $50 - 60$cm measurement distance.

4.2.4 Evaluation on View Angle

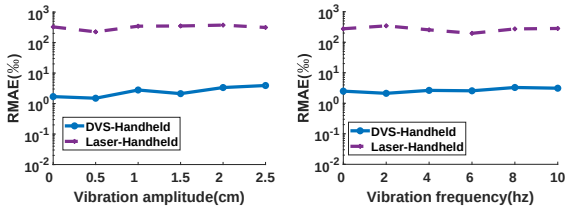
We evaluated the effect of view angle on *EV-Tach*. We keep the distance from the Davis 346 and laser tachometer to the center of the servo motor to be around $40cm$, then we conduct experiments at 90, 81, 72, 63, and 54 degrees, where 90 degrees means that the Davis 346 lens and the emitted laser is perpendicular to the rotating object. For each view angle, the average RMAEs of 300rpm to 6000rpm are showed in Fig. 11(c). Overall, we can see that the fixed-laser tachometer is not affected by the angle change because of the special *retro-reflective* label. It reflects laser directly back to its source and produce high measurement accuracy because the source (laser tachometer) is fixed. However, for the handheld scenario, we can see that, though the RMAE of **DVS-handheld** keeps increasing from 90 degrees to 18 degrees, the error is still significantly less than that of **Laser-Handheld**: the improvement is at least 75 times (2.25‰ v.s. 166.7‰ when the view angle is 54 degree).

4.2.5 Evaluation on Different Number of Blades

Generally, as three-blade propellers are the most common rotating targets to be seen [48], in the evaluations above, we set the number of blades to be three. However, it is possible the rotating targets are various in number of blades. For example, most of drones are equipped with two-blade propellers. We evaluate the accuracy of *EV-Tach* on estimating the rotational speed of propellers with two, three and four blades respectively. By gradually changing the rotational speed of servo motor, the RMAEs are calculated and presented on Figure. 11(d). From the results we can observe, *EV-Tach* achieves similar accuracy of measurement for all types of blades and the average of RMAEs are $0.32‰$, $0.31‰$ and $0.39‰$. Therefore, *EV-Tach* can work on the rotating propellers with different number of blades.

4.2.6 Evaluation on Robustness to Host Vibration

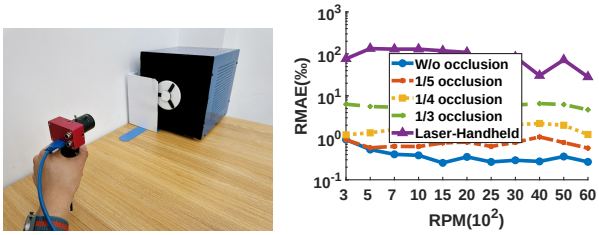
In real-world scenarios, the hosting devices are sometimes not stable, e.g., vibrating. To obtain accurate measurement, the tachometer should be able to accommodate the slight motion of the hosting device to some extent. Therefore, we simulate and quantify the vibration of host to evaluate the robustness of *EV-Tach* and laser tachometer to vibration. To generate vibration with different frequencies and amplitudes precisely, we utilize two vibration generators, namely the AUBO Robotics i16 [49] and the Modal Excite SA-JZ002 [50], to control the vibration amplitude and frequency, respectively. AUBO Robotics i16 provides vibration amplitude from 0 to 2.5cm at fixed frequency of 2Hz while Modal Excite SA-JZ002 can vibrate between 0 to 10Hz with fixed amplitude of 0.5cm. However, because the servo motor is too heavy to be carried by the vibrating platforms, we utilize the intelligent fan management software, Fan Xpert 4 intelligent, on the ASUS Z790 motherboard [51] to control the rotation speed of the TL-C12 PRO-G case fan[52]. The average RMAEs of different host vibration amplitudes and frequencies for different measurement methods are shown in Figure 12(a) and Figure 12(b). The average RMAEs of **DVS-Handheld** are 2.6 ‰ and 2.7‰, for different vibrating amplitudes and frequencies. While the average RMAEs of **Laser-Handheld** are 322 ‰ and 272‰. It is worth noting that as the motherboard does not control the fan speed as precisely as the servo motor, we are unable to obtain an accurate groundtruth. Therefore, the RMAEs of DVS-Handheld were slightly higher than the servo motor measurement results. However, the accuracy of **DVS-Handheld** is still 123.8 and 100.7 times better than **Laser-Handheld** on changing vibration amplitudes and frequencies scenarios, respectively.



(a) Different amplitudes (b) Different frequencies

Fig. 12. Relative error vs. different host vibration amplitudes and frequencies for different measurement methods.

4.2.7 Evaluation on Occlusion



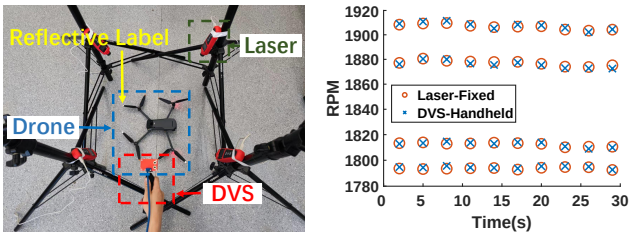
(a) The experiment setting (b) Different ratios of occlusion

Fig. 13. The experiment setting of object occlusion and the measurement results of different ratios of occlusion

It is possible part of the rotating object is occluded when measured. Thus, we evaluate the accuracy of *EV-Tach* when part of the rotating object is occluded. As shown in figure 13(a), a cardboard is placed in front of the servo motor to cover part of rotating disc. We change the ratio of occlusion by moving the cardboard from edge to the center of the disc and the ratio refers to how much the diameter of the disc is occluded. By gradually changing the rotational speed, the RMAEs of different ratio of occlusion from handheld DVS and handheld laser tachometer are calculated and presented in Figure. 13(b). From the results we can observe, the accuracy of the measurement decreases gracefully. For examples, the average RMAEs *EV-Tach* with $1/5$ occlusion and $1/4$ occlusion are $0.73‰$ and $1.73‰$, respectively. However, when the occlusion is significant, e.g., $1/2$, *EV-Tach* will fail to determine the correct number of blades which is the limit of our system. It is worth noting that, when both of the host and tachometer are in fixed state, the laser tachometer can obtain promising measurement results as long as the reflective label is not occluded during rotating. However, when the laser tachometer is held in hand, which is the practical scenario concerned in this paper, the average of RMAEs of Laser-Handheld rockets to 87‰, which is significantly higher than DVS-Handheld as shown in Figure 13(a).

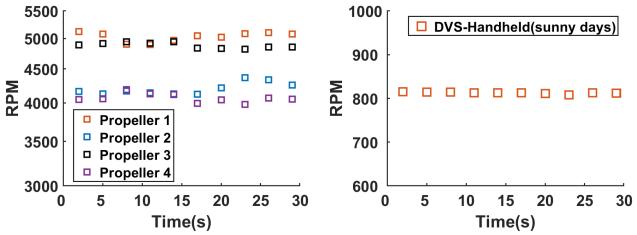
4.2.8 Evaluation on Practical Devices

The previous evaluations are based on our customized device with servo motor and the blades are printed. To prove our methods work on practical devices, we set up new experiments to measure the rotational speeds of propellers of drone, fans of air-conditioner and automobile wheels.



(a) The experiment setting (b) The measurement results

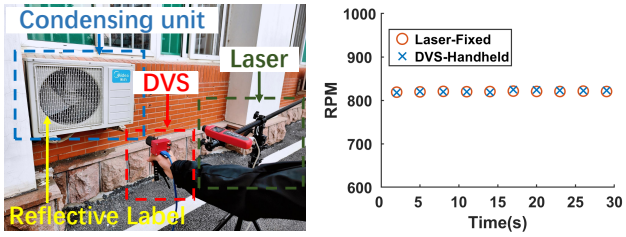
Fig. 14. The experiment setting of measuring the rotational speeds of propellers of drone and the measurement results from laser tachometer and handheld DVS.



(a) DVS-Handheld(drone hover) (b) DVS-Handheld(sunny days)

Fig. 15. The DVS-Handheld measurement results for drone propellers (a) and condensing unit of air-conditioner (b), respectively, where laser tachometer cannot work in those case.

Drone propellers: The personal drones normally have four propellers and the motions of drones are determined by changing the rotational speed of one or more propellers. As personal drones cannot provide precise rotational speed, we use laser tachometers as the tools to obtain benchmarking results. During measurement, the drone first is landing on the floor while its propellers are fast rotating for the laser tachometers cannot handle the movement of drone in the air (see Figure 14(a) for reference). To measure the four propellers with DVS and laser at the same time, four laser tachometers are screwed on the tripod arms and each laser tachometer is calibrated to one of the propellers with a reflective label attached, while a DVS is held in hand of a user as shown in the figure. The measured rotational speeds from laser tachometers and DVS are synchronized and shown in Figure 14(b). From the results, we can find the rotational speeds of the four propellers are different and those obtained from DVS match well with those from laser tachometers and the average relative difference between the two methods is less than 1‰. Another important merit of using DVS in this experiment is only one DVS is needed to measure the four rotating propellers at the same time. In addition, we also measure the rotational speeds of the propellers when the drone is hovering. As the hovering drone is too wobbly for the laser tachometer to obtain any reading, we only demonstrate the results from handheld DVS in Figure 15(a). Though the "groundtruth" is missing, we can observe the rotational speeds of the propellers are quite steady when the drone is hovering and are significantly higher than the landing status to lift the drone in the air.



(a) The experiment setting (b) The experiment results

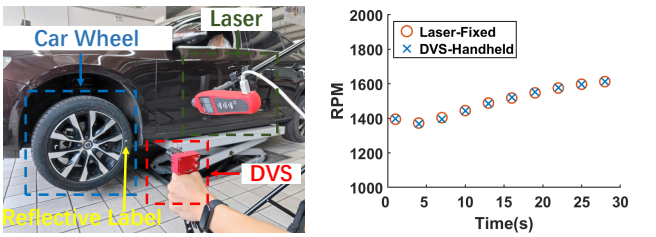
Fig. 16. The experiment setting of measuring the rotational speeds of the fan of condensing unit and the measurement results from laser tachometer and handheld DVS.

Condensing unit of air-conditioner: We also measure the rotational speed of the fan of condensing unit which is an important indicator to determine if the air-conditioner is working properly. The experiment is conducted outdoor as shown in Figure 16(a). One laser tachometer is screwed on tripod arm and faced to the rotating fan with reflective label, while DVS is held in hand of a user. The measured results shown in Figure 16(b) demonstrate the handheld DVS can produce almost the same rotational speeds as laser tachometer. Moreover, according to our experience in this experiment, the DVS-based approach shows superior convenience and robustness than laser tachometer. First, before using laser tachometer, we have to dissemble the grill of condensing unit to attach the reflective label onto the fan, and then re-assemble it during measurement for safety consideration. While the DVS can obtain the measurement directly. Second, the performance of laser tachometer is largely affected by outdoor lighting condition: it fails to produce any results during sunny days from morning to afternoon as it is too bright outside. While the DVS is robust to the outside lighting conditions. As shown in Figure. 15(b), though the "groundtruth" is missing, we can observe that handheld DVS can produce reasonable rotational speeds on a sunny day.

It is worth noting that, the evaluation also indicates the robustness of *EV-Tach* to the occlusion under practical scenario. Though the grill of the condensing unit poses significant occlusion on the blades of the fan, *EV-Tach* can achieve comparable accuracy to laser tachometer. It is because the occlusion is evenly distributed with the frames and the shape of the rotating object is preserved well which is the key for *EV-Tach*.

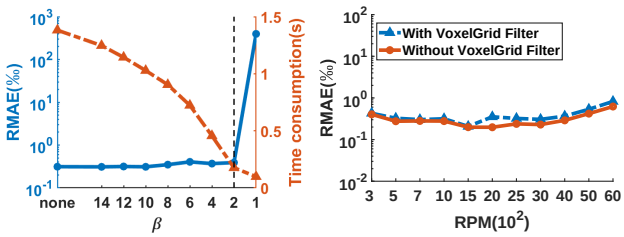
Automobile wheels: The rotational speed of car wheels is an important item to check during car maintenance. As shown in Figure 17(a), we set up a practical experiment in an indoor car workshop. The rotational speeds are measured when the car is lifted and wheels are rolling in the air. From the results in Figure 17(b), we can observe the deceleration and acceleration of the engine, and the results from both approaches match each other.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2023.3335221



(a) The experiment setting (b) The experiment results

Fig. 17. The experiment setting of measuring the rotational speeds of the wheels of automobile and the measurement results from laser tachometer and handheld DVS.



(a) Accuracy v.s. Time (b) $\beta = 2$

Fig. 18. RMAEs and time consumption of *EV-Tach* on Raspberry Pi.

4.3 Evaluation on System Overhead

By implementing *EV-Tach* on Raspberry Pi, we can evaluate its performance on embedded platform including the trade-off between the accuracy and efficiency, and resource consumption on time and energy.

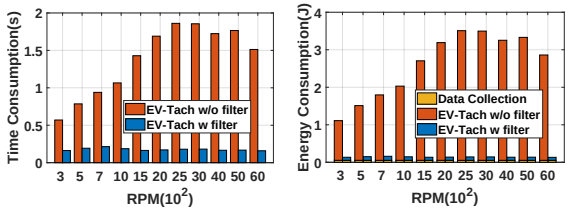
4.3.1 Trade-off Between Accuracy and Efficiency

As embedded platforms are resource-constrained, the trade-off between the accuracy and efficiency needs to be carefully tuned during implementation. Specifically, in *EV-Tach*, VoxelGrid filtering [53] is adopted to significantly reduce the number of events to be processed so that the computation time and energy consumption can be saved. However, it may also bring negative influence on the accuracy of ICP-based registration. Therefore, the size of voxel V_s for downsampling should be determined for implementation. To adapt to the sizes of rotating target and measurement of distance, V_s is calculated by the diameter of the rotating target D_{X-Y} in spatial domain (the coordinate plane formed by X and Y axes) and the number of division β in each dimension, i.e., $V_s = D_{X-Y}/\beta$. The diameter is obtained by finding the minimum squared bounding box [54] containing the whole rotating target. For example, when $\beta = 2$, the rotating target in spatial domain (the plane of X-Y axes) are divided into four equal areas and the corresponding size of voxels in spatial-temporal domain is $(D_{X-Y}/2, D_{X-Y}/2, D_{X-Y}/2)$. By decreasing β from 14 to 1, we estimate the corresponding RMAEs and time consumption of *EV-Tach* running on Raspberry Pi. The average of the results from different rotational speeds (300 to 6000rpm) are reported in Figure 18(a). There are two special cases in the figure: $\beta = none$ means no VoxelGrid filter is applied and $\beta = 1$ mean no division in spatial domain. From the results, except the special cases, VoxelGrid filtering does not bring noticeable influence on the accuracy while time consumption drops significantly with the decrease of β. Therefore, $\beta = 2$ is chosen to reduce the computation time of *EV-Tach* on embedded platforms while preserving the accuracy. Figure 18(b) shows the detailed impact of downsampling on measurement accuracy when $\beta = 2$ is chosen by comparing the RMAEs of *EV-Tach* with and without VoxelGrid filtering on measuring rotational speeds from 300rpm to 6000rpm. The results show that *EV-Tach* with filter provides accurate and stable results over different rotational speeds.

4.3.2 Resource Consumption on Raspberry Pi

At last, we profile the computation time and energy consumption of *EV-Tach* with and without VoxelGrid filter on Raspberry Pi to show how significant the VoxelGrid filtering improves the efficiency of *EV-Tach* on resource-constrained systems.

Computation Time: The computation time can be easily obtained from the console of the system. Figure 19(a) shows the computation time of *EV-Tach* with or without the VoxelGrid filtering on measuring different rotational speeds. From the results, we can observe, the filter can effectively speed up the processing of *EV-Tach* by up to 10.5 times, i.e., when the rotational speed is 5000, the computation time can be reduced from 1765.5ms to 167.5ms. The average improvement on computation time is approximately 8 times, by shortening the average computation time from 1381.6ms to 176.6ms. Accounting data collection, the average time consumption of one measurement is approximately 326ms.



(a) Time consumption (b) Energy consumption

Fig. 19. Computation time and energy consumption of *EV-Tach* (w or w/o VoxelGrid Filter) on Raspeberry Pi.

Energy Consumption: Then we estimate the energy consumption of *EV-Tach* on Raspberry Pi with or without the VoxelGrid filtering. External tool is plugged in the micro-USB of Raspberry Pi to monitor the voltage and current in different states. Table 1 presents the voltage and current of Raspberry Pi under different states, including idle, sensing and processing (with or without filter).

TABLE 1
Current and voltage of Raspberry Pi under different running states.

STATE	idle	sensing	processing	processing (filter)
Current	680mA	750mA	1050mA	780mA
Voltage	5.02V	5.02V	5.01V	5.02V

The energy consumption (E_c) can be approximated by multiplication of *Current*, *Voltage* and running time (*RT*), i.e., $E_c = Current \times Voltage \times RT$. Then the energy consumption of system in idle state should be extracted to

obtain the actual cost introduced by *EV-Tach*. Figure 19(b) presents the energy consumption of sensing (data collection) and processing (running *EV-Tach*) on Raspberry Pi. By comparing the processing cost of the two approaches of *EV-Tach*, we can observe, the VoxelGrid filtering significantly reduces the energy consumption by up to 40 times, i.e., from 3.28J to 0.084J when rotational speed is 5000rpm. The actual cost of obtaining one measurement in average, including both sensing and processing, is about 141.3mJ (52.7mJ for sensing and 88.6mJ for processing). .

5 DISCUSSION ON ADVANTAGES AND LIMITATIONS

Advantages over laser tachometer: As the description and evaluation in this paper, *EV-Tach* shows a number of superior characteristics on the task of rotational speed measurement over the state-of-the-art laser tachometer which is dominant in the market. First and foremost, *EV-Tach* is a real handheld tachometer and it is robust to subtle movement of user's hand and even vibrating host of the rotating objects. While, though the laser tachometer is designed as a portable device, its accuracy drops significantly when used as handheld. Second, the use of *EV-Tach* is more convenient than laser tachometer and no preparation is needed before measurement; while laser tachometer must be pointed to the small reflective labels or spot in high-contrast paint in few occasions). When used as handheld, it takes users tens of seconds to obtain a valid (but not necessarily accurate) reading when the target is fast rotating or in long distance. Third, *EV-Tach* is robust to the vibrating host of the rotating target which causes failed measurement for laser tachometer. Fourth, the *EV-Tach* is able to measure multiple rotating targets at the same time, which is impossible for laser tachometer. Finally, compared with the other vision-based method, it achieves significantly higher range of measurement than those with conventional RGB cameras and is more cost-effective than those applying high-speed cameras.

Limitations: However, as the principle of DVS, *EV-Tach* also shares some similar limitations to the vision-based approaches. First of all, it requires the rotating targets in form of propellers or with uneven texture so that different phases of rotation can be detected. However, like the reflective label for laser tachometer, the usability of *EV-Tach* can be extended if unique pattern or labels are allowed to be attached on the rotating object. For example, when the rotating object is a flat disc with uniform texture, a label (e.g., a straight line), which is high contrast to the disc, can be attached to aid the measurement. Second, constrained by the hardware design, the accuracy and range of measurement of *EV-Tach* in this paper is lower than laser tachometer. However considering the spatial resolution of DAVIS346 is only 346×240, the performance of *EV-Tach* is expected to be improved by using event-cameras with higher spatial and temporal resolution. Third, *EV-Tach* may fail when the occlusion is significant as it cannot determine the correct number of blades. On the contrary, for laser tachometer, when both of the tachometer and host are in fixed-state, it is able to obtain accurate measurement as long as the reflective label is not occluded while rotating. However, as laser tachometer is not robust to vibration, *EV-Tach* can achieve higher accuracy than laser tachometer when used as handheld when over half of the rotating objects are disclosed.

6 CONCLUSION

In this paper, we propose, *EV-Tach*, a rotational speed measurement system based on dynamic vision sensing on mobile devices to achieve efficient high-fidelity and convenient estimation when used handheld. *EV-Tach* starts with extracting multiple rotating targets via K-means clustering and a heatmap-based initial centroids selection method is proposed to improve the robustness of the clustering. Then angle of rotation is estimated via a coarse-to-fine ICP-based event streams registration method and rotational speed can be calculated afterwards. Extensive evaluations are conducted and the results show that the accuracy of *EV-Tach* is comparable to laser tachometer in fixed deployment but is over 210 better in handheld measurement mode, which is the focus of this work. *EV-Tach* is robust to vibrating host of rotating target in which the laser tachometer fails to provide reasonable results. At last, we implement *EV-Tach* on Raspberry Pi and apply the VoxelGrid filtering to improve its efficiency. By profiling the energy and time consumption, we can find the VoxelGrid filtering significantly reduces the time consumption by up to 10 times and energy consumption by 40 times to make *EV-Tach* run in-situ on embedded devices.

ACKNOWLEDGMENT

This work is partially supported by National Key R&D Program of China, Grant No. 2021YFE0111600, Natural Science Foundation of Shandong Province, Grant No. 2022HWYQ-040 and Grant No.ZR2021LZH006, Taishan Scholars Program and National Science Foundation China, Grant No. 61972230.

REFERENCES

[1] Haynes, "Vehicle inspections," 2022. [Online]. Available: https://haynes.com/ en-us/tips-tutorials/what-know-about-vehicle-inspections-all-50-states

[2] CHECKLINE, "Htm hand-held mechanical tachometer," 2023. [Online]. Available: https://www.checkline.com/product/HTM

[3] L. Li, H. Hu, Y. Qin, and K. Tang, "Digital approach to rotational speed measurement using an electrostatic sensor," *Sensors (Basel, Switzerland)*, vol. 19, 2019.

[4] U. A. P. F. K. R. T. J. Kathirvelan J, Babu Varghese, "Hall effect sensor based portable tachometer for rpm measurement," *International Journal of Computer Science and Engineering Communications*, vol. 2, 2014.

[5] S. A. K. B. Kim T., Adeli H., "A high accuracy method for rapid measurement of resulted code pattern radial runout of rotary optical encoder disc," in *International Conference on Control and Automation*, vol. 256, 2011.

[6] UNI-T, "Ut372 tachometer," 2022. [Online]. Available: https://meters.uni-trend.com/

[7] Y. Zhai, S. Fu, C. Yin, H. Zhou, and C. Gao, "Detection of angular acceleration based on optical rotational doppler effect," *Optics Express*, vol. 27, p. 15518, 05 2019.

[8] Y. Wang, L. Wang, and Y. Yan, "Rotational speed measurement through digital imaging and image processing," in *2017 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, 2017, pp. 1–6.

[9] J. Zhong, S. Zhong, Q. Zhang, and Z. Peng, "Measurement of instantaneous rotational speed using double-sine-varying-density fringe pattern," *Mechanical Systems and Signal Processing*, vol. 103, pp. 117–130, 03 2018.

[10] X.-d. Zhu and S.-n. Yu, "Measurement angular velocity based on video technology," in *2011 4th International Congress on Image and Signal Processing*, vol. 4, 2011, pp. 1936–1940.

[11] F. Natili, F. Castellani, D. Astolfi, and M. Becchetti, "Video-tachometer methodology for wind turbine rotor speed measurement," *Sensors*, vol. 20, no. 24, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/24/7314

[12] T. Wang, Y. Yan, L. Wang, and Y. Hu, "Rotational speed measurement through image similarity evaluation and spectral analysis," *IEEE Access*, vol. 6, pp. 46 718–46 730, 2018.

[13] Y. Zhao, Y. Li, S. Guo, and T. Li, "Measuring the angular velocity of a propeller with video camera using electronic rolling shutter," *Journal of Sensors*, vol. 2018, pp. 1–9, 03 2018.

[14] H. Kim, Y. Yamakawa, T. Senoo, and M. Ishikawa, "Visual encoder: robust and precise measurement method of rotation angle via high-speed rgb vision," *Opt. Express*, vol. 24, no. 12, pp. 13 375–13 386, Jun 2016. [Online]. Available: http://opg.optica.org/oe/abstract.cfm?URI=oe-24-12-13375

[15] 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.

[16] Wikipedia, "Event camera," https://en.wikipedia.org/wiki/Event camera, 2023.

[17] H. Li, H. Liu, X. Ji, G. Li, and L. Shi, "Cifar10-dvs: An event-stream dataset for object classification," *Frontiers in Neuroscience*, vol. 11, 2017.

[18] U. of Zurich, "Event-based vision, event cameras, event camera slam," https://rpg.ifi.uzh.ch/research dvs.html, 2021.

[19] J. A. Leñero-Bardallo, T. Serrano-Gotarredona, and B. Linares-Barranco, "A 3.6 μ s latency asynchronous frame-free event-driven dynamic-vision-sensor," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 6, pp. 1443–1455, 2011.

[20] inivation, "Davis346," 2022. [Online]. Available: https:// inivation.com/ wp-content/uploads/2019/08/DAVIS346.pdf

[21] prophesee, "Evk4 the ultra-light and compact, hd event-based vision evaluation kit built to endure field testing conditions." https://www.prophesee.ai/event-camera-evk4/, 2023.

[22] R. Benosman, C. Clercq, X. Lagorce, S.-H. Ieng, and C. Bartolozzi, "Event-based visual flow," *IEEE Transactions on Neural Networks*, vol. pp, p. 1, 11 2013.

[23] M. Heggo, L. Bhatia, and J. A. McCann, "Rftacho: Non-intrusive rf monitoring of rotating machines," in *21st ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2022.

[24] 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.

[25] Y. Wang, B. Du, Y. Shen, K. Wu, G. Zhao, J. Sun, and H. Wen, "Ev-gait: Event-based robust gait recognition using dynamic vision sensors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6358–6367.

[26] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "Ev-flownet: Self-supervised optical flow estimation for event-based cameras," *arXiv preprint arXiv:1802.06898*, 2018.

[27] Y. Bi, A. Chadha, A. Abbas, E. Bourtsoulatze, and Y. Andreopoulos, "Graph-based object classification for neuromorphic vision sensing," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 491–501.

[28] Y. Wang, X. Zhang, Y. Shen, B. Du, G. Zhao, L. C. C. Lizhen, and H. Wen, "Event-stream representation for human gaits identification using deep neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[29] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[30] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 5105–5114.

[31] Q. Wang, Y. Zhang, J. Yuan, and Y. Lu, "Space-time event clouds for gesture recognition: From rgb cameras to event cameras," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2019, pp. 1826–1835.

[32] G. Gallego and D. Scaramuzza, "Accurate angular velocity estimation with an event camera," *IEEE Robotics and Automation Letters*, vol. 2, pp. 632–639, 2017.

[33] N. Diaz, O. Gallo, J. Caceres, and H. Porras, "Real-time ground filtering algorithm of cloud points acquired using terrestrial laser scanner (tls)," *International Journal of Applied Earth Observation and Geoinformation*, vol. 105, p. 102629, 2021.

[34] Z. Zhang, J. Zhang, and H. Xue, "Improved k-means clustering algorithm," in *2008 Congress on Image and Signal Processing*, vol. 5. IEEE, 2008, pp. 169–172.

[35] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 224–227, 1979.

[36] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," ser. SODA '07. USA: Society for Industrial and Applied Mathematics, 2007.

[37] P. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

[38] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," in *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, 1991, pp. 2724–2729 vol.3.

[39] E. Recherche, E. Automatique, S. Antipolis, and Z. Zhang, "Iterative point matching for registration of free-form curves," *Int. J. Comput. Vision*, vol. 13, 07 1992.

[40] B. Krebs, P. Sieverding, and B. Korn, "A fuzzy icp algorithm for 3d free-form object recognition," in *Proceedings of 13th International Conference on Pattern Recognition*, vol. 1, 1996, pp. 539–543 vol.1.

[41] J. Konecny, M. Prauzek, and J. Hlavica, "Icp algorithm in mobile robot navigation: Analysis of computational demands in embedded solutions**this work was supported by the project sp2016/162, development of algorithms and systems for control, measurement and safety applications ii of student grant system, vsb-tu ostrava." *IFAC-PapersOnLine*, vol. 49, no. 25, pp. 396–400, 2016.

[42] M. Sinko, P. Kamencay, R. Hudec, and M. Benco, "3d registration of the point cloud data using icp algorithm in medical image analysis," in *2018 ELEKTRO*, 2018, pp. 1–6.

[43] wikipedia, "Iterative closest point," https://en.wikipedia.org/wiki/Iterative_closest_pointReferences.

[44] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 5, pp. 698–700, 1987.

[45] F. Wang and Z. Zhao, "A survey of iterative closest point algorithm," in *2017 Chinese Automation Congress (CAC)*, 2017, pp. 4395–4399.

[46] mathworld, "Eulerangles," 2022. [Online]. Available: https://mathworld.wolfram.com/EulerAngles.html

[47] R. pi, "Raspberry pi," 2022. [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/

[48] A. Chaudhuri, R. Datta, M. P. Kumar, J. P. Davim, and S. Pramanik, "Energy conversion strategies for wind energy system: Electrical, mechanical and material aspects," *Materials*, vol. 15, no. 3, 2022.

[49] AUBO, "Aubo robotics i16 collaborative robot," https://unchainedrobotics.de/en/products/robot/cobot/aubo-robotics-i16, 2023.

[50] SHIAO, "20n modal excite," https://www.chinaglobalmall.com/products/561236255397, 2023.

[51] asus, "Prime z790-p," https://www.asus.com.cn/motherboards-components/motherboards/prime/prime-z790-p/, 2023.

[52] Thermalright, "tl-c12-pro-g," http://thermalright.com/product/tl-c12-pro-g/, 2023.

[53] M. Miknis, R. Davies, P. Plassmann, and A. Ware, "Near real-time point cloud processing using the pcl," in *2015 International Conference on Systems, Signals and Image Processing (IWSSIP)*. IEEE, 2015, pp. 153–156.

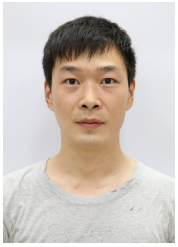
[54] J. Wood, *Minimum Bounding Rectangle*. Boston, MA: Springer US, 2008, pp. 660–661. [Online]. Available: https://doi.org/10.1007/978-0-387-35973-1$_7$83

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2023.3335221

Guangrong Zhao is a PhD candidate at Shandong University and received his BS and MS degrees from Wuhan University of Science and Technology, China, in 2018, and Harbin Engineering University, China, in 2021, respectively. He is the author and co-author of several top papers on wireless sensor networks and computer vision, such as Neurips, TPAMI, CVPR and ACM TOSN. His research interests include wireless sensor networks, computer vision, mobile computing, etc.

Yiran Shen is professor in School of Software, Shandong University. He received his BE in communication engineering from Shandong University, China and his PhD degree in computer science and engineering from University of New South Wales. He published regularly at top-tier conferences and journals. Generally speaking, his research interest is the merging area of Internet-of-Things (IoTs) and artificial intelligence. He is a senior member of IEEE.

Ning Chen Ning Chen is a graduate student in School of Software, Shandong University. He received his BE in software engineering from Shandong University, China, in 2022. His research interest is the mobile computing.



Pengfei Hu is a professor in the School of Computer Science and Technology at Shandong University. He received Ph.D. in Computer Science from UC Davis. His research interests are in the areas of cyber security, data privacy, and mobile computing. He has published more than 40 papers in premier conferences and journals on these topics. He served as TPC for numerous prestigious conferences, and associate editors for IEEE TWC and IEEE IoTJ. He is the recipient of 2022 ACM SIGBED China Rising Star Award.

Lei Liu is a full professor in the school of software, Shandong University. He obtained the master and Ph.D degree in 2005 and 2010, respectively. Dr. LIU has published over 70 research papers on international conferences and journals. His research interest includes network performance engineering, 5g technology, quality of service, IoT and UAVs.

Hongkai Wen is an Associate Professor in Department of Computer Science, University of Warwick. Before that he obtained his D.Phil at the University of Oxford, and became a postdoctoral researcher in a joint project between Oxford Computer Science and Robotics Institute. Broadly speaking, his research belongs to the area of Cyber-Physical Systems, which use networked smart devices to sense and interactive with the physical world.