Towards Robust Perception for Assistive Robotics: An RGB-Event-LiDAR Dataset and Multi-Modal Detection Pipeline

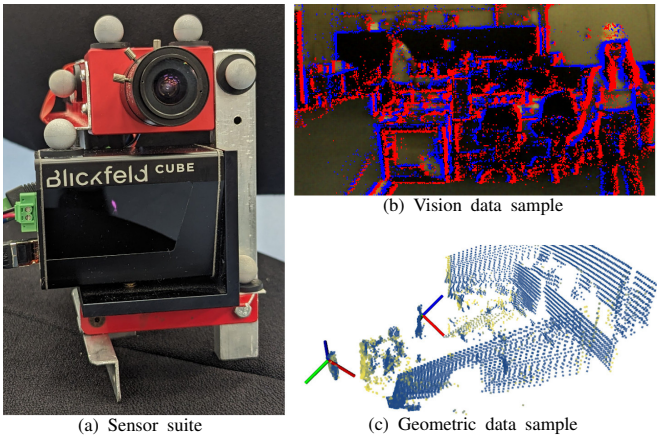
Adam Scicluna, Cedric Le Gentil, Sheila Sutjipto and Gavin Paul

***Abstract*— The increasing adoption of human-robot interaction presents opportunities for technology to positively impact lives, particularly those with visual impairments, through applications such as guide-dog-like assistive robotics. We present a pipeline exploring the perception and "intelligent disobedience" required by such a system. A dataset of two people moving in and out of view has been prepared to compare RGB-based and event-based multi-modal dynamic object detection using LiDAR data for 3D position localisation. Our analysis highlights challenges in accurate 3D localisation using 2D image-LiDAR fusion, indicating the need for further refinement. Compared to the performance of the frame-based detection algorithm utilised (YOLOv4), current cutting-edge event-based detection models appear limited to contextual scenarios, such as for automotive platforms. This is highlighted by weak precision and recall over varying confidence and Intersection over Union (IoU) thresholds when using frame-based detections as a ground truth. Therefore, we have publicly released this dataset to the community, containing RGB, event, point cloud and Inertial Measurement Unit (IMU) data along with ground truth poses for the two people in the scene to fill a gap in the current landscape of publicly available datasets and provide a means to assist in the development of safer and more robust algorithms in the future: `https://uts-ri.github.io/revel/`.**

I. Introduction

Training guide dogs requires significant time and financial resources [1], with only about half of the dogs successfully completing the programs [2]. This creates a gap between the availability of guide dogs and the needs of visually impaired individuals. Thus, there is growing interest in cost-effective robotic alternatives [3]. These robotic substitutes must replicate the "intelligent disobedience" of guide dogs, where the robot refuses unsafe commands based on its understanding of an environment. Therefore, reliable perception and decision-making algorithms are necessary to ensure user safety.

Perception and scene understanding are essential capabilities for robotic systems to be integrated into daily life. Such systems must robustly comprehend their surroundings in real-time for subsequent decision-making and actions toward safe operations. In the context of guide-dog-like assistive robotics, this translates into the accurate detection and identification of both static and dynamic objects such as cars, bicycles, pedestrians, etc, and the ability to estimate their 3D pose and dynamics. This would enable the robot to navigate and avoid hazards, enhancing safety and usability.



(a) Sensor suite (b) Vision data sample (c) Geometric data sample

Fig. 1: (a) Sensor suite featuring a DAVIS 346 frame-event camera and a Cube1 LiDAR for dataset collection. (b) DAVIS camera data sample: events (polarity-coloured in red or blue) overlaid on the RGB frame. (c) LiDAR scan sample with object motion-captured ground truth poses (frames).

Recent machine learning advances and GPU availability have enabled efficient context and scene recognition from RGB images using neural networks [4], [5]. Yet, controlling movement or manipulation demands 3D pose knowledge. Although RGBD and stereo cameras provide accurate depth information, their low dynamic range and susceptibility to motion blur, along with RGBD cameras' poor outdoor performance due to sunlight's infrared radiation, pose challenges in safety-critical, dynamic or brightly lit environments [6]. To address this, LiDAR-camera multi-modal sensor suites are used, though LiDAR's sparsity, noise, and slow acquisition rate complicate detection. Thus, RGB-LiDAR systems still struggle with standard cameras' limitations in scenarios under high dynamic range or low light conditions.

Event-based cameras [7] offer a solution to the limitations of traditional frame-based vision by capturing pixel-level changes in illumination independently. However, object detection methods for event cameras are still in their early stages compared to those for RGB data [8], [5]. To fully exploit the benefits of event cameras, developing new algorithms tailored to their unique data output is crucial.

Early RGB-based object detection used handcrafted features and classic machine learning but struggled with variability and extensive parameter tuning. CNNs revolutionised the field with superior performance in managing data variation [4], greatly aided by extensively labelled public datasets [9], [10], [11], [12], where millions of images across

All authors are with the Robotics Institute, Faculty of Engineering and Information Technology, University of Technology Sydney (UTS), Australia. Corresponding author: {adam.scicluna@alumni.uts.edu.au}

Cedric Le Gentil is supported by the Australian Research Council Discovery Project under Grant DP210101336. Sheila Sutjipto is supported by Australian Government Research Training Program Scholarships. The authors would like to acknowledge the support from the ARC Industrial Transformation Training Centre (ITTC) for Collaborative Robotics in Advanced Manufacturing under grant IC200100001.

thousands of categories provide a crucial benchmark for performance evaluation. Tailored datasets like KITTI [13] for autonomous driving have also been pivotal in advancing algorithm development. Prominent approaches include R-CNN and its iterations [14], [5], where selective searches generate regions classified by a CNN. Recent efforts have focused on improving efficiency evident with YOLO [15] and single shot detectors [16], [17]. Works have also combined object detection with semantic segmentation to enhance scene awareness and object analysis [18], [19].

Frame-based algorithms do not translate well for use with event camera data. Consequently, efforts have been made to reconstruct dense greyscale images from sparse event data to feed to a CNN, introducing an intermediate step that increases the computational burden and latency. Alternatively, CNNs [20], spiking neural networks [21], and graph neural networks [22] have demonstrated object detection in the event space. Recently, transformers like RVTs have achieved state-of-the-art performance on the Gen1 and 1-Mpx [23] datasets [24], [23], [25]. While promising, these methods lack the accessibility of algorithms like YOLO [8]. Furthermore, event camera-based techniques struggle to generalise to other scenes due to the limited variety in existing datasets.

Leveraging the complementary nature of LiDAR point clouds and RGB images enables a more comprehensive scene understanding [26]. For object detection, strategies include using RGB-trained networks on image-like data generated from LiDAR data [27], constructing pseudo-LiDAR data from RGB images [28], using 2D detectors to propose 3D search spaces [29], and employing RVTs [30]. Fusing LiDAR and camera data has proven effective for semantic segmentation, using methods like mapping LiDAR points to the output of an image-based semantic segmentation network and inputting the data into a LiDAR detector [31], and addressing sparsity with cylindrical partitioning and asymmetrical 3D CNNs [32].

To our knowledge, no publicly available dataset contains data from an event camera, an RGB camera, a LiDAR, and an IMU, while providing ground truth poses of the sensor suite and dynamic objects. In this paper, we introduce a labelled dataset and propose a multi-modal perception pipeline for 3D object detection and spatial pose estimation that combines a 2D detection step (based on RGB or event vision) and a depth estimation step using LiDAR data. Fig. 1 shows our sensor and some data samples. We evaluate the performance using the event and RGB camera, highlighting the potential and challenges for future robotic guide-dog systems.

II. Dataset

A. Sensor suite and data collection

The dataset introduced in this paper is collected indoors with a handheld sensor suite moving in the field of view of a *Vicon* motion-capture system. Two people, also tracked by the motion-capture system, are moving in and out of the sensor suite field of view. The sensor suite consists of:

- **Inivation DAVIS346** event camera: *Stream of event* data (up to 1MHz) with each event being a tuple of x and y positions in the image space, t the timestamp, and p the polarity of the corresponding illumination change; *RGB images* at 23Hz; *6-DoF IMU* at 1kHz (3-axis gyroscope and 3-axis accelerometer).
- **Blickfeld Cube1** LiDAR: *3D point clouds* at 7.9Hz with point-wise timestamps.

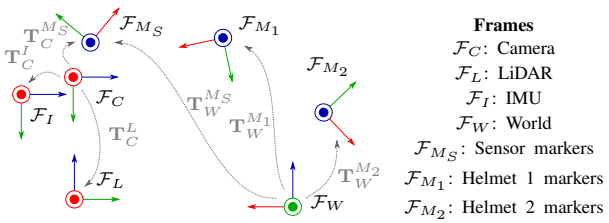


Fig. 2: Frames and geometric transformations in the dataset.

All the sensor's measurements and the output of the motion-capture system are recorded with ROS. We use the *rpg_dvs_ros* driver for the DVS camera and the Blickfeld ROS driver for the LiDAR. As illustrated in Fig. 1, the sensor suite is equipped with a set of reflective markers tracked by the *Vicon* system. Similarly, the people moving in the surroundings wear helmets with reflective markers. Subsequently, the *Vicon* system provides the 6-DoF pose of the 2 persons and sensor suite in an arbitrarily fixed reference frame. Overall, the dataset spans 14 minutes over four ROSBags, containing approximately 774 million events[1], 22000 RGB images, 6700 point clouds, and 70000 ground truth poses each for two persons in the scene. For the experimentation performed, the ROSBag entitled "dynamic.bag" was used. For convenience and utility, the dataset is labelled with the class identifier corresponding to the colour helmet worn by the person.

B. Calibration

To use our dataset effectively, we must first perform the intrinsic calibration of the camera and extrinsic calibration between the various sensors and the set of reflective markers. Fig. 2 shows the set of geometric transformations $\mathbf{T}_a^b$ estimated during calibration ($\mathbf{T}_C^L$, $\mathbf{T}_C^{M_S}$, and $\mathbf{T}_C^I$) or given by the Vicon system ($\mathbf{T}_W^{M_S}$, $\mathbf{T}_W^{M_1}$, and $\mathbf{T}_W^{M_2}$). Note that as the RGB and event data are being collected by the same cells in the *DAVIS346*, the reference frame of the event and RGB camera are collocated (labelled "Camera" in Fig. 2). Thus, the intrinsic calibration parameters obtained with the RGB camera apply to the event camera as both data types share the same optical path. Calibration sequences and parameter estimates are included with the main dataset. Table I details the estimation process for each transformation [2].

III. Multi-modal scene understanding

To enable downstream applications such as guide-dog-like assistive robots, we explore fusing vision and LiDAR data for

[1]The DVS driver cuts the event stream into variable-length event-array messages; thus, the dataset contains 25000 event-array messages.

[2]The camera's intrinsics are obtained with Matlab's calibration toolbox https://au.mathworks.com/help/vision/ref/cameracalibrator-app.html

TABLE I: Insight into the extrinsic calibration procedure of the sensor suite used to collect the proposed dataset.

Trans.	Details
$\mathbf{T}_C^{M_S}$	Camera position from checkerboard detection, then eye-in-hand calibration with Vicon poses of $\mathcal{F}_{M_S}$
$\mathbf{T}_C^L$	Checkerboard plane equation from camera and point-to-plane minimisation with LiDAR points on the checkerboard
$\mathbf{T}_C^I$	Kalibr[3]: Checkerboard for camera position and continuous-time batch state estimation

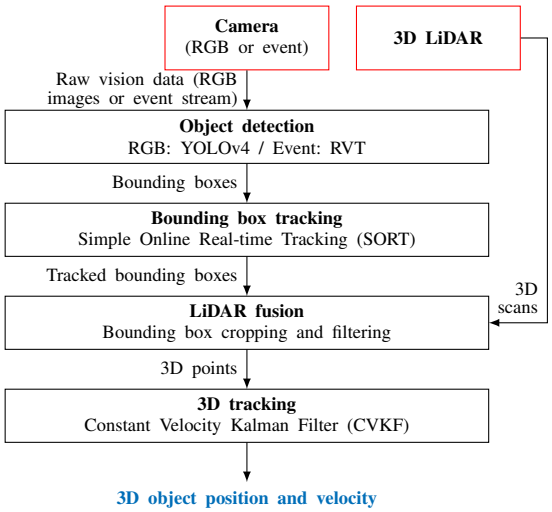


Fig. 3: Block diagram overview of the proposed vision-LiDAR object detection and tracking.

the 3D localisation of pedestrians and vehicles. The methodology can also be applied to static objects such as trees, buildings and roads. While a complete system should include these necessities, the motivation of this work is to focus on dynamic objects due to the further requirement of tracking relative motion. Fig. 3 presents an overview of the proposed pipeline. The main steps are, first, the vision-based detection of objects in the image space (2D), followed by the tracking of the resulting bounding boxes with the Simple Online and Real-time Tracking (SORT) algorithm [33]. Then, the bounding boxes are used to crop and filter the LiDAR scans before performing state estimation in the 3D space using a Constant-Velocity Kalman Filter (CVKF). Note that this pipeline can be used with an event camera or a standard RGB camera if the detection algorithm provides bounding boxes around the detected objects. The rest of this section provides details about the components of the proposed pipeline.

A. 2D object detection and tracking

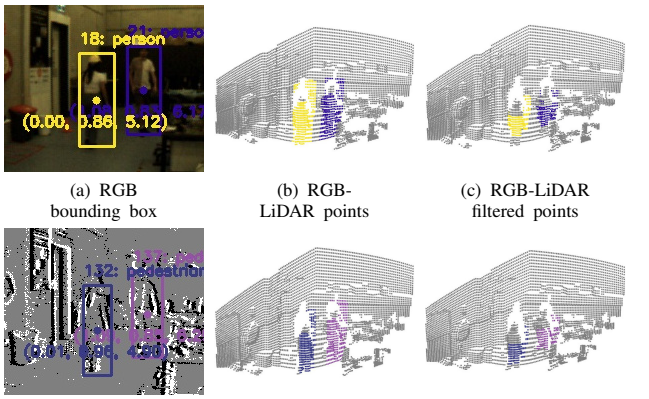
1) RGB-based detection: When the proposed framework is used with an RGB camera, we use YOLOv4 [15] for the task of object detection. YOLOv4 is a CNN-based algorithm renowned for its real-time capabilities and accuracy. It is trained on the MS-COCO dataset [10] and can classify 80 different types of objects, including pedestrians and vehicles. Its one-shot detection approach surpasses traditional two-shot detectors like Faster R-CNN in terms of inference speed. The RGB images from the DAVIS346 are undistorted using the camera's intrinsic parameters before being passed to YOLOv4. The output consists of 2D bounding boxes.

2) Event-based detection: For event-based vision, our pipeline relies on RVT [25]. The choice of RVT is motivated by its proficiency in detecting both vehicle and pedestrian data, coupled with its fast inference time relative to alternative event-based models. RVT relies on recurrent transformers to leverage the spatiotemporal nature of event data. Accordingly, the stream of events is preprocessed into a succession of 4-dimensional tensors of size $(2, T, h, w)$, with h and w the resolution of the camera, by binning the events into T temporal slices (10 slices within a 50 ms window in the publicly available model). The first dimension of the tensor represents the two polarities of the events, thus storing the events triggered by positive and negative changes separately. The authors of RVT have released pre-trained models *Gen1* and *1-Mpx* that are trained with the Gen1 [24] and 1-Mpx [23] automotive datasets, respectively. To infer objects' bounding boxes using the DAVIS346 data, we crop or pad the event tensors to fit the required input size of the models (h, w).

3) 2D tracking: The 2D bounding boxes in the image from the event and RGB object detectors are quite noisy in regards to the position and amount of misdetection. The proposed pipeline leverages the SORT algorithm for multi-frame object association and position estimation to address this issue. SORT employs the Hungarian Method in conjunction with a Linear CVKF to track objects across frames independently of other objects and camera motion. The state vector in the CVKF is $x = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}]^\top$, where u, v, $\dot{u}$ and $\dot{v}$ represent the centre coordinates and velocity in pixels/frame of the bounding box, s and $\dot{s}$ represent the bounding box area and change in area respectively, and r represents the aspect ratio of the bounding box, which is assumed to be constant. We apply small changes to parameters outlined in Table II to better handle occlusions and instances of missed subsequent associations to a tracker, which are dangers for a guide-dog-like aid.

B. 3D fusion

1) LiDAR scan filtering: Given stable 2D bounding boxes from the aforementioned vision-based detector-and-tracking step, we first select the LiDAR points that fall into a bounding box by projecting each point of a LiDAR scan into the image using $\mathbf{T}_C^L$ and the camera intrinsics as illustrated in Fig. 4 Unfortunately, the points associated with a bounding box do not only correspond to the detected object but also to the foreground and background. Accordingly, we propose a simple filtering method to only extract points belonging to the detected object. Based on the assumption that the centre of the detected object is roughly aligned with the centre of the bounding box, only points present in a square around the bounding box centre are considered. The ratio of the square's area to the bounding box's area is scaled linearly with the ratio of the bounding box's area to the image resolution. Therefore, as the bounding box gets smaller, the ratio of the square to bounding box area increases, and vice-versa. For the rest of the pipeline, the object position is represented



(a) RGB bounding box (b) RGB-LiDAR points (c) RGB-LiDAR filtered points

(d) Event bounding box (e) Event-LiDAR points (f) Event-LiDAR filtered points

Fig. 4: Bounding box point cloud segmentation and filtering examples via vision-based (RGB and event) object detection.

with a single 3D point. Accordingly, we use the median of the points inside the square to feed the tracker presented in the following subsection.

2) 3D tracking: Provided with the point representation from the LiDAR scan filtering and the bounding box tracking ID from Section III-A.3, the proposed pipeline initialises and maintains independent CVKF for each object track. Inspired by the work in [34], the CVKF state vector consists of the object position and velocity: $x_{3D} = [x, y, z, \dot{x}, \dot{y}, \dot{z}]^\top$.

IV. Experiments

A. Implementation

The quantitative results are obtained using the proposed dataset collected with an *Inivation DAVIS346* camera and a *Blickfeld Cube1* LiDAR. For the event-based detector, we empirically chose the 1-Mpx model of RVT [25] after testing Gen1 and 1-Mpx using our dataset. No significant performance difference was found. Both RGB-based and event-based object detectors were used without any retraining or fine-tuning of the networks' weights. Table II shows the parameters of the SORT algorithm for vision-based tracking (Section III-A.3). The RGB and event detectors have different noise characteristics, so SORT tracker parameters differ slightly. Experiments were conducted on a low-performance laptop with Ubuntu 20.04.6 LTS, an NVIDIA GTX GeForce 1650 GPU, an AMD Ryzen 7 5700U CPU, and 16GB of RAM. The proposed pipeline runs close to real-time with both RGB-based and event-based detection.

TABLE II: The SORT algorithm parameters for image space object tracking.

Parameter	RGB	Event
Maximum age of unmatched tracker [no. of frames]	10	10
Maximum unmatched predictions [no. frames]	5	3
Min. number of associated detections for tracking	3	1
Min. number of previous associations for prediction	10	1
IoU threshold for association	0.3	0.3

B. Vision-based object detection

To evaluate the performance of the vision-based object detectors in our pipeline for assistive robotics in dynamic settings, we performed object detection with both YOLOv4 and RVT using the proposed dataset. We only consider YOLOv4 detections above a confidence score of 0.5 while varying the RVT confidence threshold across evaluations.

Each RGB frame is associated with a 50 ms event tensor/sequence required for RVT's prediction. Table III displays RVT's precision and recall (confidence threshold of 0.3), with and without the tracking, using YOLOv4 as the ground truth due to its proven accuracy [15]. The definitions of true/false positive/negative for precision and recall are based on IoU thresholds between the bounding boxes of both methods. Table III displays results for varying IoU thresholds, while Table IV shows results for a fixed IoU threshold with varying RVT confidence thresholds.

The results show that YOLOv4 outperforms the RVT model. The precision scores suggest a higher number of misclassifications with erroneous class attribution. Interestingly, using SORT increased the recall but decreased the precision. This suggests that when true positive detections occur in one sequence of events but not in the ensuing sequences, the SORT algorithm improves the detection rate due to its ability to predict the subsequent positions of an object, reducing the number of false negatives. However, when the tracker incorrectly estimates the object dynamics or false positive detections occur, the precision score decreases.

C. 3D object tracking

1) Quantitative: Using the proposed detection pipeline and dataset, we evaluate the overall accuracy using the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE) between the predicted object 3D position in the camera reference frame and the ground truth value from the motion-capture system $\mathbf{T}_W^{M_\bullet}$, $\mathbf{T}_W^{M_S}$, and the calibration $\mathbf{T}_C^{M_S}$.

Table V shows the results from both detection methods. To achieve an unbiased evaluation of the 3D estimation framework, the event-based prediction uses a manually selected one-minute portion of the "dynamic.bag" ROSBag where the RVT detector performs well, while the RGB-based pipeline uses the full ROSBag. The ground truth for people's positions is at head level, and the LiDAR filtering focuses on the hip level. Metrics are separated by the individual axis and the XZ plane. The Y-axis (gravity-aligned axis) error of just under a metre reflects head-vs-hip tracking. Overall, both modalities result in a range of approximately 0.8 to 1 m MAE in the XZ plane, validating the proposed detection/tracking pipeline. The larger Z-axis error (depth) compared to the X-axis indicates that scan filtering does not fully isolate the object from the foreground and background, as seen in Figure 4 where the spread of 3D point cloud data contained inside a 2D bounding box is broader along the depth axis.

Given that the RMSE weights heavily outlier errors, the difference between MAE and RMSE suggests that a few tracking results are highly inaccurate, while a majority are accurate. Curiously, the CVKF does not enhance but rather worsens the final estimates. Thanks to the vision-based SORT tracking, the output of the LiDAR filtering step is already smooth. Thus, the inherent delay of the final

TABLE III: Evaluation of event-based detection vs. YOLOv4: varying IoU thresholds @ confidence threshold = 0.3.

		Event-Based Detection: IoU thresholds @ Confidence threshold = 0.3								
		0.5	**0.55**	**0.6**	**0.65**	**0.7**	**0.75**	**0.8**	**0.85**	**0.9**
Pure Detection	**Precision**	0.625	0.573	0.516	0.447	0.357	0.256	0.168	0.082	0.026
	Recall	0.423	0.388	0.349	0.303	0.242	0.174	0.114	0.055	0.017
Tracked w/ SORT	**Precision**	0.577	0.533	0.478	0.409	0.328	0.236	0.146	0.068	0.02
	Recall	0.463	0.427	0.383	0.328	0.263	0.189	0.117	0.055	0.016

TABLE IV: Evaluation of event-based detection vs. YOLOv4: varying confidence thresholds @ IoU threshold = 0.5.

		Event-Based Detection: Confidence thresholds @ IoU threshold = 0.5												
		0.3	**0.35**	**0.4**	**0.45**	**0.5**	**0.55**	**0.6**	**0.65**	**0.7**	**0.75**	**0.8**	**0.85**	**0.9**
Pure Detection	**Precision**	0.577	0.658	0.688	0.72	0.756	0.787	0.818	0.849	0.885	0.916	0.956	0.996	1.0
	Recall	0.463	0.414	0.404	0.394	0.383	0.37	0.352	0.327	0.294	0.239	0.156	0.044	0.001
Tracked w/ SORT	**Precision**	0.625	0.611	0.643	0.681	0.718	0.747	0.784	0.817	0.844	0.857	0.878	0.88	1.0
	Recall	0.423	0.45	0.441	0.432	0.423	0.408	0.391	0.368	0.33	0.274	0.19	0.055	0.001

TABLE V: Accuracy analysis of 3D object position estimation.

		RGB detection		**Event** detection*	
		MAE	**RMSE**	**MAE**	**RMSE**
Filtering only	**X**	0.283	0.513	0.232	0.41
	Y	0.765	0.79	0.86	0.888
	Z	0.724	1.397	0.929	1.613
	XZ	0.828	1.488	0.983	1.665
Filtering and CVKF	**X**	0.281	0.484	0.243	0.434
	Y	0.763	0.788	0.866	0.894
	Z	0.727	1.712	0.98	1.663
	XZ	0.831	1.779	1.033	1.719

* The event evaluation uses only one minute of the dataset

(a) Moving bus (b) Parked cars

Fig. 5: Detection samples and estimated dynamics in an urban environment.

CVKF's estimate with respect to the true state value can only result in lesser accuracy. Additionally, occlusions and missed associated detections handled by the 2D SORT algorithm lead to the wrong selection of points in the LiDAR scans - leading to high geometric errors. This correlates with the disparity between MAE and RMSE.

2) Qualitative: To test and demonstrate the ability of the proposed pipeline to provide spatial awareness for assistive robots such as guide-dog-like aid, a sensor suite consisting of an Intel RealSense camera and Velodyne VLP-16 LiDAR was utilised in an urban environment. This data was collected to inspect the effectiveness of the RGB version of our pipeline on longer-range detections and the ability to estimate vehicle dynamics. While no ground truth is available for quantitative evaluation, Fig.5 shows multiple detections of pedestrians and vehicles and their estimated velocity. These correspond to the expected velocities of the difference agents.

V. Discussion

In the LiDAR filtering step, assuming the "central square" cropping of 3D points aligns the object's centre with the bounding box centre is arbitrary and often untrue. The central part of the bounding box may correspond to background information. For instance, when a person extends an arm, the bounding box expands, which shifts the centre away from the torso, leading to an incorrect point selection. Similarly, for vehicles, the bounding box centre may align with the windshield, causing the LiDAR to observe the background. Future work will explore using efficient per-pixel semantic labels to better handle partial occlusions.

Our pipeline also assumes constant-velocity models in different trackers. While effective for wheel-based systems like autonomous vehicles, these models fall short for handheld devices, such as in our dataset, and platforms with jerky motion, such as bipedal and quadrupedal robots. In scenarios mimicking guide dogs, inaccurate vehicle tracking can have severe consequences. Investigating varied motion models and incorporating the robot's movement commands might enhance system robustness. Furthermore, employing trackers like DeepSORT [35], which utilise metrics other than IoU, can strengthen frame-to-frame association and tracking.

Expanding into 3D detection improves reliability by merging and cross-checking outputs from multiple modalities, beyond using LiDAR for depth. However, LiDAR-only detection faces limitations like vertical sparseness and motion distortion. Sensor fusion is crucial for robust robotic autonomy. Future research should refine detection synchronisation, moving beyond timestamp-based LiDAR scan matching.

Our findings indicate that event-based object detectors lack adaptability and generalisation, while frame-based detectors are ready for use without retraining, thanks to large, diverse publicly available training datasets. The lack of diverse event camera training data hinders adaptability, as event cameras are more affected by camera motion. Our dataset will enable the robotics community to investigate these issues, paving the way for safer and more robust algorithms.

VI. Conclusion

This paper has presented a dataset for comparing event-based and RGB-based multi-modal 3D object detection and tracking with LiDAR data. The dataset includes RGB, event, LiDAR, and inertial data, along with human ground-truth positions determined by a motion-capture system, addressing a gap in publicly available datasets for applications such as

guide-dog-like assistive robots. We proposed a pipeline for dynamic object detection and tracking that performs vision-based object detection followed by LiDAR-based 3D position estimation. Our experiments show that frame-based detection algorithms generalise well to various scenes, while the current state-of-the-art event models are limited to smaller, automotive-oriented scenarios.

Future work will enrich our dataset with data from various mobile platforms (wheeled, bipedal, and quadrupedal). This is important due to the spatiotemporal nature of the event data: regular movements lead to recurrent patterns in the event stream. For the proposed pipeline, our efforts will focus on refining 3D localisation and tracking to better adapt to rapid dynamic changes and employing advanced machine learning techniques for more accurate object isolation. Ultimately, we will integrate the proposed perception framework into an advanced assistive robot to help vision-impaired users navigate challenging environments safely.

REFERENCES

[1] Guide Dogs Victoria, "Facts + stats," https://vic.guidedogs.com.au/about-gdv/fact-sheet/, 2023, accessed: 2023-11-18.

[2] L. M. Tomkins, P. C. Thomson, and P. D. McGreevy, "Associations between motor, sensory and structural lateralisation and guide dog success," *The Veterinary Journal*, vol. 192, no. 3, pp. 359–367, 2012.

[3] B. Hong, Z. Lin, X. Chen, J. Hou, S. Lv, and Z. Gao, "Development and application of key technologies for guide dog robot: A systematic literature review," *Robotics and Autonomous Systems*, vol. 154, p. 104104, 2022.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[6] G. Paul, L. Liu, and D. Liu, "A novel approach to steel rivet detection in poorly illuminated steel structural environments," in *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2016, pp. 1–7.

[7] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128× 128 120 db 15 μs latency asynchronous temporal contrast vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, 2008.

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conf. on computer vision and pattern recognition*, 2016, pp. 779–788.

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[10] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.

[11] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov *et al.*, "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale," *International journal of computer vision*, vol. 128, no. 7, pp. 1956–1981, 2020.

[12] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, pp. 303–338, 2010.

[13] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[14] R. Girshick, "Fast r-cnn," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448.

[15] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," 2020. [Online]. Available: https://arxiv.org/abs/2004.10934

[16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision–ECCV 2016*. Springer, 2016, pp. 21–37.

[17] R. J. Wang, X. Li, and C. X. Ling, "Pelee: A real-time object detection system on mobile devices," *Advances in neural information processing systems*, vol. 31, 2018.

[18] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3150–3158.

[19] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.

[20] D. Gehrig, A. Loquercio, K. Derpanis, and D. Scaramuzza, "End-to-end learning of representations for asynchronous event-based data," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 5632–5642.

[21] M. Gehrig, S. B. Shrestha, D. Mouritzen, and D. Scaramuzza, "Event-based angular velocity regression with spiking networks," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 4195–4202.

[22] S. Schaefer, D. Gehrig, and D. Scaramuzza, "Aegnn: Asynchronous event-based graph neural networks," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 12 361–12 371.

[23] E. Perot, P. De Tournemire, D. Nitti, J. Masci, and A. Sironi, "Learning to detect objects with a 1 megapixel event camera," *Advances in Neural Information Processing Systems*, vol. 33, pp. 16 639–16 652, 2020.

[24] P. De Tournemire, D. Nitti, E. Perot, D. Migliore, and A. Sironi, "A large scale event-based detection dataset for automotive," *arXiv preprint arXiv:2001.08499*, 2020.

[25] M. Gehrig and D. Scaramuzza, "Recurrent vision transformers for object detection with event cameras," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, jun 2023, pp. 13 884–13 893.

[26] L. Zhao, H. Zhou, X. Zhu, X. Song, H. Li, and W. Tao, "Lif-seg: Lidar and camera image fusion for 3d lidar semantic segmentation," *IEEE Transactions on Multimedia*, vol. 26, pp. 1158–1168, 2024.

[27] B. Dai, C. Le Gentil, and T. Vidal-Calleja, "Connecting the dots for real-time lidar-based object detection with yolo," in *Australasian Conference on Robotics and Automation, ACRA*, 2018.

[28] C. Lin, D. Tian, X. Duan, J. Zhou, D. Zhao, and D. Cao, "Cl3d: Camera-lidar 3d object detection with point feature enhancement and point-guided fusion," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 18 040–18 050, 2022.

[29] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3d object detection from rgb-d data," in *2018 IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2018, pp. 918–927.

[30] X. Bai, Z. Hu, X. Zhu, Q. Huang, Y. Chen, H. Fu, and C.-L. Tai, "Transfusion: Robust lidar-camera fusion for 3d object detection with transformers," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 1080–1089.

[31] S. Vora, A. H. Lang, B. Helou, and O. Beijbom, "Pointpainting: Sequential fusion for 3d object detection," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 4603–4611.

[32] X. Zhu, H. Zhou, T. Wang, F. Hong, W. Li, Y. Ma, H. Li, R. Yang, and D. Lin, "Cylindrical and asymmetrical 3d convolution networks for lidar-based perception," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 6807–6822, 2022.

[33] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 3464–3468.

[34] X. Weng, J. Wang, D. Held, and K. Kitani, "3D Multi-Object Tracking: A Baseline and New Evaluation Metrics," *IROS*, 2020.

[35] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," 2017. [Online]. Available: https://arxiv.org/abs/1703.07402