

# Fusing Event-based Camera and Radar for SLAM Using Spiking Neural Networks with Continual STDP Learning

Ali Safa<sup>1,4</sup>, Tim Verbelen<sup>2,4</sup>, Ilja Ocket<sup>4</sup>, André Bourdoux<sup>4</sup>, Hichem Sahli<sup>3,4</sup>,  
 Francky Catthoor<sup>1,4</sup>, Georges Gielen<sup>1,4</sup>

**Abstract**—This work proposes a first-of-its-kind SLAM architecture fusing an event-based camera and a Frequency Modulated Continuous Wave (FMCW) radar for drone navigation. Each sensor is processed by a bio-inspired Spiking Neural Network (SNN) with continual Spike-Timing-Dependent Plasticity (STDP) learning, as observed in the brain. In contrast to most learning-based SLAM systems, our method does not require any offline training phase, but rather the SNN continuously learns features from the input data on the fly via STDP. At the same time, the SNN outputs are used as feature descriptors for loop closure detection and map correction. We conduct numerous experiments to benchmark our system against state-of-the-art RGB methods and we demonstrate the robustness of our DVS-Radar SLAM approach under strong lighting variations.

## MULTIMEDIA MATERIAL

Please watch a demo video of our SNN-based DVS-Radar fusion SLAM at <https://youtu.be/a7gvZWNHGoI>

## I. INTRODUCTION

Simultaneous Localisation and Mapping (SLAM) is an important problem for autonomous agents such as drones [1], [2]. Most state-of-the-art SLAM systems integrate raw odometry data with feature matching using standard RGB cameras in order to detect *loop closures* (i.e., places already visited by the agent) and to correct the drift in raw odometry accordingly [3]. However, using RGB cameras alone does not provide upmost robustness, such as insensitivity to lighting variations and environmental conditions [1]. Therefore, multi-sensor SLAM systems fusing RGB with e.g., lidar, radar and event-based cameras have emerged with increased robustness compared to RGB alone [2], [4].

Event-based cameras (also called *Dynamic Vision Sensors* or DVS) are a novel type of imaging sensor composed of independent pixels  $x_{ij}$  that asynchronously emit spikes whenever the change in light log-intensity  $|\Delta L_{ij}|$  sensed by the pixel crosses a certain threshold  $C$  [5] (see Fig. 1). In contrast to RGB cameras, DVS cameras can still perform well in low-light conditions [1] and produce a *spatio-temporal* stream that contains patterns in both the spatial and spike timing dimensions, making them a natural

This research has received funding from the Flemish Government (AI Research Program) and the European Union’s ECSEL Joint Undertaking under grant agreement n° 826655 - project TEMPO.

<sup>1</sup> Faculty of Electrical Engineering (ESAT) KU Leuven, 3001, Belgium

<sup>2</sup> IDLab, Gent University, B-9052 Gent, Belgium

<sup>3</sup> ETRO, VUB, 1050 Brussels, Belgium

<sup>4</sup> imec, Kapeldreef 75, 3001, Leuven, Belgium {Ali.Safa, Tim.Verbelen, Ilja.Ocket, Andre.Bourdoux, Hichem.Sahli, Francky.Catthoor}@imec.be, Georges.Gielen@kuleuven.be

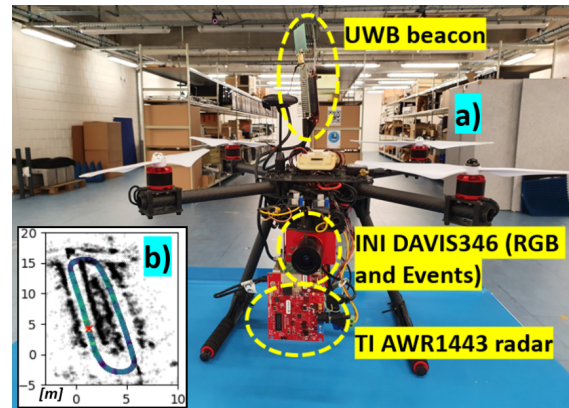


Fig. 2: **a)** Our sensor fusion drone setup used for jointly acquiring event camera and radar data. The warehouse environment (detailed in [6]) in which the drone navigates is also shown. The warehouse is equipped with Ultra Wide Band (UWB) localisation for ground truth acquisition [6]. **b)** Event and radar data are fed to SNNs that continually learn on the fly via STDP (see Fig. 4). The SNN output is used for loop closure detection to perform SLAM. Radar detections are also used to model the obstacles (black dots in b).

choice as input for *Spiking Neural Networks* (SNNs). Indeed, SNNs are bio-plausible neural networks that make use of spiking neurons, communicating via binary activations in an event-driven manner, matching the DVS principle [7].

SNNs have recently gained huge attention for ultra-low-energy and -area processing in power-constrained edge de-

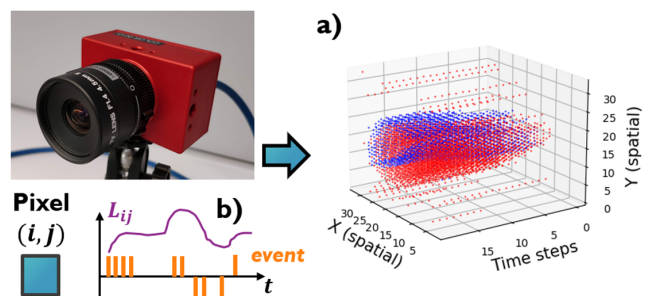


Fig. 1: **Conceptual illustration of event-based vision.** **a)** The camera outputs a stream of spikes in both space and time. **b)** Each pixel  $x_{ij}$  fires a spike when its change in light log-intensity  $\Delta L_{ij}$  crosses a threshold. The spike is positive when  $\Delta L_{ij} > 0$  (red dots) and negative (blue dots) otherwise.

vices such as drones [8]. In addition, the use of event-driven learning rules such as Spike-Timing-Dependent Plasticity (STDP) [9] enables *unsupervised* learning at the edge using emerging *sub-milliwatt* neuromorphic chips [10], [11], as opposed to power-hungry GPUs. Furthermore, SNN-STDP systems are also a good choice for continual, *on-line* learning [12], dropping the need for offline SNN training. Hence, this work studies the use of SNNs equipped with STDP learning for fusing DVS and radar data on power-constrained drones.

The use of radar sensing for drone navigation has recently been investigated in a growing number of works [13], [14], [15], [16], [17]. Indeed, fusing radar sensing with vision-based sensors such as DVS is attractive since radars intrinsically provide complementary information to camera vision, such as target velocity and position in a range-azimuth map (also called *bird's eye view* vs. projective plane in cameras) [17]. In addition, radars are robust to environmental conditions as they are not sensitive to occlusion by dirt and can sense in the dark [14].

This work deviates from most fusion-based SLAMs by proposing a *first-of-its-kind*, bio-inspired SLAM system fusing event camera with radar (see Fig. 2, 4), using SNNs that *continuously* learn via STDP, as observed in the brain [9]. It also deviates from most learning-based SLAM systems which typically require the *offline* training of a Deep Neural Network (DNN) on a dataset of the working environment captured beforehand [2]. In contrast, our DVS-Radar fusion SNN learns on the fly and keeps adapting its weights via *unsupervised* STDP as the drone explores the environment. At the same time, the SNN outputs are fed to a bio-inspired RatSLAM back-end [18] for loop closure detection and map correction. Crucially, our continual STDP learning approach enables the deployment of our system in environments *not* captured by datasets and therefore, not known *a priori* (vs. offline training in state-of-the-art DNN-based SLAMs [2]).

We use our sensor fusion drone shown in Fig. 2 to jointly acquire DVS and radar data during multiple drone flights in a challenging, indoor environment, in order to perform SLAM (see Fig. 2 b). We assess the performance of our proposed DVS-Radar SNN-STDP SLAM system against ground truth positioning, recorded via *Ultra Wide Band* (UWB) beacons [6]. The main contributions of this paper are the following:

- 1) We propose what is, to the best of our knowledge, the first continual-learning SLAM system which fuses an event-based camera and an FMCW radar using SNNs.
- 2) We propose a method for radar-gyroscope odometry, where radar sensing provides the drone's velocity, and a method for obstacle modelling via radar detections.
- 3) We experimentally assess the performance of our SLAM system on three different flight sequences in a challenging warehouse environment and we show the robustness of our system to strong lighting variations.

This paper is organized as follows. Related works are discussed in Section II, followed by background theory covered in Section III. Our proposed methods are presented in Section IV. Experimental results are shown in Section V. Conclusions are provided in Section VI.

## II. RELATED WORKS

A growing number of bio-inspired [18], [19], [20] and sensor fusion [1], [2], [4] SLAM systems have been proposed in recent years. Among the most related to this work, a DVS-RGB SLAM system has been proposed in [1], providing robust state estimations by fusing event-based cameras, RGB and raw IMU odometry. In addition, the system of [1] has been implemented on a drone for indoor navigation and was shown to be robust to drastic changes in lighting conditions and in low-light scenarios. In contrast to [1], which makes use of hand-crafted features, our system uses an SNN with STDP learning. In addition, we do not fuse the richer information obtained from RGB as in [1], but rather fuse DVS with *sparser* radar detections instead, which makes the reliable template matching of the SNN outputs challenging.

Recently, the *LatentSLAM* system has been proposed in [2] as a learning-based pipeline using a DNN encoder which provides *latent codes* for template matching, trained *offline* on a dataset capturing the environment in which the robot must navigate. The inferred latent codes are fed to the loop closure detection and map correction back-end of the popular *RatSLAM* system [18] to correct the drift in raw odometry. In contrast, our proposed continual learning system *does not* require any offline training phase, enabling its deployment in *unseen environments* without the requirement of capturing a dataset of the working environment beforehand.

As stated earlier, we make use of the RatSLAM loop closure detection and map correction back-end in this work [18], [21]. RatSLAM has been proposed as a bio-inspired system following the navigational processes of the rat's hippocampus [18]. Even though the original RatSLAM uses *raw RGB* images for template matching, further evolutions of RatSLAM, such as LatentSLAM, replace the raw RGB input by the stream of associated latent codes obtained through learned feature extraction [2]. In this work, we feed both our proposed *radar-gyroscope* odometry and the latent codes inferred by our proposed continual learning SNN-STDP fusion system to the RatSLAM back-end.

Since RGB-based SLAMs constitute today's state of the art, we will benchmark our DVS-Radar SLAM against both LatentSLAM and RatSLAM, and against ORB features [22] extensively used in state-of-the-art SLAM systems [3], [23].

## III. SNN BACKGROUND THEORY

Unlike frame-based DNNs, SNNs make use of *event-driven* spiking neurons as activation function, often modelled using the Leaky Integrate and Fire (LIF) neurons [24]:

$$\begin{cases} \frac{dV}{dt} = \frac{1}{\tau_m}(J_{in} - V) & \text{with } J_{in} = \bar{w}_{syn}^T \bar{s}(t) \\ \sigma = 1, V \leftarrow 0 & \text{if } V \geq \mu, \text{ else } \sigma = 0 \end{cases} \quad (1)$$

where  $\sigma$  is the spiking output,  $V$  the membrane potential,  $\tau_m$  the membrane time constant,  $\mu$  the neuron threshold and  $J_{in} = \bar{w}_{syn}^T \bar{s}(t)$  the input to the neuron, resulting from the inner product between the neuron weights  $\bar{w}_{syn}$  and the *spiking* input vector  $\bar{s}(t)$ . The LIF continuously integrates its input  $J_{in}$  in  $V$  following (1). When  $V$  crosses the firing

threshold  $\mu$ , the membrane potential is reset back to zero and an output spike  $\sigma = 1$  is emitted. SNNs are therefore a natural choice for processing event-driven sensors such as DVS, by *flattening* along space the spiking image in Fig. 1 into a vector of spike trains  $\bar{s}(t)$ .

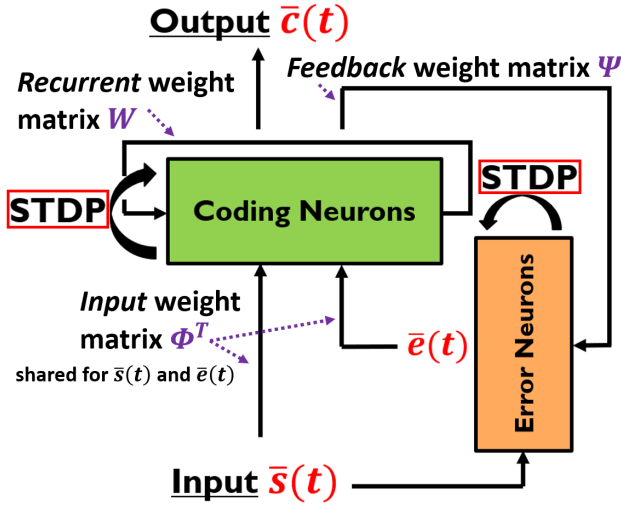


Fig. 3: **SNN-STDP architecture.** The network is composed of two distinct ensembles of LIF neurons ( $M$  coding neurons and  $N$  error neurons). The input spikes  $\bar{s}(t)$  are fed to the coding neurons via the fully-connected weights  $\Phi^T$ . The coding neurons recurrently infer the SNN output spikes  $\bar{c}(t) \leftarrow \text{LIF}\{W\bar{c}(t) + \Phi^T\bar{s}(t)\}$  by integrating  $\bar{s}(t)$  with the past outputs  $\bar{c}(t)$  through recurrent weights  $W$ . The error neurons receive  $\bar{c}(t)$  via weights  $\Psi$  and infer the error spikes  $\bar{e}(t) \leftarrow \text{LIF}\{\Psi\bar{c}(t) - \bar{s}(t)\}$  used during STDP learning. STDP (3)-(4) is locally applied to weights  $\Phi, W, \Psi$  following [25].

In order to detect loop closures and perform map correction, learning-based SLAMs seek to infer lower-dimensional latent codes  $\bar{c}$  from the sensory observation [2]. In this work, we use the SNN in Fig. 3, proposed in [25], as our baseline architecture to continuously infer *spiking* latent codes  $\bar{c}(t)$  from the input  $\bar{s}(t)$ , while jointly *learning* features  $\Phi$  from the data in an *unsupervised* way. This can be formulated as:

$$\bar{c}, \Phi = \arg \min_{\bar{c}, \Phi} \|\Phi\bar{c} - \bar{s}\|_2^2 + \lambda_1 \|\bar{c}\|_1 + \frac{\lambda_2}{2} \|\Phi\|_F^2 \quad (2)$$

with  $\bar{s}$  and  $\bar{c}$  the  $N$ -dimensional input and  $M$ -dimensional output vectors, with their elements representing the *average spike rates* of the corresponding input and output spike train vectors  $\bar{s}(t)$  and  $\bar{c}(t)$ .  $\Phi$  is the transposed SNN input weight matrix,  $\lambda_1$  is a sparsity-controlling parameter (defined by the LIF threshold  $\mu$  [25]) and  $\lambda_2$  is a weight decay parameter. The term  $\|\Phi\bar{c} - \bar{s}\|_2^2$  in (2) minimizes the re-projection error between the output latent code  $\bar{c}$  and the input  $\bar{s}$  through the weights  $\Phi$  *without supervision* (similar to an auto-encoder).

It can be shown [25] that the SNN in Fig. 3 solves (2) by *jointly* inferring  $\bar{c}(t)$  while *continuously* adapting its weights  $\Phi, W, \Psi$  via STDP learning (3)-(4) [25]. In (3) and (4),  $\eta_d$  is the learning rate,  $A_p, A_n$  the long-term potentiation (LTP) and depression (LTD) weights,  $\tau_p, \tau_n$  the potentiation and

depression decay constants,  $w_{syn,i,j}$  the  $j^{th}$  weight of neuron  $i$  and  $\tau_{ij}$  the time difference between post- and pre-synaptic spike times across the  $j^{th}$  synapse of neuron  $i$ .

$$w_{syn,i,j} \leftarrow w_{syn,i,j} + \eta_d \kappa(\tau_{ij}) \quad (3)$$

$$\kappa(\tau_{ij}) = \begin{cases} A_p e^{-\tau_{ij}/\tau_p}, & \text{if } \tau_{ij} \geq 0 \\ -A_n e^{\tau_{ij}/\tau_n}, & \text{if } \tau_{ij} < 0 \end{cases} \quad (4)$$

Eq. (2) is never solved explicitly, but rather solved *implicitly* by the SNN-STDP of Fig. 3 as it iterates. Each weight learns in a local manner, avoiding weight transport problems [26].

In Section IV-A, we use the SNN in Fig. 3 to build a DVS-Radar fusion architecture that outputs latent codes while continuously learning on the fly *without* any pre-training.

#### IV. PROPOSED METHOD

##### A. DVS-Radar fusion architecture using SNN-STDP

In order to jointly infer latent codes and continually learn a set of SNN weights capturing the DVS and radar data features, we use the unsupervised SNN-STDP ensemble of Fig. 3 (see Section III) in a *fusion* setting, by assigning: *i*) a first SNN-STDP ensemble to the *positive* polarity of the DVS; *ii*) a second SNN-STDP ensemble to the *negative* polarity of the DVS, and *iii*) a third SNN-STDP ensemble to the radar detection output (see Fig. 4).

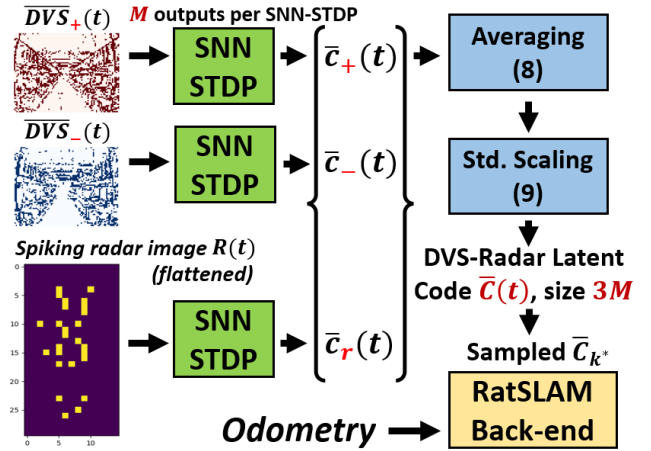


Fig. 4: **DVS-Radar SLAM using SNN-STDP ensembles.** The spiking DVS image is flattened as a vector and each event polarity (+ or -) is fed to its corresponding SNN-STDP network. Radar detections are transformed into a spiking image using (6,7) and flattened before being fed as a spiking vector to the corresponding SNN-STDP. The outputs  $\bar{c}_{+,-,r}$  of each SNN are concatenated, then averaged on a time window (8) and scaled (9) to get  $\bar{c}(t)$ . For each time-step  $k^*$ , the latent codes  $\bar{c}_{k^*}$  and the odometry are fed to a RatSLAM back-end [18] for loop closure detection and map correction.

In our experiments, we degrade the time resolution of the DVS camera to spiking bins of  $\delta T_{DVS} = 5\text{ms}$  and the DVS resolution to  $65 \times 86$ , as a good balance between computational time and performance. Each polarity channel of the DVS data is then directly connected to its respective



Therefore, each time a loop closure is detected (see Section IV-A), our pipeline refines not only the past poses  $\{X_k, Y_k, \Psi_k\}$ , but also the obstacle coordinates  $[\tilde{x}_{d,i}, \tilde{y}_{d,i}]$  by applying (11) again with the corrected  $\Psi_k$  and  $[X_k, Y_k]$ . Our obstacle aggregation is shown in both Fig. 2 b) and Fig. 8, where walls and obstacles can clearly be seen.

## V. EXPERIMENTAL RESULTS

We will now compare the *mean absolute error* (MAE) of our method against state-of-the-art RGB-based solutions on three different flight sequences (Seq.1-2-3 in Fig. 5). In addition, we test our DVS-Radar system under lighting variations (Seq.3 in Fig. 5), by randomly switching the lights on and off. Table I reports the SNN-STDP parameter values used in our experiments, tuned manually for optimising the SLAM performance, starting from the values in [25].

$\eta_d$	$\eta_c$	$M$	$N_{+,-}$	$N_r$	$\mu_{+,-}$	$\mu_r$
$3 \times 10^{-4}$	1	64	$65 \times 86$	$15 \times 30$	0.3	0.03
$\tau_m^{+,-}$	$\tau_m^r$	$A_p$	$A_n$	$\tau_p$	$\tau_n$	
0.017 s	0.17 s	1	0.8	0.0208 s	0.008 s	

TABLE I: **SNN-STDP parameters.**  $\eta_d$  is the STDP learning rate.  $\eta_c$  is the SNN coding rate [25]. The number of coding neurons  $M$  is the same for all SNN ensembles in Fig. 4. The number of error neuron  $N_{+,-,r}$  is equal to the dimensions of the input image to each SNN, where  $+, -, r$  respectively denote the SNN ensemble for the positive DVS, the negative DVS and the radar data in Fig. 4.  $\mu_{+,-,r}$  and  $\tau_m^{+,-,r}$  are the neuron thresholds and membrane constants in (1) for the respective SNNs.  $A_{p,n}, \tau_{p,n}$  are the STDP parameters in (4).

Data is acquired in a challenging warehouse environment composed of storage aisles, with high visual ambiguities between the scenes. This makes the disambiguation of the drone location hard due to the redundancy between the different views [2], [29] (see Fig. 5). Our warehouse [6] is also

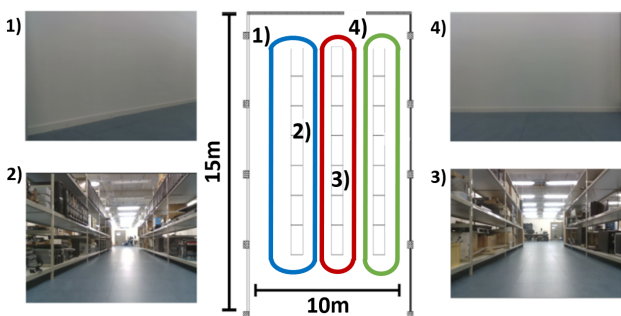


Fig. 5: **Flight data used in this work.** The drone location is hard to disambiguate visually (e.g., view 1 vs. 4, 2 vs. 3). Data were acquired during three different flight sequences: seq.1: the red path; seq.2: a combination of all paths, and seq.3: the blue and red path with strong lighting variations.

equipped with *Ultra Wide Band* (UWB) beacons for ground truth positioning, noted  $(x_k^{\text{gt}}, y_k^{\text{gt}})$  with  $k$  the time index, *only* used for error assessment. When computing the error, since the ground truth coordinates and the SLAM coordinates are

not in the same coordinate system [18], the ground truth coordinates are *translated* and *rotated* in order to match the SLAM coordinate system. We identify the rotation angle and translation vector by minimizing for the drone *localisation* MAE (identified via grid search). This *localisation* MAE is obtained as the average error between the ground truth  $(x_k^{\text{gt}}, y_k^{\text{gt}})$  and the SLAM localisation  $(x_k, y_k)$  on the complete flight sequence (with length  $T_{\text{end}}$ ):

$$\text{MAE}_L = \frac{1}{T_{\text{end}}} \sum_{k=1}^{T_{\text{end}}} (|x_k - x_k^{\text{gt}}| + |y_k - y_k^{\text{gt}}|) \quad (12)$$

In addition, the *mapping* MAE is obtained *a posteriori*, as the deviation between *all* ground truth points and the map obtained at the end of the SLAM process:

$$\text{MAE}_M = \frac{1}{T_{\text{end}}} \sum_{k=1}^{T_{\text{end}}} \min_j \{|x_k - x_j^{\text{gt}}| + |y_k - y_j^{\text{gt}}|\} \quad (13)$$

### A. Normal lighting conditions

Table II reports the  $\text{MAE}_L$  and  $\text{MAE}_M$  of our approach against state-of-the-art solutions that make use of standard RGB camera data: *i*) the original RatSLAM [18]; *ii*) the *LatentSLAM*, trained specifically for our environment following [2], on a set of 3998 frames acquired *independently* from Seq-1-2-3; and *iii*) ORB features for template matching [22] using *Lowe's Ratio Test* [30]. Table II also reports ablation results when using only one modality alone.

For LatentSLAM and our method, the similarity between two frames  $i, j$  is computed as  $s_{ij} = \frac{\bar{c}_i^T \bar{c}_j}{\|\bar{c}_i\|_2 \|\bar{c}_j\|_2}$  where  $\bar{c}_{i,j}$  is the latent code (8, 9) obtained by each method.

For ORB feature matching,  $s_{ij}$  is computed as  $s_{ij} = 1 - N_m / \max_m$  with  $N_m$  the number of matches and  $\max_m$  the maximum number of matches between the two frames  $i, j$ .

The RatSLAM back-end integrates the odometry of Section IV-B and detects *loop closures* by sampling the latent codes  $\bar{c}_i$  at 100ms intervals and by testing the similarities  $s_{ij}, \forall j$  against a threshold  $\theta$  tuned for minimum MAE ( $\theta = 0.65$  for Radar-only,  $\theta = 0.6$  for DVS-only and DVS-Radar).

Architecture	Seq-1 (L)	Seq-1 (M)	Seq-2 (L)	Seq-2 (M)
RatSLAM [18]	0.72	0.22	2.49	0.57
LatentSLAM [2]	0.54	0.21	1.68	0.39
ORB features [22]	0.46	0.17	0.95	0.39
<b>Ours</b> (Radar)	0.68	0.28	0.98	0.64
<b>Ours</b> (DVS)	1.65	0.19	0.84	0.51
<b>Ours</b> (DVS-Radar)	0.51	0.17	0.81	0.45

TABLE II: **MAE<sub>L</sub> (L) and MAE<sub>M</sub> (M).** The lower the better.

In addition to Table II, Fig. 6 and 7 visually show the SLAM results for both Seq-1 and Seq-2. It can be remarked on both Fig. 6, 7 and in Table II that the DVS-Radar fusion setup outperforms the Radar-only and the DVS-only cases, by reaching lower  $\text{MAE}_L$  and  $\text{MAE}_M$  values. This clearly shows the advantage of DVS-Radar fusion.

Compared to the previously-proposed RGB-based solutions, our proposed system outperforms both the original

RatSLAM and the LatentSLAM, which makes use of a 11-layer DNN trained *offline*. In addition, our system either outperforms or reaches close  $MAE_L$  and  $MAE_M$  performances compared to ORB features used with RGB.

This is *remarkable* given a) the small size of our SNN-STDP networks (equivalent to a 1-hidden-layer network in terms of complexity versus 11 layers in LatentSLAM), b) the fact that our network is not pre-trained offline as in LatentSLAM, but is rather initialized randomly and learns features from the input data *on the fly* via *unsupervised* continual STDP adaptation, and c) the fact that it does not use RGB as in most SLAMs and solely relies on *lower-fidelity* DVS and Radar data. In contrast to the RGB-based solutions of Table II, our DVS-Radar setup enables us to work under strong lighting variations, where RGB solutions fail.

### B. Under low light conditions

Fig. 8 shows the SLAM results obtained for Seq-3 of Fig. 5, where lighting variations were conducted by randomly turning on and off a subset of the neon lighting in the warehouse during the drone flight. Fig. 8 clearly demonstrates the *robustness* of our DVS-Radar approach in low light, compared to the other methods.

It must be remarked that systems fusing RGB with radar have been proposed for having robustness towards environmental conditions [2], [31]. However, the RGB sensor still remains unreliable in low lighting, while in our architecture, both the DVS and radar stay functional under low light. A video showcasing our system on Seq-3 is provided at <https://youtu.be/a7gvZWNHGoI>.

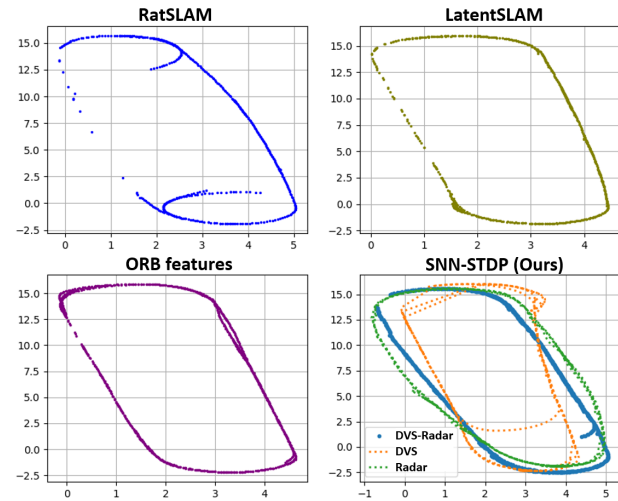


Fig. 6: *Seq-1 (dimensions in meter)*, red path in Fig. 5 (the obstacle modelling of Section IV-C is not plotted for clarity).

## VI. CONCLUSIONS

This paper has presented what is, to the best of our knowledge, a first-of-its-kind SLAM system fusing DVS and radar data with SNNs. Unlike most learning-based SLAM systems, where a DNN is trained offline using a dataset captured beforehand, our proposed system does not require

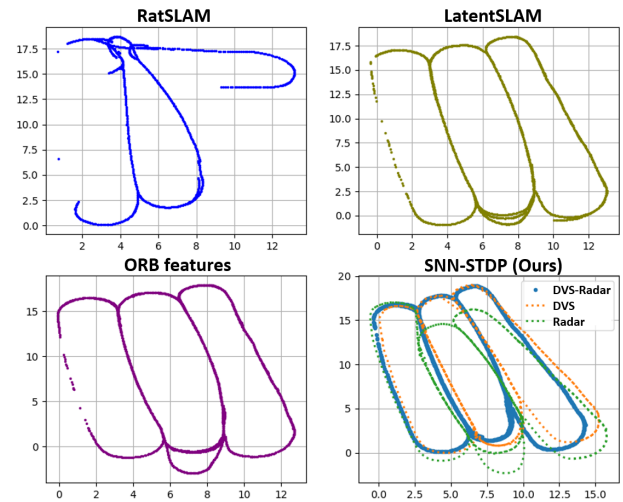


Fig. 7: *Seq-2*, all paths combined in Fig. 5 (the obstacle modelling of Section IV-C is not plotted for clarity).

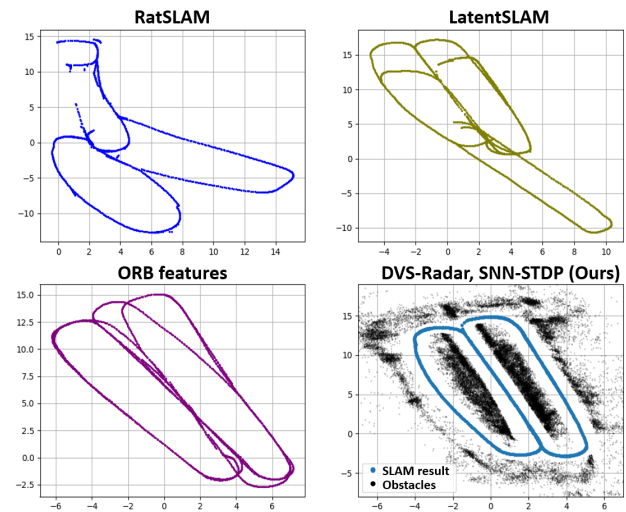


Fig. 8: *Seq-3*, blue and red paths in Fig. 5 with strong lighting variations. Our proposed DVS-Radar system is robust against lighting and clearly outperforms the other methods. Our obstacle aggregation of Section IV-C is also shown, where walls and shelves are modelled by the black dots.

any pre-training but relies on the continual and *unsupervised* adaptation of the SNN weights via STDP learning as the drone explores the environment. Finally, it has been shown that our DVS-Radar SLAM reports a competitive performance compared to state-of-the-art RGB-based solutions while enabling robust navigation and obstacle modelling under strong lighting variations. We believe that this work holds clear promise for the deployment of environmentally robust, continual learning SLAM systems for drones via low-power SNN processors.

## ACKNOWLEDGMENT

We thank Prof. J. Suykens of *KU Leuven* and dr. L. Keuninckx and D. de Tinguy of *imec* for useful discussions.

## REFERENCES

- [1] A. R. Vidal, H. Rebecq, T. Horstschaefer and D. Scaramuzza, "Ultimate SLAM? Combining Events, Images, and IMU for Robust Visual SLAM in HDR and High-Speed Scenarios," in *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 994-1001, April 2018, doi: 10.1109/LRA.2018.2793357.
- [2] O. Çatal, W. Jansen, T. Verbelen, B. Dhoedt and J. Steckel, "LatentSLAM: unsupervised multi-sensor representation learning for localization and mapping," 2021 IEEE International Conference on Robotics and Automation (ICRA), 2021, pp. 6739-6745
- [3] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel and J. D. Tardós, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM," in *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874-1890, Dec. 2021, doi: 10.1109/TRO.2021.3075644.
- [4] M. Torchalla, M. Schnaubelt, K. Daun and O. von Stryk, "Robust Multisensor Fusion for Reliable Mapping and Navigation in Degraded Visual Conditions," 2021 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), 2021, pp. 110-117, doi: 10.1109/SSRR53300.2021.9597866.
- [5] G. Gallego et al., "Event-based Vision: A Survey," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, doi: 10.1109/TPAMI.2020.3008413.9.
- [6] M. Ridolfi, N. Maccoir, J. V. Gerwen, J. Rossey, J. Hoebeke and E. de Poorter, "Testbed for warehouse automation experiments using mobile AGVs and drones," IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHOPS), 2019, pp. 919-920, doi: 10.1109/INFOCOMW.2019.8845218.
- [7] A. Safa, J. Van Assche, M. D. Alea, F. Catthoor and G. G. E. Gielen, "Neuromorphic Near-Sensor Computing: From Event-based Sensing to Edge Learning," in *IEEE Micro*, 2022, doi: 10.1109/MM.2022.3195634.
- [8] A. Vitale, A. Renner, C. Nauer, D. Scaramuzza and Y. Sandamirskaya, "Event-driven Vision and Control for UAVs on a Neuromorphic Chip," 2021 IEEE International Conference on Robotics and Automation (ICRA), 2021, pp. 103-109, doi: 10.1109/ICRA48506.2021.9560881.
- [9] Bi, G.q., Poo, M.m. (1998). Synaptic Modifications in Cultured Hippocampal Neurons: Dependence on Spike Timing, Synaptic Strength, and Postsynaptic Cell Type. *Journal of Neuroscience*, 18(24)
- [10] C. Frenkel, M. Lefebvre, J. -D. Legat and D. Bol, "A 0.086-mm<sup>2</sup> 12.7-pJ/SOP 64k-Synapse 256-Neuron Online-Learning Digital Spiking Neuromorphic Processor in 28-nm CMOS," in *IEEE Transactions on Biomedical Circuits and Systems*, vol. 13, no. 1, pp. 145-158, Feb. 2019, doi: 10.1109/TBCAS.2018.2880425.
- [11] M. Davies et al., "Loihi: A Neuromorphic Manycore Processor with On-Chip Learning," in *IEEE Micro*, vol. 38, no. 1, pp. 82-99, January/February 2018, doi: 10.1109/MM.2018.112130359.
- [12] Y. Sandamirskaya (2022). "Rethinking computing hardware for robots." *Science Robotics*, 7(67).
- [13] S. H. Cen and P. Newman, "Precise Ego-Motion Estimation with Millimeter-Wave Radar Under Diverse and Challenging Conditions," 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018, pp. 6045-6052, doi: 10.1109/ICRA.2018.8460687.
- [14] M. Sheeny, E. De Pellegrin, S. Mukherjee, A. Ahrabian, S. Wang and A. Wallace, "RADIATE: A Radar Dataset for Automotive Perception in Bad Weather," 2021 IEEE International Conference on Robotics and Automation (ICRA), 2021, pp. 1-7, doi: 10.1109/ICRA48506.2021.9562089.
- [15] N. Wessendorp, R. Dinaux, J. Dupeyroux and G. C. H. E. de Croon, "Obstacle Avoidance onboard MAVs using a FMCW Radar," 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2021, pp. 117-122, doi: 10.1109/IROS51168.2021.9635901.
- [16] A. Safa et al., "A Low-Complexity Radar Detector Outperforming OS-CFAR for Indoor Drone Obstacle Avoidance," in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 9162-9175, 2021, doi: 10.1109/JSTARS.2021.3107686.
- [17] A. Safa et al., "Fail-Safe Human Detection for Drones Using a Multi-Modal Curriculum Learning Approach," in *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 303-310, Jan. 2022, doi: 10.1109/LRA.2021.3125450.
- [18] M. J. Milford, G. F. Wyeth and D. Prasser, "RatSLAM: a hippocampal model for simultaneous localization and mapping," IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004, 2004, pp. 403-408 Vol.1, doi: 10.1109/ROBOT.2004.1307183.
- [19] R. Kreiser, A. Renner, Y. Sandamirskaya and P. Pienroj, "Pose Estimation and Map Formation with Spiking Neural Networks: towards Neuromorphic SLAM," 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 2159-2166, doi: 10.1109/IROS.2018.8594228.
- [20] G. Tang, A. Shah and K. P. Michmizos, "Spiking Neural Network on Neuromorphic Hardware for Energy-Efficient Unidimensional SLAM," 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2019, pp. 4176-4181, doi: 10.1109/IROS40897.2019.8967864.
- [21] RatSLAM python implementation, available online at: <https://github.com/renatopp/rat slam-python> (assessed 5/8/22)
- [22] E. Rublee, V. Rabaud, K. Konolige, G. R. Bradski, "ORB: An efficient alternative to SIFT or SURF." ICCV 2011: 2564-2571.
- [23] R. Elvira, J. D. Tardós and J. M. M. Montiel, "ORB-SLAM-Atlas: a robust and accurate multi-map system," 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2019, pp. 6253-6259, doi: 10.1109/IROS40897.2019.8967572.
- [24] A. Safa, A. Bourdoux, I. Ocket, F. Catthoor and G. G. E. Gielen, "On the Use of Spiking Neural Networks for Ultralow-Power Radar Gesture Recognition," in *IEEE Microwave and Wireless Components Letters*, vol. 32, no. 3, pp. 222-225, March 2022, doi: 10.1109/LMWC.2021.3125959.
- [25] Ali Safa et al., (2022). "A New Look at Spike-Timing-Dependent Plasticity Networks for Spatio-Temporal Feature Learning," (<https://arxiv.org/abs/2111.00791>)
- [26] M. Akrouf, C. Wilson, P. C. Humphreys, T. Lillicrap, and D. Tweed. 2019. "Deep learning without weight transport." Proceedings of the 33rd International Conference on Neural Information Processing Systems. Curran Associates Inc., Red Hook, NY, USA, Article 88, 976-984.
- [27] Ioffe, S. et al., (2015). "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift." In Proceedings of the 32nd International Conference on Machine Learning (pp. 448-456).
- [28] Chan, Tony F., Gene H. Golub, and Randall J. LeVeque. "Algorithms for computing the sample variance: Analysis and recommendations." *The American Statistician* 37.3 (1983): 242-247
- [29] Yu, S., Wu, J., Xu, H., Sun, R., Sun, L. (2020). "Robustness Improvement of Visual Templates Matching Based on Frequency-Tuned Model in RatSLAM." *Frontiers in Neuroinformatics*, 14.
- [30] Lowe, D.G. "Distinctive Image Features from Scale-Invariant Key-points." *International Journal of Computer Vision* 60, 91-110 (2004). <https://doi.org/10.1023/B:VISI.0000029664.99615.94>
- [31] R. Yadav, A. Vierling and K. Berns, "Radar + RGB Fusion For Robust Object Detection In Autonomous Vehicle," 2020 IEEE International Conference on Image Processing (ICIP), 2020, pp. 1986-1990, doi: 10.1109/ICIP40778.2020.9191046.