IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

HSMD: An object motion detection algorithm using a Hybrid Spiking Neural Network Architecture

PEDRO MACHADO[1] (Student Member, IEEE), ANDREAS OIKONOMOU[1], JOÃO FILIPE FERREIRA[1] (Member, IEEE) and T.M. MCGINNITY[2] (Senior Member, IEEE)

[1]Computational Neurosciences and Cognitive Robotics Group (ISTeC031), School of Sciences and Technology, Nottingham Trent University, Clifton Campus, Nottingham, NG11 8NS, United Kingdom (e-mails: {pedro.baptistamachado,andreas.oikonomou,joao.ferreira}@ntu.ac.uk

[2]Intelligent Systems Research Centre (MS225), Ulster University, Magee Campus, BT48 7JL, Derry, N. Ireland (e-mail: tm.mcginnity@ulster.ac.uk)

Corresponding author: Pedro Machado (e-mail: pedro.baptistamachado@ntu.ac.uk).

ABSTRACT

The detection of moving objects is a trivial task performed by vertebrate retinas, yet a complex computer vision task. Object-motion-sensitive ganglion cells (OMS-GC) are specialised cells in the retina that sense moving objects. OMS-GC take as input continuous signals and produce spike patterns as output, that are transmitted to the Visual Cortex via the optic nerve. The Hybrid Sensitive Motion Detector (HSMD) algorithm proposed in this work enhances the GSOC dynamic background subtraction (DBS) algorithm with a customised 3-layer spiking neural network (SNN) that outputs spiking responses akin to the OMS-GC. The algorithm was compared against existing background subtraction (BS) approaches, available on the OpenCV library, specifically on the 2012 change detection (CDnet2012) and the 2014 change detection (CDnet2014) benchmark datasets. The results show that the HSMD was ranked overall first among the competing approaches and has performed better than all the other algorithms on four of the categories across all the eight test metrics. Furthermore, the HSMD proposed in this paper is the first to use an SNN to enhance an existing state of the art DBS (GSOC) algorithm and the results demonstrate that the SNN provides near real-time performance in realistic applications.

INDEX TERMS SNN, HMSD, retinal cells, object motion sensitive ganglion cells, background subtraction, object motion detection

I. INTRODUCTION

The retina is a tiny tissue of about 1mm depth at the back of the eye, and it is responsible for the first stage of biological image processing. All vertebrate retinas possess a variable number of the same type of retinal cells, namely, photoreceptors (rods and cones), horizontal, bipolar, amacrine and ganglion cells [1], [2]. Light stimuli are sensed by the photoreceptors, which trigger electrical and chemical signals that propagate through the retinal cells and are transported via the optic nerve to the Visual Cortex. Retinal photoreceptors are sensitive to dim light (rods), colour vision (cones) and bright light (cones), and connect to bipolar and horizontal cells. Horizontal cells are responsible for regulating the signals triggered by neighbouring rods and cones. Bipolar cells receive process and transmit signals from groups of rods and cones to ganglion cells. Amacrine cells interact with groups of neighbouring bipolar cells to regulate signals transmitted to the ganglion cells, responsible for collecting the visual signals and propagating them to the visual cortex via millions of parallel channels in the optic nerve [1], [2]. The types of retinal cells vary in concentration, functionality and size. There are thousands of retinal circuits formed by types and sub-types of retinal cells wired together [1]. Different retinal circuits trigger different functionalities such as light detection, motion detection and discrimination, object motion, identification of approaching motion (looming), anticipation, motion extrapolation and omitted stimulus-response [3]. Vertebrate retinas are notable for i) incorporating millions of

these retinal circuits, ii) being extremely efficient (the whole human brain consumes approximately 20 Watts) and iii) still displaying the capability to outperform any state-of-the-art computer [4].

In computer vision, object motion detection is traditionally performed using BS methods, where the foreground (pixels or group of pixels whose light intensity values have suffered an abrupt variation) are compared with the previous image or background model [5]–[8]. BS can be static, subtracting the current image frame from the first image frame, or dynamic, subtracting the current image frame from previous image frame(s) [2], [9]–[13]. BS methods can be classified as 1) Mathematical, 2) Machine Learning and 3) Signal processing [7], [8]. Mathematical theories are the simplest way to model backgrounds using temporal average, temporal median and histograms, which can be improved using refined models (such as a mixture of Gaussians, kernel density estimation, etc.) and require low computational resources [8]. Machine learning models are more robust for performing BS, but they must be trained on the target visual features and require significant computational resources [7]. Signal processing models used to model the background using the temporal history of pixels as 1D signals and usually require moderate computational resources [8]. Although less robust, the classical mathematical BS models are better suited for real-time on near real-time applications. As real-time processing is a key objective of this work, we focus only on mathematical models in this paper.
The Hybrid Sensitive Motion Detection (HSMD) model reported in this paper was inspired by the object motion functionality exhibited by vertebrate retinas, in which object motion-sensitive retinal cells (OMS-RC) determine the difference between a local patch's motion trajectory and the background [3]. An improved version of DBS using Local Singular Value Decomposition (SVD) Binary Pattern (mathematical model) [14], [15] is enhanced by a 3-layer spiking neural network (SNN), forming a hybrid architecture.

The main contributions of the work reported in this paper are i) an object motion detection model inspired by the OMS-RC designed to work with commercial-of-the-shelf (COTS) cameras, ii) enhancement of the dynamic BS (mathematical model) using the 3-layered SNN and iii) optimisation of the proposed method for processing live capture feeds in near real-time. The algorithm was tested on the 2012 change detection (CDnet2012) [16] and 2014 change detection (CDnet2014) benchmark datasets [17] and compared with classical BS algorithms (discussed in sections II and IV). The HSMD can detect motion using commercial-off-the-shelf camera feeds and/or video clips using Spiking Neural Networks (SNN), as opposed to cameras exploiting dedicated custom architectures.
The remainder of the paper is structured as follows: related work on object detection using classical computer vision and bio-inspired computer vision is briefly reviewed in section II; the HSMD is described in section III; the training details, use-case scenarios and HSMD parameterisation are described in section IV; the results are reported and analysed in section V; and the discussion and future work are presented in section VI.

II. LITERATURE REVIEW

Reliable and optimised object motion detection in videos and live streams are an essential feature for a wide range of computer vision applications such as object tracking, intrusion detection, collision avoidance, etc. Motion detection is performed by analysing/tracking the variation of light intensities between a set of image frames.
Camera, background and foreground are three factors that affect the quality of the BS [18]. Current BS challenges include (i) abrupt illumination changes, which impact the pixel intensity values and may increase the number of false positives; (ii) dynamic objects, where background object movement may interfere with motion detection of static BS; (iii) relative motion, where both the camera and the object move at the same time, creating dynamic backgrounds; (iv) challenging weather conditions such as fog, rain, snow, air or turbulence generates errors; (v) camouflage, where camouflage regions occur when the foreground and background light intensity pixels are similar; (vi) occlusion, when another object or fixed structure obstructs the object of interest; (vii) irregular object motion - objects that suddenly increase or decrease in speed; (viii) noise, possibly arising from dirty lenses, dust, extremely high/low light intensity, etc. which decrease the quality of the detection; (ix) bumps and jitter artefacts which occur when the camera is moved; (x) image compression, where lossy compression produces a loss of information, with a consequent reduction in performance.

The OpenCV library [19] is one of the most frequently used computer vision libraries and is the reference library for computer or robot vision researchers. It includes several BS algorithms. Stauffer & Grimson [20], and KaewTraKulPong & Bowden [21] suggest modelling each pixel as a mixture of Gaussians (MOG) where the Gaussian distributions of the adaptive mixture model are analysed for determining which ones are likely to belong to the background process. All the pixel values that do not fit in the background distributions are considered foreground [20]. Zivkovic [22] proposes an efficient adaptive algorithm using the Gaussian mixture probability density (MOG2) for enhancing the MOG algorithm. MOG2 selects automatically the number of components per pixel which results in full adaptation to the observed scene. Zivkovic & Heijden [23] identified recursive equations for updating the parameters of the MOG, and proposed an enhanced algorithm using K nearest neighbours (MOG-KNN) for the automatic selection of the pixel components. The Gaussian mixture based algorithms (MOG, MOG2 and MOG-KNN) show good robustness when exposed to noise and losses due to image compression but lack sensitivity to intermittent object motion, moving background objects and abrupt illumination changes. In 2016, Sagi Zeevi [24]

proposed the CNT algorithm, which performed better on the CDnet2014 and targets embedded platforms (e.g. Raspberry PI). The CNT uses minimum pixel stability for a specified period for modelling the background; this can vary from 170 ms (default for swift movements) up to hundreds of seconds (the 60s is the default value) [25].

Godbehere *et al.* [26] suggested a single-camera statistical segmentation and tracking algorithm (GMG) by combining per-pixel Bayesian segmentation, a bank of Kalman filters and Gale-Shapley matching for the approximation of the solution to the multi-target problem. The proposed GMG algorithm is limited when processing video streams susceptible to camouflage, losses due to image compression and noise. Guo *et al.* [27] reported an adaptive BS model enhanced by a local singular value decomposition (SVD) binary pattern (LSBP) for addressing illumination changes. The proposed LSBP algorithm enhances the robustness of the motion detection to illumination changes, shadows and noise. However, it is less effective when processing video streams susceptible to camouflage, losses due to image compression and noise. More recently, in 2017, OpenCV released an improved version of the LSBP algorithm, also known as GSOC [15], [28], developed during the Google Summer of code, which enhances the LSBP algorithm by using colour descriptors and various stabilisation heuristics [14], [15]. The GSOC algorithm demonstrates better performance on the CDnet2012 and CDnet2014 datasets [14], [29] when compared to other algorithms available on the OpenCV library. The OpenCV's BS algorithms (i.e. MOG, MOG2, CNT, MOG-KNN, GMG, LSBP and GSOC) were designed for modelling the dynamic background changes (i.e. about two hundred frames are required to train the background model) and classifying all the background outliers as foreground. In this paper, the HSMD algorithm uses GSOC for performing the first stage of BS before enhancement by the SNN. The GSOC algorithm was selected over the other BS algorithms available on the OpenCV library because it is the algorithm that has performed better in the target datasets (i.e. CDnet2012 and CDnet2014) [14], [29].

Spiking neural networks have also been exploited for object motion detection. Wu *et al.* [30] proposed a bio-inspired spiking neural network to detect moving objects in a visual image sequence. The SNN was trained to extract the boundaries of moving objects from grey images. Cai *et al.* [31] expanded this work in [30] and mimicked the basic functionality of motion detection with axonal delays. These two SNN architectures [30], [31] were to perform basic detection of moving objects, but neither can process moving objects in real-time.

Lichsteiner *et al.* [32] introduced the concept of an Address-Event Representation (AER) silicon retina chip capable of generating events proportional to the log intensity changes. Farian *et al.* [33] proposed an in-pixel colour processing approach inspired by the retinal colour opponency, using the same AER concept. Brandli *et al.* [34] proposed a new version of the AER camera reported by Lichsteiner [32] called the dynamic, active pixel vision sensor (DAVIS), which exploits the efficiency of the AER protocol and introduces a new synchronous global shutter frame concurrently. In this current paper, the authors used the HSMD algorithm in conjunction with a DAVIS 240C camera and a standard RGB device, and compared the resultant performance in an object tracking scenario.

Kasabov et al. [35] proposed the deSNN that combines the use of Spike Driven Synaptic Plasticity (an unsupervised learning method for learning spatio-temporal representations) with rank-order learning (supervised learning for building rank-order models). The deSNN was tested on data collected by an Address-Event Representation (AER) silicon retina chip [32] (which generates spiking events in response to changes in light intensity) for recognising moving objects. Although able to recognise moving objects, the deSNN was designed to work specifically with AER cameras. Unlike the deSNN, the HSMD has been designed to work with commercial-off-the-shelf RGB cameras. More recently, Jiang *et al.* [36] proposed an SNN based on the Hough Transform to detect a target object with an asynchronous event stream fed by an AER camera. The algorithm [36] was able to process up to 40.74 frames per second on an Intel i7-4770 processor, accelerated by an Nvidia Geforce GTX 645. However, it is unclear whether the algorithm would work with regular commercial-of-the-shelf (COTS) cameras.

Similar to AER silicon retina chip, the HSMD algorithm mimics the basic functionality of OMS-GC with the difference that HSMD works with any COTS camera. Moreover, the HSMD uses a 3-layer SNN to enhance the GSOC algorithm, which has the best results when tested against CDnet2012 and CDnet2014 datasets. To the best of the authors' knowledge, the HSMD is the first SNN-based algorithm capable of processing image streams in near real-time(i.e. 720×480@13.82fps [CDnet2014] and 720×480@13.92fps [CDnet2012]) as a consequence of the parallel optimisations performed in terms of making use of the 96 hyper-threaded cores available the Intel(R) Xeon(R) Platinum 8160 CPU @ 2.10GHz that was used in this work.

III. HSMD ARCHITECTURE

The HSMD is a combined BS/SNN Network to create a hybrid model for detecting motion, emulating the elementary functionalities of the object-motion-sensitive ganglion cells (OMS-GC) as described in [3].

The architecture of the HSMD is shown in Figure 1. There are five layers to the overall architecture. Layer 1 performs the DBS using the GSOC algorithm (described in section IV). The resulting DBS frames are fed into Layer 2 of the Spiking Neural Network (SNN), where the pixel intensity values are converted into currents that are proportional to the light intensity (see III-C). The DBS-converted currents are fed to the Layer 2 neurons via a 1:1 synaptic connectivity. Layer 2 neurons are synaptically connected to Layer 3 neurons, which performs the first stage of motion analysis; Layer 3 neurons connect to the Layer 4 neurons via 1:1 synaptic connectivity.

Layer 4 neurons perform precise motion detection. A median filter filters their spikes to exclude random neuron activities.

A. SPIKING NEURON MODELS

The Leaky-Integrate-and-Fire (LIF) was the spiking neuron model used in this work because of its simplicity, computationally efficiency and suitability for processing images in near real-time. The LIF spiking neuron model exhibits similar, but less complex, dynamics compared to real biological neurons [37]. More complex spiking neuron models are available, e.g. Hodgkin-Huxley, but require significant computational resources and have a higher impact on the computational performance (e.g. Izhikevich [38]). The LIF neuron's dynamics are described by equation 1.

$$\tau_m \frac{\delta V_m}{\delta t} = -V_m + RI\,(t) \tag{1}$$

where $\tau_m = RC$ is the time constant, R the membrane resistance, C the membrane capacitance, $V_m(t)$ the membrane voltage and $I(t)$ is the current at time t. The membrane potential V_m is reset to the reset membrane potential (E_L) and a spike event is generated when $V_m(t)$ crosses the V_{th} (threshold voltage).

B. INPUT LAYER: DBS AND REDUCTION

Each $n \times m$ image frame (i.e. camera, video sequence or image sequences) is converted into grayscale.

The GSOC [28] delivers an adaptive DBS using colour descriptors and various stabilisation heuristics [14], [15] while processing the frames pixel-wise and leveraging the parallelism inside OpenCV [14].

C. LAYER 2: PIXEL INTENSITIES TO CURRENTS ENCODING

Pixel intensity values are converted into proportional currents and fed into the spiking neurons in Layer 2 via a 1:1 connectivity. The Layer 1 neurons were trained to trigger spike events proportional to the pixel intensity values, as described by equation 2.

$$i_c(x, y) = I(x, y).c \tag{2}$$

where $i_c(x, y)$ is the corresponding current for the image light intensity value $I(x, y)$ at coordinates x and y, and c is a conversion constant obtained experimentally (in our case, c=17.5).

D. LAYER 3: MOTION STABILITY

Layer 3 is used to perform motion stabilisation through the creation of local buffers by delaying the propagation of spike events. A delay is created when a given neuron of layer 2 connects to a neuron in layer 3, before being passed to Layer 4, instead of the direct Layer 2 to Layer 4 connection. Spike events passing through Layer 3 are buffered by neurons in Layer 3 for one simulation time-step (δt, in this work δt = 10 ms) and presented to the neurons in Layer 4. N[n] in the following simulation time-step.

The neurons in Layer 2 are connected to the Layer 3 neurons via a 1:1 connectivity. Finally, the Layer 3 neurons connect to the Layer 4 neurons via a 1:1 connectivity as shown in Figure 2. All synaptic weights from Layer 2 to Layers 3 and 4 have a value of 1370 (obtained experimentally).

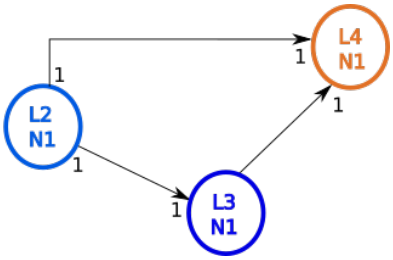

FIGURE 2. HSMD connectivity. In this example, it can be seen that the neuron 1 (N1) of each layer connects to the N1 of the subsequent layer.

E. LAYER 4: MOTION DETECTION

The Layer 4 neurons receive synaptic connections from the neurons in Layer 2 and Layer 3 via excitatory synapses and exploit these spiking events to detect motion. Spike events generated by Layer 4 neurons result from dynamic changes between sequential image frames. Signals received directly from Layer 2 neurons enable detection of changes between the current image frame n and the previous image frame n-1. In contrast, those routed via Layer 3 neurons compare the image frame n-1 with the image frame n-2. Layer 4 spike events are mapped into the corresponding area in the original image captured from the camera. The synaptic weights obtained experimentally are 1370 for all the synapses. The Layer 2 to Layer 4 weights were tuned to forward all the spike events generated in Layer 2. The Layer 3 to Layer 4 synaptic weights were tuned to produce spike events from the Layer 4 neurons for each group of two sequential spike events. The main goal is to give high importance (larger weight) to new spike events (frame [n] - frame [n-1]) and to give lower importance to older spike events (frame [n-1] - frame [n-2]).

F. LAYER 5: FILTERING

The Layer 4 neurons' spike events matrix is mapped into a motion matrix M_d of the same size as the captured image (i.e. $n \times m$). The events in the M_d matrix are filtered using an averaging filter described by equations 3 and 4:

$$H(u, v) = \frac{1}{u.v}\left(\begin{bmatrix} w_{0,0} & \dots & w_{0,u} \\ \dots & \dots & \dots \\ w_{v,0} & \dots & w_{v,u} \end{bmatrix}\right) \tag{3}$$

$$Y_d(x, y) = M_d(x, y) * H(u, v) \tag{4}$$

where $Y_d(x, y)$ is the filtered motion detection matrix, $H(u, v)$ is the averaging filter, u and v are the convolution window length and height respectively, $*$ is the convolution operator, w is the filter window.

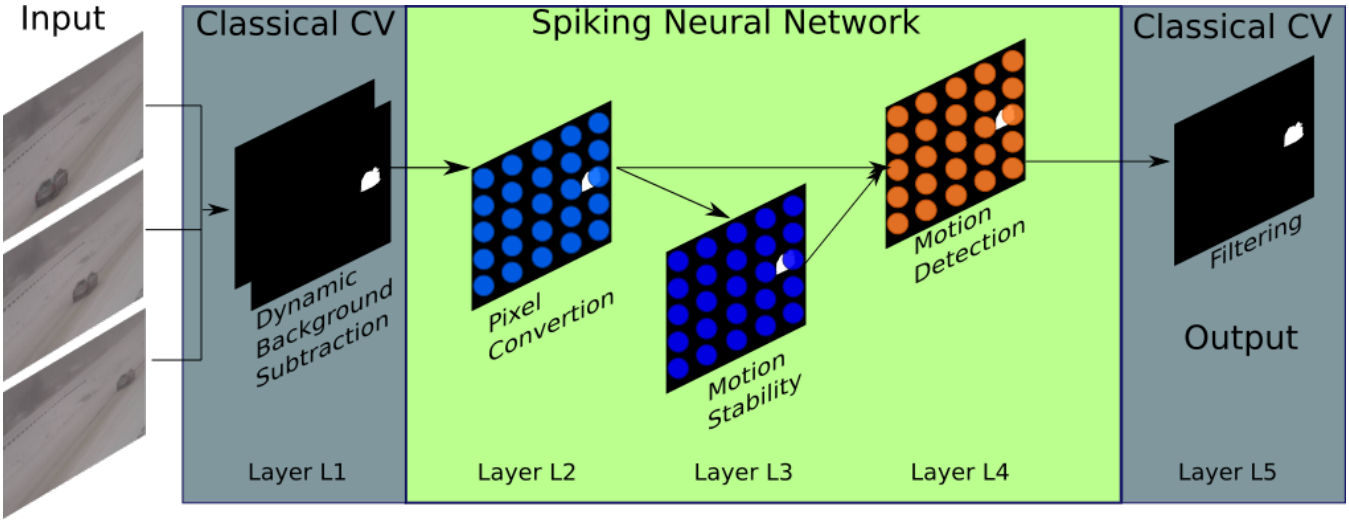

FIGURE 1. HSMD with (i) $n \times m$ image input followed by DBS, three spiking neuronal layers and filtering. Layer 1: DBS, Layer 2: pixel intensity to spike events encoding, Layer 3: Motion stability, Layer 4: motion detection and Layer 5: filtering.

IV. METHODOLOGY

The HSMD was implemented in C++ using the C++ Standard Template Library 17 (C++17) [39] (implementation of data structures), Boost 1.71 [40] (file management) and OpenCV 4.5.0 [19] (which provides common computer vision functionalities such as resize, capture and display images) in Ubuntu 20.04 LTS[1].

A. HSMD SETUP

The HSMD initial setup includes the following steps:

Step 1 - Select between live capture, video analysis or image sequences: The user can opt to run the algorithm directly on images being captured by the camera or provide the path of a video or set of image sequences for motion analysis.

Step 2 - Create the Layer 2 to Layer 4 neural network: Read the first image and compute the size of the image. The number of neurons is computed automatically from the dimensions of the first image in a sequence of images.

Step 3 - Set the neuronal parameters: The LIF parameters recommended in the references [41], [42] and frequently used in LIF SNN circuits, were used to configure the SNN. Therefore, the simulation was configured with a time step of δt=10 ms and the default neuron parameters as follows: initial V_m=-55.0 mV, E_L = -55.0 mV, C_m = 10.0 pF, R_m=1.0 MOhm, V_{reset}=-70.0 mV, V_{min}=-70.0 ms, V_{th}=-70.0 mV, τ_m=10.0 ms, t_{ref}=2 ms, w_{syn} = 1555.0 (neurons L3 and L4) and w_{p2i}=8.0 (L2 neurons only).

Step 4 - Start the image acquisition : Images are collected from devices, video streams or obtained from folders with sequences of images while the HSMD algorithm is being executed. The pseudo-code of the main algorithm is described in Algorithm 1.

[1]Available online http://releases.ubuntu.com/20.04/, last accessed 12/11/2020

B. DATASETS AND METRICS

1) Datasets

The CDnet2012 [16] (cited more than 379 times[2]) and CDnet2014 [17] (cited more than 300 times[3]) benchmark datasets were designed for benchmarking BS algorithms. While the HSMD algorithm has been designed as an object detection algorithm and not a BS algorithm, nevertheless these two datasets provide challenging scenarios for robust comparable assessment of the proposed algorithm and network. Within the two benchmark datasets tyhe HSMD was compared with the following state-of-the-art BS algorithms available on the OpenCV library: MOG [20], MOG2 [22], MOG-KNN [23], GMG [26], LSBP [27], CNT [24] and GSOC [15] methods. The OpenCV BS algorithms were used because they are highly optimised, reliable, and publicly available to anyone who wants to test or compare their algorithms.

Each of the benchmark videos in the CDnet2012 [16] and CDnet2014 [17] are considered under one or more challenge categories as follows:

CDnet2012 and CDnet2014

- **Baseline** - reference videos which are relatively simple to classify; some videos contain very simple movements from the next four categories.
- **Dynamic Background** - videos that have both foreground and background motion (e.g. water movement and shaking trees).
- **Camera Jitter** - videos captured with cameras installed on unstable structures.

[2]Retrieved from, https://ieeexplore.ieee.org/abstract/document/6238919, last accessed: 12/10/2020

[3]Retrieved from, https://ieeexplore.ieee.org/document/6910011, last accessed: 12/10/2020

Algorithm 1 HSMD main algorithm pseudo-code

```
newImage = capture_image_camera
newImageGrey = colour2grey(newImage)
set_number_neurons_from_newImageGrey_shape

build_neuronal_network
load_pretrained_weights
while frames available do
    reset_spike_events
    newImage = capture_image_camera
    newImageGrey = colour2grey(newImage)
    newImageReduced = newImageGrey
    dynSubImage = newImageReduced − previousImage
    previousImage = newImageReduced
    for I in dynSubImage do
        if dynSubImage[I]< Threshold then
            dynSubImage[I] = 0.0
        end if
        currents = convPixel2Current(dynSubImage)

        for i:=0 to timestep do
            apply_currents_to_neurons_L2
            compute_L2_neuron_spikes;
            convert_L2_neuron_spikes_to_currents;
            compute_L3_neuron_spikes;
            convert_L3_neuron_spikes_to_currents;
            compute_L4_neuron_spikes;
        end for
        spikes = get_sumSpikeEventsPerL4Neuron()

        masked_spikes = applyAveragingFilter(spikes)
        spikes = normalise(spikes)
        display(newImage)
        display(spikes)
    end for
end while
Display_spike_rates
```

- **Shadow** - videos containing narrow shadows from solid structures or moving objects.
- **Intermittent Object Motion** videos that include objects that are static for most of the time and suddenly start moving.
- **Thermal** - videos that exhibit thermal artefacts (i.e. bright spots and thermal reflections on windows and floors).

CDnet2014 only

- **Challenging Weather**: Outdoor videos showing very-low visibility winter storm conditions.
- **Low Frame-Rate**: videos capture at varying frame rates between 0.17 and 1 fps;
- **Night**: includes traffic videos with low visibility and strong headlights.
- **Pan, Tilt and Zoom (PTZ)**: videos recorded with cameras exposed to PTZ movements.
- **Air Turbulence**: videos filmed from distances of 5 to 15 km exhibiting air turbulence and frames distortion.

The BS algorithms were configured with the default OpenCV settings [19] and compared against the HSMD algorithm.The ground-truth provided by the datasets is composed of the following labels [16], [17]:

- **Static** - grayscale value 0;
- **Shadow** - grayscale value 50;
- **non-Region of Interest (RoI)** - grayscale value 85;
- **Unknown** - grayscale value 170;
- **Moving** - grayscale value 255;

The *static* and *moving* classes contain pixels that belong to the background and foreground, respectively; The *shadows*, one of the most challenging artefacts, should be classified as part of the background; The *unknown* region should not be considered either background or foreground because it contains pixels that cannot be accurately classified as background or foreground. The non-ROI pixels serve to exclude frames from being classified because some BS algorithms require several pixels for the model to stabilise (i.e. create the background model) and for preventing corruption by non-related activities to the considered category [16], [17]. Figure 3 shows the 5 class regions.

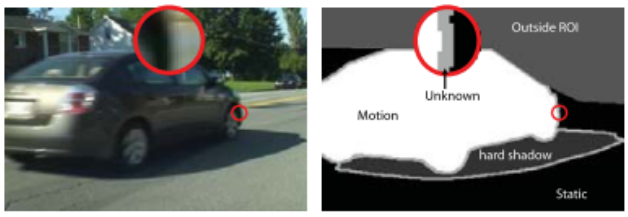

FIGURE 3. Raw image frame (left) and its respective ground-truth (right). The ground-truth images show the annotations using the datasets labels. Adapted from [16]

2) Metrics

The average performance obtained for each category using each BS method and the HSMD algorithms is characterised via eight metrics, as shown below. TP is the number of true positives, TN the number of true negatives, FN the number of false negatives, and FP the number of false positives [16], [17].

1) Recall (Re): $R_e = \frac{TP}{TP+FN}$;
   Re: rank by **descending order** order;

2) Specificity (Sp): $Sp = \frac{TN}{TN+FP}$;
   Sp rank by **descending order** order;

3) False Positive Rate (FPR): $FPR = \frac{FP}{FP+TN}$;
FPR rank by **ascending order** order;

4) False Negative Rate (FNR): $FNR = \frac{FN}{FN+TP}$;
FNR rank by **ascending order** order;

5) Wrong Classifications Rate (WCR):
$WCR = \frac{FN+FP}{TP+FN+FP+TN}$;
WCR rank by **ascending order** order;

6) Correct Classifications Rate (CCR):
$CCR = \frac{TP+TN}{TP+FN+FP+TN}$;
CCR rank by **descending order** order;

7) Precision (Pr): $Pr = \frac{TP}{TP+FP}$;
Pr rank by **descending order** order;

8) F-measure (F1): $F1 = 2 \times \frac{PR.RE}{PR+RE}$
$F1$ rank by **descending** order;

These eight metrics contribute to the overall average ranking (R) and overall average ranking across all categories ($\overline{RC}$).

Average Ranking (R): $R = \frac{\overline{Re}+\overline{Sp}+\overline{FPR}+\overline{FNR}+\overline{WCR}+\overline{CCR}+\overline{F1}}{nMet}$;
R rank by **ascending** order;

Average Ranking across all Categories ($\overline{RC}$):
$\overline{RC} = \frac{RE+SP+FPR+FNR+WCR+CCR+F1}{nMet}$;
$\overline{RC}$ rank by **ascending** order;

where $nMet$ is the number of metrics (8 in this case).

V. RESULTS

The HSMD was tested on both datasets under the same conditions to ensure an accurate and rigorous comparison. The results are presented both as overall results and per category to understand better the specific performances obtained per method. The overall results for each method are presented in section V-A and the results per method and category in section V-B.

Please note that in the tables (1 to 3), the ↑ means that the highest score is the best result and the ↓ that the lowest result is the best result. The best results are highlighted using light grey for all the methods except the HSMD results highlighted in dark grey. Re is the Recall, Sp is the Specificity, FPR is the False Positive Rate, FNR is the False Negative Rate, WCR is the Wrong Classifications Rate, CCR is the Correct Classifications Rate, Pr is the Precision, F1 is the F score or F-measure, R is the Average Ranking and $\overline{RC}$ is the Average Ranking across all Categories.

Figure 4 shows the results obtained for each of the five categories common to both CDnet2012 and CDnet2014.

A. OVERALL RESULTS

Tables 1 and 2 present the overall results obtained per method and per metric, ranked by $\overline{RC}$ (average ranking across all categories, first column) in ascendant order.

From Table 1 (2nd column), it may be seen that the HSMD algorithm ranks in first place across all eight methods with which it is compared when tested on the CDnet2012 dataset. Although the HSMD performed very well in 5 of the eight metrics, it is essential to highlight the WCR, CCR and F1 metrics results. The results show that the HSMD is sensitive to object motion due to the highest correct counts and lowest wrong counts rate, which contribute to getting the highest F-score and the second-best Precision. Furthermore, it is possible to conclude that the HSMD has improved the performance of the GSOC algorithm compared to when it is used alone when tested on the six categories of the CDnet2012.

The HSMD has also performed very well when tested on the CDnet2014 (see Table 2).

Table 2 shows that the HSMD algorithm was ranked in first place in the Average Ranking across all Categories column when tested on the CDnet2014 dataset. The HSMD performed very well in 7 of the eight metrics and exceptionally well on the Precision metrics. It can also be seen that there was a slight decrease in the HSMD performance when tested on the eleven categories available on the CDnet2014 as compared to the original six of the CDNet2012 dataset. The result is to be expected; none of the methods has excellent performance across all metrics.

B. RESULTS OBTAINED PER CATEGORY

The Average Ranking (R) for each of the methods per category is shown in Table 3.

Figures 5 and 6 show the variation of the ranks obtained per category and per method.

From the analysis of the results shown in Table 3 and Figures 5 and 6 it is possible to infer that the HSMD is sensitive to intermittent object motion, night vision, baseline and turbulence; These categories share the fact that they relate to moving objects that have high contrast, which is ideal for sensing by the spiking neurons. The HSMD has improved the results of the GSOC in 8 of the 11 categories, except for the low frame rate, dynamic background and camera jitter categories. It is also easy to visualise that the HSMD exhibits the lowest R variation, which justifies why the HSMD was ranked in the first place.

C. RESULTS ANALYSIS

The HSMD performed very severely in the dynamic background and low frame rate categories, suggesting that the spiking neuron model is not ideal for distinguishing the type of motion. i.e. the spiking neurons detect motion but are unable to distinguish between a shadow or the object itself. This result is probably because, in vertebrate retinas, only the ganglion cells are spiking cells suggesting that distinction between the main object and shadows is probably performed by other non-spiking cells. Nevertheless, the creation of the new approach incorporating both the GSOC algorithm and the SNN, which emulates the basic OMS-GC functionality, clearly improves the accuracy of the GSOC algorithm.

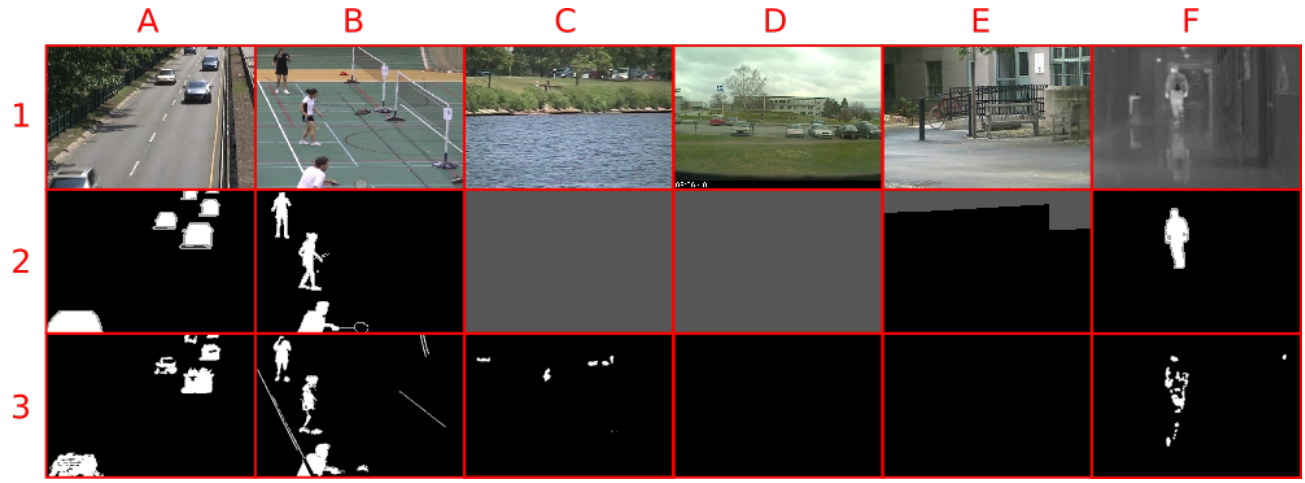

FIGURE 4. Results obtained for each of the five categories common to both CDnet2012 and CDnet2014 datasets. Column A: baseline; B: camera jitter, C: dynamic background; D: dynamic object motion; E: shadow and F: thermal. Row 1: RGB image; 2: ground-truth; and 3: HSMD binarised. The raw images, shown in the first row, demonstrate the scenarios that can be found in both datasets. The corresponding ground truth images, presented in the second row, show the 5 labels, namely, i) static [grayscale value 0], ii) shadow [grayscale value 50], iii) non-ROI [grayscale value 85], iv) unknown [grayscale value 170] and v) moving [grayscale value 255]. The corresponding binarised images generated by the HSMD are shown in the third row.

TABLE 1. CDnet2012 overall results

Method	$\overline{RC}\downarrow$	Re ↑	Sp ↑	FPR ↓	FNR ↓	WCR ↓	CCR ↑	F1 ↑	Pr ↑
HSMD	**2.8**	0.52	0.994	0.006	0.23	**0.024**	**0.976**	**0.77**	0.62
GSOC	3.5	0.54	0.993	0.007	0.25	*0.024*	*0.976*	0.75	*0.63*
MOG2	3.8	0.37	0.995	0.004	0.24	0.026	0.974	0.76	0.50
GMG	3.9	0.20	*0.998*	*0.002*	*0.21*	0.033	0.967	0.79	0.32
KNN	4.3	0.39	0.995	0.005	0.26	0.025	0.975	0.74	0.51
MOG	4.5	0.32	0.996	0.004	0.26	0.030	0.970	0.74	0.44
CNT	6.1	*0.73*	0.927	0.073	0.71	0.081	0.919	0.29	0.41
LSBP	7.3	0.57	0.90	0.096	0.80	0.109	0.891	0.20	0.29

↑: the highest score is the best.
↓: the lowest result is the best.
Best HSMD results are highlighted using dark grey, while best results per category are highlighted with light grey for other methods. Re is the Recall, Sp is the Specificity, FPR is the False Positive Rate, FNR is the False Negative Rate, WCR is the Wrong Classifications Rate, CCR is the Correct Classifications Rate, Pr is the Precision, F1 is the F score or F-measure and $\overline{RC}$ is the Average Ranking across all Categories.

The CDnet2012 and CDnet2014 datasets are composed of image files of different resolution, and accordingly, the processing times vary. The HSMD takes approximately 72.4ms (CDnet2014) and 71.9ms (CDnet2012) to process images of 720×480 on a 96-cores Intel(R) Xeon(R) Platinum 8160 CPU @ 2.10GHz equipped with 792 GB of DDR4 and 12.7 TB of disk space. The slight variations are related to other applications running in the background. Therefore, the HSMD is capable of processing images of 720×480 at an average speed of 13.82fps (CDnet2014) and 13.92fps (CDnet2012). Finally, the HSMD is the first hybrid SNN algorithm capable of processing images at such a frame rate, as far as the authors are aware.

VI. CONCLUSION AND FUTURE WORK

A bio-inspired hybrid spiking neural network (HSMD) has been proposed to detect object motion and assess against the CDnet2012 and CDnet2014 datasets. These incorporate video sequences of many moving objects under various challenging environmental conditions and are widely used for benchmarking BS algorithms. The CDnet2012 is composed of 6 categories of movements, and the CDnet2014 augments the initial 6 to 11 categories of movements. Eight metrics, utilised as standard in the CDnet datasets, were used to assess and compare the quality of the HSMD algorithm. The HSMD algorithm performed overall best in both the CDnet2012 and CDnet2014 while performing better than all the tested DBS algorithms in the intermittent object motion, night videos, thermal and turbulence categories, second best in the bad weather category and the third-best on the baseline and shadow categories. The comparatively good results are a consequence of using the SNN for emulating the basic functionality of OMS-GC, which improves the sensitivity of the HSMD to object motion. The HSMD is

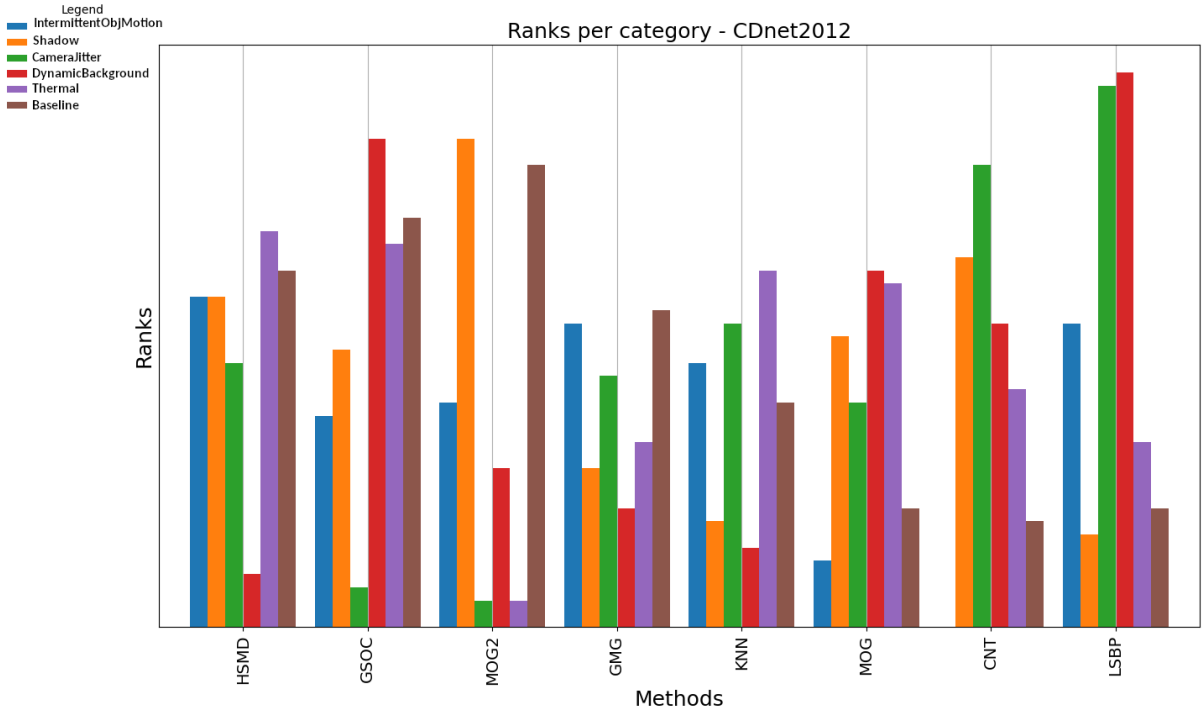

FIGURE 5. CDnet2012 overall results per category and method. The highest bars show the higher ranks, and it is clear that none of the methods had the best ranks in all the categories. Furthermore, it is possible to see that the HSMD achieved high ranks across all the categories with the exception of dynamic background.

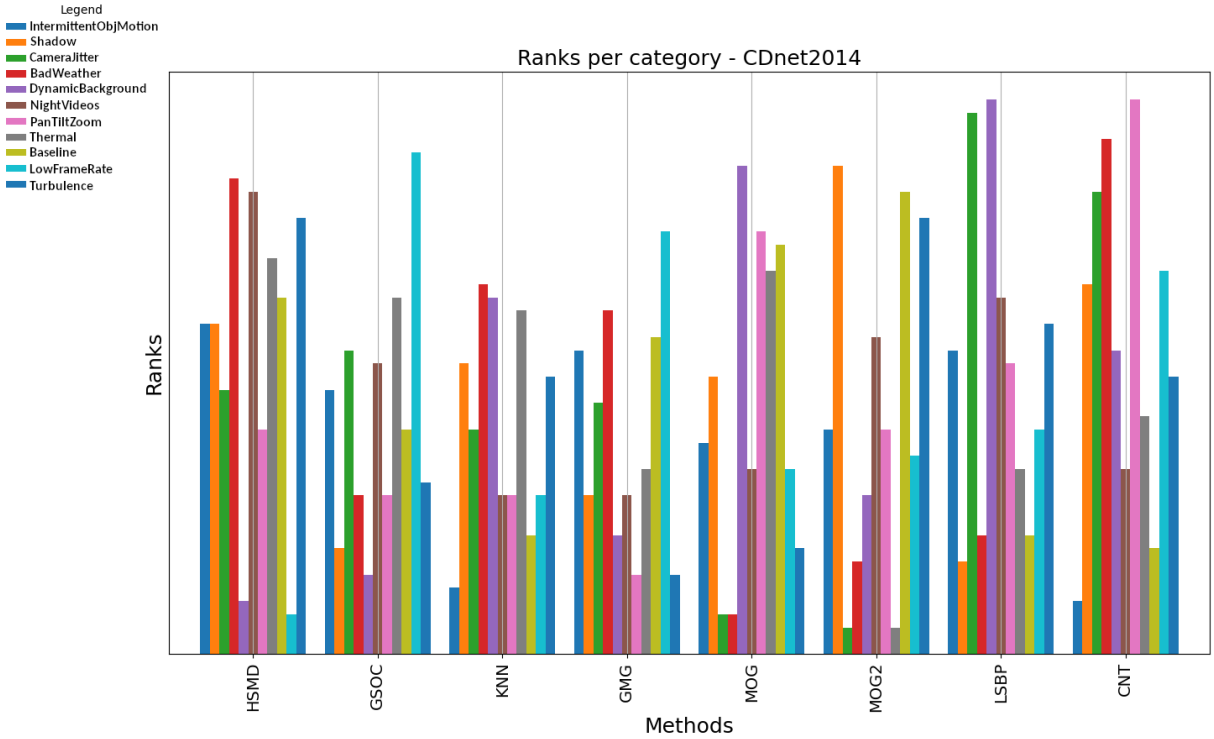

FIGURE 6. CDnet2014 overall results per category and method. The highest bars show the higher ranks, and it is clear that none of the methods had the best ranks in all categories. Furthermore, it is possible to see that the HSMD achieved high ranks across most of the categories except dynamic background and low frame rate.

TABLE 2. CDnet2014 overall results

Method	$\overline{RC}\downarrow$	Re ↑	Sp ↑	FPR ↓	FNR ↓	WCR ↓	CCR ↑	F1 ↑	Pr ↑
HSMD	**2.9**	0.55	0.993	0.007	0.35	0.018	0.982	0.65	**0.60**
GSOC	3.0	0.40	0.995	0.005	0.38	*0.017*	*0.983*	0.62	0.48
KNN	3.5	0.34	0.996	0.004	*0.32*	0.019	0.981	*0.68*	0.45
GMG	4.3	0.24	*0.997*	*0.003*	0.36	0.022	0.978	0.64	0.35
MOG	4.4	0.58	0.991	0.009	0.39	0.019	0.981	0.61	0.60
MOG2	4.5	0.39	0.994	0.006	0.42	0.018	0.982	0.58	0.47
LSBP	6.5	0.58	0.945	0.055	0.79	0.064	0.936	0.21	0.31
CNT	7.0	*0.72*	0.930	0.070	0.80	0.075	0.925	0.20	0.32

↑: the highest score is the best result.
↓: the lowest result is the best.
Best results per category are highlighted using dark grey for the HSMD and light grey for other methods. Where Re is the Recall, Sp is the Specificity, FPR is the False Positive Rate, FNR is the False Negative Rate, WCR is the Wrong Classifications Rate, CCR is the Correct Classifications Rate, Pr is the Precision, F1 is the F score or F-measure and $\overline{RC}$ is the Average Ranking across all Categories.

TABLE 3. Results per category

IntObjMotion		shadow		cameraJitter		badWeather		dynamicBackground		nightVideos		PTZ		thermal		baseline		lowFramerate		turbulence	
Method	R↓	Method	R↓	Method	R↓	Method	R↓	Method	R↓	Method	R↓	Method	R↓	Method	R↓	Method	R↓	Method	R↓	Method	R↓
HSMD	**3.875**	*MOG2*	*2.375*	*LSBP*	*1.875*	*CNT*	*2.125*	*LSBP*	*1.75*	**HSMD**	**2.625**	*CNT*	*1.75*	**HSMD**	**3.25**	*MOG2*	*2.625*	*GSOC*	*2.25*	**HSMD**	**2.875**
LSBP	4.125	CNT	3.5	CNT	2.625	**HSMD**	**2.5**	MOG	2.375	LSBP	3.625	MOG	3.0	MOG	3.375	MOG	3.125	GMG	3.0	MOG2	2.875
GMG	4.125	**HSMD**	**3.875**	GSOC	4.125	KNN	3.5	KNN	3.625	MOG2	4.0	LSBP	4.25	GSOC	3.625	**HSMD**	**3.625**	CNT	3.375	LSBP	3.875
GSOC	4.5	KNN	4.25	**HSMD**	**4.5**	GMG	3.75	CNT	4.125	GSOC	4.25	MOG2	4.875	KNN	3.75	GMG	4.0	LSBP	4.875	KNN	4.375
MOG2	4.875	MOG	4.375	GMG	4.625	GSOC	5.5	MOG2	5.5	MOG	5.25	**HSMD**	**4.875**	CNT	4.75	GSOC	4.875	MOG2	5.125	CNT	4.375
MOG	5.0	GMG	5.5	KNN	4.875	LSBP	5.875	GMG	5.875	CNT	5.25	KNN	5.5	LSBP	5.25	KNN	5.875	MOG	5.25	GSOC	5.375
KNN	6.375	GSOC	6.0	MOG	6.625	MOG2	6.125	GSOC	6.25	GMG	5.5	GSOC	5.5	GMG	5.25	LSBP	5.875	KNN	5.5	MOG	6.0
CNT	6.75	LSBP	6.125	MOG2	6.75	MOG	6.625	**HSMD**	**6.5**	KNN	5.5	GMG	6.25	MOG2	6.75	CNT	6.0	**HSMD**	**6.625**	GMG	6.25

↓: the lowest result is the best.
The HSMD results are highlighted using dark grey for the HSMD, and the best results of other methods are highlighted using light grey.
R is the average Ranking.

also the first hybrid SNN algorithm capable of processing video/image sequences with near real-time performance (i.e. 720×480@13.82fps [CDnet2014] and 720×480@13.92fps [CDnet2012]).

Future work includes optimising the HSMD algorithm to detect and track motion in challenging scenarios (e.g. low frame rate, dynamic background and camera jitter) and an investigation to verify if the SNN improves the output for all the remaining methods. Furthermore, the authors will also test if the SNN can enhance other BS algorithms available on the OpenCV library. The authors also aim to design and implement the HSMD approach on Multi-Processor System-on-Chip (MPSoC) technology and high-end Field-Programmable Arrays (FPGA) hardware to accelerate the HSMD algorithm and b) reduce the power consumption for embedded applications.

REFERENCES

[1] H. Kolb, "How the Retina Works," American Scientist, vol. 91, pp. 28 – 34, 2003.
[2] D. Holmes, "Reconstructing the retina," Nature, vol. 561, sep 2018.
[3] T. Gollisch and M. Meister, "Eye Smarter than Scientists Believed: Neural Computations in Circuits of the Retina," Neuron, vol. 65, no. 2, pp. 150–164, 2010.
[4] C. Trujillo Herrera and J. G. Labram, "A perovskite retinomorphic sensor," Applied Physics Letters, vol. 117, p. 233501, dec 2020.
[5] M. Piccardi, "Background subtraction techniques: a review," in 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583), vol. 4, pp. 3099–3104 vol.4, 2004.
[6] L. Maddalena and A. Petrosino, "Background subtraction for moving object detection in rgbd data: A survey," Journal of Imaging, vol. 4, no. 5, 2018.
[7] B. Garcia-Garcia, T. Bouwmans, and A. J. Rosales Silva, "Background subtraction in real applications: Challenges, current models and future directions," Computer Science Review, vol. 35, p. 100204, 2020.
[8] M.-N. Chapel and T. Bouwmans, "Moving Objects Detection with a Moving Camera: A Comprehensive Review," jan 2020.
[9] A. STALIN and W. AMITABH, "Bsfd : Background Subtraction Frame Difference Algorithm for Moving Object," Journal of Theoretical and Applied Information Technology, vol. 60, no. 3, pp. 623–628, 2014.
[10] A. Vacavant, T. Chateau, A. Wilhelm, and L. Lequièvre, "A Benchmark Dataset for Outdoor Foreground/Background Extraction," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 7728 LNCS, pp. 291–300, Springer, 2013.
[11] R. D. Sharma, S. L. Agrwal, S. K. Gupta, and A. Prajapati, "Optimized dynamic background subtraction technique for moving object detection and tracking," 2nd International Conference on Telecommunication and Networks, TEL-NET 2017, vol. 2018-Janua, pp. 1–3, 2018.
[12] M. Rashid and V. Thomas, "A Background Foreground Competitive Model for Background Subtraction in Dynamic Background," Procedia Technology, vol. 25, no. Raerest, pp. 536–543, 2016.
[13] J.-W. Seo and S. D. Kim, "Dynamic background subtraction via sparse representation of dynamic textures in a low-dimensional subspace," Signal, Image and Video Processing, vol. 10, pp. 29–36, jan 2016.
[14] V. Samsonov, "Improvement of the background subtraction algorithm," 2017. https://summerofcode.withgoogle.com/archive/2017/projects/6453014550282240/, last accessed: 23/11/2020.
[15] V. Samsonov, "Improved background subtraction algorithm," Nov. 2017. doi: 10.5281/zenodo.4269865, last accessed: 23/11/2020.
[16] N. Goyette, P. M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, "changedetection.net: A new change detection benchmark dataset," IEEE Computer

Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 1–8, 2012.
[17] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, "CDnet 2014: An Expanded Change Detection Benchmark Dataset," in 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 393–400, IEEE, jun 2014.
[18] R. Kalsotra and S. Arora, "A Comprehensive Survey of Video Datasets for Background Subtraction," IEEE Access, vol. 7, pp. 59143–59171, 2019.
[19] OpenCV, "Background Subtraction Algorith," 2020. https://docs.opencv.org/4.5.0/d7/df6/classcv_1_1BackgroundSubtractor.html, last accessed: 10/11/2020.
[20] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," in Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149), vol. 2, pp. 246–252, IEEE Comput. Soc, 1999.
[21] P. KaewTraKulPong and R. Bowden, "An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection," Video-Based Surveillance Systems, pp. 135–144, 2002.
[22] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004., vol. 3522, pp. 28–31 Vol.2, IEEE, 2004.
[23] Z. Zivkovic and F. Van Der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," Pattern Recognition Letters, vol. 27, no. 7, pp. 773–780, 2006.
[24] S. Zeevi, "BackgroundSubtractorCNT: A Fast Background Subtraction Algorithm," Dec. 2016. https://doi.org/10.5281/zenodo.4267853, last accessed: 23/11/2020.
[25] S. Zeevi, "Fastest background subtraction is BackgroundSubtractorCNT," 2016. https://www.theimpossiblecode.com/blog/fastest-background-subtraction-opencv/, last accessed: 09/11/2020.
[26] A. B. Godbehere and K. Goldberg, "Algorithms for Visual Tracking of Visitors Under Variable-Lighting Conditions for a Responsive Audio Art Installation," in Controls and Art, pp. 181–204, Cham: Springer International Publishing, 2014.
[27] L. Guo, D. Xu, and Z. Qiang, "Background Subtraction Using Local SVD Binary Pattern," IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 1159–1167, 2016.
[28] OpenCV, "Background Subtractor GSOC," 2020. https://docs.opencv.org/4.5.0/d4/dd5/classcv_1_1bgsegm_1_1BackgroundSubtractorGSOC.html, last accessed: 06/11/2020.
[29] OpenCV, "GSOC Background Subtraction," 2021. https://docs.opencv.org/4.5.2/d4/dd5/classcv_1_1bgsegm_1_1BackgroundSubtractorGSOC.html.
[30] Q. Wu, T. M. McGinnity, L. Maguire, J. Cai, and G. D. Valderrama-Gonzalez, "Motion Detection Using Spiking Neural Network Model," in Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence, pp. 76–83, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.
[31] R. Cai, Q. Wu, P. Wang, H. Sun, and Z. Wang, "Moving Target Detection and Classification Using Spiking Neural Networks," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 7202 LNCS, pp. 210–217, 2012.
[32] P. Lichtsteiner and T. Delbruck, "A 64x64 AER logarithmic temporal derivative silicon retina," in 2005 PhD Research in Microelectronics and Electronics - Proceedingsof the Conference, 2005.
[33] L. Farian, J. A. Lenero-Bardallo, and P. Hafliger, "A Bio-Inspired AER Temporal Tri-Color Differentiator Pixel Array," IEEE Transactions on Biomedical Circuits and Systems, vol. 9, pp. 686–698, oct 2015.
[34] C. Brandli, S. Member, R. Berner, M. Yang, S. Member, S.-c. Liu, S. Member, and T. Delbruck, "A 240 ÃŮ 180 130 dB 3 Âţs Latency Global Shutter Spatiotemporal Vision Sensor," vol. 49, no. 10, pp. 2333–2341, 2014.
[35] N. Kasabov, K. Dhoble, N. Nuntalid, and G. Indiveri, "Dynamic evolving spiking neural networks for on-line spatio- and spectro-temporal pattern recognition," Neural Networks, vol. 41, pp. 188–201, May 2013.
[36] Z. Jiang, Z. Bing, K. Huang, and A. Knoll, "Retina-Based Pipe-Like Object Tracking Implemented Through Spiking Neural Network on a Snake Robot," Frontiers in Neurorobotics, vol. 13, pp. 1–11, may 2019.
[37] W. Gerstner and W. M. Kistler, Spiking Neuron Models. Cambridge: Cambridge University Press, 2002.
[38] E. Izhikevich, "Which Model to Use for Cortical Spiking Neurons?," IEEE Transactions on Neural Networks, vol. 15, pp. 1063–1070, 9 2004.
[39] F. Standard C++, "STL C++17," 2017. https://isocpp.org/std/status, last accessed: 23/11/2020.
[40] Boost, "Boost C++ libraries," 2020. http://www.boost.org/, last accessed: 23/11/2020.
[41] R. Jolivet, T. J. Lewis, and W. Gerstner, "Generalized Integrate-and-Fire Models of Neuronal Activity Approximate Spike Trains of a Detailed Model to a High Degree of Accuracy," Journal of Neurophysiology, vol. 92, pp. 959–976, aug 2004.
[42] R. Brette and W. Gerstner, "Adaptive Exponential Integrate-and-Fire Model as an Effective Description of Neuronal Activity," Journal of Neurophysiology, vol. 94, pp. 3637–3642, nov 2005.

PEDRO MACHADO received his MSc in Electrical and Computers Engineering from the University of Coimbra (2012) and is currently doing a part-time PhD in Computer Sciences at Nottingham Trent University. Pedro's expertise includes FPGA design, computer vision, bio-inspired computing, robotics and computational intelligence. His research interests in computer science are retinal cell understanding, biological nervous system modelling, spiking neural networks, robotics and autonomous systems, applied computer vision and neuromorphic hardware.

DR ANDREAS OIKONOMOU holds a B.Sc. in Engineering (1999), M.Sc. in Information Technology for Management (2000) and Ph.D. in Computer Science with a specialisation in Biomedical Computing (2005). All completed at Coventry University in the UK. Dr Oikonomou has previously taught Computer Science and Games Development at Derby and Coventry Universities in the UK and has also worked as a Project Manager, Quality Assurance Manager and Games Studio head in the software development industry. He has +40 publications on subjects including Image Processing, Interaction Design and Software Simulations. His current research focuses on Artificial Intelligence and Robotics. He is also a member of the British Computer Society.

DR JOÃO FILIPE FERREIRA (MIEEE, MBCS) received Ph.D., M.Sc. and B. Sc. degrees in Electrical Engineering from the Faculty of Sciences and Technology, University of Coimbra (FCTUC) in 2011, 2005 and 2000, respectively. He is currently a Senior Lecturer at Nottingham Trent University (NTU) and a researcher at the Computational Neuroscience and Cognitive Robotics Research Group (CNCR). He is a member of the IEEE and the IEEE Robotics and Automation Society (RAS) since 2012 (member of the Technical Committee on Cognitive Robotics, T-CORO, since 2015, and member of the Technical Committee on Agricultural Robotics, TC AgRA, since 2019). Additionally, he is a member of the IEEE Life Sciences Community since 2013, the IEEE Systems, Man, and Cybernetics Society since 2015 and the IEEE Computational Intelligence Society since 2015. He is also a member of the British Computer Society since 2019. His main research interests are artificial perception and cognitive robotics.

PROFESSOR T. MARTIN MCGINNITY (SMIEEE, FIET) received a First Class (Hons.) degree in Physics in 1975, and a Ph.D degree in 1979. He currently holds a part-time Professorship in both Nottingham Trent University (NTU), UK and Ulster University (UU). He is the author or coauthor of 350+ research papers and leads the Computational Neuroscience and Cognitive Robotics research group at NTU. His current research is focused on the development of biologically-compatible computational models of human sensory systems, including auditory signal processing; human tactile emulation; human visual processing; sensory processing modalities in cognitive robotics; and implementation of neuromorphic systems on electronics hardware.