JOURNAL OF AEROSPACE INFORMATION SYSTEMS
Vol. 21, No. 1, January 2024

Radiation-Induced Noise Resilience of Neuromorphic Architectures

Seth Roffe*
University of Pittsburgh, Pittsburgh, Pennsylvania 15260
Himanshu Akolkar†
University of Pittsburgh, Pittsburgh, Pennsylvania 15213
Alan George‡
University of Pittsburgh, Pittsburgh, Pennsylvania 15260
and
Ryad Benosman§
Carnegie–Mellon University, Pittsburgh Pennsylvania 15213-3890

https://doi.org/10.2514/1.I011192

Neuromorphic event-based networks use asynchronous time-dependent information to extract features from input data that can allow for edge-based distributed applications such as object recognition. The noise resilience properties of such networks, especially in the context of space applications, are yet to be explored. In this paper, we use the hierarchy of time surfaces (HOTS) algorithm, which is one of the neuromorphic algorithms, to understand the least and most resilient modules in a neuromorphic network. The HOTS algorithm relies on the computing of time surfaces that maps the temporal delays between neighboring pixels into normalized features that involve many computations that are also found in other neuromorphic networks such as exponential decays, distance computations, etcetera. We implemented HOTS on a Digilent PYNQ board with a Xilinx Zynq 7020 system on a chip, and we subjected the boards running the HOTS network inference to neutron radiation at the Los Alamos Neutron Science Center. Furthermore, we used simulation models from our previous similar experiments on the event-based sensor to create a neutron induced noise model to quantify the effect of this noise on the overall performance of the network. This experiment provides the preliminary measurements of the reliability of the HOTS algorithm and proposes methods to create a more reliable HOTS architecture in future spacecraft missions.

I. Introduction

FEATURE extraction is a fundamental part of object recognition in visual processing. A problem associated with this step is understanding how features should be characterized in an image. Moreover, as these complex algorithms move to hazardous environments such as space, their reliability in performing accurately needs to be considered and evaluated.

One feature extraction algorithm for neuromorphic event-driven image classification is the Hierarchy Of Time-Surfaces (HOTS) algorithm [1]. Neuromorphic event-driven sensors are a novel form of imaging device that offer a new paradigm of artificial vision as compared to traditional frame-based cameras. Recently, people have started to investigate using these event-driven sensors in space applications, whether it be on the ground in telescopes [2] or for use in space satellites [3,4]. However, space environments are prone to far more radiative noise than most ground environments and typically involve additional thermal challenges [5]. Meanwhile, satellite designers need to use commercial-off-the-shelf processors as opposed to traditional radiation-hardened processors in order to take advantage of their better performance and energy efficiency to make these complex machine learning and computer vision applications feasible to run on board. Therefore, satellite designers typically perform additional tests on their hardware and software to ensure their system can survive in a radiative environment.

Before event-based sensors can be safely used in conjunction with computer vision applications for onboard processing, the applications need to be tested under fault injection and radiation to ensure that the algorithm will perform accurately. Radiation can cause single-event upsets (SEUs) that can cause data errors, execution errors, or complete system failures, depending on what area of memory gets upset [6]. Silent data errors could lead to incorrect decisions made by an autonomous system without any indication to the users of a failure [6,7]. Thus, understanding the possible failure modes of each application before deployment becomes vital to mission success.

This research evaluates the HOTS algorithm on classification of the Neuromorphic Modified National Institute of Standards and Technology (N-MNIST) dataset [8] under a radiative environment and under simulation-based fault injections. Our experiments provide the initial evaluation of the reliability of the HOTS algorithm as a case study to understand how different computations within a neuromorphic network are affected by these upsets. The results of this research provide designers with the information needed to add protection to their mission, whether it be additional hardware protections or use of dependable computing techniques within software. This research also presents methods to create a more reliable neuromorphic architecture for use in a hazardous environment given the observations of the fault injection and irradiation.

Received 30 August 2022; revision received 17 July 2023; accepted for publication 18 August 2023; published online 9 October 2023.

*Department of Electrical and Computer Engineering, 4420 Bayard St, Suite 560; seth.roffe@nsf-shrec.org (Corresponding Author).

†Postdoctoral Researcher, Department of Neurobiology, Biomedical Science Tower 3, Fifth Avenue, 4420 Bayard St., Suite 560; himanshu.akolkar@pitt.edu.

‡Mickle Chair Professor and Department Chair, Department of Electrical and Computer Engineering, 4420 Bayard St., Suite 560; alan.george@pitt.edu.

§University of Pittsburgh, BioScience Tower 3, Fifth Avenue; benosman@pitt.edu.

II. Background

This section provides the background needed to understand this research. This includes event-driven sensors, the HOTS algorithm, and the use of radiation testing.

A. Neuromorphic Event-Based Sensors

Biomimetic event-based sensors [9] are novel types of vision sensors that are modeled after the mammalian retina. They are made

up of independent pixels that are driven by changes in light intensity in their field of view. This method provides an asynchronous stream of events that, in contrast to the traditional frame-based method of conventional cameras, also provides temporal context about the scene. The events can either have the polarity of on, where the light intensity of the pixel is increasing, or off, where the light intensity is decreasing. This local, asynchronous event method also provides the benefit of not containing a global shutter, providing neuromorphic sensors with a very high dynamic range. Owing to the lack of a global shutter and the asynchronous nature of the pixels, individual pixels only depend on the illumination of itself. This large dynamic range means that bright sources will not oversaturate the entire field of view, allowing the sensor to still provide useful information. Similarly, because changes in light intensity are being measured, any static, redundant background information is not passed through the sensor, which significantly reduces the data rate in sparse conditions as compared to when all information was passed, especially in cases where the camera is stationary [10]. Each pixel operates asynchronously from the others, producing a very high temporal resolution [11].

Most neuromorphic sensors output visual information about a scene in the form of discrete, distinct events using address-event representation (AER) [12–14]. AER encodes the visual information from the sensor in the form of four-tuples containing the x- and y-pixel coordinates where the event occurred, the time stamp in microseconds, and the polarity of light intensity (x, y, t, p).

B. HOTS Algorithm

Because neuromorphic vision sensing is an entirely new paradigm of imaging, the corresponding computer vision applications need to be adapted to this new paradigm. Instead of working with frames, as traditional algorithms use, new methods needed to be developed to account for event streams. For image classification, the HOTS algorithm was developed [1]. The HOTS algorithm uses the temporal context of a scene around an event to learn information about the object in the scene. Specifically, the HOTS algorithm tries to answer the question of how features can be extracted from an event stream. To answer this question, Lagorce et al. introduced the concept of time surfaces [1].

The general scheme of the HOTS algorithm is shown in Fig. 1. For a more detailed description of this method, refer to the work of Lagorce et al. [1]. Time surfaces are the features that are extracted from an event stream. For each event in the stream ($e = [x, y, p, t]$), a square neighborhood with a size of R pixels around the event pixel is chosen to create a time surface using the time of old events in the neighborhood. For each pixel in the neighborhood, past events are searched until an event of the same polarity as $\boldsymbol{e}$ is found. The difference between the time in the neighborhood pixel and the current event is mapped using an exponential kernel to create a surface of temporal structures giving context to each event, which can then be used to extract features from a scene.

This process is repeated over larger spatial and temporal scales, creating a hierarchical layered network. This network architecture thus consists of a "hierarchy" of time surfaces that builds and extracts features from a stream of events. Training this network architecture creates a model that can then be used as a pattern classifier on neuromorphic sensor data. The HOTS algorithm was chosen for the fault injection simulations because it incorporates many of the widely used components, such as exponential decays, in the event-based computation algorithms.

C. Fault Injection

Fault injection is the process of introducing controlled errors and noise in a system to study their impact on the resilience and performance of the system. For environments that are hard to experiment in, such as space, fault injection can be performed through modeling and simulations. For our experiments, we used two categories of fault injections: hardware and software. Hardware-based fault injection happens at a physical level, where faults are induced by disturbing the hardware from the environment, such as a radiation testing. This disturbance can manifest as faults in the low-level hardware components, such as state changes in transistors forming the memory structure of a processor that could lead to silent data errors [6].

Srour and McGarrity [6] detailed the effects of radiation on microelectronic circuits, such as damage, ionization, and single-event effects through radiation beam testing. Therefore, beam testing is popular in the field of space computing to classify SEUs of new systems and mitigation strategies. Knowledge of how an

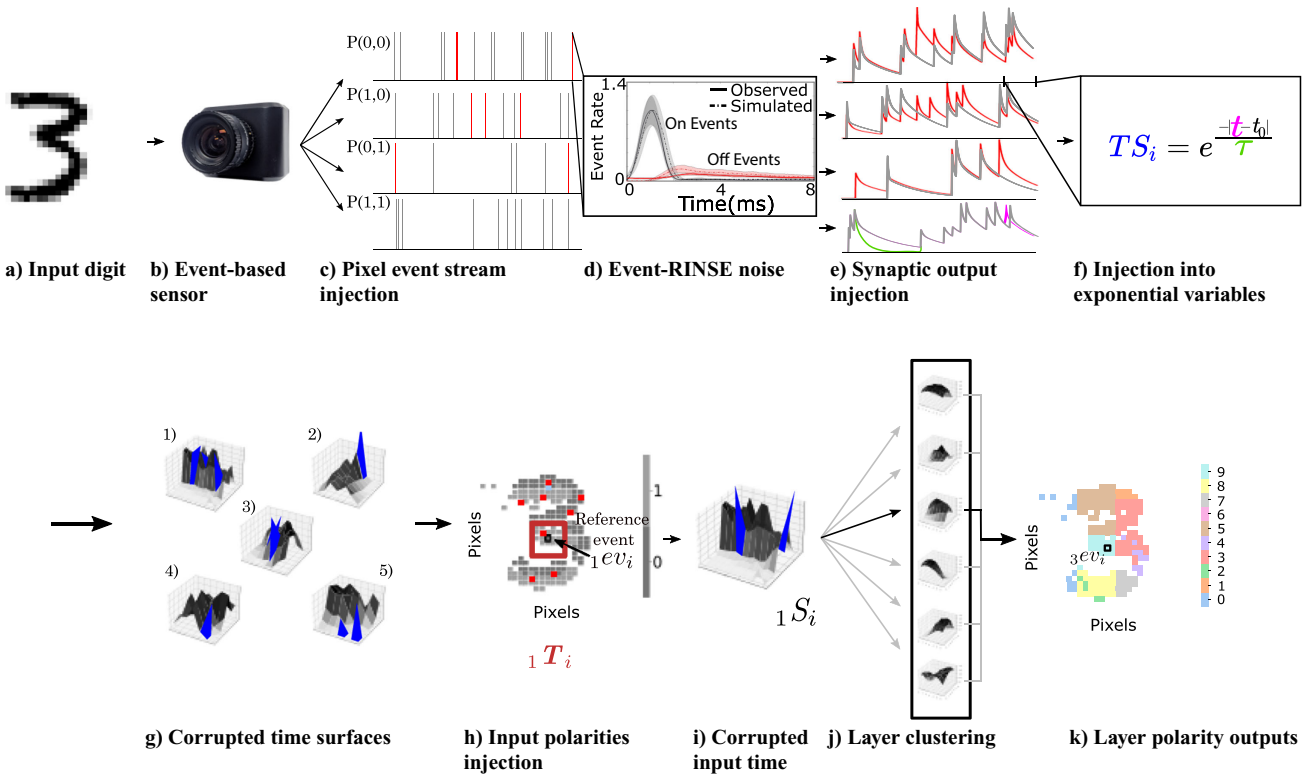

Fig. 1 Fault injection into HOTS classification architecture.

application will behave under radiation is critical to the design process for space missions, giving an overview of the upsets or faults a system might encounter.

Software-based fault injection can be performed by injecting errors at a variety of levels of abstraction to study their effects from low-level assembly instructions to high-level metainstructions. In this study, we focus on the high-level abstraction. Namely, the injection used consists of directly changing elements of data in software to observe the response of the technique under test. The effects of radiation on event-driven sensors have been measured and modeled in previous studies [15]. Our high-level fault injection model on the HOTS algorithm can be used to complement the previous study, allowing for analysis of the neuromorphic architecture starting from the input event data streams up to the final feature layer of the HOTS algorithm. Further details of the fault injections and their results are provided in Sec. V.B.

III. Related Work

This section discusses the related work on the topic of neuromorphic vision sensing. This includes the use of neuromorphic sensing for space domain awareness (SDA) and previous works done in testing the reliability of neuromorphic sensors to radiation.

A. Traditional Feature Selection

Feature selection for object recognition is a fundamental problem in visual processing. Traditionally, features are defined as a function of the static information around a neighborhood in an image [16,17]. The choice of the function determines the extracted features. Because conventional frame-based cameras primarily provide spatial information, information about temporal dynamics such as fast-moving objects in the scene is lost. The drawback of this loss of dynamic information is that frames are captured at artificially timed intervals (also known as the frame rate). These frames thus contain large amounts of redundant data if the background of an image does not change between frames.

Conventional feature extraction algorithms also typically assume that pixel illumination is the primary source of information. However, pixel luminance is not invariant to a scene [16]. Therefore, if any changes to the environment take place, such as changes in lighting conditions, the feature information will affect processing in conventional methods. Similarly, the low dynamic range of conventional cameras means that accurately measuring the luminance becomes difficult [18].

In machine learning, the primary method of image recognition has involved the use of convolutional neural networks (CNNs) ever since Krihevsky et al. won the ImageNet Challenge in 2012 [19]. What sets CNNs apart from other network types is that their layers make use of image convolution filters. Convolutional layers are composed by sliding different convolutional kernels across an image, projecting the information in the field onto a feature map. This feature map then provides local perception. In other words, areas that are close to a pixel are considered more relevant than those far away [20]. Different convolutional kernels can provide different pieces of information, such as an edge detection kernel. The problem with CNNs on embedded systems is that they are computationally complex; and convolving a filter over a large image, which may have redundant background information, becomes infeasible[21].

B. Space Domain Awareness

One particular application that the HOTS algorithm can be useful in is proximity operations for space domain awareness. Space domain awareness, previously known as space situational awareness, has been an important topic in military and civilian applications for many years [22–25]. SDA is defined as the perception of elements within a specific area in space, as well as the projection of their future states [26]. As an example of a use case for an event-based sensing architecture, optical systems play a part in "reactive SDA", where decisions are made after observing data on the field [26]. Understanding the movement of the surrounding area enables mission operators to avoid collision with potentially untracked debris and perform evasive maneuvers as needed. Vision systems with high temporal resolution and low latency would be beneficial to accurately detect objects for processing in real time.

Optical systems are particularly useful in proximity operations. These operations involve docking procedures that involve high-precision maneuvers that require real-time feedback [27]. Neuromorphic architectures, such as the HOTS algorithm, provide fast, asynchronous object classification of spacecraft components that could aid in autonomous docking procedures. Another potential use-case application is close-proximity observation, where a spacecraft moves around another object for observation [27]. Using their high temporal resolution, asynchronous event-based sensor classifications can potentially provide a faster feedback loop to the controller and optimize fuel consumption when searching for specific features, such as frame defects, on the observed system.

Because the technology is so new, the use of event-based sensors and neuromorphic algorithms in space-related applications is not very well developed. Most of the work has been carried out in the context of terrestrial telescope observation of low-brightness objects in low Earth orbit and geostationary Earth orbit. Reference [28] covers a series of experiments on using neuromorphic sensors through a fish-eye lens for all-sky observation. They show that the neuromorphic sensors are robust to difficult observing conditions and fast-moving objects. Afshar et al. [29] employed several tracking algorithms using a neuromorphic sensor fitted to a telescope. With their collected neuromorphic data, they were able to track high-speed objects moving across the field of view.

There has been extensive research into event-based cameras for real-time tracking and low-power computer systems. Many algorithms have been developed and evaluated for objects to be tracked within the visual space of an event-driven sensor. For example, Camunas-Mesa et al. were able to use neuromorphic sensors for real-time clustering and multitarget tracking, receiving an F accuracy of 95% while reducing the computational cost by 88% as compared to the conventional frame-based method [30]. Valeiras et al. developed a method for object tracking that can track many different shapes, so long as the pattern of the shapes is known a priori [31]. Similarly, Lagorce et al. provided a multikernel Gaussian mixture model tracker for the detection and tracking of different-shaped objects [32]. Another method uses spatial matching to allow objects to be tracked in even occluded conditions [33]. The low computational requirements of neuromorphic sensor data analysis even allow tracking systems to be implemented on embedded platforms [34] and Field-Programmable Gating Array (FPGAs) [33], making them perfect for space applications. Novel methods can even detect and track objects in conditions where both the camera and the objects are moving independently, as is the case for satellites [35–37].

C. Neuromorphic Sensors Under Radiation

To perform valid fault injection on the HOTS algorithm, a fault model is needed. The fault model developed in Ref. [15] was used to perform software fault injection into the data. Specifically, the introduced event-based radiation-induced noise simulation environment (event-RINSE) was useful in adding radiation-induced noise to prerecorded data streams. This added noise was then used as a fault injector, allowing the HOTS algorithm to be tested in a more controlled environment. In this experiment, the HOTS algorithm was tested for its sensitivity to noise.

IV. Scope and Methodology

This section will discuss the scope and methods to be used in the experiment. The methodology will cover software fault injection as well as the radiation experiment.

A. Scope

The operating system was a lightweight version of Linux, but the complications of Linux's monolithic kernel will be irrelevant to the reliability of the HOTS algorithm or the time surfaces. Moreover,

the operating system hosting the HOTS algorithm is arbitrary, and so this experiment does not focus on execution errors. That is, any errors involving segmentation faults, kernel panics, or similar errors are not studied because these depend on the underlying operating system.

This experiment primarily focuses on silent data errors. Silent data errors are defined here as any error causing a difference in the output of each layer. Each layer was individually analyzed and compared with the final predictions of the network. Observing the outputs of each layer enables observation on how sensitive the time surfaces are to noise. This observation also shows how robust the output predictions are to changes in the noise.

B. Methodology

This experiment took place in two distinct phases: the radiation experiment and the high-level fault injection. These two halves provide two separate measurements of reliability. Namely, the radiation experiment shows how robust the network architecture is to data errors, and the software fault injection shows how sensitive the HOTS algorithm is to noise in the input data and corruptions in the algorithm.

1. *Radiation Experiment*

The radiation experiments were performed at the Los Alamos Neutron Science Center (LANSCE) Weapons Neutron Research facility. The LANSCE provides a wide-spectrum neutron beam of energies ranging from ~1 MeV to greater than 10 MeV. Four hours of neutron beam testing at the LANSCE facility provided fluences similar to that of a 10 year mission on the International Space Station (ISS) [38]. Two Digilent PYNQ-Z2 boards, labeled in this research as PYNQ 0 and PYNQ 1 (Fig. 2), were used as the devices under test (DUTs) and were radiated while running the HOTS application for approximately four days, as allowed by the facility. This duration would be roughly equivalent to the fluence from a 240 year mission in the ISS orbit. The effective fluence on the DUTs was calculated by summing up the number of estimated neutrons passing through the DUT while it was powered, and dividing by a constant that drops off with the square of the distance from the dosimeter to account for any decay or absorption of the beam before reaching the target. This constant is defined in Ref. [39] as

$$\frac{(13.87\ \text{m})^2}{(13.87\ \text{m} + d)^2}$$

where 13.87 m is the distance from the beam source to the dosimeter, and d is the distance from the dosimeter to the DUT. In this experiment, d was 0.911 m for PYNQ 0 and 0.937 m for PYNQ 1.

The Neuromorphic Modified National Institute of Standards and Technology dataset [8], which consists of neuromorphic event streams of handwritten numerical digits (zero to nine), was used to test image classification with the HOTS algorithm. The HOTS architecture was trained prior to the radiation experiment using 2000 samples from the training dataset and 200 events per sample. Only one digit class was tested at a time to separate how data errors affect different classes, as opposed to using an average classification accuracy among all digits. Additionally, due to memory and computational constraints of the embedded platform, only the first 100 events of a sample were considered for inference. Due to the cyclic nature of the N-MNIST dataset, this did not affect the prediction accuracy significantly. The outputs of each layer as well as their respective time surfaces were logged for each execution. Three layers of time surfaces were used with neighborhood sizes of 7×7 pixels, 13×13 pixels, and 25×25 pixels around each event pixel to cover enough of the surrounding area to extract different features at each layer. Three layers were chosen, as was done in the original HOTS publication [1]. This output helps determine which layers are most sensitive to radiation effects. Similarly, the results show how robust the prediction outputs are to data errors. If there is an error in layer 1 but the prediction accuracy does not change, this shows that the layer network has intrinsic reliability associated with it. Any errors in the execution of the application, such as segmentation faults or kernel panics, were not included in this analysis. The results of the radiation experiment are discussed in Sec. V.A.

2. *Fault Injection Experiment*

Noise was added to the N-MNIST input data using the event-based radiation-induced-noise simulation environment described in Ref. [15]. This noise model was created using the actual radiation of an asynchronous time-based image sensor [9], and it is used here to insert noise into the N-MNIST event data. The noise was injected into the input data using a variable noise-event rate. Changing the event rate can be used to find how much noise needs to be added to affect the prediction accuracy. Fault injecting in software provides a much more controlled environment to test the reliability of the HOTS algorithm and the time surface's sensitivity to noise. The fault injection results are discussed in Sec. V.B.

The details of the fault injection can be seen in Fig. 1. Noise was injected into different modules of the HOTS algorithm independently to test their resilience. An input digit (Fig. 1a) is passed through the event-based sensor (Fig. 1b), which produces an asynchronous event stream for each pixel. Radiation-based noise injected into input events (Fig. 1c) modeled using the event-RINSE fault injector (Fig. 1d) described in Ref. [15] leads to erroneous spikes (in red) added to the event stream. This also leads to errors in the time surface computation, as shown in the top three rows in Fig. 1e. The bottom row of Fig. 1e further shows errors due to additional noise injected into the exponential kernel computation (Fig. 1f), such as the decay constant τ (green) and the time of the event t (in magenta): both of which lead to errors in the decayed synaptic output. The time surfaces themselves are also prone to noise (blue), which lead to erroneous time surface values for certain pixels shown in Fig. 1g. The polarities used in k-means clustering were also injected with bit flips, which are shown in Fig. 1h to see how random changes in cluster assignments affect the classification. The layer outputs (i.e., the labeled events) were then passed through a dense multilayered perceptron (MLP) as features for classification, which are demonstrated in Fig. 1k. The MLP classifier was a three-layer network consisting of an input layer with 24 nodes, which is the same size as the number of features of the last layer of the time surface hierarchy; a hidden layer with 200 nodes; and an output layer which is of size 10, which is the number of classes in the N-MNIST dataset. Each aspect of the algorithm was tested under fault injection using a simple Bernoulli test with a varying probability of injection from 10 to 90%. The targets included the time surface exponential values, the input time context, and the time decay constant, which are all highlighted in Fig. 1. All targets were injected

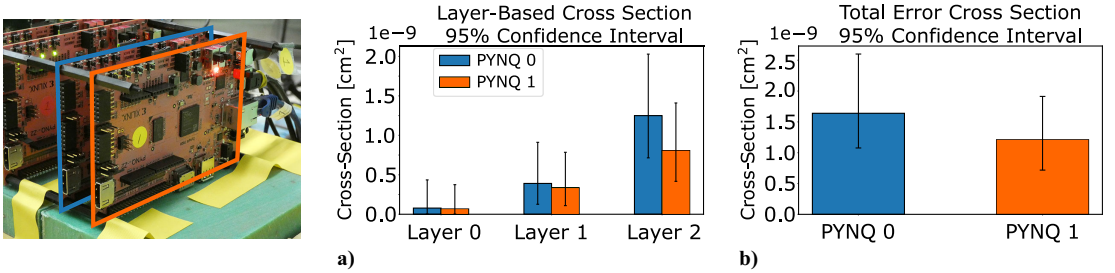

Fig. 2 Two PYNQ boards radiated at LANSCE and labeled as PYNQ 0 (blue) and PYNQ 1 (orange): a) cross sections when errors are separated by which layer the first error occurs at, b) total cross section containing the total number of errors in all layers.

into separate events to ensure a controlled response that simulated a single-event upset in memory.

C. Platform Selection

We used two separate platforms for our experiments. The radiation experiment was performed on the Digilient PYNQ-Z2 board, employing a Xilinx ZYNQ-7020 system on a chip (SOC), which has a dual-core ARM Cortex-A9 processor and an Artix-7 FPGA fabric. This SOC acts analogously to the Center for High-Performance and Reconfigurable Computing (CHREC) space processor (CSP) developed at the National Science Foundation's Center for Space, High-Performance and Resilient Computing (SHREC), which has been employed on multiple platforms aboard the International Space Station [40–42]. The SHREC also uses the CSP alongside a neuromorphic sensor on their Configurable and autonomous Sensor Processing Research mission, which is the primary reason why PYNQ is used as a device under test [3,4].

Because high-level fault injection was performed in software, the platform becomes irrelevant. Therefore, for software fault injection, a desktop computer employing an AMD Ryzen 5 3600X six-core processor was used. This platform is arbitrary and will not affect the experimental results.

V. Results

This section covers the results of the radiation experiment and insights from fault injection. The HOTS algorithm vulnerabilities are discussed herein.

A. Los Alamos Neutron Science Center Experiment

The primary purpose of the LANSCE radiation experiment is to measure the cross section of the HOTS algorithm on the Xilinx Zynq-7020 SOC. The cross section is defined as the vulnerable area on the chip where, if impacted by radiation, an error event is expected. However, the error events need to be defined in such a way that provides insight into the vulnerability of the algorithm. Error events were defined as differences from the calculated and expected time surfaces for every event in each of the three layers. The cross section, along with the 95% confidence interval, was calculated for each layer where an error event first arose, which is shown in Fig. 2a. Similarly, the total cross section and 95% confidence interval for all combined layers can be seen in Fig. 2b.

The 95% confidence interval seems to increase with each layer, as expected, because the time surface is larger in higher layers, leading to a higher chance of vulnerability. It is important to note, however, that despite these data error events, there were no events that caused a loss in prediction accuracy. Therefore, it is seen that the HOTS algorithm shows an intrinsic reliability as an application. Between the several feature points in a time surface, as well as the intrinsic reliability within the classifier, a small number of elements being affected in the time surface will most likely not cause any loss of accuracy. This result is consistent with Ref. [7], which showed intrinsic reliability within neural-network classifiers.

Furthermore, we classify the types of errors observed in the radiation experiment. There are three types of data errors that were observed: 1) silent data errors where the time surface amplitude falls within [0, 1], implying the error occurs before the exponential calculation; 2) silent data errors where the time surface amplitude does not fall within [0, 1], implying the error occurs after the exponential calculation; and 3) data errors where values are missing from the output time surface, implying the error occurs within the parameters of the architecture.

Figure 3a demonstrates an example of error type 1, where silent data errors occurred before the exponential decay calculation in the time surface generation. The residuals show two points in the observed, flattened time surface that differ from the expected result. Both cases are constrained within the range of [0, 1], which implies that the error caused an erroneous t value to be used in the calculation of the exponential decay: $e^{-(|t-t_0|/\tau)}$.

Figure 3b shows an example of a much more extreme data error. This figure shows cases of both error types 2 and 3. At around an

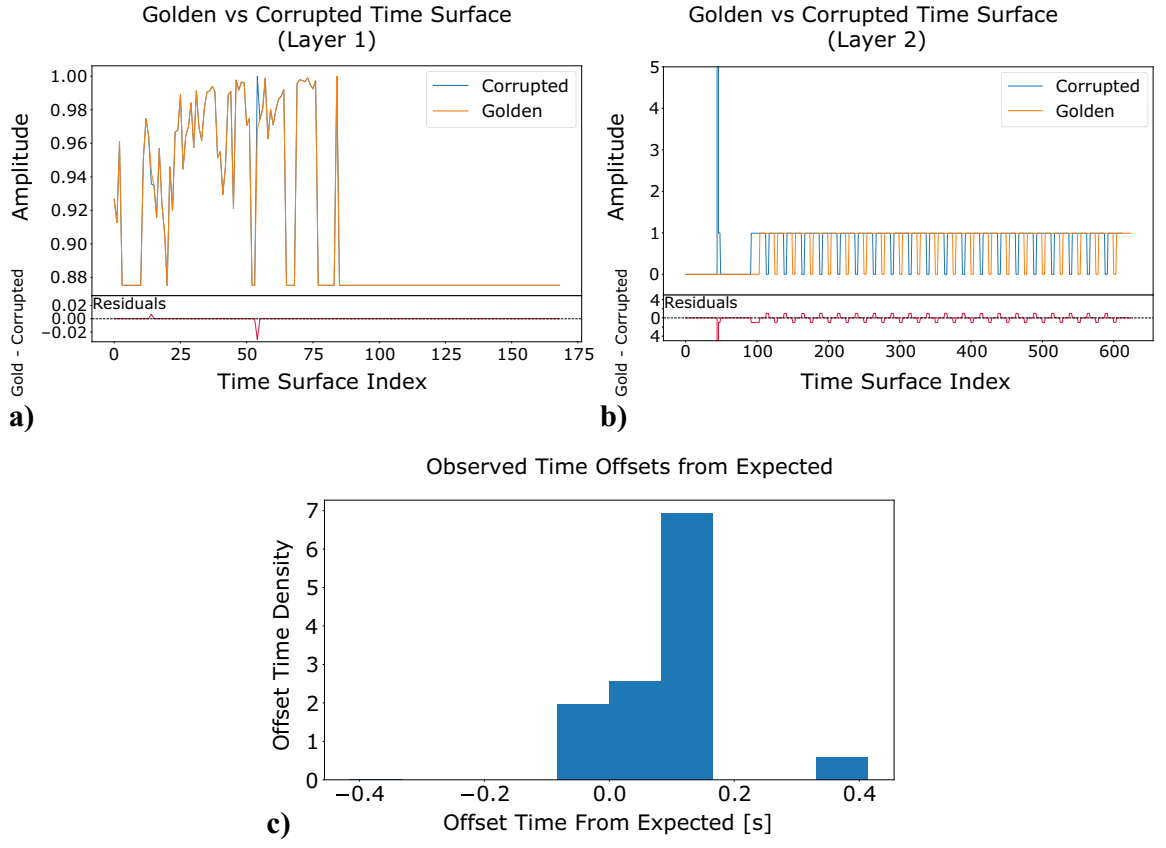

Fig. 3 Examples of data errors seen in time surfaces during irradiation. Time surfaces were flattened from two dimensions to one dimensions for easier visibility.

index of 50, the amplitude of the time surface jumps to about 5000, which is well out of the normal range of [0, 1] constrained by the exponential calculation. Therefore, it can be assumed that the error occurs in the value of the time surface after the exponential decay, and not in any of the data or parameters beforehand. Similarly, some values of the expected time surface were missing from the observed one, causing a constant offset from the expected pattern after an index of 100. Surprisingly, neither of these error events caused a drop in prediction accuracy, probably due to the fact that this time surface only represents one of many features in an event stream.

Figure 3c shows a histogram of the observed time offsets from the data errors. The majority of the time offsets were small, falling within ± 0.2 s. This small offset implies that the majority of data errors occurred before the exponential calculation, keeping the erroneous output bound within [0, 1], which was most likely due to the amount of computation time spent on the time surface exponential calculations. The majority of the runtime was spent calculating the time surface exponential values, leaving the data used in the calculation vulnerable to radiation. However, this bound from the calculation means that the majority of the output errors would not be significant enough to cause a change in the clustering output.

B. Fault Injection

The results from the event-RINSE fault injection can be seen in Fig. 4. The noise model for the event-RINSE consists of a burst of on events followed by a long relaxation period of off events, meaning that there are several hundred induced event spikes in each noise-event occurrence [15]. Induced noise rates were tested from zero to 100 noise occurrences per second. The cumulative average among all 10 classes of the N-MNIST dataset was shown to drop to a random chance at around 80 noise occurrences per second, as seen in Fig. 4a. However, even small amounts of radiation noise, such as five effects per second, cause a drop in accuracy, implying that the system is sensitive to variance in the input data. This sensitivity follows due to the creation of many random noise features created that will interfere with the classification, especially on a memory-constrained embedded system where not all events in an input file can necessarily be used for feature extraction.

Figure 4b separates the classes for digit "1" and digit "5" and compares them to the cumulative average. Classes with a diverse set of features, such as digit 5 show a higher resilience to the average, typically scoring around one standard deviation above the average. Conversely, classes without many features to extract, such as digit 1, consistently had an accuracy lower than one standard deviation below the average. This is most likely due to the random noise having a larger impact on the classification when there are fewer features to extract from a stream. However, in an actual radiative environment, it is unlikely to see significantly high enough noise rates for this data dependency to cause significant effects, except for extreme cases.

To control the response of high-level fault injections, four separate variables were individually targeted using software bit flips with varying probabilities of injection. The results of the software fault injection effects on the accuracy of the HOTS classification as compared to its

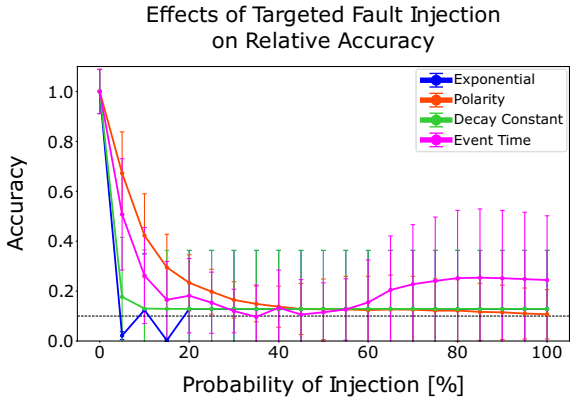

Fig. 5 Average accuracy across classes given different probabilities of bit flips in different variables. Error bars represent standard deviation.

accuracy without injection can be seen in Fig. 5. Injection into all four variables (the exponential, polarity, decay constant, and event time) caused the accuracy to drop to a random chance (10% accuracy) by 50% injected events. The exponential and the decay constant caused the sharpest decline in accuracy where the accuracy dropped to below 25%, even for 10% affected events. This is expected due to the fact that they drastically changed the shape of the time surface feature. Surprisingly, injecting into the polarities shows the slowest drop in accuracy despite directly affecting the k-means center clustering. However, this phenomenon could be due to the fact that the polarity injections only lead to reassignment from the original polarity to another polarity bounded within the number of clusters in each layer as used during training, leading to a lower probability of a data error occurring.

VI. Conclusions

The results of the current experiments demonstrate the reliability and vulnerabilities of feature extraction in neuromorphic data. The response of the hierarchy of time surface architecture to radiation as well as the response of time surface features to targeted fault injection were measured. Within the radiation experiment, no change in accuracy was seen due to any data errors observed within the time surface layers. The classifier used the output of the hierarchy of time surfaces, which means that only errors propagating to the output of the hierarchy could have any possible effect on the accuracy. However, as observed in Ref. [7], multilayered perceptrons have intrinsic resilience to data errors within the nodes, providing an overall boost to the reliability of the HOTS architecture. It is also likely that the large number of features provided by creating a time surface for every event in a data file as compared to the low chance for a malicious error in memory to occur during computation gives a higher resilience to a layer being affected. Similar results have been found in recent work by Nagarajan et al., where they studied the effect of noise injection in neurons in a spiking neural network performing digit classification on the N-MNIST dataset that could be comparable to the current

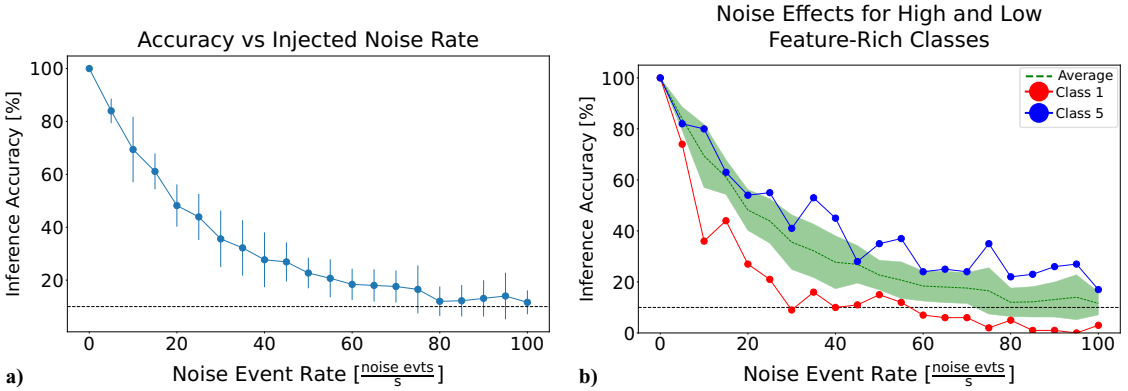

Fig. 4 Average HOTS accuracy performance for different injected noise rates. Error bars represent standard deviation.

work. The authors also found a significant drop (85.6%) in accuracy of the spiking neural network with the addition of noise. A similar study on the TrueNorth neuromorphic processor simulating a spiking neural network was reported by Brewer et al. They also showed the resilience of the spiking neural network against proton-induced single events.

Within the data errors, the time offsets were then calculated, which are shown in Fig. 3c. This figure shows a histogram of how radiation tended to shift the temporal spikes calculated through inversion of the exponential function. The majority of the calculated offsets are positive after the natural logarithm of the exponential multiplied by the negative of the decay constant. This provides evidence that the majority of the errors encountered in the radiation test occurred before or during the calculation of the exponential. This is most likely due to the fact that the creation of the time surfaces is the most computationally complex part of the algorithm. Once the time surfaces, and thus the exponential values, are calculated, the rest of the inference is quick, leaving little chance for a radiation strike to corrupt the data. The most time-consuming parts of an algorithm will generally be the most vulnerable to radiation because the time where an error may occur is larger.

To combat the randomness of a radiation experiment, controlled fault injection in software was also performed to find the most vulnerable parts of the HOTS architecture. The event-RINSE allows observations on how the classification accuracy decreases when there is corruption in the sensor data. Data errors within the algorithm can be detected or corrected via redundancy, but it becomes difficult to detect problems if the sensor providing the input itself is affected. Figure 4a shows that the accuracy of the HOTS algorithm does decrease with increased radiation noise within the input event stream. However, Fig. 4b shows that the drop in accuracy is dependent on the feature map of the data and is, therefore, data dependent. It is likely that the radiation response will be different under a different dataset. It should also be noted that due to the memory and computational constraints of embedded systems, all events in an input file were unable to be used to create time surfaces. Increasing the number of events used in each sample would most likely provide more resilience to input noise by increasing the signal-to-noise ratio. Additionally, simply knowing the radiation noise model opens up opportunities for filtering or noise reduction before processing.

High-level fault injection shows how faults within the onboard memory or cache can propagate to data errors. The primary targets for data errors within the architecture would be the exponential amplitude, the decay constant τ, the event time spike t, and the polarity used in clustering. To ensure that the results were unaffected by changes in the testing datasets, the training dataset was used for validation to ensure a 100% prediction accuracy without fault injection. To understand the slope of accuracy decay, the accuracy with fault injection was compared to the accuracy without fault injection. All four variables were shown to drop the inference accuracy very quickly. The exponential value and the decay constant were shown to have the sharpest decline in accuracy with the probability of injection. This sharp decline is most likely due to those variables drastically changing the overall feature shape of the time surface, which would lead to significant changes in clustering. The decay constant significantly impacts the shape of the feature because it affects all points in the output as opposed to one pixel in the time surface. Meanwhile, injections in the exponential were postcalculation of the exponential amplitude, meaning the erroneous value was not bounded by [0, 1], causing severe changes in the feature shape, such as is the case in Fig. 3b. However, the radiation experiment showed that this case is unlikely. Interestingly, the polarity that defines the features used for classification had the least sharp decline in accuracy. To prevent crashes, the polarity bit flips had to be bound to the number of features in each layer, which means there is a lower probability of the injected polarity being different from the expected one. With an actual radiation experiment, it is much more likely that a bit flip in the polarity variable would cause a segmentation fault, which was not accounted for in this experiment because they are platform dependent. The event time shows the second shallowest decline of the accuracy next to the polarity. The event time, as it is used in the exponential values, would not lead to as strong as a response as the others due to the fact that the exponential calculation masks the error by keeping it bound within [0, 1], and thus does not cause changes that are as severe in the output, similar to Fig. 3a.

In all variable cases, however, the accuracy still drops to a random chance before a 50% probability of injection, demonstrating a severe vulnerability to SEUs in these components. To combat this, traditional reliable computing techniques can be used. In the case of the exponential, a solution to prevent vulnerability at the cost of memory or runtime can be to calculate the value three or more times and take a majority vote on the output, similar to triple modular redundancy. Another option for error detection is to have a quick boundary check along the time surface. Due to the nature of the exponential, any values that are larger than one or less than zero would be impossible, and thus would be a data error. Similarly, the polarity calculation can also be performed multiple times and put to a vote. Multiple copies of the decay constant can be stored and voted on before any computation, allowing for redundancy with minimal time and memory overhead. Unfortunately, there is no ground truth to the event time spike during computation, and so any redundancy would have to occur in the input data files. However, the probability of any of these values being impacted by radiation causing a decline in accuracy is small. Although, adding additional redundancies and checks in these variables is a possible way to improve reliability for mission-critical software with little overhead.

The overall results of these experiments can be used to infer the response to radiation of other neuromorphic applications, such as any method using an exponential kernel. With the radiation experiment, it is shown that there is intrinsic reliability within the HOTS application, where no decline in inference accuracy is seen, even with errors in the layers. Fault injections show how propagating computational errors can lead to failures in inference. Both of these experiments together can give mission designers a starting point to creating the architectures for reliable, neuromorphic applications.

Acknowledgments

This research was supported by the Center for Space, High-Performance and Resilient Computing industry and agency members and by the Industry-University Cooperative Research Centers Program of the National Science Foundation under grant no. CNS-1738783. This work was performed, in part, at the Los Alamos Neutron Science Center, which is a National Nuclear Security Administration user facility operated for the U.S. Department of Energy by Los Alamos National Laboratory (contract 89233218CNA000001).

References

[1] Lagorce, X., Orchard, G., Galluppi, F., Shi, B. E., and Benosman, R. B., "HOTS: A Hierarchy of Event-Based Time-Surfaces for Pattern Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 39, No. 7, 2017, pp. 1346–1359. https://doi.org/10.1109/TPAMI.2016.2574707

[2] Cohen, G., Afshar, S., and Schaik, A. V., "Approaches for Astrometry Using Event-Based Sensors," *Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS)*, AMOS Tech, Kihei, Hawaii, 2018.

[3] Roffe, S., Schwarz, T., Cook, T., Perryman, N., Goodwill, J., Gretok, E., Phillips, A., Moran, M., Garrett, T., and George, A., "CASPR: Autonomous Sensor Processing Experiment for STP-H7," *34th Annual AIAA/USU Conference on Small Satellites*, 2020, https://digitalcommons.usu.edu/smallsat/2020/all2020/64/.

[4] Perryman, N., Schwarz, T., Cook, T., Roffe, S., Gillette, A., Gretok, E., Garrett, T., Sabogal, S., George, A., and Lopez, R., "STP-H7-CASPR: A Transition from Mission Concept to Launch," *35th Annual AIAA/USU Conference on Small Satellites*, 2021, https://digitalcommons.usu.edu/smallsat/2021/all2021/241/.

[5] Badhwar, G. D., and O'Neill, P. M., "An Improved Model of Galactic Cosmic Radiation for Space Exploration Missions," *International Journal of Radiation Applications and Instrumentation. Part D. Nuclear Tracks and Radiation Measurements*, Vol. 20, No. 3, 1992, pp. 403–410. https://doi.org/10.1016/1359-0189(92)90024-P

[6] Srour, J. R., and McGarrity, J. M., "Radiation Effects on Microelectronics in Space," *Proceedings of the IEEE*, Vol. 76, No. 11, 1988, pp. 1443–1469.
https://doi.org/10.1109/5.90114

[7] Roffe, S., "Evaluation of Algorithm-Based Fault Tolerance for Machine Learning and Computer Vision Under Neutron Radiation," Univ. of Pittsburgh ETD, 2020, http://d-scholarship.pitt.edu/38392/ [retrieved 29 July 2020].

[8] Orchard, G., Jayawant, A., Cohen, G. K., and Thakor, N., "Converting Static Image Datasets to Spiking Neuromorphic Datasets Using Saccades," *Frontiers in Neuroscience*, Vol. 9, Nov. 2015, Paper 437.

[9] Posch, C., Matolin, D., and Wohlgenannt, R., "A QVGA 143 dB Dynamic Range Frame-Free PWM Image Sensor with Lossless Pixel-Level Video Compression and Time-Domain CDS," *IEEE Journal of Solid-State Circuits*, Vol. 46, No. 1, 2010, pp. 259–275.

[10] Posch, C., Serrano-Gotarredona, T., Linares-Barranco, B., and Delbruck, T., "Retinomorphic Event-Based Vision Sensors: Bioinspired Cameras with Spiking Output," *Proceedings of the IEEE*, Vol. 102, No. 10, 2014, pp. 1470–1484.

[11] Benosman, R., Ieng, S. H., Clercq, C., Bartolozzi, C., and Srinivasan, M., "Asynchronous Frameless Event-Based Optical Flow," *Neural Networks*, Vol. 27, March 2012, pp. 32–37.
https://doi.org/10.1016/J.NEUNET.2011.11.001

[12] Mahowald, M., "VLSI Analogs of Neuronal Visual Processing: A Synthesis of Form and Function," Ph.D. Thesis, California Inst. of Technology, Pasadena, CA, 1992.
https://doi.org/10.7907/Z9CZ35CD

[13] Boahen, K. A., "Point-to-Point Connectivity Between Neuromorphic Chips Using Address Events," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, Vol. 47, No. 5, 2000, pp. 416–434.
https://doi.org/10.1109/82.842110

[14] Lazzaro, J., and Wawrzynek, J., "A Multi-Sender Asynchronous Extension to the AER Protocol," *Proceedings of the 16th Conference on Advanced Research in VLSI*, 1995, pp. 158–169.
https://doi.org/10.1109/ARVLSI.1995.515618

[15] Roffe, S., Akolkar, H., George, A. D., Linares-Barranco, B., and Benosman, R. B., "Neutron-Induced, Single-Event Effects on Neuromorphic Event-Based Vision Sensor: A First Step and Tools to Space Applications," *IEEE Access*, Vol. 9, May 2021, pp. 85,748–85,763.
https://doi.org/10.1109/ACCESS.2021.3085136

[16] Lowe, D. G., "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, Vol. 60, Nov. 2004, pp. 91–110.
https://doi.org/10.1023/B:VISI.0000029664.99615.94

[17] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P., "Gradient-Based Learning Applied to Document Recognition," *Proceedings of the IEEE*, Vol. 86, No. 11, 1998, pp. 2278–2323.
https://doi.org/10.1109/5.726791

[18] Reinhard, E., Heidrich, W., Debevec, P., Pattanaik, S., Ward, G., and Myszkowski, K., *High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting*, Morgan Kaufmann, San Mateo, CA, 2010.

[19] Krizhevsky, A., Sutskever, I., and Hinton, G. E., "ImageNet Classification with Deep Convolutional Neural Networks," *Communications of the ACM*, Vol. 60, No. 6, 2017, pp. 84–90.
https://doi.org/10.1145/3065386

[20] Yang, J., and Li, J., "Application of Deep Convolution Neural Network," *2017 14th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, 2017, pp. 229–232.
https://doi.org/10.1109/ICCWAMTIP.2017.8301485

[21] Qiu, J., Wang, J., Yao, S., Guo, K., Li, B., Zhou, E., Yu, J., Tang, T., Xu, N., Song, S., and Wang, Y., "Going Deeper with Embedded FPGA Platform for Convolutional Neural Network," *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, Assoc. for Computing Machinery, New York, 2016, pp. 26–35.
https://doi.org/10.1145/2847263.2847265

[22] Oltrogge, D. L., and Alfano, S., "The Technical Challenges of Better Space Situational Awareness and Space Traffic Management," *Journal of Space Safety Engineering*, Vol. 6, No. 2, 2019, pp. 72–79.
https://doi.org/10.1016/J.JSSE.2019.05.004

[23] Kennewell, J. A., and Vo, B.-N., "An Overview of Space Situational Awareness," *Proceedings of the 16th International Conference on Information Fusion*, 2013, pp. 1029–1036.

[24] Gasparini, G., and Miranda, V., "Space Situational Awareness: An Overview," *Studies in Space Policy*, Vol. 4, Springer, New York, 2010, pp. 73–87.
https://doi.org/10.1007/978-3-211-99653-9_7

[25] Pelton, J. N., "A Path Forward to Better Space Security: Finding New Solutions to Space Debris, Space Situational Awareness and Space Traffic Management," *Journal of Space Safety Engineering*, Vol. 6, No. 2, 2019, pp. 92–100.
https://doi.org/10.1016/J.JSSE.2019.04.005

[26] Holzinger, M., and Jah, M., "Challenges and Potential in Space Domain Awareness," *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 1, 2018, pp. 15–18.

[27] Carrico, T., Langster, T., Carrico, J., Vallado, D., Loucks, M., and Alfano, S., "Proximity Operations for Space Situational Awareness," *Advanced Maui Optical and Space Surveillance Technologies Conference*, Ansys Government Initiatives (AGI), PA, 2006, pp. 1–11.

[28] Ralph, N., Maybour, D., Bethi, Y., Cohen, G., Ralph, N., Maybour, D., Bethi, Y., and Cohen, G., "Observations and Design of a New Neuromorphic Event-Based All-Sky and Fixed Region Imaging System," *Advanced Maui Optical and Space Surveillance Technologies Conference*, 2019, Paper 71.

[29] Afshar, S., Nicholson, A. P., Schaik, A. V., and Cohen, G., "Event-Based Object Detection and Tracking for Space Situational Awareness," *IEEE Sensors Journal*, Vol. 20, No. 24, 2020, pp. 15,117–15,132.
https://doi.org/10.1109/JSEN.2020.3009687

[30] Camunas-Mesa, L. A., Serrano-Gotarredona, T., Ieng, S. H., Benosman, R., and Linares-Barranco, B., "Event-Driven Stereo Visual Tracking Algorithm to Solve Object Occlusion," *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 29, No. 9, 2018, pp. 4223–4237.
https://doi.org/10.1109/TNNLS.2017.2759326

[31] Valeiras, D. R., Lagorce, X., Clady, X., Bartolozzi, C., Ieng, S. H., and Benosman, R., "An Asynchronous Neuromorphic Event-Driven Visual Part-Based Shape Tracking," *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 26, No. 12, 2015, pp. 3045–3059.
https://doi.org/10.1109/TNNLS.2015.2401834

[32] Lagorce, X., Meyer, C., Ieng, S. H., Filliat, D., and Benosman, R., "Asynchronous Event-Based Multikernel Algorithm for High-Speed Visual Features Tracking," *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 26, No. 8, 2015, pp. 1710–1720.
https://doi.org/10.1109/TNNLS.2014.2352401

[33] Linares-Barranco, A., Gomez-Rodriguez, F., Villanueva, V., Longinotti, L., and Delbruck, T., "A USB3.0 FPGA Event-Based Filtering and Tracking Framework for Dynamic Vision Sensors," *Proceedings of the IEEE International Symposium on Circuits and Systems*, Vol. 2015, IEEE Publ., Piscataway, NJ, July 2015, pp. 2417–2420.
https://doi.org/10.1109/ISCAS.2015.7169172

[34] Litzenberger, M., Posch, C., Bauer, D., Belbachir, A. N., Schön, P., Kohn, B., and Garn, H., "Embedded Vision System for Real-Time Object Tracking Using an Asynchronous Transient Vision Sensor," *IEEE 12th Digital Signal Processing Workshop and 4th IEEE Signal Processing Education Workshop*, IEEE Publ., Piscataway, NJ, 2006, pp. 173–178.

[35] Ni, Z., Bolopion, A., Agnus, J., Benosman, R., and Régnier, S., "Asynchronous Event-Based Visual Shape Tracking for Stable Haptic Feedback in Microrobotics," *IEEE Transactions on Robotics*, Vol. 28, No. 5, 2012, pp. 1081–1089.
https://doi.org/10.1109/TRO.2012.2198930

[36] Ramesh, B., Zhang, S., Lee, Z. W., Gao, Z., Orchard, G., and Xiang, C., "Long-Term Object Tracking with a Moving Event Camera," *British Machine Vision Conference (BVMC)*, Sept. 2018, p. 241.

[37] Mitrokhin, A., Fermuller, C., Parameshwara, C., and Aloimonos, Y., "Event-Based Moving Object Detection and Tracking," *IEEE International Conference on Intelligent Robots and Systems*, IEEE Publ., Piscataway, NJ, 2018, pp. 6895–6902.
https://doi.org/10.1109/IROS.2018.8593805

[38] Reddell, B., O'Neill, P., Bailey, C., and Nguyen, K., "Single Event Effects Testing for Low Earth Orbit Missions with Neutrons," *2015 IEEE Nuclear and Space Radiation Effects Conference (NSREC 2015)*, IEEE Publ., Piscataway, NJ, 2015.

[39] Wender, S., and Dominik, L., "Los Alamos High-Energy Neutron Testing Handbook," *SAE International Journal of Advances and Current Practices in Mobility*, Vol. 2, No. 3, 2020, pp. 1279–1302.
https://doi.org/10.4271/2020-01-0054

[40] Rudolph, D., Wilson, C., Stewart, J., Gauvin, P., George, A., Lam, H., Crum, G., Wirthlin, M., Wilson, A., and Stoddard, A., "CSP: A Multifaceted Hybrid Architecture for Space Computing," *28th Annual AIAA/USU Conference on Small Satellites*, 2014.

[41] Sabogal, S., Gauvin, P., Shea, B., Sabogal, D., Gillette, A., Wilson, C., George, A., Crum, G., Barchowsky, A., and Flatley, T., "SSIVP: Spacecraft Supercomputing Experiment for STP-H6," *31st Annual AIAA/USU Conference on Small Satellites*, 2017.

[42] Wilson, C., Stewart, J., Gauvin, P., MacKinnon, J., Coole, J., Urriste, J., George, A., Crum, G., Timmons, E., Beck, J., Flatley, T., Wirthlin, M., Wison, A., and Stoddard, A., "CSP Hybrid Space Computing for STP-H5/ISEM on ISS," *29th Annual AIAA/USU Conference on Small Satellites*, 2015.

D. Casbeer
Associate Editor