IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 25, 2019, accepted September 14, 2019, date of publication September 23, 2019, date of current version October 4, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2943160

Bio-Inspired Stereo Vision Calibration for Dynamic Vision Sensors

MANUEL J. DOMÍNGUEZ-MORALES[1], ÁNGEL JIMÉNEZ-FERNÁNDEZ[1], GABRIEL JIMÉNEZ-MORENO[1], CRISTINA CONDE[2], ENRIQUE CABELLO[2], AND ALEJANDRO LINARES-BARRANCO[1]

[1]Robotic and Computer Technology Lab (RTC), E.T.S. Ingeniería Informática, University of Seville, 41012 Seville, Spain
[2]Face Recognition and Artificial Vision Group (FRAV), Rey Juan Carlos University, 28933 Madrid, Spain

Corresponding author: Manuel J. Domínguez-Morales (mjdominguez@us.es)

This work was supported by the Excellence Project from the Spanish Government Grant (with support from the European Regional Development Fund) COFNET (TEC2016-77785-P) and by the Spanish Government MINECO Project Bioinpad (TIN2016-80644-P).

ABSTRACT Many advances have been made in the field of computer vision. Several recent research trends have focused on mimicking human vision by using a stereo vision system. In multi-camera systems, a calibration process is usually implemented to improve the results accuracy. However, these systems generate a large amount of data to be processed; therefore, a powerful computer is required and, in many cases, this cannot be done in real time. Neuromorphic Engineering attempts to create bio-inspired systems that mimic the information processing that takes place in the human brain. This information is encoded using pulses (or spikes) and the generated systems are much simpler (in computational operations and resources), which allows them to perform similar tasks with much lower power consumption, thus these processes can be developed over specialized hardware with real-time processing. In this work, a bio-inspired stereo-vision system is presented, where a calibration mechanism for this system is implemented and evaluated using several tests. The result is a novel calibration technique for a neuromorphic stereo vision system, implemented over specialized hardware (FPGA - Field-Programmable Gate Array), which allows obtaining reduced latencies on hardware implementation for stand-alone systems, and working in real time.

INDEX TERMS Neuromorphic engineering, dynamic vision sensor, bio-inspired systems, stereo vision, calibration.

I. INTRODUCTION

Nowadays, humankind has experienced great advances in the field of computer vision. This advancement is, among other aspects, due to the use of more than one camera to collect data from the scene [1]–[3]. A two-camera system (stereo vision) is usually employed when attempting to mimic human vision. A stereo vision system allows achieving high-level goals, such as distance estimation or 3D positioning for different applications, e.g., autonomous robot navigation [4], [5]. Video processing in stereo vision covers many stages: from the calibration of the cameras to the outcome [6], such as distance measurements or 3D reconstruction [7]. Each step works with frames, processing them pixel by pixel until the desired pattern is found, or until the treatment that the system is focused on is completed [8].

The associate editor coordinating the review of this manuscript and approving it for publication was Jiachen Yang.

All applications for these systems need more than one camera and a concrete algorithm to combine the information obtained from all of them. The mechanism that performs a spatial alignment to obtain a common coordinate system is called calibration. It is the first step towards computational computer vision. Although some information concerning the measuring of scenes can be obtained by using non-calibrated cameras [9], calibration is essential when metric information is required. The use of precisely calibrated cameras allows measuring distances in a real world from their projections on the image plane [10], [11].

In a calibration procedure, there are two important aspects: camera model and calibration mechanism. A camera model explains how a 3D point is projected onto the image plane (2D) of a camera, and there are two main camera models: Pin-hole model and Gaussian model [12], [13]. Pin-hole is the most used camera model, due to its simplicity, which is why this work is focused on it.

The final objective is to obtain the projection matrix for each vision sensor (P), which determines the relation between a 3D point and a 2D point in this sensor. In a multi-camera system, all the projection matrices (from each of the vision sensors) can be combined to obtain the inverse projection matrix, used to estimate the point spatial situation with the information received from all the cameras. P is determined using images from the scene and its correspondences in the camera projections with a calibration grid [11]: commonly adopted methods are those of Tsai [14], Heikkila [15], Zhang [16], Ahmed *et al.*[17], Batista *et al.* [18], Faugeras [10] and the so-called linear method [10], among others. They are all based on the pin-hole camera model. In this work, the applied calibration technique is the one designed by Faugeras, which consists in a lineal approximation and an optimization phase.

The required computational power and speed make it difficult to develop a real-time autonomous system. However, brains perform powerful and fast vision processing using millions of small and slow cells working in parallel in a totally different way. Vision sensing and object recognition in brains are not processed frame by frame, but in a continuous way, i.e., spike by spike (a spike is an electronic pulse produced in the brain by neurons), in the cerebral cortex.

As a result, Neuromorphic Engineering emerges, with the aim of mimicking the neuron communications that take place in the brain, using several data coding systems. To mimic the whole biological system, neuromorphic engineers have developed a Dynamic Vision Sensor (DVS) [19] that emulates a biological retina. These sensors use Address-Event-Representation (AER) [20] for coding and transmitting the information using pulses (spikes), in the same way as neurons do. For real-time solutions, direct hardware implementations of these models are required, such as [21], where a retinal ganglion cell is modeled on FPGA.

There is a community of AER protocol users for bio-inspired applications in vision and audition systems. One of the goals of this community is to build large multi-chip and multi-layer hierarchically structured systems capable of performing complicated array data processing in real time [22]–[28]. The power of these systems can be used in computer-based systems under co-processing. The information in a DVS-AER system is a continuous flow that cannot be stopped: each spike contains information from a single pixel. Moreover, the intensity of a pixel dimension is encoded in the spike frequency received from that concrete pixel.

An AER retina, taking a load of 10% in the intensity of the pixels (256 values max.), transmits more than 1.9 million pulses to describe the current state of the scene (equivalent to 1 digital frame – 40ms considering 25fps). Using two retinas, this traffic flow is duplicated, which makes it very difficult to process that amount of data in real time. A fully parallel processing device with low latency and compatible for embedded applications would be optimal. It becomes essential to reduce the volume of data or to include a mechanism that simplifies the stereo vision processing step.

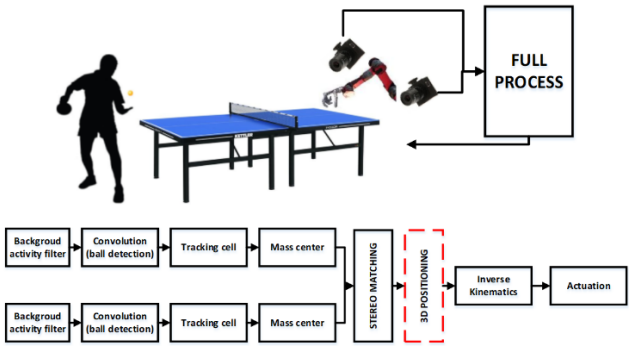

FIGURE 1. Global project vision (up) and process description (down).

In previous works, our research group developed some stereo AER algorithms to solve typical classical vision problems, such as the matching process or distance calculation [29]–[31]. However, these algorithms were not free of noise, due to inconsistencies resulting from the use of this kind of retina (relatively low resolution and temporal mismatching between them) and the AER codification itself.

This work is part of a project whose aim, as an example application, is to develop a system that can play table-tennis with a human, using a Neuromorphic stereo vision system (see Figure 1).

For this project, besides other hardware possibilities oriented to spiking systems, such as SpiNNaker [32], TrueNorth [33] or Loihi [34], among others, the full process is developed over an FPGA, because of its reduced cost, its fully parallel capabilities and its adequacy for AER systems [23], [26], [27], [34], which allows obtaining reduced latencies and stand-alone applications at lower expenses. Our proposal is composed of several processing steps. Steps 1 to 4 (see Figure 1) are developed thanks to the work of Tapiador-Morales *et al.* [35], [36] and Gómez-Rodríguez *et al.* [37], [38]. The stereo matching step has been studied in previous works by our research group [39]–[41] and by others [42]–[45]. The actuation step is implemented thanks to the work of Jimenez-Fernandez *et al.* [22], [46]. Finally, this work is focused on the 3D positioning step that depends on the calibration mechanism.

The rest of the paper is organized as follows: first, the hardware system used is presented. After that, a novel calibration implementation over a bio-inspired stereo vision system is described. Then, this mechanism is modified to be integrated into the real-time hardware system. Next, the system is tested using two different scenarios. Finally, the conclusions are presented.

II. HARDWARE SYSTEM

A DVS-AER retina only sends spikes when a brightness change is detected; therefore, if there is no change in the scene luminosity, no spikes are sent. This makes it difficult to use classical calibration models with a typical calibration grid. Usually, these retinas are used in scenarios with moving objects to obtain information about them, although in a

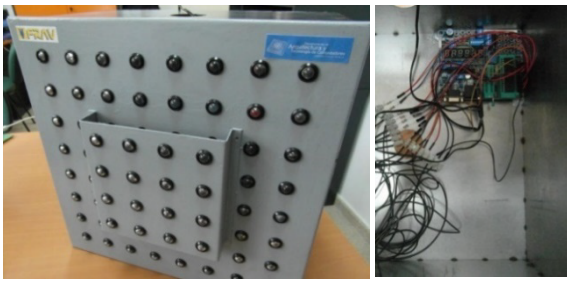

FIGURE 2. Left: LED grid for the calibration process (64 leds in two depths). Right: PIC microcontroller for controlling flashing LED lights.

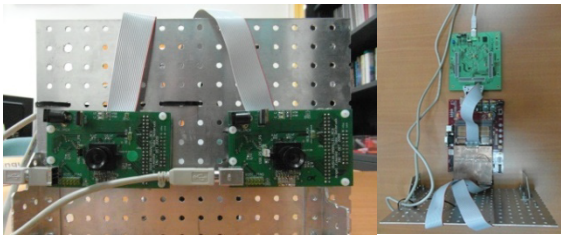

FIGURE 3. Left: Stereo AER system (two DVS AER retinas). Right: Full system (stereo DVS AER, Virtex5 FPGA and USBAERmini2).

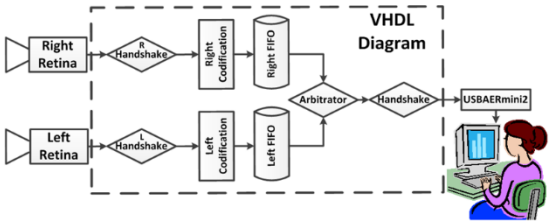

FIGURE 4. System block diagram (offline calibration): DVS retinas, Virtex-5 FPGA VHDL diagram, USBAERmini2 board and the computer.

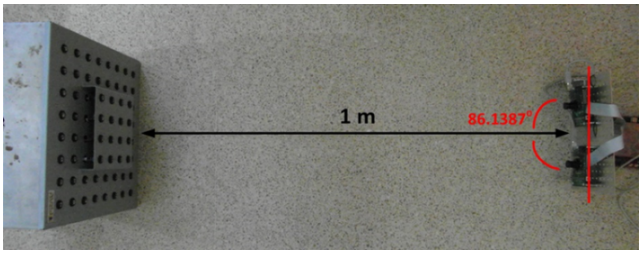

FIGURE 5. Complete system. LED grid placed 1 meter away from the retina. Retina situated in an angle of 86.1387° (as calculated and used in previous work [39]).

calibration step it is recommended to use motionless objects in order to obtain a good data accuracy.

However, a modification of the calibration grid can be done to stimulate AER DVS-retinas. The modification consists of a flashing LEDs (Light-Emitting Diodes) grid with two different depths. With the flashing LEDs, the retinas can receive information of the 3D points and as a result any calibration model can be applied. The flashing LEDs matrix was built specifically for this work using sixty-four LEDs and a PIC microcontroller that switches the LEDs on and off. This LED grid is shown in Figure 2.

For this work, a stereo AER system was used (as shown in Figure 3), with two DVS-retinas [47], a Virtex-5 FPGA board and a USBAERmini2 board [48]. The retinas send spikes to the FPGA board, which processes the information received, and sends the outcome to the bus. The USBAERmini2 receives the output of the FPGA board and packages the information before sending it via USB. After that, a PC (Personal Computer) receives the spikes and stores the data. The VHDL-based system block diagram inside the FPGA can be seen in Figure 4 and the full system schematic is shown in the same figure. During the calibration step, the FPGA configuration is focused on packaging, coding and sending the information to the PC. After finishing the offline calibration step, the projection matrices are integrated into the FPGA processing step, thus the hardware system will be able to process the information and give us the results.

As explained before, to simplify the mechanism, the shown system will calculate the projection matrix using a linear approach. To that end, the linear method was implemented [10]. Figure 2 shows the grid of sixty-four LED lights, making a total of sixty-four 3D points, whose projections are obtained for each retina. A calibration distance of one meter was used, meaning that the calibration grid was placed one meter away from the stereo system (Figure 5).

During the last years, other works have focused on this area: [49] summarizes the evolution of neuromorphic stereo vision. Moreover, some works used other platforms to compute the retinas outputs, e.g., Firouzi *et al.* [50], who used a SpiNNaker [32] board to perform the stereo-matching process. This hardware platform consists of an array of chips with 18 ARM9 cores each, consuming around 70mW/core and ∼1.25W/chip. Andreopoulos *et al.* [51] used a TrueNorth [33] cluster to develop a stereo correspondence system, with each platform consuming 100mW top; its hardware platform is composed of 9 TrueNorth chips, thus the entire cluster consumes ∼900mW. Kogler *et al.* [52] tested a stereo calibration and matching process using a computer with stored AE data. This system is not implemented in a specific hardware platform and does not run in a real-time scenario. The power consumption depends on the computer used to process the data (not less than 65W).

These referenced works aimed at solving the same problems presented in this manuscript; however, all of them have a higher power consumption than the system designed in our work, i.e., at least 900mW, whereas our FPGA implementation requires less than 100mW (a detailed power consumption study was carried out in [41] for a more complex algorithm implemented in the same FPGA).

III. CALIBRATION MECHANISM

In this section we start describing the projection matrices calculation. After that, the 3D reconstruction is explained.

A. PROJECTION MATRICES

Using the LED grid shown above, the information related to the sixty-four 3D points is stored for each retina; with a

$$A \cdot W = b$$

$$A = \begin{pmatrix} x_1 & y_1 & z_1 & 1 & 0 & 0 & 0 & 0 & -x_1u_1 & -y_1u_1 & -z_1u_1 & -u_1 \\ 0 & 0 & 0 & 0 & x_1 & y_1 & z_1 & 1 & -x_1v_1 & -y_1v_1 & -z_1v_1 & -v_1 \\ x_2 & y_2 & z_2 & 1 & 0 & 0 & 0 & 0 & -x_2u_2 & -y_2u_2 & -z_2u_2 & -u_2 \\ 0 & 0 & 0 & 0 & x_2 & y_2 & z_2 & 1 & -x_2v_2 & -y_2v_2 & -z_2v_2 & -v_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{64} & y_{64} & z_{64} & 1 & 0 & 0 & 0 & 0 & -x_{64}u_{64} & -y_{64}u_{64} & -z_{64}u_{64} & -u_{64} \\ 0 & 0 & 0 & 0 & x_{64} & y_{64} & z_{64} & 1 & -x_{64}v_{64} & -y_{64}v_{64} & -z_{64}v_{64} & -v_{64} \end{pmatrix}$$

$$b = (0 \quad 0 \quad \cdots \quad 0)^t, \quad \textit{with } 64x2 \textit{ zeros}$$

$$W = (q_{11} \quad q_{12} \quad q_{13} \quad q_{14} \quad q_{21} \quad q_{22} \quad q_{23} \quad q_{24} \quad q_{31} \quad q_{32} \quad q_{33} \quad q_{34})^t \tag{1}$$

$$F \cdot M = 0; \qquad M = \begin{pmatrix} X_i & Y_i & Z_i & \lambda \end{pmatrix}^t$$

$$F = \begin{pmatrix} P_{L1,1} - u_i \cdot P_{L3,1} & P_{L1,2} - u_i \cdot P_{L3,2} & P_{L1,3} - u_i \cdot P_{L3,3} & P_{L1,4} - u_i \cdot P_{L3,4} \\ P_{L2,1} - v_i \cdot P_{L3,1} & P_{L2,2} - v_i \cdot P_{L3,2} & P_{L2,3} - v_i \cdot P_{L3,3} & P_{L2,4} - v_i \cdot P_{L3,4} \\ P_{R1,1} - u'_i \cdot P_{R3,1} & P_{R1,2} - u'_i \cdot P_{R3,2} & P_{R1,3} - u'_i \cdot P_{R3,3} & P_{R1,4} - u'_i \cdot P_{R3,4} \\ P_{R2,1} - v'_i \cdot P_{R3,1} & P_{R2,2} - v'_i \cdot P_{R3,2} & P_{R2,3} - v'_i \cdot P_{R3,3} & P_{R2,4} - v'_i \cdot P_{R3,4} \end{pmatrix} \tag{3}$$

commercial 3D scanner, the LED's 3D position is also stored. Using classical machine vision treatments, histograms of the spikes' streams are obtained, then binarized and, finally, gravity centers are represented. Finally, the 2D points projected on each retina are calculated. Using the 3D points of the LED grid and the 2D points of the retina, the projection matrix of each retina can be obtained using a linear approach (see equation 1). Only eight non-coplanar points are needed to calculate the projection matrix terms, although the more points used, the greater the projection matrix accuracy. With the sixty-four points obtained, the system obtained is shown in equation 1, as shown at the top of the this page.

In equation 4 each element q_{ij} represents the *(i,j)* term in the projection matrix; $(x_i, y_i, z_i, 1)$ represents the space coordinates of point i from the LED grid using homogenous coordinates; and $(u_i, v_i, 1)$ is the projection in one of the retinas of point i, using a scale factor of 1. This equation system was solved using Singular Value Decomposition (SVD) for the 2D points obtained from each retina, thus it was solved twiceto calculate the projection matrix of each retina.

Using the second step in Faugeras' method (optimization) [10], intrinsic and extrinsic parameters are extracted from the system and, with the obtained information, both projection matrices are optimized. The resulting optimized projection matrices are shown in equation 2.

$$P_{RIGHT} = \begin{pmatrix} 0.0028 & 0.0001 & -0.0018 & 0.5427 \\ -0.0002 & 0.0029 & -0.0012 & 0.8398 \\ -0.0000 & 0.0000 & -0.0000 & 0.0150 \end{pmatrix}$$

$$P_{LEFT} = \begin{pmatrix} -0.0030 & -0.0001 & 0.0009 & -0.6303 \\ 0.0000 & -0.0030 & 0.0015 & -0.7762 \\ -0.0000 & -0.0000 & 0.0000 & -0.0146 \end{pmatrix} \tag{2}$$

With these matrices, the projections of a given 3D point in the space can be obtained; the 3D to 2D reconstruction process is done as well. Next, the inverse process is described.

B. 3D RECONSTRUCTION

The most complex step is the 3D reconstruction given the 2D points of both cameras. First, these two sets of 2D points must be matched with each other, although this problem has been tackled in previous works and can be assumed to be solved [40]. Thus, the important question here is: "how can the 3D points be estimated given their projections in both retinas?" There are various techniques that address this problem [52], [53]; however, almost all of them need very high computational power. We need to simplify it in order to obtain a real-time solution. That is why the method used here for 3D reconstruction is based on a triangulation process: given the projection points in both retinas and the projection matrices calculated above, using projective geometry, an inverse projection matrix can be obtained. That is, given two 2D points (one from each retina), the 3D point in space can be estimated by triangulation. The triangulation process consists in finding the intersection of two known rays in space (one from each 2D projection). Starting with the pin-hole camera model system and using both projection matrices, the inverse projection matrix (also known as the *fundamental matrix*, F) is obtained in equation 3, as shown at the top of the this page.

In these equations, M is the 3D point to be calculated; (u_i, v_i) are the coordinates of the 2D point projected on the left retina; (u'_i, v'_i) are the coordinates of the 2D point projected on the right retina; P_L is the projection matrix of the left retina and P_R is the projection matrix of the right retina.

This system (equation 3) can be solved in several ways, and the real-time process depends on the mechanism used for the triangulation process. In order to evaluate the efficiency of the triangulation, the mechanism used in the computer is the SVD decomposition; however, a real-time solution was

then implemented in order to test the efficiency of the system running directly on hardware.

IV. HARDWARE IMPLEMENTATION

The projection matrices calculation can be done offline, using a big set of 3D points and their correspondences with each retina (as seen above in equation 1); thus, the time for calculating these matrices is not critical and can be done with high accuracy. However, during the activity of the system, pairs of 2D points arrive to the 3D positioning step, and it is important to use a fast mechanism to obtain the 3D coordinate. A compromise between accuracy and real time execution is usually needed, thus it is important to distinguish which factor is more important for a given system. In our case, accuracy is relevant, although it is more important to obtain a quick response from the system, since a large number of 2D pairs are arriving at the 3D step and the ball position changes very quickly. Moreover, the AER system philosophy determines that the continuous and repetitive flow of data allows errors to be corrected over time. With these premises, an SVD transformation to obtain the 3D coordinates would be, computationally, too demanding for an FPGA. The aim is to produce a simpler mechanism in order to obtain a solution using a few operations.

The fundamental matrix of the system was previously detailed (equation 3). This matrix unifies the information of the two projection matrices of the system (one from each visual sensor) in order to obtain the inverse transformation of the system (3D coordinates). A simplification mechanism is applied to the fundamental matrix to reduce the final equation system to be solved. This simplification comes from the fact that several terms are very close to zero in the projection matrices. These terms can be set to zero and, with this simple modification, some terms of the fundamental matrix are reduced. Using this premise, $P_{L3,1-3}$ and $P_{R3,1-3}$ are very close to zero (see equation 3), thus the final fundamental matrix is shown in equation 4.

$$F_{simplified} = \begin{pmatrix} P_{L1,1} & P_{L1,2} & P_{L1,3} & P_{L1,4} - u_i \cdot P_{L3,4} \\ P_{L2,1} & P_{L2,2} & P_{L2,3} & P_{L2,4} - v_i \cdot P_{L3,4} \\ P_{R1,1} & P_{R1,2} & P_{R1,3} & P_{R1,4} - u'_i \cdot P_{R3,4} \\ P_{R2,1} & P_{R2,2} & P_{R2,3} & P_{R2,4} - v'_i \cdot P_{R3,4} \end{pmatrix} \quad (4)$$

The final simplified equation system is:

$$\begin{aligned} P_{L1,1}X + P_{L1,2}Y + P_{L1,3}Z + \left(P_{L1,4} - u_i P_{L3,4}\right) &= 0 \\ P_{L2,1}X + P_{L2,2}Y + P_{L2,3}Z + \left(P_{L2,4} - v_i P_{L3,4}\right) &= 0 \\ P_{R1,1}X + P_{R1,2}Y + P_{R1,3}Z + \left(P_{R1,4} - u'_i P_{R3,4}\right) &= 0 \\ P_{R2,1}X + P_{R2,2}Y + P_{R2,3}Z + \left(P_{R2,4} - v'_i P_{R3,4}\right) &= 0 \end{aligned} \quad (5)$$

where $P_{Lx,y}$ are constant values, (u_i, v_i) and (u'_i, v'_i) are the projection points and *(X, Y, Z)* are the coordinates of the target point. If we continue with the simplification process using, for example, the equation obtained after solving for variable Y (6), as shown at the bottom of the this page.

Renaming some terms to simplify the visualization of the results, the equations obtained are:

$$(\textit{using equations } 5.a \textit{ and } 5.c\ ) \rightarrow Y = \frac{aZ + b + cU + dU'}{e} \quad (7)$$

$$(\textit{using equations } 5.b \textit{ and } 5.d\ ) \rightarrow Y = \frac{fZ + g + hV + iV'}{j} \quad (8)$$

$$(\textit{equations } 6 \textit{ and } 7\ ) \rightarrow Z = \frac{ge - bj + heV + ieV' - cjU - djU'}{aj - fe} \quad (9)$$

$$(\textit{equation } 5.a\ ) \rightarrow X = \frac{P_{L1,2}Y - P_{L1,3}Z - (P_{L1,4} - UP_{L3,4})}{P_{L1,1}} \quad (10)$$

Finally, one of these unknown values (Z) only depends on the projection coordinates, then Y can be obtained after calculating Z value and, finally, X is calculated using Y and Z. In the results section the efficiency of this simplification is evaluated. Before that, the FPGA implementation is described.

Using the "spike-processing building blocks" presented, evaluated and used in previous works [41], [55], [56] (developed to operate with pulsed information), the hardware implementation of this bio-inspired 3D positioning system can be integrated over an FPGA. These building blocks can be summarized as:

- SG (Spike Generator): transforms a numerical signal into frequency, to work with pulsed information.
- H&F (Hold & Fire): operates with spikes as an adder/subtractor.
- I&G (Integrate & Generate): generates a pulsed output with a frequency (depends on a constant and its inputs).

Thanks to these spiking building blocks, the equations calculated to obtain X, Y and Z coordinates can be implemented in the FPGA (equations 6 to 10). For example, the implementation of the Z coordinate is designed using a scheme like a summing-amplifier [57]. With this scheme and the spike-processing building blocks, the operation of equation 11 is designed in VHDL (see Figure 6).

$$Z = \frac{ge - bj}{aj - fe} + \frac{he}{aj - fe}V + \frac{ie}{aj - fe}V' - \frac{cj}{aj - fe}U - \frac{dj}{aj - fe}U' \quad (11)$$

Renaming the constants as δ_1, δ_2, δ_3, δ_4 and δ_5, a simplification is obtained in Equation 12.

$$Z = \delta_1 + \delta_2 V + \delta_3 V' - \delta_4 U - \delta_5 U' \quad (12)$$

$$Y = \frac{Z\left(P_{L1,3}P_{R1,1} - P_{R1,3}P_{L1,1}\right) + P_{R1,1}P_{L1,4} - P_{L1,1}P_{R1,4} - P_{R1,1}P_{L3,4}u_i + P_{L1,1}P_{R3,4}u'_i}{P_{R1,2}P_{L1,1} - P_{L1,2}P_{R1,1}} \quad (6)$$

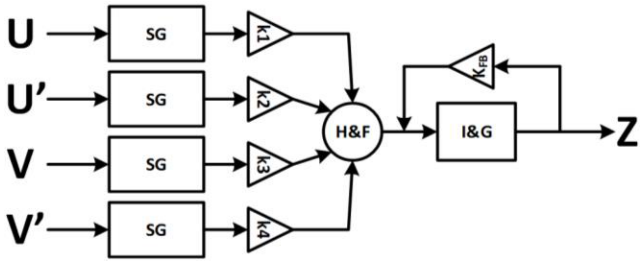

FIGURE 6. HDL block using spike-processing building blocks.

In a summing amplifier, the system is composed of two phases: attenuation and amplification. The final Z calculating block in VHDL (equation 15) is shown in Figure 6.

The values of the constants denoted as δ_n are determined by the relations between the global constant K_{FP} and its own constant k_n. Both have values between 0 and 1. The relation is shown in equation 13.

$$\delta_2 = \frac{k_3}{K_{FB}}, \quad \delta_3 = \frac{k_4}{K_{FB}}, \; \delta_4 = \frac{k_1}{K_{FB}}, \; \delta_5 = \frac{k_2}{K_{FB}} \tag{13}$$

The value of the K_{FB} constant is determined by the highest δ_n of the system and, with its value fixed, the rest of the constant values are calculated. Using the same principles and the circuit for Z coordinate calculation, X and Y are implemented and integrated in the FPGA.

At this point, the calibration step (using optimized projection matrices and the fundamental matrix) was described, simplified and integrated in a FPGA. Next, this system was tested using two different scenarios: original points reconstruction (using the same points as used in the calibration step) and moving target tests (using a pendulum and a rectangular object, both moving periodically at different distances from the stereo system). The first scenario was tested in the PC; then, the results are obtained from the FPGA implementation in order to compare the goodness of the hardware implementation. The second scenario was tested directly on the FPGA.

V. RESULTS AND DISCUSSION

In this section, the results are shown using the novel calibration mechanism modified to run in the FPGA using the building-blocks structure described above. Two different tests were evaluated: using the original calibration points and using moving objects.

A. ORIGINAL POINTS RECONSTRUCTION

Starting with the PC testing, Figure 7 shows the results obtained after the 2D reconstruction process. Only the reconstructed points and the error obtained are shown. The average error of each point was 0.3778 pixels/point.

For the 3D reconstruction process (Figure 8) the average error was 3.9958 mm/point.

Finally, this mechanism was integrated in the FPGA using the building-blocks structure described above. Then, some information of the projection matrices was simplified (some

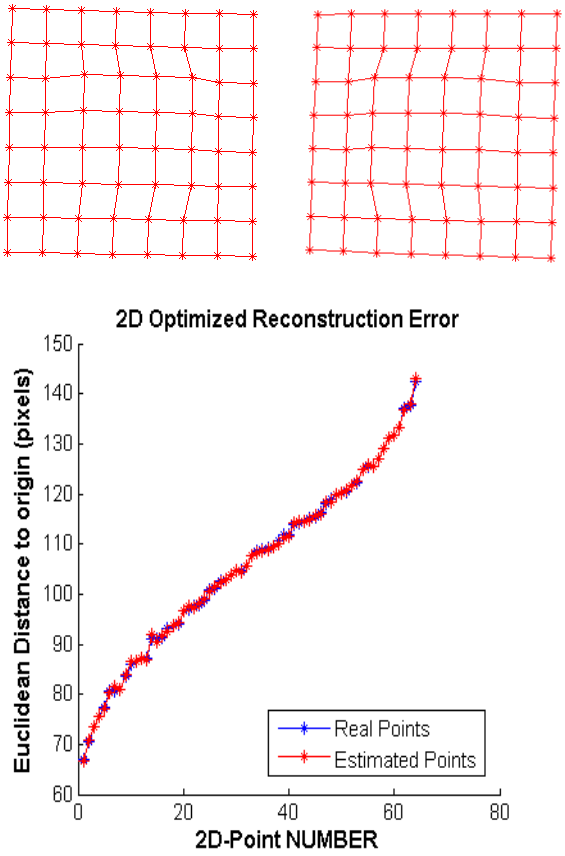

FIGURE 7. Up: Reconstructed 2D points (Left and Right Retinas). Down: Error obtained between original and reconstructed points.

error was introduced to the system). In order to quantify the error obtained during the FPGA integration, the same results were evaluated with the information directly from the FPGA. 3D reconstruction using the simplified fundamental matrix can be observed in Figure 9.

The quantitative results obtained in both cases and their comparisons can be observed in Table 1.

The error increased in the FPGA implementation from the original method, due to the fixed-point operations performed and the simplification for the hardware implementation, compared to the floating-point operations on CPU. However, a 3D positioning error of less than 1.5 cm at one meter is an acceptable result given that the vision sensors have a low resolution (128x128) and considerable random noise. Moreover, the number of operations needed to obtain the 3D coordinates was considerably reduced with the simplification of the fundamental matrix. In order to justify this conclusion, a theoretical latency test was conducted on each platform (PC and FPGA), obtaining the results observed in the last column of Table 1. As can be seen there, the PC implementation takes more than 100ms latency to obtain one 3D point (depending on the operating system computational load, it may take between 100 and 150 ms); in addition, using HDL blocks executed sequentially for X, Y and Z calculation (see Figure 6) in the FPGA (using HDL building blocks execution

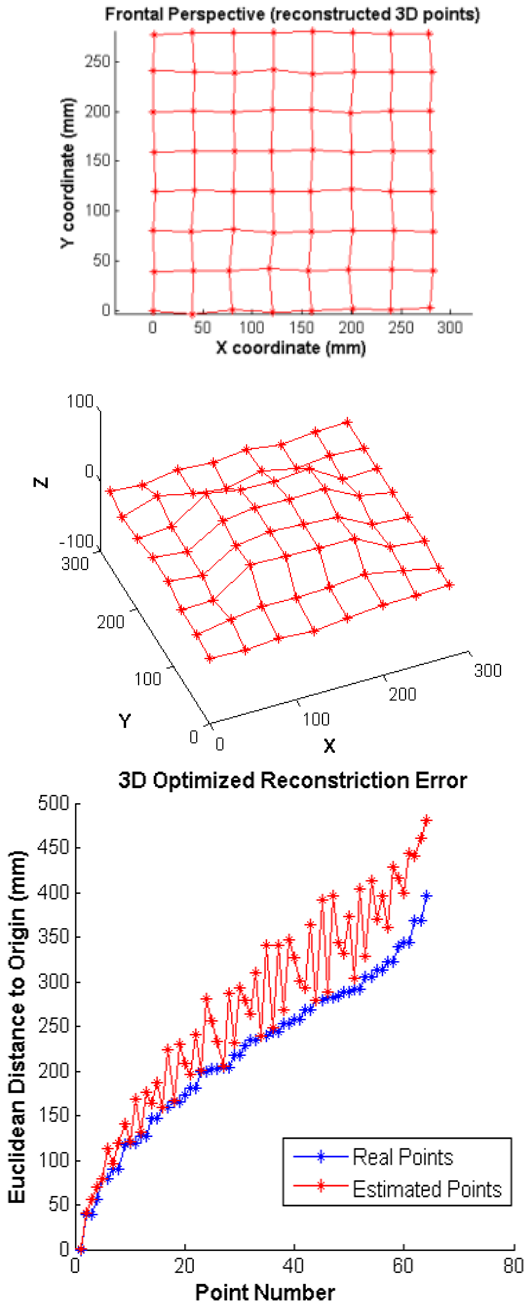

FIGURE 8. Up and Middle: Reconstructed 3D points, frontal and spatial views, respectively. Down: Error obtained between the original and the reconstructed points (optimized version).

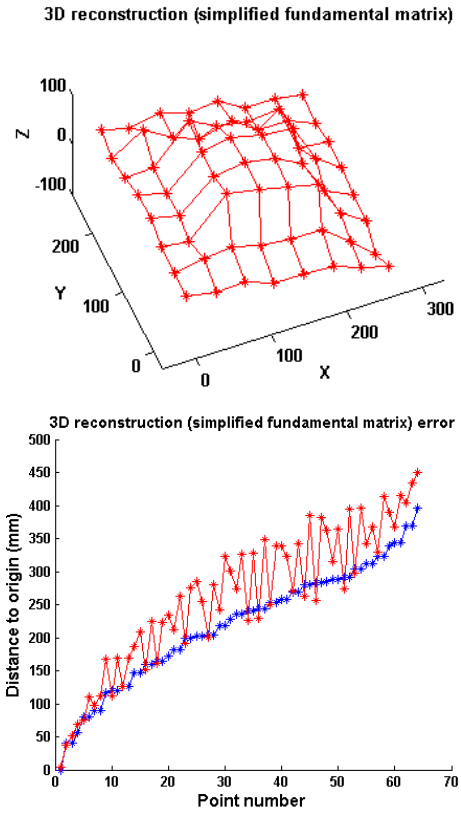

FIGURE 9. 3D reconstruction with the simplified fundamental matrix (integrated in the FPGA).

TABLE 1. Spiking-building-blocks hardware implementation comparison.

	Med. Error (mm)	Max. Error (mm)	Min. Error (mm)	σ (mm)	Frequency (Hz)
Original (PC)	*4.52*	*11.55*	*0.63*	*2.35*	*<10*
Simplified (FPGA)	*14.61*	*31.57*	*4.79*	*8.93*	*>100*

times from [55]), it takes less than 10ms latency to obtain one 3D point (around x10 improvement with 100Mhz clock on FPGA with respect to a 2.2Ghz clock on CPU).

B. MOVING TARGETS TEST

As explained at the beginning, the main difficulty of working with these AER systems in calibration is that they require the data to remain still in order to obtain the best possible accuracy, while this kind of system does not "see" anything but luminosity changes. Therefore, either flashing lights on the object (used on the calibration grid) or objects in movement are needed. As a first approach, using LED lights around a concrete object could solve this problem; however, this solution would be useless, since our aim is to use the system in an environment with moving targets. For this reason, to test the efficiency of our system under adverse situations, some tests with moving objects were done. It is important to indicate that any system is very efficient in its calculation when working around the area used for calibrating it. This means that since our system was calibrated for one-meter distance, the position of a moving object around one meter will be estimated with a great accuracy; however, if the object gets closer or further, the system's accuracy is expected to decrease.

First, we evaluated the depth estimation of the calibration system. In this case a pendulum was used. This pendulum crossed from left to right describing its trajectory over the *XY* plane. Using the tracking cells implemented by Linares-Barranco *et al.* [38], [58], the moving object was extracted from the scene and, using a gravity center calculation, the point used for the FPGA positioning system was extracted.

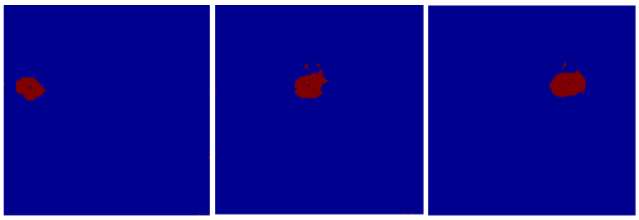

FIGURE 10. From left to right, Histograms 1, 2 and 3 obtained by the integration in time from the pendulum data of the right retina.

TABLE 2. Pendulum – distance estimation.

Real distance (cm)	Medium error (mm/point)	Max. error (mm)	Min. error (mm)	σ (mm)
25	*39.0591*	*87.0009*	*31.5253*	*11.7583*
50	*17.5305*	*45.0037*	*12.1054*	*10.2649*
75	*9.1820*	*23.0341*	*5.7063*	*5.8452*
100	***6.1574***	***17.3029***	***2.9320***	***3.9331***
125	*8.2475*	*25.9437*	*5.0367*	*4.0685*
150	*19.8530*	*49.9558*	*15.0154*	*9.8352*

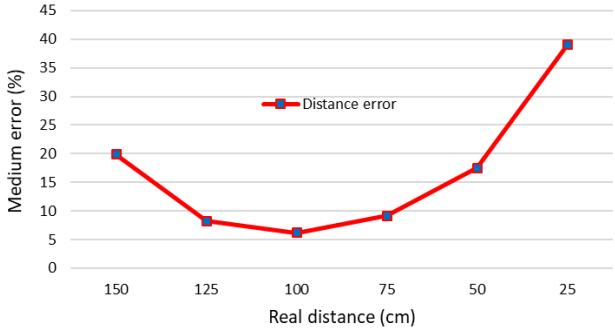

FIGURE 11. Pendulum error at several distances.

For example, the information received from one of the retinas can be seen in Figure 10.

This test was performed for several distances from the stereo vision system and the results can be observed in Table 2. A graphical representation of these results can be observed in Figure 11.

As can be observed, around the central calibration point (100cm) the error is minimal (around 6%), and it increases as the object moves away from that point (see Table 2).

The last test used a rectangular object moving in parallel to the retina plane (XY plane). This object had a real size of 100x200 mm (2dm^2) and the capture mechanism was similar to the one used with the pendulum. The test was repeated at different distances. The four corners of the tracked object were extracted; then, with these four points, a 3D reconstruction was and, using the results in spatial coordinates, the object size could be calculated offline. The graphical representation for the 100cm distance case is shown in Figure 12.

This process was repeated ten times for each distance. Both distance and object area estimation were obtained and are presented in Table 3 as the mean of these values.

Both distance and area estimation had reasonable resulting errors when close to the calibration origin (100cm), although

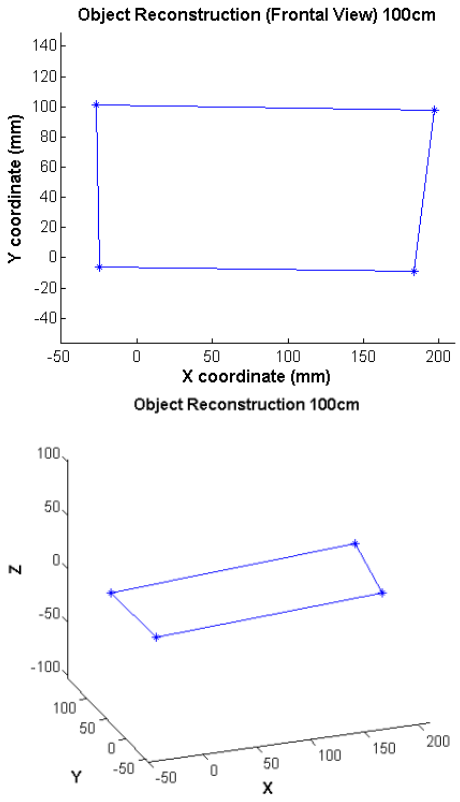

FIGURE 12. Left: Frontal view of the rectangle reconstruction at 100cm. Right: Spatial view of the rectangle reconstruction at 100cm.

TABLE 3. Rectangular object – distance and area estimation.

Real distance (cm)	Mean estimated distance (cm)	Distance error (%)	Estimated area (dm^2)	Area error (%)
25	*38.59*	*54.36*	*3.5144*	*75.72*
50	*61.91*	*23.82*	*2.9013*	*45.06*
75	*81.77*	*9.03*	*2.5609*	*28.04*
100	***100.92***	***0.92***	***2.1031***	***5.15***
125	*115.48*	*7.61*	*1.8230*	*8.85*
150	*130.03*	*13.31*	*1.5244*	*23.78*

this error increased as the object moved away from the calibration point (see Table 3). However, the distance estimation parameters presented relatively better results.

Two important points must be considered in order to understand the error values obtained in Table 2 and Table 3 at some distances: (1) Our system prioritizes real-time responses and power consumption over accuracy. That is why the process was simplified for its implementation in the FPGA. This simplification is based on, among other things, using fixed-point values and truncated low values. (2) the calibration process was conducted at a 1-meter distance, thus the best results were obtained near this point. The farther the object moves away from the calibration point, the greater the error is.

These two facts combined produce high-error results at distances far from the calibration point.

The graphical representation of Table 3 results can be observed in Figure 13.

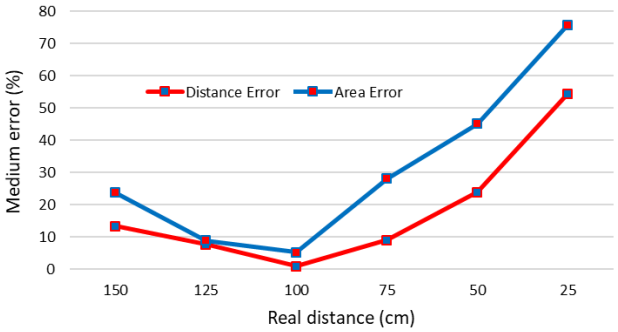

FIGURE 13. Rectangular object error at several distances.

VI. CONCLUSION

In this work, using the principles of classical machine vision calibration mechanisms, a spike-based calibration mechanism for a bio-inspired stereo DVS-AER retina system was designed and implemented. This spike-based mechanism reduces the number and complexity of the operations in a similar way as biological systems do, allowing its integration into autonomous robots (provided with FPGA) without the need for computers with high power consumption and computational resources.

To test the efficiency of this method, a novel led calibration grid with 64 LED lights controlled by a PIC microcontroller was built. 2D and 3D reconstructions were implemented on a FPGA using VHDL language, and evaluated with satisfactory results, using static and non-static targets, compared to the same implementations using floating-point computing units on processors.

To the best of our knowledge, this is the first implementation in FPGA of a 3D positioning stage for a bio-inspired stereo-vision system based on an AER-DVS retina. Thanks to the results obtained in the present work, the 3D positioning step of the robotic table-tennis player was implemented and it is currently under testing as part of the whole control system.

REFERENCES

[1] L. D. Harmon and K. C. Knowlton, "Picture processing by computer," *Science*, vol. 80, no. 3875, pp. 19–29, 1969.

[2] C. Dyer, "Volumetric scene reconstruction from multiple views," in *Foundations of Image Understanding*. 2001.

[3] I. Detchev, A. Habib, M. Mazaheri, and D. Lichti, "Practical *in situ* implementation of a multicamera multisystem calibration," *J. Sensors*, vol. 2018, Feb. 2018, Art. no. 5351863.

[4] F. A. Moreno, J. L. Blanco, and J. Gonzalez, "Stereo vision specific models for particle filter-based SLAM," *Robot. Auton. Syst.*, vol. 57, no. 9, pp. 955–970, 2009.

[5] J. Kalomiros and J. Lygouras, "Robotic mapping and localization with real-time dense stereo on reconfigurable hardware," *Int. J. Reconfigurable Comput.*, vol. 2010, Jan. 2010, Art. no. 9.

[6] J. Weng, P. Coher, and M. Herniou, "Camera calibration with distortion models and accuracy evaluation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 10, pp. 965–980, Oct. 1992.

[7] S. Barone, A. Paoli, and A. V. Razionale, "3D reconstruction and restoration monitoring of sculptural artworks by a multi-sensor framework," *Sensors*, vol. 12, no. 12, pp. 16785–16801, 2012.

[8] M. Marrón-Romera, J. C. García, M. A. Sotelo, D. Pizarro, M. Mazo, J. M. Cañas, C. Losada, and Á. Marcos, "Stereo vision tracking of multiple objects in complex indoor environments," *Sensors*, vol. 10, no. 10, pp. 8865–8887, 2010.

[9] J. L. Mundy, "Relationship between photogrammmetry and computer vision," *Proc. SPIE*, vol. 1944, pp. 92–105, Sep. 1993.

[10] O. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint* (Artificial Intelligence). 1993.

[11] P. Rathnayaka, S. H. Baek, and S. Y. Park, "An efficient calibration method for a stereo camera system with heterogeneous lenses using an embedded checkerboard pattern," *J. Sensors*, vol. 2017, Sep. 2017, Art. no. 6742615.

[12] R. Szeliski, *Computer Vision: Algorithms and Applications*. Long Beach, CA, USA, 2010.

[13] H. A. Martins, J. R. Birk, and R. B. Kelley, "Camera models based on data from two calibration planes," *Comput. Graph. Image Process.*, vol. 17, no. 2, pp. 173–180, 1981.

[14] R. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE J. Robot. Autom.*, vol. 3, no. 4, pp. 323–344, Aug. 1987.

[15] J. Heikkilä, "Geometric camera calibration using circular control points," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 10, pp. 1066–1077, Oct. 2000.

[16] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000.

[17] M. T. Ahmed, E. E. Hemayed, and A. A. Farag, "Neurocalibration: A neural network that can tell camera calibration parameters," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, Sep. 1999, pp. 463–468.

[18] J. Batista, H. Araujo, and A. T. de Almeida, "Iterative multistep explicit camera calibration," *IEEE Trans. Robot. Autom.*, vol. 15, no. 5, pp. 897–917, Oct. 1999.

[19] C. Posch, T. Serrano-Gotarredona, B. Linares-Barranco, and T. Delbruck, "Retinomorphic event-based vision sensors: Bioinspired cameras with spiking output," *Proc. IEEE*, vol. 102, no. 10, pp. 1470–1484, Oct. 2014.

[20] M. A. Sivilotti, "Wiring considerations in analog VLSI systems with application to field-programmable networks," Ph.D. dissertation, California Inst. Technol., Pasadena, CA, USA, 1991.

[21] A. Linares-Barranco, H. Liu, A. Rios-Navarro, F. Gómez-Rodríguez, D. P. Moeys, and T. Delbruck, "Approaching retinal ganglion cell modeling and FPGA implementation for robotics," *Entropy*, vol. 20, no. 6, p. 475, 2018.

[22] A. Jimenez-Fernandez, G. Jimenez-Moreno, A. Linares-Barranco, M. J. Domínguez-Morales, R. Paz-Vicente, and A. Civit-Balcells, "A neuro-inspired spike-based PID motor controller for multi-motor robots with low cost FPGAs," *Sensors*, vol. 12, no. 4, pp. 3831–3856, 2012.

[23] R. Serrano-Gotarredona, M. Oster, P. Lichtsteiner, A. Linares-Barranco, R. Paz-Vicente, F. Gómez-Rodríguez, L. Camuñas-Mesa, R. Berner, M. Rivas-Pérez, T. Delbruck, S.-C. Liu, R. Douglas, P. Hafliger, G. Jimenez-Moreno, A. C. Ballcels, T. Serrano-Gotarredona, A. Acosta-Jimenez, and B. Linares-Barranco, "CAVIAR: A 45 k neuron, 5 M synapse, 12 G connects/s AER hardware sensory–processing–learning–actuating system for high-speed visual object recognition and tracking," *IEEE Trans. Neural Netw.*, vol. 20, no. 9, pp. 1417–1438, Sep. 2009.

[24] A. Jiménez-Fernández, "A binaural neuromorphic auditory sensor for FPGA: A spike signal processing approach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 4, pp. 804–818, Apr. 2017.

[25] J. P. Domínguez-Morales, A. Jimenez-Fernandez, M. J. Domínguez-Morales, and G. Jimenez-Moreno, "NAVIS: Neuromorphic auditory visualizer tool," *Neurocomputing*, vol. 237, pp. 418–422, May 2017.

[26] C. Liu, G. Bellec, B. Vogginger, D. Kappel, J. Partzsch, F. Neumärker, S. Höppner, W. Maass, S. B. Furber, R. Legenstein, and C. G. Mayr, "Memory-efficient deep learning on a SpiNNaker 2 prototype," *Frontiers Neurosci.*, vol. 12, p. 840, Nov. 2018.

[27] A. Yousefzadeh, M. Jablonski, T. Iakymchuk, A. Linares-Barranco, A. Rosado, L. A. Plana, S. Temple, T. Serrano-Gotarredona, S. B. Furber, and B. Linares-Barranco, "On multiple AER handshaking channels over high-speed bit-serial bidirectional LVDS links with flow-control and clock-correction on commercial FPGAs for scalable neuromorphic systems," *IEEE Trans. Biomed. Circuits Syst.*, vol. 11, no. 5, pp. 1133–1147, Aug. 2017.

[28] C. Zamarreno-Ramos, A. Linares-Barranco, T. Serrano-Gotarredona, and B. Linares-Barranco, "Multicasting mesh AER: A scalable assembly approach for reconfigurable neuromorphic structured AER systems. Application to ConvNets," *IEEE Trans. Biomed. Circuits Syst.*, vol. 7, no. 1, pp. 82–102, Feb. 2013.

[29] M. J. Domínguez-Morales, A. Jimenez-Fernandez, R. Paz, M. R. López-Torres, E. Cerezuela-Escudero, A. Linares-Barranco, G. Jimenez-Moreno, and A. Morgado, "An approach to distance estimation with stereo vision using address-event-representation," in *Neural Information Processing* (Lecture Notes in Computer Science), vol. 7062. 2011.

[30] M. Domínguez-Morales, A. Jimenez-Fernandez, R. Paz-Vicente, G. Jimenez, and A. Linares-Barranco, "Live demonstration: On the distance estimation of moving targets with a stereo-vision AER system," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2012, pp. 721–725.

[31] M. Domínguez-Morales, E. Cerezuela-Escudero, A. Jimenez-Fernandez, R. Paz-Vicente, J. L. Font-Calvo, P. Inigo-Blasco, A. Linares-Barranco, and G. Jimenez-Moreno, "Image matching algorithms in stereo vision using address-event-representation: A theoretical study and evaluation of the different algorithms," in *Proc. Int. Conf. Signal Process. Multimedia Appl. (SIGMAP)*, Jul. 2011, pp. 79–84.

[32] S. B. Furber, F. Galluppi, S. Temple, and L. A. Plana, "The SpiNNaker project," *Proc. IEEE*, vol. 102, no. 5, pp. 652–665, Feb. 2014.

[33] F. Akopyan, "TrueNorth: Design and tool flow of a 65 mW 1 million neuron programmable neurosynaptic chip," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 10, pp. 1537–1557, Oct. 2015.

[34] M. Davies *et al.*, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–89, Jan. 2018.

[35] R. Tapiador-Morales, A. Linares-Barranco, A. Jimenez-Fernandez, and G. Jimenez-Moreno, "Neuromorphic LIF row-by-row multiconvolution processor for FPGA," *IEEE Trans. Biomed. Circuits Syst.*, vol. 13, no. 1, pp. 159–169, Feb. 2019.

[36] L. A. Camuñas-Mesa, Y. L. Domínguez-Cordero, A. Linares-Barranco, T. Serrano-Gotarredona, and B. Linares-Barranco, "A configurable event-driven convolutional node with rate saturation mechanism for modular ConvNet systems implementation," *Frontiers Neurosci.*, vol. 12, p. 63, Feb. 2018.

[37] F. Gómez-Rodríguez, L. Miró-Amarante, F. Diaz-del-Rio, A. Linares-Barranco, and G. Jimenez, "Real time multiple objects tracking based on a bio-inspired processing cascade architecture," in *Proc. IEEE Int. Symp. Circuits Syst., Nano-Bio Circuit Fabrics Syst. (ISCAS)*, May/Jun. 2010, pp. 1399–1402.

[38] A. Linares-Barranco, F. Gómez-Rodríguez, V. Villanueva, L. Longinotti, and T. Delbruck, "A USB3.0 FPGA event-based filtering and tracking framework for dynamic vision sensors," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2015, pp. 2417–2420.

[39] M. Domínguez-Morales, A. Jiménez-Fernndez, R. Paz-Vicente, A. Linares-Barranco, and G. Jiménez-Moreno, "Stereo matching: From the basis to neuromorphic engineering," in *Current Advancements in Stereo Vision*. Rijeka, Croatia: InTech, 2012.

[40] M. J. Domínguez-Morales, E. Cerezuela-Escudero, F. Perez-Peña, A. Jimenez-Fernandez, A. Linares-Barranco, and G. Jimenez-Moreno, "On the AER stereo-vision processing: A spike approach to epipolar matching," in *Neural Information Processing* (Lecture Notes in Computer Science), vol. 8226. 2013, pp. 267–275.

[41] M. Domínguez-Morales, J. P. Domínguez-Morales, Á. Jiménez-Fernández, A. Linares-Barranco, and G. Jiménez-Moreno, "Stereo matching in address-event-representation (AER) bio-inspired binocular systems in a field-programmable gate array (FPGA)," *Electronics*, vol. 8, no. 4, p. 410, 2019.

[42] L. A. Camuñas-Mesa, T. Serrano-Gotarredona, S. H. Ieng, R. Benosman, and B. Linares-Barranco, "Event-driven stereo visual tracking algorithm to solve object occlusion," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 9, pp. 4223–4237, Sep. 2018.

[43] J. Carneiro, S. H. Ieng, C. Posch, and R. Benosman, "Event-based 3D reconstruction from neuromorphic retinas," *Neural Netw.*, vol. 45, pp. 27–38, Sep. 2013.

[44] J. Black, T. Ellis, and P. Rosin, "Multi view image surveillance and tracking," in *Proc. Workshop Motion Video Comput.*, Dec. 2002, pp. 169–174.

[45] P. Rogister, R. Benosman, S.-H. Ieng, P. Lichtsteiner, and T. Delbrück, "Asynchronous event-based binocular stereo matching," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 2, pp. 347–353, Feb. 2012.

[46] F. Perez-Peña, A. Morgado-Estevez, A. Linares-Barranco, A. Jimenez-Fernandez, F. Gómez-Rodríguez, G. Jimenez-Moreno, and J. Lopez-Coronado, "Neuro-inspired spike-based motion: From dynamic vision sensor to robot motor open-loop control through spike-VITE," *Sensors*, vol. 13, no. 11, pp. 15805–15832, 2013.

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

[48] R. Berner, T. Delbruck, A. Civit-Balcells, and A. Linares-Barranco, "A 5 Meps $100 USB2.0 address-event monitor-sequencer interface," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2007, pp. 2451–2454.

[49] L. Steffen, D. Reichard, J. Weinland, J. Kaiser, A. Roennau, and R. Dillmann, "Neuromorphic stereo vision: A survey of bio-inspired sensors and algorithms," *Frontiers Neurorobot.*, vol. 13, p. 28, May 2019.

[50] G. Dikov, M. Firouzi, F. Röhrbein, J. Conradt, and C. Richter, "Spiking cooperative stereo-matching at 2 ms latency with neuromorphic hardware," in *Proc. Conf. Biomimetic Biohybrid Syst.*, 2017, pp. 119–137.

[51] A. Andreopoulos, H. J. Kashyap, T. K. Nayak, A. Amir, and M. D. Flickner, "A low power, high throughput, fully event-based stereo system," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7532–7542.

[52] J. Kogler, C. Sulzbachner, M. Humenberger, and F. Eibensteiner, "Address-event based stereo vision with bio-inspired silicon retina imagers," in *Advances in Theory and Applications of Stereo Vision*. 2011, pp. 165–188.

[53] K. Zhu, M. Butenuth, and P. d'Angelo, "Computational optimized 3D reconstruction system for airborne image sequences," Tech. Rep., 2010.

[54] L. Zhaohua, Y. Yuxia, Z. Liu, and Y. Yang, "Study on semi-global matching algorithm extended for multi baseline matching and parallel processing method based on GPU," *Comput. Model. New Technol.*, vol. 18, no. 3, pp. 174–178, 2014.

[55] A. Jimenez-Fernandez, M. Domínguez-Morales, E. Cerezuela-Escudero, R. Paz-Vicente, A. Linares-Barranco, and G. Jimenez, "Simulating building blocks for spikes signals processing," in *Advances in Computational Intelligence* (Lecture Notes in Computer Science), vol. 6692. 2011, pp. 548–556.

[56] M. Domínguez-Morales, A. Jimenez-Fernandez, E. Cerezuela-Escudero, R. Paz-Vicente, A. Linares-Barranco, and G. Jimenez, "On the designing of spikes band-pass filters for FPGA," in *Artificial Neural Networks and Machine Learning* (Lecture Notes in Computer Science), vol. 6792. 2011, pp. 389–396.

[57] C. M. Kukla, "Summing amplifier," *Rev. Sci. Instrum.*, vol. 40, no. 4, pp. 598–599, 1969.

[58] F. Gómez-Rodríguez, L. Miró-Amarante, M. Rivas, G. Jimenez, and F. Diaz-del-Rio, "Neuromorphic real-time objects tracking using address event representation and silicon retina," in *Advances in Computational Intelligence* (Lecture Notes in Computer Science). 2011, pp. 133–140.

MANUEL J. DOMÍNGUEZ-MORALES received the B.S. degree in computer science engineering, the M.S. degree in software engineering, in 2009, the M.S. degree in computer engineering, in 2014 and the Ph.D. degree in neuromorphic engineering, in 2014, from the University of Seville (Spain). From 2009 to 2014, he was an Assistant Professor with the Computer Architecture and Technology Department, University of Seville, where he has been a Postdoctoral Researcher and a Lecturer, since 2014. His research fields are focused on computer vision, image processing, signal processing, embedded systems, programmable hardware design, computer architecture, robotics, and e-Health.

ÁNGEL JIMÉNEZ-FERNÁNDEZ received the B.S. degree in computer engineering, the M.S. degree in industrial computer science, and the Ph.D. degree in neuromorphic engineering from the University of Seville, Seville, Spain, in 2005, 2007, and 2010, respectively. Since October 2007, he has been an Assistant Professor of computer architecture and technology at the University of Seville, where he was promoted to Associate Professor, in April 2011. His research interests include neuromorphic engineering applied to robotics, real-time spikes signal processing, neuromorphic sensors, field programmable gate array (FPGA) digital design, and embedded systems development.

GABRIEL JIMÉNEZ-MORENO received the M.S. degree in physics (electronics) and the Ph.D. degree from the University of Seville, Seville, Spain, in 1987 and 1992, respectively. After working with Alcatel, he was granted a Fellowship from the Spanish Science and Technology Commission (CICYT). He is currently an Associate Professor of computer architecture with the University of Seville. He participated in the creation of the Department of Computer Architecture, University of Seville, where he has been the Director, since 2013. He is the author of various articles and research reports on robotics, rehabilitation technology, and computer architecture. His research interests include neural networks, vision processing systems, embedded systems, computer interfaces, and computer architectures.

CRISTINA CONDE received the B.S. degree in physics (electronics) from the University Complutense of Madrid, in 1999, and the Ph.D. degree from the University Rey Juan Carlos of Madrid, in 2006. She had several years of working in the private sector and then, she joined the University Rey Juan Carlos, as an Assistant Professor, in 2001. After seven years, she was the Vicedean of the Studies with the Computer Science School. She has also coordinated several national and European projects. Her research interests include image and video analysis, pattern recognition, and machine learning in both classical and biologically inspires computation.

ENRIQUE CABELLO received the B.S. degree in physics (electronics) from the University of Salamanca, and the Ph.D. degree from the Polytechnic University of Madrid. In 1990, he joined the University of Salamanca, where he was an Assistant Professor with the School of Sciences. Since 1998, he has been with Universidad Rey Juan Carlos, where he has been the Coordinator of the Face Recognition and Artificial Vision Group, since 2001. He is currently the Head of the Computer Science and Statistics Department. His research interests include image and video analysis, pattern recognition, and machine learning. In all cases, he uses classic and bio-inspired approaches.

ALEJANDRO LINARES-BARRANCO received the B.S. degree in computer engineering, the M.S. degree in industrial computer science, and the Ph.D. degree in computer science (specializing in computer interfaces for neuromorphic systems) from the University of Sevilla, Sevilla, Spain, in 1998, 2002, and 2003, respectively. In 2001, he joined the University of Seville as an Assistant Professor with the Architecture and Technology of Computers Department. In 2009, he was promoted to Associate Professor (civil-servant). Since 2013, he has been the Secretary of the Department. He belongs to the Robotics and Technology of Computers Lab. His research interests include VLSI for FPGA digital design, neuro-inspired event-based processing, and interfaces for FPGAs. He joined the IEEE Neural Systems Applications and Technologies (NSAT) Technical Committee of the Circuits and Systems Society (CASS), in 2010, where he served as an Officer, from 2010 to 2014. He joined the IEEE Sensory System TC, in 2012.