Histogram-less direct time-of-flight imaging based on a machine learning processor on FPGA

Tommaso Milanese[*], Jiuxuan Zhao[*], Brent Hearn[†], Edoardo Charbon[*]
[*]AQUA laboratory, École Polytechnique Fédérale de Lausanne, Neuchâtel
{tommaso.milanese, jiuxuan.zhao, edoardo.charbon}@epfl.ch
[†]Imaging division, STMicroelectronics, Edinburgh, U.K.
{brent.hearn}@st.com

***Abstract*—The investigation of a novel architecture for direct time-of-flight (TOF) SPAD based imaging systems is presented. In the proposed architecture, a pulsed laser source illuminates a scene and the reflected light is captured by a SPAD, which detects photons and converts them to a digital pulse. Like in time-correlated single-photon counting (TCSPC), for each detected photon a timestamp is generated, however, unlike TCSPC, it is fed to an machine-learning processor (MLP) that was trained to recognize SPAD responses in direct TOF. The MLP generates the distance to the target directly, taking into account potential non-idealities in timestamp generation and processing. Finally, the proposed architecture was demonstrated in practical scenes and its performance reported using standard LiDAR characterization methods.**

I. Introduction

SPADs are the sensor of choice in many direct TOF and LiDAR systems, thanks to their compactness and picosecond timing resolution, which enables millimetric precision. In TCSPC, timestamps are generated whenever a photon is detected and organized in a histogram. A histogram approaches the true response of the SPAD upon for a very large – ideally infinite – number of detected photons. In practice, an estimate of the TOF is extracted from the histogram after a finite time and thus its precision is also limited. In addition, due to dark noise and background illumination, a typical histogram contains large data, much of it not useful for computing TOF. Hence, the memory allocation for a histogram is generally overestimated and thus inefficient [1], [2]. Indeed, the memory scales exponentially with respect to full scale range (FSR) and hardware timing precision and linearly with the number of depth-dots, leading to a large, possibly on-chip memory [3], [4]. To address this issue, partial histograms have been devised [5]. This approach however, in its simplest embodiment, may prevent the detection of multiple targets at separate depths. To address this shortcoming, more complex partial histograms are needed, along with complex tracking algorithms. An alternative to direct TOF, is the use of indirect TOF and frequency modulated continuous wave (FMCW) techniques, however, these techniques perform averaging at various levels of sophistication, which in effect prevents multiple depth detection as well. In this paper, we propose to use machine learning to process all photon timestamps directly, as soon as they are generated by a time-to-digital converter (TDC) driven by the SPAD image sensor. The objective is the elimination of the histograms needed in a direct TOF configuration, as shown in Fig.1. This approach provides the same advantages

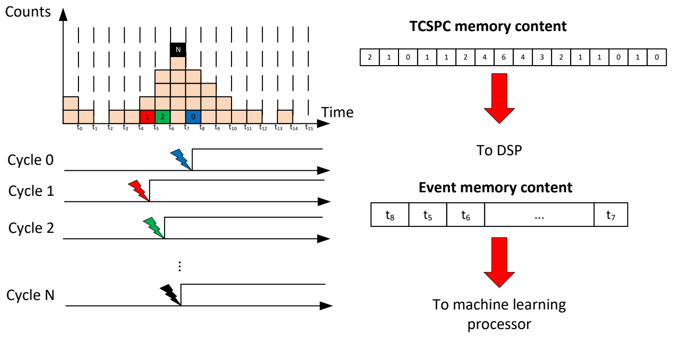

Counts
N
Time
t_0 t_1 t_2 t_3 t_4 t_5 t_6 t_7 t_8 t_9 t_{10} t_{11} t_{12} t_{13} t_{14} t_{15}
Cycle 0
Cycle 1
Cycle 2
Cycle N
TCSPC memory content
2 1 0 1 1 2 4 6 4 3 2 1 1 0 1 0
To DSP
Event memory content
t_8 t_5 t_6 ... t_7
To machine learning processor

Figure 1. Standard TCSPC and proposed event-processing.

of direct TOF with full histograms, but with much lower requirements in terms of memory and processing power. At the same time, non-idealities associated with the generation and processing of timestamps are intrinsically accounted for. Moreover, the machine-learning processor (MLP) can also be reconfigured to other tasks at a higher level, such as shape and object recognition. The novel machine learning processor was optimized for the long short-term memory (LSTM) execution. We show how to implement the LSTM efficiently in a commercial FPGA, so as to integrate it in the SPAD image sensor in the near future. In the current prototype, photon-detection is performed by a CMOS SPAD, whose raw signals are routed to the FPGA implementing an array of on-demand TDCs, which are then passed on to the LSTM accelerator, as shown in the remainder of the paper.

II. System Architecture

The general architecture is depicted in Fig.2. The SPAD signal is timestamped by the TDC and the timing data is saved into a 512 $\times$ 32b event memory. The host controls the system by means of a state machine, sending a start signal through the Opal Kelly C++ interface and waiting the end of the acquisition and data processing. The LSTM accelerator starts the processing when the event memory is filled, otherwise it stays in sleep mode, reducing the processing power consumption. When the depth retrieval ends the system sends a signal to the host and is ready to get a new depth-dot.

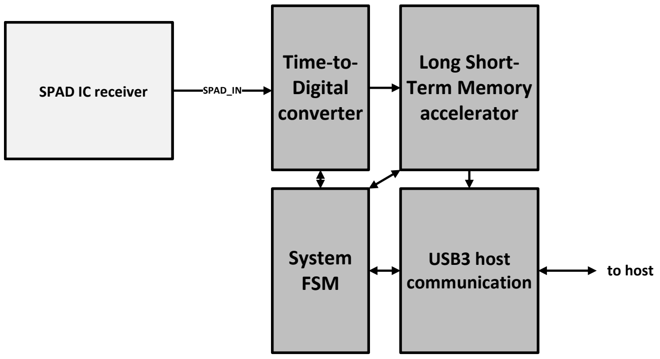


Figure 2. System architecture.

A. *Time-to-Digital converter*

The TDC is based on a tapped delay-line (TDL) latched at 400 MHz. A chain of Carry4 modules is instantiated in adjacent slices, following the place&route (PnR) tool provided by Xilinx for the ripple carry additions. The thermometer output of the TDL is sampled twice, first by a flip-flop (FF) in the same logic slice as the Carry4 and then by another FF placed by the PnR tool, so as to decrease the probability of metastability in the thermometer code.

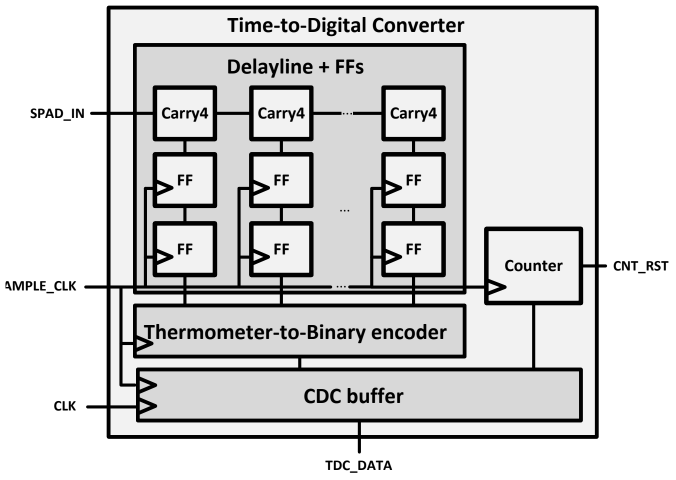


Figure 3. TDC block diagram. Two synchronized clock domains are used operating at 100 and 400 MHz.

A pipelined thermometer-to-binary encoder (T2B) converts the thermometer to binary code at 400 MHz and passes the data to a clock domain crossing synchronizer, reducing the data rate from the TDL sampling clock to the 100 MHz system clock. At the end of the signal flow the data is written in the event memory at system clock speed.

B. *LSTM accelerator*

The LSTM is a recurrent neural network (RNN), a particular type of artificial neural network (ANN) [6]. Since its conception, this processing layer has been used extensively for the processing of time series data, for instance ECG signal classification [7] and speech recognition [8]. The time series in consideration for SPAD-based D-TOF is the raw timestamp data stream, the same data that is commonly organized in a histogram for peak finding. The governing equations for this network are stated as:

$$f_t = \sigma(\mathbf{W_{xf}} x_t + \mathbf{W_{hf}} h_{t-1} + b_f) \quad (1)$$
$$i_t = \sigma(\mathbf{W_{xi}} x_t + \mathbf{W_{hi}} h_{t-1} + b_i) \quad (2)$$
$$\tilde{c}_t = \tanh(\mathbf{W_{xc}} x_t + \mathbf{W_{hc}} h_{t-1} + b_c) \quad (3)$$
$$o_t = \sigma(\mathbf{W_{xo}} x_t + \mathbf{W_{ho}} h_{t-1} + b_o) \quad (4)$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (5)$$
$$h_t = o_t \odot \tanh(c_t) \quad (6)$$

The bold quantities are matrices and the others are vectors. $\odot$ represents the element-wise multiplication between 2 vectors, $\sigma(\cdot)$ represents the element-wise sigmoid activation for all the vector elements and $\tanh(\cdot)$ is the element-wise hyperbolic tangent activation. After the LSTM layer a final fully connected layer (FCN) is used to transform the final hidden state vector into a regression value, that for this application is a number ranging from 0 to 1 representing the phase of the backscattered light pulse with respect to the laser emitter. This number is then multiplied by the FSR to extract the distance in post-processing. Eqs.1-6 embed matrix-vector multiplications, element-wise additions and multiplications and element-wise non-linear activations, all completely parallelizable operations. The design of this accelerator is based on a row-stationary data flow for the matrix-vector multiplication: each processing elements (PEs) compute one row of the multiplication, acting all in parallel. For the LSTM algorithm execution the resources needed are multipliers, adders, and non-linear activation LUTs, which form the basis of the PE design, Fig.4; muxes are added before the three operators to be able to perform element-wise vector operations. Referring to Eqs.1-6 f_t, i_t, $\tilde{c}_t$, o_t and c_t are stored in the activation registers, h_t in the hidden state memory, all $\mathbf{W}$ and b in the weight memory and x_t in the separate event-memory (not shown in the figure). By reprogramming the PEs scalar and vector calculations can be performed in parallel, lowering the processing time per timestamp. The memories in the design have been laid out in such a way that each PE has its own memory space, both in the weight memory and in the activation registers needed for the LSTM execution. A program counter dictates the state machine execution and, exploiting a masking operation, it controls the address of the weight memory. The weight memory for this design is a 42×128 bit block RAM Xilinx IP, but in the future it will be substituted by a single port SRAM for solid state implementations. The activation registers are implemented with logic, and they are subdivided in two sub-banks per PE in order to decrease the algorithm execution time. The total memory allocated for these registers is 80×28 bits, sub-divided in 10×28 bits for each of the eight PEs.

III. System training and quantization

The LSTM based network has been trained off-line after the creation of a dataset from a MATLAB simulator [9]. The

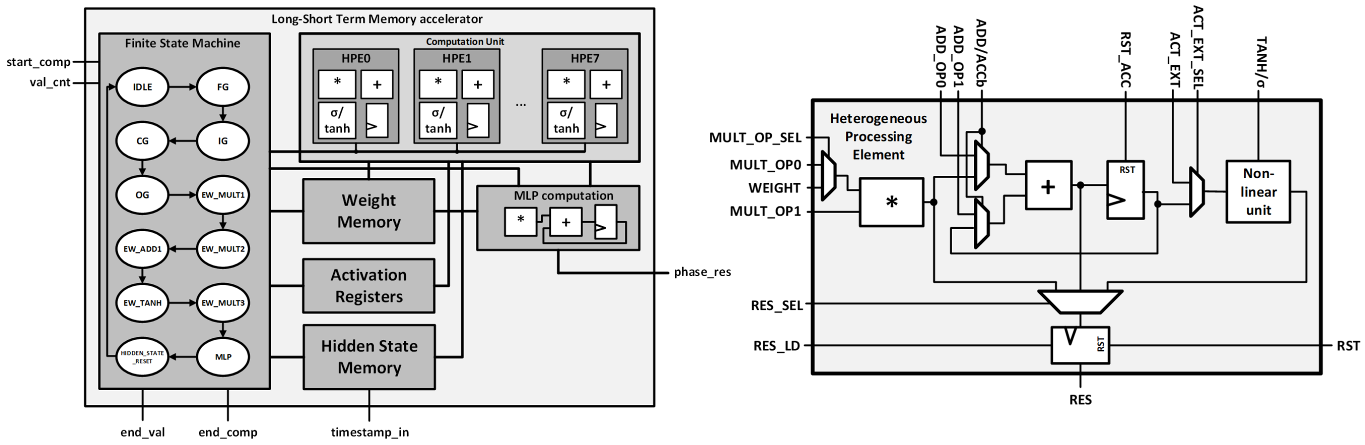


Figure 4. (Left) LSTM accelerator block diagram. (Right) Processing element RTL schematic.

LiDAR signal return is modeled as a Gaussian distribution :

$$P(t;d) = R(d)\exp\left\{\frac{(t-\frac{2d}{c})^2}{\sigma}\right\}, \tag{7}$$

where $R(d)$ represents a constant embedding the physical scene/SPAD parameters, d is the target distance, σ is the parameter related to system jitter and c is the speed of light. A background light of 1klux was chosen and a reference white target with 97% reflectivity has been used. The background is assumed to be a uniformly distributed noise source over the laser period with Poisson statistics, where the λ parameter corresponds to the background lux intensity. A gamut of distances ranging from 0 to 15 m with a step of 60 μm was selected and histograms of these distances were constructed and sampled, to obtain 10k time-series per each distance point, containing 512 timestamp values. Fig.5 shows an example of the training data, for the corresponding distance of 8.89 m. The network was implemented in PyTorch and then trained in Google Colab using mean squared error (MSE) loss function:

$$\mathrm{MSE}(y_{p,i}, y_t) = \frac{1}{N_s}\sum_{i=1}^{N_s}(y_{p,i}-y_t)^2, \tag{8}$$

with N_s being the number of timestamps processed by the network (512 in our case), $y_{p,i}$ the predicted value of the network for every time step and y_t the target ground truth. Note that the ground truth does not change through time, since the whole time series belongs to a single distance distribution, which we want to predict. Adam optimizer with a 0.001 learning rate was used, setting a total of 50 epochs to allow training convergence and input time series have been divided in batches of 64. After training, the network weights were quantized in fixed point arithmetic using simple truncation, in a Q6.10 format. After quantization the weights have been loaded into the weight memory as a memory initialization file for practical reasons, even if a simple DMA has been implemented to allow weight memory reconfigurability on-the-fly for solid state implementations.

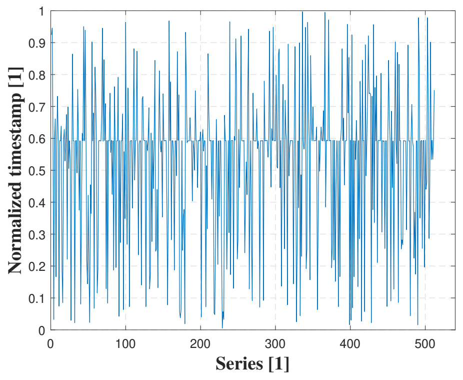


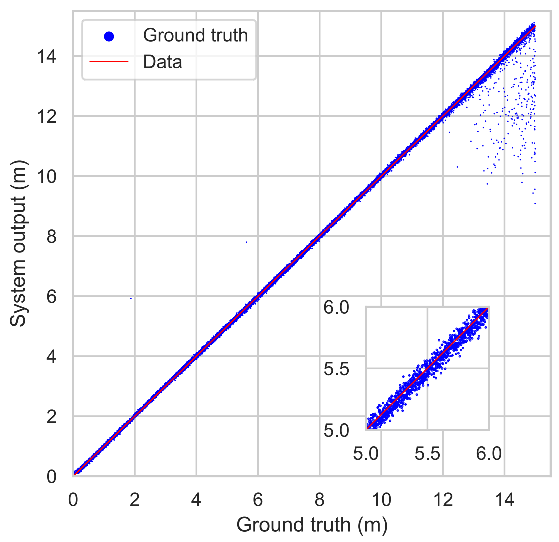
Figure 5. Example of a simulated time series used for the network training. Time values have been scaled with respect to the maximum range. In this example the distance label is 8.89 m (scaled to 0.593).

IV. Ranging Results

The system-level ranging performance of a single point SPAD sensor has been characterized. The scratchpad buffer processed by the accelerator was loaded with simulated data coming from different probability distributions for different distances, ranging from 0.05 m to 15 m with a step of 60 μm. To simulate a real-life scenario, a scanning confocal setup was built similar to [10] and a statistics of 10,000 points have been acquired for 12 different distances. The field-of-view (FoV) was covered by a generic target consisting of a white paper. The ground truth was acquired with a commercial rangefinder placed below the scanning mirrors, while parallax and offset errors were removed in post-processing. Results for the ranging measurements are shown in Fig.6.

V. 3D Imaging Results

Using the same optical setup, a 3D image was acquired. The timestamps generated by the TDCs were fed to the MLP and a standard histogram-based center-of-mass (CoM) algorithm. Since the system is event-based, no integration time is set,



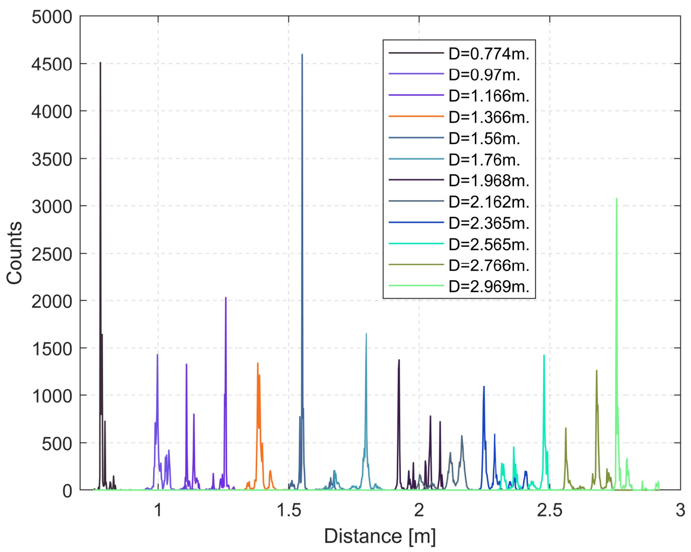


Figure 6. (Left) Simulated data. Input time series step size is 60μm. (Right) Ranging measurements. For each distance 10,0000 points are acquired and organized in a histogram, to characterize the ranging distribution of the implemented LiDAR architecture.

meaning that the acquisition proceeds when the single point has been acquired and processed. To achieve a fair comparison, for each point, we used the same scratchpad buffer where timestamps had been stored for both MLP and CoM. As expected, CoM showed less variability, while the proposed architecture can clearly distinguish the different objects in the scene. The results are shown in Fig.7.

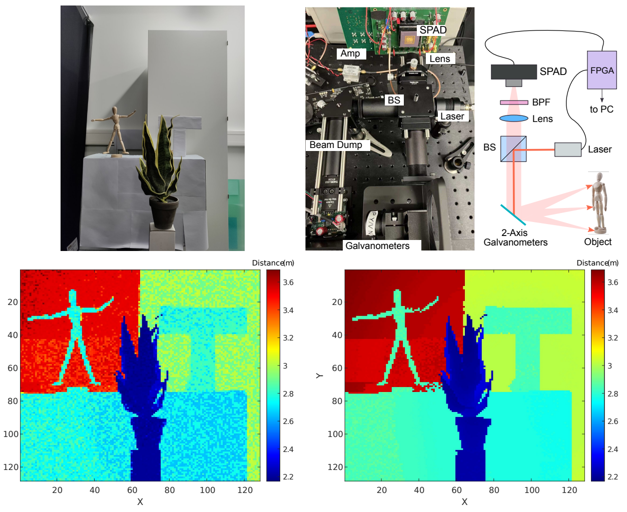


Figure 7. 3D imaging results. Top left: RGB intensity image of the target. Top right: Optical setup. Bottom left: LSTM accelerator. Bottom right: CoM.

VI. Conclusions

We presented a new histogram-less Direct Time-of-Flight architecture based on timing event processing through a machine learning processor. The processor enables individual photon timestamp processing and it is optimized for LSTM algorithm execution, with the possibility of repurposing it for other high-speed event-based applications requiring machine learning.

References

[1] A. R. Ximenes, P. Padmanabhan, M.-J. Lee, Y. Yamashita, D.-N. Yaung, and E. Charbon, "A 256× 256 45/65nm 3d-stacked spad-based direct tof image sensor for lidar applications with optical polar modulation for up to 18.6 db interference suppression," in *2018 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE, 2018, pp. 96–98.

[2] P. Padmanabhan, C. Zhang, M. Cazzaniga, B. Efe, A. R. Ximenes, M.-J. Lee, and E. Charbon, "7.4 a 256× 128 3d-stacked (45nm) spad flash lidar with 7-level coincidence detection and progressive gating for 100m range and 10klux background light," in *2021 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 64. IEEE, 2021, pp. 111–113.

[3] I. Gyongy, N. A. Dutton, and R. K. Henderson, "Direct time-of-flight single-photon imaging," *IEEE Transactions on Electron Devices*, vol. 69, no. 6, pp. 2794–2805, 2021.

[4] G. Chen, C. Wiede, and R. Kokozinski, "Data processing approaches on spad-based d-tof lidar systems: A review," *IEEE Sensors Journal*, vol. 21, no. 5, pp. 5656–5667, 2020.

[5] C. Zhang, S. Lindner, I. M. Antolović, J. M. Pavia, M. Wolf, and E. Charbon, "A 30-frames/s, 252 × 144 spad flash lidar with 1728 dual-clock 48.8-ps tdcs, and pixel-wise integrated histogramming," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 4, pp. 1137–1151, 2018.

[6] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[7] S. A. Mirsalari, N. Nazari, S. A. Ansarmohammadi, S. Sinaei, M. E. Salehi, and M. Daneshtalab, "Elc-ecg: Efficient lstm cell for ecg classification based on quantized architecture," in *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2021, pp. 1–5.

[8] D. Kadetotad, S. Yin, V. Berisha, C. Chakrabarti, and J.-s. Seo, "An 8.93 tops/w lstm recurrent neural network accelerator featuring hierarchical coarse-grain sparsity for on-device speech recognition," *IEEE Journal of Solid-State Circuits*, vol. 55, no. 7, pp. 1877–1887, 2020.

[9] A. Aßmann, B. Stewart, and A. M. Wallace, "Deep learning for lidar waveforms with multiple returns," in *2020 28th European Signal Processing Conference (EUSIPCO)*. IEEE, 2021, pp. 1571–1575.

[10] J. Zhao, T. Milanese, F. Gramuglia, P. Keshavarzian, S. S. Tan, M. Tng, L. Lim, V. Dhulla, E. Quek, M.-J. Lee *et al.*, "On analog silicon photomultipliers in standard 55-nm bcd technology for lidar applications," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 28, no. 5, pp. 1–10, 2022.