Design of a Reconfigurable 3D Pixel-Parallel Neuromorphic Architecture for Smart Image Sensor

Pankaj Bhowmik, Md Jubaer Hossain Pantho, Marjan Asadinia, Christophe Bobda
University of Arkansas
Fayetteville, Arkansas, USA
{pbhowmik, mpantho, masadini, cbobda}@uark.edu

Abstract

Power reduction and speed-up of image processing algorithms remain of high interest as image resolutions continue to increase. Neuromorphic-circuits are inspired by the nervous system aiming to reduce power consumption and speed-up. This paper presents a neuromorphic smart image sensor designed by the pixel-parallel 3D hierarchical architecture with an on-chip attention module. The module dynamically detects regions with relevant information and produces a feedback path to sample those regions at high speed. On the other hand, by sampling non-relevant regions with a low-speed, the sensor can reduce redundancy and enable high-performance computing by ensuring low-power operation. The image sensor is comprised of several hierarchical planes and each plane has small and independent reconfigurable computational units (XPU). In each plane, all XPUs operate in parallel with a different operating speed which gives a pixel-parallel architecture. When the raw image passes through the hierarchical planes, necessary image processing algorithms are performed in parallel on different planes at a variable clock rate for saving power and reducing redundancy. The goal of this work is to prototype the focal plane image sensor which emulates the brain features. The results show that the prototype achieves remarkable power saving and speed-up at different stages.

1. Introduction

Cameras are pervasively used for surveillance and monitoring applications and can capture a substantial amount of image data [1, 2]. The processing of this data is either performed a-posteriori or at powerful back-end server. On the other hand, most camera systems are used as data collection and relaying units while the processing happens at those servers. Posteriori and non-real-time video analysis may be sufficient for certain groups of applications. However, it does not suffice for applications such as accident determination and distracted driving detection image analysis using cameras on drones, that require near real-time video and image analysis, sometimes under SWAP (Size Weight and Power) constraints. Given the raw amount of data captured from cameras and the lack of reliable high bandwidth wireless connectivity that can facilitate the transfer of image data to back-end servers, we hypothesize that future data challenges in camera sensors can be overcome by pushing computation *close* to the image sensor. Such systems will exploit the massively parallel nature of sensor arrays to reduce the amount of data analyzed at the processing unit. To this end, vertically integrated technology, such as focal plane sensor processors (FPSP) [3, 4], have been developed to overcome the limitations of conventional image processing systems. FPSP refers to a block which includes a photoreceptor along with a local pixel processor. Figure 1 illustrates the focal plane processing where each block represents an FPSP and works on only the pixel received by its photoreceptor. Research on FPSPs has mostly focused on technology aspects with some proof of concepts. Vision sensors that incorporate general-purpose digital processors on the focal plane have been a subject of a number of commercial developments. While these devices are re-programmable and offer the benefits of in-sensor processing such as performance and bandwidth reduction, they exhibit many drawbacks. For instance, each column of pixels is handled by a single processor, which reduces the parallelism and all pixels are treated equally and processed at the same rate, despite differences in input relevance for the application at hand. Consequently, systems spend more time spinning on non-relevant data, which increases sensing, computation time, and power consumption. System-on-chip designs that incorporate hardware accelerators have been considered a viable solution in recent years to provide in-situ efficiency in image processing applications [5–8].

However, the conventional hardware accelerators execute in a sequential pixel read-out manner, which restricts the architecture from exploiting the full extent of the image's parallel nature. In those areas where high-speed image acquisition is required, a fast collection of image pixels and

processing should be ensured at the minimum span of time. In order to achieve this goal, highly parallel architecture design is inevitable.

To overcome the limitations of existing architectures, in this paper, we present the design of a highly parallel, hierarchical, reconfigurable and vertically-integrated 3D sensing-computing architecture for real-time, and low-power video analysis. To increase performance, while reducing power, the proposed architecture leverages the concept of the biological vision systems to reduce redundancy and deploy more resources on an important part of scene images. Visual attention is used by the brain to rapidly detect and deploy more resources to salient parts of a given scene [9], more precisely, it allows the brain to remove redundancy and transfer only useful information to high-level parts of the brain for further processing. The paradigm is implemented in the brain in a chain of fast feedforward signals that carry information to high-level part of the brain, while feedback signal provides configuration to the lower part of the visual cortex [10].

As shown in Figure 2, our presented architecture exploits saliency-based visual attention found in the brain, along with maximal parallelism, which results in a hierarchical image processing hardware architecture made of computational units that reside in three inter-twined logical planes. The first plane in the figure consists of fine-grained reconfigurable components that collaboratively analyze a collection of pixels to detect the early visual feature of the input image. The results of this step are fed into the next plane where relatively higher-level image processing (for instance line, circle, triangle, motion detection, feature extraction for recognition, etc.) is performed and mapped on salient events in an image. The map is then searched for events and objects in the third plane of computation. The higher sampling frequency is used in relevant regions, which are dynamically detected with relevant information, and produces a feedback path. In order to detect saliency, we extract the early visual features such as the number of edges or corner pixels in different regions of the image. We combined this knowledge of features for a region to calculate visual saliency. To check the viability of our design, we simulated our architecture on Virtex-7 [11]

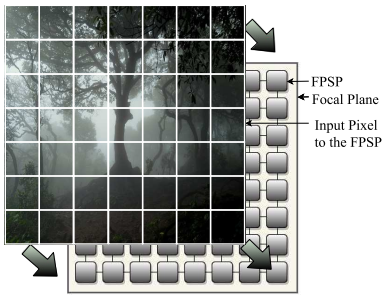

Figure 1. Focal Plane Image Processing; *Each block in the image corresponds to an Focal Plane Sensor Processor.*

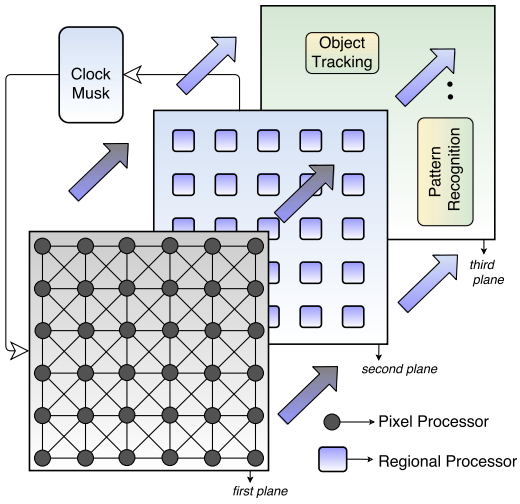

Figure 2. Overview of the 3D bottom up architecture of the smart image sensor. The computational units are organized in planes, where the output of each layer serves input for the next one.

FPGA and we extracted the basic constraints of each module from Design Compiler [12] and Innovus [13]. The results show that by trading off resource overhead we can obtain high throughput while reducing redundancy and power consumption. The proposed architecture can be applied to a large set of image processing applications where real-time operation is needed.

The remaining sections of the paper are organized as follows. Section 2 presents related work. We discuss about our proposed architecture in Section 3 . A real-time application is demonstrated as a case study and we analyze it in Section 4. Section 5 provides the implementation and experimental results that justify the viability of our design. Finally, we conclude the paper in Section 6.

2. Related Work

In this section, we will review relevant research work that have been developed to reduce redundancy and to overcome the limitations of conventional image processing systems. Then we will discuss different studies on visual attention approaches.

Understanding the concept of visual attention is complex and fortunately neuroscientists are working to devise a brain-like processor [14, 15]. In a brain-like chip, one of the challenges is the parallel image acquisition from nature by designing a on-chip sensor. Tyrrell *et al.* proposed a per-pixel architecture of CMOS digital focal plane readout for the orthogonal-transfer-based real-time digital signal processing [4]. Their per-pixel parallel architecture combines with image detection and signal processing which keeps all processors busy. They achieved high speed-up but it imposes high power consumption. The integration of complex processing may consume more power and this might

be responsible for the design bottleneck. Since high power consumption is a limitation in pixel-parallel architecture, a 3D design can be a viable solution to mitigate this problem. In [16], authors showed that by using 3D design they can achieve 50% and 35% savings in power consumption and area overhead, respectively. Adding to it, 3D connection gives a neuron-like connectivity among the circuit elements in both lateral and transverse direction which gives a brain-like design.

Alternately, the power dissipation can be further reduced by minimizing the redundancy within the system. A natural image contains a lot of irrelevant pixels or redundancies and there is research [6, 17–19] works to truncate those irrelevant pixels. Streaming data reduction is a form of reducing redundancy which is performed in image processing systems using region of interest (ROI) based strategies to limit the computation of data only to regions with high relevance. To reduce communication bandwidth, it is built and sent across the network in collaborative tracking systems such as profiled data of tracked target [17]. Many ROI-based computations [16-20], are used in image data compression to further reduce the amount of data to be transported, thus increasing the compression ratio in reducing transported data. Computational models of visual attention have been proposed in many variations [6, 18] to emulate the human reaction to scene events and allow systems to focus on the most important ones. In this line, the main concern of those presented works are related to the visual attention mapping and tracking ROI in different approaches.

It is notable that all these work concentrate on how the biological vision systems focus on an image. The knowledge was not further investigated for an efficient design aiming to reduce the power consumption in a circuit. The design of an on-chip visual attention module enables us to generate the saliency map. In [19], the authors used the attention module in an image sensor for detecting the saliency map. The design has the focal plane attention based image sensor but it is also limited to ROI generation. In addition, the use of one-bit analog to digital converter (ADC) is a limitation in high-performance computing.

Most recently, Intel introduced a prototype of neuromorphic chip loihi [22] with a self learning capability to mimic the processing in the brain. The prototype is 1000 times more energy efficient than the general purpose processor and achieves a million times faster learning rate than typical spiking neural nets. They incorporated the concept of high speed parallel computing. However, the design does not consider in-sensor computation.

Work on reconfigurable architectural design approaches, for mapping saliency map to reduce power consumption and speed-up, has not been investigated previously. In this paper, we explored the neuromorphic smart image sensor based on FPSP where we incorporated the on-chip attention module to find out the relevant regions and process those regions at high speed. In addition, the pixel-parallel 3D architecture gives the design maximal parallelism and provides low power consumption.

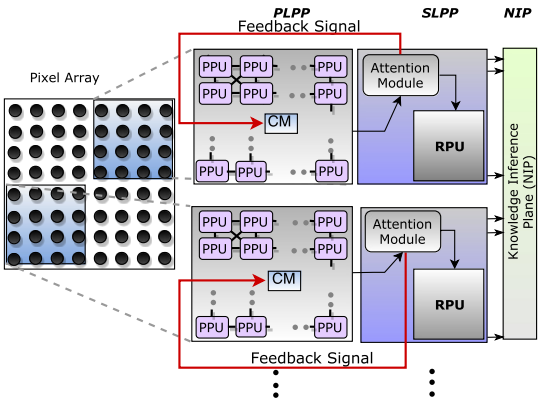

Figure 3. PPU-RPU interconnection structure with feedforward and feedback connections.

3. Architecture Overview

In this section, we describe the overall architecture of our design. Our proposed architecture is organized into three planes namely Pixel-Level Processing Plane (PLPP), Structure-Level Processing Plane (SLPP), and Knowledge Inference Plane (NIP). The planes are comprised of reconfigurable processing units to meet the computational needs of an application. We present the functional block of each hierarchical plane in Figure 3, where the PLPP receives an array of pixels in parallel and extracts early visual features. The features are then fedforward to the SLPP plane, where comparatively complicated processing is performed. The output of each processing unit is fedforward to the NIP for complex operation. We describe the functional details of the hierarchical planes below.

3.1. Pixel-Level Processing Plane (PLPP)

The PLPP is the first stage in the hierarchy which is responsible for image acquisition and low-level image processing as shown in Figure 3. This plane has two major components, Pixel Processing Unit (PPU) and Clock Musk (CM). In this figure, we have shown only two groups of PPU by showing them in two squares and each group is driven by a single CM.

The PPU contains pixel circuit, ADC, Interconnect Manager (IM), and a Digital Processor (DP). Here, each PPU is connected with its neighboring PPUs. The detail orientation of PPUs is depicted in Figure 4. The pixel circuit contains a photodiode (PD) and an amplifier. The PD creates a photocurrent when photon energy imposed on it and the amplifier converts it to a voltage. The ADC has a Sample and hold (S/H) unit which samples the analog signal and

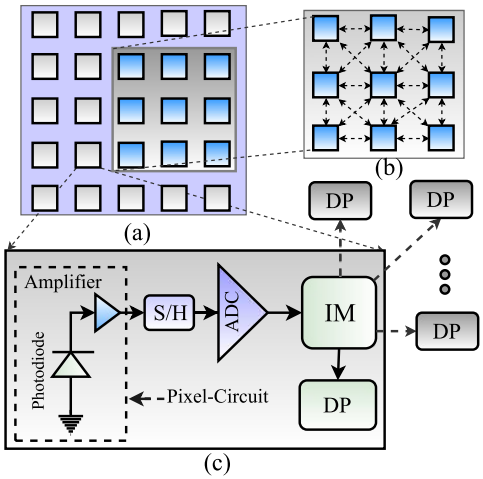

Figure 4. (a) PPU array in the PLPP layer (b) Interconnection among the PPU (c) Components of a PPU includes *Pixel Circuit, ADC, Interconnect Manager, and Digital Processor.*

maintains a constant voltage level for a specific amount of time. The ADC converts the constant analog voltage into an 8-bit digital value in that time instance. Hence, the imposed scene on the focal plane is transformed into an 8-bit gray-scale image. The ADC is followed by an IM as shown in Figure 5 which instantly routes the data to its DP and to the neighboring DPs at a time. In this figure, the IM-4 forwards the data to DP-4 and its neighboring DPs which are numbered as DP-0 to DP-8. In the same way, IM-5 routes the data from ADC-5 to DP (3˜11). Instead of storing the data, the IM updates pixel values in each clock cycle.

The architecture of the PPU gives the option to perform independent operations in the PPU in parallel which gives a pixel-parallel architecture with high throughput. Each DP has its own pixel value along with its neighboring values. With these values it performs low-level operations for instance edge or corner detection, thresholding, smoothing, inversion, filtering, and morphological operations like dilation and erosion. Those operations determine whether a pixel lies on an ROI.

A clock-musk unit is connected to each PPU for assigning appropriate clock frequency which is shown in Figure 6. A group of PPUs is connected with an XPU in the second plane, and the XPU gives a feedback signal to the corresponding

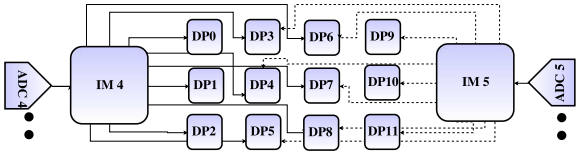

Figure 5. Signal flow from the ADC to Digital Processors(DP): *ADC-4 is connected to DP-4 and its neighbor, and similarly ADC-5 is connected to DP-5 and its neighbor.*

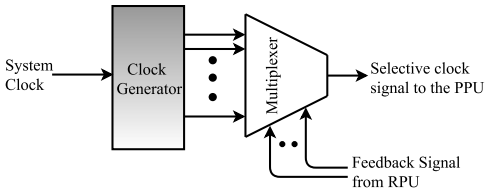

Figure 6. Clock Musk (CM) unit: *System Clock is divided into number of clocks and feedback signal determines the appropriate clock for the Digital Processor.*

CM. The XPU specifies the suitable clock frequency for the PPU group and passes the information by the feedback signal. The clock assignment allows the design to perform a multi-clock computation in the PLPP layer by maintaining the consistency among signals in parallel. The purpose of the clock musk is to slow down operations in irrelevant regions of an image.

Our proposed architecture in the PLPP leverages the parallel nature of pixel-sensing and low-level processing simultaneously. The focal plane computation in the PLPP layer provides high throughput at high-speed.

3.2. Structure-Level Processing Plane (SLPP)

The SLPP is the second stage in the hierarchical model as shown in Figure 3 and the plane takes only the features as input from the PLPP. Features have less data than the input image. Hence, the data volume reduces for the SLPP plane. This layer has a number of XPUs and they work as independent units and execute the assigned operation in parallel. The XPU generates a distributed output with these features and the output is forwarded to the NIP by a bus. All XPUs in this layer have the same configuration and generate an output at the same time. In addition, the SLPP is connected to the PLPP plane by a feedback signal. The XPU in this plane has two components, attention module and Regional Processing Unit (RPU). Figure 7 provides the closer view of the XPU of the SLPP layer and we present the detail functionalities in the following subsections.

3.2.1 Attention Module

The attention module is the brain of the SLPP layer which receives the extracted early visual features from the PLPP in parallel and generates a feedback and a feedforward signal

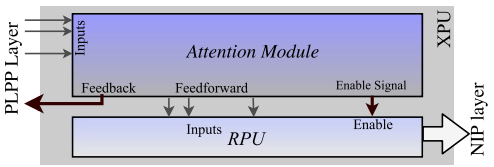

Figure 7. XPUs single flow in the SLPP: The Attention Module and RPU.

to drive the PPU and RPU respectively and it is shown in Figure 7.

The module is responsible for computing the visual saliency in a region. If a region has inadequate visual information, then the attention module generates a saliency score which is less than a threshold. In this case, this module puts zero to the enable port of the RPU and puts zero in the feedback path to select the slowest-clock from the CM. This assignment makes RPU to postpone its execution and initiates the slowest processing in the PLPP layer form the next image frame for that region. Alternately, when the saliency score is greater than threshold, then the attention module enables the RPU and selects faster-clock based on the score.

In biological vision systems, the brain focuses on the points where there is enough visual information and other portions are ignored. Similarly, our architecture temporarily ignores irrelevant regions by assigning a slow-clock to them. Alternately, when saliency score becomes significant, our design assigns fast-clock to that region to execute those pixels with high importance.

3.2.2 Regional Processing Unit (RPU)

The RPU has a more coarse-grained processor and operates on broader regions of the image than the PPU. Where the PPU is responsible for only one pixel, the RPU processes on the group of pixels as shown in Figure 3. The figure indicates, there is one RPU for each PPU group in the PLPP.

Figure 7 presents an elaborated functional view of the RPU. The RPU receives the input from the attention module with an enable signal. After that, it starts executing and sends the distributed output to the NIP by a bus for further processing. This processor is always assigned with the fast-clock and maintains an asynchronous connection with the attention module. The RPU is a reconfigurable unit and we can implement different kinds of complicated computer vision applications for instance line, circle, blob, shape, object detection on this processor. The PLPP performs the preliminary job of a complex algorithm and this reduces the execution overhead. The computation in the RPU is discrete and don't have the knowledge of the entire image.

One of the objectives of the design is to save power. When an RPU is not receiving the enable signal from the attention module, it remains idle and doesn't consume power. Hence, the possible power saving is proportional to the number of idle RPUs in the SLPP layer.

3.3. Knowledge Inference Plane (NIP)

The output of the SLPP is combined with the inference knowledge of a scene. Sequences of distributed features such as lines, circles, rectangles, and contours can be combined with infer knowledge of a scene. As opposed to the PLPP and SLPP, which operate on large amounts of data, the features are vastly reduced in the NIP to the extent that they can be processed sequentially by an embedded processor with average performance. The integration of our relevance-feedback method as explained earlier, further limits features to only relevant regions of an image. The knowledge inference plane implements clustering and some other processing preferred by the user to accumulate all the segregated information obtained from the RPUs. In this design, the NIP receives inputs from each RPU in parallel. In addition, those RPUs which are inactive and don't have the line equation will be discarded by the NIP. As a consequence, the number of effective inputs for the sequential processor further decreases and this feature enhances the system performance by speeding up the sequential operation.

The designed system-on-chip architecture, with a low-frequency (in the 100 MHz range) processor, enhances the speed-up and performance by reducing the effective inputs.

3.4. The Overall Architectural Evaluation

The key feature of the hierarchical architecture using XPUs is the maximal parallelism provided vertically and horizontally within and across processing planes. In addition, the three layers described above maintain a hierarchy and each layer communicates with its adjacent layers in real-time. The layered design introduces a 3D pixel-parallel structure. If we come down from the PLPP to the NIP, there is a gradual degradation in data volume but gradual increase in the image processing complexity; which is a common feature of a bottom-up architecture in the brain. In the nervous system, as we go deeper from the retina to the deep layer, the complexity of processing increases. The visual attention scheme in the nervous system decreases the data volume from layer to layer. In the human visual system, from Retina to Layer-4 early visual features are extracted, complex processing is carried out in Layer-5, and the deep-layer accumulates all information and gives the final output and there are feedback and feedforward connections among the layers [10]. Transitioning from the human visual system to our design, the PLPP emulates retina to Layer-4, the SLPP imitates Layer-5, and the NIP acts like the deep layer.

Therefore, in our work, we have emulated the basic concept of the biological visual system in a circuit by designing a pixel-parallel focal plane smart neuromorphic image sensor with a bottom-up hierarchical 3D architecture.

4. Case Study

To rationalize our architecture, we reconfigure our system for a specific real-time application and compare our design to existing solutions found in the literature [3], [23]. In this section, we provide a real-world use case scenario of lane detection method and analyze the performance in terms of speed-up and power saving.

The conventional approaches for lane detection are followed by some sequential operations as represented by a flowchart depicted in Figure 8. The image sensor reads an image and forwards it to the back-end processor. The processor performs preliminary computations such as noise reduction, image enhancement or image smoothing which are followed by an ROI generation. After that, an edge detection algorithm is applied. The edge detected image provides the features for line detection. The output of the line detection algorithm can generate several lines but a few of them represent lanes. A clustering or classifying algorithm gives possible lanes from the multiple lines.

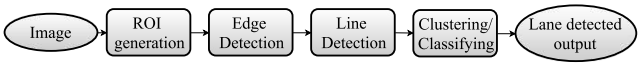

Figure 8. Block diagram of Lane detection.

When a sequential processor computes all those computations, it becomes hard to meet the demand for real-time operation. In Figure 8, the edge detection and line detection steps are computationally expensive for a sequential processor. To overcome this limitation, researchers normally achieve speed-up by introducing hardware-accelerators and run individual portions of the execution which are computationally expensive. In [23], the authors proposed a system-on-chip architecture for real-time lane detection. Images were acquired from an image sensor mounted on the windshield of a car to provide lane departure notification. The edge and the line detection were performed in a hardware accelerator and the rest of the operations were computed on the sequential processor. The authors reported an overall speed-up of $2.09\times$ with respect to software implementation. For a larger image, the sequential operation will require more computation time and reduce the frame rate as a consequence. In [24], authors presented a nice correlation with frame rate with image size. They showed the gradual degradation in frame rate with the gradual increase in pixel count in a sequential processor implemented on a hardware accelerator.

However, our architecture solves this problem by dividing and assigning the computer vision application on those three layers. In the first layer, we implement edge detection on DPs and these processors convert a grayscale image into a binary image which is fed into the SLPP plane. Since the PLPP takes an 8-bit input image and gives a one-bit output image, the PLPP layer contributes to reduce $8\times$ data volume. The SLPP divides the entire image into several regions. The attention module receives data and computes the saliency score for that region. When a score is greater than a threshold, we assign fast clock in the PLPP and the RPU looks for the possible line equations on the block. Conversely, the RPU halts the execution when the score is less than a threshold and Digital Processors in the group are assigned to a $10\times$ slower clock from the next clock cycle. At the same time, From the SLPP layer, a distributed line equations are forwarded to the NIP. When a region does not have enough visual information, will not send any data to the NIP. Based on the input image, there will be a number of XPUs in SLPP that will remain idle. This inactive RPUs will reduce the data volume in the NIP. In NIP, we execute a clustering algorithm which collects the distributed knowledge and populates lines on the lane. We have reported the performance of each stage in section5.

5. Experimental Result

We organize this section with the Evaluation and implementation detail of our model and then followed with a discussion on the performance of our architecture, which includes resource utilization, power consumption, area projection, timing information, and speed-up.

5.1. Evaluation Infrastructure

We used the Virtex-7 [11] FPGA board from Xilinx as an evaluation platform to prototype our model. The RTL analysis of the prototype gives the latency, resource utilization, and power consumption of each hierarchical plane. We tested each unit in the Application Specific Integrated Circuit (ASIC) domain to meet the industry standard for fabrication. We used Design-Compiler [12] from Synopsys and Innovus [13] from Cadence to achieve the layout design for each module in 90nm technology. From the layout design, we accumulate the area, power, timing constraints and then analyze those values with different operating conditions.

5.2. Implementation Detail

This architecture we have presented can be modeled for different computer vision applications by splitting the entire operation into three segments and distributing them in the three layers. In this section, we describe our implementation for the real-time high-speed lane detection as described in the section 4. The following sections describe the performance in different planes.

5.2.1 Performance of the PLPP Layer

In the reconfigurable PPU, we have implemented Sobel edge detection for achieving early visual features. We used Design Compiler Ultra [12] and Innovus [13] to design the layout of the DP, IM, and CM using 1 GHz clock frequency. We have summarized the layout extracted parameters in Table1.

The table shows that area overhead of the processor is low and this gives an advantage to improve the fill-factor of the image sensor. In [3], authors spend 43% area for the Digital Processor and 27% for a memory unit in the focal plane which reduce the fill-factor.

It is notable that, we didn't include any memory unit to store the pixel data in the PLPP layer rather we routed the data in vertical and horizontal direction. Hence, the overall

Table 1. Key Functional Modules Specification in the PLPP Layer

Performance Matric	DP	CM	IM
Area (um^2)	380.11	596.37	45.22
Power (uW)	918.7	561.6	100.1
Timing (ns)	0.943	0.786	0.469
Wire-Length (um)	1121	1947	21.58
No. of Cells	199	170	17
Cell Density(%)	65.67	59.6	60.2

Table 2. The possible Power Saving in the Processor by increasing Time-Period with 2GHz reference-clock

	Time Period					
	1X	2X	4X	8X	10X	20X
Power Consumption (uW)	953.7	480.4	243.8	125.5	101.8	54.5
Power Saving (%)	0	49.62	74.43	86.84	89.33	94.28

area reduces which improves the fill-factor. In 90nm technology, the pixel circuit consumes an area of 401.6um^2 [25] and the ADC consumes 0.021mm^2 [26]. The estimated active area for a VGA image (640 x 480) size will be 7250mm^2 which has 307200 PPUs. The chip size is compatible with the available neuromorphic chip [22]. In the PPU, an ADC occupies major portion. The use of newer technology (32nm) with the advanced method (spintronics) can reduce the ADC area to 10um^2 [27].

In the PLPP layer, the CM saves power by assigning multiple clocks in the plane. In this processor, we applied different clock frequencies ranging from 2GHz to 25MHz to observe the variation in power consumption. The simulation result obtained from design compiler [12] is given and have been tabulated in Table 2. The analysis shows that when 10 times slower clock is applied, the processor saves 89.33% power compared with the reference clock-frequency (2GHz).

We designed those modules in an FPGA to analyze the behavior and summarized in Table 3. Here, the DP takes 2 and IM takes 1 clock-cycle and the PLPP updates the output every 2 clock-cycle.

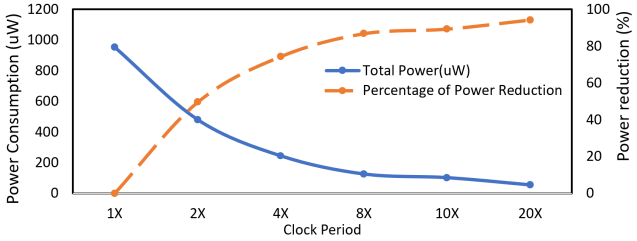

Figure 9. Power Consumption versus time period analysis in the DP: The 10 times slower clock has 89% power savings.

Table 3. Resource Utilization and Timing in FPGA

Module	FFs	LUTs	Clock Cycles
DP	1	91	2
IM	1	8	1
CM	25	35	2

5.2.2 Performance of the SLPP Layer

In the SLPP layer, we implement Hough Line Transform (HLT) algorithm to detect lines in each region. This algorithm looks at many possible lines in an image and among them, the horizontally inclined lines do not represent a lane. In our implementation, we prevent those lines by adding a filter in HLT algorithm. When a line makes an angle in between 30° -150° with the horizontal axis, our RPU considers that as a line. This approach prevents false line detection and lessens the burden to the sequential processor.

Although the reconfigurable units can be tailored to a larger region, for simplification, we have considered an RPU is connected to an 8x8 region (64 PPUs) for extracting parameters. Table 4, which presents different constraints of the XPU unit shows that the attention module consumes only 4.98% power with an 2% area overhead in the XPU. To this extent, we can save 95.02% power consumption from an XPU when the RPU is inactive. In a complete design, numbers of RPUs will remain inactive which determines the amount of power savings from the SLPP layer.

In Table 5 we provide the FPGA resource utilization and timing performance for line detection. This table shows that the attention module generates outputs in every 2 clock-cycles and continues its surveillance for finding an ROI. However, the SLPP layer has 8-times reduced data volume than the PLPP layer and inactive RPUs save power and reduce redundancy of an image.

Table 4. Key Functional Modules Specification in the SLPP Layer

	RPU	Attention Module
Area (um^2)	42438.438	894.824
Power (uW)	26710	1402
Timing (ns)	1.459	0.959
Wire Length	696018	151
No. of Cells	29048	25
Cell Density(%)	74.81	58.81

Table 5. Resource Utilization and Timing performance in FPGA

Module	FFs	LUTs	Clock Cycles
RPU	13198	85446	1650
Attention Module	74	90	2

Table 6. Overall Resource utilization in FPGA

	PLPP		SLPP	
	Usages	Utilization(%)	Usages	Utilization(%)
LUTs	57339	18.9	257148	84.69
FFs	1377	0.23	40260	6.64
Clock	3	-	1652	-

5.2.3 The Overall Implementation

We have tested the design for a comparatively small image in an FPGA to check the end to end connectivity. The design is comprised with 576 PPUs and 9 RPUs. The main objective of the FPGA implementation is to test the design integrity for a small image by analyzing the performance and resource utilization. We have implemented the layers separately in the FPGA and tabulated the result in Table 6. Those results show that when an image imposed on the PLPP, with a 100MHz system-clock, we get discreet line equations after 165ns. The drawback of the design is that it consumes more resources for the nature of parallel execution. A biological vision systems also have huge resources but the utilization of these resources remains low. Transitioning to our design, we need more resources and with have redundancies in resource utilization. The prominent feature of the design is that the pixel-parallel architecture is highly scalable and has the same behavior for large images.

Besides, we performed software simulations to test the behavior of different planes with pragmatic image sizes. Figure 10 shows the software simulated output images of each layer. Figure 10 (a) is the input image to the PLPP and this layer gives edge detected image using Sobel edge detection and shown in Figure 10(b). We have considered six RPUs which divides the image into six groups. If each group has enough edge pixels, then the six RPUs will work in parallel to generate line equations. In Figure 10(c), three regions haven't enough edge pixels and our architecture keeps them inactive. As a result, we can achieve nearly 47% power savings in the SLPP layer according to the simulation in section 5.2.2. Finally, in Figure 10(d), we showed the lane detected output image which is obtained in the NIP.

5.3. Performance Analysis

PLPP an SLPP perform two operations: edge detection and line detection. In [3], edge detection took 84 cycles and in [23], they took 2.589ms. Comparing with them, our PLPP takes only 3 clock-cycles or 30ns for edge detection. Alternately, the author in [23], reported 41ms was required to compute lines in an image with an image size of (320x37). In contrast, our SLPP layer completes line detection in 165ns in FPGA with 100MHz clock. However, the execution time doesn't depend on the image size in our design and the parallel nature achieves high speed-up in each stage.

The ASIC implementation gives the CMOS level information for each module. We perform area, power, and timing analysis for all modules to design a neuromorphic chip. Alternately the FPGA implementation gives the end to end integrity of the design. This implementation gives the proof that the design is reconfigurable and scalable by maintaining the same performance. Finally, the software simulation shows the outcome of each stage of the hierarchical planes.

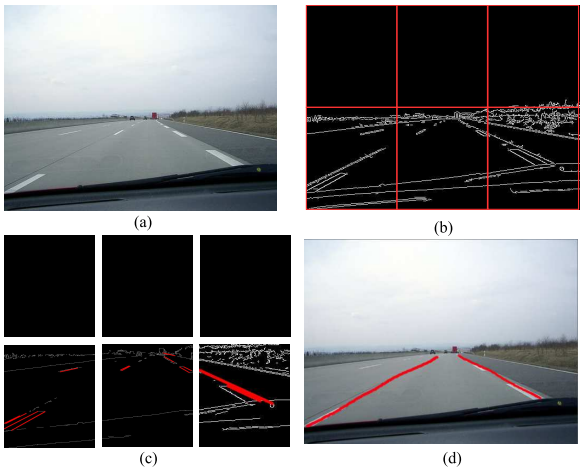

Figure 10. Images in the Hierarchical Plane, (a) Image input in the PLPP layer (b) Output of the PLPP (c) Output of the SLPP (d) Output of the NIP layer. The lanes are highlighted in the main image with red marks [28].

6. Conclusion

In this paper, we presented a pixel-parallel 3D architecture of a neuromorphic image sensor. The sensor uses different sampling frequencies in different regions of an image to enable high-performance computing at low power. The smart sensor is designed as a 3D bottom-up architecture composing of several computational planes where each plane performs different image processing algorithms in a highly parallel manner. The model emulates the hierarchical process of the nervous system by providing feedforward and feedback information flow between different planes. The designed attention module dynamically detects regions with relevant information and produces a feedback path to sample those regions with a higher clock frequency. We have tested the viability of our design by prototyping it on an FPGA and ASIC platform. The results showed that by trading off resource overhead we can obtain high speed-up while reducing redundancy and power consumption.

7. Acknowledgement

This work was supported by the National Science Foundation (NSF) under Grant-1618606.

References

[1] T. Winkler and B. Rinner, Applications of trusted computing in pervasive smart camera networks, In *Proceedings of the 4th Workshop on Embedded Systems Security*, ser. WESS 09. New York, NY, USA: ACM, 2009, pp. 2:12:10.

[2] F. J. Streit, M. J. H. Pantho, C. Bobda, and C. Roullet. Vision-Based Path Construction and Maintenance for Indoor Guidance of Autonomous Ground Vehicles Based on Collaborative Smart Cameras, In *Proceedings of the 10th International Conference on Distributed Smart Camera*, 2016.

[3] W. H. Robinson, D. S. Wills. Design of an integrated focal plane architecture for efficient image processing, *15th International Conference on Parallel and Distributed Computing Systems*, 2002.

[4] B. Tyrrell *et al.* Time Delay Integration and In-Pixel Spatiotemporal Filtering Using a Nanoscale Digital CMOS Focal Plane Readout, *IEEE Transactions on Electron Devices*, vol. 56, no. 11, pp. 2516–2523, 2009.

[5] J. Anders *et al.* A hardware/software prototyping system for driving assistance investigations, *Journal of Real-Time Image Processing*, vol. 11, 2016.

[6] M. S. Park, C. Zhang, M. DeBole, and S. Kestur. Accelerators for biologically-inspired attention and recognition, In *Proceedings of the 50th Annual Design Automation Conference (DAC–13)*, 2013.

[7] B. da Silva *et al.* Comparing and combining GPU and FPGA accelerators in an image processing context, *2013 23rd International Conference on Field programmable Logic and Applications*, 2013.

[8] V. Dworak *et al.*Strategy for the development of a smart NDVI camera system for outdoor plant detection and agricultural embedded systems, *Sensors*,vol. 13, no. 12, pp. 1523–1538, 2013.

[9] R. Desimone and J. Duncan. Neural mechanisms of selective visual attention, *Annu. Rev. Neurosci.*, vol. 18, no. 1, pp. 193–222 vol 18, 1995.

[10] G. Michalareas, J. Vezoli, S. Van Pelt, J. M. Schoffelen. Alpha-Beta and Gamma Rhythms Subserve Feedback and Feedforward Influences among Human Visual Cortical Areas, *Neuron* vol. 89, no. 2, 2016.

[11] Xilinx. VC707 Evaluation Board for the Virtex-7 FPGA, 2016.

[12] Synopsys. Design Compiler Ultra, *Concurrent Timing, Area, Power, and Test Optimization*, https://www.synopsys.com [Online; accessed 15-March].

[13] Cadence. Innovus Implementation System, https://www.cadence.com, [Online; accessed 12-March-2018].

[14] J. Sabarad et al. A reconfigurable accelerator for neuromorphic object recognition, In *Proceedings 2012 17th Asia and South Pacific Design Automation Conference*, 2012.

[15] J. Kogler, C. Sulzbachner, and W. Kubinger, Bio-inspired stereo vision system with silicon retina imagers, Comput. Vis. Syst., vol. 5815, pp. 174183, Jan. 2009.

[16] Swaminathan and Madhavan. Electrical design and modeling challenges for 3D system integration, *Design Conference*, 2012.

[17] Y. Wang, S. Velipasalar, and M. Casares. Cooperative Object Tracking and Composite Event Detection With Wireless Embedded Smart Cameras, *IEEE Transactions on Image Processing*, vol. 19, 2010.

[18] L. Itti and C. Koch. Computational Modelling of Visual Attention, *Nature reviews neurosci.*, vol. 2, 2001.

[19] M.C. Park, K.J. Cheoi, and T. Harnarnoto. A smart image sensor with attention modules, *Seventh International Workshop on Computer Architecture for Machine Perception (CAMP05)*, 2005.

[20] Z. Chen, G. Barrenetxea and M. Vetterli. Event-driven video coding for outdoor wireless monitoring cameras, *19th IEEE International Conference on Image Processing*, 2012.

[21] S. Bae, Y. C. P. Cho, S. Park, K. M. Irick, Y. Jin, and V. Narayanan. An FPGA Implementation of Information Theoretic Visual-Saliency System and Its Optimization, *19th Annual International Symposium on Field-Programmable Custom Computing Machines*, 2011.

[22] M. Davies et al. Loihi: A Neuromorphic Manycore Processor with On-Chip Learning, *IEEE Micro*, vol. 38, no. 1, pp. 82-99, 2018.

[23] C. Bobda, M. J. H. Pantho, C. Roullet, A. Bensrhair, and S. Ainouz. SoC Design Of A Novel Cluster-Based Approach for Real-Time Lane Detection in Low Quality Images, *11th International Conference on Distributed Smart Cameras*, 2017.

[24] T. Santti, O. Lahdenoja, A. Paasio, M. Laiho, and J. Poikonen. Line Detection on FPGA with parallel sensor-level segmentation, *2014 14th International Workshop on Cellular Nanoscale Networks and their Applications (CNNA),* 2014.

[25] Fossum et al. *CMOS ACTIVE PIXEL SENSOR TYPE IMAGING SYSTEM ON A CHIP*, 5841126, 1998.

[26] P. Harpe, C. Zhou, X. Wang, G. Dolmans and H. de Groot. A 12fJ/conversion-step 8bit 10MS/s asynchronous SAR ADC for low energy radios, In *Proceedings of ESSCIRC, Seville*, 2010.

[27] Q. Dong *et al.* Low-Power and Compact Analog-to-Digital Converter Using Spintronic Racetrack Memory Devices, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2017.

[28] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, ImageNet: A Large-Scale Hierarchical Image Database, www.image-net.org/papers/imagenet_cvpr09.bibCVPR09, *CVPR*, 2009.