An SoC FPGA-based Integrated Real-time Image Processor for Uncooled Infrared Focal Plane Array

Hyun Woo Oh*, Cheol-Ho Choi, Jeong Woo Cha, Hyunmin Choi, Joon Hwan Han, and Jung-Ho Shin
Infra Technology R&D Center
Hanwha Systems, Seongnam, Republic of Korea
{hyunwoo.oh, cheoro1994, jeongwoo.cha, hyunmin.choi, joonhwan.han, jh.hoya.shin}@hanwha.com

***Abstract*—This paper presents an integrated image processor architecture designed for real-time interfacing and processing of high-resolution thermal video obtained from an uncooled infrared focal plane array (IRFPA) utilizing a modern system-on-chip field-programmable gate array (SoC FPGA). Our processor provides a one-chip solution for incorporating non-uniformity correction (NUC) algorithms and contrast enhancement methods (CEM) to be performed seamlessly. We have employed NUC algorithms that utilize multiple coefficients to ensure robust image quality, free from ghosting effects and blurring. These algorithms include polynomial modeling-based thermal drift compensation (TDC), two-point correction (TPC), and run-time discrete flat field correction (FFC). To address the memory bottlenecks originating from the parallel execution of NUC algorithms in real-time, we designed accelerators and parallel caching modules for pixel-wise algorithms based on a multi-parameter polynomial expression. Furthermore, we designed a specialized accelerator architecture to minimize the interrupted time for run-time FFC. The implementation on the XC7Z020CLG400 SoC FPGA with the QuantumRed VR thermal module demonstrates that our image processing module achieves a throughput of 60 frames per second (FPS) when processing 14-bit 640×480 resolution infrared video acquired from an uncooled IRFPA.**

***Index Terms*—thermal imaging, infrared camera, image processor architecture, non-uniformity correction, uncooled infrared focal plane array**

I. Introduction

Recent advancements in emerging embedded technologies, such as robotics and advanced driver assistance systems, have spurred the demand for thermal imaging and processing based on infrared (IR) sensors due to the benefits, including enhanced night visibility and privacy protection [1], [2].

Uncooled infrared focal plane array (IRFPA) plays a pivotal role in thermal imaging applications, particularly in lightweight embedded systems. Due to the functionality under normal temperature conditions, the uncooled IRFPA enables the implementation of a thermoelectric cooler-less (TEC-less) thermal imaging camera. The inherent compactness of uncooled IRFPA resulting from this attribute boosts them to be adopted in lightweight vision-based systems and applications.

Non-uniformity correction (NUC) has been a fundamental feature in the processing of the thermal image from the IRFPA. Fixed pattern noise (FPN) originating from the manufacturing process and thermal characteristics generated by the dark current noise emanates functional differentiation for each pixel on the same IR radiation intensity. These non-uniformities significantly degrade the overall image quality.

Two-point correction (TPC) is a commonly used algorithm for mitigating FPN due to its robust performance and simplicity, based on a linear expression that is suitable for execution on lightweight embedded platforms [3]. However, despite the benefits, applying the TPC algorithm is avoided in real-time video processing due to intrinsic drawbacks such as the necessity of run-time repetitive flat field correction (FFC), which disturbs the continuous visualization of the image frames [3], [4].

To address these challenges, many studies have been conducted to design and implement NUC algorithms without the need for the FFC. These algorithms are categorized as scene-based NUC and neural network-based NUC (NN-NUC) [3]–[5]. However, as these algorithms rely on real-time image frames as a reference source, the results of the images involve the ghost effect and image blurring. Although some studies mitigated these negative effects using novel approaches, scene-based NUC still involves these drawbacks from innate characteristics [3]. Indeed, the TPC algorithm remains a powerful approach and serves as a baseline for performance evaluation of other NUC algorithms, as the TPC affords robust performance [4]. Nevertheless, reducing interruption caused by repetitive FFC is required to minimize the drawbacks of the TPC.

Another important consideration is the presence of dark current noise arising from thermal drift [6]. In traditional thermal imaging systems, this noise is mitigated by maintaining sensor temperature low through the use of external thermoelectric coolers [7]. However, in compact imaging systems utilizing uncooled IRFPA, a subsidiary method for thermal drift compensation (TDC) is necessary to address the variable characteristics associated with temperature. One of the most commonly used and suitable methods for the TDC is pixel-wise polynomial modeling [8], [9].

Executing pixel-wise processing algorithms, such as the TDC based on polynomial modeling, TPC based on linear expression, and repetitive FFC, in a general-purpose processor (GPP) for real-time applications is inefficient due to memory bottlenecks originating from the necessity of accessing numerous coefficients, which vary for each term and pixel. As a result, the FPGA-based approach is widely adopted in many studies in IR video processing.

Nevertheless, applying these algorithms for real-time high-resolution IR video with a high refresh rate is challenging as the memory size and bandwidth are proportionate to each

2771-2508/23/$31.00 ©2023 IEEE
DOI 10.1109/DSD60849.2023.00095

pixel count and refresh rate. Previous approaches address these issues by either lowering frame rate [10] or utilizing additional random access memory (RAM) [4], [11], [12]. However, these approaches still have crucial drawbacks. The former approach reduces reactivity and the visibility of the video, while the latter compromises the compactness of the system.

On the other hand, contrast enhancement methods (CEM) adaptive to variable data ranges, e.g., min-max contrast stretching and histogram equalization (HE), are suitable to be processed by a GPP with external memory as these methods require a frame buffer with multiple pipelines to analyze the entire image frame by frame and store the entire buffer to the block RAM in FPGA requires exploiting extremely high-cost FPGA. Despite these advantages, exploiting the external processor has drawbacks in terms of compactness and power due to mounting another chip and overheads generated by video signal communication between different chips. Therefore, employing a modern system-on-chip field-programmable gate array (SoC FPGA) for both the NUC and CEM processes is one of the appropriate methods to address these issues.

In this paper, we propose an integrated thermal image processor architecture designed for real-time interfacing and processing of high-resolution thermal video with a high refresh rate from the IRFPA. Our processor acquires real-time IR video from the uncooled IRFPA, performs NUC algorithms and CEM, and generates the video through transition minimized differential signaling (TMDS) to visualize the video directly with a high-definition media interface (HDMI).

Organized to co-processing architecture utilizing modern lightweight SoC FPGA, the image processing workload is split as operating the NUC algorithms at the programmable logic (PL) block and CEM at the processing system (PS) block. Additionally, we adopted the architecture that PS and PL share the DDR3 RAM to exclude extra memories, preserving the compactness of the entire system.

The internal NUC algorithm comprises polynomial modeling-based TDC, TPC, run-time FFC, and dead-pixel correction (DPC). To address the memory bottlenecks of pixel-wise NUC algorithms based on multi-parameter polynomial expressions, we designed accelerators and parallel caching modules. Moreover, we designed a specialized accelerator to minimize the interrupt time for the run-time FFC.

The proposed architecture was implemented using a low-cost Xilinx Zynq-7000 SoC FPGA and a thermal imaging module, the Hanwha Systems QuantumRed VR, which supports the path-through of raw image frames from uncooled IRFPA. The implementation results demonstrate the successful real-time processing of 14-bit 640×480 resolution IR video at 60 frames per second (FPS) while performing the NUC and the scene-based CEM.

II. Non-uniformity Correction Algorithm

Fig. 1 shows the comprehensive process of the NUC algorithm. The NUC algorithm begins by receiving the raw video signals and the FPA temperature (T_{FPA}) from the IR sensor.

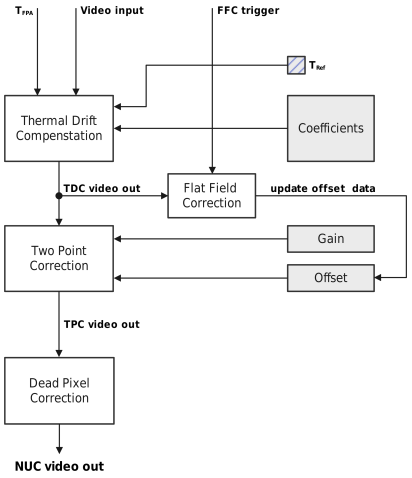


Fig. 1. The non-uniformity correction process in the proposed architecture

The first step is the TDC, which eliminates the FPA temperature response from the raw video signal. This adjustment ensures that the intensity of a pixel, regardless of varying sensor temperatures, exhibits a certain level of similarity in response to the same infrared radiation. Subsequently, the TPC is applied to remove the FPN in the video output of the TDC stage, equalizing the intensity response of all pixels at the same radiation level. Finally, the dead pixels on the output are removed by the DPC by exploiting both pre-measured dead maps and real-time thresholding techniques. To compensate for the imperfections in noise characteristic modeling, the TPC requires run-time FFC, which updates the offset value using the shutter as a reference source during operation. This provides effective compensation for imperfect modeling.

A. Thermal Drift Compensation

The TDC process is based on polynomial fitting for each pixel that regards the variable as T_{FPA} and the response as pixel intensity. As a baseline, axis translation is applied to T_{FPA} through subtraction with reference temperature (T_{Ref}) in both the coefficient derivation and run-time TDC, as shown in (1). This reduces the dynamic range of the coefficients, lowering the quantization error innate to digital arithmetic.

$$T_d = T_{FPA} - T_{Ref} \tag{1}$$

During run-time, the thermal drift response is mitigated by executing two procedures for each pixel: calculating the approximate thermal response ($P_{TDC}(i,j)$) using polynomial expression as shown in (2) and subtracting the calculated response as shown in (3). The coefficients for approximation are pre-calculated using calibration data measured on varying T_{FPA} with the same black body temperature.

$$P_{TDC}(i,j) = \sum_{n=0}^{k} C_n(i,j) \cdot T_d^{k-n} \tag{2}$$

$$I_{TDC}(i,j) = I_{RAW}(i,j) - P_{TDC}(i,j) \tag{3}$$

B. Two-Point Correction

The TPC process is performed by calculating linear expression (4) for each pixel to transform the modeled response function into the reference linear expression.

$$I_{TPC}(i,j) = G(i,j) \cdot I_{TDC}(i,j) + O(i,j) \tag{4}$$

Similar to the TDC, the gain ($G(i,j)$) and offset ($O(i,j)$) are pre-calculated using calibration data, but different to those of the TDC, the calibration data for TPC is obtained under conditions where the T_{FPA} is maintained and the black body temperature varies for two points, cold and hot. The TDC pre-processes these measured data before deriving the TPC coefficients to ensure the input image is identical to the run-time NUC process structure (see Fig. 1).

To derive $G(i,j)$ and $O(i,j)$, first, the mean intensity values (I'_C, I'_H) are calculated with pre-processed data ($I_C(i,j)$, $I_H(i,j)$) according to the (5). The w and h refer to the width and height of the image.

$$I' = \sum_{i=0}^{h-1} \sum_{j=0}^{w-1} \frac{I(i,j)}{wh} \tag{5}$$

The next step is to determine the transformation function for each pixel. The formula to derive $G(i,j)$ and $O(i,j)$ are expressed as follows.

$$G(i,j) = \frac{I'_H - I'_C}{I_H(i,j) - I_C(i,j)} \tag{6}$$

$$O(i,j) = \frac{I_H(i,j)I'_C - I'_H I_C(i,j)}{I_H(i,j) - I_C(i,j)} \tag{7}$$

C. Flat Field Correction

The FFC adjusts the $O(i,j)$ value for each pixel to equalize the response for the varying input data received from the flat reference source ($I_{FFC}(i,j)$) to the same as the average value of the data (I'_{FFC}), gathered by closing and calibrating the IR intensity of the shutter. According to the TPC algorithm, each pixel has a different linear response to $I_{TDC}(i,j)$ as the $G(i,j)$ and $O(i,j)$ differ by pixel position. The $G(i,j)$ data are fixed to match the response with only one reference source, and only the $O(i,j)$ should be changed. This process starts with deriving a new linear algebra with the same $G(i,j)$ that passes through the reference coordinate $(I_{FFC}(i,j), I'_{FFC})$ as shown in (8).

$$I_{TPC}(i,j) = G(i,j) \cdot (I_{TDC}(i,j) - I_{FFC}(i,j)) + I'_{FFC} \tag{8}$$

Using this equation, the new offset value is calculated through the following equation.

$$O_{new}(i,j) = G(i,j) \cdot -I_{FFC}(i,j) + I'_{FFC} \tag{9}$$

Due to the closed housing of the camera, thermal noise, which is the most influential type of noise, gradually reaches a saturation level over time. As the TDC coefficient calculation uses the reference data acquired from the atmosphere in that thermal noise is saturated, the required frequency of the FFC to occur in run-time gradually decreases over time.

Algorithm 1 Dead pixel correction pseudo code

Input:
I: 2-dimensional array of a single frame
D: 2-dimensional array indicating dead pixels
T: Threshold value
Output:
O: 2-dimensional array of the output frame

```
for i ← 0; i < h; i ← i + 1 do
    for j ← 0; j < w; j ← j + 1 do
        W ← get_around_pixels(I, i, j)
        W ← remove_dead_pixels(W, D, i, j)
        m ← mean(W)
        if D[i][j] == true or |m − I[i][j]| > T then
            O[i][j] ← median(W)
        else
            O[i][j] ← I[i][j]
        end if
    end for
end for
```

D. Dead Pixel Correction

The DPC method applied in our processor is based on a conditional median filter combined with multiple methods. The methods involve the utilization of a static dead pixel map extracted during the parameter derivation process and dynamic thresholding of the difference between the local mean of neighboring pixels and the value of the current pixel during run-time. The detailed procedure of the DPC process is presented in algorithm 1.

For pixels that are not identified as dead according to the dead pixel map, the algorithm extracts the surrounding pixels using a predetermined window size while excluding pixels identified as dead in the map. By comparing the average value of the extracted pixels with the value of the current pixel, the algorithm determines whether the pixel is dead based on whether the gap between values exceeds the threshold. If the current pixel is determined to be dead, the output value is replaced with the median value of the extracted pixels. For pixels identified as dead in the dead pixel map, the replacement process is executed without considering any other conditions.

III. Proposed Architecture

Fig. 2 illustrates the architecture of the proposed image processor. The PL block comprises the accelerators for the NUC algorithms (TDC, TPC-FFC, and DPC), the subsidiary accelerator to assist the CEM process executed by the PS block (CEM assistant), the video direct memory access (VDMA) for buffering the video frames in the advanced extensible interface (AXI) memory space and transmitting the processed frames, and the video scaler as well as RGB to HDMI module for visualizing the processed frames. The accelerators, VDMA,

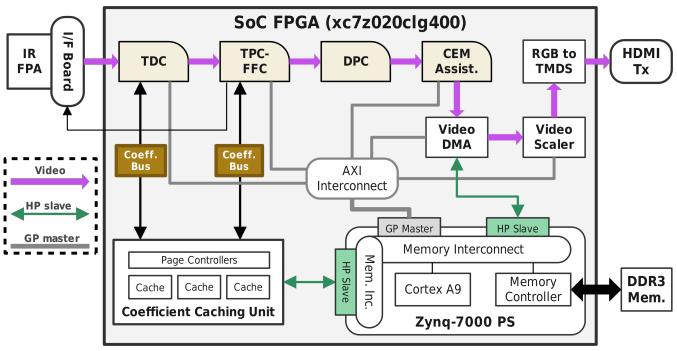


Fig. 2. Top architecture of the proposed thermal image processor.

and video scaler module are connected to the general-purpose (GP) master bus in the PS through the AXI interconnect, enabling the PS to manipulate the entire processing chain.

The PL accelerators are daisy-chained to process the image frame in a horizontal raster scan order. Through this, the memory requirements for the frame buffer for each NUC process are annihilated. Nevertheless, parallel memory accesses for receiving numerous coefficients are still necessary to receive multiple coefficients as the algorithms have varying coefficients for each pixel. As mentioned before, mounting and using an additional memory IC violates the compactness of the entire system. Thus, we designed the coefficient caching unit to exclude the IC addition by sharing the DRAM with the PS, exploiting the AXI high-performance (HP) slave port on PS.

The entire operation of the image processor for one frame is as follows:

- Receive the parallel IR video stream in a raster scan order and obtain temperature from the IRFPA with the I/F board.
- Execute the pipelined NUC accelerators, TDC, TPC, and DPC, and the CEM assistant.
- Store the frame to the DRAM, which organizes a part of the address space, using the VDMA.
- Perform contrast enhancement using the software running on the core of the PS block.
- Visualize the processed frame using the VDMA, video scaler, and RGB to TMDS module.

The FFC is sporadically executed through the TPC-FFC accelerator when triggered by the software. This accelerator reduces the interruption while providing high-quality video, minimizing the time required for calculation.

A. *Coefficient Caching Unit*

Fig. 3 shows the architecture of the coefficient caching unit (CCU), while Fig. 4 depicts the timing of the CCU's caching operations. The CCU includes a virtual read bus and a virtual write bus to facilitate both read and write access for the NUC algorithms. These virtual buses abstract the direct memory access (DMA) operations based on AXI stream read/write accesses, enabling the memory accesses to operate as those for simple RAM that has a latency of one clock cycle for both operations.

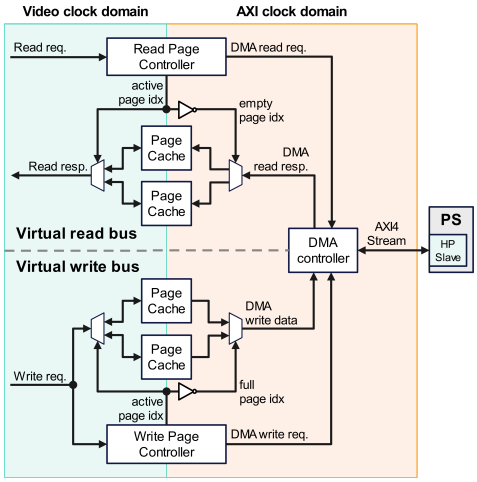


Fig. 3. The architecture of the CCU.

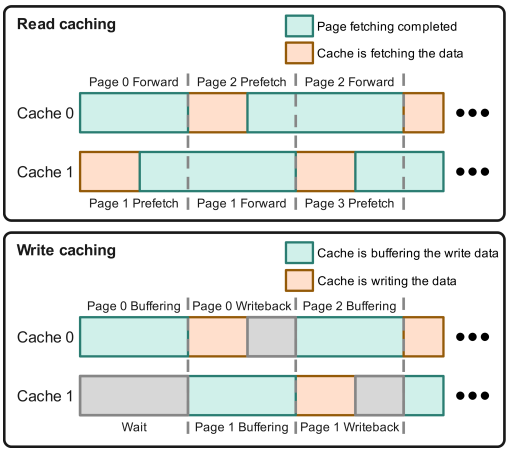


Fig. 4. Timing characteristic of the caching operations on sequential access.

Since the throughput of the DRAM in burst access is higher than the required throughput performance, which is proportional to the video clock period, the virtual buses can be abstracted with suitable prefetching and buffering mechanisms. Each virtual bus contains a pair of page caches that operate crossly and simultaneously for both read and write operations.

In the case of the read bus, the page controller manages one cache to forward the data corresponding to the read request on the video clock domain, while sending the read requests to the DMA controller and manipulating another cache to fetch the data from the DMA read responses. On the other hand, for the write bus, the page controller receives data from the video clock domain and saves the data to one cache while sending the request to the DMA controller to write the buffered data from another cache.

For both virtual buses, the exchange of operations for caches occurs when the page change triggers are set. This architecture is well-suited for video processing with varying coefficients for pixel-by-pixel operations because the memory is accessed sequentially in a raster scan order due to the signal transmission.

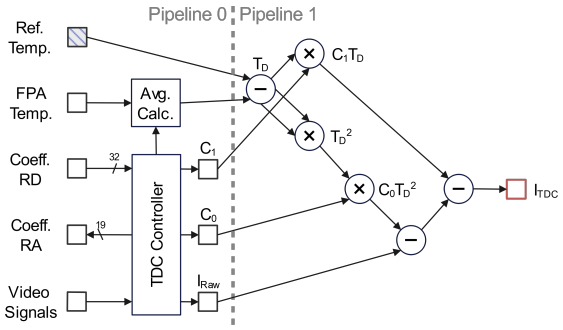


Fig. 5. The architecture of the TDC accelerator.

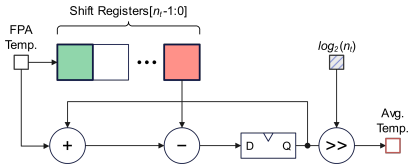


Fig. 6. Average temperature calculator circuit.

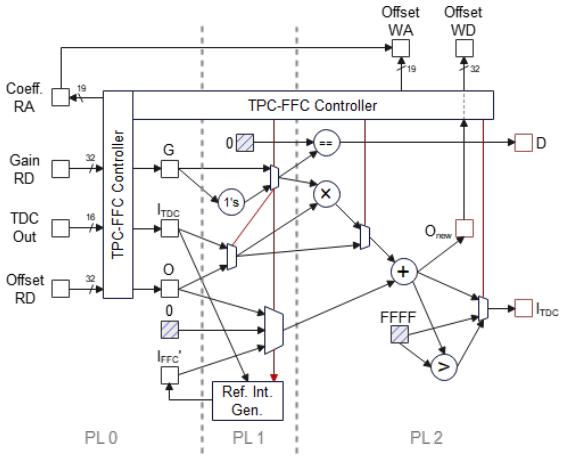


Fig. 7. The architecture of the TPC-FFC accelerator.

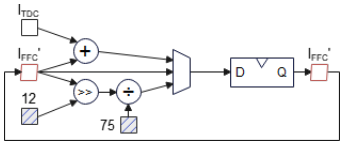


Fig. 8. Reference intensity generator circuit.

B. TDC Accelerator

In our processor, the TDC process applied second-order polynomial fitting. The internal calculations of the TDC accelerator employ adaptable fixed-point arithmetic to accommodate varying quantization levels, with a 14-bit quantization for raw input and 16 bits allocated for each coefficient. As the last coefficient of the TDC process can be combined with the TPC offset, the adder for the last component is omitted, reducing the required number of memory buses.

Fig. 5 depicts the architecture of the accelerator, consisting of two pipeline stages. In the first stage, the accelerator buffers the input video signals, e.g., data, horizontal sync (HSYNC), and vertical sync (VSYNC), generates the address to read, and propagates the coefficients of the current pixel position, which are bundled into the 32-bit data read, through the bus transaction. The read data is then split into two 16-bit coefficients. Additionally, the average value of the FPA temperature is derived through the average calculator (see Fig. 6 to be used in the next pipeline. This reduces the uncertain noise of the temperature data. In the next stage, the polynomial expression is calculated, with ancillary video signals such as HSYNC and VSYNC also propagated in a pipelined manner.

C. TPC-FFC Accelerator

The TPC-FFC accelerator performs both the TPC process and the FFC process. As both the FFC and TPC computation requires the gain ($G(i,j)$) value for each pixel, combining the architecture for TPC and FFC minimizes the memory bandwidth requirements efficiently. Fig. 7 shows the architecture of the TPC-FFC accelerator. During normal operation, the accelerator reads the gain ($G(i,j)$) and offset ($O(i,j)$) from the two read buses and calculates the TPC output intensity ($I_{TPC}(i,j)$). When the FFC trigger occurs by the processor, the accelerator performs the FFC process by calculating the required equation (see (9)) while exploiting only the memory space for the $O(i,j)$. The following sequence accomplishes this process:

- Accumulate a certain number of frames for each pixel and for all pixels to generate I_{FFC} and I'_{FFC}, respectively.
- Get I'_{FFC} by dividing by the pixel count, which is 307200 in 640×480 resolution. This is substituted to executing the right shift by 12-bit as well as the division by 75.
- Derive the new offset ($O_{new}(i,j)$) and save through the write bus.

The I'_{FFC} is calculated through the generator circuit (see Fig. 8) manipulated by the TPC-FFC controller.

Fig. 9 shows the timing comparisons of the FFC operations between the accelerator and the software-only implementation. Assuming that the FFC process uses four frames to build a flat reference source, the accelerator only requires five frames to complete the FFC process, meaning that only one frame is required for offset calculation, due to the synchronization of frames and utilization of the virtual buses (see Fig. 3, 4). In contrast, the software-only implementation requires additional time to handle the latency originating from the VDMA. Moreover, the bottleneck from cache misses due to the numerous coefficients degrades the performance. Although the prefetcher on the PS allows the accumulations to be done before the next frame is received, it is not enough to efficiently handle the O_{new} calculation, which requires huge memories compared to the cache size. These characteristics increase the number of discarded frames to be prepared for normal operation, highlighting the drawbacks of the TPC. Therefore, leveraging the FFC accelerator is key to undermining the major weakness of the TPC as the interruption time is minimized.

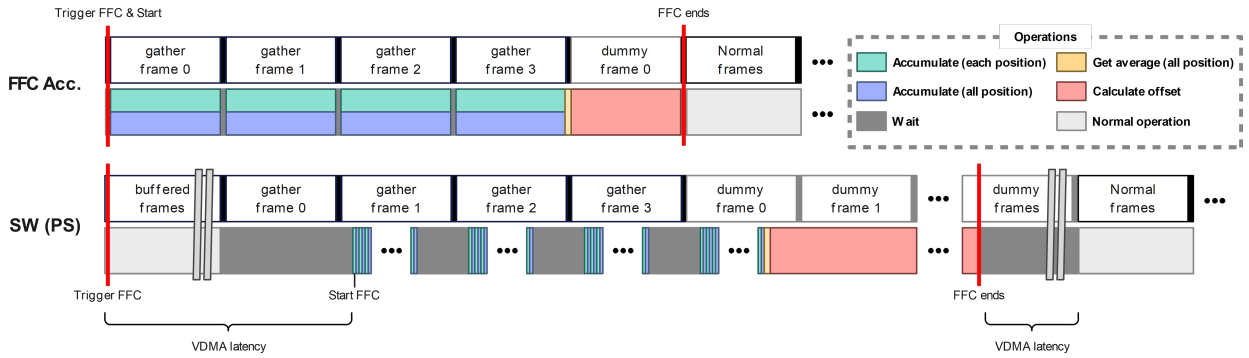


Fig. 9. Timing comparisons between the accelerator-based and PS-only implementation for the FFC operations.

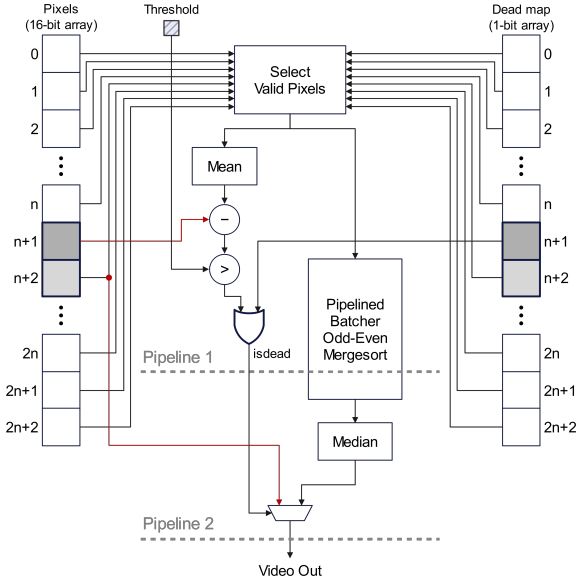


Fig. 10. The architecture of the DPC accelerator.

D. DPC Accelerator

The architecture of the DPC accelerator is depicted in Fig. 10. To process the pixels in a raster scan order, the $2n + 3$ of bundled shift registers, which contain a 16-bit pixel array, 1-bit dead map, and synchronization signals (HSYNC, and VSYNC), are mounted for organizing the 3×3 window for around pixels, assuming the n as a sum of the width and required clock count for HSYNC.

Similar to the previous accelerators, the DPC accelerator is pipelined to satisfy the timing requirements. In the first pipeline, the final dead flag (isdead) is identified by thresholding the gap between the current pixel value and the mean value of around pixels, as well as seeking the dead indicator of the current pixel. The dead pixels around are not extracted, utilizing the dead map. In addition, the first pipeline of the Batcher odd-even mergesort is executed. In the second pipeline, the lasting sorting algorithm is executed and finally, the appropriate video output is selected from the original pixel value and median value of the sorted output through the flag.

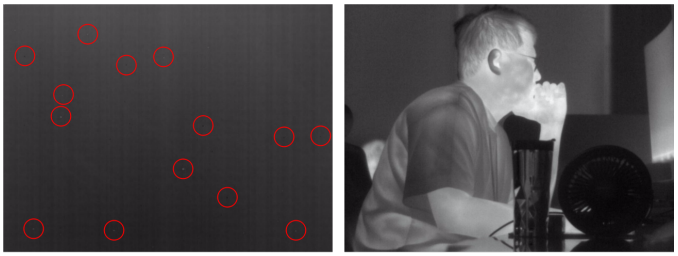
IV. Implementation & Evaluation

To evaluate the proposed image processor, we implemented the processor using the Zybo Z7-20 board with Vivado 2022.2 tool and the custom interface board that receives the IR video, which is 14-bit depth, 640×480 resolution, and 60 FPS, from the Hanwha Systems QuantumRed VR thermal module. This module supports the pass-through of raw image frames from uncooled IRFPA. The calibration for the NUC parameter derivation is performed using only the thermal module. This is because the Z7-20 board is inappropriate to endure a variety of temperature conditions in the thermal chamber as the SoC FPGA mounted on the board is consumer graded.

Fig. 11 presents the raw frame and the frame fully processed by the NUC accelerators implemented on the PL and scene-based CEM running on the PS software. As shown in the figure, the raw image from the uncooled IRFPA cannot be classified and also includes several dead pixels, marked by red circles. In contrast, the output frame presents an image that reaches the human-visible level, making it available for surveillance applications while not showing any negative effects, for example, dead pixels, ghosting effects, and image blurring. Furthermore, the scene-based CEM on the image processor enables visual identification of objects not only in sparse IR emissions but also in dense IR emissions, as the CEM is adaptive to varying intensity ranges for each frame.

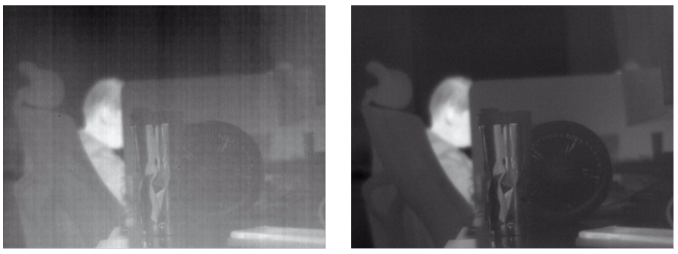
The CEM process we applied is based on adaptive linear min-max stretching, which utilizes the minimum and maximum values within each frame. Though the histogram equalization shows better contrast enhancement, it is hazardous for surveillance due to the alteration of the objects in the image, making the objects with dense intensities hard to identify. Therefore, we adopted the adaptive linear min-max stretching, which preserves the object information.

Fig. 12 visually compares the frame with and without the run-time FFC process, with all other algorithms fully applied. The left image is captured in the condition that untreated noises are accumulated to the visible level. According to the right figure, the image distortion is prominently removed as the FFC process is executed in this state. As a result, the FFC is necessary for the TPC-based NUC algorithms because such distortion is a crucial disadvantage in terms of image quality.



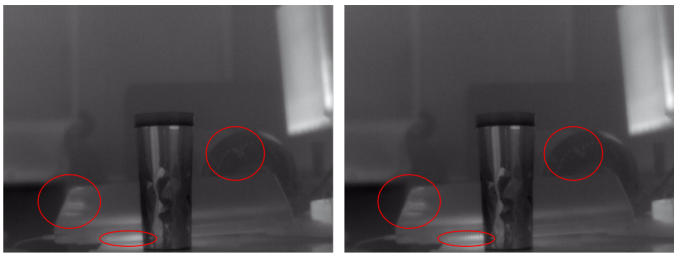
(a) Raw Frame (b) Processed Frame

Fig. 11. Comparison between the raw frame and the fully processed frame. (Red circles indicate the dead pixels)



(a) Before FFC (b) After FFC

Fig. 12. Comparison between the frames with and without the FFC process.



(a) TPC only (b) TDC-TPC

Fig. 13. Comparison between the frames with and without the TDC process. (Red circles indicate the points that are blurred or mismatching intensity)

Fig. 13 presents the frames with and without the TDC process. The frame on the left side, which is processed without the TDC process, exhibits several instances of blurring and intensity mismatch, marked by red circles. Compared to the frame on the left side, the fully processed frame on the right side shows a clear output. Though the image quality improvement is not as impacting as the other processes, applying the TDC is reasonable as it relieves the temperature characteristics, reducing the frequency of the occurrence of FFC triggers for maintaining the image quality.

To evaluate our NUC algorithm, we set the quantitative metric as the roughness index (ρ) presented in the (10). This metric is the ratio of the edge component relative to the original image and derived through the edge mask h [13], which refers to the horizontal vector $(1, -1)$. The lower ρ value indicates better performance as it reflects the variations in the image, which are amplified when non-uniformity present in the image increases.

TABLE I
ROUGHNESS INDEX ANALYSIS

Frame type	ρ	
	$20°C$	$80°C$
Raw	5.663×10^{-2}	4.313×10^{-2}
Processed	4.063×10^{-3}	3.271×10^{-3}

TABLE II
RMSE ANALYSIS

Frame type	RMSE	
	$20°C$	$80°C$
Raw	6.555×10^{-2}	6.298×10^{-2}
Processed	3.322×10^{-3}	3.332×10^{-3}

$$\rho(I) = \frac{||h * I||_1 + ||h^T * I||_1}{||I||_1} \tag{10}$$

We compared the ρ of the raw and the processed frames gathered from the black body with a temperature of 20 and 80 ^oC. Table I presents the average roughness indexes for both frames. The result presents that the NUC algorithm achieved 92.83% and 92.41% reduced roughness index at 20 and 80 $°C$, respectively.

For further evaluation, we set another metric as the normalized root mean square error (RMSE) derived from the frames gathered from the black body, previously used in roughness index analysis. The lower value indicates better performance.

$$RMSE(I, I_{ref}) = \frac{1}{2^p - 1}\sqrt{\frac{1}{wh} \cdot \sum_{i=0}^{h-1}\sum_{j=0}^{v-1}(I(i,j) - I_{ref})^2} \tag{11}$$

The p refers to bit-width of the data, which is 14 for raw frames and 16 for processed frames. The reference value (I_{ref}) is the mean value of the raw frames and processed frames, respectively, as every pixel value is depicted as the same value in ideal condition. The result in Table II presents that the NUC algorithm achieved 94.9% and 94.7% reduced RMSE at 20 and 80 $°C$, respectively. Both the roughness index and RMSE analysis confirms that the TPC-based NUC we adopted is effective to address the non-uniformity of raw infrared video.

In the case of memory bandwidth, the required throughput values of read/write operations are measured as 230.31 and 76.77 megabytes per second (MB/s), respectively, according to the video clock of the uncooled IRFPA, which is 19.19 MHz. These results are significantly lower than the maximum achievable throughputs using the AXI HP slave port, measured as 365.6 MB/s and 448.1 MB/s for one port according to [14]. The timing reports present that the processor can be operated on a maximum of 57.69 MHz frequency. This result is above 45.7 MHz, the condition that the maximum bandwidth of the AXI HP slave port is reached when utilizing the two ports for coefficients while other ports are used for the VDMA. Therefore, the PL part of our processor can handle the 640×480 video up to 142.87 Hz without a design change. However, the PS should be further verified for a higher FPS.

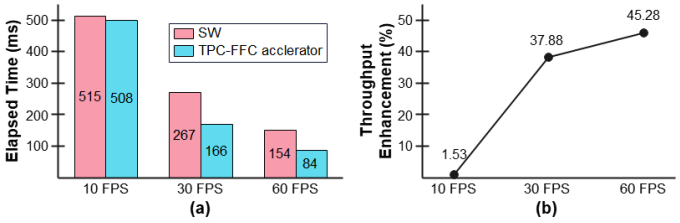


Fig. 14. Performance breakdown on the FFC processes with the SW and HW. (**a**) Elapsed time. (**b**) Throughput enhancement.

TABLE III
RESOURCE UTILIZATION OF THE ENTIRE IMAGE PROCESSOR

Element Type	Used	Available	Percentage
LUTS	18,690	53,200	35.13%
Flip-Flops	25,122	106,400	23.61%
BRAM	28.5	140	20.36%
DSP blocks	44	220	20.00%

To evaluate the designed hardware (HW) accelerators, we compared the maximum reachable FPS of both HW acceleration and software implementation of the NUC algorithms, composed of the TDC, TPC, and DPC processes. The required time for processing one frame using the software is measured as approximately 17.54ms on average, showing that the software NUC can process up to 57.02 FPS. This result indicates that utilizing hardware accelerators can achieve up to 2.49 times the throughput compared to the software NUC. Consequently, exploiting the designed hardware accelerator is a suitable method to overcome the performance limitations of the PS and distribute the workload of the PS to allow more complex CEM to be executed in the PS.

We measured the elapsed time of the FFC processes, which utilize four frames, of both hardware (HW) acceleration and software implementation to evaluate the FFC acceleration. The TPC-FFC accelerator achieved 1.53%, 37.88%, and 45.28% improved throughput compared to the SW on 10, 30, and 60 FPS, respectively (see Fig. 14). These results originate from the characteristics of the accelerator, which are synchronized to the video clock domain. The improvement is expected to be higher where that video has higher resolution or FPS as the performance of the accelerator becomes faster while that of the PS is limited by the fixed operating frequency.

Table III presents the resource utilization of the proposed image processor. The result shows that our image processor occupies less than 40% of each element type in the Xilinx XC7Z020CLG400, proving that the proposed architecture is feasible on a low-cost SoC FPGA. Moreover, post-processing logic such as color adjustment, sharpening, and object detection can be added to the processor for further enhancements.

The overall implementation results suggest that with a high-performance SoC FPGA and DDR4 memory along with certain optimizations such as parallelism and splitting pipelines, better throughput performance, including resolution and frame rate, is expected to be achieved while preserving the NUC algorithm applied in this work.

TABLE IV
POWER BREAKDOWNS OF THE ENTIRE IMAGE PROCESSOR

Type	Max Power (mW)	Share
Total	2,098	100%
PS block	1,417	67.54%
PL	436	20.78%
IO	132	6.29%
Device static	113	5.39%

Table IV shows the power breakdowns of the entire image processor. The overall maximum power consumption is acceptable for low-power devices. In addition, the PL part, including logic, BRAM, DSP, and PLL, consumes only 20.78% of the overall power. This indicates that utilizing the proposed accelerators is more efficient than exploiting only the GPP.

Table V shows the comparisons between our work and the prior works based on a variety of NUC algorithms. In the case of throughput, which is proportionate to the resolution and FPS, our processor achieved 18.43 and 43.89 megapixels per second (MP/S) with and without adaptive CEM, respectively, showing the best performance compared to other works except for the work in [4]. Compared to the work that adopted TPC-based NUC algorithms [10], our work achieved approximately 49.96% higher throughput. This result indicates that our processor architecture with multiple caching units effectively deal with the memory bottleneck of pixel-wise algorithms. However, our work still has throughput gap compared to the work with scene-based NUC.

In the case of power, our processor shows a higher consumption than the work with scene-based NUC. Considering the lower FPS of our work, this gap will be more on the same condition. This originates from the utilization of the PS that dissipates most of the power in our design. Despite the drawback in terms of power, the proposed architecture still has advantages in executing GPP-friendly high computational operations such as adaptive CEM.

V. CONCLUSION

This paper presents the integrated image processor architecture for processing real-time high-resolution thermal video from uncooled IRFPA utilizing a low-cost SoC FPGA. Our processor provides a one-chip solution for the NUC algorithms and the scene-based CEM to be executed seamlessly. The technical barriers and drawbacks that obstacles the calibration-based TPC and the polynomial modeling-based TDC from being applied to thermal image processing were addressed by several techniques, including exploiting the specialized caching units, TPC-FFC accelerator, and other NUC accelerators. The implementation on a low-cost XC7Z020CLG400 SoC FPGA with QuantumRed VR thermal module demonstrates that our processor achieved a throughput of 60 FPS on 14-bit 640×480 resolution infrared video received from uncooled IRFPA while using less than 40% of the SoC FPGA and consuming 2098 mW. We expect our architecture to be applied to a variety of thermal imaging applications that requires robust image quality and minimized interruption.

TABLE V
COMPARISONS BETWEEN THE PRIOR WORKS

	This work	[3]	[4]	[10][b]
NUC Type	TPC-based NUC	NN-NUC	Scene-based NUC	TPC-based NUC
NUC Algorithms	TDC + TPC + FFC + DPC	Scribner's + Guided Filter	Constant Range	TPC + DPC
Resolution	640×480	256×256	640×480	640×512, 64×48
FPS	60 (142.87[a])	180	238	25, 4000
Throughput (MP/s)	18.43 (43.89[a])	11.80	73.11	8.19, 12.29
Power (mW)	2098[c]	Not presented	1184	Not presented

[a]Only the NUC accelerators are confirmed for operation.
[b]Supports dual mode configuration.
[c]Reported for 60 FPS configuration.

In future work, we plan to develop our work in three different directions. The first direction is applying the complex CEM, e.g., contrast limited adaptive histogram equalization, utilizing the proposed image processor to improve overall image quality while preserving the throughput. Another direction is upgrading the proposed image processor architecture for the IRFPA with a higher resolution over the high definition resolution or a higher frame rate. This objective is regarded as can be accomplished by utilizing the DRAM with better performance and certain modifications on the accelerator architecture. Finally, we will design the integrated module, which directly receives the video signals from the uncooled IRFPA, different from the current implementation that uses pass-through features of the thermal module. This subject consists of changing the SoC FPGA to satisfy the industrial grade and designing the specialized printed circuit board.

REFERENCES

[1] P. Hurney, P. Waldron, F. Morgan, E. Jones, and M. Glavin, "Review of pedestrian detection techniques in automotive far-infrared video," *IET Intelligent Transport Systems*, vol. 9, no. 8, pp. 824–832, Oct. 2015. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1049/iet-its.2014.0236

[2] N. Kowalczyk, M. Sobotka, and J. Rumiński, "Mask Detection and Classification in Thermal Face Images," *IEEE Access*, vol. 11, pp. 43 349–43 359, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10113628/

[3] S. Rong, H. Zhou, Z. Wen, H. Qin, K. Qian, and K. Cheng, "An improved non-uniformity correction algorithm and its hardware implementation on FPGA," *Infrared Physics & Technology*, vol. 85, pp. 410–420, Sep. 2017. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S135044951630696X

[4] R. Redlich, M. Figueroa, S. N. Torres, and J. E. Pezoa, "Embedded nonuniformity correction in infrared focal plane arrays using the Constant Range algorithm," *Infrared Physics & Technology*, vol. 69, pp. 164–173, Mar. 2015. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1350449515000389

[5] N. Celedon, R. Redlich, and M. Figueroa, "FPGA-based Neural Network for Nonuniformity Correction on Infrared Focal Plane Arrays," in *2012 15th Euromicro Conference on Digital System Design*. Cesme, Izmir, Turkey: IEEE, Sep. 2012, pp. 193–200. [Online]. Available: http://ieeexplore.ieee.org/document/6386892/

[6] H. Du, Z. Gong, D. Li, Y. Wang, Y. Zhao, J. Xu, and D. Sun, "Dark Current Measurement and Noise Correction Method for LWIR QWIP Detection System Based on Focal-Plane Temperature," *Applied Sciences*, vol. 13, no. 9, p. 5549, Apr. 2023. [Online]. Available: https://www.mdpi.com/2076-3417/13/9/5549

[7] J. Nazemi, J. Battaglia, R. Brubaker, M. Delamere, and C. Martin, "A low-power, TEC-less, 1280 x 1024, compact SWIR camera with temperature-dependent, non-uniformity corrections," B. F. Andresen, G. F. Fulop, and P. R. Norton, Eds., Baltimore, Maryland, May 2012, pp. 83 530B–83 530B–10. [Online]. Available: http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=1353401

[8] D. Lin, X. Cui, Y. Wang, B. Yang, and P. Tian, "Pixel-wise radiometric calibration approach for infrared focal plane arrays using multivariate polynomial correction," *Infrared Physics & Technology*, vol. 123, p. 104110, Jun. 2022. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1350449522000913

[9] O. Gonzalez-Chavez, D. Cardenas-Garcia, S. Karaman, M. Lizarraga, and J. Salas, "Radiometric Calibration of Digital Counts of Infrared Thermal Cameras," *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 11, pp. 4387–4399, Nov. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8611149/

[10] Q. Liu, H. Wang, H. Bian, and H. Wang, "Dual Mode High Speed Uncooled Short Wave Infrared Camera Based on FPGA," *Journal of Physics: Conference Series*, vol. 1952, no. 3, p. 032042, Jun. 2021. [Online]. Available: https://iopscience.iop.org/article/10.1088/1742-6596/1952/3/032042

[11] A. Kumar, S. Sarkar, and R. Agarwal, "A novel algorithm and FPGA based adaptable architecture for correcting sensor non-uniformities in infrared system," *Microprocessors and Microsystems*, vol. 31, no. 6, pp. 402–407, Sep. 2007. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0141933107000373

[12] T. Sosnowski, G. Bieszczad, M. Kastek, and H. Madura, "Processing of the image from infrared focal plane array using FPGA-based system," in *Proceedings of the 17th International Conference Mixed Design of Integrated Circuits and Systems*, 2010, pp. 581–596. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/5551280

[13] M. M. Hayat, S. N. Torres, E. Armstrong, S. C. Cain, and B. Yasuda, "Statistical algorithm for nonuniformity correction in focal-plane arrays," *Applied Optics*, vol. 38, no. 5, p. 772, Feb. 1999. [Online]. Available: https://opg.optica.org/abstract.cfm?URI=ao-38-5-772

[14] Xilinx, "Evaluating high-performance ports," 2022, Accessed: 04-Apr.-2023. [Online]. Available: https://xilinx.github.io/Embedded-Design-Tutorials/docs/2021.1/build/html/docs/User_Guides/SPA-UG/docs/6-evaluating-high-performance-ports.html