Reply Letter

Masahiro Nishimura, Taito Manabe, and Yuichiro Shibata

We deeply appreciate your careful review and valuable suggestions. We have carefully revised the paper according to the reviewers’ comments as follows. In the revised manuscript, corrections were made to address the comments and additional typo corrections were made. In the revised manuscript, corrections corresponding to the comments are indicated in red. (Figure is corrected with caption in red.)

Reply to Associate Editor

> Thank you for your submission to IEICE Transactions. Both of our reviewers think that the paper is interesting and could be published after the revision.
> ”Mandatory:” parts by two reviewers are the mandatory conditions of the conditional acceptance, and other parts are optional. Please revise based on their comments.

Thank you very much for your email and for the opportunity to revise our manuscript submitted to IEICE Transactions. We sincerely appreciate the reviewers’ constructive and encouraging comments. We are pleased to know that both reviewers found our paper interesting and potentially publishable after revision. We will carefully address all the “Mandatory” points raised by both reviewers, as well as consider the optional suggestions to further improve the quality of the manuscript.

Reply to Reviewer A

> This manuscript proposes an estimation method of FPGA’s HW resource for real-time HDR synthesis. The method estimates the required BRAM resource usage with reasonable accuracy (around 10% error) and the authors demonstrate that it is useful for the design space exploration under the tradeoff between HW resources and image quality.
> The manuscript is written in good organization and good English. However, there are several points that need to be updated.

Mandatory:
Section 3.3 in p.3 says ”the compression is used only after the fifth layer”. I think it means that, for instance, the 7-layer configuration implements a compression circuit only for 6th and 7th layers and not for 1st to 5th layers. (i.e. partially compressed and partially not-compressed in the implementation) However, Section 4.2 in p.5 says ”in the modeling, ... compression is performed in all the layers ... to simplify the equation”. Why the model did not estimate layers before and after the fifth layer separately? It seems not so difficult to separately estimate the resources.

As you correctly pointed out, the actual hardware implementation applies compression only to the layers after the fifth one, resulting in a partially compressed architecture. However, in Sec. 4.2, we chose to simplify the resource estimation model by assuming that all layers are compressed, in order to keep the equation tractable and easy to use for design space exploration. To clarify this point and show that we have indeed considered the layer-wise breakdown, we have added a detailed derivation in the Appendix, where both compressed and uncompressed parts are modeled separately. This includes the resource usage for each layer and explains how the simplified model was derived based on assumptions about dominant contributors and trade-offs between accuracy and model complexity.

Mandatory:
In p.5, equation (9) - (12) contains many ”magic numbers” such as 7, 3, 67, 4, 11, and so on. Since this part (the estimation method) would be the key of the proposed method, a detailed explanation is required. If the rationale for the numbers in the formulae is not provided, it is difficult to determine whether they are correct or not.

We have added a comprehensive derivation of Eq. (9) - (12) in the Appendix, where each constant (e.g., 7, 3, 67, 4, 11, etc.) is explained based on the buffer sizes, bit widths, pipeline delays, and module structures described in Sec. 3. The derivation is shown step by step using the same variables and architecture diagrams to provide a transparent and reproducible basis for the estimation model. We believe this addition enhances the credibility and clarity of the proposed method.

Mandatory:
Are W, G, and L in equation (9) the BRAM usage for all layers or single layer? (Since ’d’ is the number of processing layers, it implies the equation is for single layer, right?)

Eq. (9) represents the total BRAM usage across all layers. The variable d denotes the number of layers, and the formula incorporates the cumulative usage of modules W, G, and L over the entire pipeline. We agree that it may seem as though the equation is for a single layer due to the appearance of d, and we have revised the manuscript to clarify this point. While it is possible to formulate BRAM usage per layer, the resulting expression becomes considerably more complex. To address this concern, we have added a detailed derivation of Eq. (9) - (12) in the Appendix, showing how the total BRAM usage is computed from per-layer contributions.

Mandatory:
* Typos and confusing sentences

1. What is the definition of 'n' in equation (9) - (12)? Is it the number of input images?
2. Explanation of Fig. 7 is expected. The manuscript says "See [10] for detailed algorithm and implementation" in p.3. However, since you have included the figure, it is worth explaining a bit more. According to [10], it is ADPCM. Then, at least, the word "ADPCM" should be in the manuscript. (P.3 says "adaptive differential coding". Is it different from ADPCM?)
3. The x-axis of Fig. 8 and Fig. 9 is "Depth". Is it "# of layers"?
4. Section 5.3 in p.6: "is requires 952.7" -> "requires 952.7"?
5. Section 5.3 in p.6: "The graph also shows that the increase in BRAM ..." -> Which graph? Do you mean the table (Table 1)?
6. Section 5.4 in p.7: "we can easily find that the 6-layer and 7-input configuration achieves the best quality..." -> I think 5-input is better quality (MEF-SSIMd) than 7-input for the series of 6-layer (orange bars).

We are grateful for your careful review and showing typos or other problems. For each of the listed items, we have revised the manuscript as follows:

1. n is the number of pieces of input, and the definition of n has been added to the description of Eq. (9)
2. We have added a brief explanation of Fig. 7 to clarify the function and structure of the module.
3. The x-axis in Fig. 9 and Fig. 10 is the number of layers. The figure has been corrected to replace the figure.
4. The word "is requires 952.7" has been replaced with "requires 952.7."
5. It means Table1 and Table2, not graph. The word "The graph ..." has been replaced with " Table 1 and Table 2 ⋯ ."
6. The correct configuration is "7-layer and 6-input".

Optional:
In p.2, right-column: "Mertens' algorithm ... generates a Gaussian pyramid of the input image and a Laplacian pyramid of the weights." However, the previous paragraph says "The Laplacian pyramid can be regarded as a decomposition of an image ..." Are they consistent? The caption of Fig. 3 in [1] also says "Laplacian decomposition of the images and a Gaussian pyramid of the weight maps". [10] also says "a Laplacian pyramid of the input image and a Gaussian pyramid of the weights".

The explanation previously provided was incorrect. The correct description should be: "Mertens' algorithm ... generates a Laplacian pyramid of the input image and a Gaussian pyramid of the weights."

Section 5.2 in p.5: "Resource usage ... is also shown in Fig. 9 as in Fig. 8" -> I think "as in Fig. 8" is not needed.

The redundant "as in Fig. 8." has been removed.

Section 5.3 in p.6 says ”Since the difference between ... 912 and 1024 is relatively larger than I am also interested in the resource usage of DSP in FPGA. Are DSPs not critical in the architecture?

In our implementation[10,11], DSP is not very important as we rarely use it, and BRAM is the most dominant.

Section 5.4 in p.7: Fig. 10 uses ”translucent bars”, however, it is difficult to distinguish. Please try to improve (ex. adding a dotted-line border to the translucent bars).

Added a border to the graph and replaced Fig. 10.

In the list of references, the author list of [10] is all capital (e.g. M. NISHIMURA). Please use the same style as other references (ex. M. Nishimura). ”adpcm”, ”hrd” and ”fpga” in the title can be capital letters

The inconsistent capitalization has been corrected.

Reply to Reviewer B

This paper proposes a method to accurately estimate BRAM usage in HDR compositing on FPGAs using a mathematical model, and is a practical result that contributes greatly to the efficiency of design space exploration. The approach that enables the evaluation of resource consumption for each configuration in advance is solid and useful, and can be evaluated as having engineering significance. On the other hand, the circuit configuration itself is a common configuration in image processing, and its novelty is limited.

The reviewer requests the following corrections. First, there is no quantitative evaluation of the throughput and latency of this implementation. To claim real-time performance, numerical indicators such as processing time and frame rate are essential. Second, the increase in latency is cited as the reason for avoiding DRAM, but this is unconvincing due to the lack of quantitative verification. To show the benefits of BRAM design, comparison with when DRAM is used and discussion of bandwidth constraints are necessary. Third, HDR processing is generally performed on CPUs and ASICs, and a simple comparison with these in terms of performance and flexibility is desirable. In light of the above points, we would like to ask you to add experimental data and discussion.

Strengths:

- The proposed BRAM usage model is well formulated, practically useful, and helps significantly reduce design iteration time in FPGA-based HDR processing systems.
- The evaluation results demonstrate that the estimation error is small and predictable, which enhances the model's reliability for design exploration.
- The structure of the manuscript is clear, and figures are well prepared.

Weakness:

- The idea of using modeling to guide hardware configuration under resource constraints is sound and commendable.
- However, the architectural design (fully pipelined Mertens-like HDR synthesis with Gaussian/Laplacian pyramids) follows well-known image processing practices and offers limited novelty.

Mandatory:

1. The paper claims that DRAM usage increases latency and implementation cost. However, there is no measurement or estimation of actual system latency, either for the proposed BRAM-only design or in comparison with DRAM-based approaches. Please add comments on quantitative latency or throughput (e.g., clock frequency, cycles per frame, fps at given resolution) to support the claims of real-time capability and low latency. The authors can refer their past paper to show their system's performance.

Thank you for pointing this out. In response to your comment, we have added a discussion on the system's latency and throughput performance. Specifically, we refer to our previous work [10,11], which presents a DRAM-based implementation of HDR image fusion using a similar algorithm. To clarify and support our claims on low latency and real-time capability of the proposed BRAM-only architecture, we have included the quantitative performance data (clock frequency, frame processing latency, and fps at target resolution) in Table 4 of the revised manuscript. These additions demonstrate the effectiveness of the BRAM-only architecture in reducing latency and maintaining real-time throughput.

> Mandatory:
> 2. There is no justification for excluding DRAM besides a qualitative assertion of ”latency increase.” Please discuss whether DRAM could have been a viable alternative under different scheduling/buffering strategies, and clarify whether memory bandwidth limitations played any role in this design choice. The authors can show the performance difference (bandwidth and latency gaps) between DRAM and BRAM by referring a paper such as the following one.
> H. Peng et al., ”Binary Complex Neural Network Acceleration on FPGA : (Invited Paper),” 2021 IEEE 32nd International Conference on Application-specific Systems, Architectures and Processors (ASAP), NJ, USA, 2021, pp. 85-92, doi: 10.1109/ASAP52443.2021.00021.

In the revised manuscript, we have added a discussion in Sec. 1 to clarify the rationale behind excluding DRAM in our design. Specifically, we now include a comparison of the typical memory bandwidth and latency between DRAM and BRAM, citing relevant references including the work by H. Peng et al. [9]. This comparison helps justify our decision from a hardware performance perspective. Furthermore, we briefly discuss that although DRAM could be used under different scheduling or buffering strategies, such approaches would introduce additional complexity and control overhead, which may compromise the low-latency requirement of our target application. These aspects are now addressed in the revised introduction.

> Mandatory:
> 3. The paper lacks any comparison with CPU or ASIC-based HDR implementations, which are more common in industry. Even a brief discussion in the related work or conclusion section would help contextualize the advantages and trade-offs of the FPGA approach.

we have added a new subsection (Sec. 5.5) to the revised manuscript, where we briefly compare our FPGA-based HDR implementation with CPU- and GPU-based approaches. This section discusses typical performance characteristics (such as latency and throughput) based on previous studies and reported benchmarks. We highlight that FPGA offers advantages in deterministic low-latency processing and parallelism, making it suitable for real-time HDR applications, particularly when frame-by-frame responsiveness is critical.

Other Modifications

Additional corrections were made, including typos in the text.

1. Page 7, left : The word “8-layers” has been replaced with “8-layer”