

FlowAcc: Real-Time High-Accuracy DNN-based Optical Flow Accelerator in FPGA

Yehua Ling, Yuanxing Yan, Kai Huang, Gang Chen*

School of Computer Science and Engineering, Sun Yat-sen University, China
{lingyh6,yanyx8}@mail2.sysu.edu.cn;{huangk36,cheng83}@mail.sysu.edu.cn

Abstract—Recently, accelerator architectures have been designed to use deep neural networks (DNNs) to accelerate computer vision tasks, possessing the advantages of both accuracy and speed. Optical flow accelerator is however not among these architectures that DNNs have been successfully deployed. Existing hardware accelerators for optical flow estimation are all designed for classic methods and generally perform poorly in estimated accuracy. In this paper, we present FlowAcc, a dedicated hardware accelerator for DNN-based optical flow estimation, adopting a pipelined hardware design for real-time processing of image streams. We design an efficient multiplexing binary neural network (BNN) architecture for pyramidal feature extraction to significantly reduce the hardware cost and make it independent of the pyramid level number. Furthermore, efficient hamming distance calculation and competent flow regularization are utilized for hierarchical optical flow estimation to greatly improve the system efficiency. Comprehensive experimental results demonstrate that FlowAcc achieves state-of-the-art estimation accuracy and real-time performance on the Middlebury dataset when compared with the existing optical flow accelerators.

Index Terms—Optical flow, BNN, FPGA, real-time

I. INTRODUCTION

Accurate optical flow information is essential to many higher-level real-time computer vision tasks, such as robotic navigation, driving assistance, visual simultaneous localization and mapping (vSLAM), and 3-D reconstruction. In optical flow estimation, the most essential problem is to precisely establish pixel-wise correspondences between consecutive image frames. However, finding accurate correspondences in optical flow estimation is nontrivial since the 2-dimensional searching process is computationally highly complex and time-consuming.

In general, two kinds of methods can be employed to infer a dense correspondence field between two images. On one side, there are gradient-based methods [1], [2], [3] that yield accurate dense flow fields, but fail when the light intensity changes or displacements get too large. This is due to the fact that the gradient equation of these methods must satisfy stringent assumptions, which assume the displacements of the objects between two consecutive images are small and light intensity remains constant within a neighborhood of the corresponding pixels. However, in many real-world applications such as in autonomous driving, the complex environment as well as fast moving objects generally make these perfect assumptions not satisfied anymore, and therefore

results in low accuracy. On the other side, there are hierarchical block matching methods [4] based on feature descriptors that allow for large displacements, but correspondences are very sparse and have limited processing speed due to the complex structure of regularity constraints. Furthermore, the existing block matching methods are based on handcrafted features, which can easily cause the outlier pixels to generate minimal cost and result in low accuracy, due to light intensity change, warp, noise, etc in complex real-world scenarios.

Through using deep learning networks, optical flow estimation has achieved great performance improvement in recent years. Therefore, it is desirable to develop a dedicated hardware accelerator for DNN-based flow estimation combined with using hierarchical block matching to improve efficiency. Firstly, the huge resource demands of DNNs render it largely difficult to deploy such systems on resource-constrained platforms, such as FPGAs. Secondly, to enable hierarchical block matching, multiple DNNs are needed to perform parallel processing for feature extraction on the image pyramid, which significantly increases hardware resource consumption. Last but not least, the matching errors tended to be amplified and propagated during hierarchical block matching. Flow regularization at each level can alleviate this problem, but brings excessive resource consumption.

To overcome these challenges, in this paper, we present FlowAcc, the first DNN-based accurate real-time optical flow estimation architecture on FPGA. FlowAcc comprises two main modules, i.e., pyramidal feature extraction with BNNs and hierarchical optical flow estimation. FlowAcc exploits an efficient data flow for pyramid feature extraction of BNN that pyramid images can reuse a BNN. Using this data flow, FlowAcc can parallelly calculate the matching costs at different levels with minimal hardware resource consumption. Compared to existing FPGA-based optical flow estimation accelerators [1], [2], [3], [4], FlowAcc takes the advantages of both BNN and block matching for precisely establishing pixel-wise correspondences between consecutive image frames to calculate the exact optical flow endpoint, which significantly improves the accuracy. The major contributions of this paper are as follows:

- We propose FlowAcc, a DNN-based accelerator for real-time high-accuracy optical flow estimation on FPGA.
- We design an efficient flow regularization module for hierarchical optical flow estimation to improve the hardware efficiency.

*Corresponding author: Gang Chen. Email: cheng83@mail.sysu.edu.cn

- We design a temporal multiplexing BNN for pyramid feature extraction to eliminate the reduplicative hardware costs and make it independent of the pyramid level number.
- We design a novel block matching unit based on LUT-6s for reducing hardware resource consumption for hierarchical costs calculation.

We conduct comprehensive experiments and comparisons to evaluate our FlowAcc system on the challenging Middlebury optical flow dataset. Experimental results show that compared with existing FPGA designs, FlowAcc achieves state-of-the-art estimation accuracy and real-time performance of 131.5 frames/s at the resolution of 640×480.

II. RELATED WORK

Optical flow estimation accelerators implemented on FPGAs can be divided into three categories: gradient-based methods [5], [1], [6], phase-based methods [2], [3], and block matching [4]. Phase-based methods [2], [3] use quadrature filter to compute phase for optical flow estimation. Gradient-based methods such as Lucas–Kanade (L&K) methods [5], [1] solve an over-determined equation to generate the optical flow. Honegger et al. [6] exploit global and iterative processes to improve the accuracy for gradient-based methods. However, these methods use image gradient information to determine the direction of motion for pixels to generate optical flow. This leaves the exact endpoint of optical flow not calculated, and causes high endpoint error (EE). Furthermore, the assumption that the displacement of the pixel between consecutive image frames is small is not satisfied in real-time scenarios, for which gradient-based methods are thus not suitable. To solve this problem, Seyid et al. [4] implement a block matching method on FPGA for optical flow calculation and achieve 39 fps at 640×480 resolution and lower EE compared with gradient-based methods. But the handcrafted unreliable SAD features they used for block matching generally limit their matching accuracy.

Recently, researchers propose DNN-based optical flow calculation and achieve extremely high accuracy [7]. Nevertheless, these state-of-the-art methods are only accelerated on high-end GPUs and perform poorly in real-time performance because of the millions of parameters and floating-point calculations. In summary, the accelerator for real-time high-accuracy DNN-based optical flow calculation on FPGA there is still lacking. In this paper, we aim to develop an efficient BNN-based optical flow estimation system on FPGA to achieve a better balance between accuracy and real-time performance.

III. OPTICAL FLOW CALCULATION WITH BNN

As shown in Fig. 1, we follow [8] and design a temporal multiplexing BNN for pyramidal features extraction to significantly reduce the computational complexity. Instead of estimating the optical flow at a single level, we employ hierarchical optical flow estimation to reduce the hardware resource consumption. In hierarchical optical flow estimation,

coarse optical flows are obtained at level 1 and then refined and improved to sub-pixel accuracy by levels 2 and 3, as shown in Fig. 1. To obtain different level accuracy of the optical flows, the endpoints of the previous coarse levels are the start points for the next. At each pyramid level, the matching cost between pixel p of image I_1 and endpoints pixel $p + 2 \times \bar{m}\bar{v}_f^{l-1}(p)$ of image I_2 from previous level is established through computing hamming distance between BNNs extracted binary descriptors as shown in Eqn. (1).

$$C_l(p, \bar{m}\bar{v}) = H(f_1^l(p), f_2^l(p + \bar{m}\bar{v} + 2 \times \bar{m}\bar{v}_f^{l-1}(p))) \quad (1)$$

where the $H(\cdot)$ function returns hamming distance of the two binary descriptors; $l = 1, 2, 3$ is the pyramid level; $\bar{m}\bar{v} \in d \times d$ is the candidate motion vectors at level l ; $\bar{m}\bar{v}_f^{l-1}(p)$ is the motion vectors for pixel $p(x, y)$ at level $l-1$, and $\bar{m}\bar{v}_f^0(p) = 0$; $f_1^l(x)$ and $f_2^l(p + \bar{m}\bar{v} + 2 \times \bar{m}\bar{v}_f^{l-1}(p))$ are the binary descriptors of pixel p and pixel $p + \bar{m}\bar{v} + 2 \times \bar{m}\bar{v}_f^{l-1}(p)$, respectively. C_l is the matching cost. At each level, we use a winner-takes-all strategy (WTA) to select the coarse optical flow $\bar{m}\bar{v}_m^l(p)$, as computed as follows:

$$\bar{m}\bar{v}_m^l(p) = \operatorname{argmin}_{\bar{m}\bar{v} \in d \times d} C_l(p, \bar{m}\bar{v}) \quad (2)$$

The coarse optical flow can't be rectified at the next levels when the estimated correct endpoint is out of the short-range. To tackle this problem, we exploit local smoothness constraints [9] to regularize the optical flow after the WTA strategy. Local smoothness constraints define an energy function $E(\bar{m}\bar{v}_m^l(i))$ for each optical flow (i) on a support region (SR). And the energy minimum corresponds to the optimal optical flow. The defined energy function is computed as in Eqn. (3).

$$E(\bar{m}\bar{v}_m^l(i)) = H(f_1^l(p), f_2^l(p + \bar{m}\bar{v}_m^l(i))) + \lambda \Theta(\bar{m}\bar{v}_m^l(i)) \quad (3)$$

where $f_1^l(p)$ and $f_2^l(p + \bar{m}\bar{v}_m^l(i))$ are the extracted features from BNNs; $E(\bar{m}\bar{v}_m^l(i))$ is the energy function of optical flow $\bar{m}\bar{v}_m^l(i)$; $\Theta(\bar{m}\bar{v}_m^l(i))$ is the penalty factor for smoothness, which can be calculated as the distance between two vectors; λ is the smoothness weight for the penalty factor.

The smoothed optical flow is computed as:

$$\bar{m}\bar{v}_s^l(p) = \operatorname{argmin}_{1 \leq i \leq 9} E(\bar{m}\bar{v}_m^l(i)) \quad (4)$$

Before passing the optical flow to the next level, we employ a median filter to improve the accuracy of the optical flow at the end of each level. The final optical flow of pixel p at level l is computed as Eqn. (5).

$$\bar{m}\bar{v}_f^l(p) = \bar{m}\bar{v}_s^l(p) + 2 \times \bar{m}\bar{v}_f^{l-1}(p) \quad (5)$$

IV. PROPOSED HARDWARE ARCHITECTURE

Although short-range matching is inherently hardware-friendly, it still consumes huge hardware resources due to the following reasons. (1) We need to calculate the matching costs for hierarchical optical flow estimation at each level, which occupies considerable hardware resources. (2) At level 1, we use BNNs to extract a feature map for the down-sampled image to calculate the matching costs. The size of the

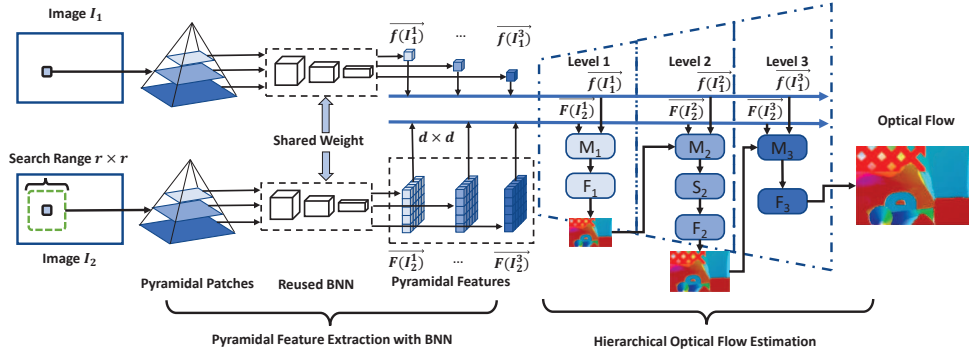


Fig. 1. Architecture overview of the proposed BNN-based optical flow estimation algorithm. Hierarchical optical flow estimation yields multi-scale flow fields that generated by module M (matching), S (smoothness constraint) and F (median filter).

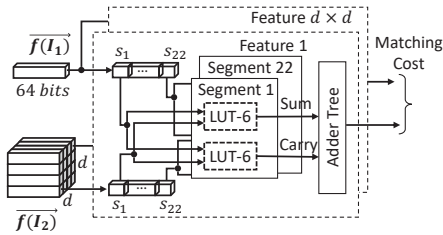


Fig. 2. Proposed LUT-6s based hardware architecture of block matching unit.

down-sampled feature map is a quarter of the original image. To match the original image and improve the accuracy, we use four down-sampled features to calculate a matching cost, which consumes more resources. (3) The feature map of level 3 is 4 times bigger than the original image because of pixel interpolation. Therefore, in level 3, four matching units are used to calculate the matching costs of sub-pixel accuracy.

To resolve these problems, we design a hardware-friendly hamming distance calculation unit for block matching, as shown in Fig. 2. In FPGA, logic circuits have synthesized the form of LUTs. In our work, we use Altera Stratix V FPGA with LUT-6s as the experimental chip. To this end, we utilize LUT-6s to build the circuit for hamming distance calculation. We integrate the XOR operation and full adder for 3-bit hamming distance calculation into two LUT-6s. As shown in Fig. 2, the 64-bit reference and matching features are split into 22 3-bit segments. Then, we exploit this integrated module to calculate the hamming distance for corresponding segments. Finally, we employ an adder tree to sum up the sum and carry bits to determine the final hamming distance. By using this architecture, the hardware costs are significantly reduced.

To improve the accuracy, we design a fully pipelined hardware structure for flow regularization process. The flow regularization contains two parallel modules: smooth cost calculation and penalty factor calculation. For penalty factor calculation, a 3×3 size support region for optical flow is obtained from the matching unit. Then, the penalty factor of each optical flow is calculated in parallel by a penalty factor

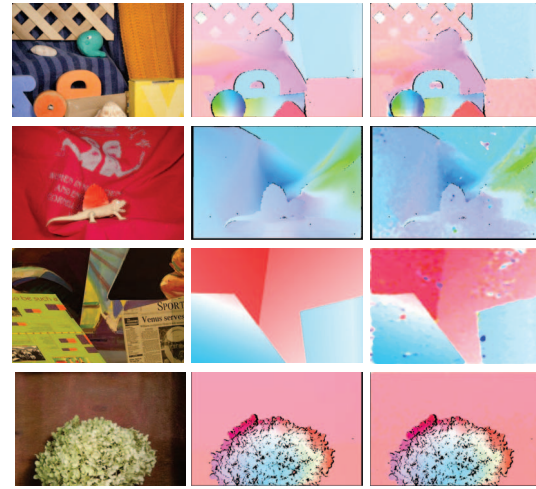


Fig. 3. Evaluation proposed system results on Middlebury benchmark. First column: input matching images. Second column: ground truth. Third column: the optical flow produced by FlowAcc.

calculation module. Then matching with a feature of I_1 to generate the smooth cost term. Finally, a parallel adder and a WTA module are used to sum up the smooth cost term and penalty factor term and select the smoothed optical flow. To further save resources, we reuse the feature support region of the block matching unit.

V. EXPERIMENTAL RESULTS

Tab. I shows the comparisons between FlowAcc and two software-based implementations and three FPGA-based implementations for the optical flow estimation on the Middlebury dataset. In our experiments, two software-based implementations including L&K and H&S methods from Piotr's Computer Vision MATLAB Toolbox [10] and three FPGA-based implementations are adopted for comparisons. As shown in Tab. I, compared to software-based and FPGA-based optical flow estimation systems, FlowAcc achieves the best accuracy in terms of AEE and AAE. This is attributed to the following reasons: (1) Instead of solving an overdetermined

TABLE I
PERFORMANCE COMPARISON ON MIDDLEBURY DATA SET.

Sequence	L&K [10]		H&S [10]		Seong et al. [1]		Seyid et al. [4]		Jang et al. [6]		Ours	
	AEE	AAE	AEE	AAE	AEE	AAE	AEE	AAE	AEE	AAE	AEE	AAE
RubberWhale	0.39	11.91 ^o	0.46	14.27 ^o	0.78	25.35 ^o	0.28	8.59 ^o	-	10.04 ^o	0.15	4.87^o
Dimetrodon	0.3	5.97 ^o	0.53	11.4 ^o	0.81	14.19 ^o	0.44	8.23 ^o	-	6.03 ^o	0.28	5.24^o
Venus	0.78	9.22 ^o	1.08	12.18 ^o	1.73	21.02 ^o	0.47	6.41 ^o	-	13.3 ^o	0.46	5.8^o
Hydrangea	0.6	4.35 ^o	0.64	5.93 ^o	1.18	9.73 ^o	1.98	14.8 ^o	-	5.07 ^o	0.23	2.85^o
Grove2	0.35	4.95 ^o	0.4	5.18 ^o	0.94	8.08 ^o	0.42	5.8 ^o	-	4.9 ^o	0.29	4.26^o
Grove3	1.45	10.19 ^o	1.18	9.44 ^o	1.76	15.8 ^o	0.99	10.9 ^o	-	15.51 ^o	0.81	7.96^o

TABLE II
REAL-TIME PERFORMANCE COMPARISON AMONG DIFFERNET OPTICAL FLOW ESTIMATION FPGA IMPLEMENTATION.

Evaluated Design	Method	Resolution	FPS	Density (%)	MPPS	FPGA
FlowAcc	BNN+Block matching	640×480	131.5	100%	40.4	Altera Stratix V
Tomasi et al. [3]	Phase-based	512×512	57.2	82.8%	12.41	Xilinx Virtex-4
Barranco et al. [5]	L&K	640×480	32	58.5%	9.8	Xilinx Virtex-4
Seyid et al. [4]	SAD+Block matching	640×480	39	-	-	Xilinx Virtex-7
Tomasi et al. [2]	Phase-based	640×480	31.5	92%	8.9	Xilinx Virtex-4
S.Jang et al. [6]	Global	640×480	64	100%	19.66	Xilinx Virtex 7
Seong et al. [11]	L&K	800×600	170	35.35%	28.85	Xilinx Virtex 6

gradient equation in L&K and H&S methods [10], FlowAcc uses block matching to find the corresponding pixel to generate more accurate optical flow. (2) Compared with SAD-based block matching method [4], FlowAcc achieves high accuracy because of BNN providing robust feature descriptors. (3) Compared with global optical flow method [6], FlowAcc uses hierarchical optical flow estimation to refine the accuracy to sub-pixel level and achieves higher accuracy. Additionally, some qualitative examples of FlowAcc on the Middlebury dataset are shown in Fig. 3. We can see that FlowAcc can generate smooth flow in texture-less and continuous regions and sharper flow in edges.

Tab. II compares the real-time performance between FlowAcc and several reported FPGA-based real-time optical flow estimation implementations. Compared to the phase-based [3], [2], L&K [5] and global method [6], FlowAcc achieves 2.3×, 4.17×, 4.11× and 2.05× speed up, respectively. This is because FlowAcc employs hierarchical optical flow estimation to reduce the circuit scale for improving the system frequency. Compared to the SAD+block matching work [4], [12], FlowAcc achieves 3.37× speed up while significantly improving the matching accuracy, as illustrated in Tab. I. The reason is that FlowAcc is designed as a fully pipelined end-to-end architecture for BNN and block matching which can estimate the optical flow in a streaming manner. Although FlowAcc has lower FPS than the L&K method [11], it is worth noting that this implementation provides only 35.35% density optical flow, which reduces the MPPS. FlowAcc achieves 1.4× processing throughput in terms of MPPS due to the 100% optical flow density.

VI. CONCLUSION

In this paper, we present FlowAcc, an efficient FPGA design for BNN-based real-time high-accuracy optical flow estimation. FlowAcc exploits multiplexing BNN to extract feature pyramid for hierarchical optical flow estimation and can be deployed on an FPGA. The experiment results show

that FlowAcc can achieve higher accuracy and processing speed when compared with existing FPGA implementations.

VII. ACKNOWLEDGEMENT:

This work was supported in part by the National Natural Science Foundation of China (NSFC) (Grant No.62072478) and the Science and Technology Planning Project of Guangzhou city of China (Grant No.202007050004).

REFERENCES

- [1] Han-Soo Seong and Hyuk-Jae Lee. A vlsi design of real-time and scalable lucas-kanade optical flow. In *2014 International Conference on Electronics, Information and Communications (ICEIC)*, 2014.
- [2] Matteo Tomasi et al. High-performance optical-flow architecture based on a multi-scale, multi-orientation phase-based model. *IEEE transactions on circuits and systems for video technology*, 2010.
- [3] Matteo Tomasi et al. Massive parallel-hardware architecture for multiscale stereo, optical flow and image-structure computation. *IEEE transactions on circuits and systems for video technology*, 2011.
- [4] Kerem Seyid et al. Fpga-based hardware implementation of real-time optical flow calculation. *IEEE Transactions on Circuits and Systems for Video Technology*, 2016.
- [5] Francisco Barranco et al. Parallel architecture for hierarchical optical flow estimation based on fpga. *IEEE transactions on very large scale integration (VLSI) systems*, 2011.
- [6] Sung-Joon Jang and Chong-Min Kyung. Resource-efficient and high-throughput vlsi design of global optical flow method for mobile systems. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2020.
- [7] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.
- [8] Gang Chen et al. Stereoengine: An fpga-based accelerator for real-time high-quality stereo estimation with binary neural network. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2020.
- [9] Chris Bartels and Gerard de Haan. Smoothness constraints in recursive search motion estimation for picture rate conversion. *IEEE Transactions on Circuits and Systems for Video Technology*, 2010.
- [10] Piotr Dollár. Piotr's Computer Vision Matlab Toolbox. <https://github.com/pdollar/toolbox>.
- [11] Han-Soo Seong et al. A novel hardware architecture of the lucas-kanade optical flow for reduced frame memory access. *IEEE transactions on circuits and systems for video technology*, 2015.
- [12] Guillermo Botella et al. Robust bioinspired architecture for optical-flow computation. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2009.