Histogram-Oriented Gradients Based Video Deblurring on FPGA

Hemalatha.N[1], Dr.T.Menakadevi[2], Dr.A.Kavitha[3]

[1]Ph.D Scholar, [2]Professor, [3]Professor
[1]Department of Electronics and Communication Engineering,
[1]Sudharsan Engineering College, Pudukkottai, Tamil Nadu

***Abstract*—When shooting videos while holding the camera, motion blur from camera shake is a significant problem. Video-based approaches, as opposed to single-image deblurring, may make extensive use of information between neighboring frames. Therefore, aligning adjacent frames is one of the best methods. However, image alignment is a computationally expensive and delicate process; therefore, algorithms that aggregate information must be able to distinguish between areas that have successfully aligned and those that have not, a task that demands a high level of scene knowledge. The approach for removing motion blur from the video is suggested in this work using a histogram of Oriented Gradients (HOG) features. By evaluating the video frame quality using the four performance measures PSNR, SSIM, MSE, and RMSE, it is validated. On a Xilinx Virtex 7 VX485T, the planned work was implemented.**

***Index Terms*— FPGA, Video Deblurring, fragile, Histogram of Oriented Gradients** *(key words)*

Abstract – When shooting videos while holding the camera, motion blur from camera shake is a significant problem. Video-based approaches, as opposed to single-image deblurring, may make extensive use of information between neighboring frames. Therefore, aligning adjacent frames is one of the best methods. However, image alignment is a computationally expensive and delicate process; therefore, algorithms that aggregate information must be able to distinguish between areas that have successfully aligned and those that have not, a task that demands a high level of scene knowledge. The approach for removing motion blur from the video is suggested in this work using a histogram of Oriented Gradients (HOG) features. By evaluating the video frame quality using the four performance measures PSNR, SSIM, MSE, and RMSE, it is validated. On a Xilinx Virtex 7 VX485T, the planned work was implemented.
Keywords – FPGA, Video Deblurring, fragile, Histogram of Oriented Gradients

I. INTRODUCTION

Computer vision is currently one of the most dynamic subfields in the world of information technology. One or more images are taken by a camera, which are later processed to extract information from them. When doing tasks like identifying features, detecting edges, and registering images, this information may then be utilised. However, it is not always possible to rely on the image quality because there is a chance that the shots contain noise or motion blur. If there are any movements between the camera and the subject during this time, the picture may be fuzzy. When taking a photograph, the camera must keep the shutter open for a specific amount of time to capture the necessary quantity of light and create a clean image. [1].

In addition to degrading the quality of visual perception, motion blur caused by camera shake and object movement also makes analysis tasks like object tracking more difficult to complete. In recent years, new tracking benchmarks have been developed to compare the accuracy of different trackers on videos with blurred frames in order to judge how well modern trackers can handle motion blur. The purpose of these benchmarks is to assess how effectively contemporary trackers can manage motion blur. These benchmarks, however, do not account for the impact of additional potential interferences such occlusion, subpar resolution, illumination shift, etc., which results in an erroneous assessment of a tracker's capabilities. Additionally, we are unable to provide in-depth research into how motion blur impacts tracking performance because we are utilising the current blur-related tracking benchmark, which prevents us from evaluating trackers' resistance to various levels of motion blur objectively. [2].

The majority of hand-shake motion blur is both transient and temporally uncorrelated; hence the most efficient video deblurring systems make use of information from nearby frames to sharpen hazy ones. By leveraging "sharp" pixels from nearby frames, a high-quality result can be recreated. Prior studies have shown that the patch-based synthesis, which uses either lucky imaging or weighted Fourier aggregation, significantly outperforms conventional deconvolution-based deblurring approaches [3].

However, without specifically addressing alignment, this strategy is unable to utilise the data from surrounding frames and is ineffective with movies that have a lot of motion. This shows that this method works well when the input frames are not overly blurry but fails miserably when there is a lot of blur in the input frames. Therefore, carry out experimental research to confirm that alignment and deblurring are required for deep networks to restore sharper frames to blurry movies [4].

Deblurring algorithms based on segmentation estimate many movements, many kernels, and related picture segments simultaneously. Since complex object movements and camera rotations result in locally varying blurring, this method is therefore ineffective [5].

Tugay and Dincer [6] suggested an inertial measurement unit (IMU) was used to measure the blur brought on by the movement of the IR detector. IMU IR detector angular position data was used to create the point spread function for each pixel. Both spatially invariant and variant blur situations can be applied to IR detector movements. After quantification, a Wiener-filter-based method was used to eliminate spatially invariant blur.

According to Sumi and Santha [7], preprocessing and detection are the two main tasks involved in pedestrian identification. Motion blur was introduced into the captured video during the preprocessing stage as a result of sensor and object movement. Before being analysed and categorised using histogram equalisation, this was eliminated using the Wiener filter. By removing the blur effect, the suggested method improves the rate of pedestrian detection.

Based on the principles of the histogram of oriented gradients (HoG), Ahlad Kumar [8] established an efficient approach for calculating the parameters of the point spread function (PSF). The estimate of the angle and velocity parameters of the motion-based PSF benefits particularly from HoG and statistical features. The blurred image is then reconstructed using a recently developed non-blind method in the moment domain after these parameters have been estimated. Rapid convergence and resistance to changes in the PSF's parameters are benefits of the moment domain. Experiments on artificial and real photographs show that the method successfully recovers the original images and produces outcomes comparable to those of the state-of-the-art techniques. By examining the image quality with PSNR and the well-known quality indicators BRISQUE and SSIM.

Reconfigurable hardware devices, which are quicker than software implementations, and VLSI technology were suggested by Sowmya et al. [9]. The use of descriptive hardware language allows the field of circuits to be extended to medical applications. To swiftly prototype various medical devices, Verilog Hardware Description Language (HDL)-based image enhancement approaches are required.

Nie et al. [10] proposed A high frame rate video reconstruction and deblurring technique is based on DAVIS events and low-speed blurred image sequences. First, we used an image denoising technique to represent events as a collection of event photos. Then, using the extended 3D partial recursive search (E-3DPRS) method, we developed high-performance motion vector fields (MVF) between close denoising event photos at a low computational cost. The object motion trajectory is represented by high-time resolution MVF. Furthermore, employing MVF as a priori knowledge to recover potentially crisp images reduces the amount of very ill-posed deblurring that needs to be done. Finally, we built a high frame rate sharp video utilising MVFs and a sharp picture sequence.

In Section II, the proposed approach is thoroughly explained, and the functions and implementation of each module are examined. Section III presents the findings and related discussions. The proposed architecture is contrasted with existing work in Section IV. The results were compiled in Section V, which also provided a conclusion for the suggested task.

II. PROPOSED HOG BASED VIDEO DEBLURRING FOR FPGA BASED REAL TIME APPLICATIONS

Fig.1. Four components, including the Hog computation module, the Response Matrix Generation Module, the convert to block Module, and the deconvolution Module, make up the proposed framework. There are also several additional fundamental FPGA modules. The control logic module, among others, has control over the FPGA's other modules. Data is initially received by the single port RAM. The HOG computation module receives the data after it has been transferred from the RAM into the RAM. The Hog feature of the input image is retrieved and used in the Hog computation module. Based on a desired frequency response, the Response Matrix Generation module can generate a magnitude matrix by computing the frequency response for a two-dimensional filter. The entire image is converted to blocks or divided into several sections in the convert to blocks module. The blocks are next subjected to the deconvolution process, and then the data is loaded into the RAM. The pixel intensity value modified using the image normalization block.

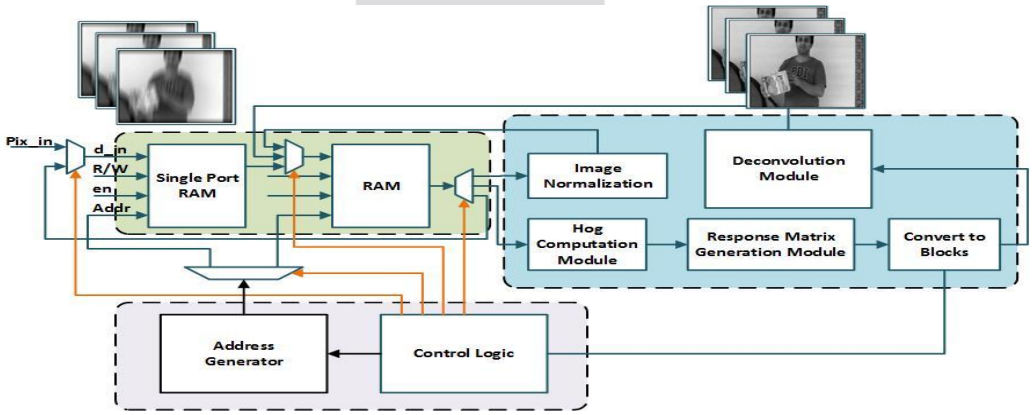


Fig.1RTL Architecture of proposed method

A. Histogram Oriented Gradient (HOG)

By counting the instances of angle direction in a subset of a picture, the HOG features connected with image blur reduction are found. The appearance and condition of facial features can be recognised by the distribution of force inclinations, which determines the HOG properties. The resulting inclusions effectively portray a visual signature and are quite selective. After being motivated by the outstanding results achieved for the confirmation of inclination, we applied HOG features for picture recovery [11].

For a M by N input picture, the first step of the HOG method typically entails 9xMxN multiplications and 2xMxN adds to determine the gradient in the x and y axes. The second step of the process involves calculating each pixel's magnitude and arctangent. As a result, MxNx2, MxN additions, and MxN evaluations of the square root and arctangent are required. The third stage involves comparing each pixel's angle to the range of 0 to 180 (or -90 to 90) degrees. Depending on the method and the data, this stage may include a different number of comparisons. However, nine bins can only support a maximum of nine comparisons. The bin value [12–15] for each pixel is then adjusted to reflect the magnitude value.

Therefore, MxN additions are present in this step. The histograms of a block are normalised in the next stage of the process, called normalisation. We must perform 9K divisions, 9K multiplications, 9K additions, and 1 square root computation for each block of K cells. The normalisation of each block's histograms is the next stage of the HOG approach. After the magnitudes are categorised into various bins, the histograms of the cells inside a block are normalised. The display of the HOG algorithm's cell and block is shown in Fig. 2.

$$g = \sqrt{g_x^2 + g_y^2}$$
$$\theta = arctan\frac{g_y}{g_x}$$

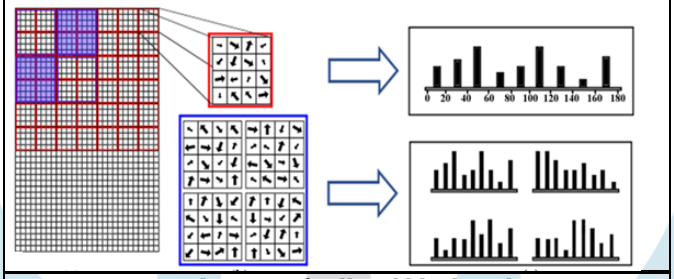


Fig.2 Visualization of cell and block in the HOG algorithm.

B. Image Normalization

The normalising technique in image processing modifies the range of pixel intensity values to achieve greater consistency. Applications can include glare-induced poor contrast images. In some cases, normalisation is also referred to as histogram stretching or contrast stitching. In more general data processing fields like digital signal processing, it is known as dynamic range expansion [16–17]. The term "normalisation" refers to the process of putting a picture or other signal into a range that is most recognisable or normal to the senses, which is the goal of dynamic range extension in most applications. To lessen mental fatigue or distraction, it is frequently desirable to provide consistency in dynamic range for a group of data, signals, or pictures [18]. For instance, a newspaper makes an effort to guarantee that the grayscale range of every photo in an issue is uniform. The linear normalisation of a grayscale digital image is described by the formula below.

$$I_N = (I - Min)\frac{newMax - newMin}{Max - Min} + newMin$$

Ensures that the database always has consistent data. More flexible database design. Better protection for the database.

C. Deconvolution Module

A technique called deconvolution, which is based on an algorithm, can undo the effects of convolution on record data. Deconvolution generally aims to find a solution to a convolution equation of the specified form:

$$f * g = H$$

The process of reversing the optical distortion that The process of undoing the optical distortion caused by imaging equipment, which ultimately produces cleaner images, is referred to as "deconvolution" in the domains of optics and imaging. Deconvolution can also be used to sharpen images that have been blurred by fast motion or errors during the capturing process. The accepted practice is to assume that the optical route through the device is flawless optically. The point spread function, a mathematical

function that characterizes the distortion in terms of the path that a hypothetical point source of light (or other waves) takes through the instrument, is then convolved using this assumption.

III. RESULTS AND DISCUSSIONS

The algorithmic verification of the sample rate conversion algorithm is done with the MATLAB 2020a tool.

Quality Metrix

In this section, present the experimental results supporting the proposed method for motion blurring based on the Histogram of directed gradients. There have been three experimental groups.

1) Mean Square Error (MSE)

The MSE assesses the consistency of an estimator; it is typically non-negative, with values closer to zero being higher [19]. Represents the contrast between the original and decrypted pictures in MSE depicted in Equation (6).

$$MSE = \frac{1}{Px*Px}\sum_{i=1}^{Px}\sum_{j=1}^{Px} | \hat{I}(i,j) - I(i,j)|^2 (6)$$

2) PSNR

PSNR is the ratio of a signal's full potential strength to the input power when it is entirely contaminated [20]. PSNR is expressed as Equation (7).

$$PSNR = 20.\log_{10} MAX_{PY} - 10.\log_{10} MSE \quad (7)$$

Where MAX_{PY} represents a maximum image pixel value.

3) SSIM

SSIM is a viewpoint paradigm that interprets image loss as a perceived change in structural features and often integrates fundamental visual phenomena [21], such as the strength of light masking and intensity masking concepts shown in Equation (8).

$$SSIM(i,j) = \frac{(2k_i k_j + r1)(2l_{xy} + r2)}{(k_i^2 + k_j^2 + r1)(l_i^2 + l_j^2 + r2)} \quad (8)$$

4) Root-Mean-Square Error (RMSE)

The standard deviation of the residuals may be computed with the help of the RMSE. A statistic that measures how far distant the data points are from the regression line is called the residuals; it denotes described in Equation (9).

$$\text{RMSE} = \sqrt{\text{E} - \text{K}} (9)$$

Where E is the expected value and K is known results.

D. Hardware Performance

The performance matrices that are most often used have been used. It is comprised of the area of the chip as well as the wire length.

1) Area of estimation

It is the rectangle of the smallest possible size that can include all of the bricks.

$$A = (\max(x + w) - \min(x))(\max(y + h) - \min(y)) (10)$$

Where *h* represents the height and *w* represents the width. Fig.3.depicts the sample blurred and deblurred video frame. Fig.4 depicts photographs of real-world scenes that have been blurred together with the distribution of the gradients in those video frames, as well as the Hyper-Laplacian function that most accurately matches that functions. Fig.5 shows the performance of PSNR. Fig.6 shows the comparative Performance of SSIM. Fig.7 shows the comparative value of MSE. Fig.8 shows the comparative value of RMSE. Table I. shows the resource usage for Xilinx Virtex 7 VX485T FPGA device. When the image size changes, it needs a reconfiguration controller. Control Unit for LUTs are 1347 and FFs are 113.Gradients Calculator for LUTs are 112, FFs are 35and BRAMs is 1. The reconfiguration controller has 150 LUTs and 66FFs. Used total LUTs are 8,716 (2.87%), FFs are 1,613 (0.27%) BRAMS are 72 (3.50%) and DSPs are 192 (6.86%).

Table 1. Resource Usage for Xilinx Virtex 7 VX485T FPGA device

Module	**FPGA Area Occupation**			
	LUTs	**FFs**	**BRAMs**	**DSPs**
Control Unit	1,347	113	-	-
FFT(y)	2,207	376	16	94
FFT(w)	2,207	376	16	94

Gradients Calculator	112	35	1	-
α estimator	315	34	2	-
w calculator	265	53	37	-
Reconfiguration controller	150	66	-	-
Formula Solver	2,113	560	-	4
Total	8,716 (2.87%)	1,613 (0.27%)	72 (3.50%)	192 (6.86%)

Table 2 Comparison of device utilization with other methods

Method	**Image size**	**LUT**	**Slice Register**	**Delay(ns)**	**Throughput (Mbps)**	**Power (mW)**
Kumar A, 2017 [8]	256x256	-	14	0.83	9.6	-
Remez, Tal, et al., 2016 [18]	192x192	6271	387	6.085	80.142	103
Scholz, Nikolas, et al., 2016 [19]	1280x1024	1034	-	-	188	992
Proposed work	128x128	252	66	0.72	87.8	89
	256x256	284	72	0.836	93.15	103
	512x512	329	83	0.12	97.15	128

.

Table 2 shows the comparison of device utilization with other methods. The method [8] with Image size 256x256 has Slice Register of 14, 0.83 delay and 9.6 throughput. The 192x192 image size methodology has 6271 of LUT, 387 of slice register, 6.085 of delay, 80.142 of throughput and 103 of power. The method [19] 1280x1024 image size has LUT of 1034, slice register of 188 and delay of 992. In this work, 128x128 image size achieves LUT of 252, slice register of 66, delay of 0.72, throughput of 87.8 and power of 89. The 256x256 image size achieves return LUT of 284, slice register of 72, delay of 0.836, throughput of 93.15 and power of 103 mW. When using 512x512 image size achieves LUT of 329, slice register of 83, delay of 0.12, throughput of 97.15 and power of 128 mW. With comparison to previous works discussed here, the size of the LUT, slice register, delay, and power in this work are reduced. Simultaneously, the throughput is increased.

Table 3 Performance comparison of video quality

Method	**PSNR**	**SSIM**	**MSE**	**RMSE**
[8]	48.14	0.93	0.07	0.43
[18]	24.97	0.84	0.042	0.62
[19]	39.12	0.79	0.072	0.64
Proposed work	52.36	0.95	0.006	0.04

Table 3 shows the performance comparison of video quality. The method [8] achieves 48.14 of PSNR, 0.93 of SSIM, 0.07 of MSE and 0.43 of RMSE. The technique [18] returns 24.97 of PSNR, 0.84 of SSIM, 0.042 of MSE and 0.62 of RMSE. [19] has 39.12of PSNR, 0.79 of SSIM, 0.072 of MSE and 0.64 RMSE. This work achieves 52.36 of PSNR, 0.95 of SSIM, 0.006 of MSE and 0.04 of RMSE.

IV. COMPARISON WITH RELATED METHODS

A comparison with related methods shows that the method represented in this paper has lesser delay and power, higher PSNR and SIM and produced a lower error.

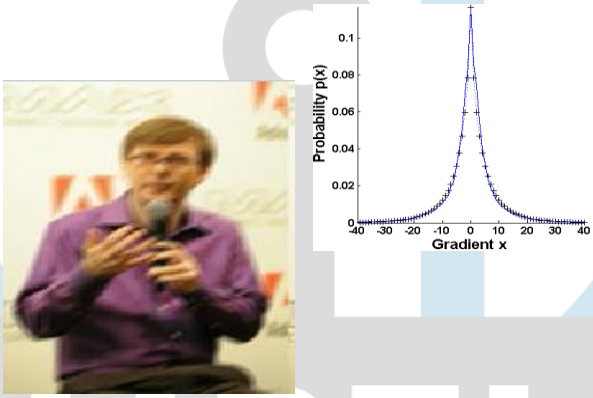
The authors in [8] did not discuss LUT and power but they tried to reduce the slice register and delay. To compare our algorithm with the method presented in [8], applied the algorithm to a similar size image with the same parameters, which resulted in the value of LUT=284, Slice register=72, Delay=0.836ns, Throughput=93.15Mbps and Power=103mW.

In another paper [18] researchers presented the value of LUT=6271, Slice register=387, Delay=6.085ns, Throughput=80.142Mbps and Power=103mW which are not as good as the ones obtained with our method.

In [19], the experimental results were presented briefly and there were no overall experimental results. The authors did not provide any information about the Slice register and Delay. As the authors said, the estimated value of LUT is 1034, Throughput is 188Mbps and Power=992mW.

The amount of MSE and RMSE in this work is reduced compared to the prior efforts presented here. At the same time, the PSNR and SSIM values are raised.

Fig.3.Sample resized video frame



Original video frame Image gradients and Hyper Laplacian fitting

Fig.4 Video frame of real-world scenes that have been blurred, together with the distribution of the gradients in those video frames, and the Hyper-Laplacian function that best matches those functions

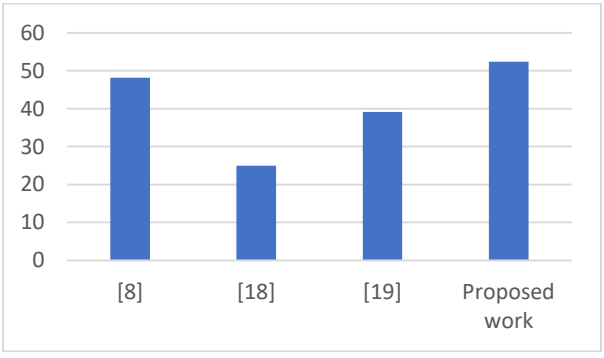


Fig.5 Comparative performance of PSNR

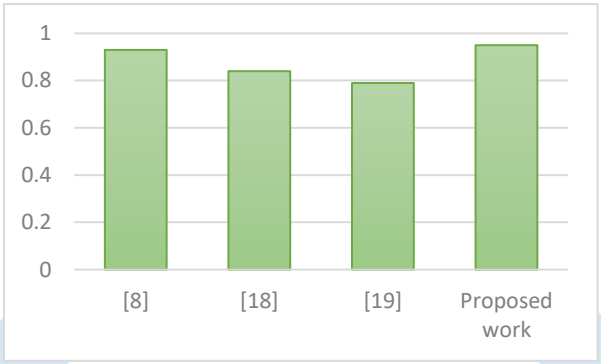


Fig.6 Comparative performance of SSIM

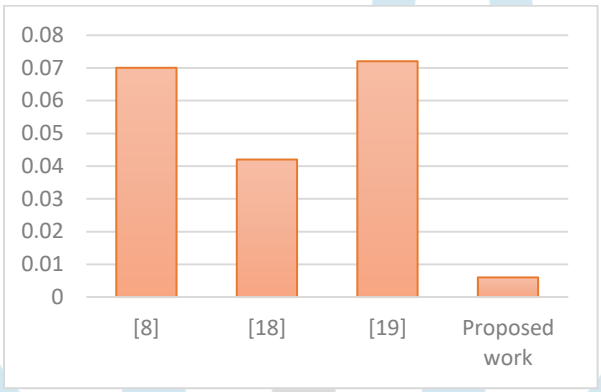


Fig.7 Comparative value of MSE

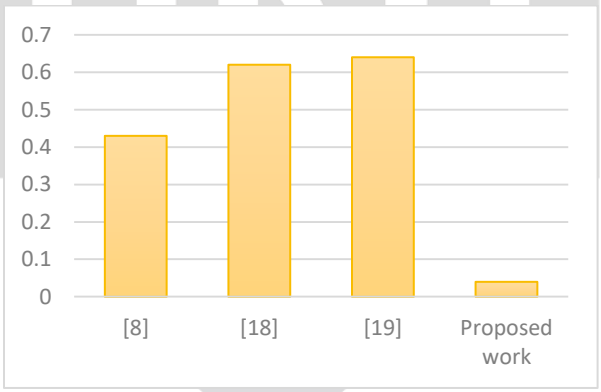


Fig.8 Comparative value of RMSE

Table 4 Performance of sample video frames

Image	PSNR	SSIM	MSE	RMSE

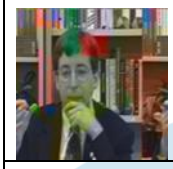
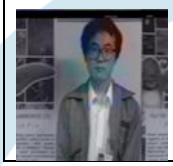
	51.46	0.98	0.008	0.02
	49.76	0.96	0.002	0.06
	50.16	0.97	0.010	0.03

Table 4 shows the performance of sample video frames. The PSNR, SSIM, MSE, and RMSE for the Akiyo video frame are 51.46, 0.98, 0.008, and 0.02, respectively. The deadline video frame gets 49.76 of PSNR, 0.96 of SSIM, 0.002 of RMSE and 0.06 of MSE. 50.16 of PSNR, 0.97 SSIM, 0.010 of MSE and 0.03 of RMSE are achieved by the bowing video frame.

V. CONCLUSION

This paper presents HOG, an FPGA-based high-performance adaptive deblurring core for real-time applications. HOG can automatically adjust the deblurring parameter to the characteristics of the image, producing results that are more accurate. The experimental results showed that the FPGA uses fewer hardware resources than a static deblurring technique and that the quality of the recovered latent picture is higher. These experimental results demonstrate an improvement in the precision of all subsequent image processing modules that use the deblurred image as an input, such as edge detectors. 52.36 PSNR, 0.95 SSIM, 0.006 MSE, and 0.04 RMSE are achieved in this work. Compared to prior findings, our implementation architecture is significantly more efficient, needing fewer LUTs (252) and slices (66) while generating high throughput (87.8 Mbps) with minimal overhead.

VI. CONFLICT OF INTEREST

The authors do not have any conflict of interest.

References

[1] Zhang, Xiaoqin, Runhua Jiang, Tao Wang, and Jinxin Wang. "Recursive neural network for video deblurring." IEEE Transactions on Circuits and Systems for Video Technology 31, no. 8 (2020): 3025-3036.

[2] Guo, Qing, Wei Feng, RuijunGao, Yang Liu, and Song Wang. "Exploring the effects of blur and deblurring to visual object tracking." IEEE Transactions on Image Processing 30 (2021): 1812-1824.

[3] Su, Shuochen, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. "Deep video deblurring for hand-held cameras." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1279-1288. 2017.

[4] Zhou, Shangchen, Jiawei Zhang, Jinshan Pan, HaozheXie, WangmengZuo, and Jimmy Ren. "Spatio-temporal filter adaptive network for video deblurring." In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2482-2491. 2019.

[5] Vedavyas, Y., S. Sri Harsha, M. Sai Subhash, and S. Vasavi. "Quality Enhancement for Drone Based Video using FPGA." In 2022 International Conference on Electronics and Renewable Systems (ICEARS), pp. 29-34. IEEE, 2022..

[6] Doner, Tugay, and DincerGokcen. "FPGA-based infrared image deblurring using angular position of IR detector." The Visual Computer 37, no. 7 (2021): 2039-2050..

[7] Sumi, A., and T. Santha. "Motion deblurring for pedestrian crossing detection in advanced driver assistance system." In 2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), pp. 1-4. IEEE, 2017.

[8] Kumar, A., 2017. Deblurring of motion blurred images using histogram of oriented gradients and geometric moments. Signal Processing: Image Communication, 55, pp.55-65.

[9] Sowmya, K. B., T. S. RakshakUdupa, and Shashank K. Holla. "Implementation of an FPGA Real-Time Configurable System for Enhancement of Lung and Heart Images." In Advances in Multidisciplinary Medical Technologies─Engineering, Modeling and Findings, pp. 199-213. Springer, Cham, 2021.

[10] Nie, Kaiming, Xiaopei Shi, Silu Cheng, ZhiyuanGao, and JiangtaoXu. "High Frame Rate Video Reconstruction and Deblurring Based on Dynamic and Active Pixel Vision Image Sensor." IEEE Transactions on Circuits and Systems for Video Technology 31, no. 8 (2020): 2938-2952.
[11] Hier, R. G., and R. C. Puetter. "Real-time hardware realization of a deblurringanisoplanaticism corrector (DAC) for improvement of long-range video imagery." In Unconventional Imaging IV, vol. 7094, pp. 95-100. SPIE, 2008.
[12] Wang, Meng, ShengyuHou, Huafeng Li, and Fan Li. "Generative image deblurring based on multi-scaled residual adversary network driven by composed prior-posterior loss." Journal of Visual Communication and Image Representation 65 (2019): 102648.
[13] Liang, Chih-Hung, Hung-Ting Su, and Winston H. Hsu. "Learn from the past–sequentially one-to-one video deblurring network." Journal of Visual Communication and Image Representation 78 (2021): 103159.
[14] Saptalakar, Bairu K., and Mrityunjaya V. Latte. "FPGA-based reflection image removal using cognitive neural networks." Applied Nanoscience (2022): 1-15.
[15] Popovic, Vladan, KeremSeyid, EliévaPignat, ÖmerÇogal, and Yusuf Leblebici. "Multi-camera platform for panoramic real-time HDR video construction and rendering." Journal of Real-Time Image Processing 12, no. 4 (2016): 697-708.
[16] Szydzik, Tomasz. "Contributions to non-iterative super-resolution algorithms enabling efficient real-time FPGA implementation for resolution enhancement of video sequences." PhD diss., 2015.
[17] Kim, Michael Duckjune, and Jun Ueda. "Real-time panoramic image generation and motion deblurring by using dynamics-based robotic vision." IEEE/ASME Transactions on Mechatronics 21, no. 3 (2015): 1376-1387.
[18] Remez, Tal, Or Litany, ShacharYoseff, HarelHaim, and Alex Bronstein. "FPGA system for real-time computational extended depth of field imaging using phase aperture coding." arXiv preprint arXiv:1608.01074 (2016).
[19] Scholz, Nikolas, Jochen Moll, Moritz Mälzer, Konstantin Nagovitsyn, and Viktor Krozer. "Random bounce algorithm: real-time image processing for the detection of bats and birds." Signal, Image and Video Processing 10, no. 8 (2016): 1449-1456.
[20] Arslan, Ahmet, GökhanKorayGültekin, and AfşarSaranlı. "Adaptive Mesh-Grid Based Spatially Non-Uniform Video Motion Deblurring Using Imu-Camera Sensor Fusion." Available at SSRN 4039888 (2021).
[21] Van Toan, Nguyen, and Jeong-Gun Lee. "FPGA-based multi-level approximate multipliers for high-performance error-resilient applications." IEEE Access 8 (2020): 25481-25497.