HARDWARE ACCELERATION OF LUCKY REGION FUSION ALGORITHM FOR IMAGING

by

William Maignan

A thesis submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Master of Science in Electrical and Computer Engineering

Spring 2013

© 2013 William Maignan
All Rights Reserved

HARDWARE ACCELERATION OF LUCKY REGION FUSION ALGORITHM FOR IMAGING

by

William Maignan

Approved: _____
Fouad E. Kiamilev, Ph.D.
Professor in charge of thesis on behalf of the Advisory Committee

Approved: _____
Kenneth E. Barner, Ph.D.
Chair of the Department of Electrical and Computer Engineering

Approved: _____
Babatunde A. Ogunnaike, Ph.D
Interim Dean of the College of Engineering

Approved: _____
James G. Richards, Ph.D.
Vice Provost for Graduate and Professional Education

ACKNOWLEDGMENTS

First and foremost, I thank my advisor Fouad Kiamilev. Working under him has truly been a phenomenal experience. I am extremely grateful that he allowed me to join his research group. I would never have had the opportunity to write this thesis without his continual support and encouragement. He made me feel welcome from the very beginning with the family-oriented environment that he promotes.

Next I want to thank Dean Michael Vaughan, Dr. Stephen Cox, Ms. Veniece Keene, and the Greater Philadelphia Region Louis Stokes Alliance for Minority Participation (LSAMP) for awarding me the Bridge to the Doctorate (BTD) Fellowship. Without this fellowship, graduate school might not have been possible for me.

I would also like to thank Dr. Jiang Liu, Dr. Mathieu Aubailly, and Mr. Gary Carhart at the Intelligent Optics Lab, part of the U.S. Army Research Laboratory (ARL) Computational and Information Sciences Directorate (CISD) for their collaboration and for funding the project through the co-operative agreement from the US ARMY RDECOM under contract number W911NF-11-2-0088.

I also thank everyone in CVORG, especially David Koeplinger who played an instrumental role in the project.

Last, but certainly not least, I would like to thank my support system: my Mom, Dad, brother Philip, and sister Maureen for their love, support, and encouragement.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

"Lucky-region fusion" (LRF) is an image processing technique that has proven successful in enhancing the quality of images distorted by atmospheric turbulence. The LRF algorithm extracts sharp regions of an image obtained from a series of short exposure frames, and "fuses" them into a final image with improved quality. In previous research, the LRF algorithm had been implemented on a PC using a compiled programming language. However, the PC usually does not have sufficient processing power to handle real-time extraction, processing and reduction required when the LRF algorithm is applied not to single picture images but rather to real-time video from fast, high-resolution image sensors. This thesis describes a hardware implementation of the LRF algorithm on a Virtex 6 field programmable gate array (FPGA) to achieve real-time video processing. The novelty in our approach is the creation of a "black box" LRF video processing system with a standard camera link input, a user controller interface, and a standard camera link output.

Chapter 1

INTRODUCTION

1.1 Background

Similar to haze rising from the ground on a hot day causing a heat mirage, atmospheric turbulence causes distortions and mutations in images. As the sun heats the ground during the day (most strongly at midday) it also heats the layer of air closest to the ground. This causes air pockets to rise and variations of the refractive index of air. Besides uneven heating from the sun, the refraction index of the air varies based on several other atmospheric characteristics including the air's temperature, humidity, pressure, carbon dioxide level, and dust density. When the air pockets are disturbed by wind current, the atmosphere can be characterized as "turbulent." Turbulence also occurs at night when air temperature is higher than the ground temperature [1,2].

Atmospheric turbulence caused by variation of refractive index along the optical transmission path strongly affects the performance of long range imaging systems. It can produce combinations of geometric distortion (motion), space and time varying blurs, locally varying blurs, and out-of-focus blurs when the turbulence is violent [3,4]. The optical effects of atmospheric turbulence hinder the recognition range along horizontal optical transmission paths at ground and sea level[5]. All of these effects increase in severity as the distance between the target image and imaging system increases.

Imaging through atmospheric turbulence is one of the oldest and perhaps most frequently repeated experiments. However, the problem of image quality

preservation and restoration remains one of the most complicated problems in modern optics. This is especially true in military scenarios where it is of crucial importance to observe targets over long distances. The solutions to the problem of imaging through turbulence generally fall in two categories: hardware solutions and software solutions. Many of these solutions have been developed in the past. Some hardware solutions use adaptive optics and are used for astronomical applications. These solutions are often expensive, large, and complex [5,6,7]. Existing software solutions include a variety of post detection image-processing techniques, such as lucky-region fusion, speckle imaging, blind deconvolution, phase diversity, or intensity transport [1-12]. Software solutions are much cheaper, smaller and less complex than hardware. Nonetheless with the fast, high-resolution imaging sensors available today even simple software solutions to the optical problems caused by turbulence face the danger of overloading any computer memory or communication link within a fraction of a second. Furthermore, PCs usually do not have the processing power to handle real-time extraction, processing, and reduction of information from a real-time visual data stream [6,11].

This thesis describes the lucky-region fusion (LRF) algorithm, which is a computationally inexpensive but effective software image processing technique, being accelerated in hardware to achieve real-time, live image processing. Section 1 reviews the results of "Automated video enhancement from a stream of atmospherically-distorted images: the lucky-region fusion approach." [8]. Section 2 describes the lucky-region fusion algorithm and its implementation in a field programmable gate array (FPGA) embedded in a high-speed camera. Section 3 details the expanded implementation of the LRF algorithm where the image processor is a "black box" with the FPGA "decoupled" from the camera.

1.2 Lucky Region Fusion

Real-time, high resolution, long-range imaging is often useful in military field applications like remote surveillance and target identification. Unfortunately, the compounded effects of natural atmospheric turbulence over long distances can introduce variations in local image quality, causing the overall image or video to appear blurred and distorted. The "lucky-region" fusion algorithm is one technique currently being developed to combat these effects. The variations in image quality caused by turbulence include high-resolution regions ("lucky regions"). The LRF technique locates and combines these regions together in order to create a final synthetic image with an overall improved image quality.

The LRF algorithm improves local image quality using the following steps:

1. Compute the image quality map (IQM) for each image in the incoming video stream. The IQM gives a spatial representation of the clarity of the image based on the distribution and sharpness of edges in that image.
2. Compare each IQM to that of the current fused image to determine which areas – if any – are clearer in the incoming frame than in the current synthetic image.
3. Merge the selected regions into the fused video stream.

The LRF algorithm has previously been implemented on a workstation PC using a compiled programming language. However, even when accelerated with a GPU, this platform may not have adequate processing power to handle real-time processing of data from high-resolution image sensors [8].

1.3 Motivation

Carhart, Vorontsov, and Aubailly developed the LRF algorithm. The algorithm is proven to mitigate the effect of atmospheric turbulence, but was implemented on a workstation PC using a compiled programming language. The PC does not have adequate processing power to handle real-time processing of video stream. The motivation of the Hardware Accelerated Lucky Region Fusion Algorithm is to take the

proven algorithm and using hardware to accelerate the algorithm to make it practical in real world applications.

The LRF algorithm is implemented and accelerated on hardware - a field programmable gate array (FPGA). A FPGA is an integrated circuit intended to be configured "in the field" using a hardware description language (e.g. Verilog or VHDL). A single FPGA can contain hundreds of thousands to millions of configurable digital gates. FPGAs have the distinct advantage of being reconfigurable like a software design while also being capable of parallel processing like a custom application-specific integrated circuit (ASIC) design. The maximum processing bit rate of an FPGA is higher than most sequential processors because of this parallel processing capability. Consequently, implementing an algorithm on a FPGA can accelerate the algorithm by approximately 20 times when compared to a desktop PC.

Chapter 2

LITERATURE REVIEW

2.1 Atmospheric Turbulence

Atmospheric turbulence caused by a variation of refractive index along the optical transmission path strongly affects the performance of long-distance imaging systems. Turbulence produces geometric distortion, motion blur (if the exposure time is not sufficiently short), and sometimes out-of-focus blur when the turbulence is violent [3]. Videos captured by optical sensors in the presence of atmospheric turbulence are very unstable and degraded in their resolution and geometry [14].

Overcoming the distortions caused by atmospheric turbulence in long distance imaging systems has been studied for a long time and carries a lot of interest in many fields such as astronomy (i.e. planet exploration), geology (i.e. topographical measurements), ecology, traffic control, remote sensing, and homeland security (surveillance and military intelligence) [14]. A great number of solutions have been created to mitigation of these effects. The solutions could be generalized into two categories, hardware and software solutions [10].

2.2 Hardware Solutions

The hardware solutions refer to the use of Adaptive Optics (AO) applications in the field of astronomy [10]. The astronomical community is given credit for revolutionary research and the Adaptive Optics solution regarding atmospheric degradation due to turbulence. The solution was developed by astronomers that used ground-based telescopes to look through kilometers of the Earth's atmosphere

and needed to find a way to mitigate the effect of the turbulence. AO systems measure the wavefront phase errors generated by variations in the atmosphere refraction index and use a deformable mirror (DM) actuated in real time to compensate the effect of the turbulence to achieve an angular resolution close to the diffraction limit of a telescope. The wavefront reconstruction is normally computed using fast Fourier transforms (FFT) and spatial filtering. The computations have previously been done on a normal PC using the Central Processing Unit (CPU) [1, 15-17].

However, adaptive optics for the next generation of large (10 m diameter) and giant telescopes (50-100 m diameter) requires a massive amount of processing power that goes beyond the practical limits of today's CPUs for real time compensation of the atmospheric turbulence. In [17], Graphical Processing Units (GPUs) were used to handle the computations necessary for the AO system. The GPUs provided abundant computational power with satisfactory execution time results but the results were not optimized due to the rigid architecture of the GPU. Magdaleno implemented an FPGA prototype to accelerate the calculation of wavefront reconstruction, taking advantage of the parallel architecture and high-performance signal processing of the FPGA. The implemented system was faster than the CPU or the GPU solution. Furthermore, the implemented design incorporating the FPGA met current and future adaptive optics image processing frame rate requirements [15,17,18].

This has been the industry trend in the astronomy community. FPGAs have been added to AO systems to process the large amount of data coming from the large telescopes and to compute calculations at the necessary rates. In [19], Gavel used an FPGA to handle the large amount of computation needed to implement multi-conjugate adaptive optics at atmospheric turnover rates. In [16], Rodríguez Ramos used a FPGA as the processor in the FastCam system, which involves a hardware implementation of Lucky Imaging. The FPGA dramatically improved the performance of the instrument because of the real-time behavior. In [20],

Venugopal utilized FPGAs and GPUs for the real-time parallel processing of a four meter Advanced Technology Solar Telescope (ATST) adaptive optics (AO) system. These projects validate how FPGAs can be used as the processing heart of an adaptive optics system control. They provide simplicity, low power consumption, small physical size, and low relative cost [15]. Although the astronomy community relies on adaptive optics systems, cost, size and complexity are the main drawbacks of building adaptive optics system. Many of the other fields that use long-range imaging rely on software to mitigate atmospheric turbulence [16].

2.3 Software Solutions

Current software solutions for atmospheric imaging systems can generally be divided into two main categories or two types of algorithms. One important category is based on multi-frame reconstruction. This type of approach first requires a non-rigid image registration technique to register each observed frame and use the registration parameters to further estimate the deformation [3].

The second class of algorithms employs image selection and fusion methods to reduce the blurring effects caused by turbulence. The frame selection technique finds frames of the best quality (lucky frame) from a short- exposure video stream, and the output image is produced by fusing the lucky frames. The "lucky region" technique takes a series of short-exposure images, with turbulence created "mutations" in local image quality, and randomly makes some regions sharper, which are called lucky regions. The lucky regions can be detected by a sharpness metric, and then fused to produce a high quality image [3].

All other algorithms can be categorize into one of the above two classes. For example in [10] Huebner's work combines the two categories to create a new algorithm. Huebner use software-based approach is Motion-Compensated Averaging (MCA), which is a relatively simple averaging technique that has proven quite effective for the mitigation of turbulence effects and combined it with the "Lucky Imaging" technique. In [1], Baumgartner created a new video quality

measure quantifying the success of turbulence mitigation on real data where no reference imagery is available, using the lucky region imaging framework.

The computational complexity of software solutions can grow exponentially, and easily can reach a point where the algorithm is too complex to be considered practical. Many algorithms that have been created to mitigate the effects of atmospheric turbulence work in laboratory experiments, but are too computationally intense to be implemented for real-time processing on modern desktop PCs.

2.4 Hardware Acceleration of LRF

Hardware Acceleration of the Lucky Region Fusion Algorithm takes a proven atmospheric turbulence mitigation algorithm, Lucky Region Fusion, and accelerates the algorithm to make it practical. The black box approach to atmospheric turbulence is cutting edge.

In comparison, EM Photonics sells a hardware-accelerated version of Lawrence Livermore National Labs (LLNL) algorithm. This is the only known similar hardware acceleration for imaging through turbulence product. The LLNL average-bispectrum speckle algorithm combines information from several images, taken a short time apart from one another. The images can be from a series of multiple short-exposure still shots from a conventional camera or, more commonly, a sequence of consecutive video frames. This information is processed using "averaging" procedures in the bispectral domain, where the magnitude and phase are estimated independently and subsequently recombined in the real space. To accommodate the spatially varying point-spread functions experienced in earth-bound imaging, overlapping sub- fields of the image, or tiles, are separately processed and reassembled to form the full field. As a result, the algorithm produces a single corrected image with quality near the diffraction limit [21]. The image processing method LLNL uses is different from LRF. Their algorithm is computationally heavy compared to LRF since it does 3-D FFT computations. L-3 Cincinnati Electronics implemented an FPGA based version of a multi-frame joint

TMSR (turbulence mitigation (TM) and super-resolution (SR)) algorithm create by University of Dayton that provides a real-time, low-power, compact hardware implementation suitable for terrestrial applications. The technique that the University of Dayton uses (deblurring and dewarpping) also differs from the LRF algorithm [22].

EM Photonics GPU accelerated version processes 720p (1 megapixel/frame) images at 10 frames per second. The FPGA accelerated version processes 720p images at 60 frames per second. EM Photonics hardware takes S-Video in and outputs VGA out [21]. The Hardware LRF black box takes camera link in and camera link out at a 1004*1004 resolution at approximate 50 frames per second. The hardware accelerated lucky region fusion black box video data format and image-processing technique makes it a cutting edge, unique product.

Chapter 3

HIGH SPEED SMART CAMERA

3.1 FastVision FastCamera13

The high-speed smart camera that was chosen to accelerate the LRF algorithm is FastVision's FastCamera13. The FastCamera13 utilizes a Micron MI-MV13 1.3 megapixel CMOS image sensor which sends ten pixels per clock at a66-megahertz clock rate, delivering frames at up to 500 frames per second at the camera's full 1280 x 1024 resolution. The Xilinx Virtex II FPGA then receives the high-speed data for processing.

3.2 Verilog Lucky Region Fusion

The first processing step of the LRF algorithm is edge detection. In this case a relatively simple operator, Sobel Edge Detection was used. It works by computing the derivatives of the intensity signal in each of the x and y directions. These are the places where the intensity is changing most rapidly. The Sobel method approximates the derivative in each direction for each pixel by a process called convolution. This involves adding the pixel and its eight nearest neighbors, each multiplied by a coefficient. The coefficients are represented in a 3*3 convolution mask. The derivative image is computed by centering each of the convolution masks over successive pixels in the original image. Then multiplying the coefficient in each mask by the intensity value of the underlying pixel and sum the nine products together to form two partial derivatives, D_X and D_Y for the derivative image. Ideally, Sobel Edge Detection would

compute the magnitude of the derivative image pixel as $|D| = \sqrt{(D_X^2 + D_Y^2)}$. However, since the algorithm searches for the maxima and minima in the magnitude and computing the square and square-root functions would require a lot of hardware resources, a sufficient approximation is $|D| = |D_X| + |D_Y|$ The last part of the Sobel Edge Detection algorithm is to repeatedly compute the approximate magnitude for each pixel position. Note that the pixels around the edge of the image do not have a complete set of neighboring pixels, so we need to treat them separately [13]. Once this step is complete, the "lucky" lines (not regions) are found.

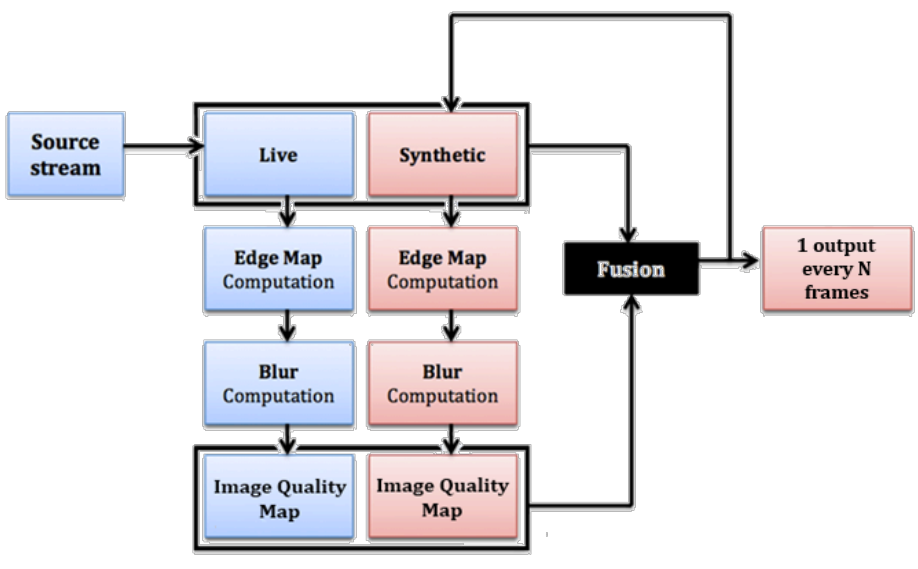

Figure 3.1: 1st Generation Block Diagram

The second step in the LRF process is to take the image resulting from the Sobel edge detection and blur it to determine the original image's lucky-regions. The blur is computed by averaging over a one-dimensional 15-pixel blur radius, first horizontally and then vertically. Once the blur is computed the image is considered to be the original image's Image Quality Map (IQM). After the IQM is created for both the live and synthetic image, the two IQMs are compared pixel by pixel to create a binary mask. Each "pixel" in this binary map has a value of one if the live image's pixel is greater than the synthetic image's corresponding pixel, and is a zero otherwise. This binary mask is then blurred using a five by five averaging blur to create a gradient "Keep" Map. This Keep Map determines what proportion of each new synthetic pixel will be from the newest, live image and what proportion will be from the previous synthetic image. The synthetic frames are stored in SRAM, and one synthetic frame is output once every 30 frames. After the LRF image processing is complete, the output data is sent through camera link to a frame grabber and finally displayed on a computer monitor.

3.3 Results

Figure 3.2 shows the result of the Lucky-Region Fusion algorithm successfully implemented in hardware. Note that in the experiment in Figure 3.2 was taken with no atmospheric turbulence present, so the Keep Map contains mainly "white" pixels, meaning no turbulence is being removed during the experiment and the image quality can't improve further.

Figure 3.2: 1st Generation Results. Top left is original image. Top right is image after Sobel Edge Detection. Bottom left is Image Quality Map. Bottom right is Keep Map.

The initial result of LRF running in real-time on the high-speed smart camera is groundbreaking and serves as a proof of concept. Nevertheless, to make the entire system more practical improvements were necessary. For instance, the Virtex II FPGA was released in 2002 and 2,160 Kbits of BRAM space. To compensate for the limited size of the Virtex II FPGA, important features were left out of the original implementation. Figure 3.3 shows an example of one of the limitations of using the FastCamera13 system. Since only one full frame could be stored at a time, the synthetic image calculation could only combine independent sets of N frames together, where N was typically set to be 30. This memory limitation in turn limited the synthetic output to 1 frame for every 30 input frames.

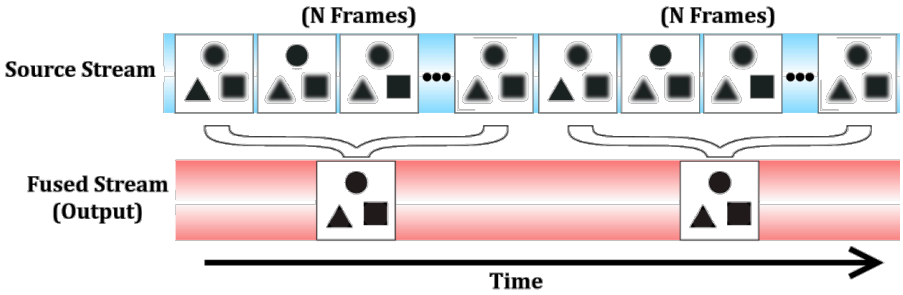

Figure 3.3: Synthetic Frames 1st Generation Output Example. The smart camera's frame storage space was limited to a single frame, making the output frame rate 1/Nth the input frame rate.

Chapter 4

BLACK BOX

4.1 Black Box

Figure 4.1 shows a system diagram of the “black box”. The initial experimental platform for the black box consists of a Xilinx ML605 evaluation board (containing a Virtex 6 FPGA) and a Toyon Boccaccio FPGA Mezzanine Card (FMC) camera link card. The main benefit of switching the hardware acceleration to the black box approach is that it removes any camera dependency, allowing the creation of an architecture capable of receiving and outputting video data in any base camera link configuration. Additionally, since the processor hardware is independent from the camera hardware, the number of choices for FPGA is greatly increased. The Virtex 6 FPGA used in the second implementation of the LRF algorithm 14,976 Kbits of BRAM blocks. This is 7 times the BRAM capacity that the Virtex II FPGA on the FastCamera13.

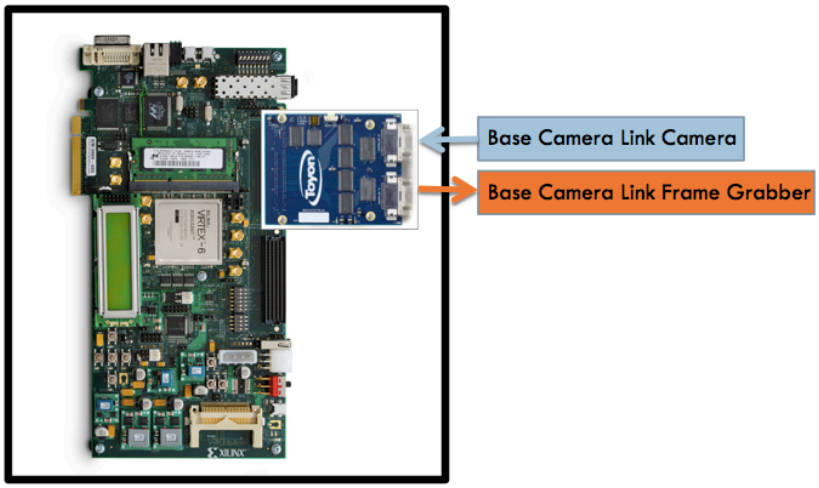

Figure 4.1: "Black box" System. Toyon Boccaccio camera link card is connected to ML605 FMC expansion port. The Boccaccio FMC card receives and transmits camera link data. ML605 processes the data.

Figure 4.2 shows the 2nd generation system diagram. This Black Box implementation can support an input/output frame rate of up to about 100 fps for 1280 x 1024 image, with intermediate synthetic frames being stored in the ML605's DDR3 SDRAM. The Bocccaccio card's 100 frames per second maximum frame rate would have limited the original LRF implementation to fusing no more than 5 images per synthetic image in order to output at a semi-reasonable video rate of 20fps. Fortunately, this frame rate is not a problem for the expanded version of the LRF algorithm. The ML605 is capable of both storing and computing at least 30 synthetic frames in parallel, as seen in Figure 4.3. This increased capacity and processing ability benefits the LRF algorithm in two related ways. First, it means that synthetic frames are no longer fusions of independent sets of N frames but rather a sort of "sliding" fusion of successive groups of N frames, as shown in Figure 4.4. Second, and perhaps more importantly, it means that the LRF unit will be capable of outputting synthetic frames at the same rate it receives new frames. A high-speed camera will no longer be required just to achieve bare minimum output video rate when N is set to 30.

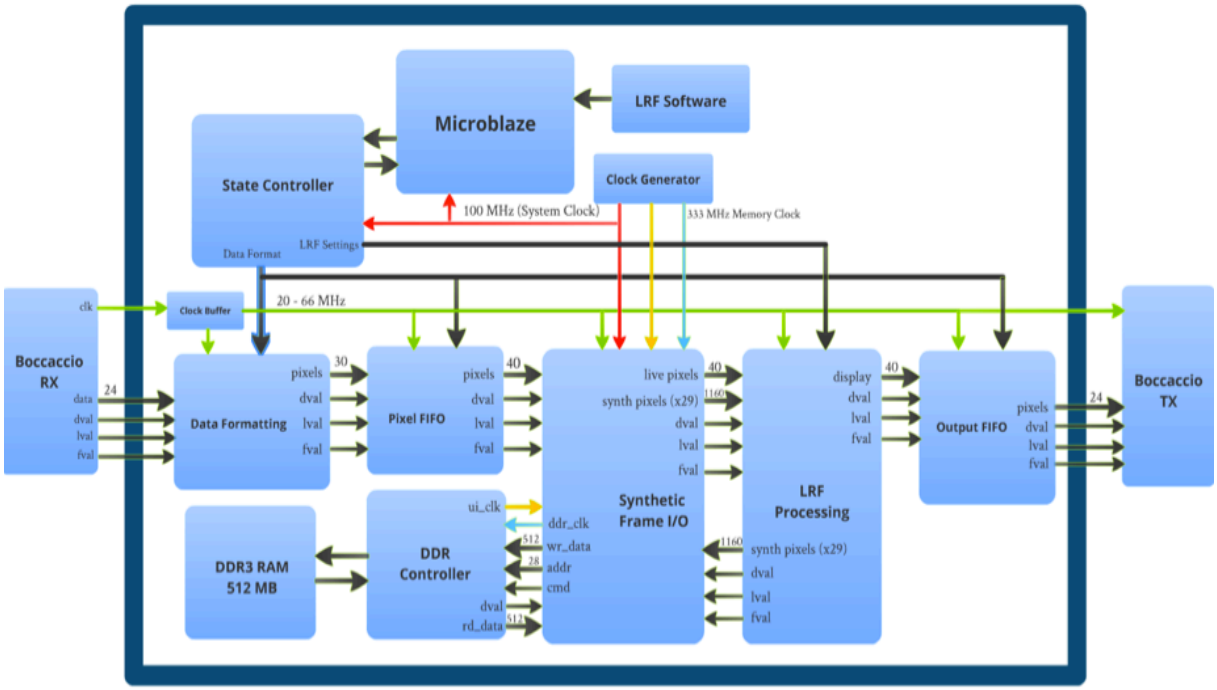

Figure 4.2: 2nd Generation System Diagram

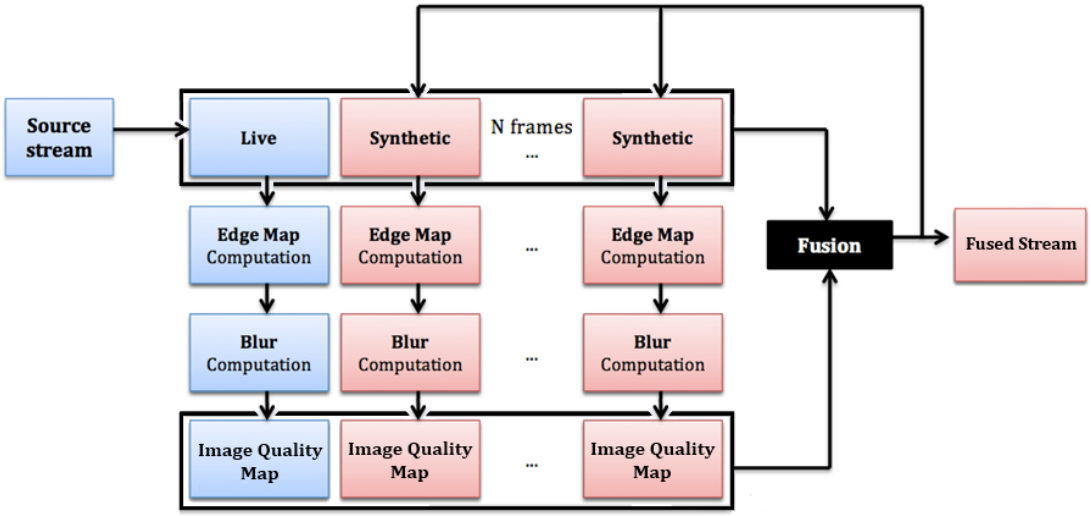

Figure 4.3: 2nd Generation Block Diagram

As seen in Figure 4.2, the second-generation design also includes a MicroBlaze soft processor core for controlling four settings via a serial connection with a computer terminal. The first is a selectable number of input frames per

synthetic image (N) between 1 and 30. The second is the display mode or output format of the data. The third is the camera input format. The final and most important changeable setting is the blur kernel radius, selectable from 1 to 50 pixels. The fusion kernel radius is the most critical fusion parameter of LRF, as it allows for the algorithm to compensate for different levels of severity in atmospheric turbulence [8]. The inclusion of a MicroBlaze processor also allows us to create a "hybrid" implementation with both sequential and parallel processing elements. While the LRF algorithm's relative simplicity lends itself to hardware implementation, this hybridization will be needed when more complicated video processing algorithms like local jitter minimization are added to the LRF module in future system designs.

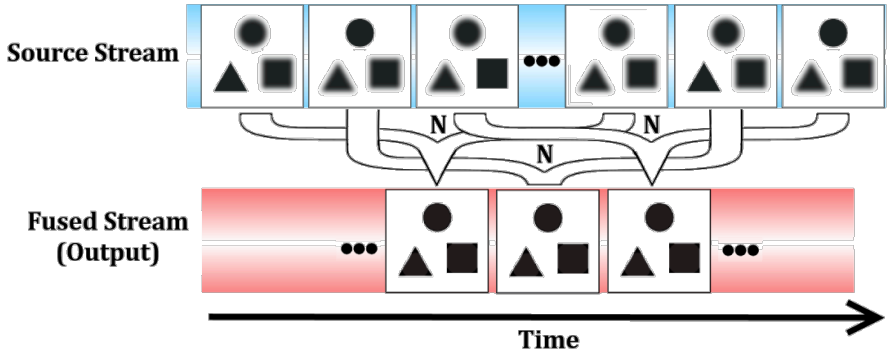

Figure 4.4: Synthetic Frames 2nd Generation Output Example. The increased number of logic gates, frame storage capacity, and frame storage transfer rate on the ML605 increases the maximum number of synthetic frames that can be calculated at the same time to around 30.

Chapter 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

Two different implementations of a hardware accelerated lucky-region fusion algorithm were presented, a high-speed smart camera and a "Black Box" implementation. The 1st generation served as a proof of concept of LRF running in hardware. The 2nd generation is a practical product that improves the overall image quality of an image distorted by atmospheric turbulence. Both generations describe real-time image processing.

5.2 Future Work

Future work will add to the LRF algorithm to compensate for jitter removal and creating another implementation that can operate with a color camera link camera. Since the LRF algorithm has a performance limit, augmenting the current algorithm with another image processing techniques such as deconvolution may improve the performance limit. General movement causes problems for the LRF algorithm. Removing motion artifacts, objects that appear in the image for a brief moment and disappear but remain in the fused image, would improve the algorithm. Also, accounting for moving target crossing through an image is desired.

REFERENCES

[1] Baumgartner, Dustin D., and Bruce J. Schachter. "Improving FLIR ATR performance in a turbulent atmosphere with a moving platform." In *Proc. of SPIE Vol*, vol. 8391, pp. 839103-1. (2012).

[2] Oreifej, Omar, Xin Li, and Mubarak Shah. "Simultaneous video stabilization and moving object detection in turbulence." (2013): 1-1.

[3] Zhu, Xiang, and Peyman Milanfar. "Image reconstruction from videos distorted by atmospheric turbulence." In *IS&T/SPIE Electronic Imaging*, pp. 75430S-75430S. International Society for Optics and Photonics, (2010).

[4] Zhu, Xiang, and Peyman Milanfar. "Removing atmospheric turbulence via space-invariant deconvolution." (2013): 1-1.

[5] van Eekeren, Adam WM, Klamer Schutte, Judith Dijk, Piet BW Schwering, Miranda van Iersel, and Niek J. Doelman. "Turbulence compensation: an overview." In *SPIE Defense, Security, and Sensing*, pp. 83550Q-83550Q. International Society for Optics and Photonics, (2012).

[6] Carhart, G. W., and M. A. Vorontsov. "Synthetic imaging: nonadaptive anisoplanatic image correction in atmospheric turbulence." *Optics letters* 23, no. 10 (1998): 745-747.

[7] Vorontsov, Mikhail A., and Gary W. Carhart. "Anisoplanatic imaging through turbulent media: image recovery by local information fusion from a set of short-exposure images." *JOSA A* 18, no. 6 (2001): 1312-1324.

[8] Aubailly, Mathieu, Mikhail A. Vorontsov, Gary W. Carhart, and Michael T. Valley. "Automated video enhancement from a stream of atmospherically-distorted images: the lucky-region fusion approach." In *SPIE Optical Engineering+ Applications*, pp. 74630C-74630C. International Society for Optics and Photonics, 2009.

[9] Gilles, Jérôme, and Stanley Osher. "Fried deconvolution." In *SPIE Defense, Security, and Sensing*, pp. 83550G-83550G. International Society for Optics and Photonics, (2012).

[10] Huebner, Claudia S., and Corinne Scheifling. "Software-based Mitigation of Image Degradation due to atmospheric Turbulence." In *Remote Sensing*, pp. 78280N-78280N. International Society for Optics and Photonics, 2010.

[11] Vorontsov, Mikhail A. "Parallel image processing based on an evolution equation with anisotropic gain: integrated optoelectronic architectures." *JOSA A* 16, no. 7 (1999): 1623-1637.

[12] Mao, Yu, and Jérôme Gilles. "Non rigid geometric distortions correction-application to atmospheric turbulence stabilization." *Inverse Problems and Imaging Journal* (2012).

[13] Ashenden, Peter. *Digital Design An Embedded Systems Approach Using VHDL*. Burlington: Morgan Kaufmann, 2008. pp. 400-425.

[14] Fishbain, Barak, Leonid P. Yaroslavsky, and Ianir A. Ideses. "Real-time stabilization of long range observation system turbulent video." *Journal of Real-Time Image Processing* 2, no. 1 (2007): 11-22.

[15] Rodríguez-Ramos, Luis F., A. Alonso, Fernando Gago, José V. Gigante, Guillermo Herrera, and Teodora Viera. "Adaptive optics real-time control using FPGA." In *Field Programmable Logic and Applications, 2006. FPL'06. International Conference on*, pp. 1-6. IEEE, 2006.

[16] Ramos, LF Rodríguez, JJ Piqueras Meseguer, Y. Martin Hernando, A. Oscoz, and R. Rebolo. "Real-time lucky imaging in FastCam project." In *Astronomical Telescopes and Instrumentation: Synergies Between Ground and Space*, pp. 701449-701449. International Society for Optics and Photonics, 2008.

[17] Magdaleno, Eduardo, Manuel Rodríguez, José Manuel Rodríguez-Ramos, and Alejandro Ayala. "Modal Fourier Wavefront Reconstruction Using FPGA Technology." *Micro. Nanosyst* 1 (2009): 72-82.

[18] Rodriguez-Ramos, Luis F., Teodora Viera, Jose V. Gigante, Fernando Gago, Guillermo Herrera, Angel Alonso, and Nicolas Descharmes. "FPGA adaptive optics system test bench." In *Optics & Photonics 2005*, pp. 59030D-59030D. International Society for Optics and Photonics, 2005.

[19] Gavel, Donald, Marc Reinig, and Carlos Cabrera. "Fast hardware implementation of tomography for multi-guidestar adaptive optics." In *Optics & Photonics 2005*, pp. 59030F-59030F. International Society for Optics and Photonics, 2005.

[20] Venugopal, V., K. Richards, S. Barden, T. Rimmele, S. Gregory, and L. Johnson. "Accelerating Real-time processing of the ATST Adaptive Optics System using Coarse-grained Parallel Hardware Architectures." In *International Conference on Engineering of Reconfigurable Systems and Algorithms*, pp. 296-301. 2011.

[21] Bodnar, Michael R., Petersen F. Curt, Fernando E. Ortiz, Carmen J. Carrano, and Eric J. Kelmelis. "An embedded processor for real-time atmoshperic compensation." In *SPIE Defense, Security, and Sensing*, pp. 734102-734102. International Society for Optics and Photonics, 2009.

[22] Droege, Douglas R., Russell C. Hardie, Brian S. Allen, Alexander J. Dapore, and Jon C. Blevins. "A real-time atmospheric turbulence mitigation and super-resolution solution for infrared imaging systems." In *SPIE Defense, Security, and Sensing*, pp. 83550R-83550R. International Society for Optics and Photonics, 2012.