

# Hardware Design of correlation Filters for Target Detection

Naeem Akbar<sup>1</sup>, Sara Tehsin<sup>1</sup>, Haseeb ur Rehman<sup>1</sup>, Saad Rehman<sup>1</sup> and Rupert Young<sup>2</sup>

<sup>1</sup> National University of Sciences and Technology, Islamabad, Pakistan

<sup>2</sup> University of Sussex, Brighton, UK

**Abstract**— Correlation filters have been implemented in software and have proven very effective for automatic target detection, biometric verification and security applications. In this paper, these filters are implemented in hardware keeping in view their importance in real time applications. Hardware implementation results are placed in comparison with results generated through software. These results vary by as little as  $10^{-4}$  which is demonstrated in the experimental section. The hardware design of these filters is implemented in LabView which can be subsequently employed in real-time security applications. This design may be expanded for other advanced variants of correlation filters in future work.

**Key words:** Hardware design, Correlation filter, Automatic target recognition.

## I. INTRODUCTION

Human and machine vision are often analogous but identification of targeted objects by machines has been a vital and essential requirement of image processing as compared to humans when there is distortion and noise within cluttered backgrounds. By closely relating the response of artificial sensors to human vision perception, better performance can be developed. Thus sensors which are chosen for aim detection should follow the best approximation of human senses no matter what variations and occlusions are present in test images.

Correlation filters, which have been successfully used in the past decades for better object and shape recognition are designed for implementation in software or hardware. These filters may accommodate a wide range of distortions in which they can achieve reliable object recognition. These filters can be implemented in hardware which can be effectively used for real time applications. The advanced variants of correlation filters have been used for biomedical, defense and security applications. Kumar *et al.* proposed an Optimal Trade-off Maximum Average Correlation Height (OT-MACH) filter which minimized the energy function. Different correlation filters can be implemented by changing the value of the Optimal Trade-off (OT) parameters [1]. Its advanced variants are resistant to distortion, noise and background clutter. Another interesting attribute of the correlation filter is that it gives pronounced peaks for test images by simple computations.

There are several other solutions for the hardware implementation of the filter but these are less flexible and cost effective, such as hardwired solutions. Media-processors include new Very-Long-Instruction-Word (VLIW) and programmable super-scalar Digital Signal Processors (DSPs) embedded with heavy-duty computational blocks / cores. These are highly efficient, low cost, flexible processors best suited for the hardware design of correlation filters.

In this paper, the hardware design of a correlation filter has been implemented in the National Instruments LV 8.6 which is a programming environment (graphical and dataflow based) for embedded design. LV combines a Single Cycle Time Loop (SCTL) block feature which is a special loop of the LV FPGA Timed Loop structure. This loop computes all functions inside a block when it uses with an FPGA source in *one specified clock cycle*. SCTL can be used with derived clocks to regularize the loop on speeds different from 40 MHz by default. Timing properties of the Timed Loop cannot be changed when it is used with an FPGA source. The code can be written for conventional sub-systems in a hardware description language (HDL) and incorporated into LV. The optimized architecture of the OT-MACH has been proposed in the LV (FPGA) design environment and put into operation using the architecture of the FPGA. It invokes the LV (FPGA) synthesis tool first followed by mapping the resultant code and engaging the platform specific tools of the targeted FPGA. Xilinx Kintex-7 is the target FPGA used in our experimentation. The National Instruments FlexRio board is our experimental platform; more specifically it comprises a Xilinx Kintex-7 device integrated in conjunction with the host.

This paper is segmented into five sections; after this introduction a literature review is presented in Section II which presents some related work starting with early techniques and reviewing some related implementations. Section III describes the detailed methodology used to implement correlation filters on the FPGA using LV. Section IV illustrates the computed results of the hardware architecture and gives a comparison with software implementation in MATLAB. Section V gives brief conclusions.

## II. BACKGROUND REVIEW

Young *et al.* proposed a hybrid optical correlator in which the input image is Fourier transformed digitally at video rate with a digital signal processor [2]. The input data is combined with a digital template and then loaded onto a high frame rate Spatial Light Modulator (SLM). The output of the optical Fourier transform then implements a correlation between the input image and reference image. This hybrid hardware approach of optical correlator gave high speed results as compared to previous implementations.

Various optimized approaches for 1-D and 2-D Fast Fourier Transformation (FFT) implementation, used for implementing correlation filters, have been proposed in the past. The high level framework including 1D-FFT and 2D-FFT FPGA implementations for instantaneous applications was proposed by Uzun *et al.* [3]. The parallel approach in 2D-FFT calculation is achieved by numerous 1D-FFT processors which share additional external memory.

Kumar *et al.* presented a survey paper of all variants of correlation filters and their mathematical expressions [4]. A hard problem of pattern recognition is to detect a targeted item in the presence of distortion in orientation, position and scale, especially in cluttered backgrounds for which a methodology was proposed by Rehman *et al.* in which a logarithmic correlation filter is combined with a band-pass difference of Gaussian (DOG) filter [5]. The resultant method was capable of creating invariance for object identification despite several forms of distortion.

Research work on a method of combining of two existing techniques was introduced by Bone *et al.* which was capable of producing invariance to several kinds of rotation and scale distortions [6]. In this procedure, the Maximum Average Correlation Height (MACH) filter was used to provide orientation invariance tolerance to the noisy backgrounds along with the logarithmic mapping algorithm which is invariant to in-plane rotation and scale changes and was implemented in the form of a memory shift.

Recently Awan *et al.* proposed a combined framework of the EEMACH filter and the DOG filter to develop a system which can give a pronounced peak in the presence of background clutter and distortions [7]. The proposed filter gave enhanced target recognition results leading to a higher percentage of correct automated decisions in all situations as compared to previously proposed correlation filters.

The values of the optimal trade-off correlation parameters can be selected randomly on the basis of experiments. These values are then optimized depending upon the dataset and application through an hierarchal particle swarm optimization algorithm by Tehsin *et al.* [8].

Various variants of correlation filters with enhanced performance have been proposed by many researchers in the past few decades [9-13]. These filters are efficiently used in many applications such as biomedical, defense and security applications. By keeping in view its effectiveness, the OT-MACH filter has been implemented by us in hardware for the usage in real time applications.

## III. METHODOLOGY

An optimal tradeoff approach proposed by Kumar, is used to minimize the following energy function [4]:

$$E(h) = \alpha(ONV) + \beta(ACE) + \gamma(ASM) - \delta(ACH) = \alpha h^+ C h + \beta h^+ D_x h + \gamma h^+ S_x h - \delta |h^T m_x|$$

The resultant optimal trade-off MACH filter given as [1]:

$$h = \frac{m_x^*}{\alpha C + \beta D_x + \gamma S_x}$$

Here  $\alpha, \beta$  and  $\gamma$  are optimal trade-off parameters which are used to implement three different variants of the correlation filter,  $m_x$  is the average of training data  $x_1, x_2, \dots, x_N$ ,  $X_i$  is a diagonal matrix of the training images and  $D_x$  is a diagonal average power spectral density of the training image. [1]

$$D_x = \frac{1}{N} \sum_{i=1}^N X_i^* X_i$$

$S_x$  is a diagonal matrix measuring the similarity of training data to the class mean in the frequency domain [1]:

$$S_x = \frac{1}{N} \sum_{i=1}^N (X_i - m_x)^* (X_i - m_x)$$

and  $C$  is spectral density matrix of additive input white noise.

In this paper we implement the hardware design of the MACH filter which gives a correlation peak against a targeted object. It maximizes the performance measure called Average Correlation Height (ACH) while minimizing Average Similarity Measures (ASM). The value of  $\alpha=0.01$ ,  $\beta=0.1$  and  $\gamma=0.3$  has been chosen for the MACH filter implementation [6].

Correlation filters are implemented in frequency domain as this involves a simple frequency domain multiplication. For the 2D-FFT we accelerate a 1D-FFT algorithm in the Lab View (LV) FPGA using an SCTL block. The SCTL is a particular loop of the LV FPGA Timed Loop structure. This computes every function inside the block by one selected clock cycle when used with an FPGA. SCTL can be used with derived clocks to regularize the loop on speeds different from 40 MHz by default. In the FPGA target one cannot dynamically change timing properties of the Timed Loop. Other than SCTL employment, to execute one iteration takes a minimum of 3 clock cycles for one *while loop* in FPGA VI. It executes due to the enable chain for FPGA VI compilation. The enable chain is applied for assurance of dataflow when the FPGA VI is compiled into a bit file. The performance of SCTL in FPGA VI varies depending on what is inside in the loop. Logic is implemented in combinatorial hardware so the FPGA configuration for code generation uses less resources. Instead of doing addition, the result is saved, and then multiplied and again the result is saved. The SCTL can do this in 1 x clock cycle and the step involved in storing the intermediate results is avoided. FPGA resources are conserved as no flip flop is required in between the operations for storing the result.

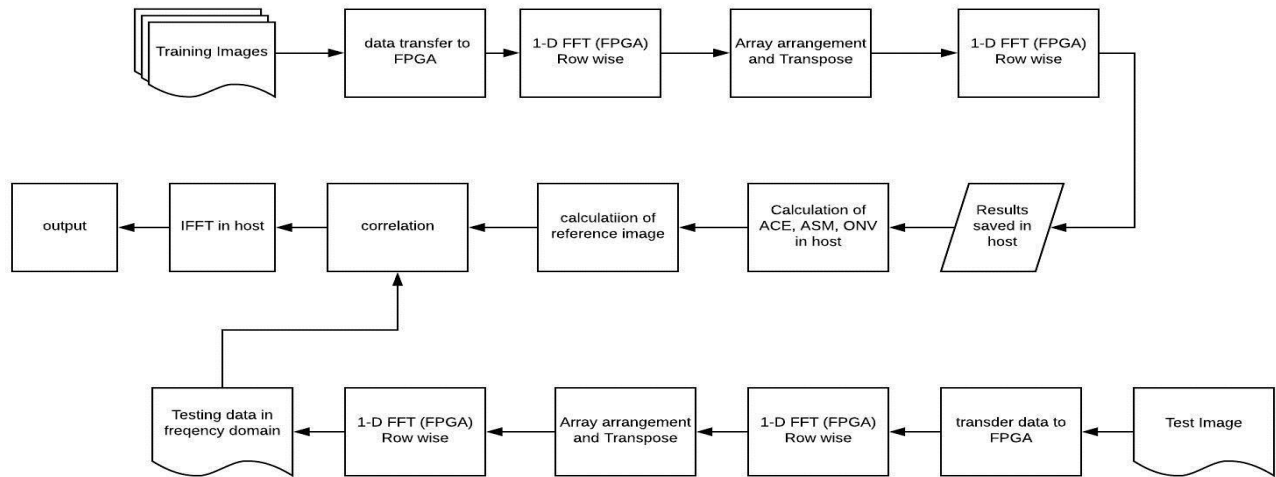


Figure 1: Block diagram of hardware design of MACH

In figure 1, the complete block diagram of the hardware design of the MACH is given. The FFT of the images is implemented on an FPGA in LV and the remaining implementation is done on the host of LV. In host mode, LV uses the building blocks which run on the system's processor. In our design, we are using the FPGA for the FFT as it takes more computations. Thus a separate processor has been used and once we compute the FFT for the training images we store these on the host for further experimentation which reduces computations.

#### IV. EXPERIMENTAL RESULTS

For the hardware implementation on FPGAs, the MACH filter was trained from 0-40 degrees out of plane rotated images as shown in figure 2. Out of plane rotation is one of the important factors in tracking of any object. The data set of images of the Amsterdam image library that contain out of plane rotated images of size 16x16 resolution were used [14].



Figure 2: Amsterdam image dataset [14]

Details of FPGA hardware used for the complete implementation after place and routing is provided in table 1 below.

Only 1% DSP 48s blocks, 18% of block RAMs, 56.4 % of slice LUTs and 22.1% of slice registers are used of the total hardware resources. All timing requirements are met; it takes 324  $\mu$ sec for computation for an image of 16x16 pixels. However, the FPGA VI code is generic for some of the latest types of NI LV FPGAs like the 7975R and 7972R etc.

Device Utilization	Used	Total	Percent
Slice Registers	112549	508400	22.1
Slice LUTs	143286	254200	56.4
Block RAMs	145	795	18.2
DSP48a	16	1540	1.0

Table 1: FPGA hardware usage

The implementation of a sub block in LV is given in figure 3 in which the FFT is calculated on an FPGA of LV and then data transferred back to the host for further computation.

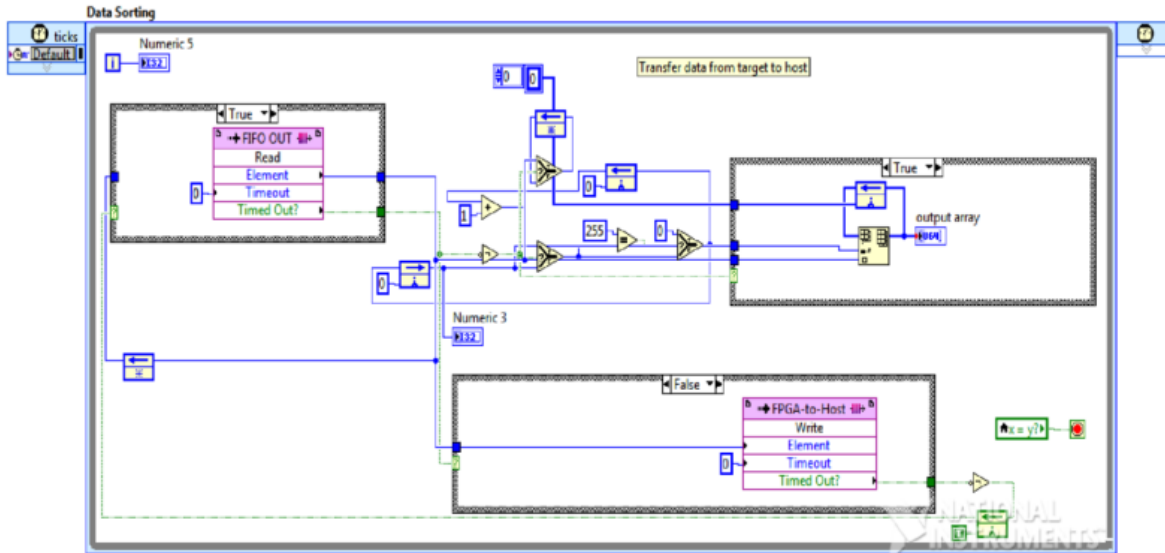


Figure 3: Sub section of LV implementation

The LV hardware design has been tested on 15, 25 and 40 out of plane angles and the same testing is repeated using MATLAB through software implementation. The results generated from MATALB are similar to the LV results with only a difference is at  $10^{-4}$  which is very minor. Table 2 illustrates correlation results the first 16 values in the 1<sup>st</sup> column, tested on 15, 25 and 40 degree out of plane angles.

15 degree out of plane rotation		Difference	25 degree out of plane rotation		Difference
LV result	MATLAB result		LV result	MATLAB result	
9.70307 + 0.0i	9.70306 + 0.0i	0.00001+0.0i	9.58505 +0.0i	9.58504+0.0i	0.00001+0.0i
5.20207 + 0.814591i	5.20206 + 0.814591i	0.00001+0.0i	4.99109+1.28628i	4.99109+1.28628i	0.0+0.0i
2.04243 + 0.61897i	2.04244 + 0.61897i	0.00001+0.0i	1.85900+0.81521i	1.85901+0.81521i	0.00001+0.0i
0.91842+ 0.35864i	0.91842+ 0.35864i	0.0+0.0i	0.95973+0.54588i	0.95974+0.54588i	0.00001+0.0i
0.0868+ 0.08171i	0.08697+ 0.08171i	0.00001+0.0i	0.15086+0.07848i	0.15087+0.07848i	0.00001+0.0i
0.04847+ 0.02104i	0.04847+ 0.02104i	0.0+0.0i	0.01076+0.01825i	0.01076+0.01825i	0.0+0.0i
0.01512+ 0.03416i	0.01511+ 0.03416i	0.00001+0.0i	0.00540+0.02879i	0.00541+0.02879i	0.00001+0.0i
-0.03042+ 0.00663i	-0.03042+ 0.00663i	0.0+0.0i	-0.0056+0.01008i	-0.0057+0.01008i	0.00001+0.0i
0.00266 + 0.0i	0.00266 + 0.0i	0.0+0.0i	0.00232+ 0.0i	0.00232+0.0i	0.0+0.0i
-0.03042- 0.00663i	-0.03042- 0.00663i	0.0+0.0i	-0.00573-0.01008i	-0.00574-0.01008i	0.00001+0.0i
0.01510 - 0.03416i	0.01511 - 0.03416i	0.00001+0.0i	0.00541- 0.02879i	0.00541-0.02879i	0.0+0.0i
0.04851- 0.02104i	0.0485- 0.02104i	0.00001+0.0i	0.01075- 0.01825i	0.01076-0.01825i	0.00001+0.0i

Table 2: Correlation Results Between Reference Image and Test Image

There is very small difference between hardware and software generated results. The results after the inverse FFT are shown in table 3 in which the first 16 values are given. The implementation in LV is same as the MATLAB implementation which can be used efficiently for a real time environment.

15 degree out of plane rotation		Difference	25 degree out of plane rotation		Difference
LV result	MATLAB result		LV result	MATLAB result	
-0.002924	-0.002934	0.00001	0.000252	0.000252	0.00001
-0.004113	-0.004123	0.00001	1.397551e-06	1.397551e-06	0.0
0.005661	0.005671	0.00001	0.012580	0.012590	0.00001
0.015848	0.015848	0.0	0.025365	0.025375	0.00001
0.024411	0.024421	0.00001	0.030455	0.030465	0.00001
0.033787	0.033787	0.0	0.038149	0.038149	0.0
0.045211	0.045201	0.00001	0.047551	0.047561	0.00001
0.053298	0.053298	0.0	0.055156	0.055166	0.00001

0.057388	0.057388	0.0	0.057718	0.057718	0.0
0.052361	0.052361	0.0	0.045303	0.045313	0.00001
0.034122	0.034112	0.00001	0.024030	0.024030	0.0
0.020139	0.020129	0.00001	0.015532	0.015542	0.00001

Table 3: Correlation Results Between Reference Image and Test Image

The correlation peak of 25 out of plane rotated test images generated by MATLAB, is given in figure 4. The COPI value of the simple MACH filter for the test image is 0.0016.

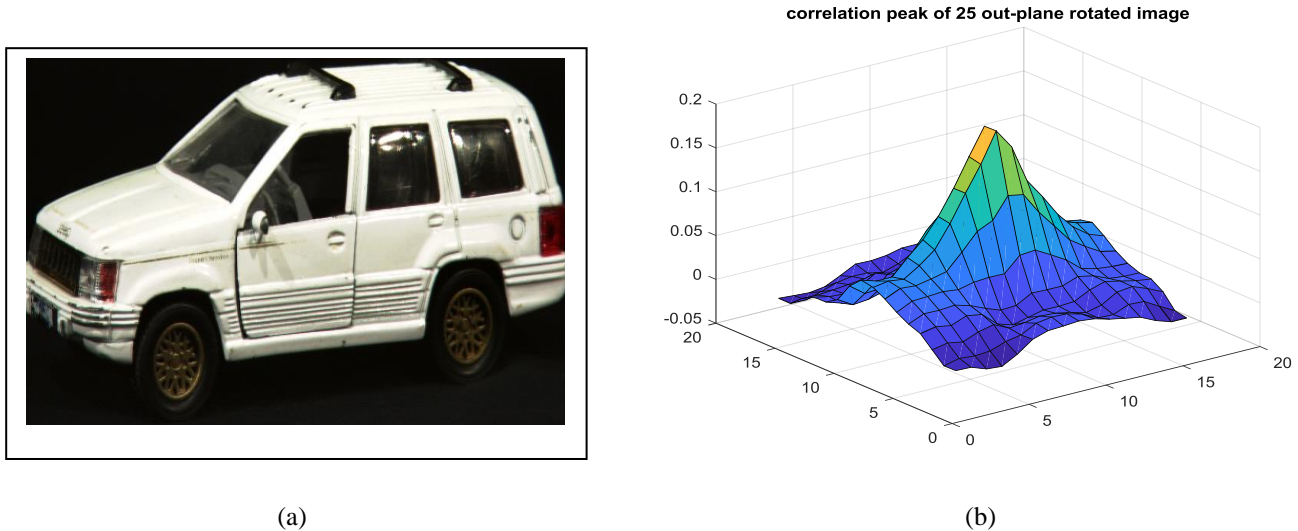


Figure 4: (a) Test Data (b) Correlation Output of Test Data

### V. Conclusion

Hardware implementation of software is the ultimate goal for most developmental work and research. However, practical implementation of any study or research has its own complications and variations, which need to be addressed during the development phase. Many synthesis applications have been developed which are being used for FPGA implementations. LabView is one of these which is very accurate and reliable. Correlation filter implementation in LabView, after a lot of effort, showed excellent results, which opened different avenues for real time implementation of correlation filters in the fields of medical, security and military applications.

### REFERENCES

1. Zhou, H., & Chao, T. H. (1999, March). MACH filter synthesizing for detecting targets in cluttered environment for grayscale optical correlator. In *Optical pattern recognition X* (Vol. 3715, pp. 394-399). International Society for Optics and Photonics.
2. Young, R. C., Claret-Tournier, F., Li, G., Birch, P., Budgett, D. M., Koukoulas, T., & Chatwin, C. R. (2000, March). Hardware implementation details of a hybrid digital/optical correlator system. In *Optical Pattern Recognition XI* (Vol. 4043, pp. 25-40). International Society for Optics and Photonics.
3. Uzun, I. S., Amira, A., & Bouridane, A. (2005). FPGA implementations of fast Fourier transforms for real-time signal and image processing. *IEE Proceedings-Vision, Image and Signal Processing*, 152(3), 283-296.

4. Kumar, B. V., Fernandez, J. A., Rodriguez, A., & Boddeti, V. N. (2014, May). Recent advances in correlation filter theory and application. In *Optical Pattern Recognition XXV* (Vol. 9094, p. 909404). International Society for Optics and Photonics.
5. Rehman, S., Young, R., Birch, P., Chatwin, C., & Kypraios, I. (2005). Fully scale and in-plane invariant synthetic discriminant function bandpass difference of gaussian composite filter for object recognition and detection in still images. *Journal of Theoretical and Applied Information Technology*, 5(2), 232-241
6. Bone, P., Young, R. C., & Chatwin, C. R. (2006). Position-, rotation-, scale-, and orientation-invariant multiple object recognition from cluttered scenes. *Optical Engineering*, 45(7), 077203.
7. Awan, A. B., Rehman, S., & Bakhshi, A. D. (2018). Composite filtering strategy for improving distortion invariance in object recognition. *IET Image Processing*.
8. Tehsin, S., Rehman, S., Saeed, M. O. B., Riaz, F., Hassan, A., Abbas, M., ... & Alam, M. S. (2017). Self-organizing hierarchical particle swarm optimization of correlation filters for object recognition. *IEEE Access*, 5, 24495-24502.
9. Awan, A. B., Rehman, S., & Latif, S. (2014, November). Synthesis of an adaptive CPR filter for identification of vehicle make & type. In *Software Engineering Conference (NSEC), 2014 National*(pp. 25-29). IEEE.
10. Tehsin, S., Rehman, S., Awan, A. B., Chaudry, Q., Abbas, M., Young, R., & Asif, A. (2016, April). Improved maximum average correlation height filter with adaptive log base selection for object recognition. In *Optical Pattern Recognition XXVII*(Vol. 9845, p. 984506). International Society for Optics and Photonics.
11. Rehman, S., Bilal, A., Javed, Y., Amin, S., & Young, R. (2013). Logarithmically pre-processed EMACH filter for enhanced performance in target recognition. *Arabian Journal for Science and Engineering*, 38(11), 3005-3017
12. Rehman, S., Riaz, F., Hassan, A., Liaquat, M., & Young, R. (2015, April). Human detection in sensitive security areas through recognition of omega shapes using MACH filters. In *Optical Pattern Recognition XXVI* (Vol. 9477, p. 947708). International Society for Optics and Photonics.
13. Bone, P., Kypraios, I. I., Young, R. C., & Chatwin, C. R. (2005, February). Fully invariant object recognition in cluttered scenes. In *Information Technologies 2004* (Vol. 5822, pp. 1-13). International Society for Optics and Photonics.
14. Amsterdam library of images: <http://aloi.science.uva.nl/>