Journal of Theoretical and Applied Information Technology
30th April 2012. Vol. 38 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

VLSI IMPLEMENTATION OF AN EFFICIENT TEMPLATE MATCHING ARCHITECTURE BASED ON FEATURE EXTRACTION

[1]T. MAHALAKSHMI, [2]R. MUTHAIAH

[1]P.G. Scholar, VLSI Design, SASTRA University, Thanjavur

[2]Assoc. Prof., School of computing, SASTRA University, Thanjavur

E-mail: [1]mahalakedu@gmail.com, [2]sjamuthaiah@core.sastra.edu

ABSTRACT

This paper presents a spectral architecture for template matching that combines edge detection, to find the similarity between input image and template. Generally template matching algorithms based on cross correlation suffers either from computational complexity or larger detection time. This architecture overcomes both problems and makes the system more reliable. Experiment results show the accuracy of detection and computational time. Since this model was developed on Matlab system generator and synthesized using Xilinx platform, it can be easily implemented on FPGA. Resource utilization can be done, which speeds the computation when the system is implemented on FPGA. This makes the architecture more comfortable for real time applications.

Keywords: *Spectral Architecture, Edge detection, Cross correlation, Matlab System generator, Xilinx*

1. INTRODUCTION

Template Matching (TM) in image processing is finding a target among various objects. It is widely used in many applications, such as tracking and monitoring in military applications, satellite image monitoring, image registration and medical image combination for medical applications. Matching is commonly done in many methods and still many researches are working to increase accuracy and reduce computation time [1-3].

Generally, Mean Absolute Difference, Mean Square Error, Luminance similarity, Correlation, Contrast similarity and Hausdorff distance are used to find the similarity [9, 10]. Number of works has been proposed to detect the object along with feature extraction such as edges, regions and corners and then to locate the target by combining the features but these works have high computational complexity [4-8].

Cross correlation is an important tool in image processing which correlates two signals to find the match but the insufficient accuracy and computational complexity makes the method unaffordable for real time applications [11-13]. Correlation can be made through both spatial and spectral domain, it is known that FFT based correlation is quite fast compared to other methods, so we proceed with a spectral architecture which overcomes the limitations mentioned. Edge detection can be done in many ways still in all methods making is done only in horizontal and vertical direction which limits the accuracy of feature extracted from the original image. In our architecture masking is done for all eight directions. This combined architecture implementation can be performed in either software or hardware.

2. PROPOSED ARCHITECTURE FOR TEMPLATE MATCHING

Fig. 1 shows the combined architecture of template matching which contains two major functional blocks. This architecture is designed in such a way that it works for a real time application. The input image in which the target is to be detected is brought inside the model through “from workspace” blocks in Simulink. It loads a variable from Matlab workspace to working model when an m-file is compiled. This m-file includes set of functions; first it reads the input image and converts it to a gray image. The converted gray image variable will be in matrix (two-dimension) form, which cannot be processed by the Xilinx blocks. So

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the two-dimensional array is converted into a one-dimensional array. The variable that contains this one-dimensional array is loaded inside the model. Template image is also processed in the same way and loaded inside the model.

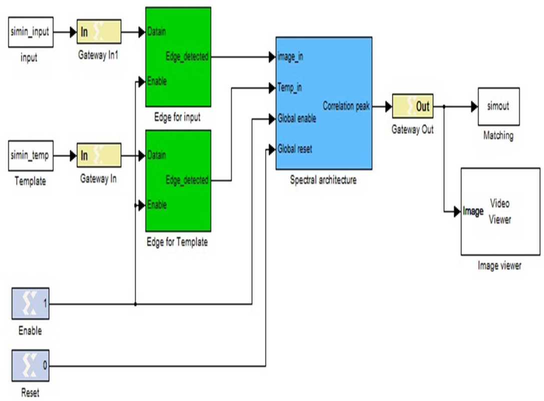


Figure1. Combined Architecture

Edges of both input and template image are extracted by edge detection block which are designed using Xilinx blocks. It cannot directly get the values from Simulink blocks so the values are routed through a gateway. Then both edges are cross-correlated by a spectral architecture to find the match. Correlation peaks will be maximum when there exists complete match and zero when no matching. Else peaks will vary depending on the matching position. Fig. 2 shows the input image and two templates taken for verification.

Figure2. a) Input image b) Template1 c) Template2

3. EDGE DETECTOR

In image processing and computer vision edge detection is a process which attempts to capture the significant properties of objects in image. Edge detector must be more reliable and efficient because the accuracy of template matching depends on it. They are categorized into two, based on their behavioral study.

- Gradient edge detectors or First derivative
- Zero crossing or Second derivative

Gradient edge detectors can be easily designed using Xilinx blocks; since they require only LUT's and ALB's. Sobel, Prewitt and Robert cross comes under first derivative. Suitable type is selected depending on the applications. We consider Prewitt detector for our experiment. Horizontal and Vertical kernel of Prewitt is shown below.

$$\begin{bmatrix}-1 & -1 & -1\\ 0 & 0 & 0\\ 1 & 1 & 1\end{bmatrix}\begin{bmatrix}-1 & 0 & 1\\ -1 & 0 & 1\\ -1 & 0 & 1\end{bmatrix}$$

Horizontal and Vertical kernel for Prewitt

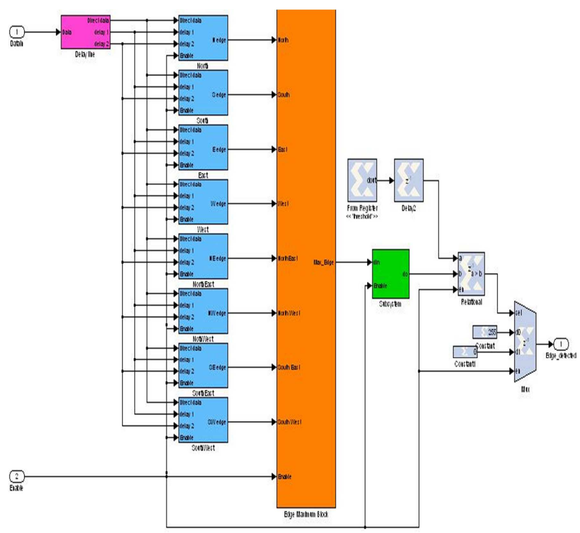


Figure3. Modified Edge detector

Accuracy of edges detected is not sufficient it can be overcome by this modified edge detector shown in Fig. 3. There are four major blocks in it which are described below.

3.1 Delay line

Image pixel values loaded from workspace is fed as input; the pixel values loaded are stored in the memory to delay them. Depth of memory depends on the pixel values in each delay line. The pixel values are then passed to the convolution blocks.

3.2 Convolution block

Instead of convoluting the image pixel values in horizontal and vertical direction; it is done in all eight directions. The kernel of Prewitt is rotated in eight directions as shown in Fig. 4 that is each edge orientation ranges from 0° to 315° at the step of 45°.

Journal of Theoretical and Applied Information Technology
30th April 2012. Vol. 38 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

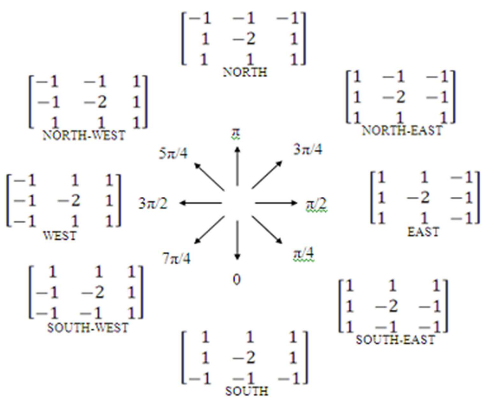


Figure4. Masking orientation in eight directions

Each mask is convoluted with the incoming delayed pixel values and maximum edges in all eight directions are estimated. It is done by multiplying each row of convolution kernel with the input pixel values and added.

3.3 Maximum Estimation

Maximum edge is estimated using a multiplexers and comparators. The inputs for mux are the compared results of edges from all eight directions.

3.4 Threshold unit

Thus the magnitude of edge detected values is fed into this block as input. These types of detectors are susceptible to noise. To overcome a threshold block is added; by doing so noises get suppressed. The threshold value is decided manually to get proper edges.

4. SPECTRAL ARCHITECTURE FOR CORRELATION

Correlation theorem states that the correlation computed in spatial domain is similar to point wise multiplication in the Fourier domain.

$$\text{Cross Correlation} = F^{-1}(F^{*}(T)^{*} F(I))$$

Fig. 5 shows the spectral architecture which directly implements the CC theorem with reduced number of elements. To implement the theorem directly we require two FFT computations and one IFFT computation with complex multiplication.

But our architecture simplifies it by using two temporary memories in which one holds the information of input image at initial stage. These memories are utilized again and again for storing the temporary values at the time of calculation. One memory to place the template pixel values permanently. A complex multiplier is used which performs multiplication between Template and Image. Fourier treatment is done by FFT slices. The operation of slices is controlled by multiplexer. It is programmed to perform FFT operation and IFFT operation to compute the theorem.

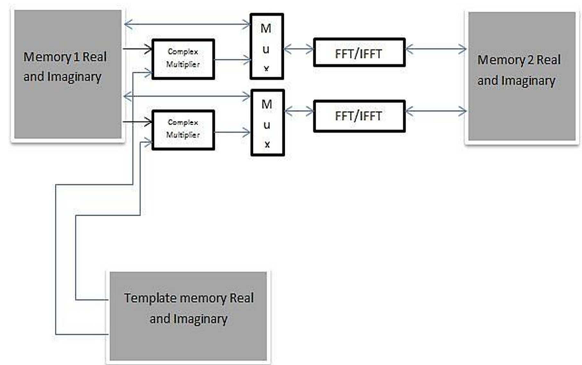


Figure5. Spectral architecture

Two memories Memory 1 and Memory 2 are used which groups two independent memories; one for real part and other for imaginary part. Memory 1 holds the pixel values of input image. Template memory is the permanent memory which holds the conjugate values of template image; it also has separate access for real and imaginary. Depth of memory and number of slices used varies depending on the size of the image used. Independent memories and parallel FFT/IFFT computations are done to speed up the process. The working principle of this architecture is explained below.

1. Conjugate of template value is calculated and stored in Template memory_re and Template memory_img is loaded with zeros.

2. Input pixel values are loaded in memory1_re and memroy1_img is filled with zeros.

3. Row wise FFT computation for elements in memory1 is done and the results are stored in memory2_re and memory2_img.

4. Column wise FFT computation for elements in memory2_re and memory2_img is done and the results are stored in memory1_re and memory1_img.

Journal of Theoretical and Applied Information Technology
30th April 2012. Vol. 38 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Column wise selection is done by a separate control unit. (Now F (I) is done).

5. Values from memory1 and template memory multiplied by a complex multiplier.

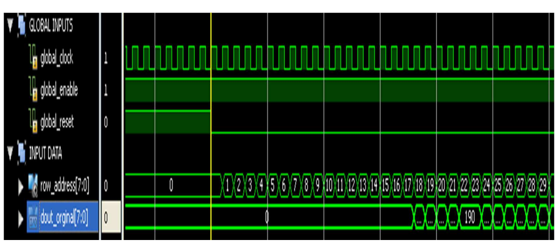
6. IFFT computation for the products obtained is done and the results are stored in memory2.
7. FFT is computed for the values stored in memory2 and the result is stored in memory1.

The correlation peaks (maximum value) obtained is the point where the input image and template matches. This can be further enhanced by Zero normalized cross correlation (ZNCC) which is left for future work

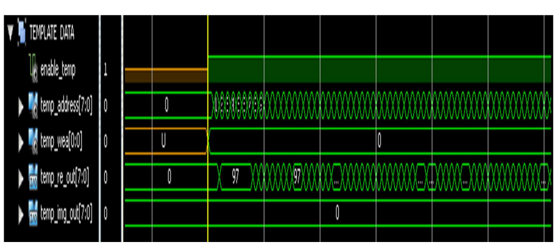
5. IMPLEMENTATION DETAILS AND RESULTS

The architecture has been designed in such a way that it can be easily implemented and reliable for real time application. This has been simulated and synthesized using Xilinx ISE version 12.1. Edge detection model was designed using Matlab system generator blocks. This model is converted into HDL code by "make HDL" option. Template matching block is designed using Xilinx IP core. Both the models are combined together in Xilinx and synthesized. Simulation and Synthesis results are shown below.

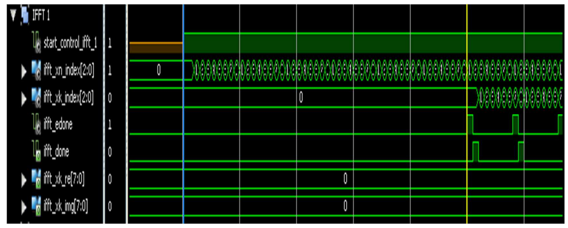
Simulation Results on Template Matching



Input image pixel values

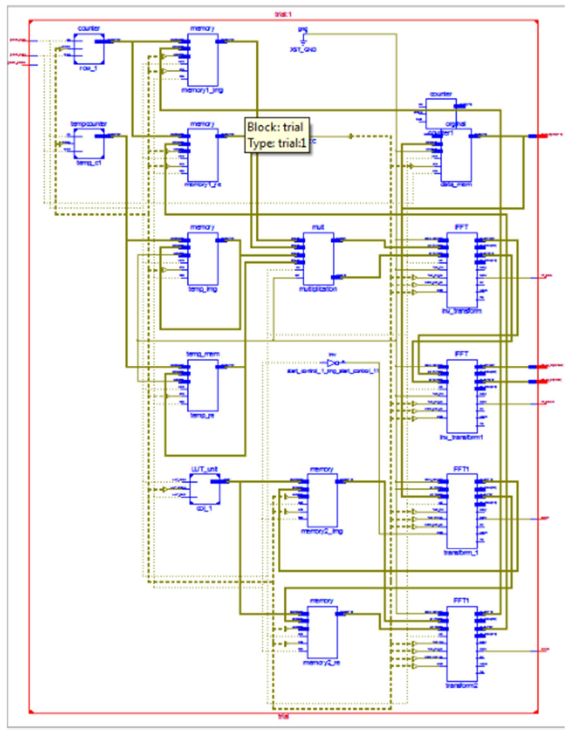


Template image pixel values



Correlated output value

Synthesis Results for Template Matching



6. CONCLUSION

The efficiency of the proposed architecture is comparatively faster than the existing architectures, which has been achieved by the logic utilization of FPGA. The accuracy of the matching, when the template is not rotated is about 98% and the computation time is calculated for the operating speed of 500MHz. The Cross-Correlation is reduced further when implemented in hardware.

RANGE	DIRECT IMPLEMENTATION (Sec)	CROSS-CORRELATION IMPLEMENTATION (Sec)
16 X 16	11.68	20.64
32 X 32	17.90	6.88
50 X 50	10.12	2.29

Table 1 Implementation Results

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

REFRENCES:

[1] Dufour, R.M.; Miller, E.L.; Galatsanos, N.P. "Template Matching Based Object Recognition With Unknown Geometric Parameters," *IEEE Transaction on Image Processing*, Vol. 11, No. 12, Dec. 2002, pp. 1385-1396.

[2] Gupta, Rahul Gupta, Amardeep Singh and Matt Wytock, "Object Recognition using Template Matching," *http://www.stanfordedu/class/cs229 /proj2008.*

[3] R. Brunelli, "Template Matching Techniques in Computer Vision: Theory and Practice," *Wiley*, 2009. doi:10.1002/9780470744055 .

[4] Li, Y., Tsin, Y., Genc, Y. and Kanade, T., "Object detection using 2D spatial ordering constraints," Computer Vision and Pattern Recognition, *CVPR 2005. IEEE Computer Society Conference on*, 20-25 June, 2005.

[5] Heisele B., Rocha C., "Local shape features for object recognition," Pattern Recognition, 2008. *ICPR 2008 19th International Conference on*, *Tampa*, FL, 8-11 Dec. 2008

[6] Neal R. Harvey, Reid Porter, and James Theiler, "Ship detection in satellite imagery using rank-order grayscale hit-or-miss transforms," *Proc. SPIE*, Vol. 7701, 2010.

[7] E. G. M. Petrakis, A. Diplaros, and E. Milios, "Matching and retrieval of distorted and occluded shapes using dy-namic programming," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 11, 2002. doi:10.1109/TPAMI.2002.1046166

[8] Feng Ge, Tiecheng Liu, Song Wang and Joachim Stahl, "Template-Based Object Detection through Partial Shape Matching and Boundary Verification," *International Journal of Information and Communication Engineering*, Vol. 4, No. 2, 2008.

[9] H. Chang, J. Zhang, "New metrics for clutter affecting human target acquisition," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 42, no. 1, pp. 361-368, Jan. 2006.

[10] V. G. Hansen and J. H. Sawyers, "Detectability loss due to greatest of selection in a cell-averaging CFAR", *Proc. of the IEEE International Radar Conference*, pp. 325-332, 1973.

[12] R. C. Gonzalez and R. E. Woods, *Digital Image Processing* (third edition), Reading, Massachusetts:Addison-Wesley, 1992.

[13] Cavadini, M., Wosnitza, M., Troster, G.: Multiprocessor system for high-resolution image correlation in real time. *IEEE Trans. Very Large Scale Integr. VLSI Syst.* 9(3), 439–449 (2001).