ISSN No. 0976-5697

Volume 1, No. 3, Sept-Oct 2010

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Digital Modeling to Resolution Scaling in Soc for Video Coding

S. K. Fairooz*
Research Scholar,
JNTU Hyderabad, A.P, India,
fairoozsk@gmail.com

Dr. B. K. Madhavi
Prof.ECE department,
Vardhaman Colloge of Engineering, Hyderabad , A.P. India,
bkmadhavi2009@gmail.com

***Abstract*:** Unlike most other information technologies, which have enjoyed an exponential growth for the past several decades, display resolution has largely stagnated. Low display resolution has in turn limited the resolution of digital images. Scaling is a non-trivial process that involves a trade-off between efficiency, smoothness and sharpness. As the size of an image is increased, so the pixels, which comprise the image, become increasingly visible, making the image to appear soft. Super scalar representation of image sequence is limited due to image information present in low dimensional image sequence. In this work, we analyze re-enhancement work on pixel classification based resolution enhancement, namely, resolution synthesis, and discuss its applicability to low complexity customer grade display systems. In the paper an FPGA implementation of the proposed algorithm is presented, and computational complexity analysis is observed.

***Keywords*:** Image, Super Resolution, FPGA, Super Scalar.

I. INTRODUCITON

Digital image coding is developed from a long time. In the area of image processing there is a need to improve the resource requirement for progressive image processing using resource optimization techniques. In earlier efforts it is observed that image sequencing can be improved by optimized usage of available resources using scheduling schemes, multiplexing schemes, representation schemes etc. The proposed conventional methods were observed to be developed keeping available resources and there constrains in mind. Today's applications demand high-resolution representation of image data for real time interfacing and communications. With the incorporation of developed optimization scheme as outlined above can provide a significant improvement in coding but in current scenario and for future applications these methods may get constrained. As the available resources such as bandwidth, power, coding techniques are limited to certain minimum values.

To achieve high resolution representation images are to be retained for good visual quality. As resource optimizations are constrained, coding techniques are to be improved to achieve the stated quality in image processing. A new approach of image representation has emerged in recent past with high-resolution projection approach for low dimensional image sequence. Where low data representation is very low in resource requirement and has low visual quality then it is suggested to be interpolated on a high-resolution grid for better visual quality. To achieve higher visual quality the stated interpolation approaches were carried out in frequency representation using transformation techniques. In this paper, we propose a modified training scheme top improve the performance under certain conditions. An FPGA [11, 12] based implementation of the proposed algorithm is been proposed, the goal it to design a low-complexity resolution enhancement method that can be implemented in the next generation display systems.

II. SYSTEM DESIGN ISSUES

The problem of image high-resolution is observed to estimate a high resolution (HR) version of an image sequence from its low-resolution (LR) equivalent. Even in the case of a sufficient number of LR samples, the presence of distortion and noise produces contradictory source information. In many ways the solution is similar to
previous approaches for still-image scenarios, but certain modifications have been introduced to improve performance and tackle some complications that arise specifically for the case of image.

There are several attractive features for this proposed method. First, the required models can also be adjusted locally for a higher-quality HR reconstruction. The second major quality of the algorithm is its efficiency, a necessary requirement for a practical image enhancement solution. Finally, the proposed method offers significant flexibility in several aspects of the solution: the scaling factor can be arbitrarily adjusted (even separately on each dimension for aspect ratio conversion), the model information (distortion, noise, and motion) can be changed locally, and the amount of source data can vary allowing a more selective inter-frame registration.

III. ARCHITECTURE OF PROPOSED METHOD

Top-level block diagram of the hardware architecture designed for the modified RS algorithm is shown in figure below.

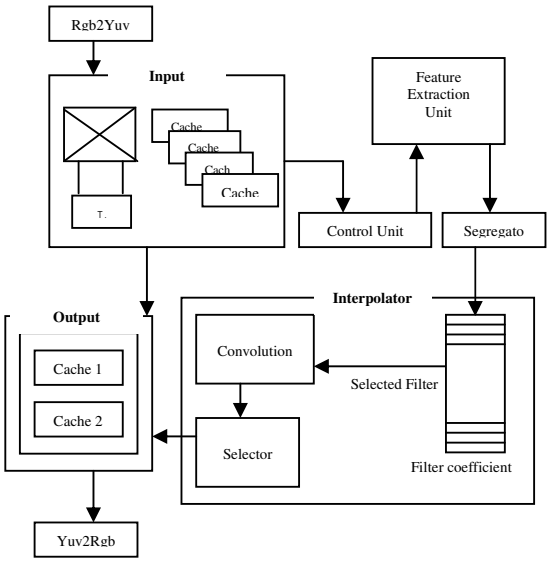

Figure 1. Top Level Block Diagram

There are seven main blocks; Modified RS is performed only on the luminance (Y) path. Chrominance signal is interpolated by pixel replication. At each input pixel arrival, the control unit reads a 4x1 pixel column from the Cache Buffer and provides the 3x3 and 5x5 pixel windows to he feature extraction and classification units. The feature extraction unit then extracts feature vector dimension of 8x1 which is then fed to the Segregator Unit for further processing. The purpose of Segregator is to perform distance calculation between prototypes of predetermined vectors and feature vector, the output is the index of the class having minimum distance of feature vector. Interpolator unit then uses this index to address filter coefficient LUT, selecting the appropriate filter for input pixel neighborhood. Constant coefficient multipliers are used form performing convolution and interpolated output pixels are then stored t output cache. Because of dynamic sample rate it is require having memories at the output to comply with the rate and order or data. The hardware implementation of the proposed scaling algorithm can be performed in two different ways:

1. Output - For each HD pixel coordinate, find the corresponding 5x5 low-resolution window, and perform feature extraction, classification, and interpolation on this neighborhood.
2. Input - For each SD pixel coordinate, find the corresponding four HD pixel coordinates. Since these HD pixels are to be interpolated from the same SD pixel neighborhood, perform feature extraction, and classification steps only for once.

Then perform interpolation for all four HD pixels corresponding to the SD pixel, and arrange the interpolated pixels in raster-scan order using cache buffers. Discard redundant pixels at the interpolation output if L_V or L_H (Scaling ratios, upper bounded by two) is a non-integer value.

Two main factors that can affect the implementation efficiency are the input/output data rate, and the target implementation platform. To provide an efficient implementation for different data rates, and different implementation platforms, the degree of resource sharing should be variable. The degree of resource sharing in feature extraction and classification units could be defined as:

D_r = Ne(# of elements in the feature vector)/N_p(# of processing paths in feature extraction)

eq.1

To achieve the desired data throughput with different resource sharing levels, core clock frequency, F_{clk_core}, must be related to the input pixel clock frequency, F_{Clk_in},with the following formula:

$$F_{clk\_core} = [D_r] F_{Clk\_in}$$

eq.2

A. *Operation of Control Unit*

Control unit (CU) provides input data to the data-path blocks at appropriate timing and format The control unit:

1. Generates a sliding window to be used by FE and IN units, by shifting the 5 x 1 pixel column from the input memory unit into the 5 x 5 pixel window.

2. Generates the memory address and control signals for IM and OM blocks, and pipeline control signals for other blocks.

3 Generates the synchronization signals defined at the video standards (hsync,vsync,data enable).

4. Generates the Pv and PH values defined in equation below.

$$P_V = [Q_v(x_V/L_V - z_V) + \varepsilon] \qquad \text{eq.3}$$

$$P_H = [Q_H(x_H/L_H - z_H) + \varepsilon] \qquad \text{eq.4}$$

Where,

P_V =Vertical phase.

PH = Horizontal phase.

x_V = Vertical coordinates of high resolution pixel.

x_H = Horizontal coordinates of high resolution pixel.

ε = Very small number like 10^{-6}.

z_V = Vertical coordinates of low resolution pixel.

z_H = Horizontal coordinates of low resolution pixel.

5. For scaling ratios less than two, selects the pixels to be omitted using equation shown below.

$$z_V = [x_V/L_V] \quad z_H = [x_H/L_H]$$

eq.5

B. *Operation of Input Memory*

Input memory (IM) unit operates at input pixel clock frequency F_{clk_in}. The block consists of four cache buffers, to provide a 5 x 1 pixel column to the CU. The length of the line buffers is equal to the input video's horizontal size. Shown below is the block diagram of the Input Unit.

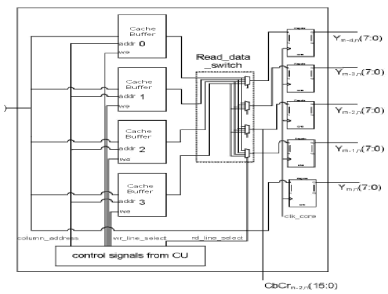

Figure 2. Input Unit

C. Operation of Segregator

Segregator unit operates at core clock frequency CU, and generates four high resolution pixels using the selected F_{clk_core}. figure below shows the architecture of the feature extraction and context classification units. parallel implementation where Dr = 1, will use 15 adders, 32 multipliers, and a serial-parallel implementation, where Dr= 4 reduces the number of adders/subtractors to four and the multipliers to eight, with a negligible increase in the number of pipeline registers.

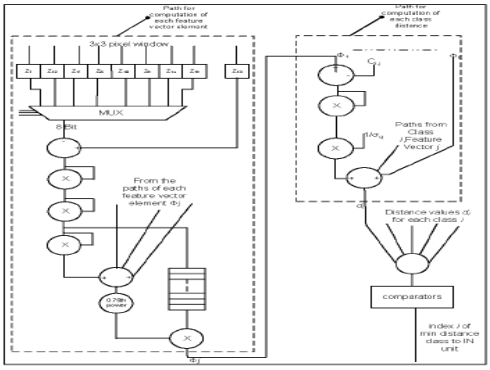

Figure 3. Context Classification and Feature Extraction.

D. Operation of Interpolation

Interpolation unit (IN) unit operates at core clock frequency. 100 multiplications, and 96 additions required to perform 5 x 5 convolution for four HD pixels is resource shared with D_r= 4 to reduce the number of multipliers to 25, adders to 24. Therefore at each F_{clk_core} clock cycle, one convolution operation is performed, generating four HD pixels at every $F_{clk\text{-}in}$, clock cycle.

E. Output Unit

Output memory (OM) unit also operates at core clock frequency. It basically chooses the appropriate pixels between the high resolutions pixels generated by IN unit, and arranges them in raster scan order according to the scaling ratio.

IV. RESULTS & OBSERVATIONS

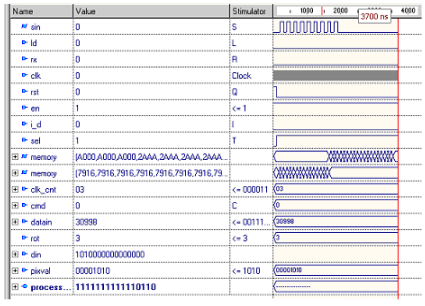

Figure 4. Simulation plot for memory interface unit

Figure 4 illustrates the simulation results obtained form the simulation of FPGA unit under various images processing operation.

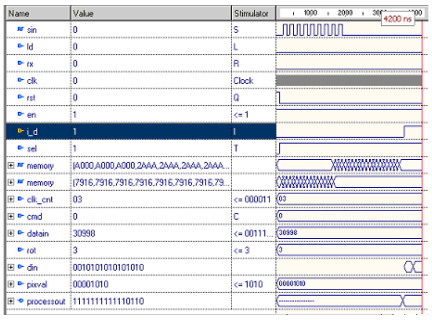

Figure 5. Simulation plot for memory interface unit.

The figure above illustrates the operation of pixel enhancement (increment). A total of 8 bit bus width is provided for this signal to provide a rotation of 256 pixels at a time.

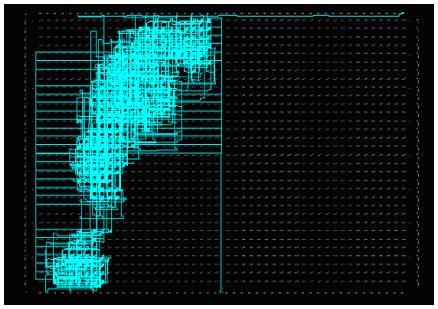

Figure 6. Routing of the implemented core into the targeted FPGA.

Figure 6 illustrates the implemented of the video interface unit developed on VHDL and implemented onto the targeted FPGA (xcv300-bg432-6) for the real time realization. The obtained implementation detail are BEL=1041 and timing report is 9.058ns. The observations of multiple frameset are carried out with the interface of a test bench and are interfaced with Matlab tool to observe the results. The original frame sequence is taken at a very low resolution with pixel representation of 150x250 size frame. These 5 frame sequences are passed to the developed system for pre-processing and the results obtained is shown below.

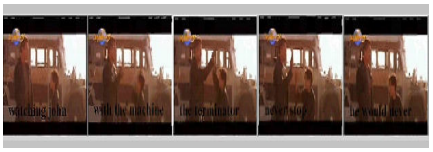

Figure 7. Original image sequence considered

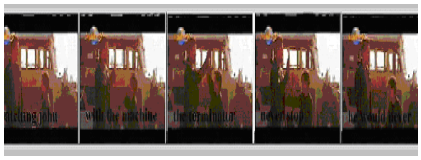

Figure 8. Scaled image sequences at 1:2.5 ratio

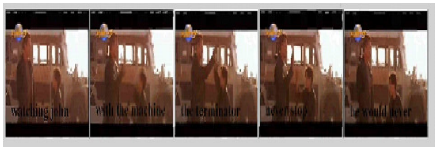

Figure 9. Scaled image sequences at 1:2.5 ratio

The observation clearly illustrates the accuracy in retrieval in terms of visual quality as compared to the conventional Fourier based coding technique.

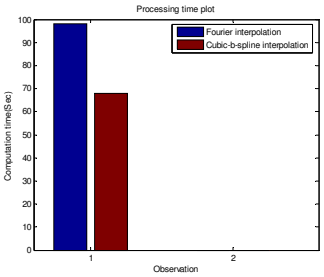

Figure 10. Computation time taken for the two methods.

The system developed is also evaluated for the computation time taken for the computation and projection of the frame sequence for interpolation. The total time taken for reading, processing and projecting is considered for the processing system.

V. CONCLUSIONS

A hardware implementation for a classification based resolution enhancement method has not been presented previously. Proposed method is therefore compared with several FPGA implementations of simpler LSI scaling methods.

By eliminating the need for soft interpolation and reducing the number of context classes. The computational complexity of resolution synthesis with negligible visual quality loss is lowered. The modified RS algorithm is simple enough to be implemented on low cost FPGA boards.

VI. REFERENCES

[1] S. P. Kim, N. K. Bose, and H. M. Valenzuela, “Recursive reconstruction of high resolution image from noisy under-sampled multiframes”, IEEE Trans. Acoust., Speech, Signal Processing, vol. 38, pp. 1013-1027, June 1990.

[2] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar, “Fast and robust multiframe super resolution”, IEEE Trans. Image Processing, vol. 13, pp. 1327-1344, Oct. 2004.

[3] X. Li and M. T. Orchard, “New edge-directed interpolation”, IEEE Trans. Image Proc., vol. 10, pp. 1521-1527, Oct. 2001.

[4] H. A. Aly and E. Dubois, “Specification of the observation model for regularized image up-sampling,” IEEE Trans. Image Processing, vol. 14, pp. 567-576, May 2005.

[5] R. S. Prendergast and T. Q. Nguyen, “Spectral modeling and Fourier domain recovery of high-resolution images from jointly under sampled image sets”, under review for IEEE Trans. Image Proc., submitted Dec. 18, 2006.

[6] K. S. Ni and T. Q. Nguyen,“Image superresolution using support vector regression”, IEEE Trans. Image Proc., vol. 16, pp. 1596- 1610, June 2007.

[7] S. C. Park, M. K. Park, and M. G. Kang, “Super-resolution image reconstruction: a technical overview”, IEEE Signal Processing Mag., vol. 20, pp. 21-36, May 2003.

[8] Narayanan, R. C. Hardie, K. E. Barner, and M. Shao, “A computationally efficient super-resolution algorithm for image processing using partition filters,” IEEE Trans. on Circ. Syst. For Image Technology, vol. 17, no. 5, pp. 621-634, May 2007.

[9] S. Farsiu, M. Elad, and P. Milanfar, “Image-to-image dynamic superresolution for grayscale and color sequences,” EURASIP Journal of Applied Signal Processing, Special Issue on Superresolution Imaging, vol. 2006, pp. 1–15, 2006.

[10] R. C. Hardie, “A fast image super-resolution algorithm using an adaptive Wiener filter,” IEEE Trans. Image Proc., vol. 16, no. 12, pp. 2953-2964, Dec. 2007.

[11] S. S. S. Ramachandran, "Design and FPGA implementation of a video scalar with on-chip reduced memory utilization complexity," in Euro micro Symposium on Digital Systems Design (DSD '03), 2003.

[12] M. Aurelio, Nuo-Maganda, and M. 0. Arias-Estrada, "Real with time fpga-based architecture for bicubic interpolation: An application for digital image scaling," in International Conference on Reconfigurable Computing and FPGAs (ReCon- Fig'05), 2005.

