Resolution and Scaling Video Coding For SoCs in Digital Modeling

S. K. Fairooz and B. K. Madhavi

***Abstract*—Soc Designing has attained high attention in recent past. For the past several decades there is an exponential growth in the demand for the improvement in display resolution. Resolution of digital imaging has been turn limited to low display resolution due to resource overhead. A non-trivial process such as scaling gives a trade-off between efficiency, smoothness and sharpness. For the visualization improvement of imaging samples the resolutional increment is achieved by increasing the size of display, so the pixels, which comprise the image, get expanded. Super scalar representation of image sequence is limited due to image information present in low dimensional image sequence. This paper outlines an Digital implementation of the video coding algorithm for low resource overhead in SoC designing for resolution enhancements. The Complexity of designing and implementation overhead for such system is presented and a generic model for low complex digital design is proposed.**

***Index Terms*—Digital designing, super scalar coding, resolution representation, computational complexity.**

I. Introduction

From a long time, digital image coding is under developed. In the area of image processing there is a need to improve the resource requirements, for progressive image processing using resource optimization techniques [1]. In earlier approaches [2] it is observed that the optimizations were obtained by usage of available resources using multiplexing, representation schemes, and scheduling schemes. These proposed conventional methods [3] were developed keeping available resources and there constraints in mind. For all real time interfacing and communications, the system demands high-resolution representation of an image data [4],[5]. With the optimization schemes 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 retain good visual quality a high resolution representation is required. To achieve the stated quality in image processing the coding techniques [6], [7] are to be improved due to resource optimizations. A new technique of image representation has emerged in recent past with high-resolution projection approach for Low dimensional image sequence [8]. Wherein data representation in low dimension is very low in resource requirement, but has low visual quality. Hence it is suggested to be interpolated on a high-resolution grid for better visual quality.

The stated interpolated approaches were carried out in frequency representation using transformation techniques to achieve higher visual quality [9]. In this paper, we proposed a modified operation scheme to [10], [11] for the improvement of performance under variable conditions. A low complexity resolution enhancement method that can be implemented in the next generation displays systems using FPGA based implementation of the proposed algorithm is presented.

Manuscript received May 22, 2012, revised July 31 2012.

S. K. Fairooz is with JNTU Hyderabad, A.P, India (e-mail: fairoozsk@gmail.com).

B. K. Madhavi is with the Geethanjali College of Engineering and Technology, Hyderabad , A.P. India.

II. System Design Issues

There is a problem of image high resolution to estimate a high resolution (HR) version of a image sequence from its low- resolution (LR) equivalent. The presence of distortion and noise produces contradictory source information in the case of a sufficient number of LR samples. 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. The features for this proposed method are first the required models can also be adjusted locally for a higher – quality HR reconstruction. The second feature is the quality of the algorithm is its efficiency 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. Proposed Architecture

The hardware architecture designed for the modified RS algorithm is shown as the top level diagram in the Fig. 1.

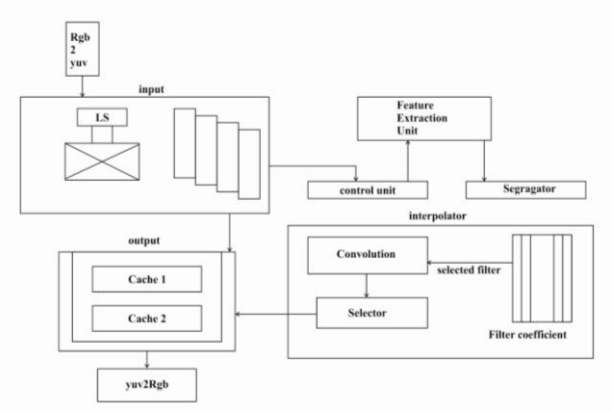

Fig. 1. Top level block diagram.

DOI: 10.7763/IJIET.2012.V2.187

In the above diagram there are different blocks, Modified RS algorithm is performed on the luminance (Y) path. Chrominance signal is interpolated by the pixel replication. At each input pixel arrival, the control unit reads a 4×1 pixel column from the Cache Buffer and provides the 3×3 and 5×5 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 neighbourhood. Constant coefficient multipliers are used form performing convolution and interpolated output pixels are then stored at 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 5×5 low-resolution window, and perform feature extraction, classification, and interpolation on this neighbourhood.
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 neighbourhood, 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:

$$D_r = N_e(\#\text{ of elements in the feature vector})/N_p(\#\text{ of processing paths in feature extraction}) \quad (1)$$

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:

$$F_{clk\_core} = [D_r]\, F_{Clk\_in} \quad (2)$$

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 5×1 pixel column from the input memory unit into the 5×5 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.

$$P_V = [Q_V(x_V/L_V - z_V) + \varepsilon\,] \quad (3)$$
$$P_H = [Q_H(x_H/L_H - z_H) + \varepsilon\,] \quad (4)$$

where

P_V =Vertical phase.
P_H = 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.

$$z_V = [x_V/L_V]\; z_H = [x_H/L_H] \quad (5)$$

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 5×1 pixel column to the CU. The length of the line buffers is equal to the input video's horizontal size. Fig. 2. shows the block diagram of the Input Unit.

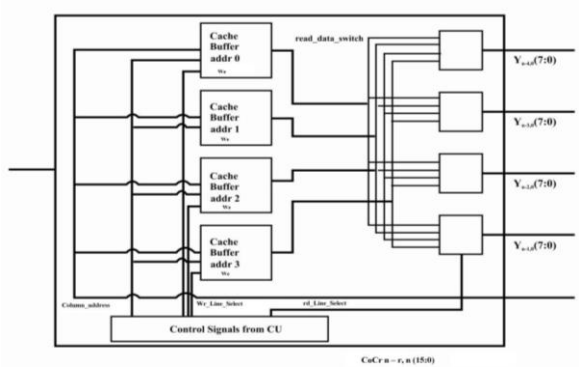

Fig. 2. Block diagram of input unit.

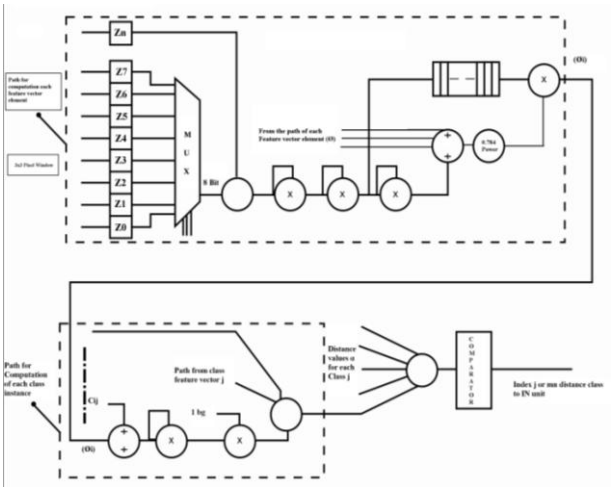

Fig. 3. Context classification and Feature Extraction.

C. Operation of Segregator

Segregator unit operates at core clock frequency CU, and generates four high resolution pixels using the selected F_{clk_core}. Fig. 3. shows the architecture of the feature

extraction and context classification units. Parallel implementation where D_r = 1, will use 15 adders, 32 multipliers, and a serial- parallel implementation, where D_r= 4 reduces the number of adders/subtractors to four and the multipliers to eight, with a negligible increase in the number of pipeline registers.

D. Operation of Interpolation

Interpolation unit (IN) unit operates at core clock frequency. 100 multiplications, and 96 additions required to perform 5×5 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. Observations

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

Fig. 5. 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.

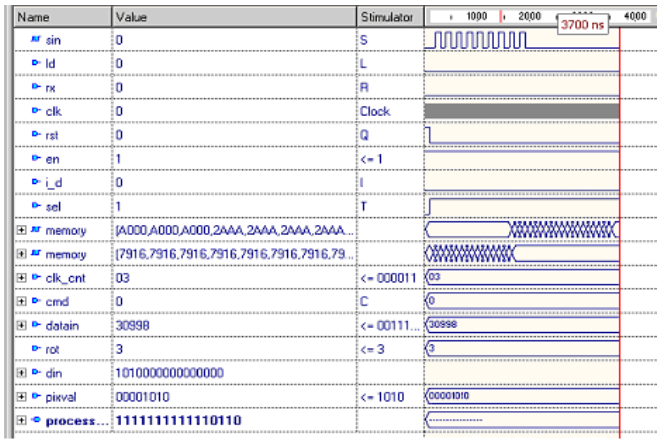

Fig. 4. Simulation plot for memory interface unit.

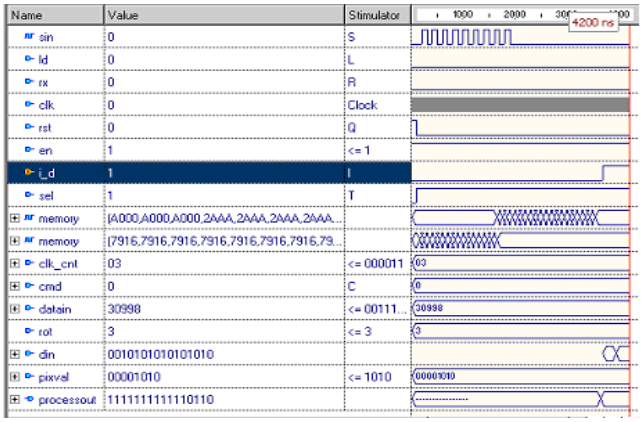

Fig. 5. Simulation plot for memory interface unit.

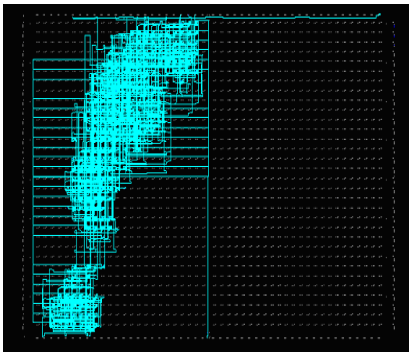

Fig. 6. Routing of the implemented core into the targeted FPGA.

Fig. 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 150×250 size frame. These 5 frame sequences are passed to the developed system for pre-processing and the results obtained is shown in Fig. 7 to Fig. 9.

The observation clearly illustrates the accuracy in retrieval in terms of visual quality as compared to the conventional Fourier based coding technique, as show in Fig. 10.

Fig. 7. Original image sequence considered

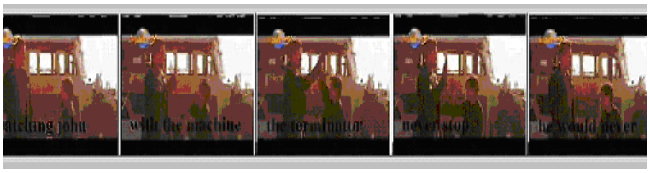

Fig. 8. Scaled image sequences at 1:2.5 ratio

Fig. 9. Scaled image sequences at 1:2.5 ratio

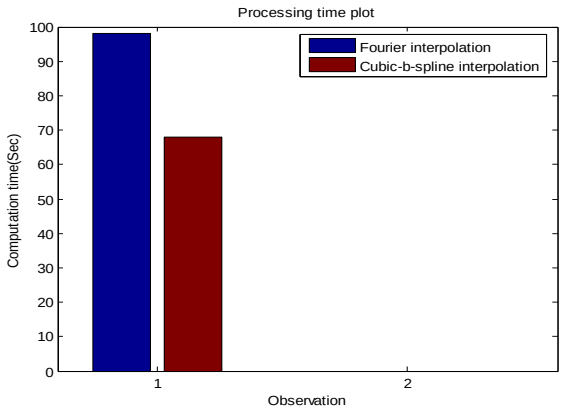

Fig. 10. Computation time taken for the two methods.

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. The 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.

References

[1] S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruction: a technical overview," *IEEE Signal Processing Magazine*, vol. 20, pp. 21-36, May 2003.

[2] S. Farsiu, M. Elad, and P. Milanfar, "Image-to-image dynamic super resolution for greyscale and color sequences," *EURASIP Journal on Applied Signal Processing, Special Issue on Super resolution Imaging*, vol. 2006, pp. 1–15, Jan. 2006.

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

[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] B. 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.

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

[7] 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.

[8] X. Li and M. T. Orchard, "New edge-directed interpolation," *IEEE Transaction Image Processing*, vol. 10, pp. 1521-1527, Oct. 2001.

[9] R. S. Prendergast and T. Q. Nguyen, "Spectral modeling and Fourier domain recovery of high-resolution images from jointly under sampled image sets," *IEEE Trans. Image Processing*, Dec. 18, 2006.

[10] S. S. S. Ramachandran, "Design and FPGA implementation of a video scalar with on-chip reduced memory utilization complexity," present at Euro Micro Symposium on Digital Systems Design, 2003.

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