A Novel Error Free Dimensional Projection Coding

1 Shaik Mohammed Rafi, 2 G D Basavaraj, 3 Dr. Shaik Fairooz.
1,2 Assistant Professor, Sreyas Institute of Engineering and Technology, India
3 Associate Professor, Sreyas Institute of Engineering and Technology, India
mohammed.rafi@sreyas.ac.in, basavaraj.g@sreyas.ac.in, fairooz.sk@sreyas.ac.in

Abstract: In this paper, dimensional coding of projection has been developed with the hardware design for RS algorithms. The result of simulation of Memory Interface Unit for basic and dimensional projection coding with scale and Implemented Video Interface module using VHDL and Target on to the FPGA implemented (XCV300-bg432-6). It provided operating speed of 732.5 MHz, power consumption of 1249 BEL and 128.3 mW in FPGA.

Keywords: FPGA, BEL’s, VHDL, HD, Video Codec, Projectional Coding.

1. Introduction:

Unlike most other information technologies, whose exceptional growth has increased in recent decades, the resolution of the screen has largely stopped being reduced to less resolution digital video screens[1] and now offers high-resolution screen scaling, high-resolution digital video scaling [4] open opportunity, they have proposed a general framework for video presentation and flexibility[5-7]. As the size of the video increases, pixels, which are part of the video, begin to appear faster, make the video visible. On the contrary, it is obvious to reduce a video which leads to better and sharpness. It has a codec and a numerical video screen, which allows the user to convert the low-resolution [2] content to the signal in the form of HD content in that controller. Based on the quality of the projector located above the launch, the yield feature of the output video shown can be improved.

2. Traditional System Design Issues:

High resolution video problems when evaluating a high-resolution (HR) [3,8] version that is equivalent to low-resolution equivalent (LR) video stream. First, the properties of this recommended method to reproduce more selective human resources between high quality and frames, locally required models can be proposed.

3. Proposed Architecture:

Architectures designed for RS algorithms have been modified according to the upper level block diagram, shown in Figure 1.
3.1 (a) Functional Description

The structure shown above is made up of different functional blocks; the RS adjustment algorithm is used in color-coded signals by luminance path (Y) and pixel replication. On each pixel input, the controller reads the column cache of 4 x 1 pixels and provides 5 x 5 pixel windows and 3x3 extraction and classification. The function is to eliminate the vector and then there is another dimension other than 8 x1, which again selects a predefined model, the division of vector characteristics and the appropriate filter for the pixel input environment. Fixed transaction multiplier was used to adjust and the output result of the pixel was placed in the cache memory.

1. Output - With HD Pixel format, find the window analogous to the 5x5 precision in this neighborhood and remove, compile and interpolate features.

2. Input - For each SD pixel format, select the analogous coordinates of 4 HD pixels.

End the final pixel interpolation output if the LV or LH (decreasing ratio is high) [9] is the wrong value. Two key factors affecting the efficiency of implementation is a fast data entry / exit and a destination that is defined as below:

\[ D_r = \frac{N_e}{N_p} \]  

(1)

Ne represents the number of elements for feature vector

Np represents the number of processing paths for feature extraction

To achieve the essential performance levels with dissimilar data and resource sharing, the core clock frequency, Fclk_core, the frequency input of the FClk_in pixel clock must be associated with the given formula below:
Control unit (CU) provides information access to the data path blocks in a timely manner and format of control unit.

1. Feature Extraction uses a sliding window, memory conversion pixel window input column 5 x 1 by 5 x 5 pixels.
2. Control unit produce memory address and control signals for input memory and output memory blocks
3. It produces exact synchronization signal in video standards
4. It produces $X_H$ and $X_V$ values mentioned in the subsequent equation.

$$X_{Vs} = [Q_{Vs}(Y_{Vs}/L_{Vs} - R_{Vs}) + \epsilon_s] \quad (3)$$

$$X_{Hs} = [Q_{Hs}(Y_{Hs}/L_{Hs} - R_{Hs}) + \epsilon_s] \quad (4)$$

Where,

$X_V$ = Projection plane Vertical.

$Y_H$ = Projection plane Horizontal.

$Y_V$ = Represents the high resolution pixel using vertical coordinates.

$Y_H$ = Represents the high resolution pixel using horizontal coordinates.

$\epsilon$ = Constant ($10^{-6}$.)

$R_V$ = Low resolution pixel using vertical coordinates.

$R_H$ = High resolution pixel using horizontal coordinates.

5. For projection ratios less than 2, the pixels are given as,

$$R_{Vs} = [Y_{Vs}/L_{Vs}] \quad (5)$$

$$R_{Hs} = [Y_{Hs}/L_{Hs}]$$

3.1(c) Input Memory

Input memory module (IM) input clock frequency $F_{clk_in}$ runs in blocks of 4 cache memory stores, 5x1 pixel columns for the CU is provided. The extent of the store line is the horizontal size of the video input.
3.1(d) Segregator Unit

The function works with the central clock of the CPU and produces 4 high-resolution pixels using specified Fclk_core. The number of pipeline registers is shown in Figure 3 with a slight increase in reference classification and pixel extraction unit.

Figure 3: Context Classification and Pixel Extraction.

3.1(e) Interpolation Unit

The interpolation unit (IN) is used as an adjustment function, which produces 4 high definition pixels every cycle in Fclk.

3.1(f) Output Unit

The output memory module (OM) works by the original clock frequency. Basically, select the appropriate pixel between the high-resolution pixels generated by these units and adjust the scanning scans according to the mapping relationship.
4. Simulation Results:

Simulation result of memory interface unit for original and dimensional projection coding with scale ratio shown below.

**Figure 4:** Simulation plot for memory interface unit

The total width of the 8-bit bus has been provided so that the signal can provide a rotation of 256 pixels at a time. The data stored in the buffer is stored in the memory unit, where the input is done through Signal and the processed data is distributed as the output of the process. Depending on the control signal RD and the signal line SL, the writing and reading task is selected.

**Figure 5:** Simulation plot for memory interface unit.
Figure 6: Routing of the implemented core into the targeted FPGA

Figure 6 shows the implementation of FPGA (XCV-300-BG 432-6) in the FPGA, through implementation in basic routing and advanced VHDL video interface module. To monitor the collection of results from several frames, it is executed with the Interface and Multisim Tool with the testbench. The processing of five frames are passed to a system and the results obtained are shown. in the Figure 7, Figure 8 with the ratios of 1:2:5.

Figure 7: Sequence of input video

Figure 8: Sequence of video scaled at 1:2.5 ratios
An illustration shows clearly the accuracy of the recovery in terms of visual quality compared to traditional technology based on Fourier coding, as shown in Figure 10. Comparison time is better compared to comparison to B-bar cubic interpolation, as shown in Figure 10, Fourier interpolation techniques.

![Figure 9: Computation time taken for the two methods](image)

While computing the sequence of tables and calculating the time used in the project, the system has been developed to see if it has been completed.

5. Conclusion:

In the recommended approach, data for pixel-encoded frame coding passes through decimal methods at lower resolutions. At the reception, it has been suggested that the interpolation logic uses horizontal / vertical projection to achieve high resolution screens. This proposed interpolation argument shows better operation in each encoding coding using B-band projection. The objective is 732.5 MHz operating speed, 1249 BEL power consumption and FPGA provides 128.3 mW.

References

[1] Dzung Nguyen, Minh Dao, Trac Tran, “Video Error Concealment Using Sparse Recovery And Local Dictionaries”, IEEE International Conference On Acoustics, Speech, And Signal Processing, 2011.

[2] Yui-Lam Chan, Hoi-Kin Cheung, and Wan-Chi Siu, “Compressed-Domain Techniques for Error-Resilient Video Transcoding Using RPS”, IEEE Transactions on Image Processing, Vol. 18, No. 2, 2009.

[3] Muhammad Shoaib, “Key Picture Error Concealment Using Residual Motion-Copy in Scalable Video Coding”, World Congress on Computer Science and Information Engineering, 2009.

[4] M. Tun, K. K. Loo and J. Cosmas, “Error-Resilient Performance of Dirac Video Codec Over Packet-Erasure Channel”, IEEE Transactions on Broadcasting, Vol. 53, No. 3, 2007.

[5] Zhenyu Wu and Jill M. Boyce, “Adaptive Error Resilient Video Coding Based On Redundant Slices of
H.264/AVC”, IEEE International Conference on Multimedia and Expo, 2007.

[6] Daniel Persson, Thomas Eriksson, and Per Hedelin, “Packet Video Error Concealment with Gaussian Mixture Models”, IEEE Transactions on Image Processing, Vol. 17, No. 2, 2008.

[7] Yi J. Liang, John G. Apostolopoulos and Bernd Girod, “Analysis of Packet Loss for Compressed Video: Effect of Burst Losses and Correlation between Error Frames”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 18, No. 7, 2008.

[8] Antonios Argyriou, “Error-Resilient Video Encoding and Transmission in Multirate Wireless LANs”, IEEE Transactions On Multimedia, Vol. 10, No. 5, 2008.

[9] Dieu Thanh Nguyen, Michio Hayashi and Joern Ostermann, “Adaptive Error Protection for Scalable Video Coding Extension of H.264/AVC”, IEEE International Conference on Multimedia and Expo, 2008.