ISSN: 2395-7639

International Journal of Multidisciplinary Research in Science, Engineering, Technology & Management (IJMRSETM)

Visit: www.ijmrsetm.com

Volume 2, Issue 4, April 2015

An Effective Colour Interpolation in CFA Images using Adaptive Enhancement Technique and SIFT Design

Punitha M

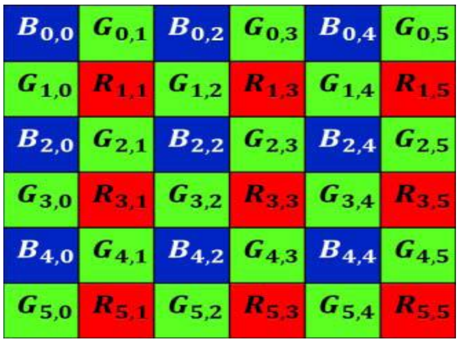
Sri Shakthi Institute of Engineering and Technology, Coimbatore, India

ABSTRACT: The algorithm consists of an anisotropic weighting model, which is designed to catch more information in horizontal than vertical directions. The filter-based compensation methodology which includes a Laplacian and spatial sharpening filters, which are developed to improve the edge information and reduce the blurring effect and an edge detector. By using these technique hardware cost is successfully reduced by hardware sharing and reconfigurable design techniques.

KEYWORDS: Camera, colour Filter Array, Edge detector , Sharpening spatial filter,

I. INTRODUCTION

Recently, digital cameras are integrated into many consumer electronic products, such as digital camera, digital video, smart TV, smart phone, tablet PC, and so on. The digital cameras are developed by a charge-coupled device or a CMOS image sensor that can capture images by color filter array (CFA) technique. The red (R), green (G), and blue (B) colors are sampled as one color in each pixel [1]. Fig. 1 shows a CFAs called Bayer CFA, in which two colors have disappeared in each pixel. Thus, it is important to reconstruct the images from CFA to full RGB formats. Many efficient high-quality algorithms [2]–[18] have been proposed for reconstructing the full RGB color from CFA images. An adaptive color interpolation technique that used a 2-D locally stationary Gaussian process and an edge indicator was proposed in [2]. Several efficient techniques such as selecting [7], [8] or fusing [9] the information of the vertical direction (DV) and horizontal direction (DH) has also been presented. A gradient based scheme with a Gaussian low-pass filter to enhance the performance of the color interpolation was proposed in [10]. The high-quality color interpolation algorithms have the characteristics of high complexity and high memory requirement. This implementation requires a two line buffer memory, which is much less than a frame memory in [19]. Second, a low-complexity edge detector was created to enhance the edge information. It used only addition, subtraction, and absolute operations to obtain the edge information. The hardware cost of the edge detector is much less than the previous designs which use dividers and multipliers to obtain the edge and gain information. Third, a novel anisotropic weighting model was designed for the proposed color interpolator. It can improve the quality of the interpolated image by acquiring more information from the DH than the vertical without adding line-buffer memory. Fourth, the proposed filter design can achieve a good quality of the interpolated images because it uses various colors of the original CFA pixels and double interpolated green pixels as elements of the filters rather than single color of original CFA pixels and single color pixel to compensate the interpolated pixel as presented in [20]–[22]. It consists of an edge detector to enhance the edge information in the images, an anisotropic weighting model to reduce the memory requirement, a filter-based RB compensator to improve the quality, and a register bank to process streaming data directly using only a two-lime-buffer memory.



$B_{0,0}$	$G_{0,1}$	$B_{0,2}$	$G_{0,3}$	$B_{0,4}$	$G_{0,5}$
$G_{1,0}$	$R_{1,1}$	$G_{1,2}$	$R_{1,3}$	$G_{1,4}$	$R_{1,5}$
$B_{2,0}$	$G_{2,1}$	$B_{2,2}$	$G_{2,3}$	$B_{2,4}$	$G_{2,5}$
$G_{3,0}$	$R_{3,1}$	$G_{3,2}$	$R_{3,3}$	$G_{3,4}$	$R_{3,5}$
$B_{4,0}$	$G_{4,1}$	$B_{4,2}$	$G_{4,3}$	$B_{4,4}$	$G_{4,5}$
$G_{5,0}$	$R_{5,1}$	$G_{5,2}$	$R_{5,3}$	$G_{5,4}$	$R_{5,5}$

Fig .1 Bayer CFA.

Objective : To convert CFA sensor image into RGB image with low power and minimum area .

1.1 EXISTING BLOCK:

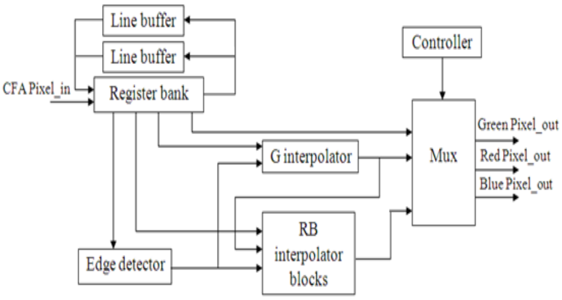


Fig 1.1 Block diagram of color interpolation processor.

Advantage :

- High quality
- The hardware cost of the edge detector is much less than the previous design sused only addition and subtraction process

1.2 PROPOSED BLOCK:

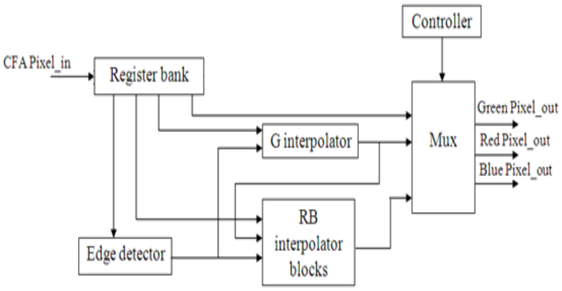


Fig 1.2 Block diagram of Proposed color Interpolation processor

Fig. 1.2 shows the block diagram of the proposed color interpolation processor. It consists of seven main blocks: a register bank, an edge detector, a green color interpolator (G interpolator), a red and blue colors interpolator model 1 (RB_M1 interpolator), red and blue colors interpolator model 2 (RB_M2 interpolator), a red and blue colors interpolator model 3 (RB_M3 interpolator), and a controller. The details of each part will be described in the following sections.

Advantage :

- Area is minimized (Removed unwanted registers)
- Fast execution (by using parallel operation for store the register value)
- Low power (By using unsigned addition process, it reduced over all power consumption)

PROPOSED TECHNIQUE

1.3 PROPOSED SIFT ARCHITECTURE WITH COLOR FILTER ARRAY :

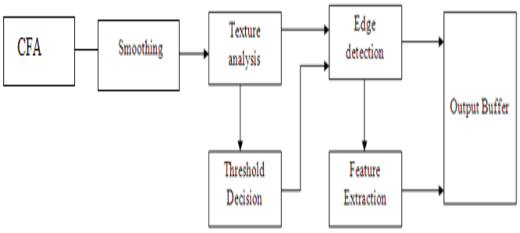


Fig. 1.3 Proposed architecture diagram.

Fig 1.3 shows the proposed architecture that consists of two stages: scale-space extrema detection and keypoint localization in the stage one, and orientation assignment and the local frame descriptor in the stage two. In the stage one, we adopt the proposed LPSIFT with integral image to reduce the computation significantly, and further reduce the required integral image buffer by the on-the-fly computation scheduling. In the stage two, we adopt the proposed brightness threshold to simplify computation and implement the normalization in keypoint localization with a low cost universal operation unit with precision equivalent cycles (PEC).

1.4 RED INTERPOLATOR BLOCK DIAGRAM:

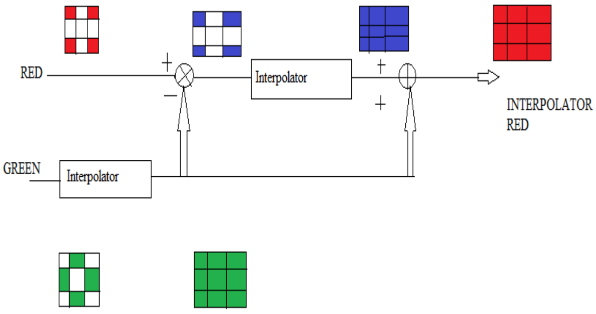


Fig 1.4 Red Interpolator Block diagram

1.5 CONSTRUCTION *OF COLOR IMAGE FROM COLOR PLANES:*

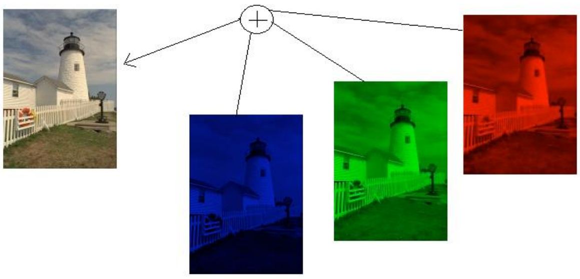


Fig 1.5 Construction of Color Image From Color Planes*:*

By Adding three color interpolators original color image is formed. Color Interpolation algorithm is used to develop a low-cost, low power, high performance and high quality color interpolation processor for real time application. An anisotropic weighting model ,edge detector, laplacian and sharpening filter have been used to reduce the memory and improve the quality of the image.

1.6 GREEN COLOR INTERPOLATION

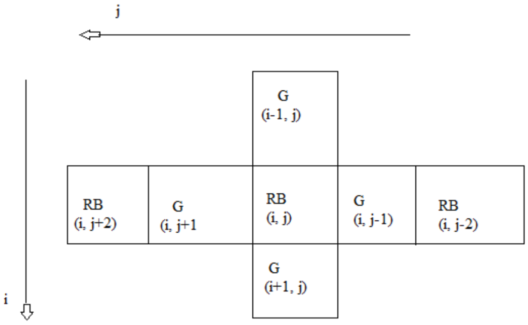


Fig 1.6 (a) Relative locations and reference neighbouring samples of G(*i*, *j*) in CFA format for green color interpolation.

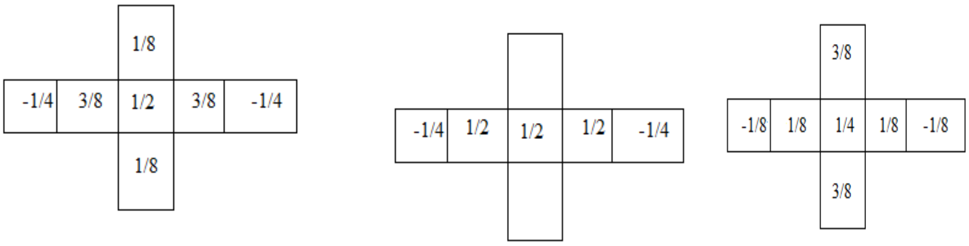


Fig 1.6.1 Parameters of green color interpolation (a) without edge enhancement model, (b) with edge enhancement in DH model, and (c) with edge enhancement in DV model.

To improve the quality of interpolated images, an anisotropic weighting model was created for this design. As shown in Fig. 1.6(a), the reference neighboring pixels in DH are much more than those in DV. The location of G(*i*, *j*) could be the color pixel of B(*i*, *j*) or R(*i*, *j*), as shown in Fig. 1.6(a). Hence, the interpolated pixel G(*i*, *j*) is described as G(RB) *(i, j)* where the location of P*(i, j)* is R*(i, j)* or B*(i, j)* in the original CFA images, in which the P*(i, j)* represents the pixel at the location of *i* and *j* in the *y* and *x* coordinates, respectively.

$$G_{i,j}^{(RB)G} = \frac{3}{8}(G_{i,j-1}+G_{i,j+1})+\frac{1}{8}(G_{i-1,j}+G_{i+1,j})+ RB_{i,j}- \frac{1}{2}(\frac{1}{2}(RB_{i,j}+RB_{i,j-2})+\frac{1}{2}(RB_{i.j}+R_{i,j+2}) \qquad (1)$$

where G*(*RB*)*G *i, j* is the result of G*(i, j)* where the location of P*(i, j)* is R*(i, j)* or B*(i, j).* Fig. 1.6.1(a) shows the parameters of green color interpolation without edge enhancement model. Otherwise, if the value of the TD is larger than the threshold value and DH is less than DV, the value of G*(*RB*) (i, j)* can be obtained by the edge enhancement in DH model as

$$G_{i,j}^{(RB)G} = \frac{1}{2}(G_{i,j-1}+G_{i,j+1})+RB_{i,j} - \frac{1}{2}(\frac{1}{2}(RB_{i,j}+RB_{i,j-2})+\frac{1}{2}(RB_{i.j}+R_{i,j+2}) \qquad (2)$$

Fig. 1.6.1(b) shows the parameters of green color interpolation with edge enhancement in DH model. Also, if the value of TD is larger than the threshold value and the value of DH is larger than DV, the value of G*(*RB*) (i, j)* can be computed by the edge enhancement in DV model as

$$G_{i,j}^{(RB)G} = \frac{3}{8}(G_{i,j-1}+G_{i,j+1})+)+\frac{3}{8}(G_{i-1,j}+G_{i+1,j})+\frac{1}{2}\{RB_{i,j}- \frac{1}{2}(\frac{1}{2}(RB_{i,j}+RB_{i,j-2})+\frac{1}{2}(RB_{i.j}+R_{i,j+2}) \qquad (3)$$

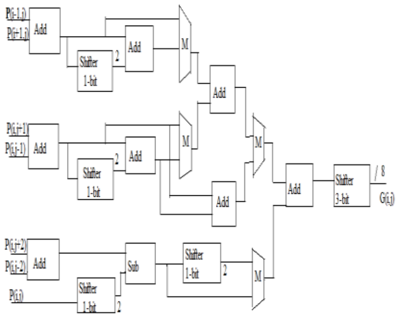


Fig 1.6.2. Architecture of the green color interpolator (G interpolator).

1.7 RED AND BLUE COLORS INTERPOLATION

1/4	-1/4	1/4
-1/4	1	-1/4
1/4	-1/4	1/4

1/4	-3/8	1/4
-1/8	1	-1/8
1/4	-3/8	1/4

1/4	-1/8	1/4
-3/8	1	-3/8
1/4	-1/8	1/4

Fig 1.7 Parameters of red and blue colors interpolation at G*(i, j)* (a) without edge enhancement model, (b) with edge enhancement in DH model, and (c) with edge enhancement in DV model.

Fig. 1.7(a) shows the relative locations and reference neighbouring samples of G*(i, j*+1*)*, G*(i*−1*, j)*, G*(i, j*−1*)*, and G*(i*+1*, j)* where the pixel in CFA format is R*(i, j)* or B*(i, j)*. If the value of TD is less than a threshold value, the value of R*(*B*) (i, j)* or B*(*R*) (i, j)* can be calculated by the without edge enhancement model as

$$RB_{i,j}^{(BR)G} = \frac{1}{4}(RB_{i-1,j-1} + RB_{i-1,j+1} + RB_{i+1,j-1} +RB_{i+1,j+1)} +g_{i,j} - \frac{1}{4}(G_{i-1,j} +G_{i+1,j} +G_{i,j-1} +G_{i,j+1}) \quad (4)$$

where g*(i, j)* is produced by green color interpolation which is mentioned above. Otherwise, if the value of TD is larger than the threshold value and DH is less than DV, then the value of R*(*B*) (i, j)* or B*(*R*) (i, j)* can be obtained by the edge enhancement in DH model as

$$RB_{i,j}^{(BR)G} = \frac{1}{4}(RB_{i-1,\, j-1} + RB_{i-1,\, j+1} + RB_{i+1,\, j-1} + RB_{i+1,\, j+1)} + g_{i,\, j} - \frac{1}{8}(3 * (G_{i-1,\, j} + G_{i+1,\, j}) + G_{i,j-1} + G_{i,\, j+1}) \qquad (5)$$

Fig. 1.8(b) shows the parameters of red and blue colors interpolation with edge enhancement in DH model. Also, if the value of TD is larger than the threshold value and DH is larger than DV

$$RB_{i,j}^{(BR)G} = \frac{1}{4}(RB_{i-1,\, j-1} + RB_{i-1,\, j+1} + RB_{i+1,\, j-1} + RB_{i+1,\, j+1)} + g_{i,\, j} - \frac{1}{8}(G_{i-1,\, j} + G_{i+1,\, j}) + 3 * (G_{i,j-1} + G_{i,\, j+1})) \qquad (6)$$

To analyze (4)–(6), all input signals should be the same except the values of their parameters. Hence, the reconfigurable technique can be used to design the hardware architecture of the red and blue colors interpolator.

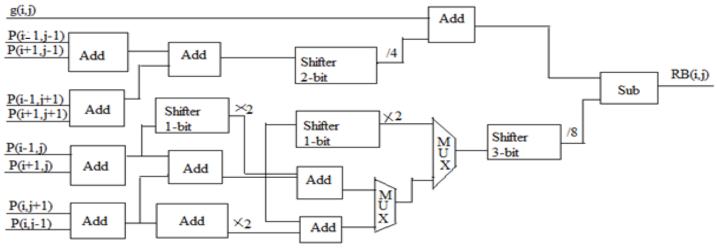


Fig 1.7.1. Architecture of the red and blue colors interpolator model 1 (RB_M1 interpolator).

1.8 SOFTWARE REQUIREMENTS

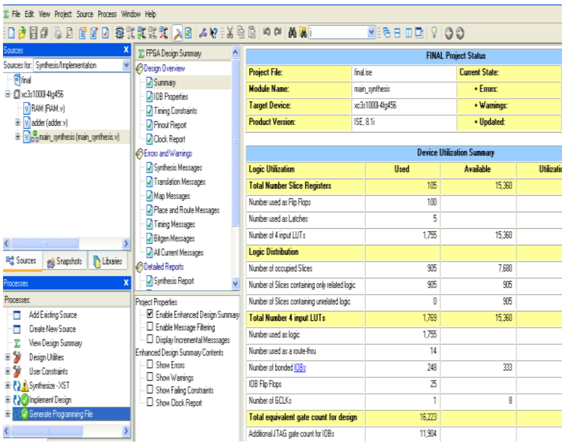
Modelsim SE 6.3f

In this paper the modelsim software is used for simulation and verification. ModelSim is a verification and simulation tool for , Verilog, SystemVerilog, and mixed- language designs. The best standards and platform support in the industry make it easy to adopt in the majority of process and tool flows. Model Technology's award-winning Single Kernel Simulation (SKS) technology enables transparent mixing of Verilog in one design. ModelSim's architecture allows platform independent compile with the outstanding performance of native compiled code. An easy-to-use graphical user interface enables you to quickly identify and debug problems, aided by dynamically updated windows. For example, selecting a design region in the Structure window automatically updates the Source, Signals, Process, and Variables windows. These cross linked ModelSim windows create a powerful easy-to-use debug environment. Once a problem is found, can edit, recompile, and re-simulate without leaving the simulator.

1.9 XILINIX ISE:

Xilinx ISE 8.1i & Xilinx Platform Studio 8.1 are the programming tools for the system. This is used for generate RTL & FPGA schematic. This is used for create logic design like micro blaze design XPS includes a graphical user interface (GUI), along with a set of tools that aid in project design. From the XPS GUI, you can design a complete embedded processor system for implementation within a Xilinx FPGA device.

II. SIMULATION RESULTS



Project Properties
- ☑ Enable Enhanced Design Summary
- ☐ Enable Message Filtering
- ☐ Display Incremental Messages

Enhanced Design Summary Contents
- ☐ Show Errors
- ☐ Show Warnings
- ☐ Show Failing Constraints
- ☐ Show Clock Report

FINAL Project Status

Project File:	final.ise	Current State:
Module Name:	main_synthesis	• Errors:
Target Device:	xc3s1000l-4fg456	• Warnings:
Product Version:	ISE, 8.1i	• Updated:

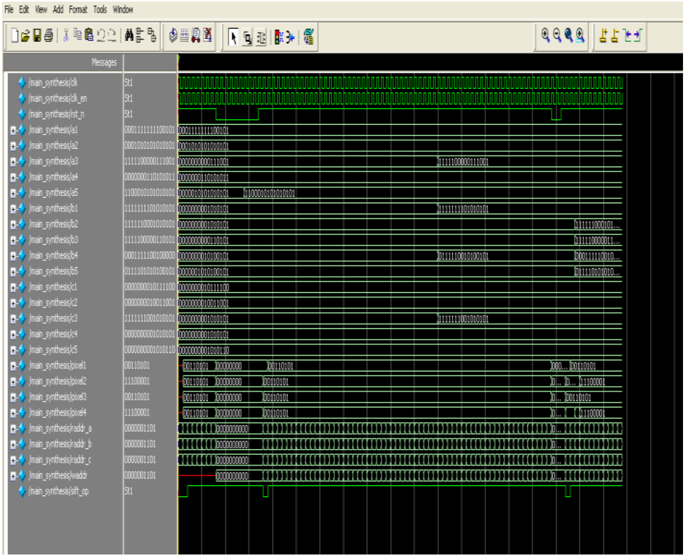
Device Utilization Summary

Logic Utilization	Used	Available	Utilizati
Total Number Slice Registers	105	15,360	
Number used as Flip Flops	100		
Number used as Latches	5		
Number of 4 input LUTs	1,755	15,360	
Logic Distribution			
Number of occupied Slices	905	7,680	
Number of Slices containing only related logic	905	905	
Number of Slices containing unrelated logic	0	905	
Total Number 4 input LUTs	1,769	15,360	
Number used as logic	1,755		
Number used as a route-thru	14		
Number of bonded IOBs	248	333	
IOB Flip Flops	25		
Number of GCLKs	1	8	
Total equivalent gate count for design	16,223		
Additional JTAG gate count for IOBs	11,904		

Fig 2. Total Equivalent gate count

The above figure describes that the total equivalent gate count is reduced upto 16223.

The total power consumption is reduced using CFA Interpolation technique. The total power consumption is reduced upto 4,499.



/main_synthesis/clk	St1
/main_synthesis/clk_en	St1
/main_synthesis/rst_n	St1
/main_synthesis/a1	0001111111100101
/main_synthesis/a2	0001010101010101
/main_synthesis/a3	1111100000111001
/main_synthesis/a4	0000000110101011
/main_synthesis/a5	1100010101010101
/main_synthesis/b1	1111111101010101
/main_synthesis/b2	1111110001010101
/main_synthesis/b3	1111100000110101
/main_synthesis/b4	0001111100100000
/main_synthesis/b5	0111101010100101
/main_synthesis/c1	0000000010111100
/main_synthesis/c2	0000000010011001
/main_synthesis/c3	1111111001010101
/main_synthesis/c4	0000000001010101
/main_synthesis/c5	0000000001010110
/main_synthesis/pixel1	00110101
/main_synthesis/pixel2	11100001
/main_synthesis/pixel3	00110101
/main_synthesis/pixel4	11100001
/main_synthesis/raddr_a	0000001101
/main_synthesis/raddr_b	0000001101
/main_synthesis/raddr_c	0000001101
/main_synthesis/waddr	0000001101
/main_synthesis/sift_op	St1

Fig 2.1 Simulated Output

If the pixel network in coding matches the input in the waveform the output is 1 otherwise the output is 0.

III. CONCLUSION

In this work adaptive edge enhanced color interpolation is developed, Which offers low complexity, low power and low memory requirement and high speed. In this work the area and delay are calculated. The total area and delay are 16223 logic gates and delay of 25.004 ns. This design was synthesized to Xilinx.

IV. ACKNOWLEDGMENT

First and foremost, we wish to express our deep gratitude and indebtness to o institution and our department for providing us a chance to fulfill our long cherished of becoming Electronics and Communication engineers.

We wish to acknowledge with thanks the excellent encouragement given by the management of our college. We wish to express our hearty thanks to the Principal of our college and HOD. We are committed to place our heartfelt thanks to all teaching supporting staff members, lab technicians and friends, and all the noble hearts that gave us immense encouragement towards the bestowing the gifts of life on us and also for providing us the necessary help through his lovely creation in this endeavor of us.

REFERENCES

[1] N. X. Lian, L. Chang, Y. P. Tan, and V. Zagorodnov, "Adaptive filtering for color filter array demosaicking," *IEEE Trans. Image Process.*, vol. 16, no. 10, pp. 2515–2525, Oct. 2007.

[2] K.-H. Chung and Y.-H. Chan, "Color demosaicing using variance of color differences," *IEEE Trans. Image Process.*, vol. 15, no. 10, pp. 2944–2955, Oct. 2006.

[3] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans. Image Process.*, vol. 17, no. 1, pp. 53–69, Jan. 2008.

[4] D. Menon and G. Calvagno, "Regularization approaches to demosaicking," *IEEE Trans. Image Process.*, vol. 18, no. 10, pp. 2209–2220, Oct. 2009.

[5] D. Doswald, J. Hafliger, P. Blessing, N. Felber, P. Niederer, and W. Fichtner, "A 30-frames/s megapixel real-time CMOS image processor," *IEEE J. Solid-State Circuits*, vol. 35, no. 11, pp. 1732–1743, Nov. 2000.

[6] S. C. Hsia, M. H. Chen, and P. S. Tsai, "VLSI implementation of lowpower high-quality color interpolation processor for CCD camera," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 4, pp. 361–369, Apr.2006.

BIOGRAPHY

M .Punitha did her Bachelor of Engineering in Electronics and Communication Engineering at Vickram college of Engineering, Sivaganga and doing Master of Engineering in VLSI Design at Sri Shakthi Institute of Engineering and Technology, Coimbatore, India. Her research interests include Digital Electronics. Presented a paper in International Conference on "VLSI Implementation of Visual Feature Extraction.