International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 4, Issue 2, February 2015

VLSI Implementation of Low Cost Visual Feature Extraction Using parallel SIFT Algorithm

Punitha. M, Anitha. R

***Abstract*— Visual feature extraction with scale invariant feature transform (SIFT) is widely used for object recognition. Scale-invariant feature transform or (SIFT) is an algorithm in computer vision to detect and describe local features in images. SIFT detects and uses a much larger number of features from the images, which reduces the contribution of the errors caused by these local variations in the average error of all feature matching errors. SIFT can robustly identify objects even among clutter and under partial occlusion, because the SIFT feature descriptor is invariant to uniform scaling, orientation, and partially invariant to affine distortion and illumination changes. However, its real-time implementation suffers from long latency, heavy computation, and high memory storage because of its frame level computation with iterated Gaussian blurs operations. a layer parallel SIFT (LPSIFT) with integral image, and its parallel hardware design with an on-the fly feature extraction. Feature extraction involves reducing the amount of resources required to describe a large set of data. Analysis with a large number of variables generally requires a large amount of memory and computation power or a classification algorithm which over fits the training sample generalizes poorly to new sample.**

***Index Terms*— Feature extraction, LPSIFT, SIFT, Keypoint localization.**

I. Introduction

Scale-in variant feature transform (SIFT) is an algorithm in computer vision to detect and describe local features in images. For any object in an image, interesting points on the object can be extracted to provide a "feature description" of the object. This description, extracted from a training image, can then be used to identify the object when attempting to locate the object in a test image containing many other objects. To perform reliable recognition, it is important that the features extracted from the training image be detectable even under changes in image scale, noise and illumination. Such points usually lie on high-contrast regions of the image, such as object edges [3]. However, in practice SIFT detects and uses a much larger number of features from the images, which reduces the contribution of the errors caused by these local variations in the average error of all feature matching errors.

SIFT keypoints of objects are first extracted from a set of reference images and stored in a database. An object is recognized in a new image by individually comparing each feature from the new image to this database and finding candidate matching features based on Euclidean distance of their feature vectors. From the full set of matches, subsets of keypoints that agree on the object and its location, scale, and orientation in the new image are identified to filter out good matches.

1.1 Layer Parallel SIFT (LPSIFT)

Layer parallel SIFT (LPSIFT) with integral image and its parallel hardware design for we adopted the layer parallel restricted box kernel to replace iterated Gaussian blur operations. The computation of box kernel was further simplified by the integral image approach with Reuse of sub-kernel sum. With this, the latency of the first feature point was reduced to several image lines (depending on the location of that feature point) from several frames.

1.2 SIFT ALGORITHM

SIFT is an approach for detecting and extracting local feature descriptors from images In order to generate the set of image features, the algorithm consists of five major computation steps: Building Gaussian Scale Space, Key Point Detection and Localization, Orientation Assignment, Key Point Descriptor .

1.3 Keypoint Localization

We simplified the complex low contrast analysis to be a low brightness test. For hardware design, we adopted the on-the fly feature extraction flow so only partial temporal results have to be stored. . Furthermore, the costly inverse square root and divider in keypoint localization were implemented by a low cost universal operation unit with precision equivalent cycle (PECs) to reduce the gate count.

1.4 Orientation Assignment

In the orientation assignment step, each keypoint is assigned one or more orientation based on local image gradient direction to achieve invariance to rotation. In which, SIFT selects the largest, s-vector, to represent the orientation assignment. These steps ensure invariance to image location, scale and rotation.

1.5 Keypoint Descriptor

The local image gradients are measured at the selected scale in the region around each keypoint. These are transformed into a representation the Fallows for significant levels of local shape distortion and change in illumination.

1.6 Gaussian Scale Space

Key point candidates for SIFT features are obtained potentially from local extrema of difference-of-Gaussian (DoG) pyramid. The scale space pyramid (octave) is constructed by convolving the initial image with Gaussian kernels repeatedly. After that, the first image in this octave is downscaled by a factor of two and convolved with the same set of Gaussian kernels to construct the next octave until a certain image size is reached.

1.7 SIFT ARCHITECTURE

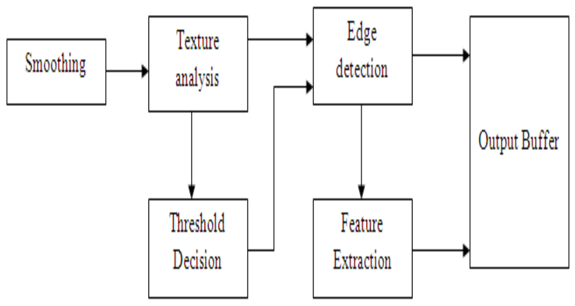


Fig. 1.7 SIFT architecture diagram.

Fig 1.7 shows the SIFT 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).

1.8 SIFT ANALYSIS AND LPSIFT DESIGN

Design analysis of SIFT

The design analysis of SIFT for parameters [octave, scale] = [2, 4].In this table, A is the number of pixels in a frame and its first level down sampled frame defined a

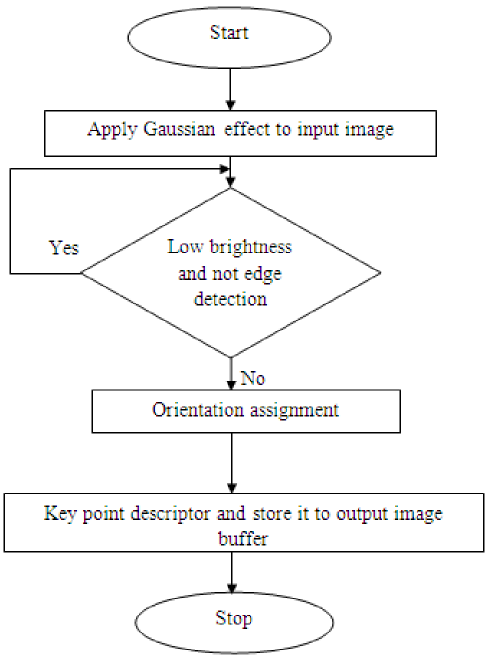
$$A = WH + \frac{1}{4}WH = 1\frac{1}{4}WH$$

Where W and H are width and height of the original frame. B represents the number of feature points. From this table, we can find that the scale-space extrema detection and the keypoint localization contribute to most of overall computation, and require a lot of memory to save intermediate data frames. In addition, the keypoint localization, orientation assignment, and the local frame descriptor calculation use a lot of dividers to compute Taylor expansion and normalization. This causes hardware implementation difficult. Thus, how to solve these problems is an important issue for SIFT hardware design. Another problem for SIFT hardware is its highly data dependent computational structure. In the data flow of SIFT, a new scale image has to wait for the completion of the previous scale image in the Gaussian pyramid. Thus, different scale images are computed sequentially. This results in long latency and prohibits the following DoG computation in parallel that needs at least three scale images at the same time to calculate the candidates of features points. Thus, these intermediate scale images have to be stored until completing DoG computations, which results in large memory storage.

To assist this parallel computation while reduce the computational complexity, we use the integral image approach to create a box kernel with the response similar to that of the Gaussian filter so as to keep the feature matching performance as close as that of SIFT but without complex Gaussian filters. By adopting the integral image approach for multiple box kernels in each scale, we reduce computational complexity significantly and enable parallel computation for all scales.

1.9 LPSIFT ALGORITHM

1.9 Proposed LPSIFT algorithm


Start
Apply Gaussian effect to input image
Yes
Low brightness and not edge detection
No
Orientation assignment
Key point descriptor and store it to output image buffer
Stop

To solve the data dependency problem, which not only calculate Gaussian pyramid and DoG pyramid at the same time but also calculate keypoint candidates proposed LPSIFT algorithm, which is based on SIFT with three major modifications: fast scale space extreme detection with layer parallel Gaussian pyramid and integral image, and simplified keypoint localization with a brightness threshold. In this flow, proposed a layer parallel algorithm simultaneously. This parallel computation reduces storage and latency from multiple whole images to several image rows. To assist this parallel computation while reduce the computational complexity, we use the integral image approach to create a box kernel with the response similar to that of the Gaussian filter so as to keep the feature matching performance as close as that of SIFT but without complex Gaussian filters. By adopting the integral image approach for multiple box kernels in each scale, we reduce computational complexity significantly and enable parallel computation for all scales. Another problem is the high complexity of Taylor expansion. It proposes a low brightness method instead of the low contrast method to reduce complexity.

Fast Scale Space Extreme Detection with Layer Parallel Gaussian Pyramid and Integral Image: Fig. 1.9 shows the concept of the layer parallel Gaussian pyramid.

$$\begin{aligned} G[m,k] &= GF(x,y,k\sigma) * G[m,k-1] \\ &= (GF(x,y,k\sigma) * GF(x,y,(k-1)\sigma) \cdots \\ &\quad *GF(x,y,\sigma)) * G[m,0] \\ &= GF'(x,y,k) * G[m,0] \\ &\qquad \text{for positive integer } m, \text{ and } k. \end{aligned}$$

With this independent Gaussian filtering, we can compute all scale layers in parallel and generate the DoG images on demand. However, the merged kernel will become larger with more kernels merged as shown in Fig. 1.9 which will demand high computational complexity and storage. To solve this, we use the integral image concept that can compute this large kernel with constant time. By using integral image, no matter what the kernel size is, we need only four points of information and three additions, and reduce a lot of computation. With this simple computation, we can compute all scale layers in parallel without worry about the heavy complexity and storage due to layer parallel computation. The integral image works best for the box kernel, but the box kernel results in inferior matching performance. Thus, we use restructured box kernels to approach the performance by the Gaussian filter. Take the first scale 5 ×5 box kernel in Fig. 4(c) as an example. Simplified Keypoint Localization with a Brightness Threshold: SIFT uses Taylor expansion to exclude the low contrast candidates, but Taylor expansion needs complex computation. In practical, the low contrast candidates almost have low brightness, too. Therefore, low brightness candidates can be excluded according to this threshold.

Threshold $_{\text{new}}$ **=**$(2^{\text{precision of DoG}}$ **-1)*coefficient**$_{\text{bright}}$

2. STAGE ONE WITH ON-THE-FLY COMPUTATION SCHEDULING

Fig 2.shows the architecture of the stage one, which applies the noise smoothing to the input image and builds the integral image for following filtering operations. With enough rows of integral image, begin to compute scale images and corresponding DoG images to find feature point candidates at the same time.

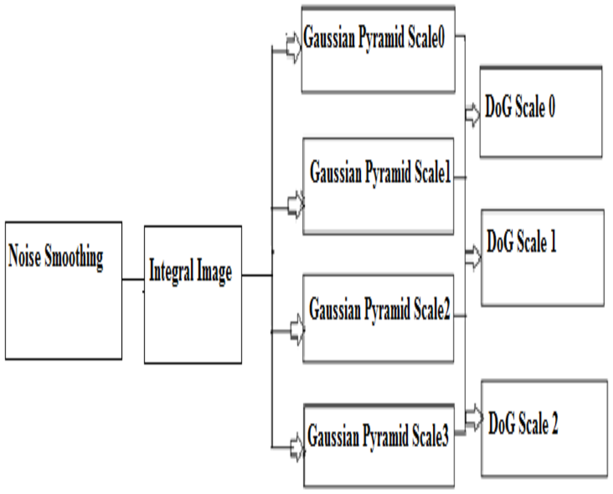


Fig 2. Architecture of the stage one

Fig 2.shows the architecture of the stage one, which applies the noise smoothing to the input image and builds the integral image for following filtering operations. With enough rows of integral image, begin to compute scale images and corresponding DoG images to find feature point candidates at the same time.

2.1 STAGE TWO ARCHITECTURE

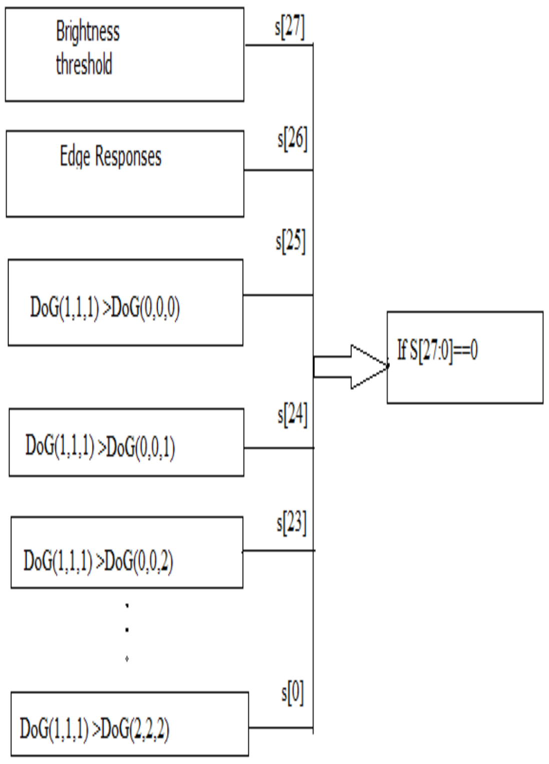


Fig 2.1 Hardware architecture of the keypoint localization.

Fig. 2.1 shows the design of the keypoint localization after obtaining the DoG. We can compute the feature point candidate by examining conditions such as the local maximum and minimum value, low brightness and edge detection. Since all these conditions are independent to each other, we can compute each condition simultaneously with a flag bit to denote if it matches the feature point criteria. If all flag bits are zeros, In this design, 19×19 pixels of integral image will be fetched again to re-compute 15×15 information of scale 1 since no scale image is stored. In this recomputation step, we will compute one row at a time, and store the results at the buffer.

2.2 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. Unified mixed language simulation engine for the fastest regression suite throughput. Native support of Verilog, SystemVerilog for design, and System C for effective verification of the most sophisticated design environments. Fast time-to-debug causality tracing and Multilanguage debug environment. The basic simulation flow is given by working libraries, Compile design files, Load and run simulation, Debug results. ModelSim offers numerous tools for debugging and analyzing the design. . The project flow for Modelsim is given by Create a project, add files to the project, compile the project, run the simulation, Debug the results. Advanced code coverage and analysis tools for fast time to coverage closure. ModelSim uses libraries in two ways: 1) as a local working library that contains the compiled version of other design; 2) as a resource library. The contents of your working library will change as you update your design and recompile. A resource library is typically static and serves as a parts source for other design. It can create it's own resource libraries, or they may be supplied by another design team or a third party.

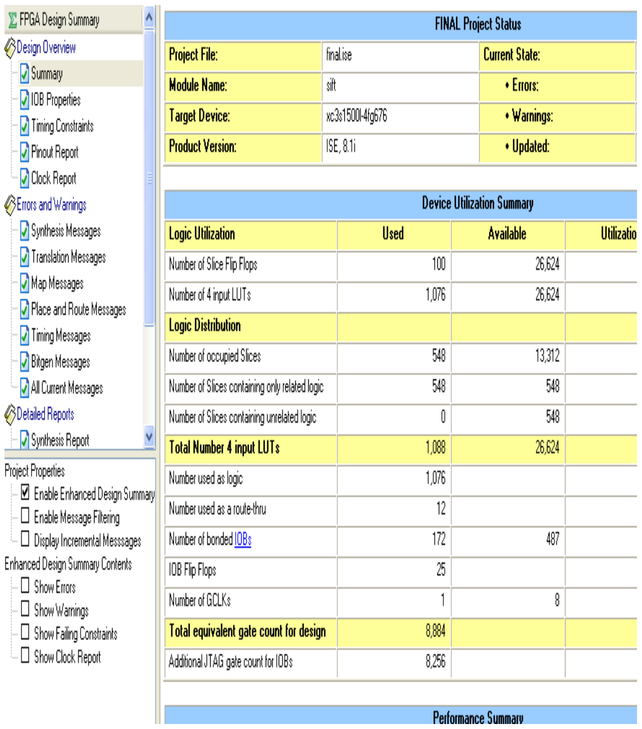
2.3 XILINIX ISE

Xilinx ISE 7.1i & Xilinx Platform Studio 7.1 are the programming 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.

Xilinx Platform Studio (XPS)

XPS is a part of the Xilinx Embedded Development Kit Kit (EDK) and includes the Xilinx Platform Studio (XPS) GUI and all tools run by the GUI to process hardware and software system components.

2.4 RESULT ANALYSIS OF SIFT ARCHITECHTURE



FPGA Design Summary
Design Overview
- ✓ Summary
- ✓ IOB Properties
- ✓ Timing Constraints
- ✓ Pinout Report
- ✓ Clock Report

Errors and Warnings
- ✓ Synthesis Messages
- ✓ Translation Messages
- ✓ Map Messages
- ✓ Place and Route Messages
- ✓ Timing Messages
- ✓ Bitgen Messages
- ✓ All Current Messages

Detailed Reports
- ✓ Synthesis Report

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:	sift	• Errors:	
Target Device:	xc3s1500l-4fg676	• Warnings:	
Product Version:	ISE, 8.1i	• Updated:	

Device Utilization Summary			
Logic Utilization	Used	Available	Utilizatio
Number of Slice Flip Flops	100	26,624	
Number of 4 input LUTs	1,076	26,624	
Logic Distribution			
Number of occupied Slices	548	13,312	
Number of Slices containing only related logic	548	548	
Number of Slices containing unrelated logic	0	548	
Total Number 4 input LUTs	1,088	26,624	
Number used as logic	1,076		
Number used as a route-thru	12		
Number of bonded IOBs	172	487	
IOB Flip Flops	25		
Number of GCLKs	1	8	
Total equivalent gate count for design	8,884		
Additional JTAG gate count for IOBs	8,256		

Performance Summary

Fig 2.4 Total equivalent gate count

The total equivalent gate count of the design is reduced using Layer parallel Scale Invariant Feature Transform (LSIFT) technique. The above figure describes that the total equivalent gate count is reduced upto 8,88.

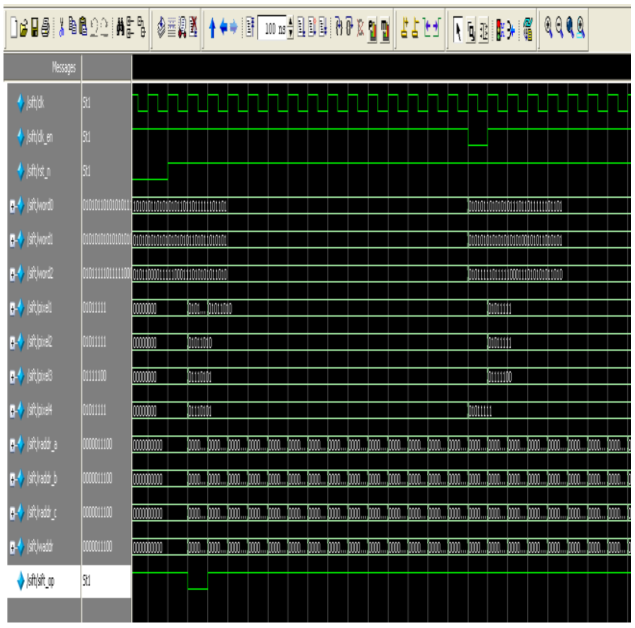
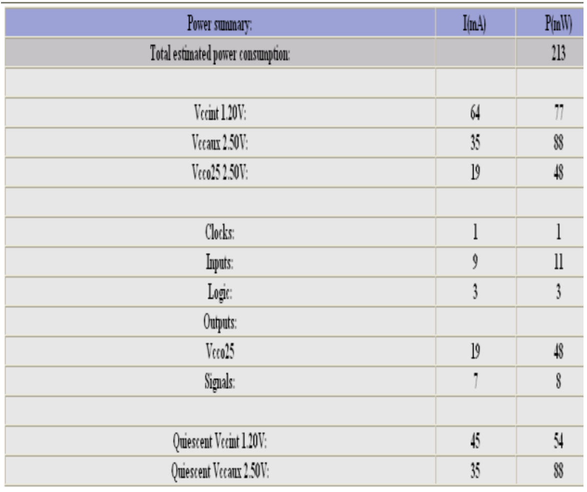
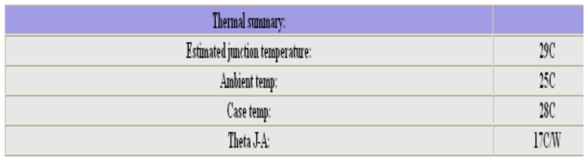


Fig 2.5 Simulated Output For SIFT Algorithm

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



Power summary:	I(mA)	P(mW)
Total estimated power consumption:		213
Vccint 1.20V:	64	77
Vccaux 2.50V:	35	88
Vcco25 2.50V:	19	48
Clocks:	1	1
Inputs:	9	11
Logic:	3	3
Outputs:		
Vcco25	19	48
Signals:	7	8
Quiescent Vccint 1.20V:	45	54
Quiescent Vccaux 2.50V:	35	88



Thermal summary:	
Estimated junction temperature:	29C
Ambient temp:	25C
Case temp:	28C
Theta J-A:	17C/W

Fig 2.8 Power Analysis

The total power consumption is reduced using layer parallel scale invariant feature transform (LSIFT) technique. The above figure describes that the total power consumption is reduced upto 213.

2.5 CONCLUSION

When SIFT algorithm is used in real time image processing application it suffers from long latency, heavy computation and high memory storage. So proposed system uses layer parallel SIFT algorithm which reduces the computational and memory. With these technique, the presented design easily achieves the real time demand with significantly lower cost, which saves 88% count and memory cost. This also reduces the latency to a few image lines instead of several frames. Also by using Xilinx instead of modelsim hardware complexity and latency is reduced.

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] Bay H., Tuytelaars T., and Gool L.J. V., 2008 “SURF: Speeded up robust features,”in Proceedings European Conference. on Computer Vision., ,pp. 404-417.

[2] Bonato V., Marques E., and Constantinides G. A., 2008 “A parallel hardware architecture for scale and rotation invariant feature detection,” IEEE Transaction. Circuits System. Video Technology. volume. 18, no. 12, pp. 1703–1712,.

[3] Cornelis N., and Van Gool L., 2008 “Fast scale invariant feature detection and matching on programmable graphics hardware,” in Proc. IEEE Computer Society Conference Compuert. Vision. Pattern Recognitation. Workshops,. pp. 1–8.

[4] Derpanis K.G., Leung E.T., and Sizintsev M., 2007, “Fast scale-space feature representations by generalized integral images,” in Proceedings. IEEE International Conference. Image Process., pp. 521–524.

[5] Feng H., Chen., E. Li, Y., and Zhang Y., 2008 “Parallelization and characterization of SIFT on multi-core systems,” in Proceedings. IEEE International Symposium. Workload Characterization., , pp. 14–23.

[6] Grabner M., Grabner H.,and Bischof H., 2006 “Fast approximated SIFT,” Proceedings. Asian Conference. Computer Vision., pp. 918–927.

[7] Heymann S., Mülle K.r, Smolic A., Froehlich B., and Wiegand T., 2007, “SIFT implementation optimization for general-purpose GPU,” in Proceedings. International Conference Central Europian. Computer. Graph., Visualization. Computer. Vision., pp. 144–159.

[8] Hsu P.H., Tseng Y.C., and Chang T.S., 2010, “ Low memory cost bilateral filtering using stripe-based sliding integral histogram,” in Proceeding. IEEE International. Symposium. Circuits Systems., pp. 3120–3123.

[9] Huang F.C., Huang S.Y., Ker J.W., and Chen Y.C., 2012. “High-performance SIFT hardware accelerator for real-Time image feature extraction,” IEEE Transaction Circuits Systems. Video Technology., volume. 22, no. 3, pp. 340–351.

[10] Kim D., Kim K., Kim J.Y., Lee S., Lee S J., and YooH.J. , 2008 “Vision platfrom for mobile intelligent robot based on 81.6 GOPS object recognition processor,” in Proceedings. 45th ACM/IEEE Design Autom. Conferance, ., Jun. pp. 96–101.

[11] D.G., 2004. “Distinctive image feautres from scale-invariant keypoints,” Int. J. Comput. Vision., volume 60, no. 2, pp. 91–110,

[12] Sawasaki N., Nakao M., Yamato Y., and Okabayahi K., 2006, "Embedded vision system for mobile robot navigation," in Proceedings International Conference. Robot Autom., pp. 2693–2698.

[13] Terriberry T., French L., and Helmsen J., 2008, "GPU accelerating speeded-up robust features," in Proceedings 4th International Symposium. 3D Data Process.Visualization. Transmisson., pp. 1–8.

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. She has Attended one day workshop on ASIC Design and verification conducted by Vinchip system. Attended two days workshop on Device Modeling and Simulation at KPR institution of engineering and technology, Coimbatore.

R .Anitha did her Bachelor of Engineering in Electronics ınd communication Engineering at Karpagam college of ɜngineering, Mileripalayam .Her completed Master of Engineering in VLSI Design at Sri Shakthi Institute of Engineering and Technology,Coimbatore, India. Presented ı paper in International Conference on" FPGA mplementation of Wu-Manber Algorithm For BLASTN DNA Sequence Matching". Presented a paper in a National Conference on" Reduction in crosstalk and low power RC coupled VLSI Interconnects"