ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

A Comparative Study of Design of Hough Transform Implementation with two Different Methods

Vijaykumar S. Kawde
PG student, Electronics Department, WCE, Sangli, India
kawdevijay@gmail.com

Abstract

Line detection is very important task in image processing field. It is mainly used in auto focusing camera input sensors. Many techniques used for line detection but gives improper result if noise is present in an input image. Hough transform is famous technique present for line detection which gives proper results in presence of noise and discontinuous shapes in an image. This paper gives comparative study of two techniques of hough transform, both the methods are simulated in Xilinx 13.2 design suite which is popularly used for FPGA hardware implementation and compared there simulation results. First method is general hough transform, this is very basic method of hough transform in which binary feature image is given as input and vote procedure of hough transform is performed over it. Second method is given by Zhong-Ho Chen, Alvin W. Y. Su, and Ming-Ting Sun in august 2012, this method use incrementing property of hough transform and process all pixels in parallel. Also locality of hough transform is used to reduce the size of vote memory.

Keywords: Voting procedure, incremental and locality property, Xilinx simulation.

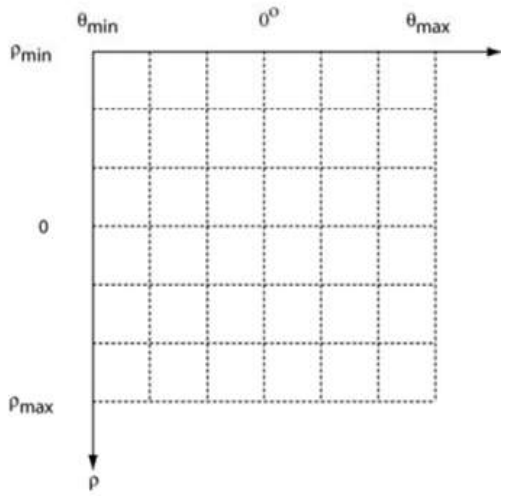
Introduction

Cartesian and polar these are two types of hough transform. But in cartesian type, when line is parallel to co-ordinates then slope becomes infinite so polar type of hough transform is mostly used in image processing.

In polar type x-y plane is transformed into ρ-θ plane by using equation one

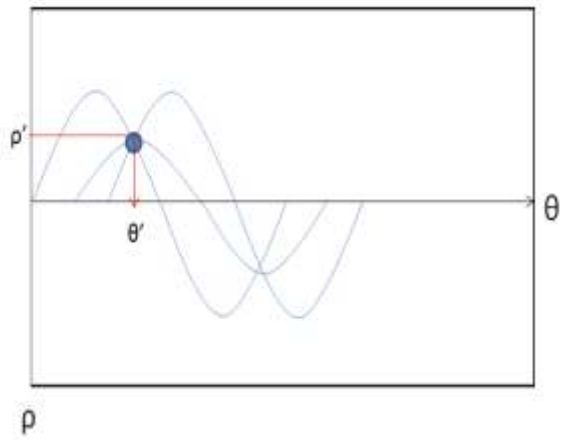
$$\rho = X*\cos\theta + Y*\sin\theta \quad (1)$$

Where ρ is perpendicular distance from origin and θ is angle made by it with x co-ordinate. Each point in the x-y plane gives a sinusoidal curve in the ρ-θ plane. N number of collinear point lying on the line in X-Y plane will give N curves that are intersects at point (ρ, θ) in the parameter (ρ-θ) plane. Parameter space is divided into accumulator cells and each (ρ, θ) value of parameter space is considered as vote.
One particular cell of parameter space getting maximm votes gives us the parameters (ρ and θ) of line present in input image. [2]



Parameter space with accumulator cells

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852



ρ-θ plane

General hough transform

Methodology

Preprocessing on input image has to be done in MATLAB and input colour image is converted into gray binary feature image. Input specification of an image is observed from MATLAB and used in simulation. Xilinx 13.2 design suite used for simulation of both methods, verilog language used for coding because it is more C based and no need of GATE packages.

Hough matrix is memory which stores all votes and is created using register bank. Input image is also stored in register bank memory (memory within LUT of FPGA) [10] [2]. Then voting procedure of hough transform is performed, local maxima of hough matrix memory is observed which gives us the parameters of the line in given input image. Flow chart for steps of simulation of general hough transform is given below in results.

Hough transform with incremental property and locality property

Methodology

Block divdation and rle used for each block

Input image is divided into blocks and blocks containing all zero pixels is skipped in processing, because line is pass through only nonzero pixels. Then run length encoding is applied for each block to get Rb, Code and Zl values, Rb is used to indicate beginning of block-row in row x col image, Code part represents the pixel values in a block and Zl is the number of successive zero-blocks after the current block, used to avoid zero blocks. These Rb, Code and ZL are used in incremental and locality property [1] .

Incremental and locality property

ρ for pixel present at (x, y) in X-Y plane is given by equation (1). Another pixel present at (X+dx, Y+dy) in X-Y plane ρ for it is given by

$$\rho\theta_{(x+dx,\ y+dy)} = \rho\theta_{(x,\ y)} + dx*\cos\theta + dy*\sin\theta \quad (2)$$

block size is fixed so dx and dy of equation (2) are constants and are precomputed and stored in register bank memory. Difference among ρ values for every pixel in a block is small. Pixels giving same ρ value can be jointly accumulated in hough memory. ρ value of first pixel in block p0(x, y) is

$$\rho(p0) = i0 + f0 \quad (3)$$

where i0 is integer part and f0 is fractional part. ρ value pixel p1 at (x+dx, y+dy) is given by

$$\rho(p1) = i0 + (f0 + dx*\cos\theta + dy*\sin\theta) \quad (4)$$

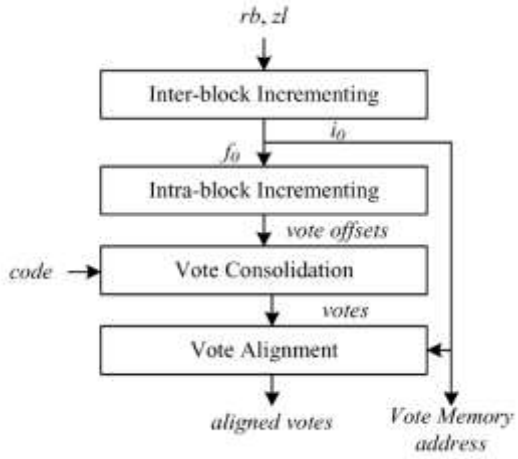
where (f0 + dxcosθ + dysinθ) Vθ is called vote offset which gives difference of ρ(p0) and ρ(p1) [1]. Flow chart for steps of simulation is given below in results.

Vote consolidation

for block size M by N and given θ

$$V\theta = f0 + (M-1)*\cos\theta + (N-1)*\sin\theta \quad (5)$$

for 2×4 block Vθ is in between 0 to 4. So max 5 different integer values are present in a block. Vote consolidation consolidates eight votes from eight pixels into five consolidated votes. Memory access is reduced because votes are jointly stored [1].

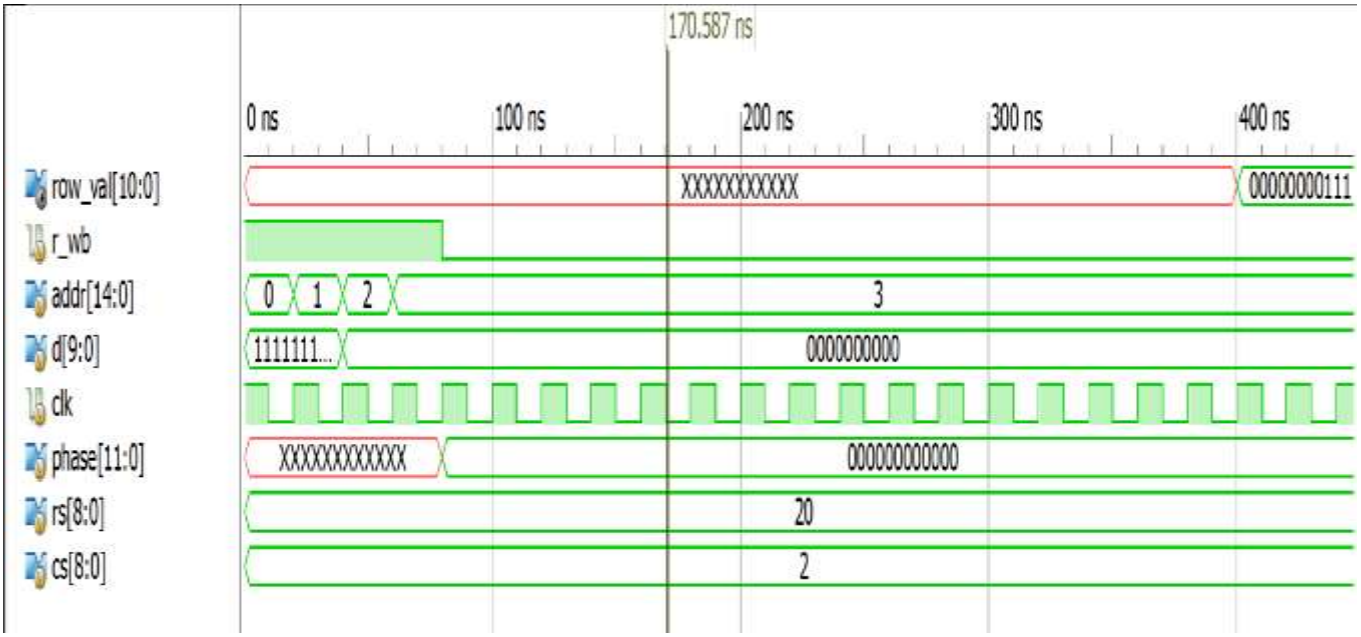


System architecture

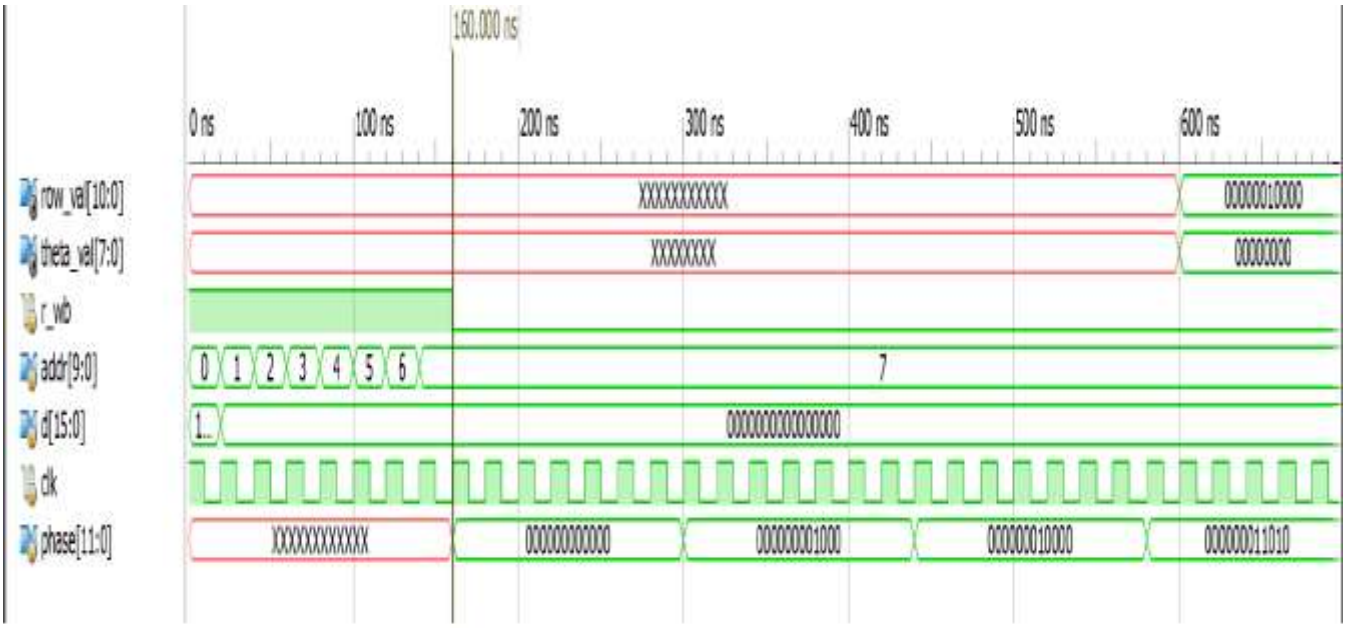
ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

Results and discussion

Simulation result of general hough transform:

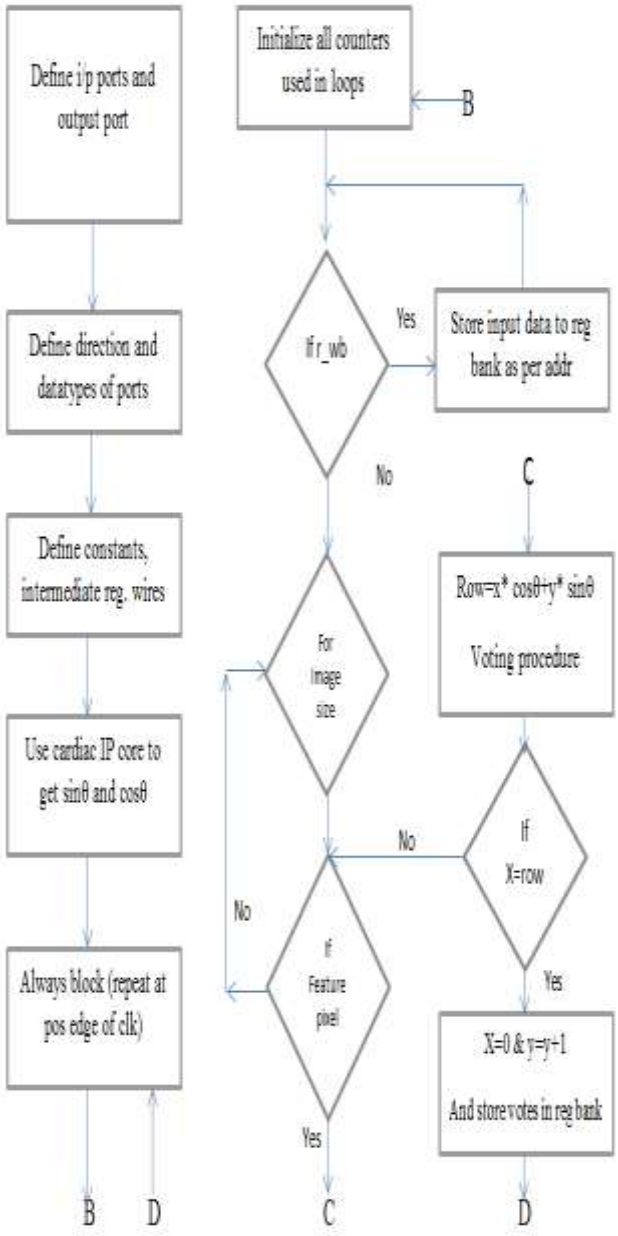


Simulation result of hough transform with incremental property and locality property:

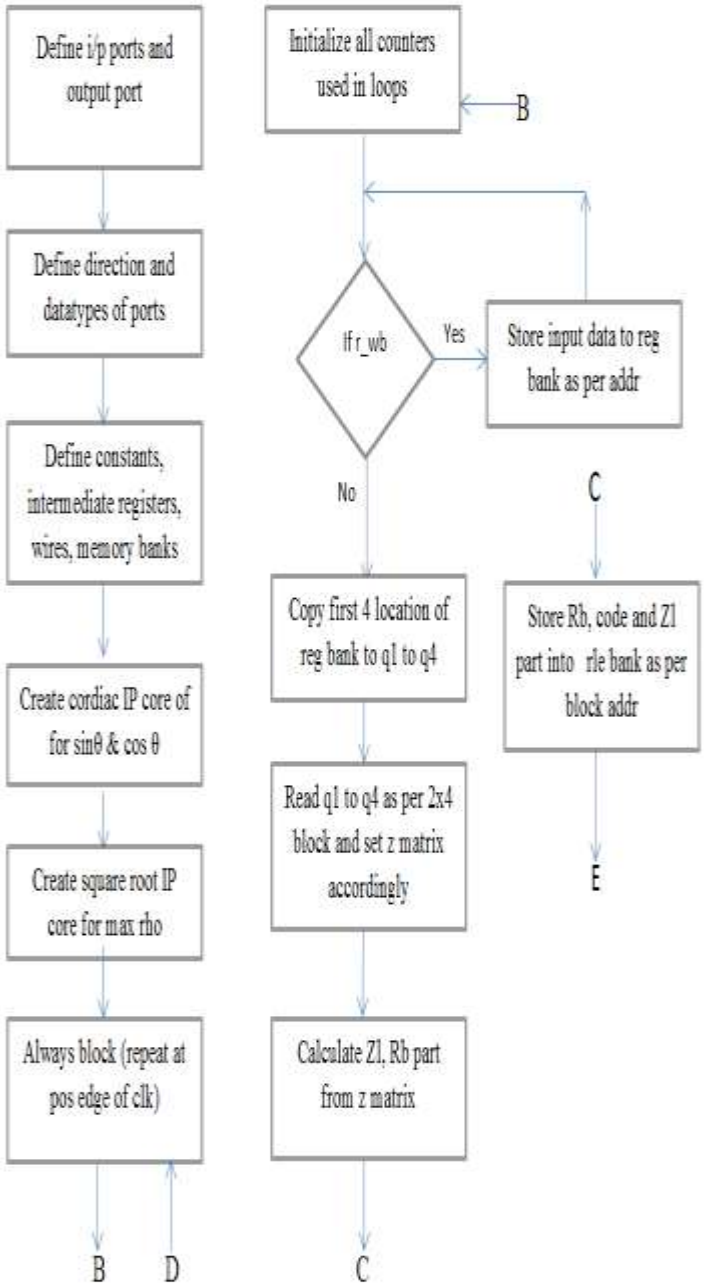


ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

Flowchart for general hough transform:

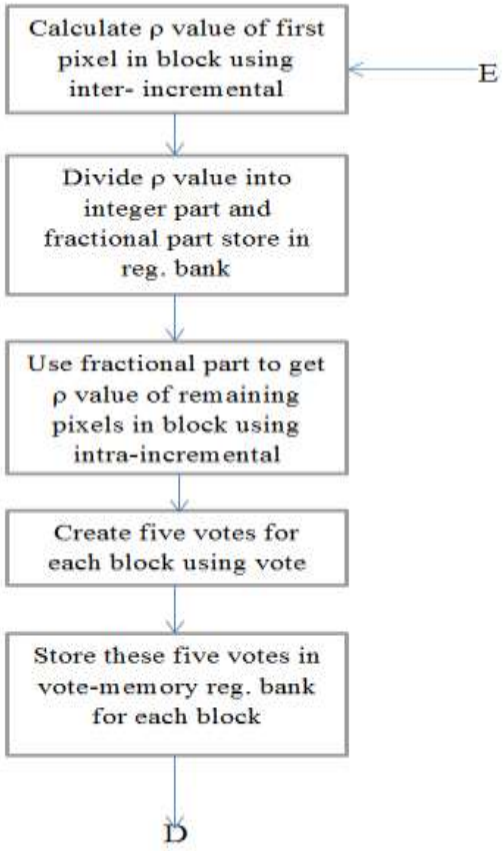


Flowchart 1 for hough transform with incremental property and locality property:



Flowchart 2 for hough transform with incremental property and locality property:

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852



Tables:

Table 1. Comparison table

Parameters	General hough transform	Hough transform using incremental and locality property
Image size processed at 300 MHZ	20 X 2	16 X 8
Memory usage	2612912 kilobytes (for 20X2)	2131416 kilobytes (for 16X8)
Total CPU time	946.77 secs	28 secs

Discussion:

Simulation of general hough transform is done using Xilinx 13.2 design suite and code written in verilog is synthesized. Phase is input angle to system ranges from -90° to +90°, d is input image data stored in register bank, input image size is 20X2, addr is input address of register bank memory which stores input image data. r_wb, clk are controlling signals and row_val is output of system which gives final rho (ρ) value getting maximum votes in hough vote memory for given phase.

Simillarly simulation of hough transform with incremental property and locality property is done using Xilinx 13.2 design suite and code written in verilog is synthesized. Input image size is 16X2. Table 1 gives comparision of synthesis reports of both methods which is generated after synthesis of both method's code.

General hough transform is basic type of hough transform used for line detection thus this method is selected as first method. Hough transform with incremental property and locality property processes all pixels in block parallely and also avoids zero pixels. It also do angle level parallelism by using same hardware for all angles (-90° to +90°).Thus this mehod is selected as second method

Conclusion

Hough transform gives robust line detection even if noise is present or shapes are discontinuous. This paper mainly focuses on comparison of designs used for hardware implementation of general hough transform and hough transform with incremental property and locality property. This paper compares Xilinx simulation results and synthesis report of both method's code synthesised in Xilinx.

General hough transform processes 40 pixels at 300 MHZ while hough transform with incremental property and locality property processes 48 pixels in same time. General hough transform method requires 2612.9 MB distributed memory to perform hough transform of 20X2 image. While hough transform with incremental property and locality property takes 2131.4 MB distributed memory to perform hough transform of 16X8 image. Synthesis of general hough transform in Xilinx takes 946.77 sec CPU time and synthesis of hough transform with incremental property and locality property code takes only 28 secs. So we may observe that hough transform with incremental property and locality property require much less memory and very less processing time than general hough transform.

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

References:

[1] *Zhong-Ho Chen, Alvin W. Y. Su, and Ming-Ting Sun,"Resource-Efficient FPGA Architecture and Implementation of Hough Transform," IEEE Transactions on VLSI systems, vol. 20, no. 8, August 2012.*

[2] *Vijaykumar S. Kawde and Jagdish shete, "Line detection using Hough transform technique with two different approaches," International Journal of Engineering Associates (ISSN: 2320-0804) # 59 / Special Issue -1 (Volume 3)*

[3] *R. Rajbanshi, Q. Chen, A.Wyglinski, G. Minden, and J. Evans, "Quantitative comparison of agile modulation technique for cognitive radio tranceivers," in Proc. IEEE CCNC, Jan. 2007, pp. 1144–1148.*

[4] *Xin Zhou, Yasuaki Ito, and Koji Nakano, "An FPGA Implementation of Hough Transform using DSP blocks and block RAMs", Bulletin of Networking,Computing, Systems, and Software, ISSN 2186–5140,Volume 2, Number 1, pages 18–24, January 2013.*

[5] *Karthik Mayasandra, Hanif M. Ladak, Wei Wang "A Distributed Arithmetic Hardware Architecture for Real-Time Hough Transform Based Segmentation", 0-7803-8886-0/05/$20.00 ©2005 IEEE.*

[6] *Samir Tagzout, Karim Achour and Oualid Djekoune, "Hough Transform Algorithm for FPGA Implementation, " 0-7803-6488-0/00/$10.00 ©2000 IEEE.*

[7] *Costin-Anton Boiangiu, Bogdan Raducanu, "Robust Line Detection Methods", 9th WSEAS International Conference on AUTOMATION and INFORMATION (ICAI'08), Bucharest, Romania, June 24-26, 2008 ISBN: 978.*

[8] *O.Djekoune , K. Achour, M. Halimi and S. Kahlouche, "Incremental Hough Transform: an Improvement Algorithm for Digital Devices Implementation", Advanced Technologies Development Center, Algeria*

[9] *C. V. Hari, "Performance Evaluation of Different Line Detection Algorithms", International Journal of Modeling and Optimization, Vol. 2, No. 4, August 2012.*

[10] *Jagdish shete, Sunil kore and Vijaykumar S. Kawde, "Pipelined VLSI Architecture for CDF (2, 2) Lifting Based 1-D Discrete Wavelet Transform,International Journal of Engineering Associates (ISSN: 2320-0804) # 45 / Special Issue -1 (Volume 3)NC-EMPIRES.*

[11] *Subbiah, A.Rega, "Implementation of Hough Transform Using Resource Efficient FPGA Architecture, " International Journal of Advanced Information Science and Technology (IJAIST), ISSN: 2319:2682, Vol.14, No.14, June 2013.*

[12] *F. Zhou and P. Kornerup , "A high speed Hough transform using CORDIC," Univ. Southern Denmark, Tech. Rep. PP-1995-27, 1995.*

[13] *Rafael C. Gonzalez, Richard E. Woods,"Image Segmentation", in Digital Image Processing, 3rd edition.*

[14] *J. D. Bruguera, N. Guil, T. Lang, J. Villalba, and E. L. Zapata , "Cordic based parallel/pipelined architecture for the Hough transform," J. VLSI Signal Process., vol. 12, no. 3, pp. 207–221, 1996*

Author Biblography

Vijaykumar S. kawde
Author received B.E. in E&TC from university of pune. Currently he is perceiving M.Tech in Electronics from WCE, Sangli. His research field is DSP-VLSI.
Email: kawdevijay@gmail.com