

CAD for VLSI 1

Project - Hough Transform

Problem

In this project you will design a system that will perform Hough Transform and detect lines in a given image. Hough Transform uses parametric descriptions of a line to find out if the given image has lines in it. The algorithm transfers cartesian co-ordinates (x, y values), to polar co-ordinates(r, θ). The equations for a straight line are as below:

$$\text{Linear Model : } y = a \cdot x + b \quad (1)$$

$$\text{Parametric Model : } r = x \cdot \cos\theta + y \cdot \sin\theta \quad (2)$$

In the equations above, r represents the perpendicular distance of the line from the origin, and θ represents the orientation of the line from the X-Axis measured anti-clockwise. A line passing through the point (x_1, y_1) in cartesian space represents a sinusoidal curve $r = x_1 \cdot \cos\theta + y_1 \cdot \sin\theta$ in the parameter space (r, θ) . Collinear points on the binary image correspond to intersections of sinusoids on the parameter space.

Hough transform uses a parametric matrix P . The rows of P correspond to the range of r and the columns correspond to the range of θ . Ideally, the ranges must be

$$-\sqrt{N_1^2 + N_2^2} \leq r \leq \sqrt{N_1^2 + N_2^2} \quad (3)$$

$$\frac{-\pi}{2} \leq \theta < \frac{\pi}{2} \quad (4)$$

for a binary image of size $N_1 \times N_2$. However, we cannot represent the whole continuous space required for both r and θ . Usually, r is incremented in units of 1 and θ is increased in some units of degrees.

Refer to the notes for the algorithm performing Hough Transform. The notes are taken from the book titled "Digital Image Processing Algorithms" by Ioannis Pitas, Prentice Hall, ISBN : 0131458140. The algorithm iterates over the two dimensions of the image. For all pixels that are above a specific threshold (color value), the algorithm finds out what possible values of r and θ can the pixel be in. If a pixel at (x_1, y_1) satisfies the parametric form for some r_1, θ_1 , then the entry $P[r_1][\theta_1]$ is incremented by 1. After all the pixels are tested, the array is scanned for peaks. Since collinear points result in peaks in the matrix P , we will have large peaks if there are lines, and very low values if there is no line. We end up with the peaks denoting lines in the image.

Requirements and Constraints

You have to design a machine that uses Hough Transform to detect lines in an image. The colors in the image are in the range 0 to 255. The image size is fixed at 30×30 . The design must take in the image one pixel at a time.

1. The design must be a parallel hardware design. You cannot use instruction set based implementation for Hough Transform.
2. You should have increments of 5 degrees for θ and increments of 1 pixel for r .
3. The threshold for pixels are set at 128. This means only pixels of value greater than 127 can form the lines. Even if there are collinear points all having pixel values less than 128, we don't detect a line.
4. Simulate your design for the following images :
 - (a) An image with a line in the main diagonal with all pixels bigger than 128 in value.
 - (b) A horizontal line at $y=15$.
 - (c) A vertical line at $x=15$.
5. The design must be synthesizable. Synthesize the design for the Flex family and by using Magma. Optimize your design for area and delay. Detailed synthesis instructions will follow.

Project Report

You must turn in a neatly typed report that describes the project in brief. You must describe your design clearly, starting with the hardware description. Clearly describe how the loops are unrolled and how the sine and cosine operators are approximated. Use a lookup table for storing the values. There is no synthesis requirement on the table. The report will be graded as follows:

1. Introduction [5 Points]
 - Abstract
 - Table of Contents
 - Basic summary of your design and results.
2. Hardware Description [15 Points]
 - Basic Hardware Description
 - Functional Block Diagram
 - Discussion about arithmetic units used in computation and precisions that you use
3. System Design [45 Points]
 - Well Annotated Verilog Code
 - Characterization of the hardware - Area and Delay
 - Detailed Simulation. Should be clearly annotated.
4. Testbench and Synthesis[20 Points]
 - Test Strategy

- Testbench for the sample images mentioned. Your simulation must show how the table is changed for different calculations of r and θ .
- Analysis of the synthesized design - Library cells used, their characterizations. You must synthesize the design both using FPGA and Magma flows. Detailed synthesis instructions will follow.

5. Report Format and Style [5 Points]

6. Discussion [5 Points]

Talk to another group whose project was different from yours. Find out details about their project and explain briefly how they designed their project. Also, give details of their test strategy. Write a one-page report about the other group's work.

Clearly describe how you divided your tasks among your group members. Each of you must write a brief evaluation of the other members work. Be honest. Hand them back to us individually. Do not attach it with your project report. The review that you give about your partners will be held confidential.

7. Progress Report [5 Points] Turn in a 2-page progress report that describes your preliminary studies and your hardware design strategy. The progress report is due on April 13 2007 in class.