Real-time Classified Hough Transform Line Detection Based on FPGA

Xinming Wang[1, a], Haishu Tan[2, b], Fuqiang Zhou[1, c]and Yao Zhao[1, d]

[1]School of Instrumentation Science and Opto-electronics Engineering, Beihang University,Beijing 100191,China

[2]Department of Electronic Information Engineering, Foshan University, Foshan 528000, China

[a]xmw19890815@163.com,[b]tanhaishu@foxmail.com, [c]zfq@buaa.edu.cn, [d]joeAsir0560@outlook.com

Keyword: Line Detection, Gradient, Hough Transform, FPGA.

Abstract. Hough Transform (HT) is a popular tool for line detection due to its robustness to noise and missing data. This paper proposes a classified Hough Transform (CHT)to achieve real-time line detection on one kind of hardware circuit called FPGA. It makes full use of the gradient information calculated in the edge detection stage compared with other Hough Transforms.Theresults show that the proposed hardware algorithm is feasible. Asthe classified Hough Transform is divided into two independent parts according to gradient, every edge pixel is substituted into a single calculation formula, it accelerates computation with the same accuracy.

1. Introduction

Automatic detection of lines in images is a classic problem in computer vision. It is also relevant to computer graphics in applications such as image-based modeling and user-interfaces[1].In computer vision, line detection is a fundamental primitive in a wide range of applications including camera calibration, autonomous robot navigation, industrial inspection, object recognition, line based image stitching and remote sensing.

The Hough Transform is a standard method to find line features in an image [2]. It is an attractive technique because of its robustness to noise and changes in the illumination level. The main problems of Hough Transform are the large memory capacity required and the time-consuming computations. In some applications such as robot navigation, object tracking and auxiliary driving, real-time performance is a necessary condition. Software can’t meet the requirement, so the implementation of Hough Transform on hardware circuit is a good choice to achieve real-time line detection.

A lot of work has been carried out about the implementation of Hough Transform on hardware. Cucchiara et al [3]implemented the basic HT on the edge points and the Gradient-Weighted Hough transform (GWHT) for gray-level images on a pipelined architecture using Field Programmable Gate Arrays (FPGA). However,these efforts have a common disadvantage that the trigonometric calculation requires large resource and is time- consuming.

So in this paper, we achieve real-time line detection through a classified Hough Transformon one kind of hardware circuit called FPGA. By using the improved slope-intercept equation, the trigonometric calculation is avoided and the hardware architecture is simplified. Section 2 presents the traditionalHough Transform principle. Section 3 describes our classified Hough Transform based on the gradient and the proposed incremental calculation. Section 4 shows the specificmodules of the line detection system and the proposed hardware architecture. Section 5 gives the experimental results and the evaluation.

2. Traditional Hough Transform Principle

The traditional Hough Transform uses the concept of point-line duality to locate lines in an image. A point P in an image can be defined using a pair of coordinate (x, y) or a set of lines passing through it. The original method used in the HT [4] is to represent every possible line using the slope-intercept form like Eq. (1):

$$c = y - m \bullet x \tag{1}$$

Another parameterization of a line is like Eq. (2) where θ is the angle of the line and ρ represents the closest distance between the line and the image origin.

$$\rho = x \bullet \cos\theta + y \bullet \sin\theta \tag{2}$$

Themainproblem of the traditional Hough Transform is its computational expense. This comes from two sources. The first is the calculation of the sine and cosine. The second is the memory access bandwidth. To solve thisproblem caused by the unified voting scheme, we proposeone classified Hough Transform based on the pixel gradient information which reduces the calculationwith the same accuracy.

3. Classified Hough Transform Based on Gradient Information

Compared with other Hough Transforms which vote for parameter θ from 0 to π, our classified Hough Transform simplify the voting scheme using pixel gradient information. Not only does it reduce the amount of calculation, but it also reduces the clutter in parameter space, making peak detection more reliable. Fig. 1 shows the relationship between gradient and orientation of one line. We divide the lines in an image into two types depending on the gradient information expressed by Eq.(4).

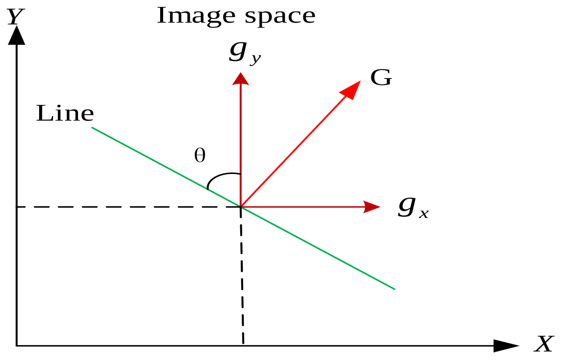


Fig. 1 Line gradient-orientation relationship.

$$\tan\theta = \frac{g_y}{g_x} \tag{3}$$

$$\begin{cases} c = y - m \bullet x, |g_x| \le |g_y| \\ l = x - k \bullet y, |g_x| > |g_y| \end{cases} \tag{4}$$

The incremental form of Eq. (4) is as Eq. (5) and Eq. (6).

$$\begin{cases} m_n = m_{n-1} + \varepsilon \\ c_n = c_{n-1} - \varepsilon \bullet x \end{cases} with \begin{cases} m_0 = -1 \\ c_0 = x + y \end{cases} \tag{5}$$

$$\begin{cases} k_n = k_{n-1} + \varepsilon \\ l_n = l_{n-1} - \varepsilon \bullet y \end{cases} with \begin{cases} k_0 = -1 \\ l_0 = x + y \end{cases} \tag{6}$$

In Eq. (5) and Eq. (6), make $\varepsilon = 1/2^N$, where N is a natural number, and they are replaced by shift operations [5].

4. The FPGA Implementation of Classified Hough Transform

The complete procedure of line detection is shown in Fig. 2. First, the image is processed to detect edges. Each edge pixel then votes for the classified Hough Transform parameter space according to gradient information. Then, detected peaksin parameter space determines candidate lines.

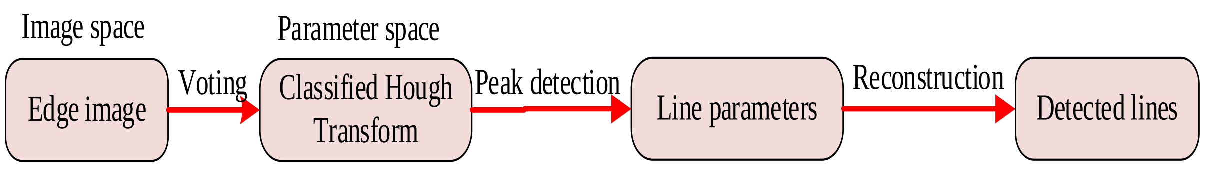


Fig. 2 Line detection procedure

4.1 Edge Extraction.

Considering the accuracy and complexity, we choose Sobel filter like Fig. 3 to achieve edge extraction and obtain the gradient [6] information. The hardware architecture of the Sobel filter is shown in Fig. 3.

The image pixels are shifted through the line buffers[7] that create a delay line. These delay lines feed the filter array simultaneously. With a certain threshold, we determine whether the center pixel of the filter window is an edge point. If it is an edge point, we store the coordinates of the pixel and the comparing result of thevalues g_x and g_y .

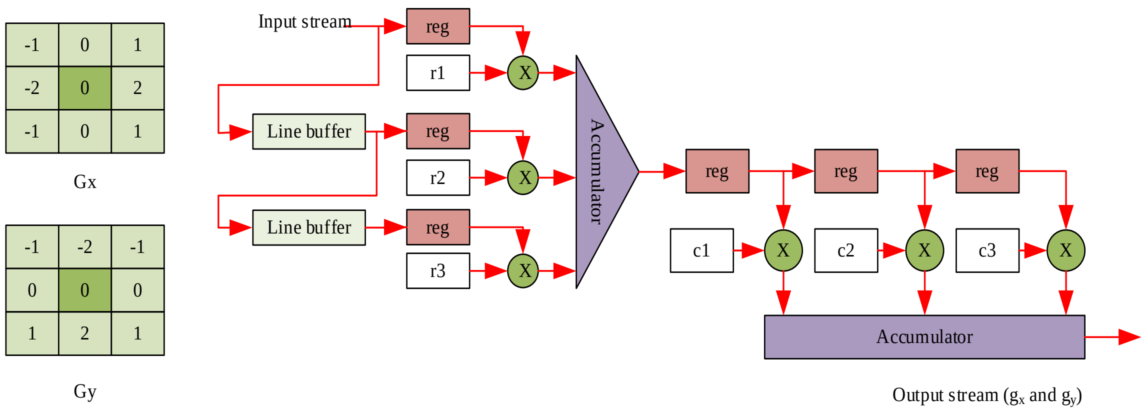


Fig. 3 Sobel filter and its hardware architecture

4.2 Line Parameter Estimation Based on Our Classified Hough Transform.

Since the processing speed of upstream and downstream is inconsistent, we need a buffer to store the output of the upstream pipeline that includes the coordinates of each edge point and the comparing result ofg_x and$g_y$which is coded by a gradient flag. We choose FIFO memory to achieve it. Fig.4 shows the specific architecture. Our classified Hough Transform unit operate independent of the system clock, so it can be processed at a higher speed by using a higher frequency clock.

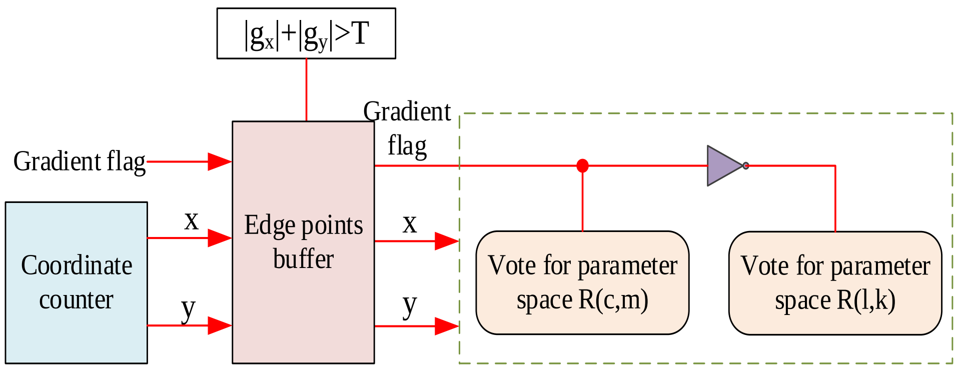


Fig. 4 Edge points buffer unit

As mentioned above, the parameter space is divided into two parts R(c,m) and R(l,k), so two independent hardware circuits are used for voting. Fig.5 shows the specific circuit configuration of our classified Hough Transform. It includes three modules. One is the “line equation calculator” module that calculates the slope (m or k) and intercept (c or l) using Eq. (6) and Eq. (7). It has three adders, one multiplierand two multiplexers which are used to distinguish the first value of the parameter sequence from the others. When ε is defined as the form of $1/2^N$, the multiplier can be substituted by a shift operator. The second is the "address block" module that transforms the two parameters of a line to a memory address. The two parameters and the address have a one to one relationship, so we can gain the liner parameters according to the address. The third is the"voting memory" module. The voting memory has two operationmodes. One is increasing mode and the other is clearing mode. This voting memory accumulates the address value in increasing mode. First, it reads the value of a memory cell that is indicated by address, and the memory cell's value is increased and updated. In clearing mode, the content of the voting memory are transmitted to the peak detector, and then the data is cleared. The address of the voting memory is sequentially increased in clearing mode.

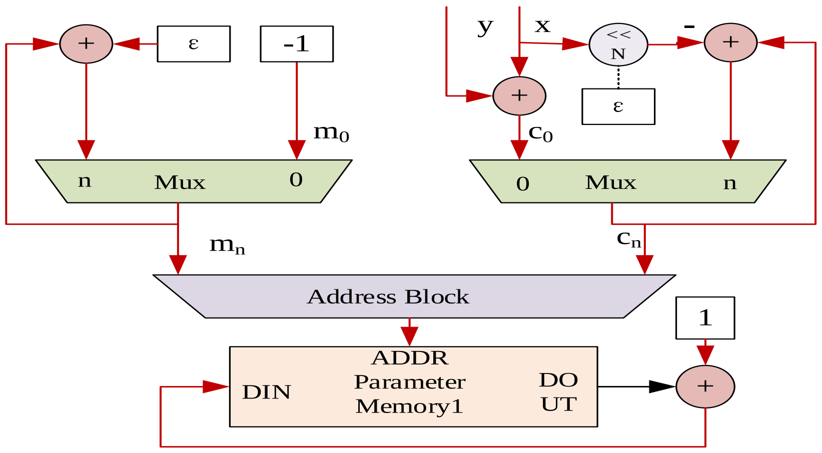


Fig. 5 Classified Hough Transform voting unit

4.3 Peak Detection.

The schematic diagram of peak detection is shown in Fig. 6.A scan in parameter space using a 3 ×3window is suitable for finding out peaks. The scanning operations about parameter (c, m) and parameter (l, k)are parallel. Once a peak is found, we compare it with a preset threshold to removenoise interference and reconstruct the line using parameters that correspond to the peak point. For each edge pixel, when they have the samem_i, the value of the$c_i$is treated as the address$A_j$and the value in$A_j$represents the number of the points with the same line parameters $m_i$and c_i.If the value in the position $(m_i, A_j)$is one peak, the corresponding line is$A_j = y - m_i \cdot x$.For an image sizedH × W, the max value of$c_i$is$c_0$which equals H+W，so for images sized256×256, the max value of the$A_j$is 512.The same rule applies to line parameters $k_i$andl_i.

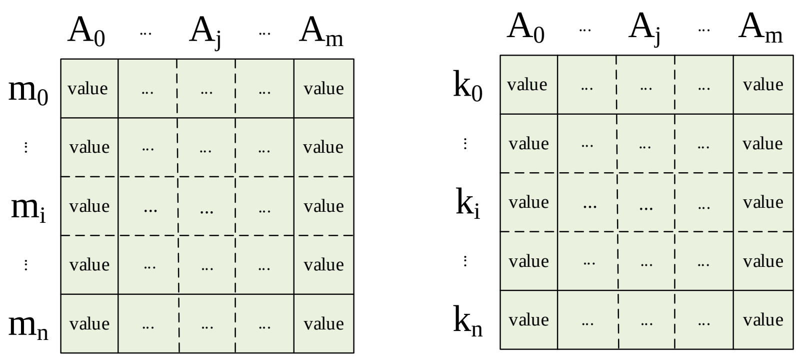
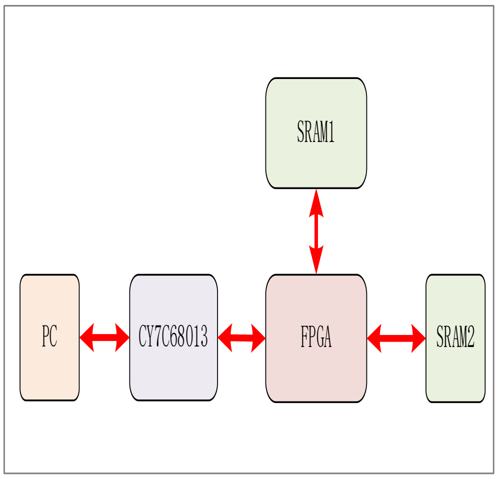


Fig.6 Peak detection schematic diagram

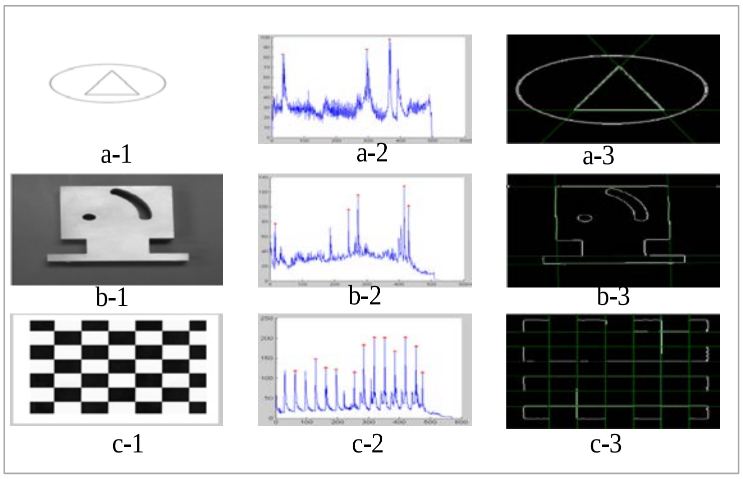
5. Experiments and Evaluation

5.1 ExperimentalResults of FPGA Implementation.

The whole architecture has been implemented on the Xilinx 6slx45fgg484-3. Fig. 7 shows the system components. The image data is transmitted to FPGA from PC by a USB interface chip named CY7C68013,and two off-chip SRAMs are used here. We use gray images sized 256×256 to test the whole algorithm. The line parametersare transmitted to PC and we reconstruct the lines on PC. The experimental results are shown in Fig.7. The left column are the testingimages, the middle column are the peak points and the right column are the reconstructed lines which correspond to line parameters from FPGA.Fig.7shows that our classified Hough Transform proposed in the paper has a good result on line detecting, especially for long lines. For shot lines,ifthe threshold in the peak detection stage is set too high, they are ignored such as the undetected lines in b-3 and c-3.



System components



Experimental results

Fig. 7 System components and experimental results

5.2 Accuracy and Robustness of Classified Hough Transform.

To test the accuracy and robustnessof our classified Hough Transform, six hand-generated straight lines are given with m being -1, 0, 1 and k being 0, -0.58 and 0.58 which are shown in Fig. 8. They aretestedon FPGA 10 times, and the deviation of the detected m and k are shown in Fig.8.We can conclude that ourClassified Hough Transformare robust from the curves in Fig. 8. Our Classified Hough Transform has a good stability and it satisfies the accuracy requirement. The deviation is related to the measurement step ε .If ε is little enough, the deviation can be close to zero. When m isthe initial valuem_0, the deviation is 0, because m_0 is not affected by ε .

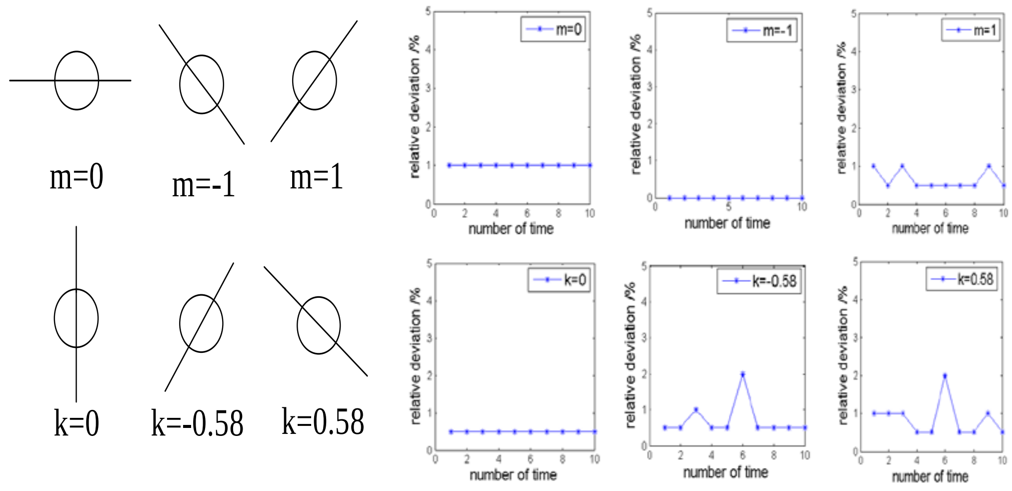


Fig. 8Hand-generated lines and the deviation

5.3 The Comparison between Traditional Hough Transforms and Classified Hough Transform.

To compare our Classified Hough Transform with traditional Hough Transforms, several meaningful parameters relative to the effect are defined as follows: maximum calculation error of line parameter that refers topin traditional Hough Transforms and c_i, $l_i$in Classified Hough

Transform, computational efficiency that represents the computed number of line parameters per cycle, overall efficiency that represents the number of processed points per second. Then the performance are shown in Table1. From Table 1, we can see that ourClassified Hough Transform has a high overall efficiency for the reason that each edge pixel only needs to be substituted into a single calculation formula and two edge points are processed within one clock circle.

Table1 Comparison between algorithms

	Maximum Calculation Error	Computational Efficiency	Overall Efficiency (M/s)	Image Resolution
HT using CORDIC	0.177	2	384	256*256
HT Based on Parallelism	0.125	1	387	256*256
Algorithm Proposed	0.095	2(N+1)	600	256*256

5.4 Calculation Time.

Table 2 shows that when the images used for different algorithms are unified into the common size, our classified Hough Transform has the shortest calculation time (0.71<0.90<1.30).

Table 2 Calculation time

Algorithm	Calculation Time	Image Resolution
algorithmin paper[5]	3.61ms	512*512
algorithm in paper[3]	15.57ms	1024*768
CHT proposed	0.71ms	256*256

5.5 Resource Consumption.

Table 3 shows the resource consumption of our classified Hough Transform algorithm on FPGA architecture.

Table 3 Design summary

	Used	Available	Utilization
Slice Registers	2041	54576	4.00%
Slice LUTs	1976	27288	7.00%
Bonded IOBs	30	316	9.00%
Block RAMs	7	116	6.00%
BUFG/BUFGCTRLs	3	16	19.00%

6. Conclusion

The classified Hough Transform proposed in this papercan achieve a real-time line detection on onehardware circuit called FPGA. It has a good advantage over traditional Hough Transforms.Thewhole line detection system and its specific hardware structure makes the proposed algorithm be implemented on medium-sized FPGA, which is of vital importance for real time detection.

Acknowledgements

The work was supported by the National Natural Science Foundation of China (No. 61471123).

Reference

[1] Fernandes L A F, Oliveira M M. Real-time line detection through an improved Hough transform voting scheme[J]. Pattern Recognition, 2008, 41(1):299-314.

[2] Chen Z H, Su A W Y, Sun M T. Resource-Efficient FPGA Architecture and Implementation of Hough Transform[J]. IEEE Transactions on Very Large Scale Integration Systems, 2012,

20(8):1419-1428.

[3] Cucchiara R, Neri G, Piccardi M. A real-time hardware implementation of the Hough transform[J]. Journal of Systems Architecture, 1998, 45(1):31–45.

[4] Bailey D. Design for embedded image processing on FPGAs[M]// Wiley, 2011.

[5] Dyakonov K M. Two problems on coinvariant subspaces of the shift operator[J]. Integral Equations & Operator Theory, 2014, 78(2):151-154.

[6] Baliga J, Grant A J, Kind A P, et al. Calculating peak-to-average power ratio reduction symbols for multi-carrier modulated signals using a gradient-descent approach: US, US8175179[P]. 2012.

[7] Lim H. Number of tunable wavelength converters and internal wavelengths needed for cost-effective design of asynchronous optical packet switching system with shared or output fibre delay line buffer[J]. Iet Communications, 2013, 7(13):1419-1429.