On a fast discrete straight line segment detection

ALI ABDALLAH, ROBERTO CARDARELLI, GIULIO AIELLI

University of Rome Tor Vergata

Abstract

Detecting lines is one of the fundamental problems in image processing. In addition for real time applications, detection should be achieved in real time. In this paper we investigate the use of fast trigger processor technology used in high energy physics experiments. We propose a method for detecting discrete straight lines segments in binary images based on a simple resistor network trigger processor.

I. INTRODUCTION

Detecting lines is one of the fundamental problems in image processing. For real time applications there is an additional computational time constraint. A fast straight line segments detection is essential in these applications. For example in augmented reality (AR) systems, augmentation of artificial information is conventionally in real time in order to provide real time vision for Human-Computer interaction.

The weighting resistor matrix (WRM) is developed as a fast trigger processor in high energy physics [1], where detecting signal (pattern) in data at high speed is crucial. A first parallel test processor have been developed and tested. It consists on 7 interconnected boards, where each board is able to detect segment lines of a certain slope. The main two advantages of the WRM chip are:

- It is extremely fast, basically no computations are performed, but rather the interconnections of the resistor network give the desired response. This depends only on the signal propagation delay inside the circuit.
- The WRM naturally, using the voltage propagation inside the circuit, performs a best fit correlation instead of relying on perfect instances, allowing to overcome imperfectness in input data.

In this paper we will describe a miniaturized serial version of the WRM, as well as its usage as a fast discrete straight line detector in binary images. The rest of the paper is organized as follows: Section II contains basic definitions as well as an introduction to the WRM circuit. Section III explains the working principles of the WRM, while Section IV illustrates the usage of the WRM data for discrete straight line segment detection.

II. BASICS

Since in binary images we are dealing with discrete planes and there is no such thing as of straight lines in continuous plane, it is fundamental to define what is straight line in the discrete case. A binary image can be viewed as a finite grid inside $\mathbb{Z}^2$. Given a point (n,m) on the $\mathbb{Z}^2$, we define its 4-neighbors and 8-neighbors pixels as follows:

Figure 1: Neighborhood 4 and 8 connectedness.

Given two points $A, B \in \mathbb{Z}^2$. We say A and

B are 4-connected, respectively 8-connected, if B is a 4-neighbor, respectively 8-neighbor, of A.

Definition II.1 A subset X of $\mathbb{Z}^2$ is 4-connected, respectively 8-connected, if and only if for every a in X , its 4-neighbor, respectively 8-neighbor, is not void.

Definition II.2 A discrete straight line if 4-tuple $DSL = (a, b, \mu, \omega) \in \mathbb{Z}^4$, satisfying:

$$\mu \leq ax - by < \mu + \omega,$$

where $\gcd(a, b) = 1$.

In the definition, μ is *intercept* of the line and ω its *thickness*. Given $L = (a, b, \mu, \omega) \in \mathbb{Z}^4$ we have

- if $w < \max(|a|, |b|)$, L is disconnected,
- if $w = |a| + |b|$, L is 4-connected,
- if $w = \max(|a|, |b|)$, L is 8-connected,

Figure 2: DSL examples with different w . (a) $w < \max(|a|, |b|)$; (b) $w = \max(|a|, |b|)$; (c) $w = |a| + |b|$;

We consider in this paper $\omega = \max(|a|, |b|)$ to work on 8-connected, 1 bit-thick discrete lines.

Figure 3: Example of prefixed WRM patterns

Figure 3 illustrates a subset of the WRM prefixed patterns. The complete set of patterns of

one WRM chip contains also the corresponding vertical symmetries. In fact these predefined patterns are hardware readouts of the WRM circuit, for this we will call them RODs. Each device covers -90° to -45° and 45° to 90° . In order to process the full image covering all angles, a simple circuit can be used to rotate the input data by 90° . Then another WRM chip is used in parallel on the rotated data.

As we said in the introduction, the WRM does not look for DSL pattern that matches perfectly the predefined patterns. Data are fitted with the WRM RODs, and the ROD that maximize the likelihood will be taken in consideration. That is to say, even if a certain DSL does not match any of the predefined pattern, it will be still detected with the best fit. A key point for doing that is the natural voltage diffusion inside the WRM chip. Each point of the input binary image can be viewed as 1Volt input and it causes one dimensional voltage diffusion inside the resistor circuit, as shown in figure 4.

Figure 4: Voltage diffusion of 1 "on" pixel

Mathematically, The voltage diffusion inside the circuit is equivalent to a convolution of the binary image Img with a kernel K

$$S = Img * K,$$

Where $K = (\dots (\frac{1}{2})^n \dots \frac{1}{2} 1 \frac{1}{2} \dots (\frac{1}{2})^n \dots)$.

The effect of this convolution, which is performed naturally by the WRM, is to smooth the binary image data reinforcing linear correlation.

III. 8×8 PATCH DSL SEGMENT DETECTION

In this section we explain how the binary image data is processed inside the WRM chip to detect small DSL segments. Since this is the basic functionality of the WRM circuit, the processing and the detection are performed almost instantly.

Given $S = (s_{i,j})_{1 \leq i \leq 8, 1 \leq j \leq m}$ a $8 \times m$ matrix data produced by the convolution of a $8 \times m$ binary image, as explained in the previous section. A series of sums are performed at each $(., j)$ pixel point from $1 \leq j \leq m$ using a N prefixed patterns p^k as follows:

$$f_k^j = \sum_{k=1}^N p^k(S(., j))$$

For example in our case, these sums are performed following 8×8 WRM RODs as illustrated in Figure 3.

Given a threshold th , a DSL pattern fits the data if $|\frac{d^2 f_k^j}{dj^2}| \geq th$. The derivatives are performed following the j columns direction.

Data are loaded inside the WRM serially. At the beginning a 8×16 pixels are loaded inside the circuit, then at each hardware clock this 8×16 window is shifted one pixel to the right inside a $8 \times n$ pixel data. The above sums are computed, in a very fast way, by means of resistors. This hardware clock can be seen as the columns index j of the image data. Once the $8 \times n$ data are processed, the derivatives are performed to detect patterns.

Figure 5: Input data example

Let's take now a concrete example to make things clearer. Given the input data presented in figure 5, we plot in figure 6 only the relevant RODs outputs of the WRM after computing the second derivative.

Figure 6: 2st derivative of the WRM RODs

The grids in the image to the left correspond to the prefixed DSL patterns that are tested against the input data. The plots to the right correspond to the 2st derivative performed over the ROD outputs. The peaks in the data correspond to where the ROD patterns fit well the input data, detecting this way the 5 discrete straight lines in the input data.

As you can see from the above, the peaks amplitude vary according to how much the RODs fits the data at a specific hardware clock. More the pixels points are linearly correlated the greater the amplitude will be.

IV. DSL SEGMENT DETECTION IN IMAGES

In this section we explain how to use the WRM output for detecting arbitrary DSL segments in images.

The first required step is to transform RGB images to binary images, the natural way is to use an edge detector [4]. RGB image is usually transformed to grayscale image. Then an edge detector method looks for sharp changes to produce the final binary image.

Figure 7: Zero Crossing Edge Detector

Gradient based methods consist of thresholding the gradient to detect edges. Since the change in the gradient happens on more than one pixel, thresholding gives thick edges, which are post-processed by a thinning algorithm. On the contrary, zero crossing on the 2nd derivative occurs at one pixel level, producing 1 bit thick edges, ideal input for the WRM.

Another important consideration for our choice, is that the zero crossing edge detector can be implemented easily in electronics. This is usually required when the edge detector needs to provide data to the WRM at a very high speed. In less speed demanding cases, an FPGA implementation can be considered.

Figure 8: RGB image processing chain

Up to this point we have a processing chain 8 that starts from an RGB image, then goes through and edge detector, then the edge detected binary image is processed inside the WRM chip to detect small segments. Since the WRM takes an $8 \times n$ input format, the full image is sampled in order to scan it fully. At the end of the chain, the WRM produces different sets of small 8 bits DSL segments. Each set corresponds to DSL segments that are detected withing certain inclinations. A set can be seen as 2D parametric space, where the parameters are the sample number and the hardware clock. These disjoints parametric spaces give an overall descriptors of image DSL segments. For

example, DSL segments that go from $\begin{smallmatrix} | \\ \backslash \end{smallmatrix}$ to $\begin{smallmatrix} / \\ \backslash \end{smallmatrix}$ we can consider RODs that cover three areas:

 Figure 9: 8×8 ROD covered regions

Since inside each parametric space the points follow a specific direction, detecting segments of arbitrary length consist of joining these points from sample to sample. Following is a pseudo-code for detecting DSL from WRM sampled data set.

Data: WRM sampled data set

Result: DSL segments list

```

while not at end of samples do
    read current sample;
    find 8 bit segment;
    foreach 8 bit segment do
        segment start;
        while Next sample do
            find next corresponding 8 bit segment;
            if found then
                Update segment details;
            else
                segment finished;
                add segment to list;
                break;
            end
        end
    end
end
    
```

Algorithm 1: Detect DSL from a WRM sampled data set.

At the end a list of segment is given where each segment is provided by its starting point (x_1, y_1) and its ending point (x_2, y_2) .

V. EXPERIMENTS

In this section we present experimental results of the speed of our method for detecting DSLs. Just to give a baseline for the speed of our implementation, we will compare our result with the standard classical method for detecting DSLs which is the Hough Transform (HT). The Hough Transform is implemented on a *Intel(R) Core(TM) i7-3632QM CPU @ 2.20GHz* Linux machine. Also the last part of our chain, which is the final DSLs detection algorithm, is implemented in C on the same hardware.

Our input data is a 800×600 pixels binary images produced by the zero crossing edge detector from RGB images. We don't count the time needed for the edge detector to produce the binary image as it is a common time for both methods.

The experiment shows that with our method, we are able to process about 60 *frames/s*. Using the Hough Transform method, we were able to reach a maximum of 8 *frames/s*. Also the experiment shows that our method was more successful in filling gaps in DSLs. Showing a very good performance as shown in figure 10.

Figure 10: Original grayscale image is processed by the zero crossing edge detector. The binary edge detected data is then processed by the WRM. Finally the reconstructed segments are detected and overlaid on the grayscale image.

VI. DISCUSSION

The process of creating different sampled parametric spaces from input data is similar to the voting procedure of the Hough Transform [2]. The voting procedure in our case selects only points which have strong local linear correlation on a specific ROD direction.

The computational complexity and the storage requirements are the main problems for the voting procedure of the HT. Many approaches have been taken in order to address these problems In HT based method. It is important to mention that methods based on the Hough Transform for detecting straight line segments, need also to perform post processing. That is to say, no end points or length are given [3] but only straight lines coefficient. In some sense, the final part of the DSL segment detection is a common problem for the Hough based methods and for the WRM method. The main advantages of the WRM method regarding the last step are:

- WRM Sampled data is smaller than the input data, which reduce computational complexity.
- Sampled data are separated in disjoint parametric spaces for each inclination. Thus the problem is highly parallelizable.

In addition, our method is able to overcome gaps due to the voltage diffusion that reinforces linear correlation. A very high speed in processing binary images can be reached by using the WRM, which fill perfectly the requirement for real time processing.

VII. CONCLUSION

In this paper we discussed the adaptation of a fast trigger processor board for digital image analysis. We showed a formal view of what the device is actually doing. We proposed an algorithm that is able to construct from the WRM samples, a list of DSLs present in the image. The main advantage of this device is that it is extremely fast and highly parallelizable. In

a visual tracking system, this device can play a hardware acceleration role. By providing a list of detected features and patterns, it allows tracking algorithm to save valuable computational time and to have more space to achieve real time tracking.

REFERENCES

- [1] On a very fast topological trigger. R. Cardarelli, V. Chiostrì, L. Di Stante, E. Reali, R. Santonico, M. Travaglini, E. Tusi (Rome U., Tor Vergata) 1993.
- [2] R.O. Duda and P.E. Hart. Use of the hough transformation to detect lines and curves in pictures. Technical Report 36, AI Center, SRI International, 333 Ravenswood Ave, Menlo Park, CA 94025, Apr 1971. SRI Project 8259 Comm. ACM, Vol 15, No. 1.
- [3] Mohammed Atiquzzaman and Mohammed W. Akhtar. A robust hough transform technique for complete line segment description. *Real-Time Imaging*, 1(6):419–426, 1995.
- [4] Djemel Ziou and Salvatore Tabbone, Edge Detection Techniques - An Overview, INTERNATIONAL JOURNAL OF PATTERN RECOGNITION AND IMAGE ANALYSIS, 1998, volume 8, 537-559
- [5] R. Klette and A. Rosenfeld. *Digital Geometry: Geometric Methods for Digital Picture Analysis*. The Morgan Kaufmann series in computer graphics and geometric modeling. Elsevier, 2004.