A improved GPGPU-accelerated parallelization for a rotation invariant Thinning algorithm

Weiguang Yang
School of Software Technology, Dalian University of Technology, Dalian, China
lityangweiguang@163.com

Abstract

This is the text of the abstract.

Categories and Subject Descriptors CR-number [*subcategory*]: third-level

General Terms term1, term2

Keywords keyword1, keyword2

1. Introduction

key-word INSERT-SORT TRUE out-degree $A.len$ $A.len$ $A.len$ $A[1\,..\,n]$
$i = j - 1$
$i = j - 1$
$high$ == low
$high == low$

Thinning is the process of reducing the thickness of each line of patterns to just a single pixel. Thinning is an important pre-processing step for many image analysis operation, such as image processing, character recognition, pattern recognition and so on.

A comprehensive survey of thinning algorithms is described by Lam et al. in this paper, the author reviewed about 100 thinning algorithms. There two kinds of the thinning algorithm, the sequential thinning algorithm(STAs) and the parallel thinning algorithm(PTAs). As PTAs scan remove all the contour pixels that should be deleted in one iteration rather than only one pixel like STAs. So PTAs are usually faster than SPAS. There are three classes of PTAs: n-subiteration parallel Thinning algorithms, n-subfield parallel Thinning parallel algorithms and fully parallel Thinning algorithms.

With the increase of image size, such as satellite image and medi6cal image, result in the longer execution time of thinning algorithms. For example, thinning a 2D image(2048*2048) takes about 17s on i7 CPU platform by using the A-W algorithm. It is a challenge to implement thinning algorithms in real-time. K.Kim et al implemented a PTA by using FPGA , but the implement is hard for general engineer and it limits the image size. Hu BingFeng et al implement 12- subiteration parallel 3D thinning algorithm on GPU and the speed-up achieved was 152x.

CONF 'yy, Month d–d, 20yy, City, ST, Country.

http://dx.doi.org/10.1145/nnnnnnn.nnnnnnn

Algorithm 1 Calculate $y = x^n$

Require: $n \geq 0 \vee x \neq 0$
Ensure: $y = x^n$
  $y = 1$
  if $n < 0$ **then**
    $X = 1/x$
    $N = -n$
  else
    $X = x$
    $N = n$
  end if
  while $N \neq 0$ **do**
    if N is even **then**
      $X = X \times X$
      $N = N/2$
    else $\{N$ is odd$\}$
      $y = y \times X$
      $N = N - 1$
    end if
  end while

With the multi-thread parallel processing power, many problems, such as large data sets and intensive computation, can be resolved efficiently on GPGPU.as s modern GPU architecture, Compute Unified Device Architecture (CUDA) is supported by Nvidia GPUs for general-purpose parallel computing. The parallel thinning algorithms can be well implement on GPU computation plat form owing to its data parallelism. we implement several parallel thinning algorithms(PTAs) on GPU in figure 1.Figure 1 results show PTAs achieves a good performance on average 60 speed up improvement. But the speedup of A-W algorithm is not ideal. Because there are too many branch s in the algorithm flow which will lead a low warp efficiency.

In this paper, we manly focus on A-W algorithm, improved the algorithm with a look-up table, which decrease the branch. This paper makes the following contributions. 1. we implement several parallel thinning algorithms(PTAs) on GPU and proved the PATs could obtain a good speedup on GPU. 2. we present a novel and sample parallel strategy on A-W algorithm which could decrease the branch and logic complexity. 3. we discussed in detail about the relationship between the speedup

The rest of the paper is organized as follows. Section 2 briefly describe the A-W algorithm. Then section 3 illustrates the detailed design of A-W algorithm on CUDA. Experimental analysis and results are shown in Section4 and Finally we offer the future work and conclusion in Section5.

A. Appendix Title

This is the text of the appendix, if you need one.

Acknowledgments

Acknowledgments, if needed.

References

[1] P. Q. Smith, and X. Y. Jones. ...reference text...