Parallel Automatic Text-Background Separation in Picture Books

Chang Hu
Dept. of Computer Science, University of Maryland
October 29, 2007

1. Problem

The International Children's Digital Library (ICDL, www.childrenslibrary.org) contains 1925 children's books in 39 different languages, visited monthly by more than 50,000 users worldwide. All the book pages presented on the web site are scanned images from physical books, which suffer from readability problems due to the wide range of screen sizes and browsers. The ICDL Readability project is a major effort to address this problem by extracting textual information from the pages and presenting it on top of the original drawings (without text) in those pages. This extraction-representation process is especially crucial to picture books because the authors' creative intent can be preserved.

The first step of this project is a text-background separation system. This system involves document image processing techniques such as layout analysis, text location and background inpainting[1], which can be time consuming when applied sequentially to large images.

Parallel image processing systems[2][3] has been discussed and built since the early days of the field of computer vision itself, including hardware as well as software design. Such systems are also built in practice[4]. On the other hand, however, there's no image processing system built under the MapReduce framework reported so far.

The proposed project explores the ways in which text-background separation can be implemented in parallel under the MapReduce framework. A MapReduce-based translation of the existing sequential algorithms will first be implemented. The text-background separation system will be hence built and deployed. Other approaches to a parallel system will also be explored.

2. Resources

The data set will be a subset of all ICDL books. A sequential prototype of the Text-background separation system has been implemented and tested using the Matlab Image Processing Toolbox. The Sun Java Advanced Imaging API[5] will be used for interfacing with the Java-based Hadoop system. JMagick[6] is an alternative if more image processing support is needed.

3. The MapReduce Perspective

The existing sequential algorithms are mostly localized, or independent to the global features of the page image. Image processing algorithms like these fit into the MapReduce framework naturally because an image can thus be divided (mapped) into sub-images with which processing can be done in parallel, and processed sub-images can then be stitched (reduced) back into one image.

Another aspect that has a large potential benefit from cloud computing is image inpainting, which is the current bottleneck in terms of processing time. Inpainting reduces to solving a partial differential equation (PDE) iteratively. Being a much-discussed topic in parallel computing, PDE solvers can be treated by MapReduce, too.

A more challenging problem, however, is that if each one of the image operations in the system can have a parallel implementation in the MapReduce framework. It would also be very interesting to see if the MapReduce implementations are more efficient.

4. Interesting Extensions

Based on the implantation of this system, it is also possible to identify a generic model for parallel image processing on a cluster of commonplace computers.

5. REFERENCES

[1] Bertalmio, M., Sapiro, G., Caselles, V., and Ballester, C. 2000. Image inpainting. In *Proceedings of the 27th Annual Conference on Computer Graphics and interactive Technique*. DOI= http://doi.acm.org/10.1145/344779.344972

[2] Rosenfeld, A. Parallel image processing using cellular arrays. *Computer. Vol. 16, no. 1, pp. 14-20. 1983*

[3] Bräunl, T., Parallel Image Processing, , *Springer, 2001*

[4] The Beowulf Cluster Site, http://www.beowulf.org/

[5] Java Advanced Imaing, http://java.sun.com/javase/technologies/desktop/media/jai/

[6] JMagick, http://www.yeo.id.au/jmagick/