High-Quality ASCII ART GENERATION with GPU Acceleration

Koji Nakano
Department of Information Engineering, Hiroshima University
Kagamiyama 1-4-1, Higashi-Hiroshima, 739-8527, JAPAN
HIROSHIMA UNIVERSITY

Abstract

We propose a new technique to generate a high-quality ASCII art image using Local Exhaustive Search(LES). We have implemented our new technique in a GPU to accelerate its computation. The experimental results show that the GPU implementation can achieve a speedup factor up to about 57 over the CPU implementation.

ASCII Art

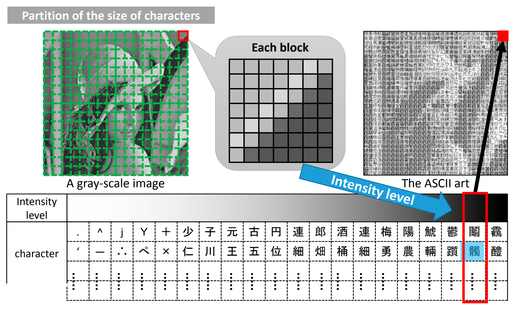
An ASCII art is a matrix of characters reproducing an original image.

Original gray-scale image

ASCII art

A conventional method

The idea of a conventional ASCII art generation is to partition an original image into blocks of the same size as characters. Each block is assigned to a character such that each character reproduces the intensity level of the corresponding block.



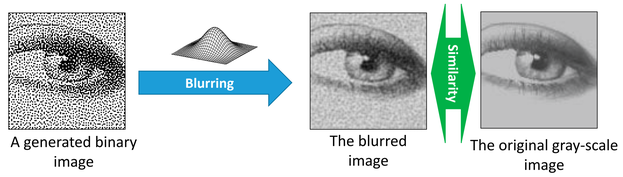
[1] Yasuaki Ito and Koji Nakano, FM Screening by the Local Exhaustive Search with Hardware Acceleration, International Journal of Foundations of Computer Science, Vol. 16, No.1, pp.89–104, February 2005.

Our proposed method

Our technique is inspired by the digital half-toning using the local exhaustive search(LES) to optimize binary images based on the human visual system [1]. Because generated ASCII arts are binary images, LES can be applied to ASCII art generation.

The evaluation method based on the human visual system

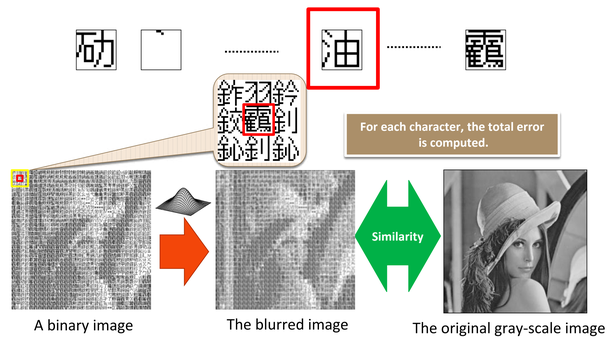
If the blurred image is very similar to the original image, the generated binary image reproduces the original image.



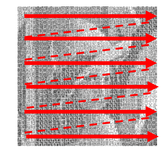
The similarity can be computed with the sum of the difference between the blurred image and the original gray-scale image with respect to intensity level.

Outline of our algorithm

We find a replacement character which minimizes the total error over all characters.



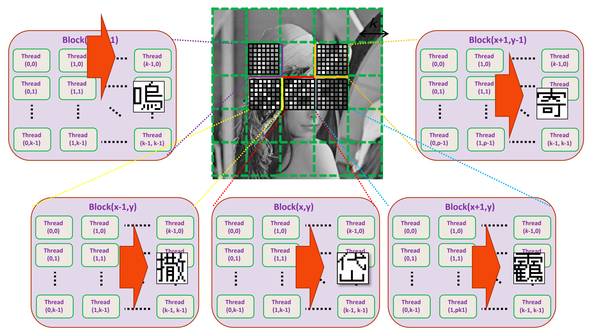
This replacement procedure by the raster scan order is repeated until one round of the raster scan order search does not replace characters.



Raster scan order

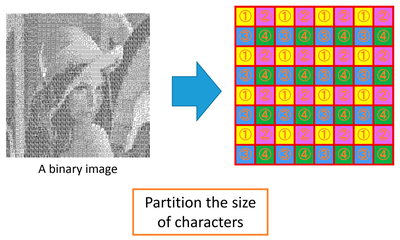
Acceleration using the GPU

Each CUDA block replaces the assigned block by the selected character obtained with LES in parallel.



Parallel Execution

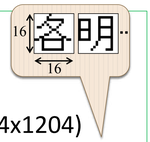
Since LES for adjacent blocks cannot be executed in parallel, we partition blocks into four groups. In each group, the affected regions of all blocks do not overlap each other.



Experimental Result

For the purpose of comparison, we also implemented the sequential algorithm on the CPU.

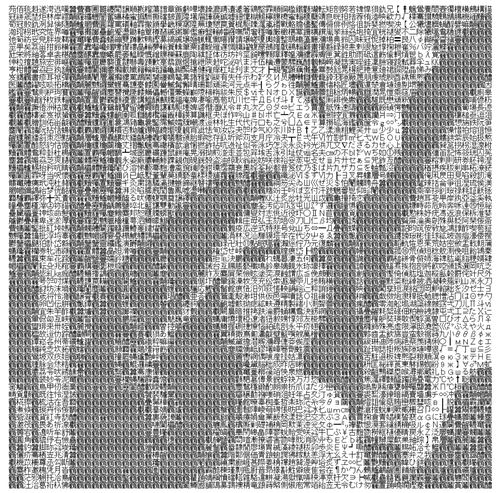
Experimental environment
CPU : Intel Xeon X7460
GPU : NVIDIA GeForce GTX 680
Input image: Lena (256x256, 512x512, 1024x1204)
Character code: JIS Kanji code (7310 characters, 16x16)



Computing time

Image size	256 × 256	512 × 512	1024 × 1024
CPU [s]	4.06	16.1	64.2
GPU [s]	0.125	0.331	1.12
Speed-up	32.6	48.5	57.1

The experimental results show that the GPU implementation can achieve a speedup factor up to about 57 over its CPU implementation.



Conventional method

Our method