A Block-Based Union-Find Algorithm to Label Connected Components on GPUs

Stefano Allegretti, Federico Bolelli, Michele Cancilla and Costantino Grana
DIEF, Università degli Studi di Modena e Reggio Emilia, Italy

Abstract
Connected Components Labeling (CCL) is a fundamental image processing algorithm that extracts objects from an input binary image, giving each of them a different label. In the last decade, the fast development of GPUs supported the design of a few parallel approaches to efficiently solve the problem.

Goal: improve the performance of a Union-Find GPU-based algorithm by applying the 2x2 Grana mask, to reduce the number of threads and memory accesses.

GPU Union Find CCL
The Union Find algorithm consists of three functions executed by multiple threads in parallel (kernels). Each pixel is assigned a thread.
1. Initialization
Initial labels are pixels raster indexes.
2. Merge
Foreground pixels are linked to their neighbors, by means of Union-Find procedures. The thread working on pixel \( x \) checks neighbors \( p, q, r \) and \( s \).
3. Compression
At the end of the kernel, pixels belonging to the same connected component are linked together in a Union-Find tree.

Experimental Results
Connected Components Labeling (CCL) is a fundamental image processing algorithm that extracts objects from an input binary image, giving each of them a different label. In the last decade, the fast development of GPUs supported the design of a few parallel approaches to efficiently solve the problem.

The block-based mask
Foreground pixels in a 2x2 block are always connected.

For this reason, a Connected Components Labeling algorithm can assign labels to blocks in the first step. This approach allows to divide by 4 the number of threads, and to decrease the amount and average height of Union-Find trees.

Blocks connectivity depends on lower level pixels connectivity. The neighborhood mask becomes larger:

Green pixels are responsible for connecting block \( X \) to \( P, Q, R \) and \( S \). They are read by the thread working on block \( X \), in order to know which blocks must be linked together.

GPU Block-based Union Find CCL
Block-based Union Find has the same kernels of Union Find, plus a final one to assign labels to single pixels.

1. Initialization
Block labels are initialized in raster scan order.
2. Merge
Neighbor blocks are linked together with Union-Find, using the larger mask.
3. Compression
Final block labels are set to the roots of trees.
4. FinalLabeling
Block labels are copied into internal foreground pixels.

Block-based Union Find (BUF) has been compared to state-of-the-art using the common benchmark YACCLAB. Test cases are:
- Real world datasets of video surveillance, fingerprints, text, medical images and natural images.
- Synthetic datasets of images with varying density and granularity of foreground pixels.

Test environment is a Quadro K2200 NVIDIA GPU (Maxwell architecture) with CUDA 10.0. Code has been compiled with NVCC V10.0.130 with optimizations enabled.

![Diagram](image.png)