# CUDA COMPETITION

Connected Components Labeling Algorithm

# WHAT IS CCL?

Connected-component labeling (alternatively connected-component analysis, blob extraction, region labeling, blob discovery, or region extraction) is an algorithmic application of graph theory, where subsets of connected components are uniquely labeled based on a given heuristic.  
<http://en.wikipedia.org/wiki/Connected-component_labeling>

# SETUP YOUR PROJECT IN WINDOWS OS

Software requirement: CUDA 4.0 or above (driver, toolkit); Windows Based C++ Compiler e.g. Visual Studio.

Hardware requirement: CUDA enabled gpu (<http://developer.nvidia.com/cuda-gpus>).

If you are using Visual Studio, please make sure you add all the \*.cu and \*.cuh files to your CUDA Project and take use of the right project configuration.

Run: [name of execution file] –infile=”[infile name]” –cpu\_outfile=”[CPU output file name]” –gpu\_outfile=”[GPU output file name]”  
Where [infile name] is the name of input file; [CPU output file name] is the name of the output of CPU CCL algorithm; [GPU output file name] is the name of the output of GPU CCL algorithm. For example, .\CCL –infile=”1.bmp” –cpu\_outfile=”cpu.bmp” –gpu\_outfile=”gpu.bmp”

# SETUP YOUR PROJECT IN LINUX

Software requirement: CUDA 4.0 or above (driver, toolkit); g++ Compiler;

Hardware requirement: CUDA enabled gpu (<http://developer.nvidia.com/cuda-gpus>).

Compile: Makefile has already been added to the folder of Project. You can just type “make” in your command line under the DIR of the folder to compile the program.

Run: .\CCL –infile=”[infile name]” –cpu\_outfile=”[CPU output file name]” –gpu\_outfile=”[GPU output file name]”  
Where [infile name] is the name of input file; [CPU output file name] is the name of the output of CPU CCL algorithm; [GPU output file name] is the name of the output of GPU CCL algorithm. For example, .\CCL –infile=”1.bmp” –cpu\_outfile=”cpu.bmp” –gpu\_outfile=”gpu.bmp”

# DEVELOPING GPU CCL ALGORITHM

A CPU version CCL module has been called in the main function for demo and comperation. What you need to do is designing a GPU version in ccl\_gpu.cu to achieve as much speed as you can (no need to follow the same alg of CPU). The interface of the gpu module has been defined in ccl\_gpu.cuh.

Don’t modify the files except ccl\_gpu.cu and ccl\_gpu.cuh if not necessary. If you need a modification please add a comment after it.

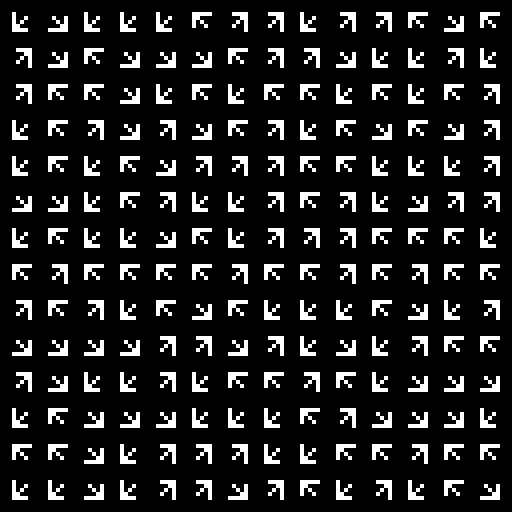
# RESULT VERIFICATION

The output of GPU CCL function “gpuLabelImage” is the image data with the depth of 32 (int type) and “iNumLabels” which is the number of Labels (targets) in the image. In the project, the image data is maped to true color image of 24bit depth for easy verification.

For the given input, GPU output “iNumLabels” should be the same as CPU algorithm. One target in the output file should be marked with only one color. 4-connectivity algorithm is also acceptable. In this case, your iNumLables may be different with the Output of CPU algorithm.

Note that: the output image of GPU CCL not necessarily has the same look as the CPU version, since you just need to give each target a unique label as the definition of CCL.

Example of input file:



Example of output file:

