Assignment Sheet 3

Reduction

Compaction

Todo

- **Download exercise03.tar.gz from course web page.**

```
tar –xzf exercise03.tar.gz
```

Reduction

- **Another variant of the dot product. This time we want to perform the reduction in the shared memory.**

- **Use the provided skeleton and fill in the missing gaps**
  - Read a block of data into shared memory and reduce it in shared memory to a single value
  - You obtain an array of as many values as you had blocks
  - Run the kernel a second time to reduce these values to a single number.

Compaction

- **Sometimes the number of output items per thread needs to be variable.**
    - In this case parallel threads do not know to which location to write, because this depends on the number of items generated by the previous threads.
    - The scan (prefix sum) algorithm solves the problem by generating an index array of the correct locations.

- **Task: Segment an image into a set of Voronoi cells, defined by local maxima in the image**
    - The maximum detection and Voronoi cell generation are already included
    - Only the compaction with the scan needs completion
    - Because of the important parallel computing pattern of the scan, this is a highly recommended exercise.

- **Use the provided skeleton and fill in the missing gaps**

- **Data**
    - In folder images/ you will find example input images
    - In folder referenceImages/ we have pre-computed the solution for different input images
    - Use these pre-computed solutions to check that your code works correctly!