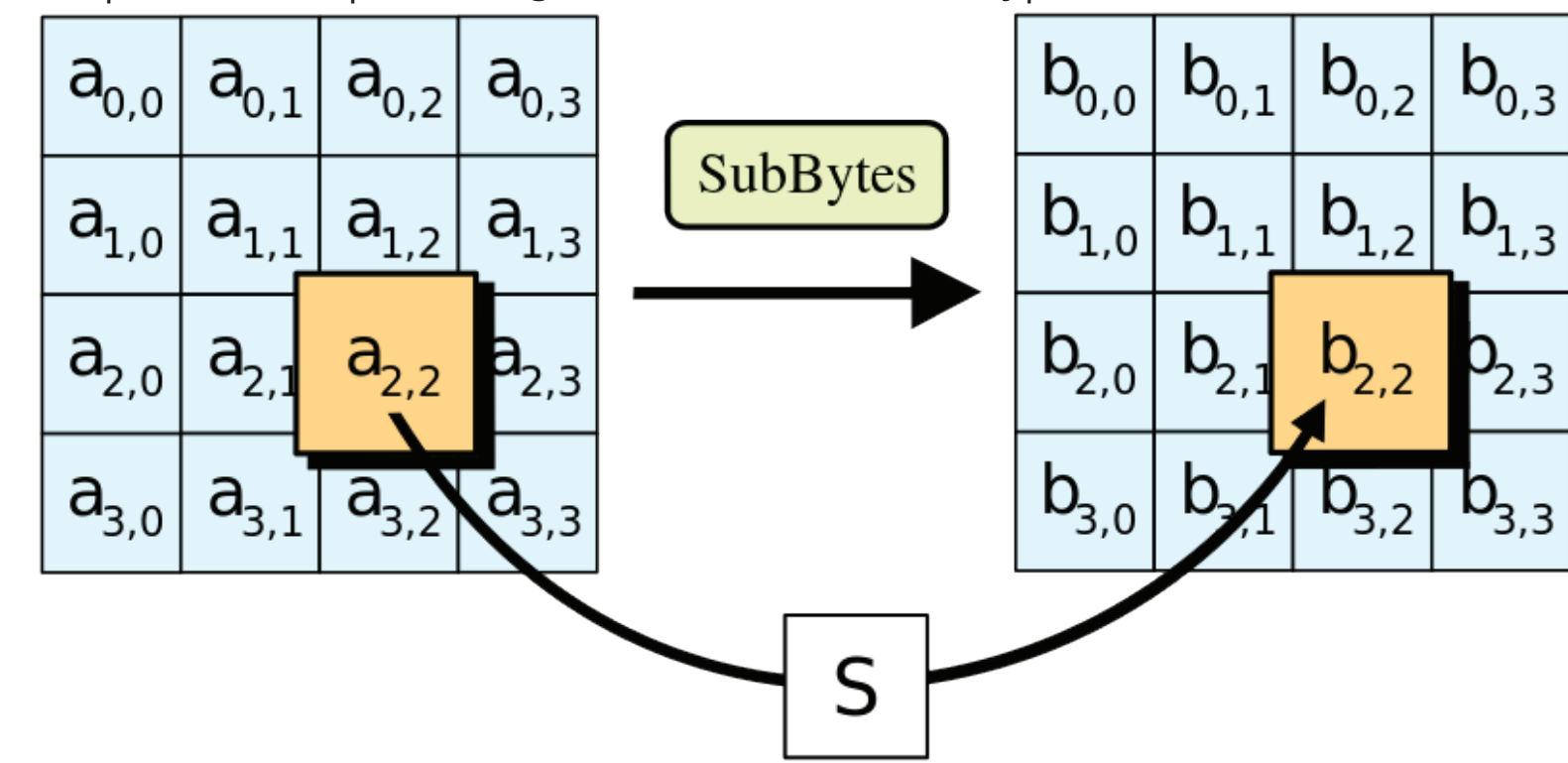
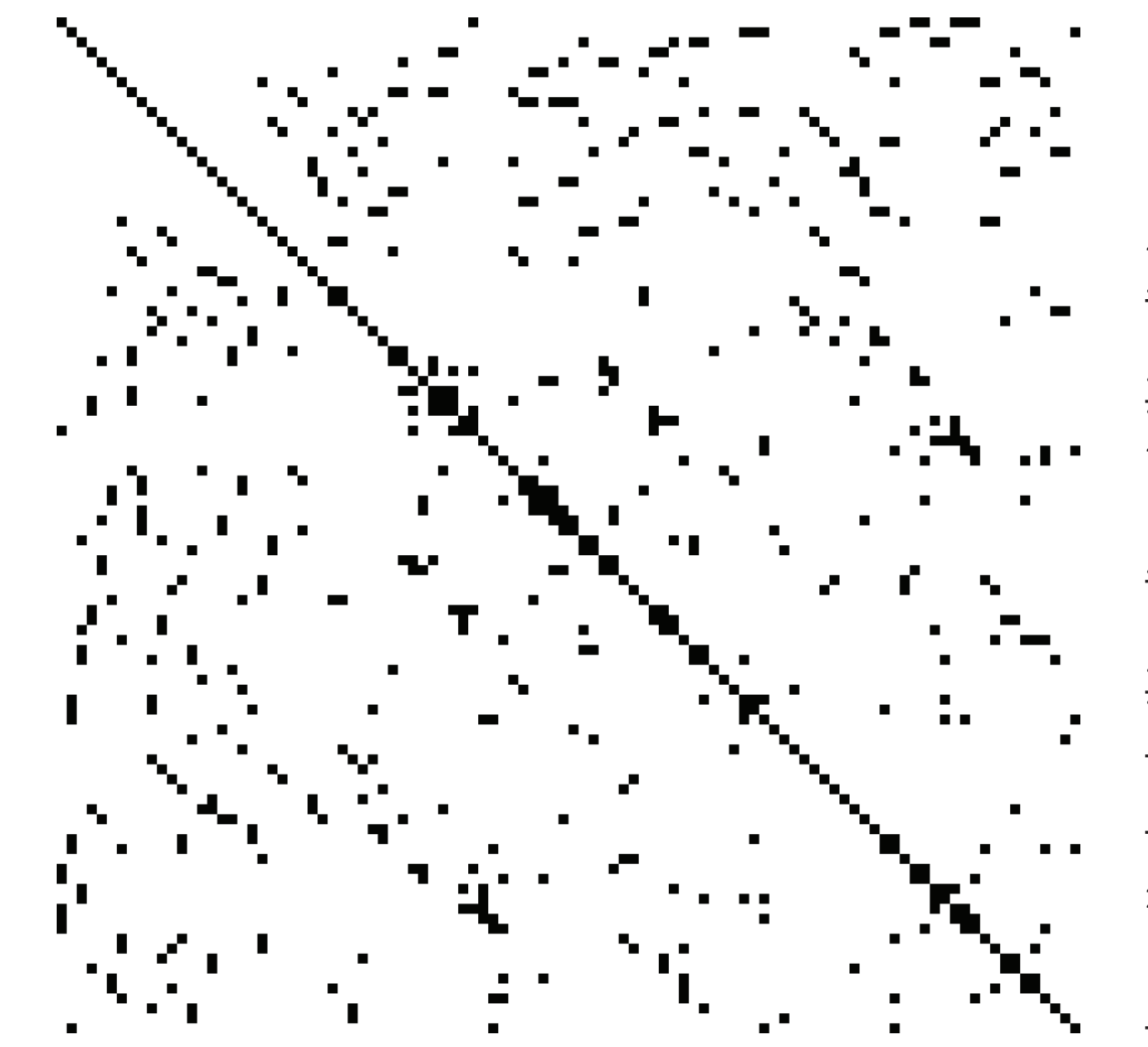


https://en.wikipedia.org/wiki/Advanced_Encryption_Standard#/media/File:AES-SubBytes.svg



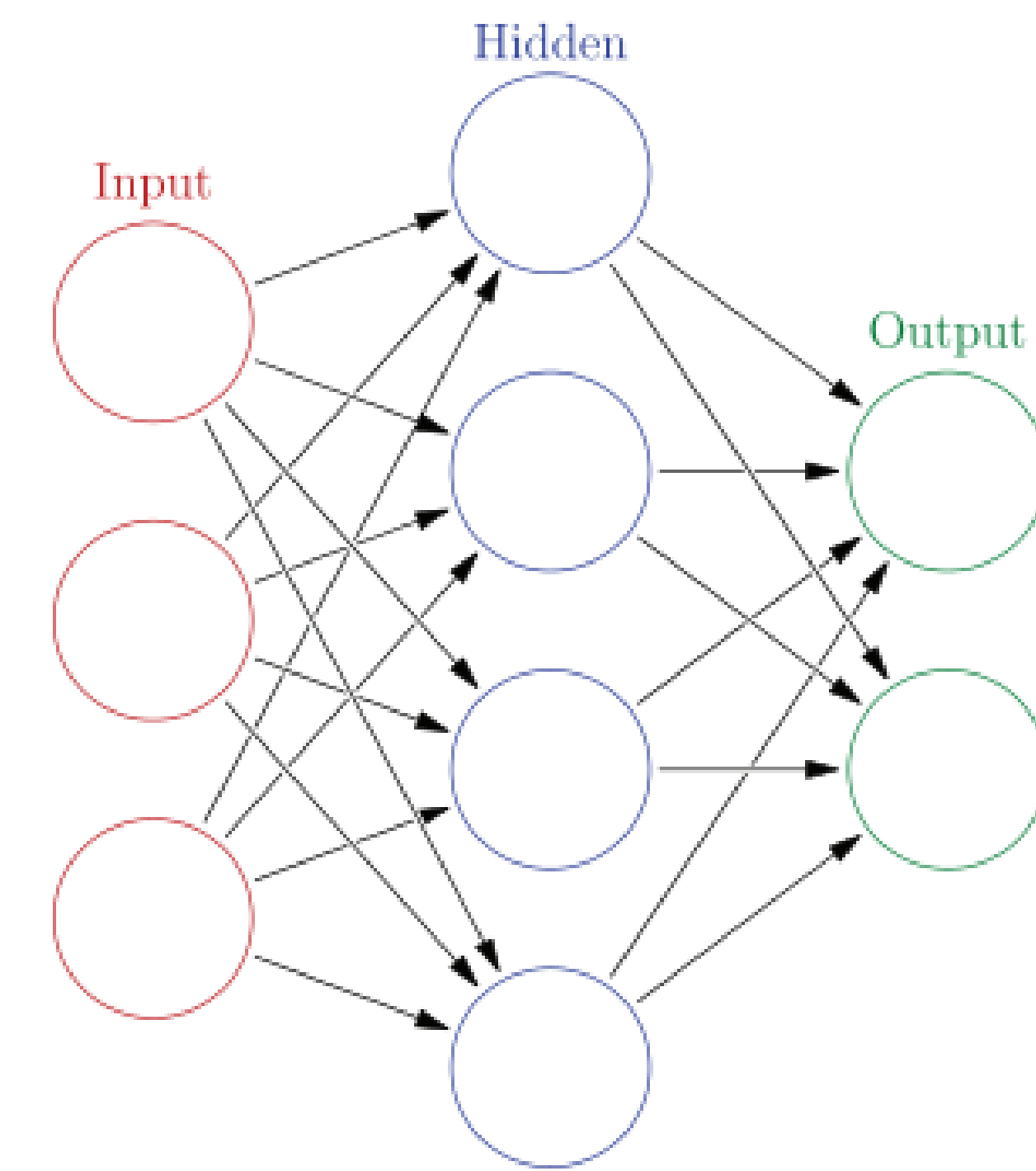
https://en.wikipedia.org/wiki/Total_variation_denoising



https://upload.wikimedia.org/wikipedia/commons/8/8a/Finite_element_sparse_matrix.png

GPU Programming

Cuda Projects



https://en.wikipedia.org/wiki/Artificial_neural_network#/media/File:Colored_neural_network.svg

$$P = e^{-rT} \int_{\mathbb{R}^d} \max \left(0, \exp \left(m + \sum_{j=1}^d a_j \Phi^{-1}(x_j) - K \right) \right) dx$$

Project rules

- All code is yours.

Project rules

- All code is yours.
- Document performance optimization.

Project rules

- All code is yours.
- Document performance optimization.
- Projects are open ended.
 - › You can pivot it into a direction that interests you.

Project rules

- All code is yours.
- Document performance optimization.
- Projects are open ended.
 - › You can pivot it into a direction that interests you.
- We discuss current status each week in the tutorial.

Project rules

- All code is yours.
- Document performance optimization.
- Projects are open ended.
 - › You can pivot it into a direction that interests you.
- We discuss current status each week in the tutorial.
- At the end:
 - › 10 min presentation + 2 page write-up

Time line

- 10/1: CPU impl. + proposal for Cuda implementation
- 16/1: Consultations
- 23/1: Presentations (in lecture)
- 30/1: hand-in write-ups

Projects

Matrix transpose

- Implement efficient matrix transpose
- Your implementation should be efficient irrespective of matrix size and data type

$$A_{ij}^T = A_{ji} \quad , \quad A \in \mathbb{R}^{n \times m}$$

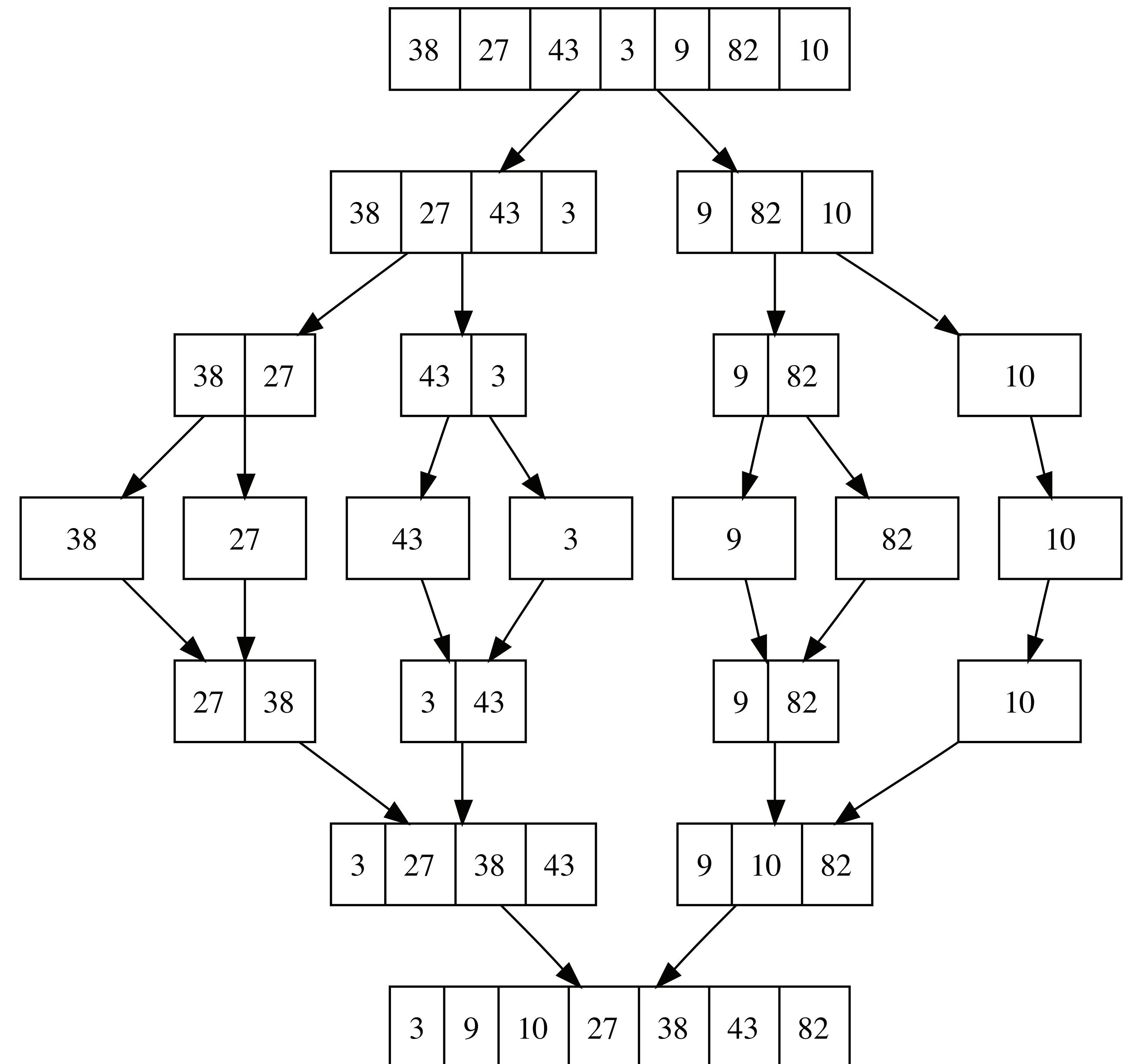
Matrix multiplication

- Implement efficient matrix multiplication
- Your implementation should be efficient irrespective of matrix size and data type

$$C_{ij} = \sum_{k=1}^n A_{ik} B_{kj}$$

Merge sort

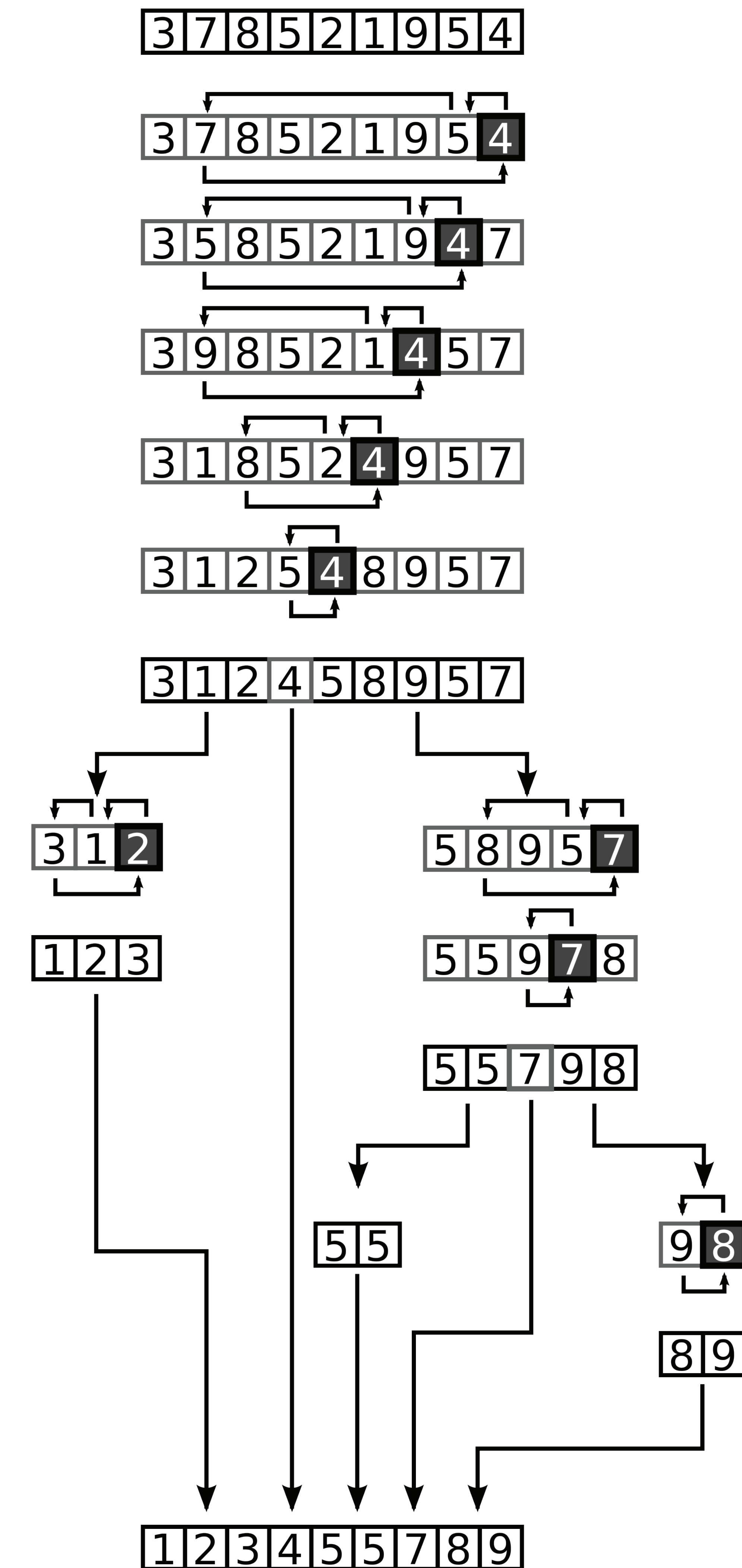
- Develop an efficient merge sort implementation
- Use grid synchronization on the device to improve performance



from https://en.wikipedia.org/wiki/Merge_sort

Quick sort

- Develop an efficient quick sort implementation
- Use dynamic parallelism to improve performance



<https://en.wikipedia.org/wiki/Quicksort>

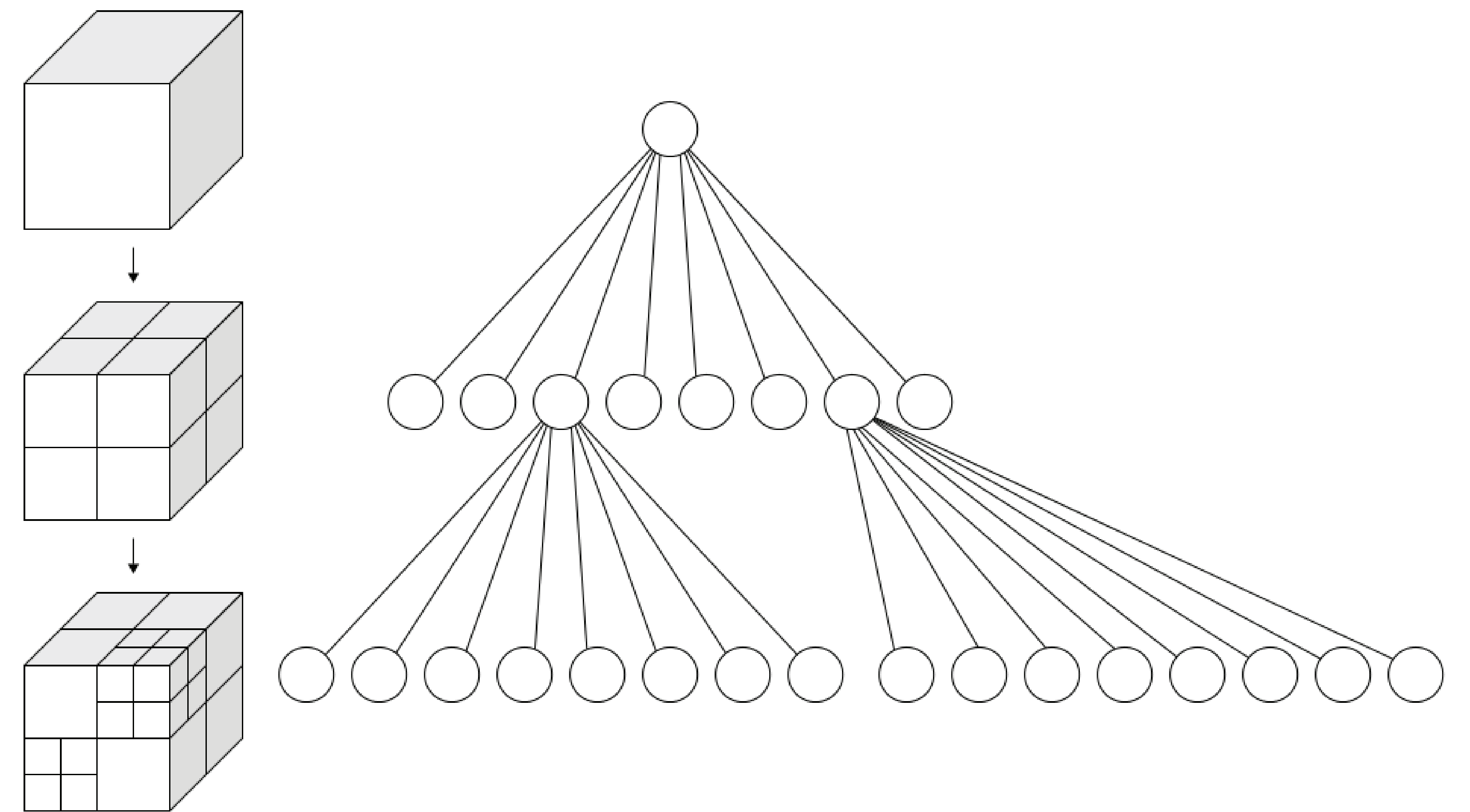
Placement-new

- Implement memory arena placement-new to seamlessly transfer a set of instances to the device

```
class Sphere {  
    void*  
    operator new( size_t size) {  
        ...  
    }  
};
```

Octree

- Implement an octree as a spatial data structure for geometric primitives
- Use dynamic memory allocation during construction



<https://en.wikipedia.org/wiki/Octree>

Denoising using convolution

- Implement 2D convolution-based denoising
- Explore performance gains using a mixed precision implementation

Original



Noisy image



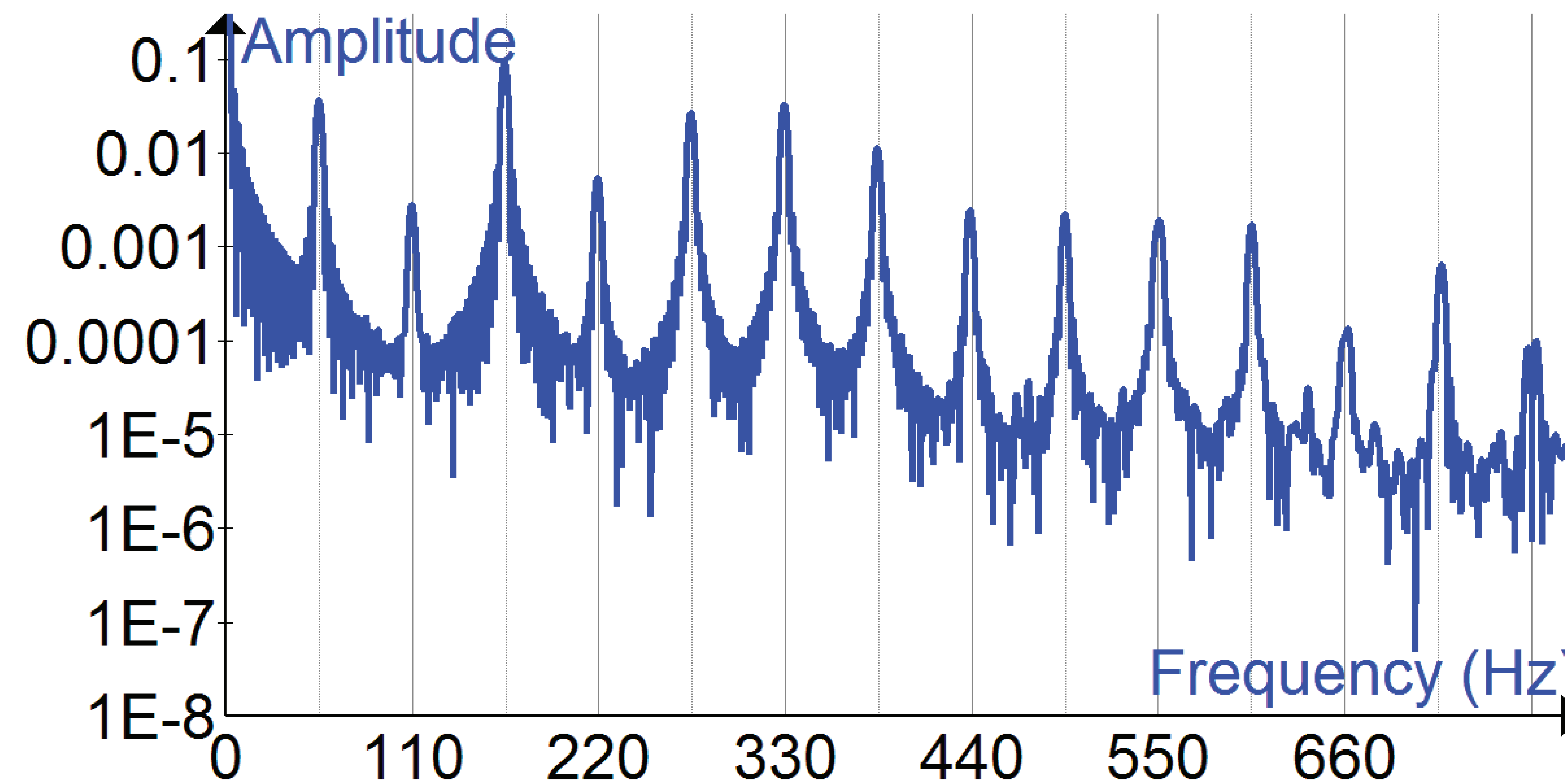
Denoised image



https://en.wikipedia.org/wiki/Total_variation_denoising

Audio Processing using FFT

- Implement FFT in Cuda (and compare to CuFFT)
- Demonstrate simple audio processing



https://en.wikipedia.org/wiki/Fourier_analysis#/media/File:Fourier_Transform_of_bass_guitar_time_signal.png

Tone mapping using histograms

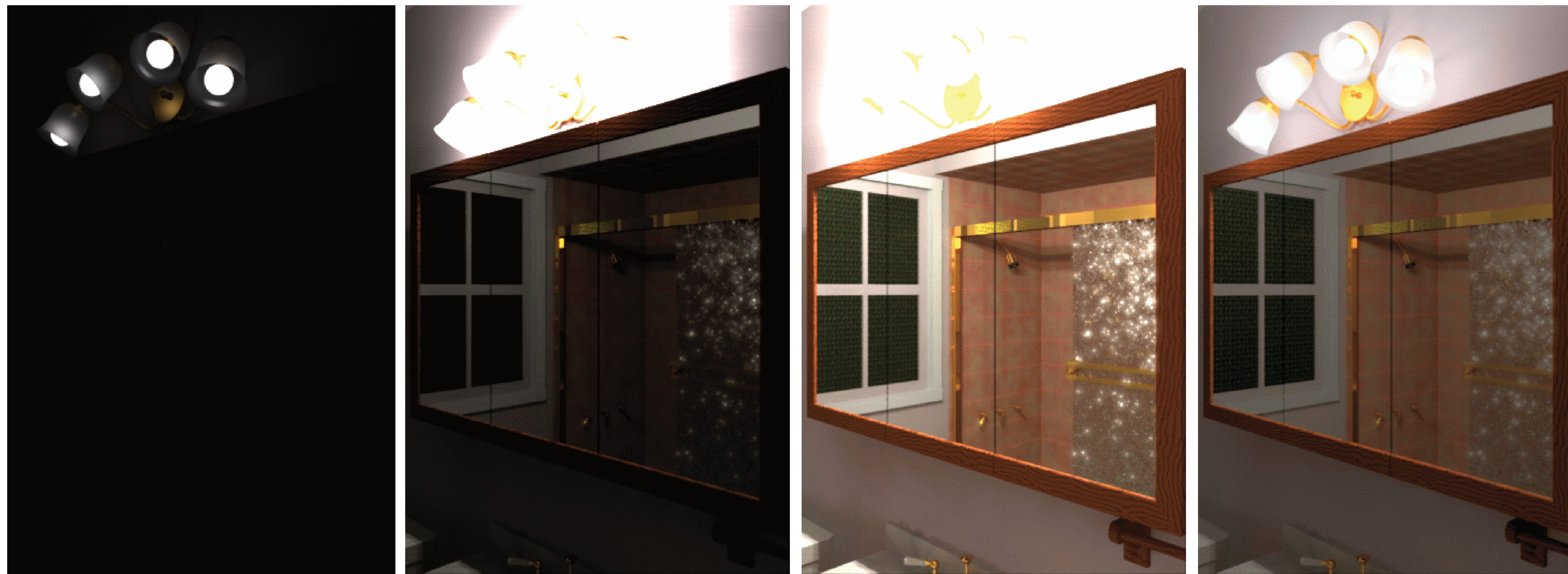
- Implement image histogram computation in Cuda
- Use this for histogram-based tone mapping



<http://resources.mpi-inf.mpg.de/departments/d4/teaching/ws200708/cg/slides/CG13-ToneMapping.pdf>

Tone mapping using histograms

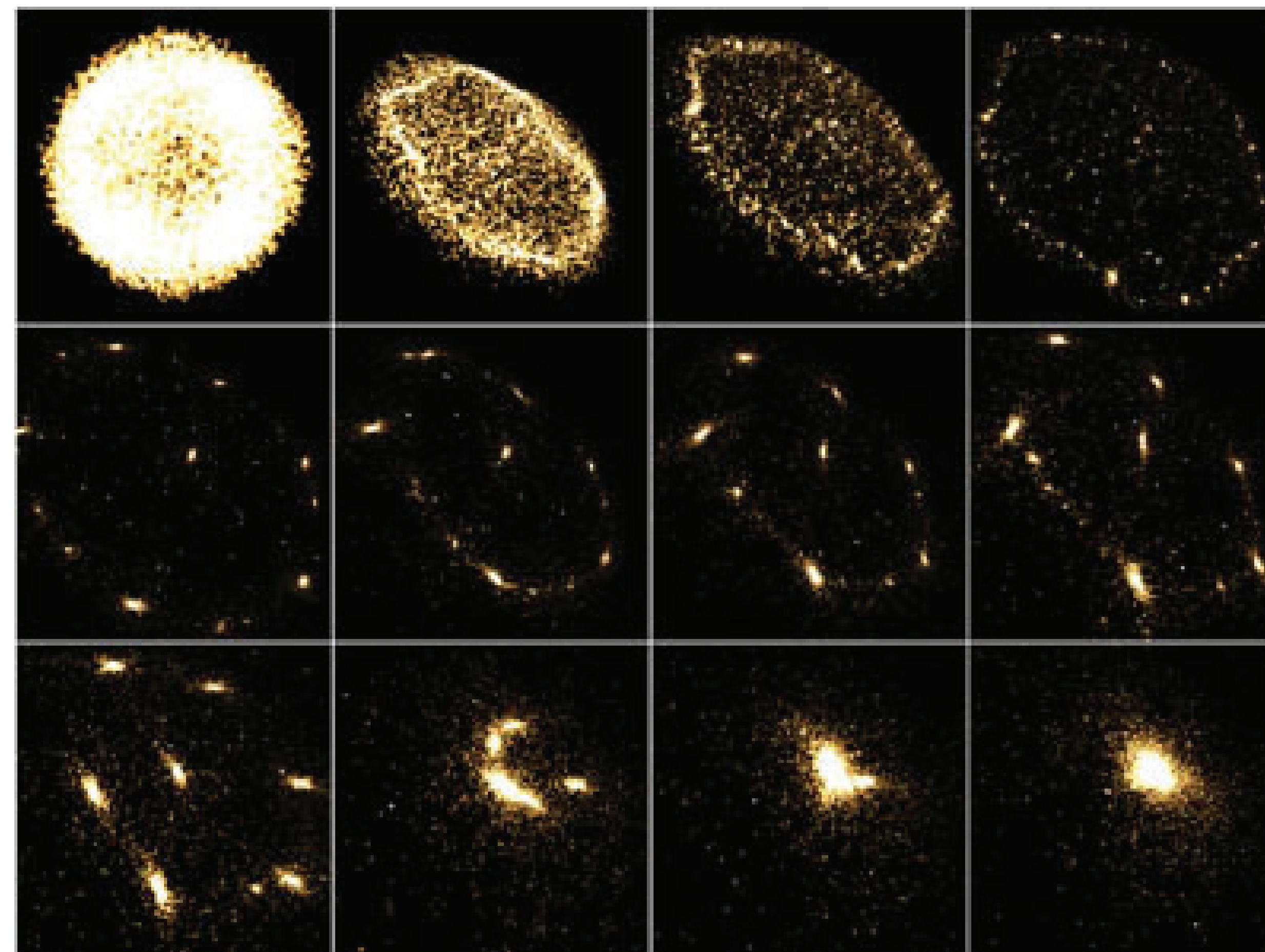
- Compute image histogram in Cuda
- Use this for histogram-based tone mapping



<http://resources.mpi-inf.mpg.de/departments/d4/teaching/ws200708/cg/slides/CG13-ToneMapping.pdf>

N-body simulation

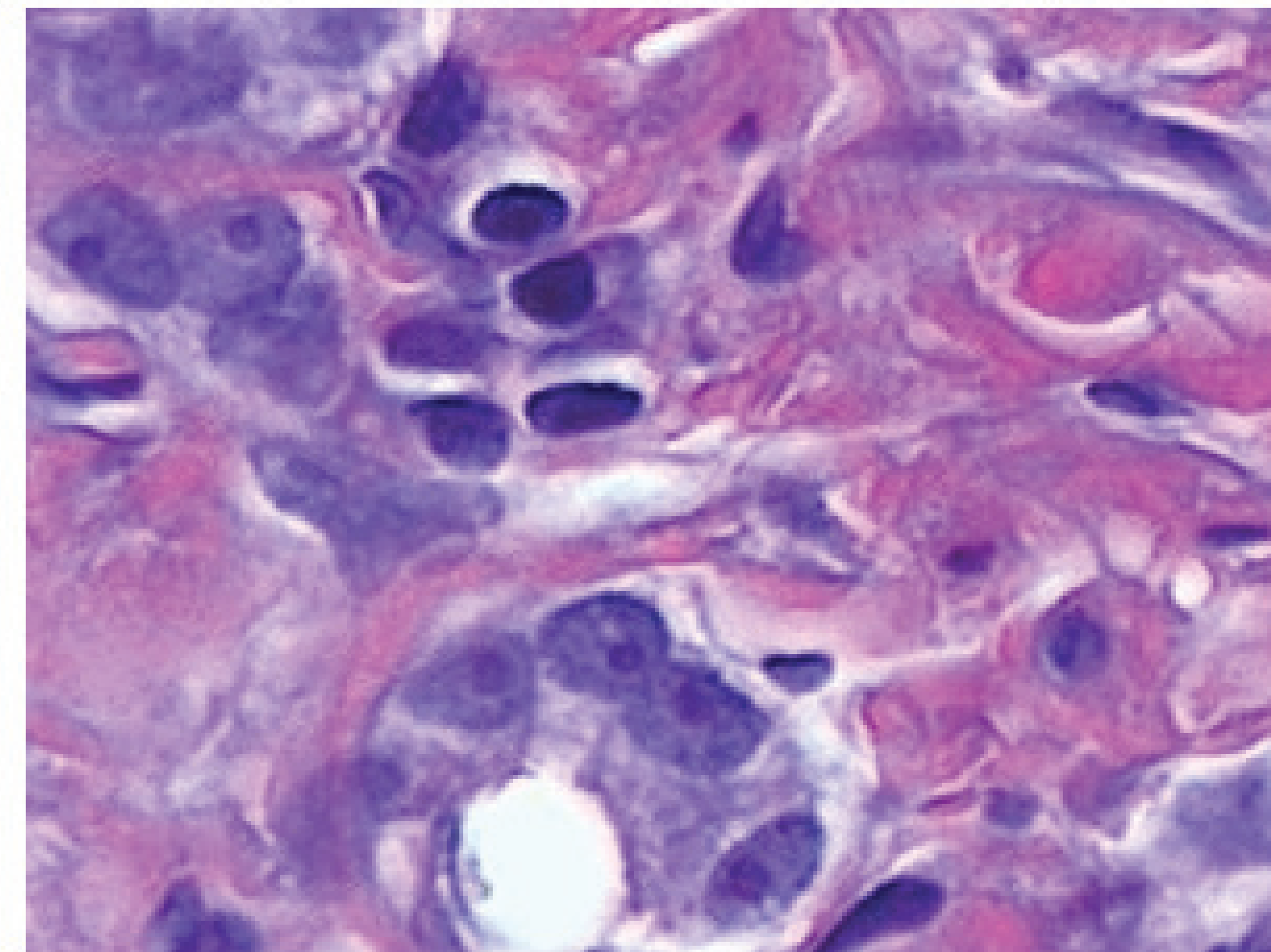
- Implement N-body simulation for gravitational potential
- Use OpenGL interoperability to visualize data



http://http.developer.nvidia.com/GPUGems3/gpugems3_ch31.html

Image segmentation using clustering

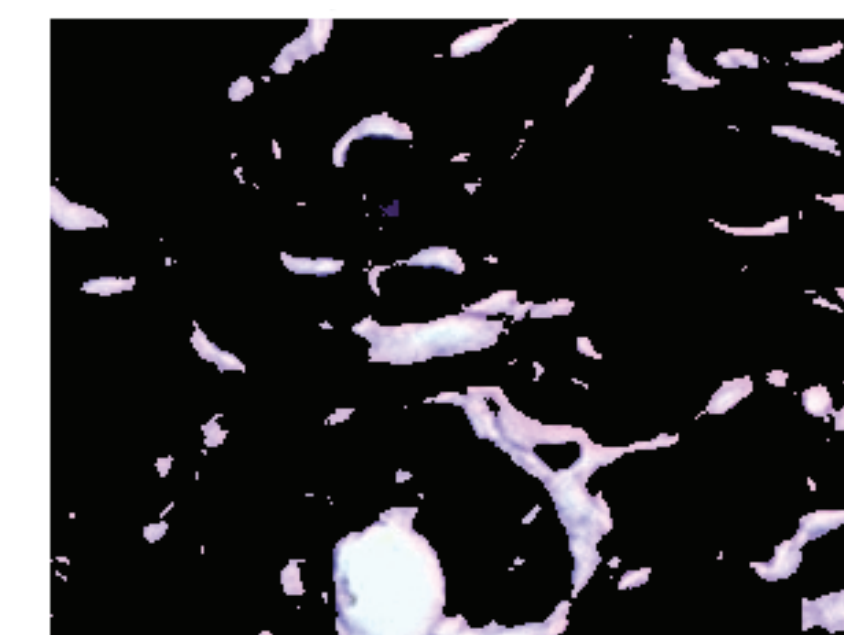
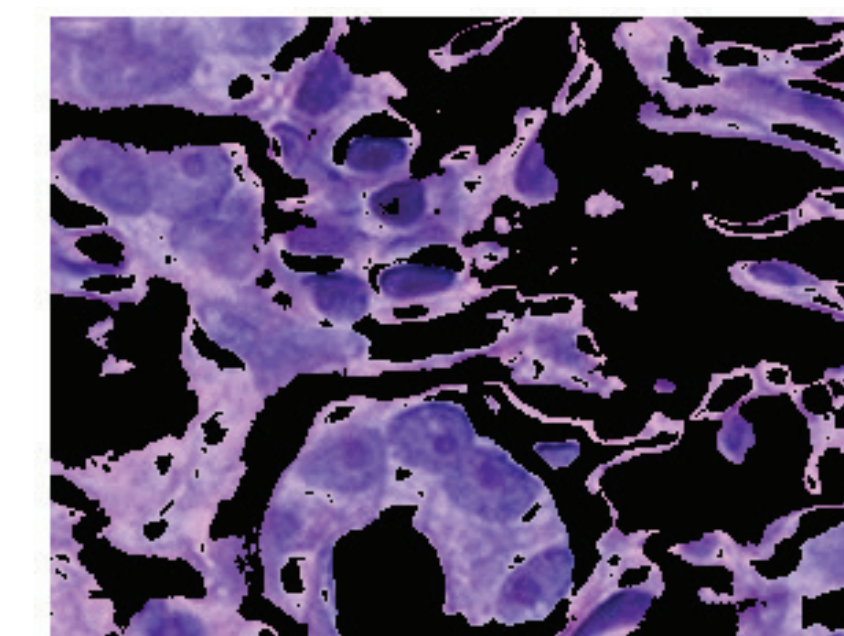
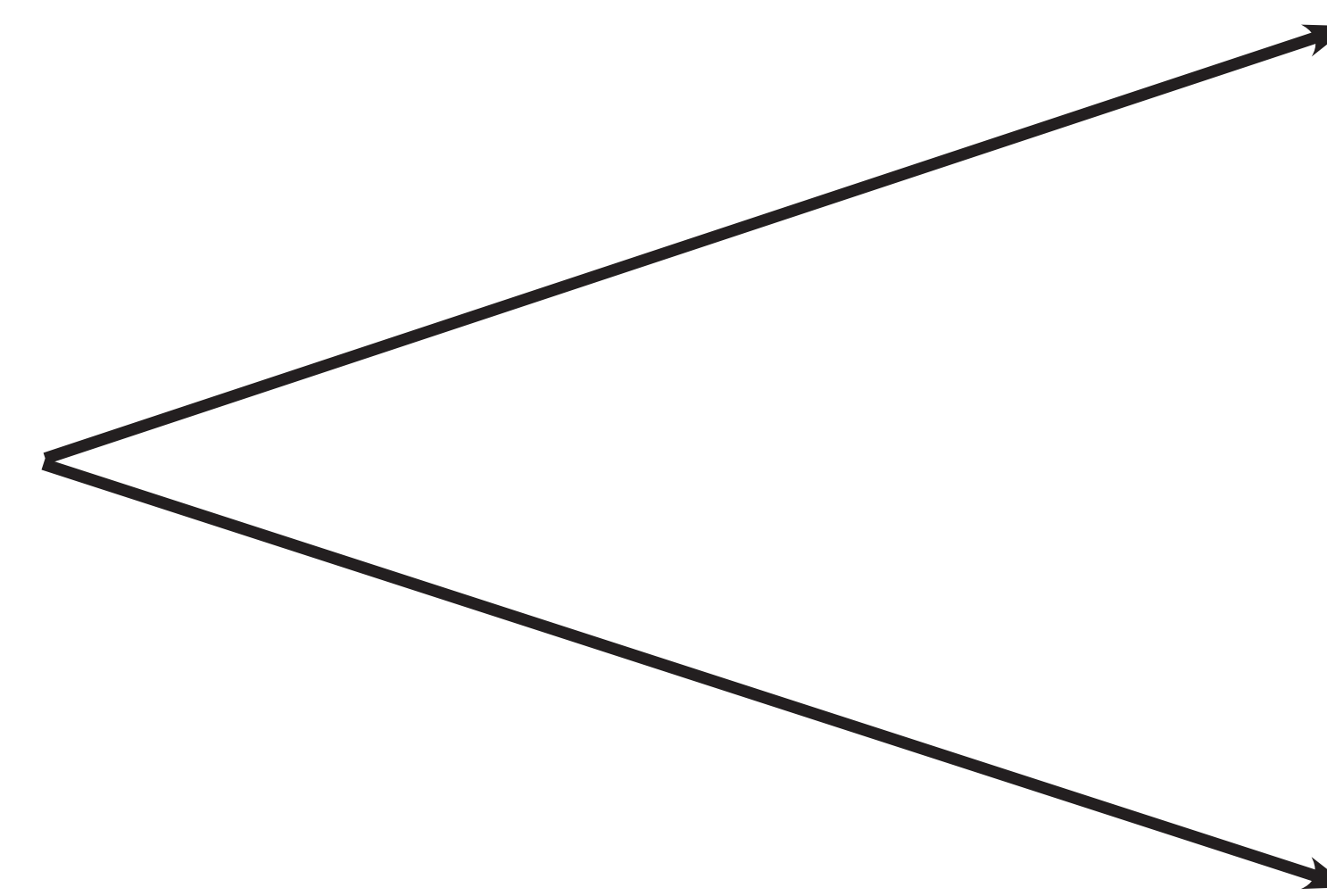
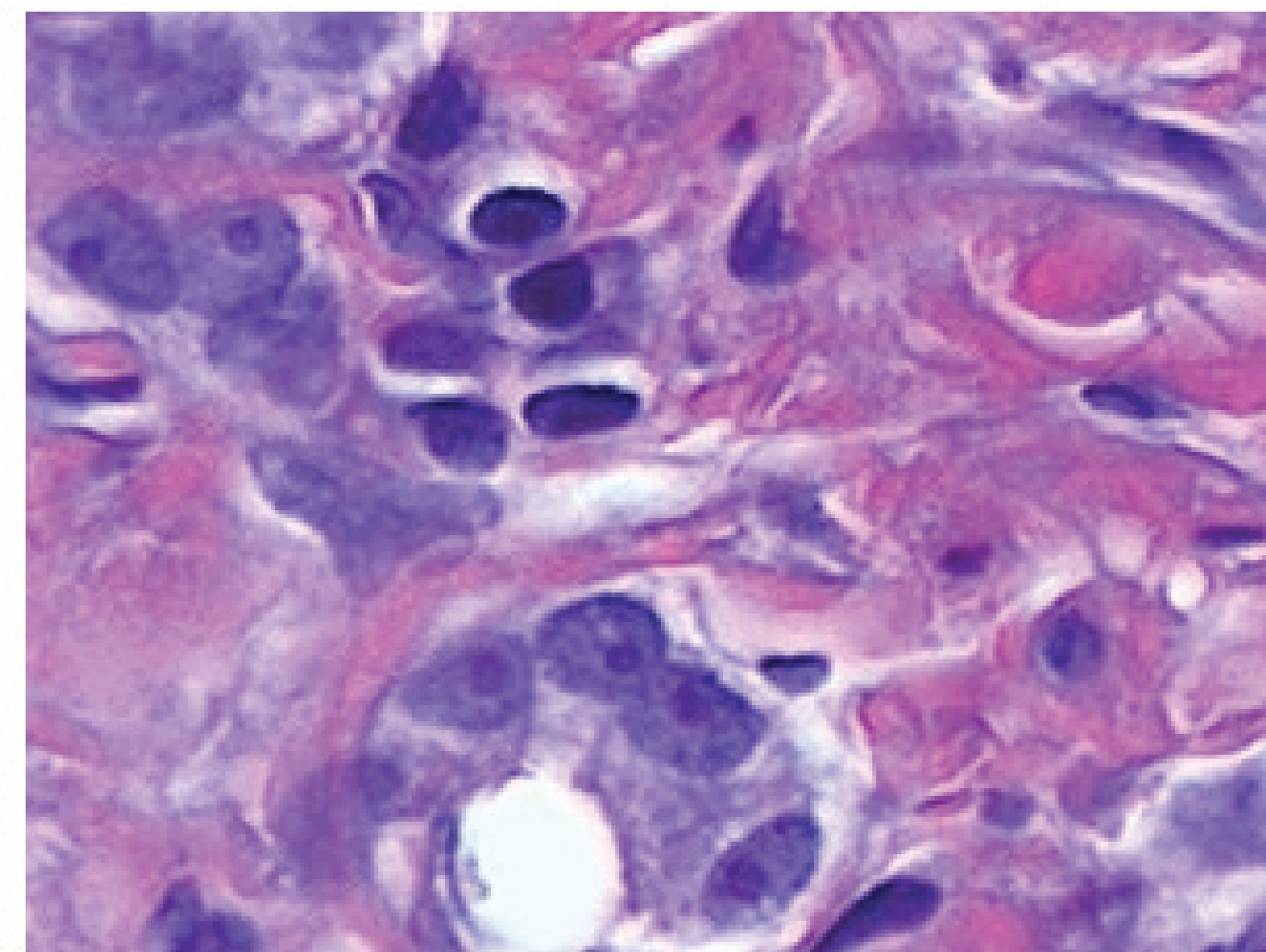
- Implement k-nearest neighbor
- Use it to perform image clustering



<https://de.mathworks.com/help/images/examples/color-based-segmentation-using-k-means-clustering.html?requestedDomain=www.mathworks.com>

Image segmentation using clustering

- Implement k-nearest neighbor
- Use it to perform image clustering



<https://de.mathworks.com/help/images/examples/color-based-segmentation-using-k-means-clustering.html?requestedDomain=www.mathworks.com>

Fast wavelet image compression

- Implement image compression / decompression using wavelets
- Explore the use of dynamic parallelism

