OpenGL GPGPU 2

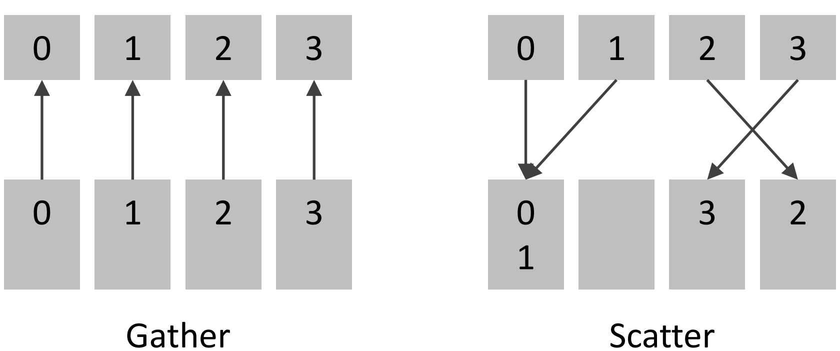
Ivo Ihrke **Tobias Ritschel** Mario Fritz

Today

- 3 Topics
  - **Histograms**
    How often does a certain value appear in a population?
  - **Compactification**
    Produce a compact list of primitives from a sparse input, e.g. features
  - **Voronoi diagrams**
    For every pixel, compute the index of the point in a set closest to it
- Underlying principle: Gather & Scatter

Change

- Until now:
  - Iterate over all pixels with index i
  - Read from some constant inputs (uniforms, textures,…)
  - Produce one or a low number for each pixel i
- Today, i will vary, e.g. in the vertex program
- This is called “scatter”, the old one “gather”



Problems

- What primitive: Points, lines, …
- Where?
- What combination: sum, multiply, …
- How to skip? Culling, discarding
- No order guarantees!

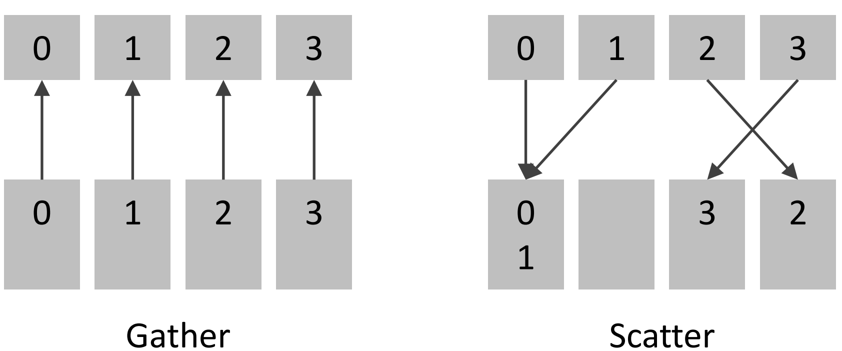
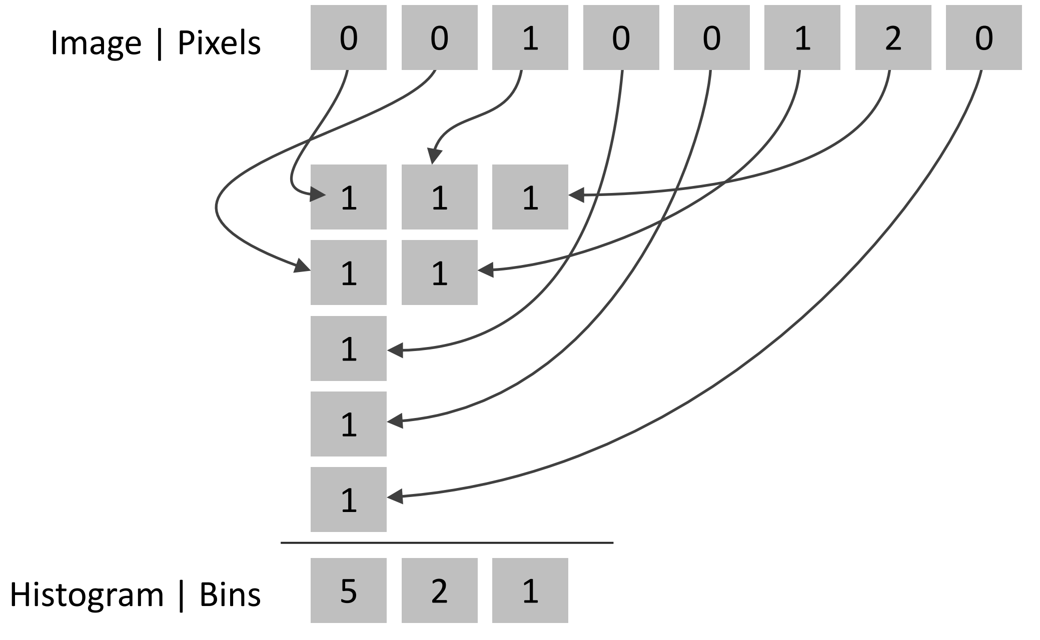


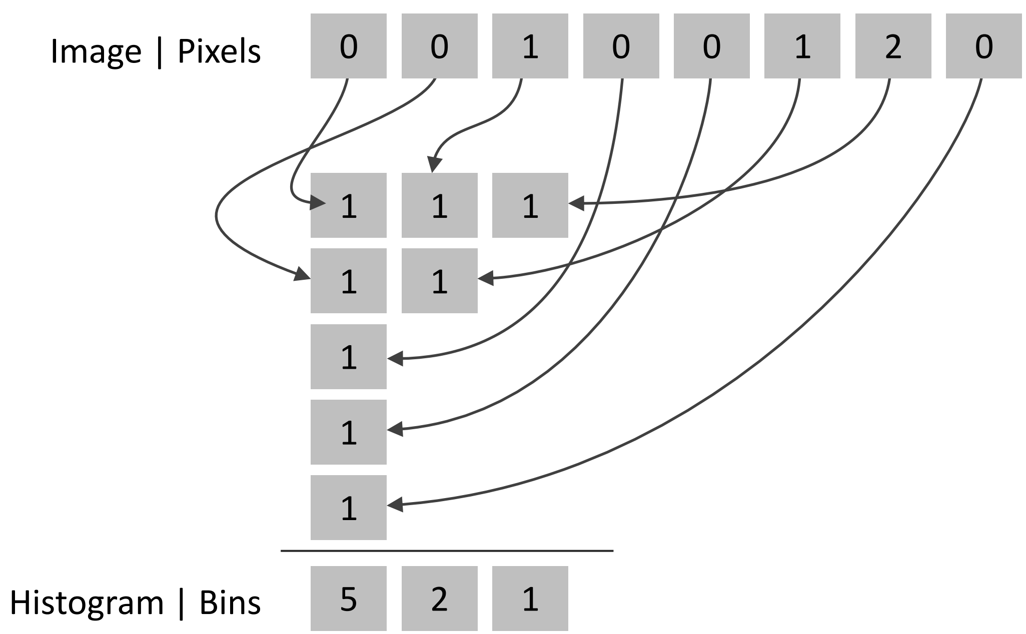
Image Histogram

- Draw a point for every pixel
- Use a vertex shader to sort it into a bin
- Combine using additive blending



Discussion

- Pro: Order-independent, parallelizes well
- Con: Scatter and atomic pressure



Visualization

- Result is 1D (or one pixel height-2D) texture
- Hard to visualize
- Cant draw to screen directly
- Solution:
  - Draw into 1D / flat texture
  - Display this texture in a new way
    - By stretching it
    - By drawing lines (Assignment)
- Drawing into textures: FBOs
- Basically these allow to render into a texture

CODE

Programming breakout

Histograms

Compactification: Problem

- The result of a computation is often sparse
- Example feature detection
- What we want is a **compact** feature list, e.g. their
  - Locations
  - Values

Image	0	0	1	1	1	1	0	0
Features	0	0	1	0	0	1	0	0
Compact features locations	2	5						
values	1	-1						

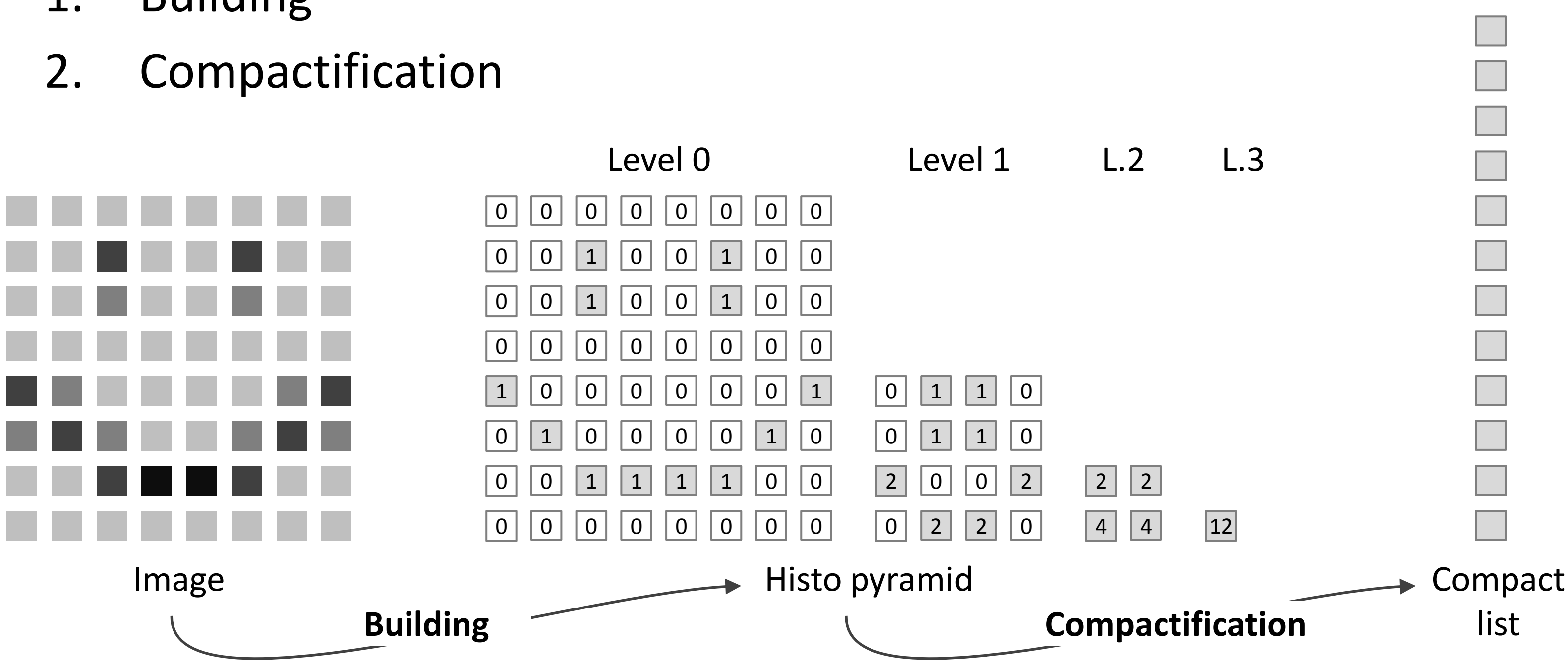
Solution: PrefixScan

- Two-step solution
  - First: Compute output count you have at input element i
  - Second: Compute output and put it where it belongs

Image	0	0	1	1	1	1	0	0
Output count	0	0	1	0	0	1	0	0
Prefix scan	0	0	0	1	1	1	2	2
Output	0	0	1	0	0	-1	0	0
Compact features locations	2	5						
values	1	-1						

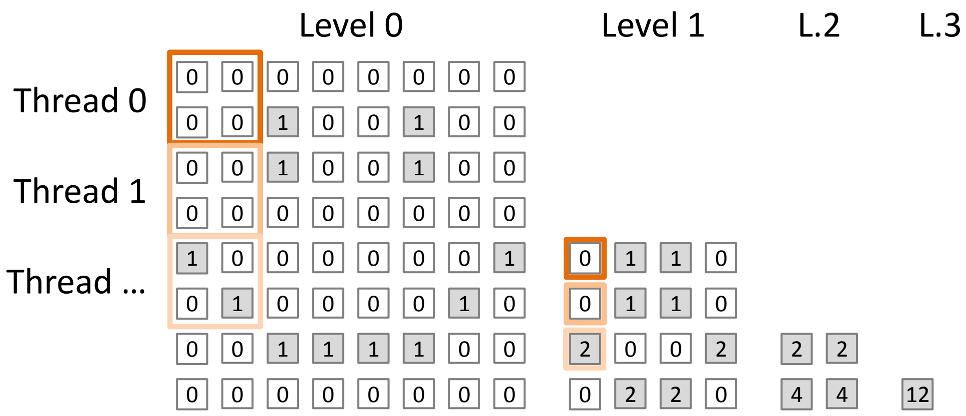
Example: HistoPyramids

- For 2D Images
- Using MIP hierarchy
- Two steps:
  1. Building
  2. Compactification



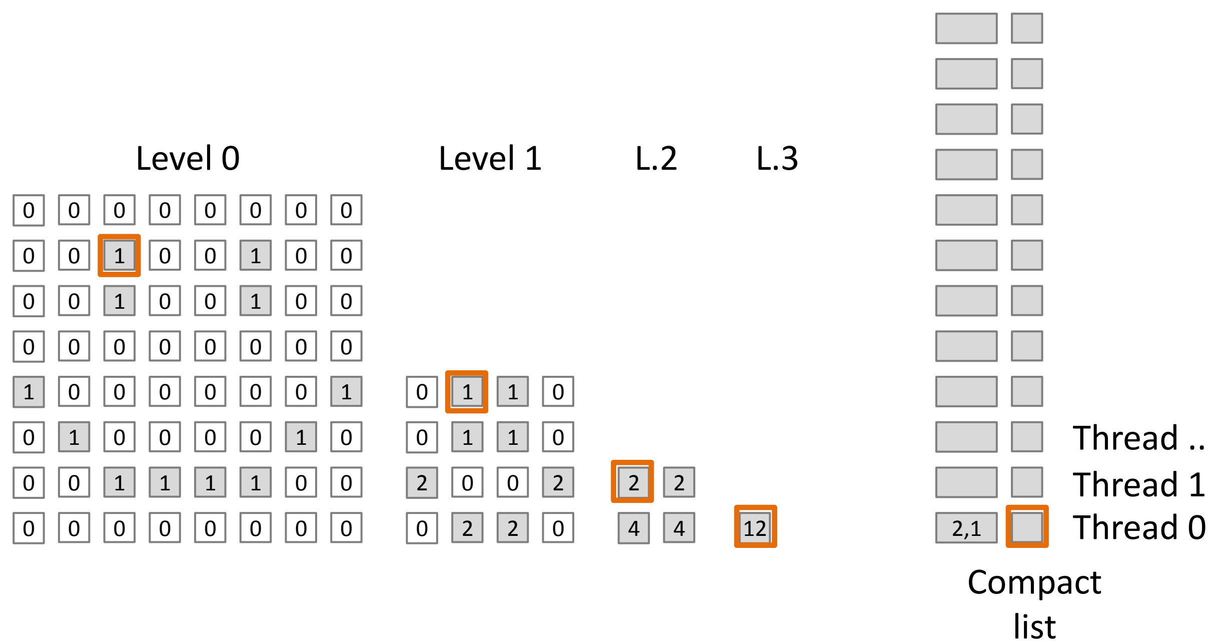
Build step

- Put 0/1 into level 0 (finest!) of a MIP map
- In higher levels, sum all 4 pixels below it
- Multiple passes
- “Final” pixel contains count



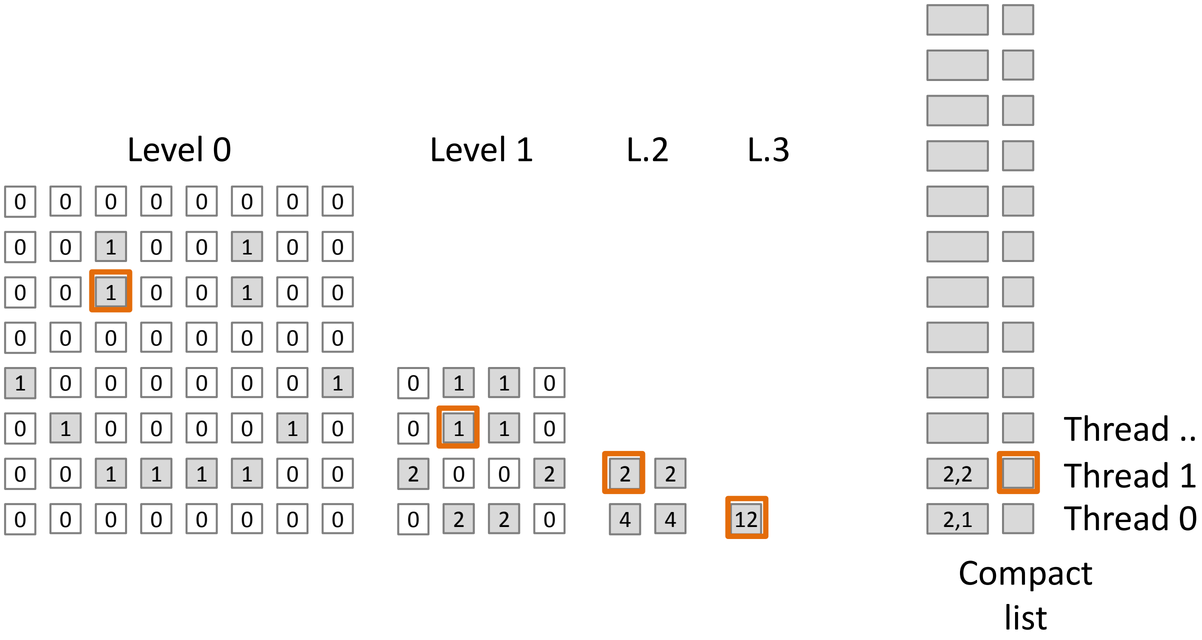
Compactification step

- Start a thread for every



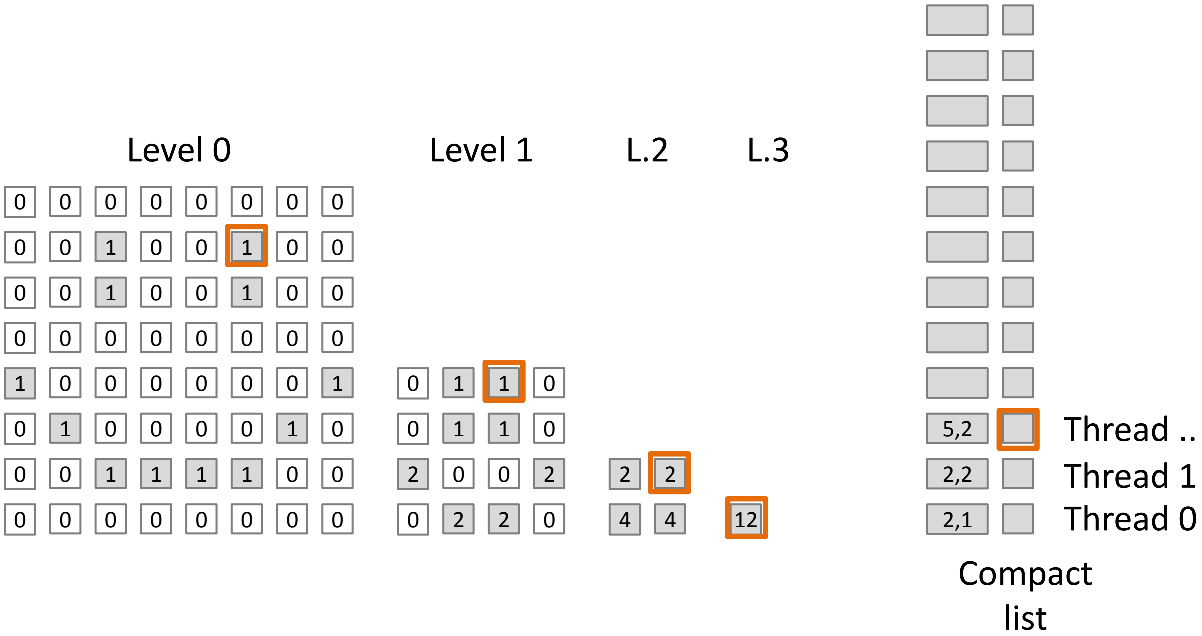
Compactification step

- Start a thread for every



Compactification step

- Start a thread for every

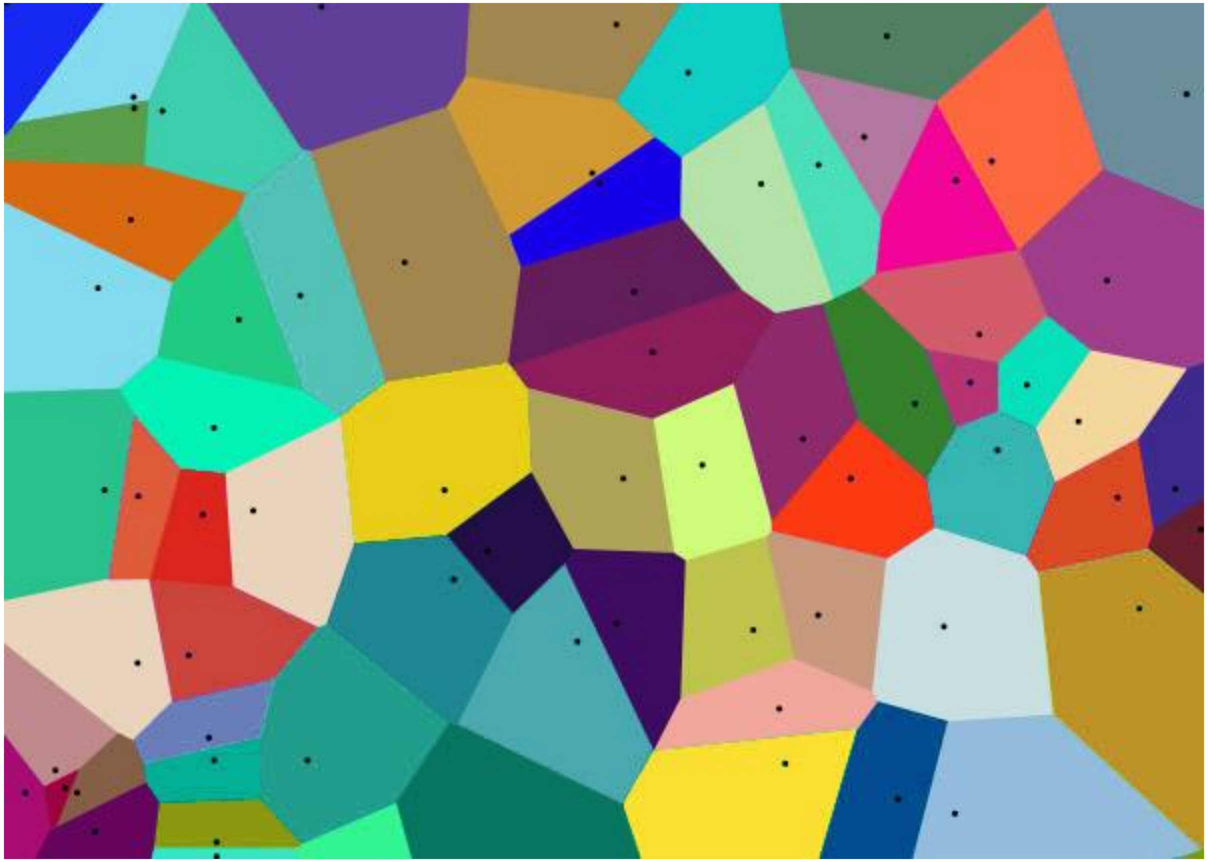


CODE

Programming breakout

Histo Pyramids

Voronoi Diagram

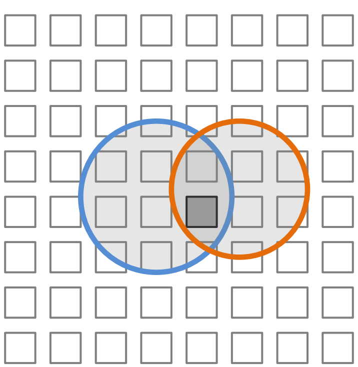
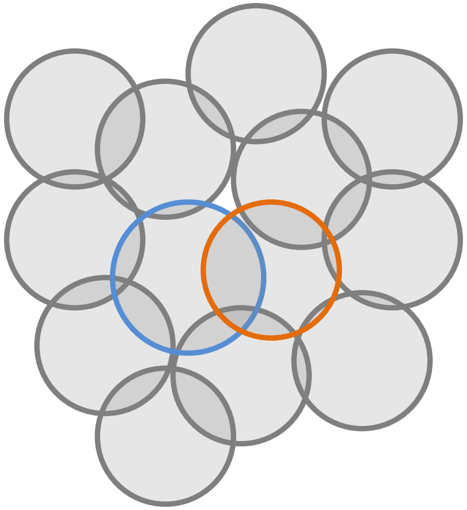
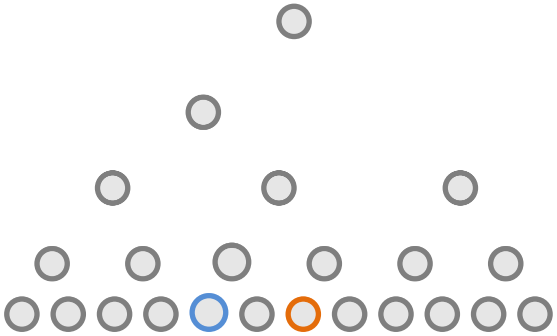


Paralle Voronoi Diagram

- Trivial:
  - For all pixels in parallel
  - Loop over all centers and find the smallest
- $O(nm)$ for n pixels and m centers, parallel over n
- Goal:
  - Get m down to constant by making some assumptions
  - Parallel over n and m

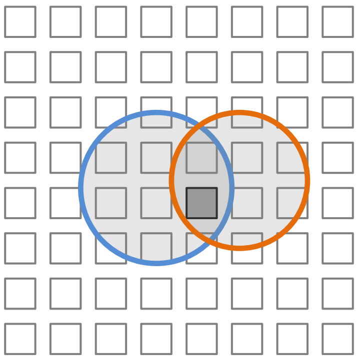
m constant

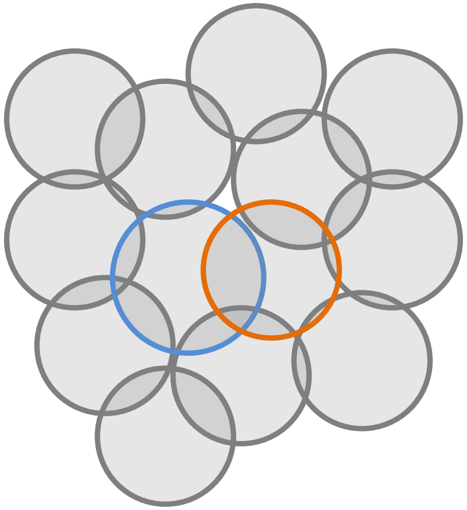
- Assume a maximal distance r
- Visit only those pixels in a radius r around every center
- Gather-type operation:
  - For all n pixels in parallel
  - Find centers closer than r, maybe using k-d trees?

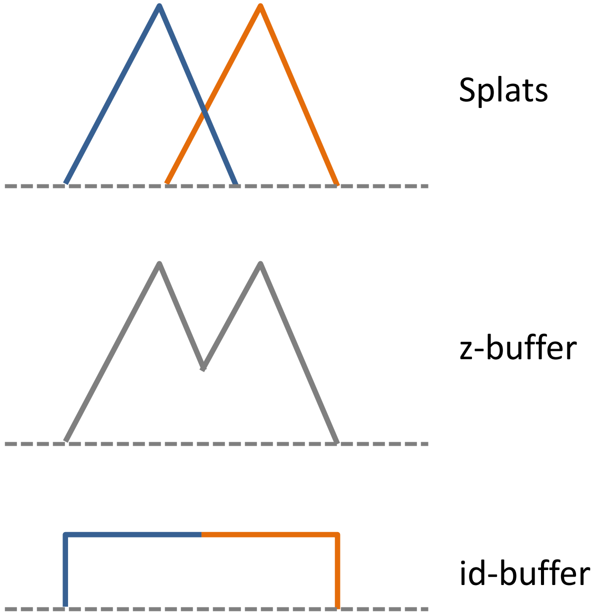




nm parallel

- Just draw 2D disks / boxes / … of size r
- This **visits** only those pixels in a radius r around every center
- Scatter-type operation:
  - For all $m\ r^2$ pixels in parallel
  - Find minimum index using z-buffering







CODE

Programming breakout

Voronoi Diagrams