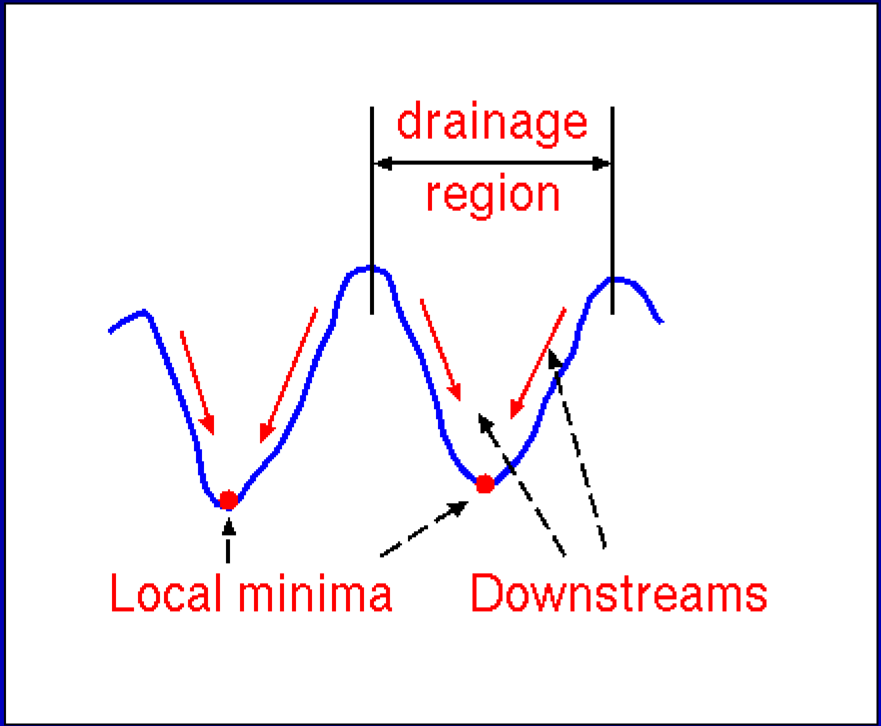
Segmentation of Tomographic Data by the Hierarchical Watershed Transform

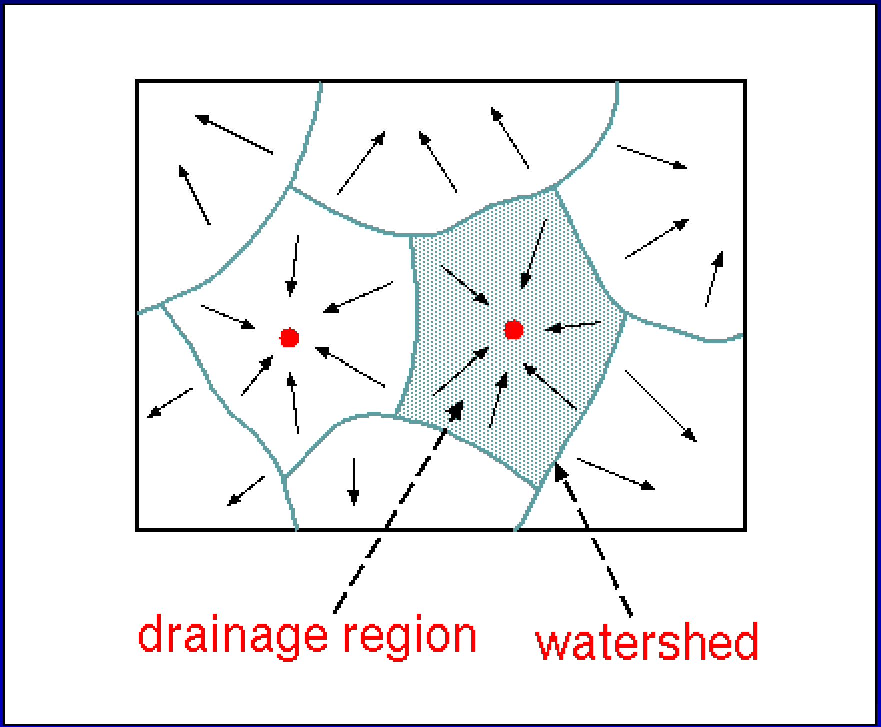
Goal: partition a volume to homogeneous region

The Watershed Concept (1)

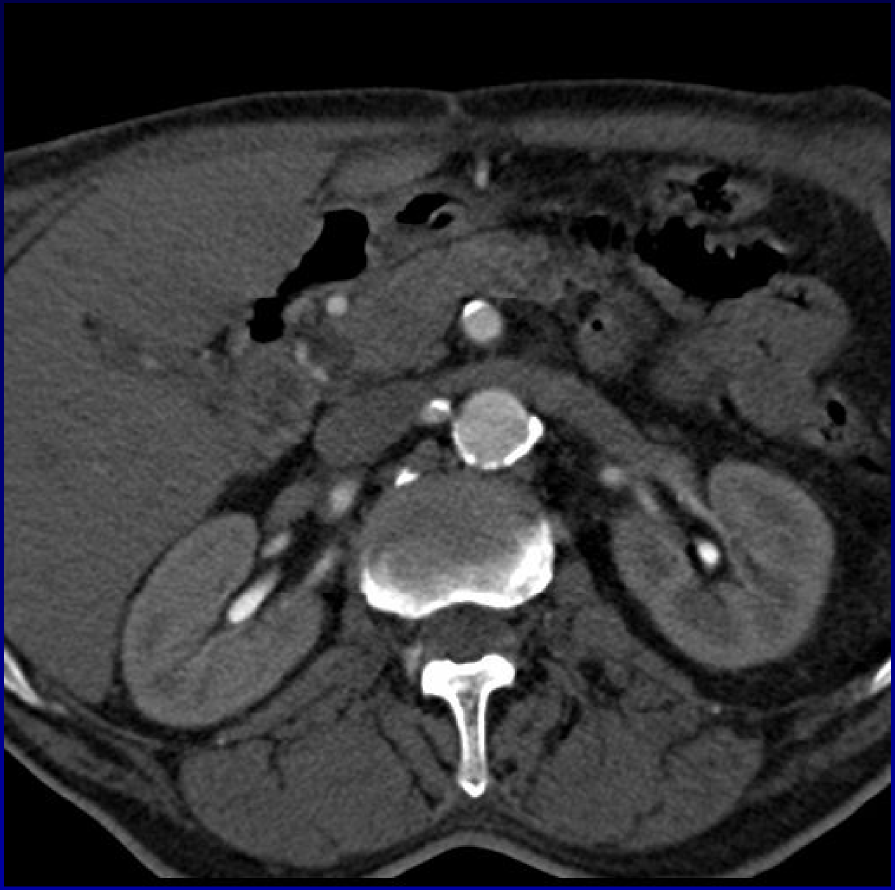
The Watershed Concept (2)

- **Waterflow simulation on gradient images:**
  - **Catchment basins & watershed lines**



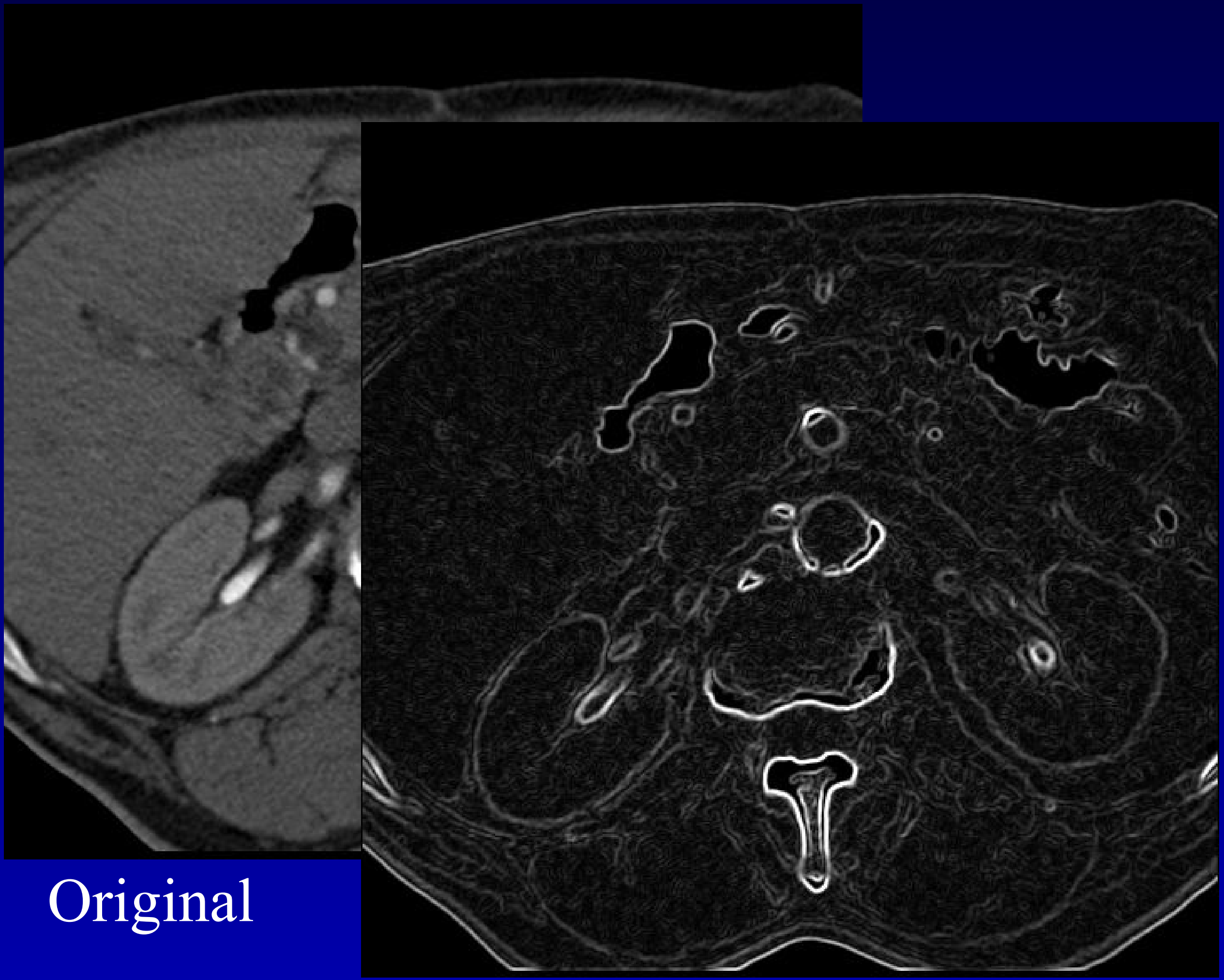


Watershed Implementation



Original

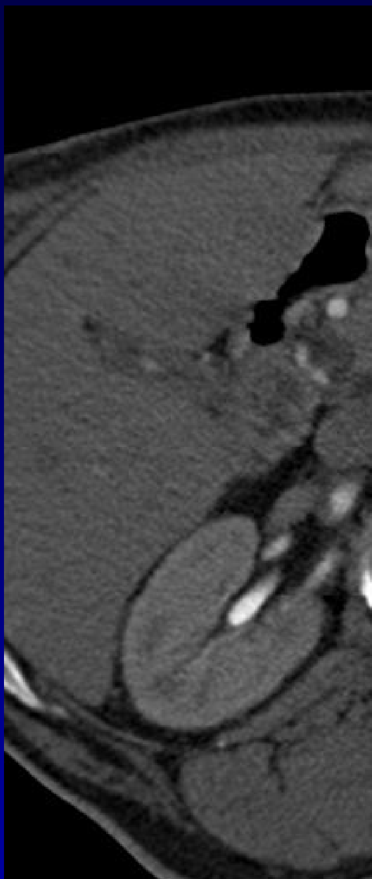
Watershed Implementation



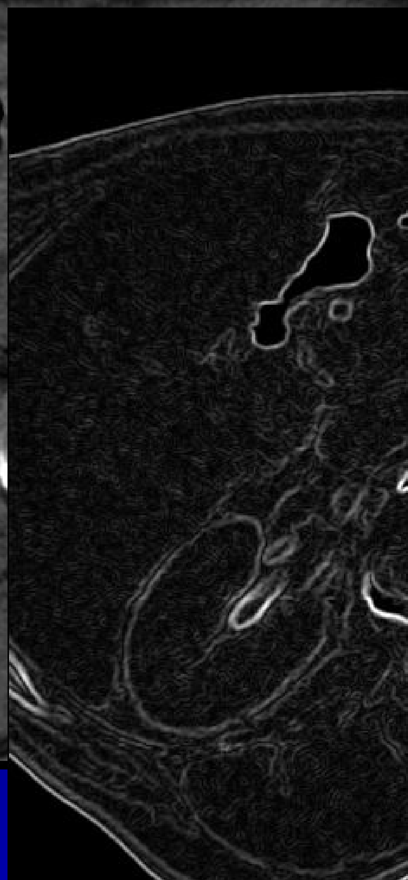
Original

Sobel edges

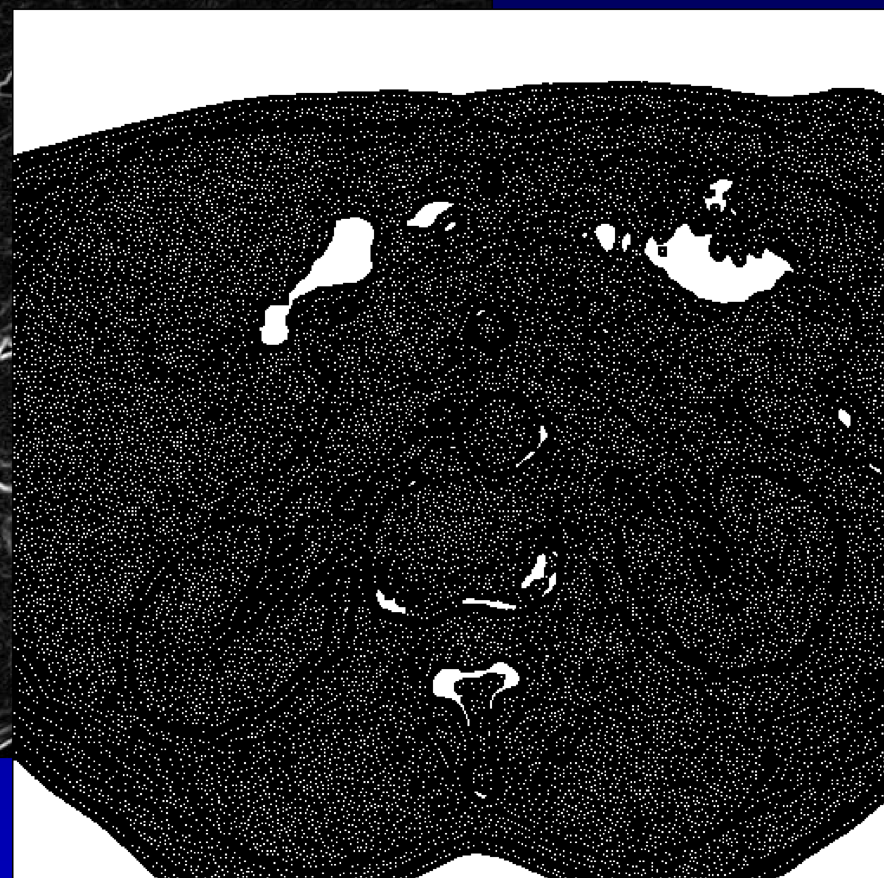
Watershed Implementation



Original

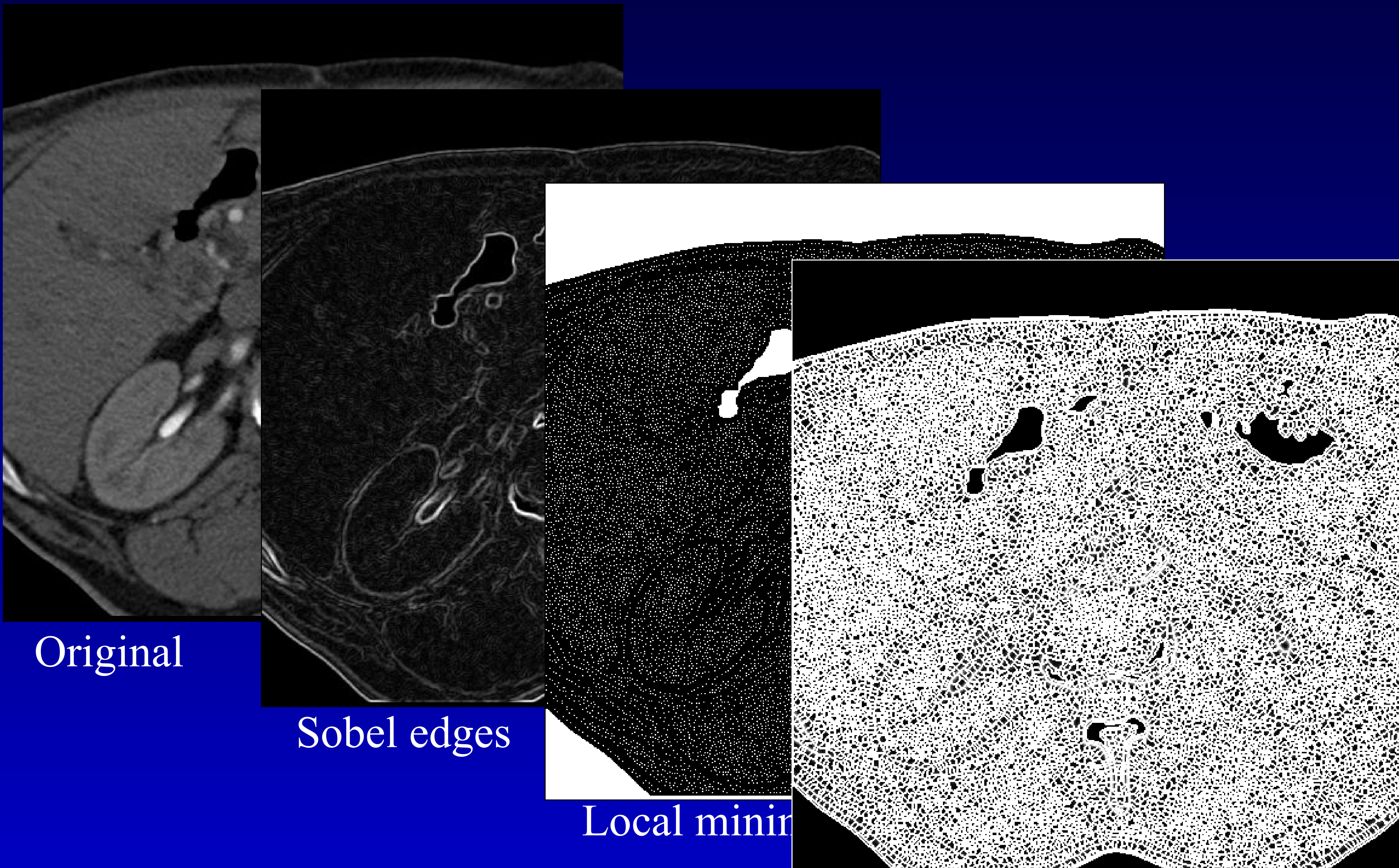


Sobel edges



Local minima

Watershed Implementation



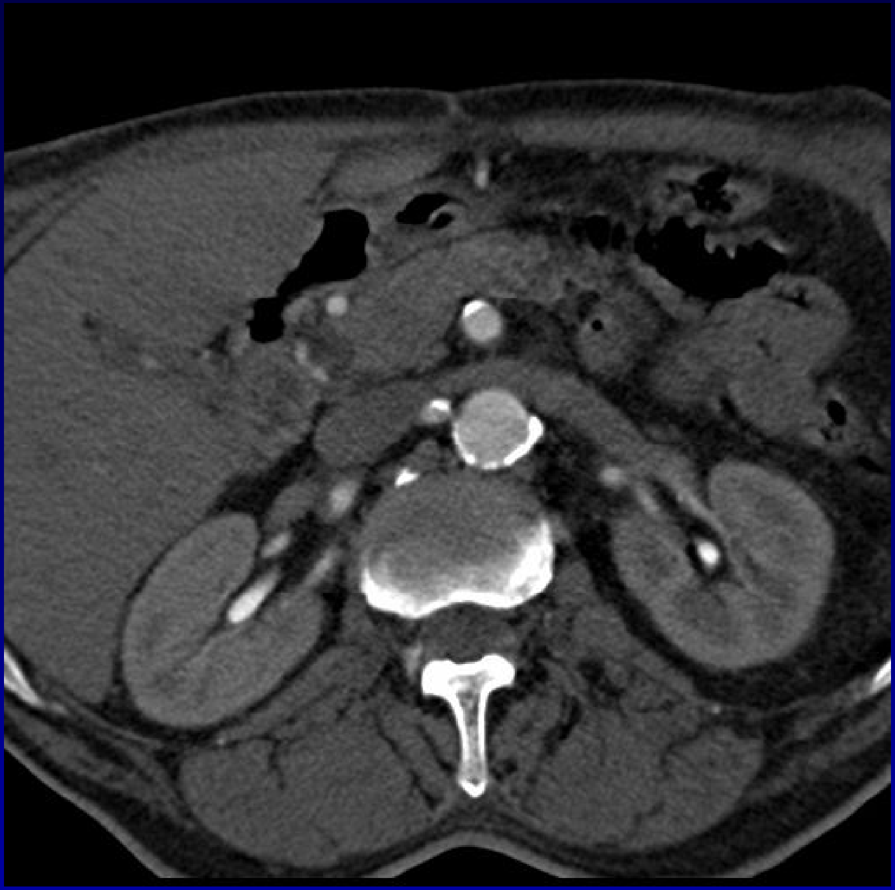
Original

Sobel edges

Local minin

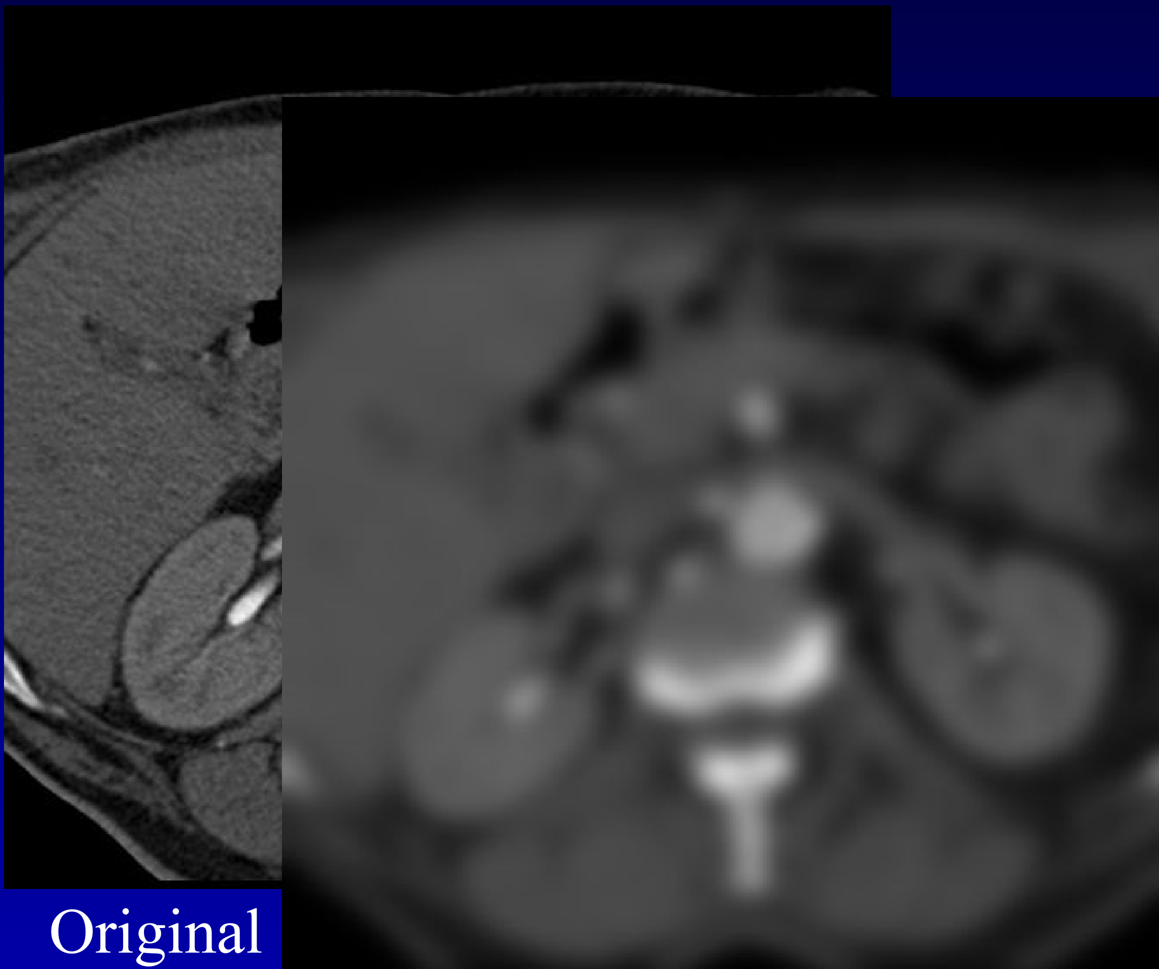
Region bondaries – watersheds

Large Regions by Gaussian Smoothing



Original

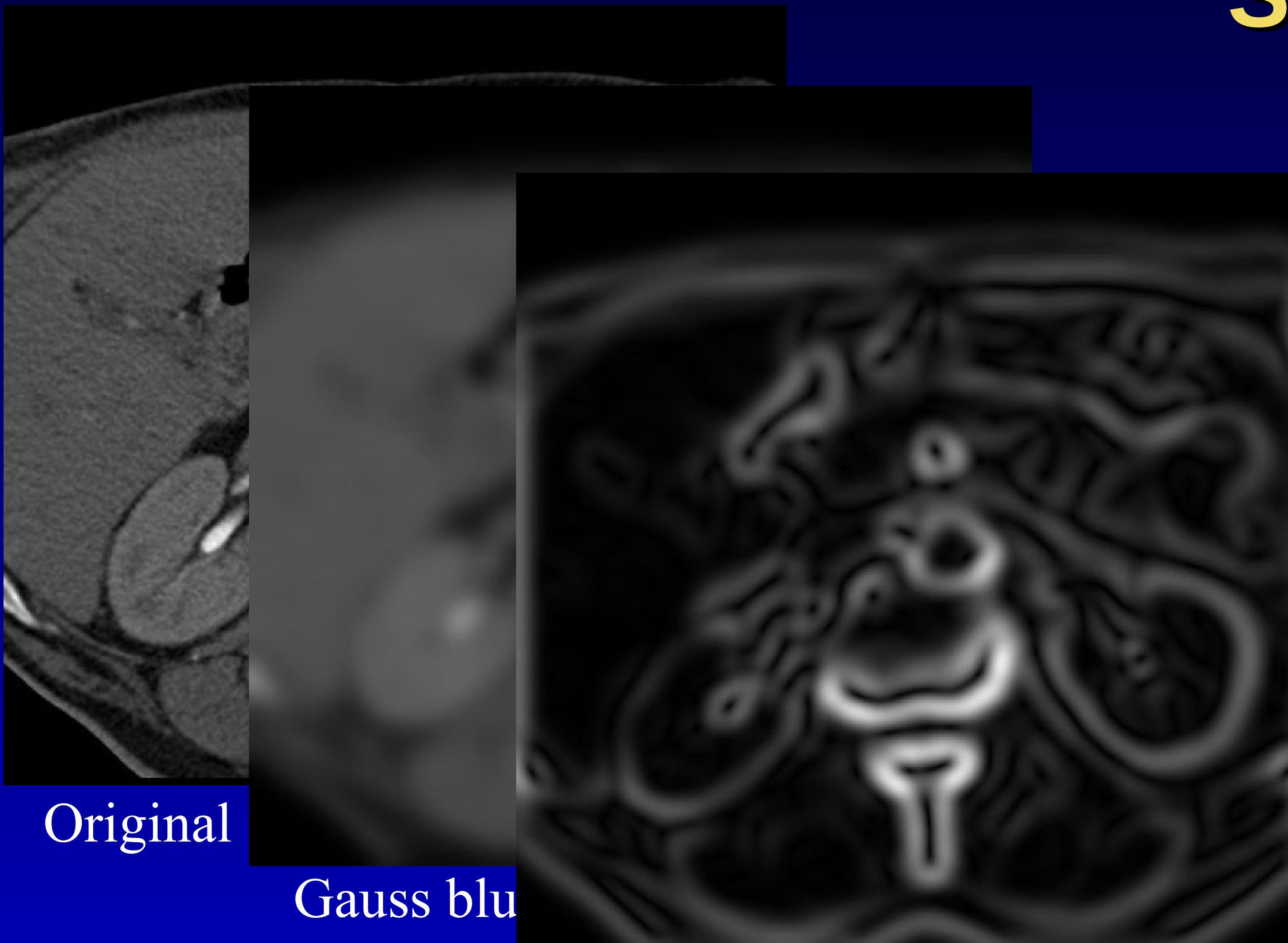
Large Regions by Gaussian Smoothing



Original

Gauss bluring, σ=8.0

Large Regions by Gaussian Smoothing

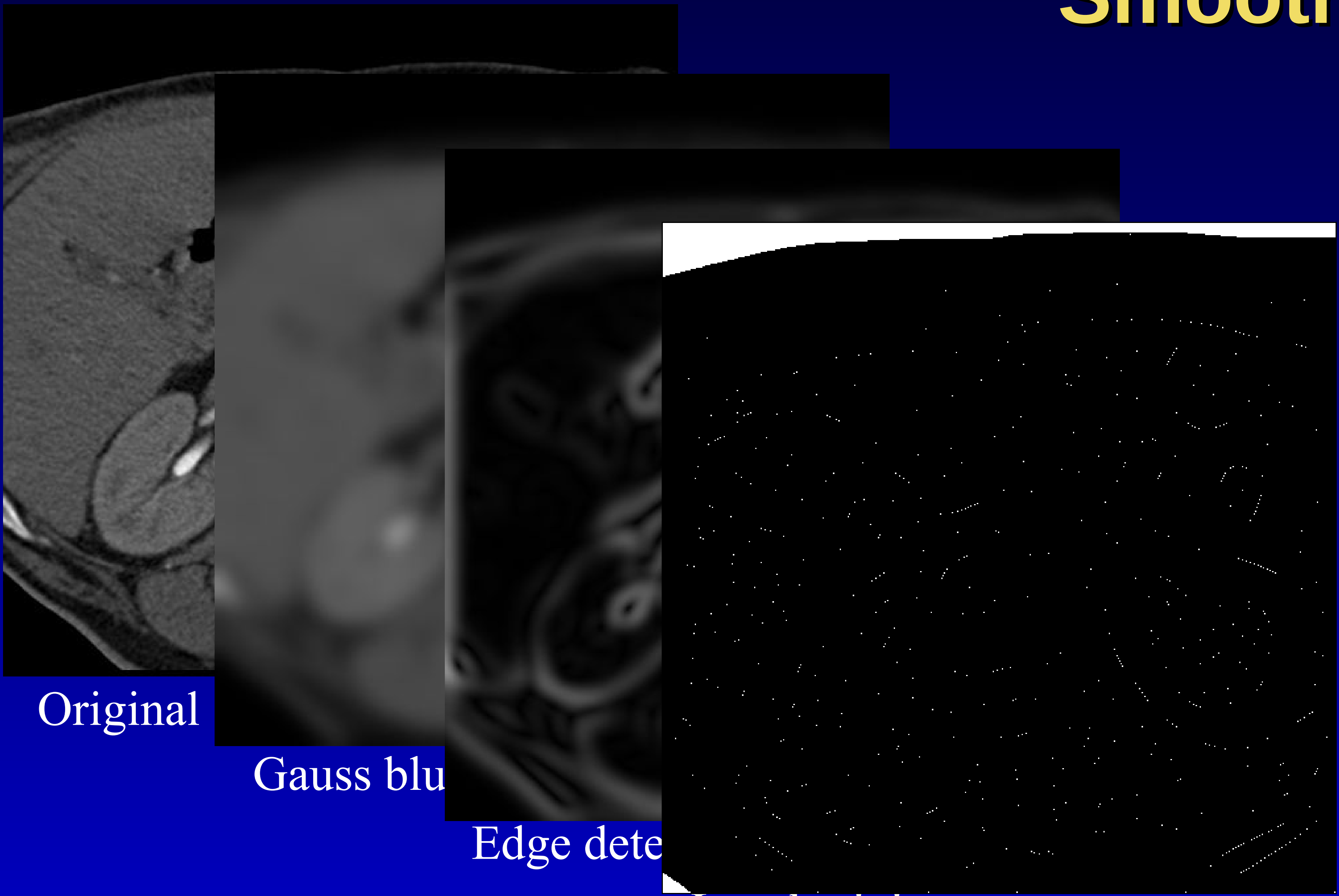


Original

Gauss blu

Edge detection

Large Regions by Gaussian Smoothing



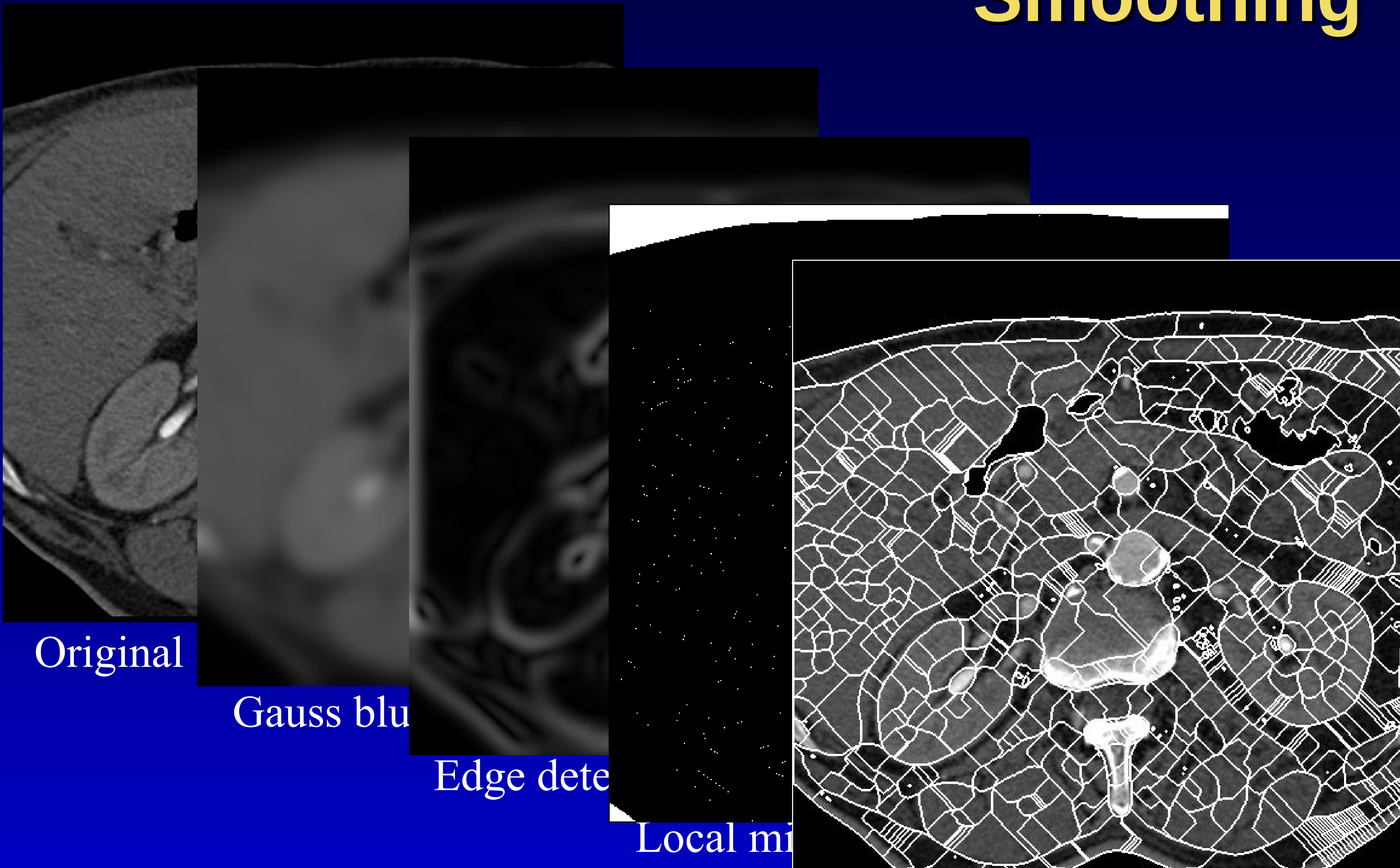
Original

Gauss blu

Edge dete

Local minima

Large Regions by Gaussian Smoothing



Original

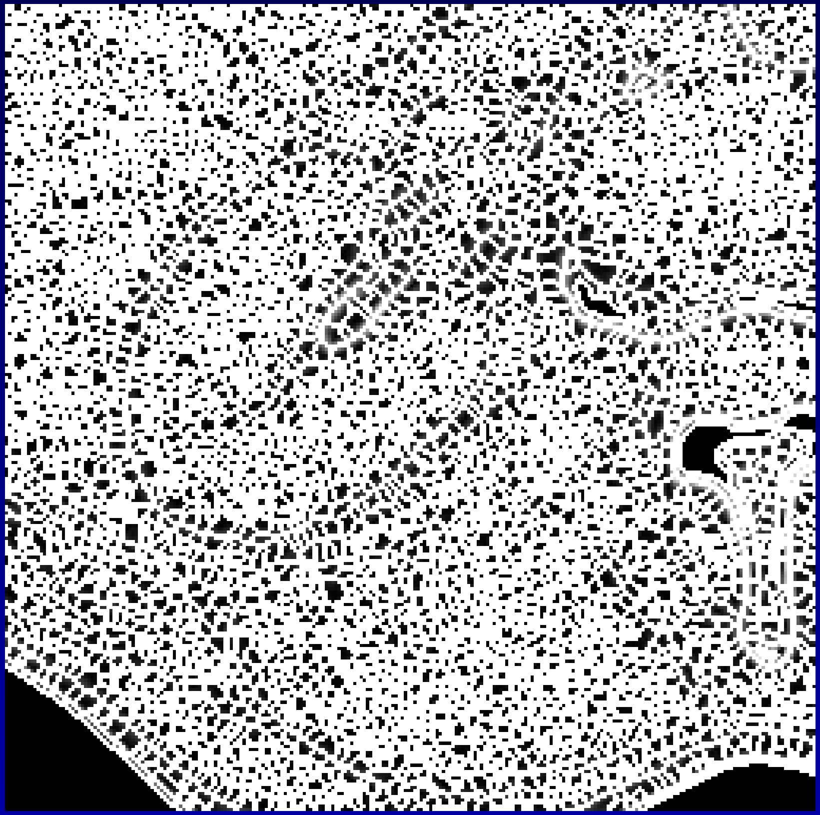
Gauss blu

Edge dete

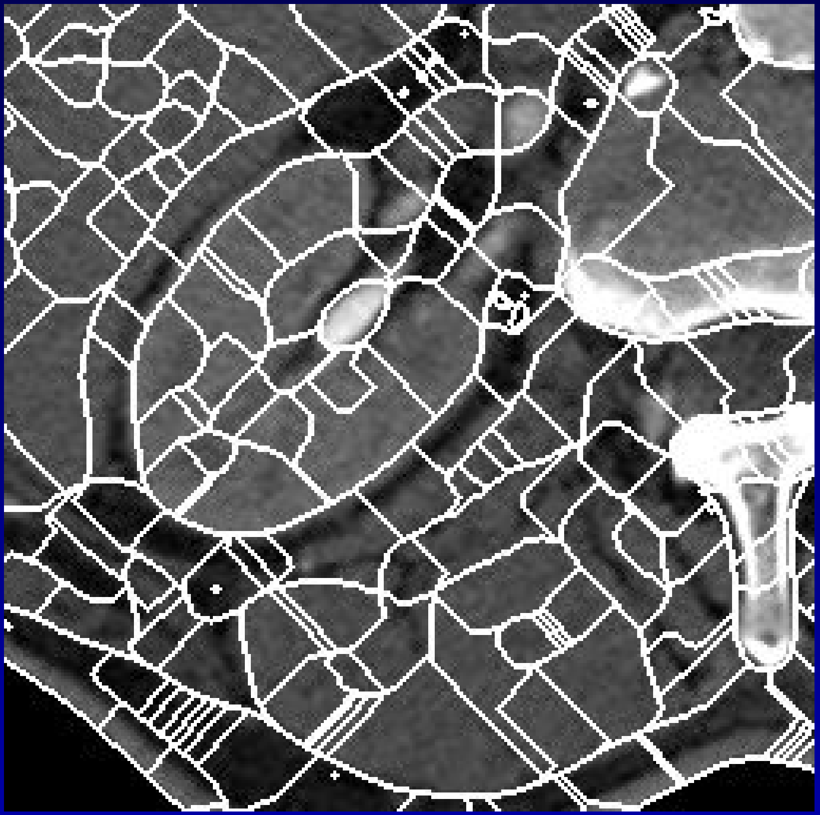
Local mi

Region bondaries – watersheds

Watershed Drawbacks



No smoothing: numerous small regions



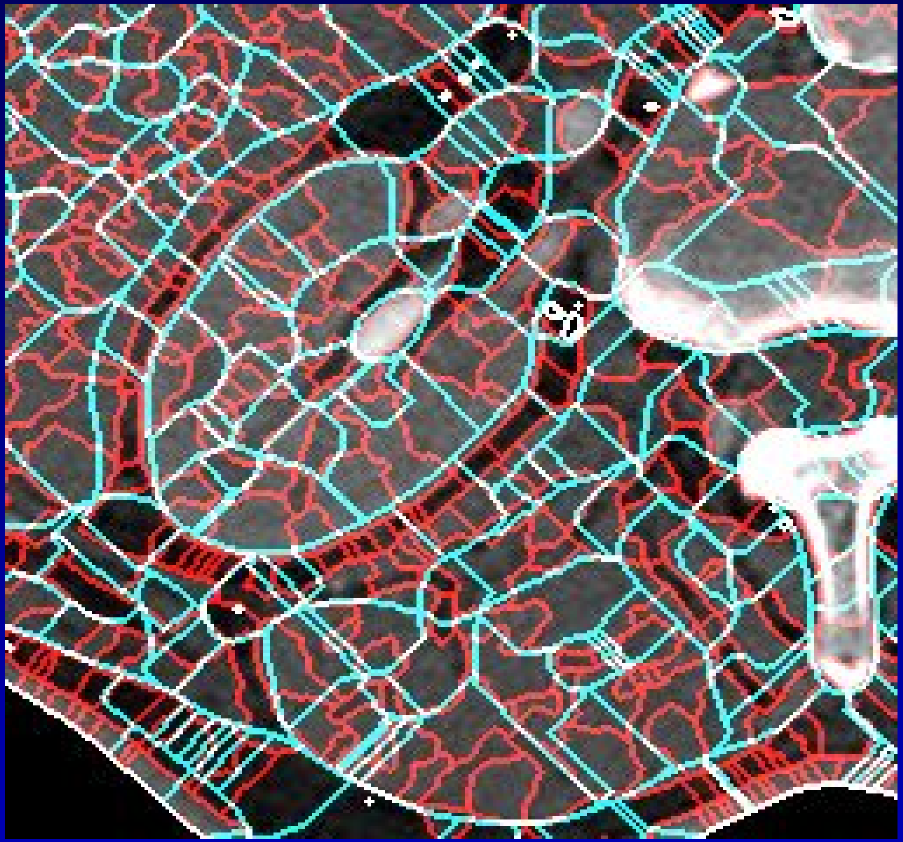
Smoothing: fewer regions but imprecise contours

The Hierarchical Watershed Transform (HWT)

- **Aimed to override the problems:**
  - ***Build large regions with precise contours***
- **The algorithm:**
  1) **Build a sequence of watershed segmentations with an increasing Gaussian σ_i**
  2) **Starting from level σ_0, label each region at level σ_i with a label of the region at level σ_{i+1}** with the highest number of overlapping pixels.

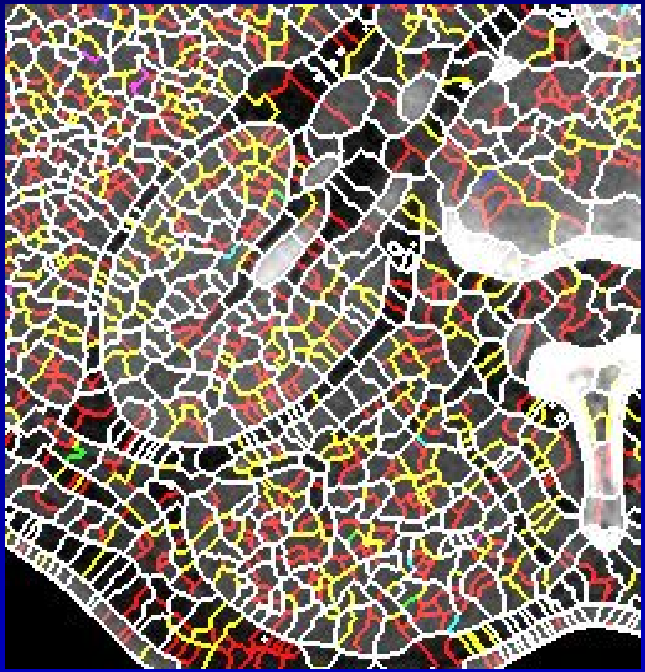
Region Overlap & Merging

- **Red contours, precisely at region boundaries: level σ_i**
- **White/Cyan contours, imprecise: level σ_{i+1}**

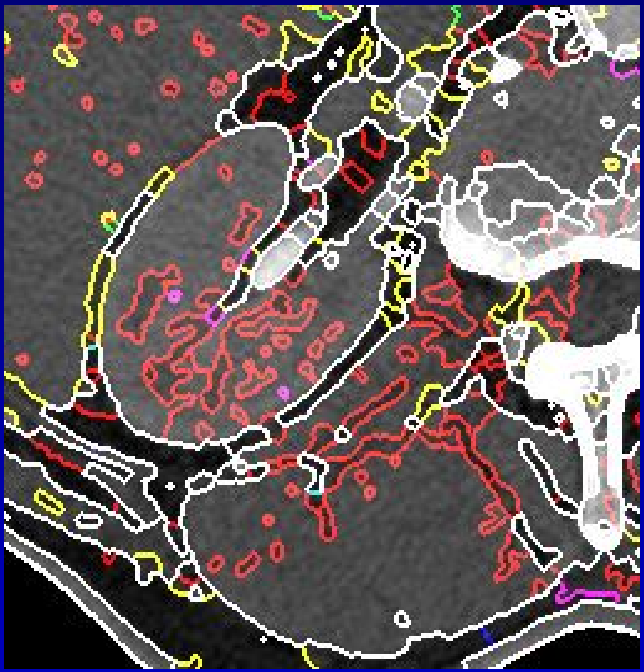


HWT Applications

- **Multiresolution region hierarchy for image analysis**
- **Region merging based on spectral similarity or edge height**



Multiresolution hierarchy



Mean density merging: distance 10, 15 and 20

Current Implementation

- A 2 pass algorithm
  - Find local minima
  - For each voxel
    1. Go down, unless minimum or labelled voxel is found
    2. Go uphill, label the downhill path by the minimum label
- No streaming
- Local, but unpredictable

A Similar Task: Region Labeling (RL)

- Task: assign individuall labels to isolated foreground regions
- A two pass algorithm
  1. Pre-label and build corresponcence table
     - If FG neighbor labeled – get the label
     - If not labeled – start new label
     - If two nighbors have a different label – store in the table
  2. Final labeling using the table

Labeling of Local Minima

- Similar to RL, differences in
  - FG voxel: minimum in 3x3x3 neighborhood (alternatives possible)
  - Such voxel may belong to plateau
  - Correspondence table
  - Passes
    1. Is 3x3x3 local
    2. Is plateau
    3. Dilation, closing (merge close minima)
    4. Region labeling
  - Output: Labeled minima

Watershed Detection – Thick Slab

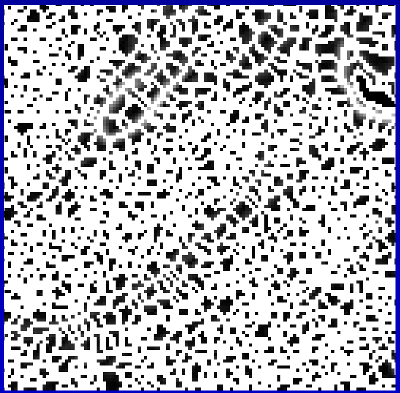
- Slice loading and releasing according to number N of processed voxels :
  - Loaded if N>0 (a downhill processing reaches the slice)
  - Released if N==nx*ny (all voxels processed)
- Advantage: straightforward
- Disadvantage:
  - Unpredictable number of slices stored
  - Cache unfriendly

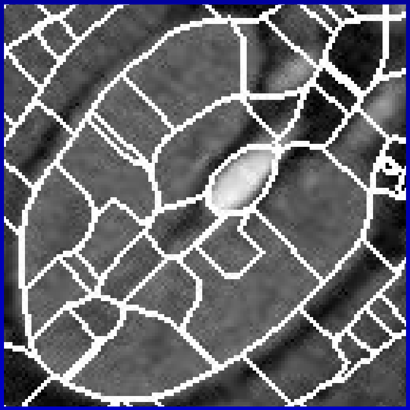
Watershed Detection – RL Style

- Features
  - Two slices required
  - Local decision about downhill descend
  - Labeling and correspondence table
- Advantages: Two slices
- Disadvantages
  - Is possible at all?

Hierarchy Building

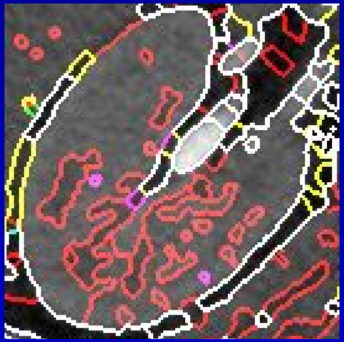
- WT at two sigmas $\sigma_1 < \sigma_2$
- Labeling of σ_1 level according to the σ_2 level by overlapping
- The thick slab approach?
  - All active regions in memory





Region Merging

- Merge regions with similar density
  - Build neighbor list
  - Sort the list according to density similarity
  - Start merging from the nearest neighbors up to a predefined threshold



Visualization and Segmentation

- Interactive segmentation based on real-time GPU-based visualization
  - Region specification by
    - Size, neighbor relation, WS hierarchy etc...
- Visualization
  - Transfer functions based on
    - Region density
    - Density difference between neighboring regions

Implementation Status

- In-memory HWS transform in the f3d v.5

Implementation Status

- In-memory HWS transform in the f3d v.5

Thank You!