

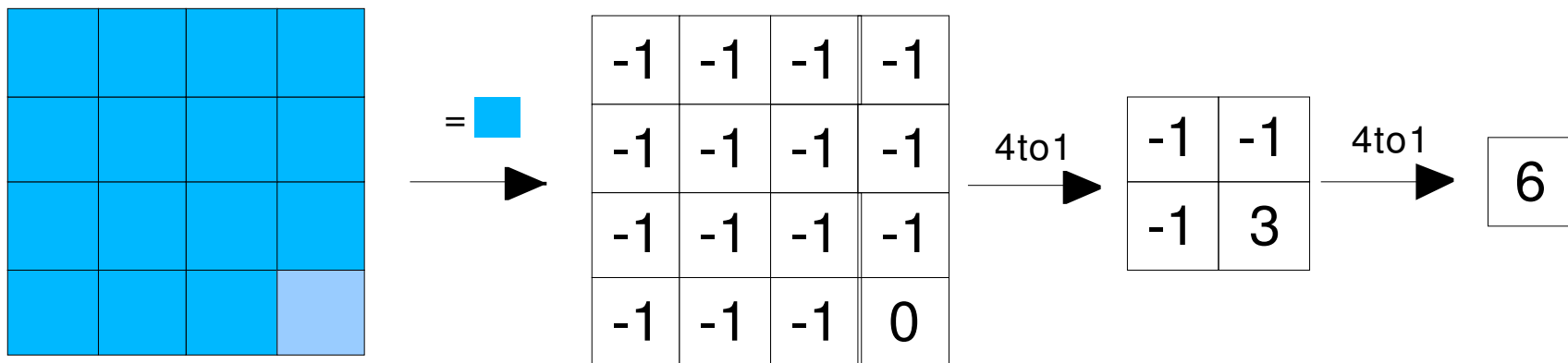
# Real-time QuadTree analysis

- Vision task: Detect regions with common features
  - “Common feature”: Low variance around average color
  - Usually very time-consuming, by far not real-time
  - Our Answer:
    - Extension of GPU point list algorithm
    - Only create simple region geometry, region quadtrees
    - Typical analysis time: 640x480 in 15 ms
  - Useful for:
    - Color grouping
    - Motion vector clustering
    - Acceleration of grid calculations
-

# Real-time QuadTree Analysis:

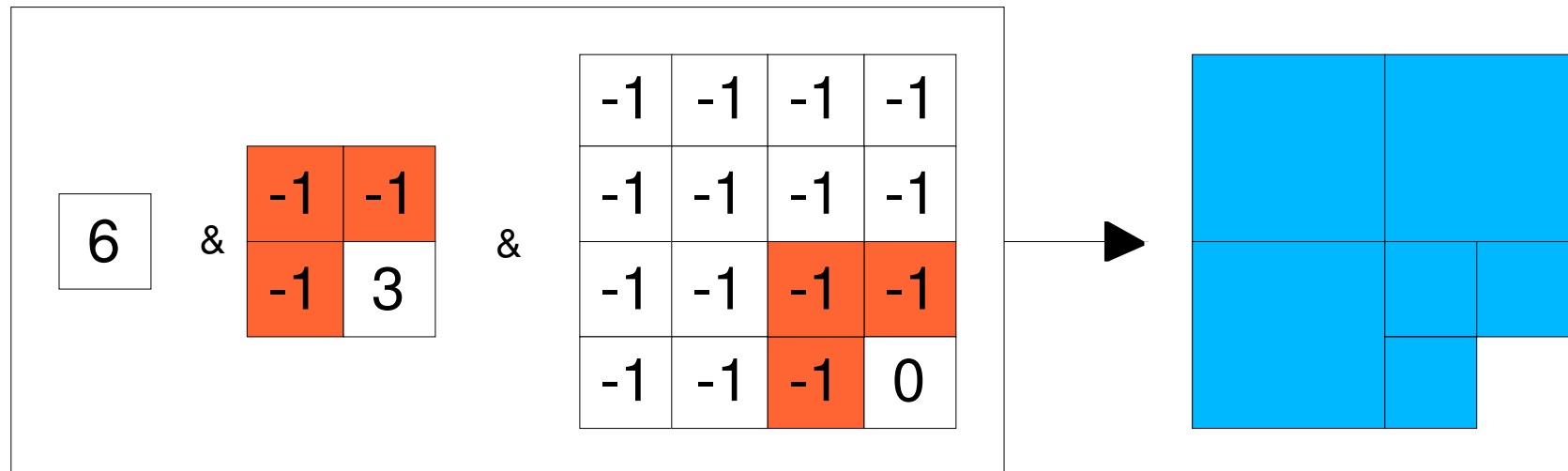
## Algorithm overview

- Extends GPU PointList algorithm with a new concept: QuadTree leaf (marked with -1)
- HistoPyramid base is filled with QuadTree leaves
- New rules during cell reduction (in HistoPyramid Builder):  
*If all four children are QuadTree leaves  
then create a new QuadTree leaf cell,  
else store absolute sum of children cells.*



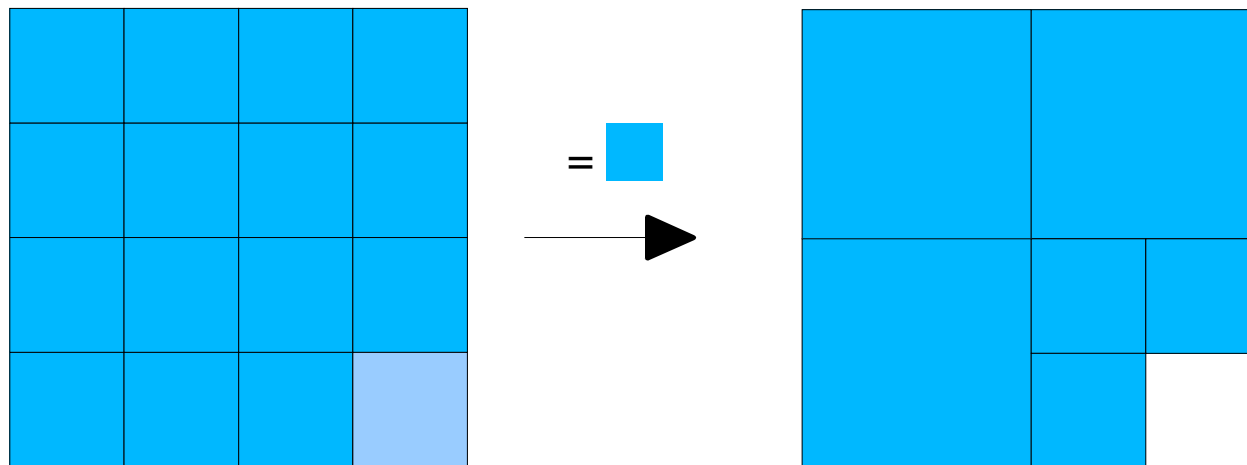
# Real-time QuadTree Analysis: Algorithm overview

- Item reconstruction similar to PointList reconstruction.
- But extend GPU PointList Builder rules:
- If* a quadtree leaf is encountered and its cell index is correct, *then* terminate traversal, output quad of corresponding size.  
(In example, red: Traversal terminators)  
*else* descend into children cells.



# Real-time QuadTree Analysis: Algorithm variants

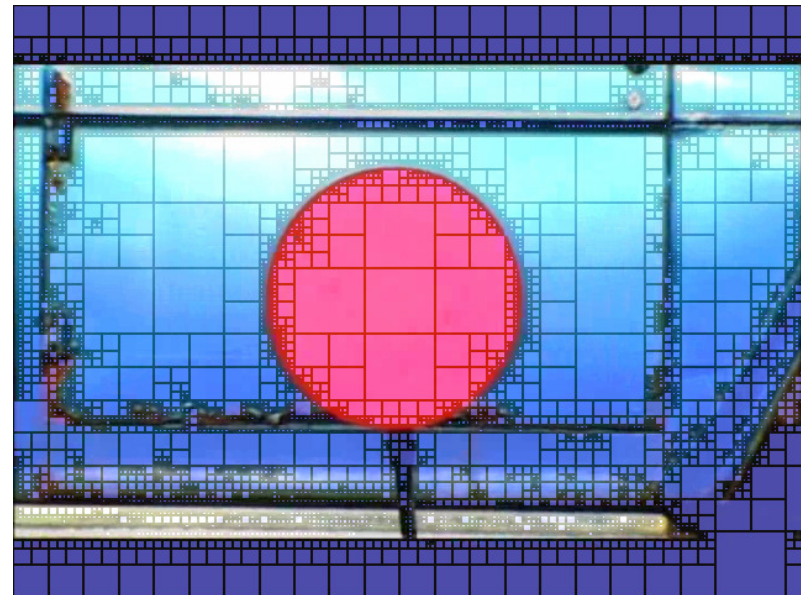
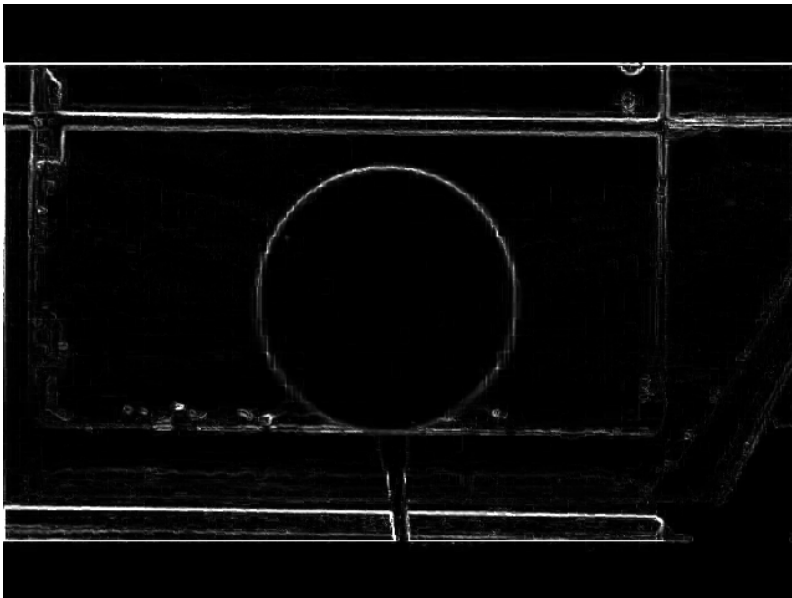
- Simple: isolate one feature
- Useful for:
  - Sparse matrix analysis (Occupation)
  - Data Compression (efficient representation)
  - Silhouette quadtree/octree



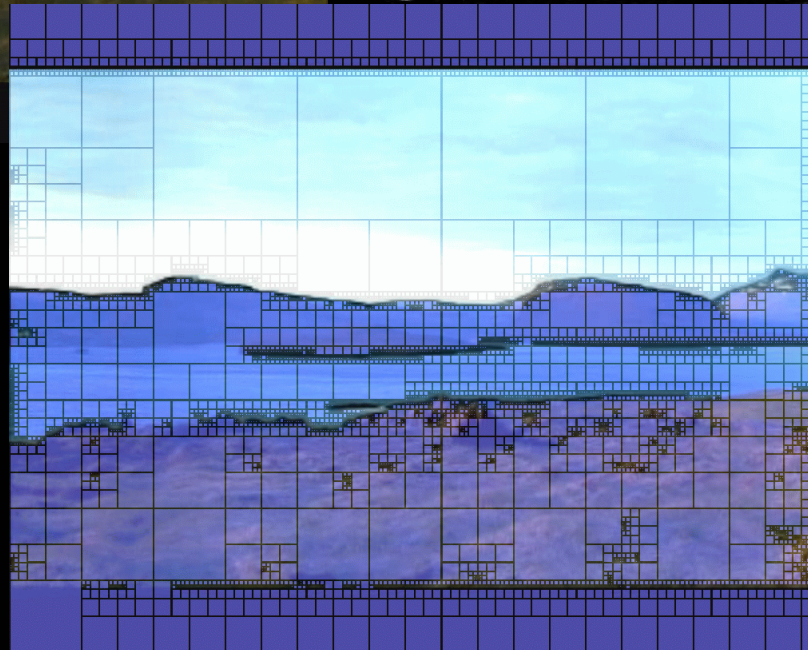
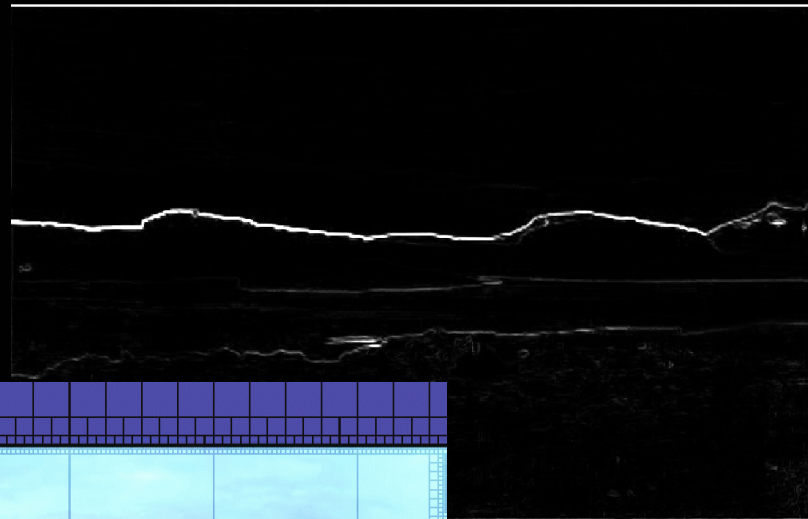


# Real-time QuadTree Analysis: Edge thresholding variant

- Run edge-detection on input, fusion regions with low activity
- Good: intuitive behaviour, low storage
- Bad: pre-processing;  
edges consume one pixel, thus suboptimal tessellation



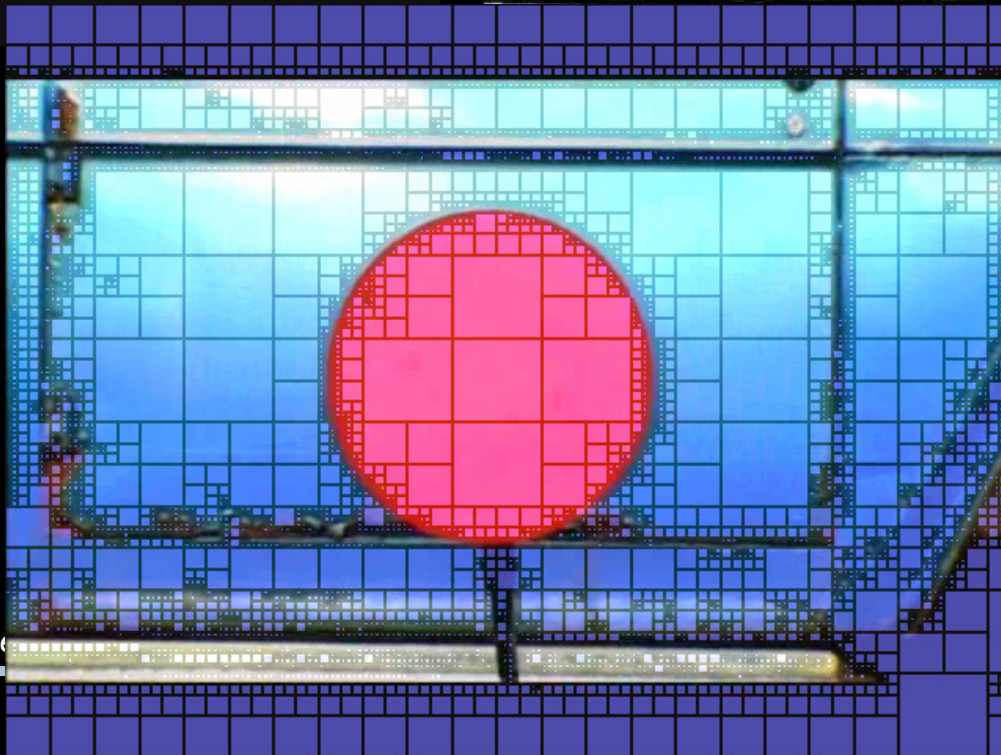
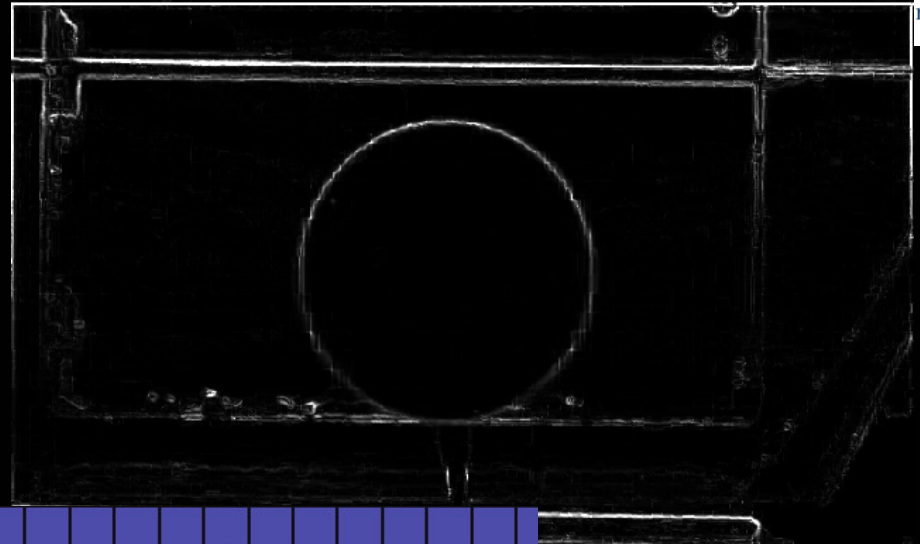
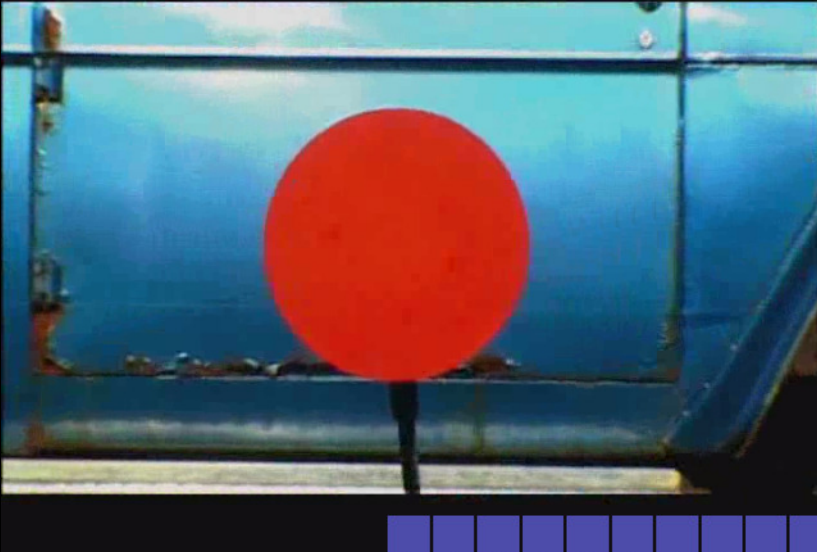
# Results



*ffkvadrat: Real-time quadtree analysis*

Video frame, Edge analysis, Quadtree Overlay

(Footage by GusGus, *Desire*)



*ffkvadrat: Real-time quadtree*

Edge analysis, Quadtree Overlay

# Results



Original video



Num. Quads: 1089  
Threshold: 0.20  
Video + AVG Color Outline

Analysis result  
(at different thresholds)

*Prototype implementation by Rouslan Dimitrov*

# Results



*ffkvadrat: Real-time quadtree analysis*

Video frame, Edge analysis, several results

(Footage by GusGus, *Call of The Wild*)