Computer
Graphics
Charles
University

Raster Image Encoding

© 1995-2015 Josef Pelikán & Alexander Wilkie
CGG MFF UK Praha

pepca@cgg.mff.cuni.cz
http://cgg.mff.cuni.cz/~pepca/

Use

- **Efficient storage** of images
  - Two-dimensional nature of the data can be exploited for better compression

- Effective operations on images with **bitmasks**
  - Set operations with bitmasks
  - Superposition of images

RLE („Run-Length Encoding”)

- Use of **coherence** in horizontal direction
  - – Neighbouring pixels often have the same value
  - – Most efficient at low bit depths

- Special character to start a **„run”**

  ESC {#} {pixel} (PCX)

- Two types of run - **„copy”** and **„iterate”**

  COPY {#} {data ...} (Targa, BMP, ...)

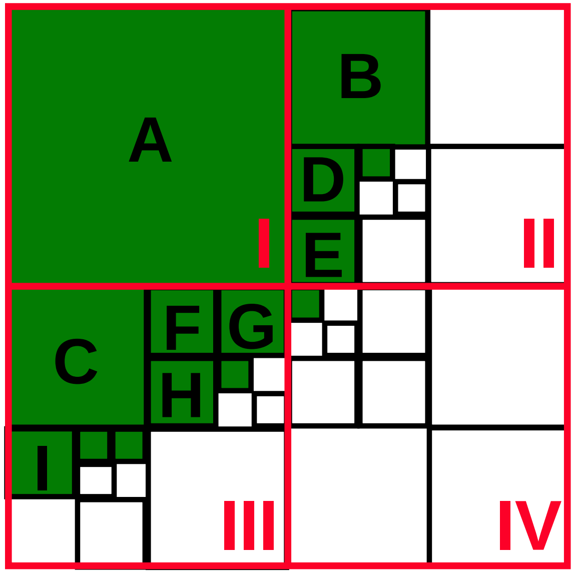
  FILL {#} {pixel}

Quadrant Tree („quadtree”)

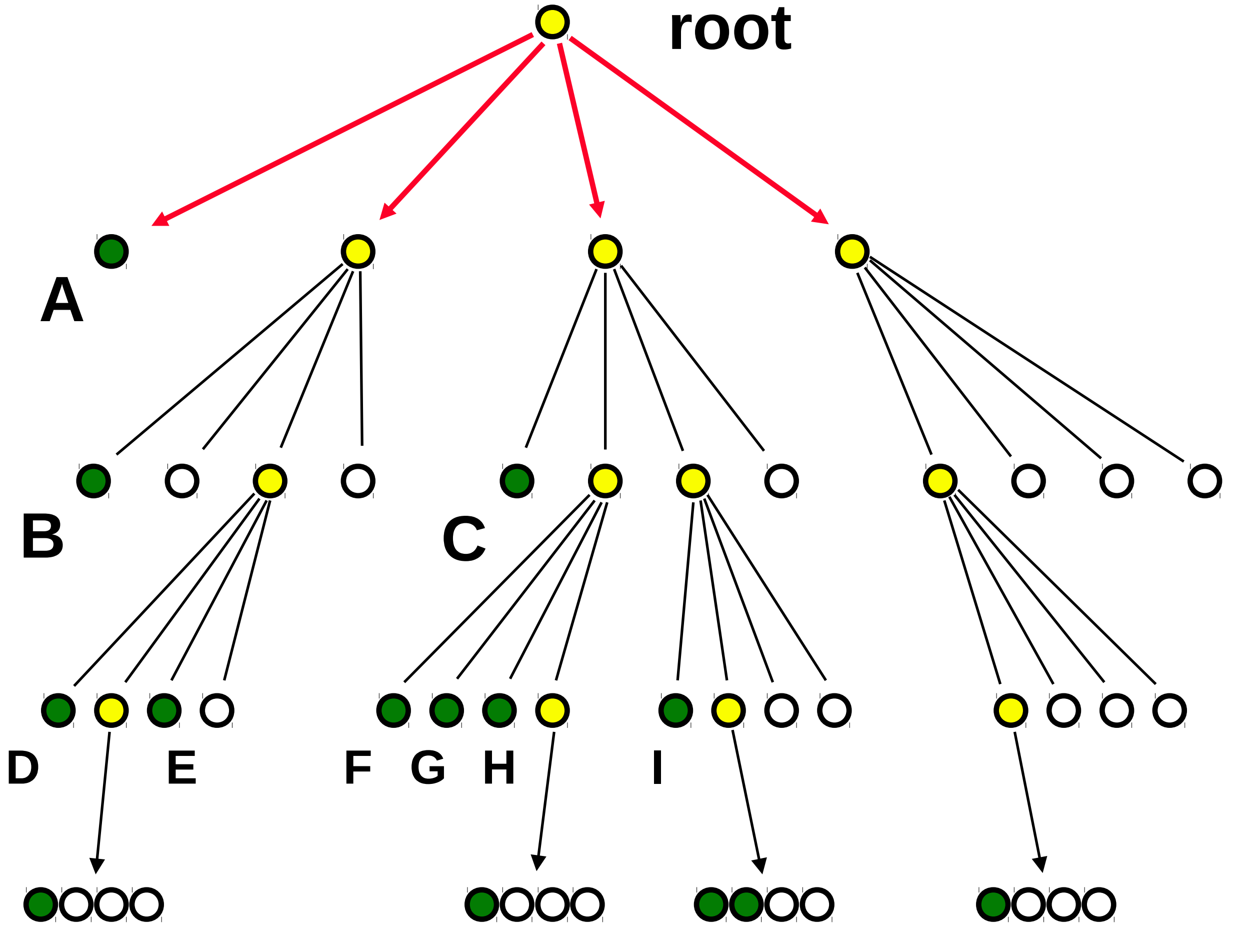
- Exploits coherence in scanline and vertical direction
  - Stores large areas of similar colour
  - **Adaptive** (gradual sub-division of „interesting“ areas)

- **Applications for quadtrees:**
  - Image storage
  - Space-saving storage of **bitmasks** (set operations)
  - Auxiliary data structure for **fast searching**

Quadrant Tree („quadtree”)



16 × 16
(256 bytes)



12 entries (96 bytes)

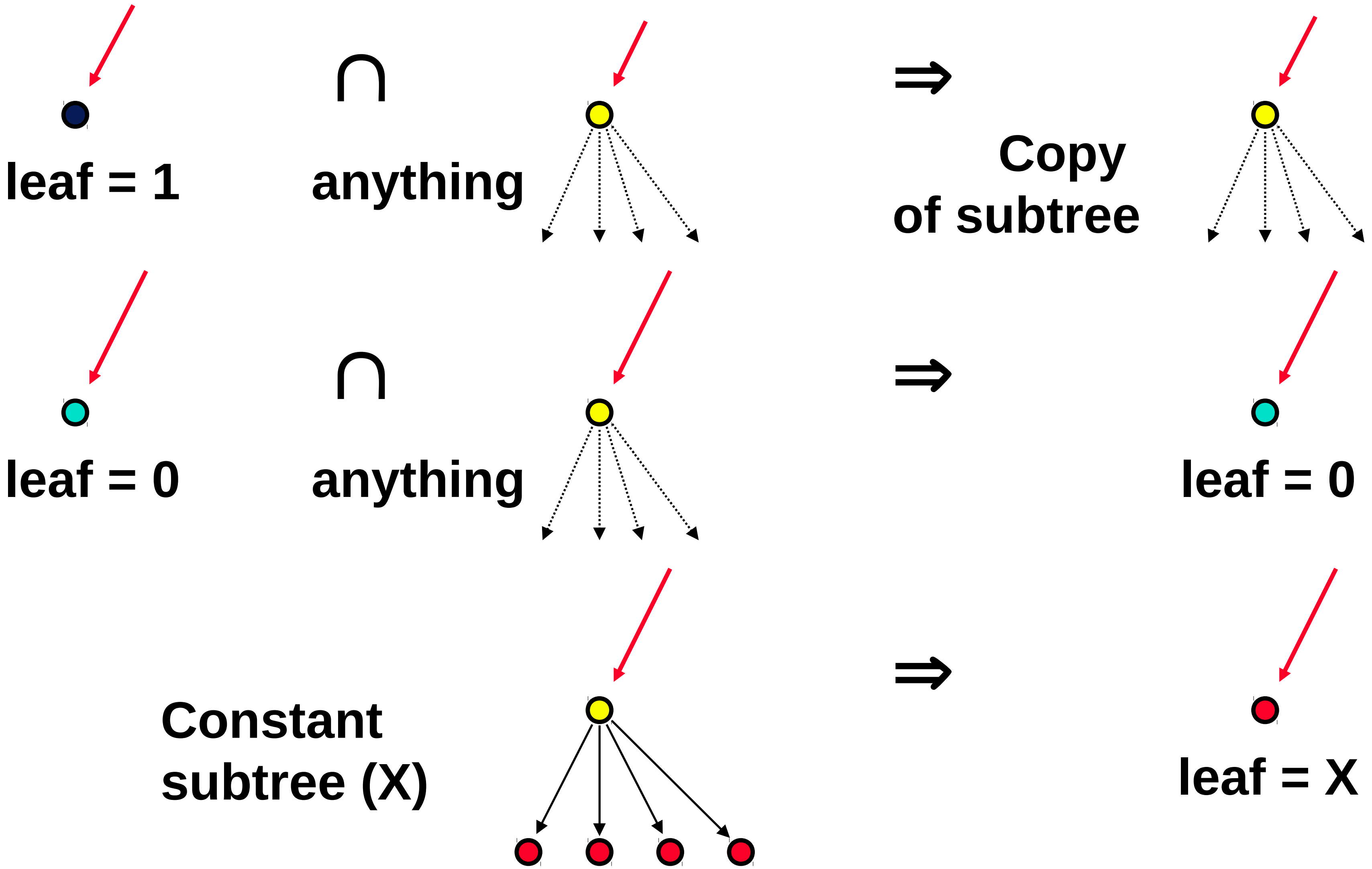
Coding of Quadtrees

- Top-down
  - Check area of potential quadtree node: if it is not uniform, subdivide
  - Each pixel gets queried multiple times
- Bottom-up
  - Start with 2x2 pixel blocks, test if they are uniform
  - On the way up, combine uniform areas
  - Every pixel is only read once

Set theory operations

- Quadtrees represent one bit informations (set, mask,...)
  - Set operations (union, intersection, ...)
  - Requires similar definition area
- Parallel walk of the input tree, and construction of the output tree
  - All input nodes are disjoint: divide and conquer

Rules for pruning operations



leaf = 1 ∩ anything ⇒ Copy of subtree

leaf = 0 ∩ anything ⇒ leaf = 0

Constant subtree (X) ⇒ leaf = X

Implementation notes

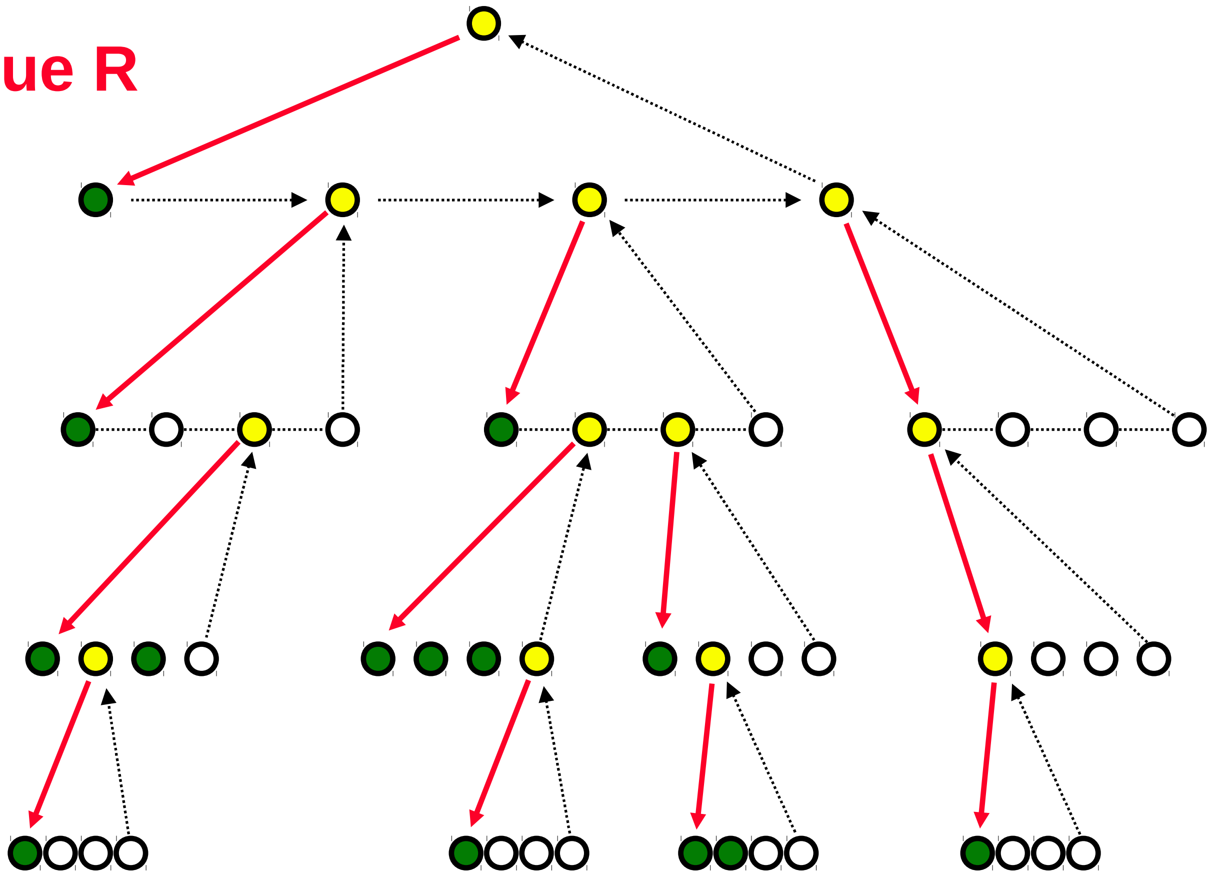
- Coding of common areas:
  - Use a smallest size of $\mathbf{2^n \times 2^n}$ blocks as “terminal symbols” of the tree
  - Pixels outside the area get a special code
- Efficient hybrid coding:
  - If a sub-tree is larger than the corresponding bitmap, use the bitmap instead

Implementation notes

- Shared branches of a quadtree:
  - If a subtree occurs several times, store it only once, and refer to it from different locations (postprocess)
  - Turns the tree into an acyclic directed graph
  - Banch joins can be used at various levels
- Linearisation of a quadtree
  - Traversal in unique top-down, left-to-right order („pre-order”)

Linear Storage of a Quadtree

Special value R



R1R10R1R1000100R1R111R1000R1R1100000

RRR1000000000 ... 49 entries

End

Further information:

- **J. Foley, A. van Dam, S. Feiner, J. Hughes**: ***Computer Graphics, Principles and Practice***, 844-846, 552-555, 992-996