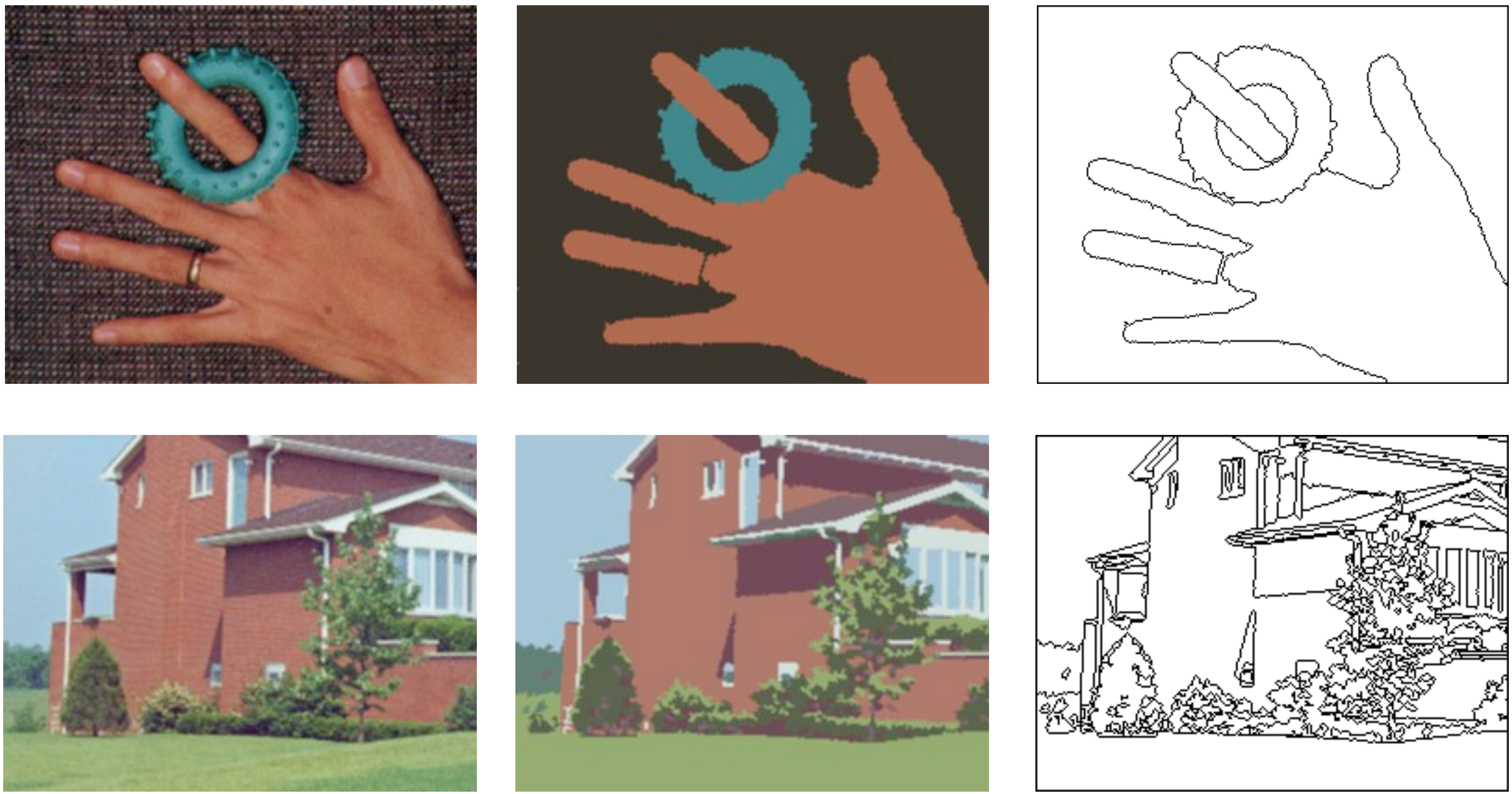
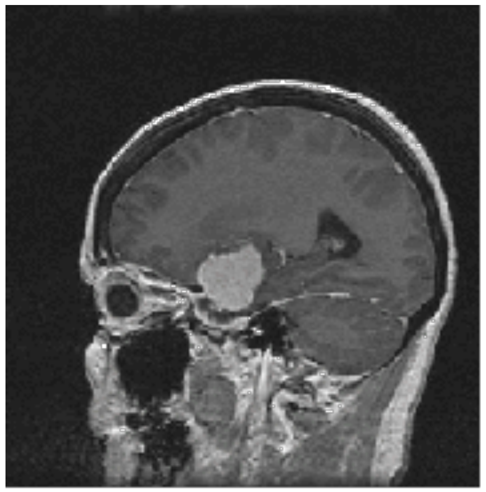
# Image Segmentation

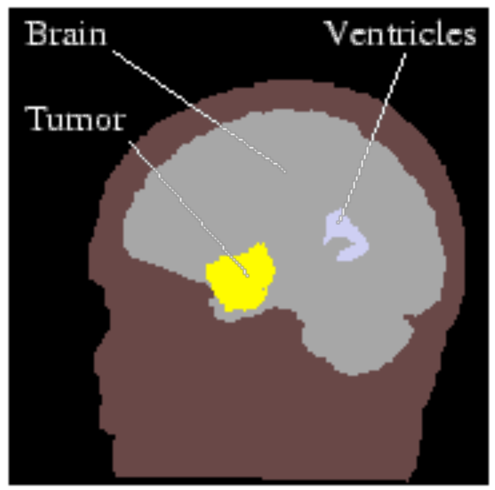
# Natural Images



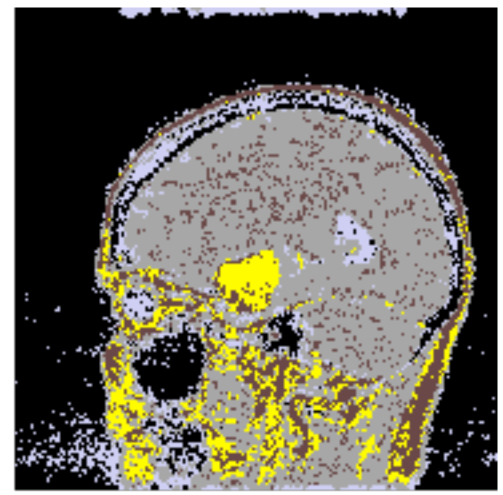
# MRI images



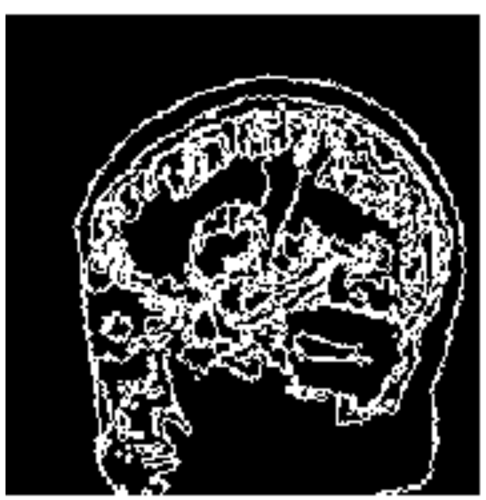
(a) Saggittal cross-section of MRI volume (Meningioma)




(b) Manual segmentation

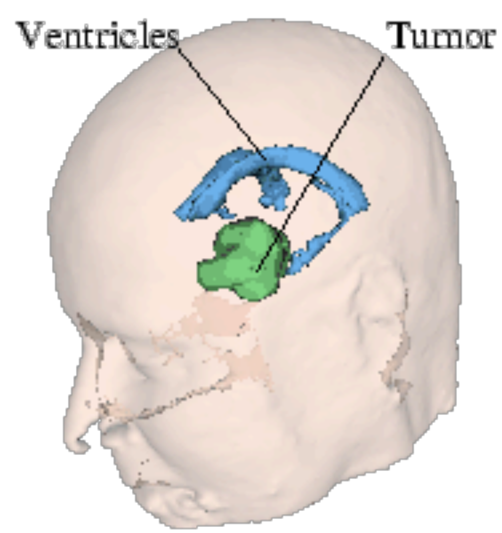


(c) Statistical classification



(d) Outline of all structures in the registered anatomical atlas

(e) Combination of statistical classification and model-based methods



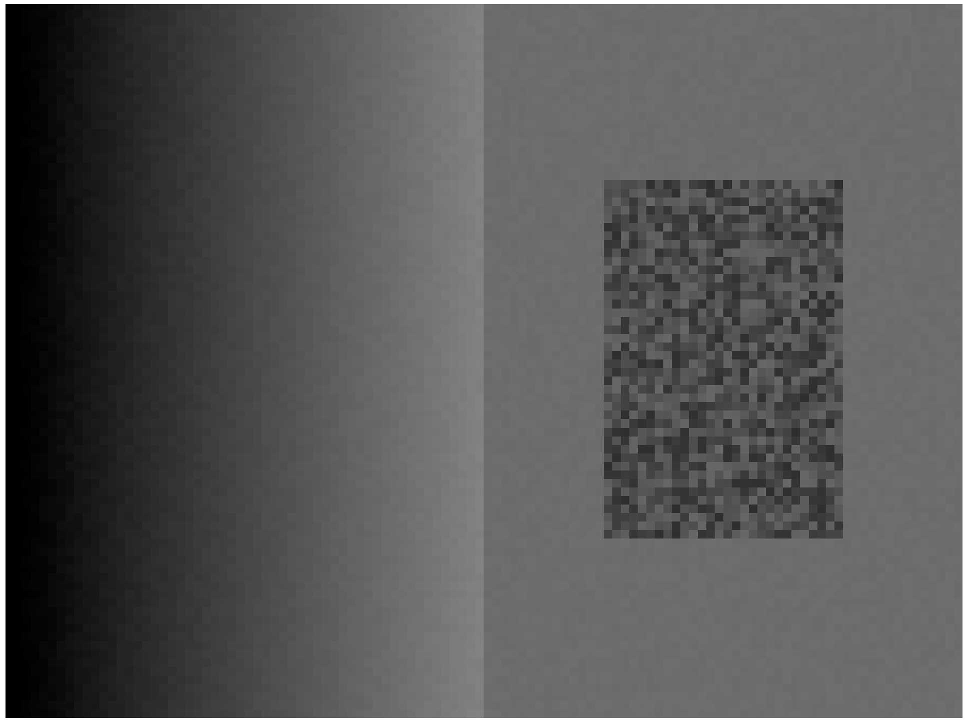

(f) 3D model

# Classical methods

- Agglomerative approach
  - start with one region per pixel
  - repeatedly merge regions with similar properties
- Merge regions with similar average color: get regions with near constant color
- local criteria: group neighboring pixels that have similar color (and use transitivity)

# Motivating example

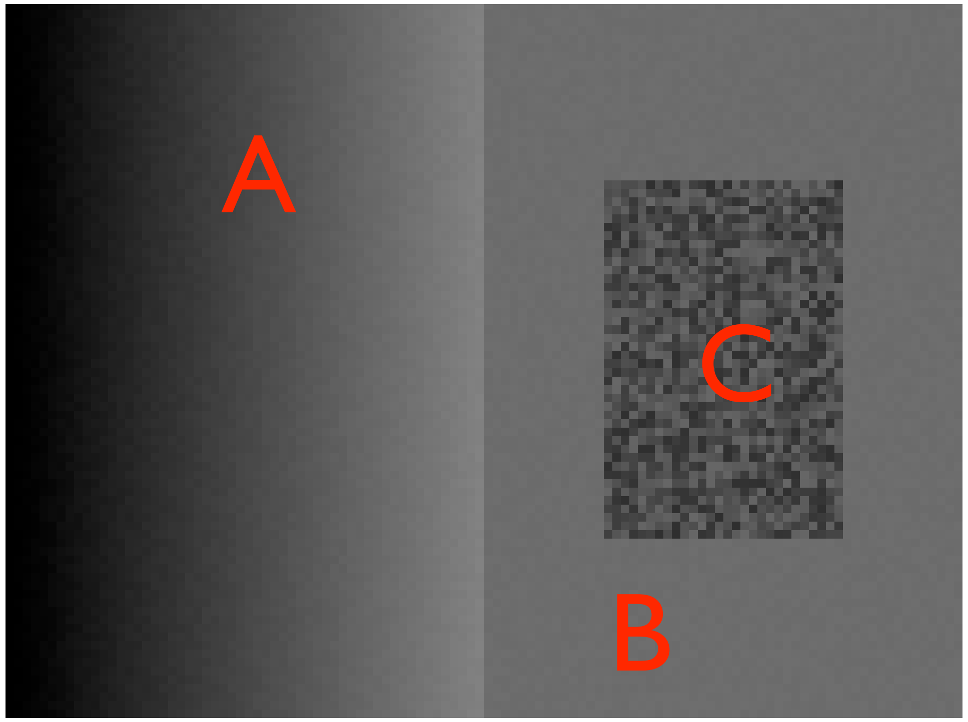
- Three coherent regions (independent of high-level knowledge)



- where are large intensity differences?

# Motivating example

- Near constant assumption inadequate
  - would split A




- Purely local criteria inadequate
  - difference along border of A and B less than differences within C

# Adaptive criteria

- Graph based approach G=(V,E)
- V are pixels, E connect nearby pixels
- weights on edges
  - w(u,v) = difference(color(u), color(v))
- Compare differences along boundary of regions to differences within regions

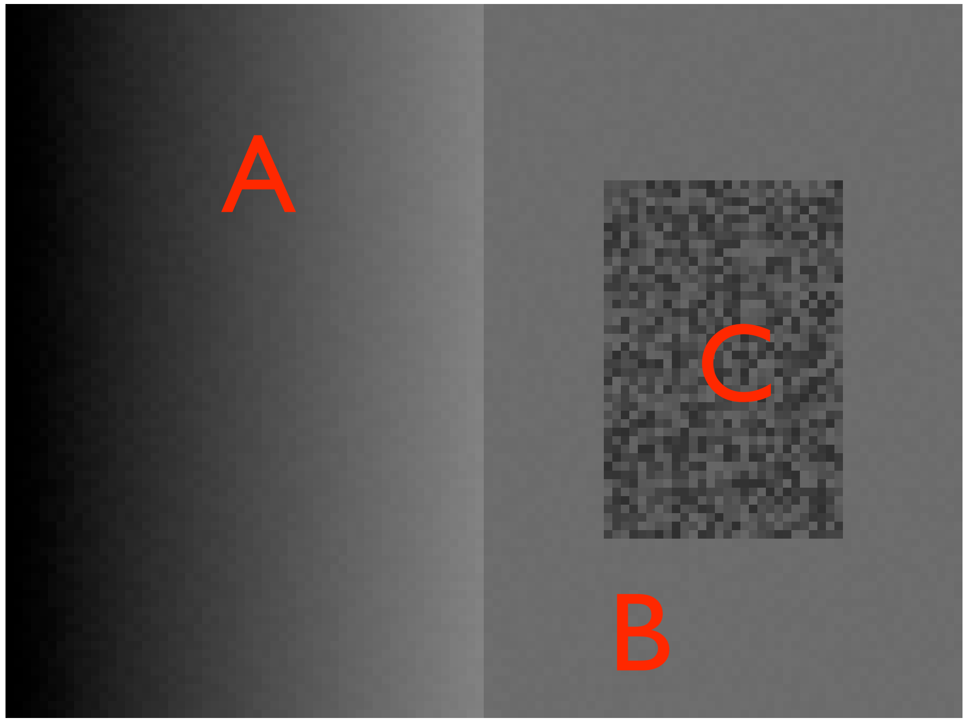
# Criteria

- dif(A,B): difference along boundary of regions
  - dif(A,B) = min weight edge between A and B
- int(A): internal differences within region
  - int(A) = max weight in MST of A
  - small even if A is far from constant
- Evidence for boundary between A and B if
  - dif(A,B) > min(int(A)+T, int(B)+T)

# Good segmentation

- Too fine if there are regions A and B with no evidence for boundary between them
- Too coarse if there exists a refinement that is not too fine
- Good if neither too fine nor too coarse

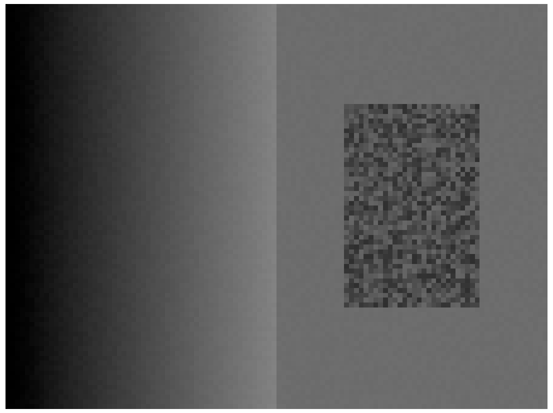
# Motivating example



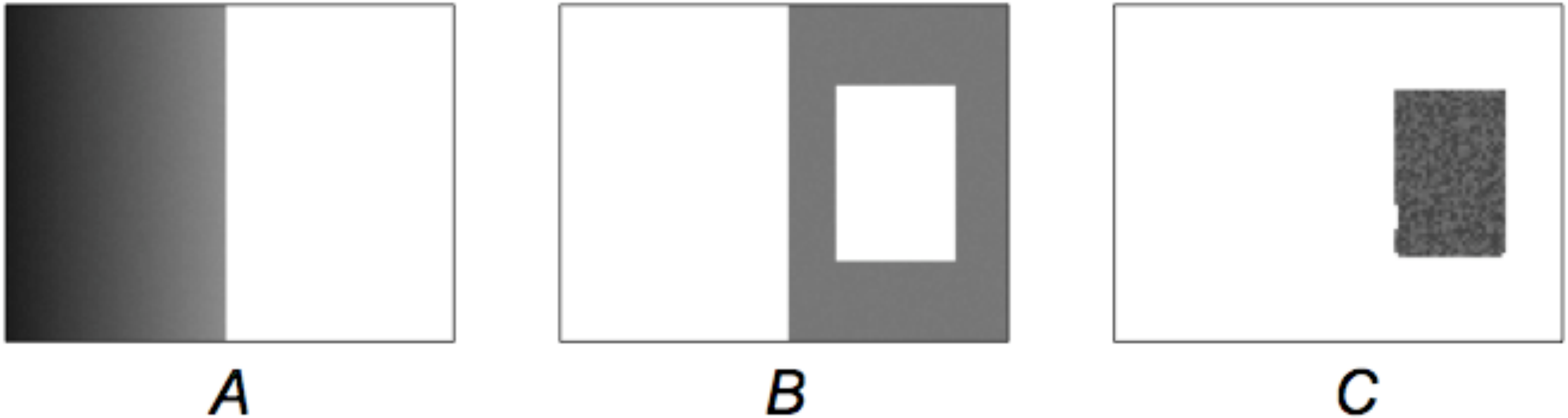

- For any subset of C, int(C') large
- Splitting C would be too fine
- int(A), int(B) small
- dif(A,B) larger than int(A) and int(B)
- Merging A and B would be too coarse

# Efficient algorithm

- Initialize S with with component per pixel
- int(A) = 0 for all components
- Consider edges (u,v) in increasing weight order
  - let U and V be components containing u and v
  - if U=V, skip edge
  - if w(u,v) < min(int(U)+T, int(V)+T)
    - merge U and V into W
    - int(W) = w(u,v)

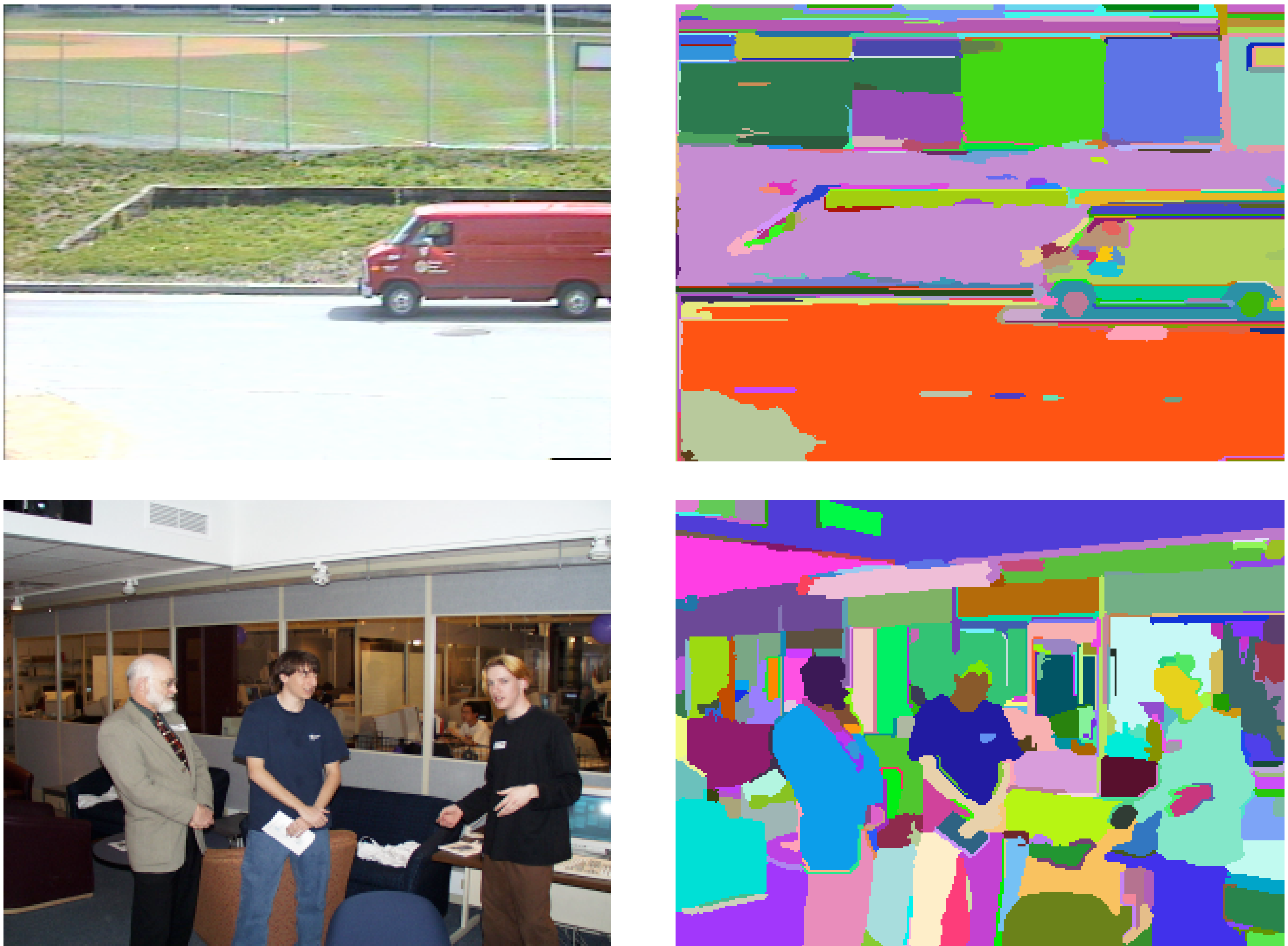


# Motivating example




- First B forms, then A, then C
- Weights between A and B are large wrt to int(A)
- Weights between B and C are large wrt to int(B)
- int(C') large for any subpart of C

# Example results



# More results

If we connect non-neighboring pixels by edges we can get regions that are non-local

# Grouping pixels by proximity and color is not enough