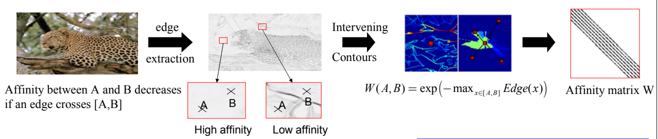
Spectral Segmentation with Multi-Scale Graph Decomposition

Timothee Cour Florence Benezit Jianbo Shi

University of Pennsylvania

- Affinity graph needs long-range connections to gather sufficient grouping information
- Long-range connections can be compressed on a multiscale grid
- With constrained segmentation, non-adaptive grid can produce precise object boundary

Introduction: Graph-based image segmentation



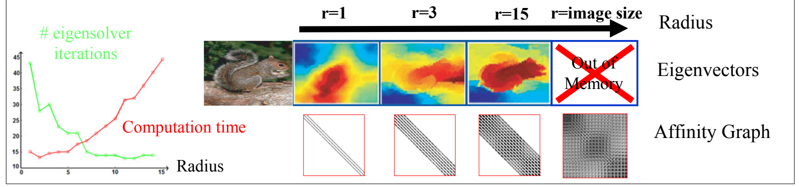
$W(A,B) = \exp(-\max_{x \in [A,B]} Edge(x))$

"similar pixels should be in the same segment, dissimilar pixels should be in different segments" => **normalized cuts:** $Ncut(U,V) = \frac{cut(U,V)}{Vol(U)} + \frac{cut(U,V)}{Vol(V)}$

We introduce indicator vector X: $X^k(i) = 1$ iff $i \in region(k)$

Goal: maximize $\varepsilon(X) = \frac{1}{K}\sum_k \frac{X^{kT} W X^k}{X^{kT} D X^k}$ => relaxed problem solved with eigenvectors $WX = \lambda DX$

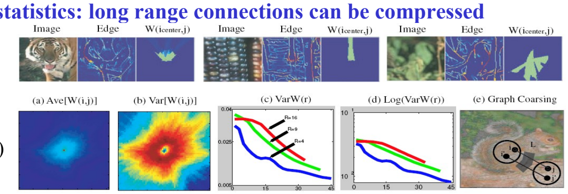
long-range connections: speed / accuracy tradeoff



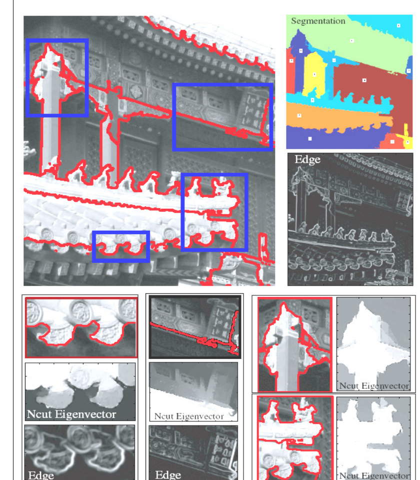
Ecological statistics: long range connections can be compressed

Affinity Graphs W on natural images

Statistics on $W(i_{center}, :)$

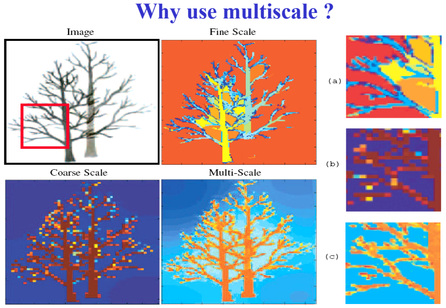


Results



Multiscale segmentation of a 800x700 image.
Notice the faint roof boundary is segmented clearly

Why use multiscale ?



fine scale: foreground is split.
coarse scale: details are lost.
Multiscale: local accuracy and global consistency.

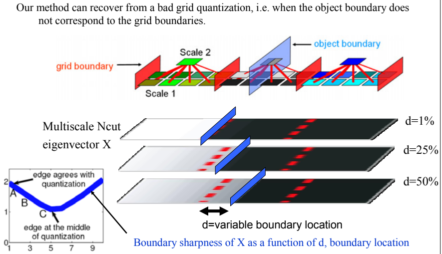
Previous work on multiscale segmentation

1) multi-scale signal processing: edge cues at different image scales
2) image-region similarity cue for texture/shape
3) data-driven adaptive coarsening: multiscale mesh refined iteratively (works well, but difficult to tweak and no clear objective function)

non-adaptive grid usually suffers from grid artifacts

Non-adaptive grid does not suffer from grid artifacts

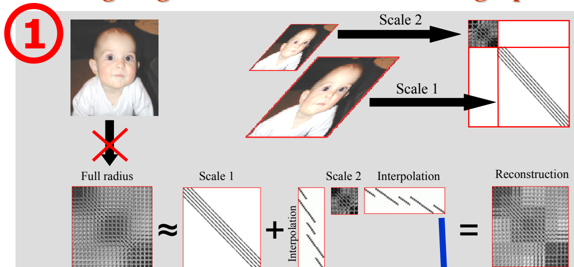
Our method can recover from a bad grid quantization, i.e. when the object boundary does not correspond to the grid boundaries.



Boundary sharpness of X as a function of d, boundary location

Image segmentation with multiscale graph compression and cross-scale constraints

1



$W_{Full} \approx W_{scale\,1} + C_{1,2}{}^T W_{scale\,2} C_{1,2}$

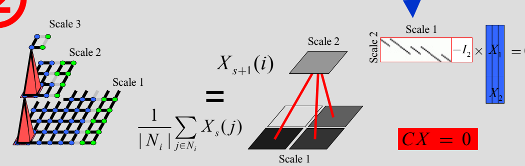
Graph compression into multiple scales

Given an image I of size $p \times q$: for $s = 1:S$ ($S = \#scales$):

a) sample $(p/d) \times (q/d)$ pixels $i \in I^s$ from I^{s-1} on a regular grid
b) compute affinity W^s on I^s with radius r

Multiscale affinity matrix: $W = \begin{pmatrix} W_1 & & 0 \\ & \ddots & \\ 0 & & W_S \end{pmatrix}$

2



$X_{s+1}(i) = \frac{1}{|N_i|}\sum_{j \in N_i} X_s(j)$

$[\,\cdots\; -I_2\,] \times \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = 0$

$CX = 0$

Cross-scale constraint

$X = \begin{pmatrix} X_1 \\ \vdots \\ X_S \end{pmatrix}$, each X_s is a $n_s \times K$ indicator of the K partitions at scale s

Super-node label = average of labels of nodes in that super-node.

$C_{s,s+1}(i,j) = \begin{cases} 1/|N_i|, & \text{if } j \in N_i \\ 0, & \text{else} \end{cases}$

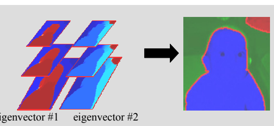
Constraint matrix: $C = \begin{pmatrix} C_{1,2} & -I_2 & & 0 \\ & \ddots & \ddots & \\ 0 & & C_{S-1,S} & -I_S \end{pmatrix}$

Upper triangular !

3

maximize $\varepsilon(X) = \frac{1}{K}\sum_k \frac{X^{kT} W X^k}{X^{kT} D X^k}$

subject to: $CX = 0$



$$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix}^T \begin{pmatrix} W_1 & 0 \\ 0 & W_2 \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} = X_1{}^T W_1 X_1 + (C_{1,2} X_1)^T W_2 (C_{1,2} X_1) = X_1{}^T \underbrace{(W_1 + C_{1,2}{}^T W_2 C_{1,2})}_{\approx W_{Full}} X_1$$

$$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix}^T \begin{pmatrix} D_1 & 0 \\ 0 & D_2 \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} = X_1{}^T \underbrace{(D_1 + C_{1,2}{}^T D_2 C_{1,2})}_{} X_1$$

Denominator is full matrix !!
Hard to invert

We need a more clever solution

Constrained Ncut optimization

Using Lagrange multipliers, the constrained Ncut optimization amounts to finding eigenvectors of $\overline{W}X = \lambda \overline{X}$

$\overline{W} = Q D^{-1/2} W D^{-1/2} Q$,

with $Q = I - D^{-1/2} C^T (C D^{-1} C^T)^{-1} C D^{-1/2}$

subspace projection matrix

compute $\overline{X}, K$ first eigenvectors of $\overline{W}$
by iterated chained matrix-vector multiplications.

discretize $X = D^{-1/2}\overline{X}$

Running time: O(N), N=#pixels

Accurately segments large images that previously could not be handled: < 400 seconds for 1000x1000 image

Proof of running time:

$nnz(W) = \sum_s N \cdot (2r+1)^2 / d^{2s} = O(N)$

computing $(CD^{-1}C^T)^{-1}X$ in $\overline{W}$ is usually $O(N^3)$,
but C is upper triangular => solve 2 triangular systems with $O(N)$ elements