Segmentation Using Superpixels: A Bipartite Graph Partitioning Approach

Zhenguo Li   Xiao-Ming Wu   Shih-Fu Chang
Columbia University

DVMM lab — digital video | multimedia laboratory

COLUMBIA UNIVERSITY

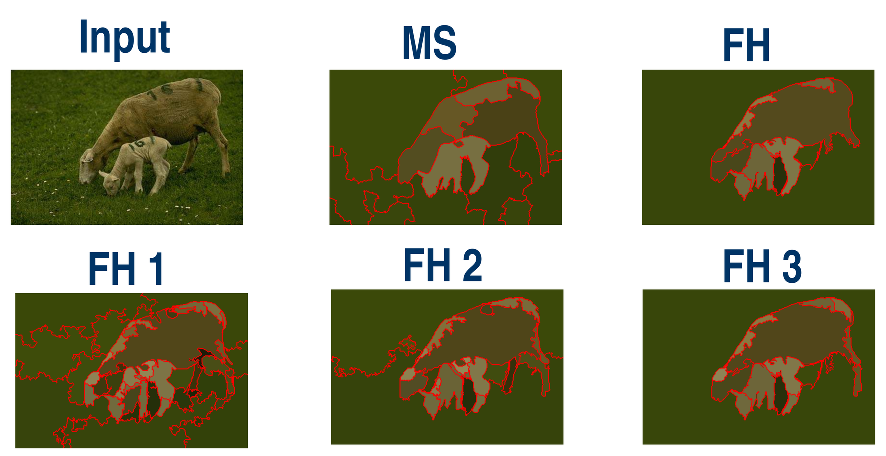
Problem and Motivations

Segmentation is crucial for high-level vision. It remains challenging due to visual ambiguity and variety.

◆Observations

◆Different methods behave differently.

◆Each method gives different results under different parameters.



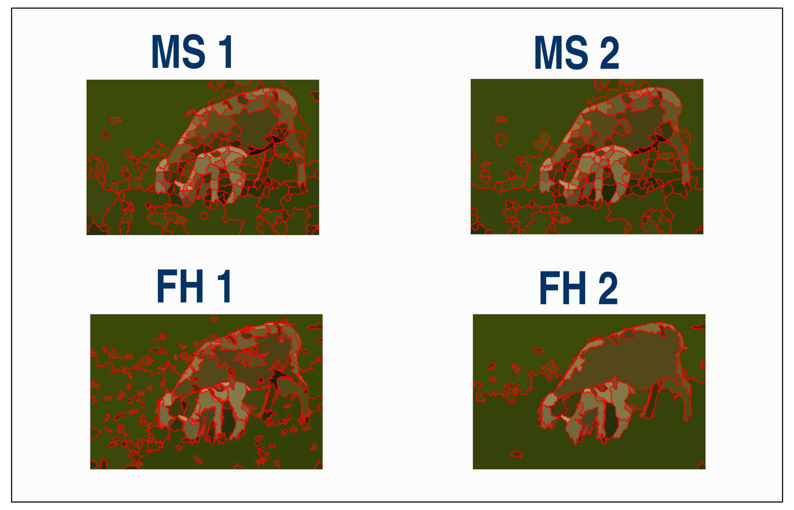
MS [Comaniciu and Meer'02]
FH [Felzenszwalb and Huttenlocher'04]

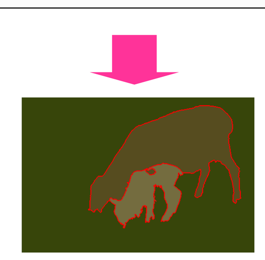
How to capture and model a variety of visual patterns simultaneously?

◆Motivations

◆Combine complementary information to improve performance.

◆Capture visual patterns using superpixels generated by different methods with varying parameters.





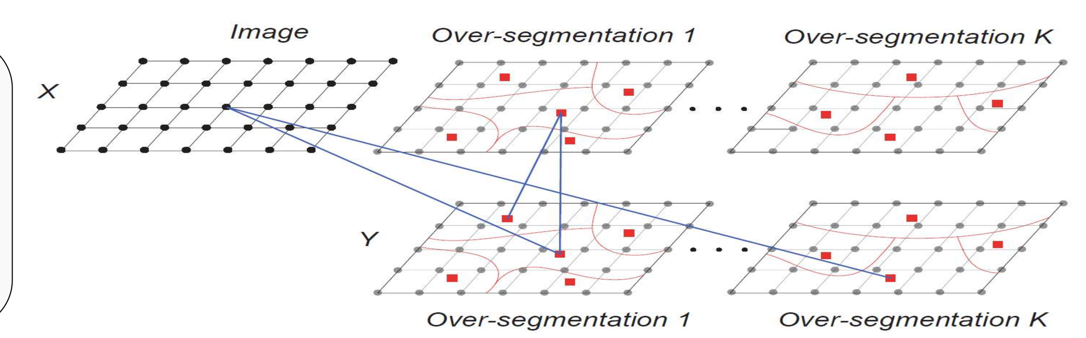
Ours (SAS)

Superpixel Aggregation and Bipartite Graph Partitioning

◆Combine pixels and multiple/multi-scale segmentations by a bipartite structure. Using superpixels as grouping cues:

◆Pixels in a superpixel tend to belong together.

◆Similar neighboring superpixels tend to belong together.



Algorithm 1 Transfer Cuts

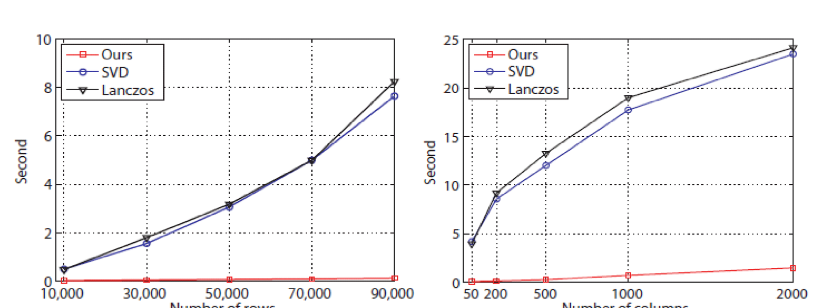
Input: A bipartite graph $G = \{X, Y, B\}$ and a number k.
Output: A k-way partition of G.

1: Form $D_X = \mathrm{diag}(B\mathbf{1})$, $D_Y = \mathrm{diag}(B^\top \mathbf{1})$, $W_Y = B^\top D_X^{-1} B$, and $L_Y = D_Y - W_Y$.
2: Compute the bottom k eigenpairs $\{(\lambda_i, \mathbf{v}_i)\}_{i=1}^k$ of $L_Y \mathbf{v} = \lambda D_Y \mathbf{v}$.
3: Obtain γ_i such that $0 \le \gamma_i < 1$ and $\gamma_i(2-\gamma_i) = \lambda_i$, $i = 1, \ldots, k$.
4: Compute $\mathbf{f}_i = (\mathbf{u}_i^\top, \mathbf{v}_i^\top)^\top$, with $\mathbf{u}_i = \frac{1}{1-\gamma_i} D_X^{-1} B \mathbf{v}_i$, $i = 1, \ldots, k$.
5: Derive k groups of $X \cup Y$ from $\mathbf{f}_1, \ldots, \mathbf{f}_k$.

Speedup by the bipartite structure

Table 1. Complexities of eigen-solvers on bipartite graphs

Algorithm	Complexity
Lanczos [26]	$O(k(N_X + N_Y)^{3/2})$
SVD [31]	$O(k(N_X + N_Y)^{3/2})$
Ours	$2k(1+d_X)N_X$ operations $+ O(kN_Y^{3/2})$

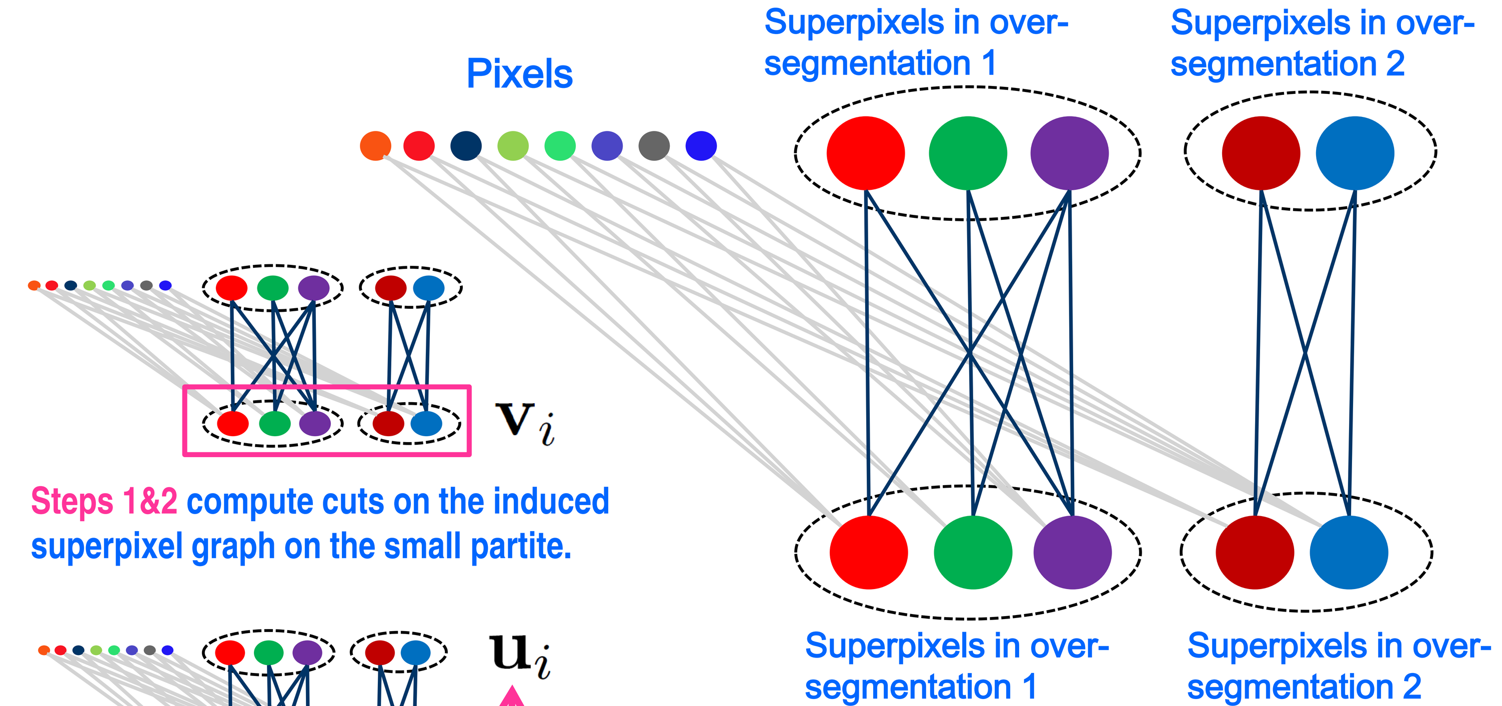


Algorithm 2 Segmentation by Aggregating Superpixels

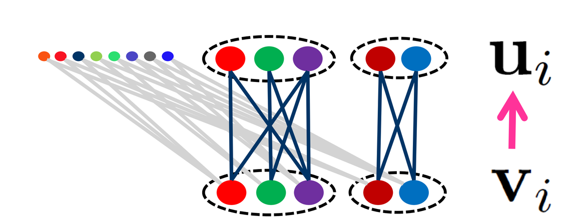
Input: An image I and the number of segments k.
Output: A k-way segmentation of I.

1: Collect a bag of superpixels $\mathcal{S}$ for I.
2: Construct a bipartite graph $G = \{X, Y, B\}$ with $X = I \cup \mathcal{S}$, $Y = \mathcal{S}$, and B defined in (1-3).
3: Apply Tcut in Algorithm 1 to derive k groups of G.
4: Treat pixels from the same group as a segment.

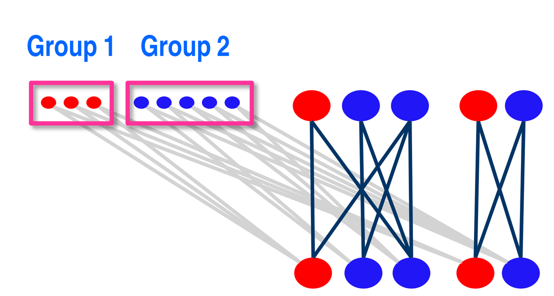
SAS takes 6.44s per image of size 481×321, where 4.11s for generating superpixels and 0.65s for Tcut. MNcut, MLSS, Ncut and TBES take more than 30s, 40s, 150s, and 500s, respectively. Codes of SAS are available at: www.ee.columbia.edu/dvmm.



Steps 1&2 compute cuts on the induced superpixel graph on the small partite.



Steps 3&4 transfer cuts from the small partite to the entire graph.



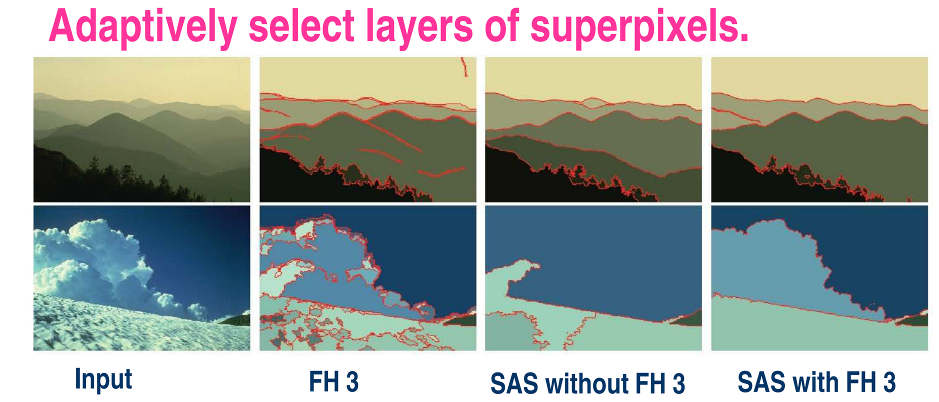
Step 5 groups the pixels and superpixels.

$$G = \{X, Y, B\} \quad B = (b_{ij})_{N_X \times N_Y}$$
$$X := I \cup \mathcal{S} = \{\mathbf{x}_i\}_{i=1}^{N_X}$$
$$Y := \mathcal{S} = \{\mathbf{y}_j\}_{j=1}^{N_Y}$$
$$b_{ij} = \alpha, \text{ if } \mathbf{x}_i \in \mathbf{y}_j, \mathbf{x}_i \in I, \mathbf{y}_j \in \mathcal{S};$$
$$b_{ij} = e^{-\beta d_{ij}}, \text{ if } \mathbf{x}_i \sim \mathbf{y}_j, \mathbf{x}_i \in \mathcal{S}, \mathbf{y}_j \in \mathcal{S};$$
$$b_{ij} = 0, \text{ otherwise},$$

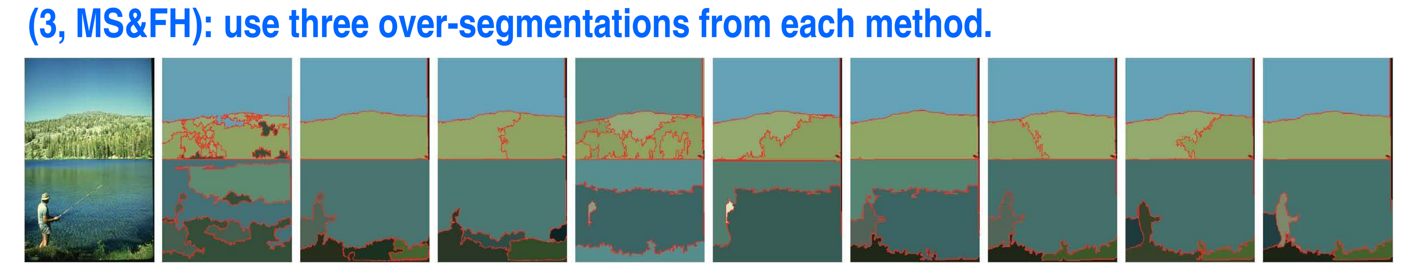
$\mathbf{x} \sim \mathbf{y}$ If x = y, or if y is adjacent to x and is its nearest neighbor in feature space, or vice versa.

Adaptively select layers of superpixels.

Include large superpixels for complex images to suppress strong edges within objects.



Segmentations with different combinations of layers of superpixels.
(3, MS&FH): use three over-segmentations from each method.



Segmentation Results

Results on Berkeley segmentation database (BSDS)

Methods	PRI	VoI	GCE	BDE
Ncut	0.7242	2.9061	0.2232	17.15
Mean Shift	0.7958	1.9725	0.1888	14.41
FH	0.7139	3.3949	**0.1746**	16.67
JSEG	0.7756	2.3217	0.1989	14.40
MNcut	0.7559	2.4701	0.1925	15.10
NTP	0.7521	2.4954	0.2373	16.30
SDTV	0.7758	1.8165	**0.1768**	16.24
TBES	0.80	1.76	N/A	N/A
UCM	0.81	**1.68**	N/A	N/A
MLSS	0.8146	1.8545	0.1809	**12.21**
SAS	**0.8319**	**1.6849**	**0.1779**	**11.29**
SAS(MS)	0.7991	1.9320	0.2222	15.37
SAS(FH1)	0.8070	1.8690	0.2167	14.28
SAS(FH2)	0.8007	1.7998	0.2105	17.17
SAS(MS+FH1)	**0.8266**	1.7396	0.1868	**11.83**
SAS(MS+FH2)	**0.8246**	**1.7144**	0.1904	12.63

Methods	FH	Ncut	Mean Shift	UCM	SAS
BFM	0.58	0.62	0.63	**0.71**	**0.64**
RSC	0.51	0.44	0.54	**0.58**	**0.62**

PRI: Probabilistic Rand Index; VOI: Variation of Information;
GCE: Global Consistency Error; BDE: Boundary Displacement Error;
BFM: Boundary-based F measure; RSC: Region-wise segmentation covering.

Sensitivity of SAS w.r.t. the parameters.

α	$\{10^{-9}, 10^{-5}, 10^{-1}, 10^{3}\}$				10^{-3}			
β	20				$2 \times \{10^{-5}, 10^{-1}, 10^{3}, 10^{7}\}$			
PRI	0.806	0.813	0.818	0.818	0.821	0.821	0.814	0.815
VoI	1.867	1.810	1.836	1.840	1.811	1.811	1.836	1.831
GCE	0.209	0.203	0.201	0.202	0.194	0.194	0.210	0.209
BDE	13.76	13.33	13.27	13.31	12.40	12.35	13.70	13.70

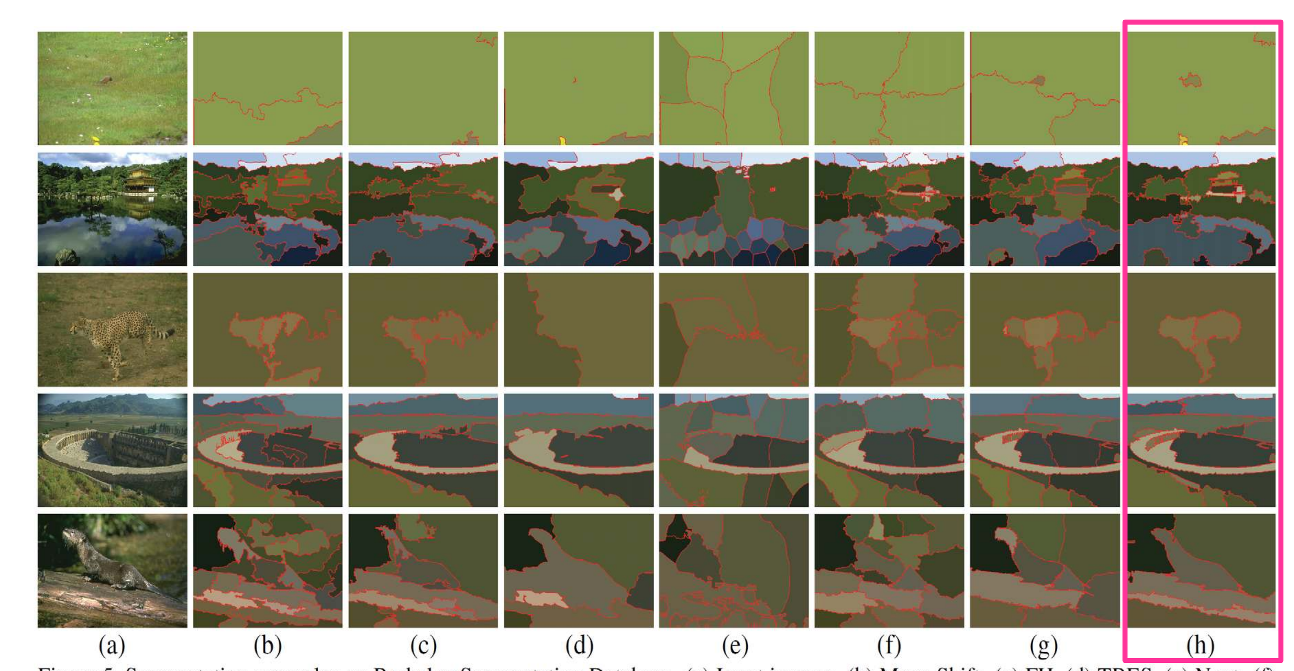


Figure 5. Segmentation examples on Berkeley Segmentation Database. (a) Input images. (b) Mean Shift. (c) FH. (d) TBES. (e) Ncut. (f) MNcut. (g) MLSS. (h) SAS.