UNIVERSITY of ROCHESTER

Segmentation

CSC 249/449 Spring 2019

http://www.cs.rochester.edu/~cxu22/t/249S19/

Instructor: Chenliang Xu
chenliang.xu@rochester.edu

Outline

- Graph-based segmentation:
  - Normalized Cuts
  - Minimum spanning forest

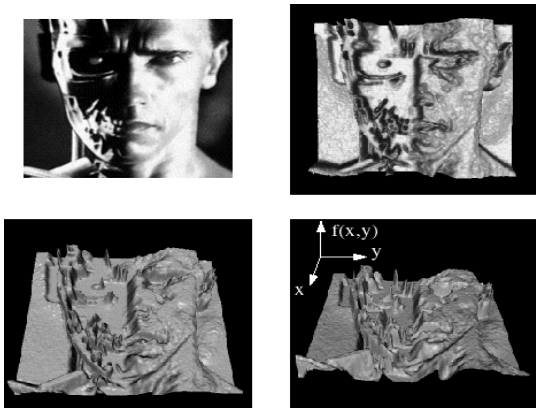
- Bottom-up video segmentation
  - LIBSVX: A Supervoxel Library and Benchmark for Early Video Processing
  - Streaming graph-based video segmentation

- Case Study: Combining bottom-up and top-down

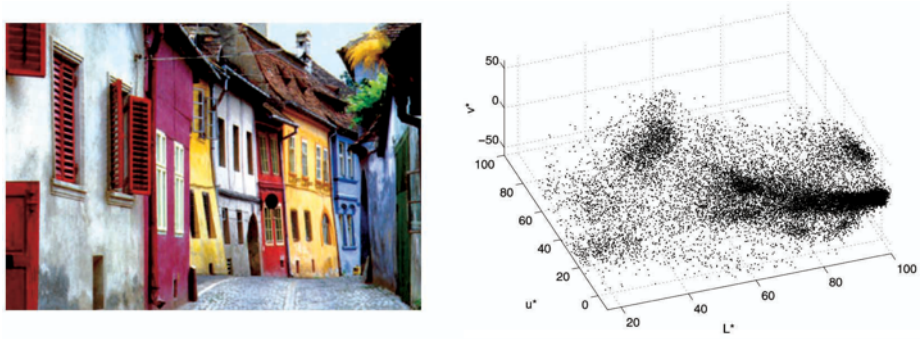
Graph-based Segmentation

Different ways of modeling images

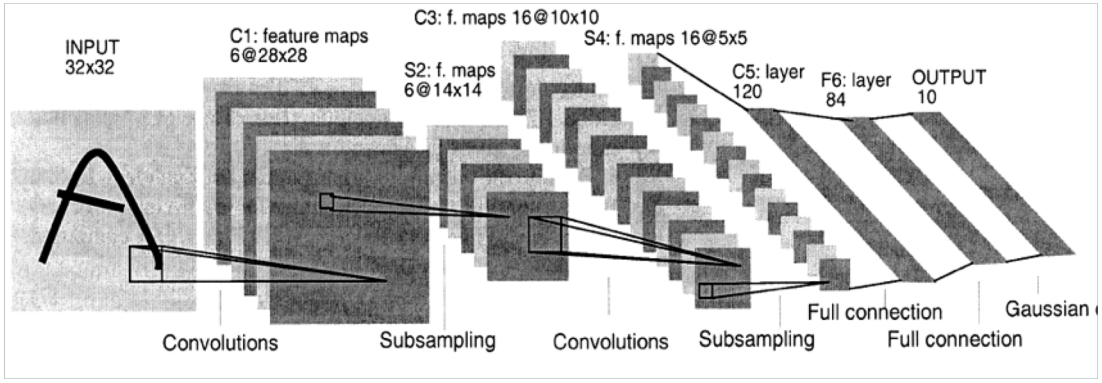
- Images as functions

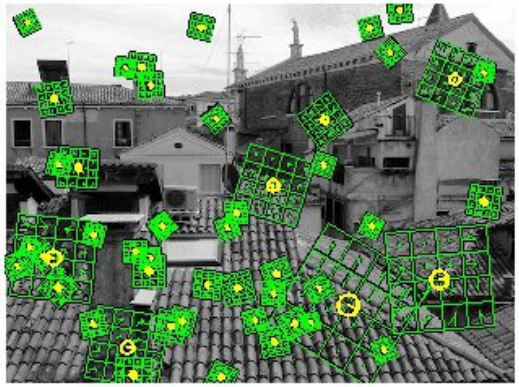


- Images as a collection of points/features

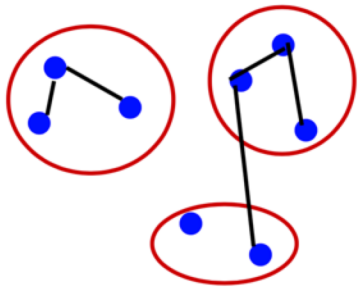


- Images as vectors/matrices/tensors





Problem Formulation



Graph G = (V, E)

V is set of nodes (i.e. pixels)

E is a set of undirected edges between pairs of pixels

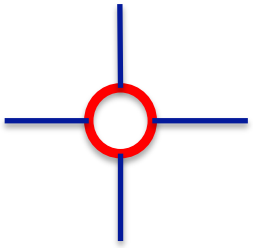
$w(v_i, v_j)$ is the weight of the edge between nodes v_i and v_j.

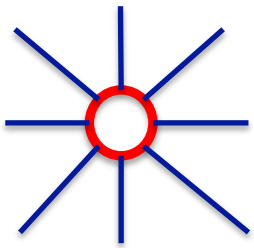
S is a segmentation of a graph G such that G' = (V, E')

where $E' \subset E$.

S divides G into G' such that it contains distinct components (or regions) C.

4-connected pixel lattice or 8-connected pixel lattice



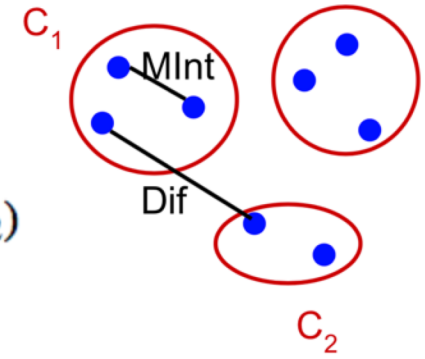


Efficient Graph-Based Image Segmentation (Felzenszwalb and Huttenlocher)

Predicate for Segmentation

Predicate D determines whether there is a boundary for segmentation.

$$D(C_1, C_2) = \begin{cases} \text{true} & \text{if } Dif(C_1, C_2) > MInt(C_1, C_2) \\ \text{false} & \text{otherwise} \end{cases}$$



Where

Dif(C_1, C_2) is the difference between two components.

MInt(C_1, C_2) is the internal different in the components C_1 and C_2

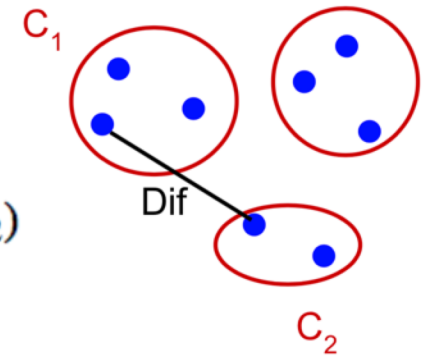
Efficient Graph-Based Image Segmentation (Felzenszwalb and Huttenlocher)

Predicate for Segmentation

Predicate D determines whether there is a boundary for segmentation.

$$D(C_1, C_2) = \begin{cases} \text{true} & \text{if } Dif(C_1, C_2) > MInt(C_1, C_2) \\ \text{false} & \text{otherwise} \end{cases}$$

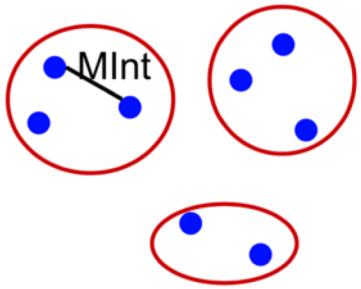
$$Dif(C_1, C_2) = \min_{v_i \in C_1, v_j \in C_2, (v_i, v_j) \in E} w(v_i, v_j).$$



The different between two components is the minimum weight edge that connects a node v_i in component C_1 to node v_j in C_2

Efficient Graph-Based Image Segmentation (Felzenszwalb and Huttenlocher)

Predicate for Segmentation



Predicate D determines whether there is a boundary for segmentation.

$$D(C_1, C_2) = \begin{cases} \text{true} & \text{if } Dif(C_1, C_2) > MInt(C_1, C_2) \\ \text{false} & \text{otherwise} \end{cases}$$

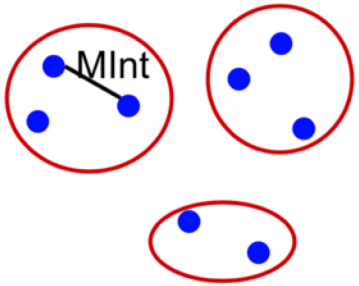
$$Dif(C_1, C_2) = \min_{v_i \in C_1, v_j \in C_2, (v_i, v_j) \in E} w(v_i, v_j).$$

$$Int(C) = \max_{e \in MST(C,E)} w(e).$$

Int(C) is to the maximum weight edge that connects two nodes in the same component.

Efficient Graph-Based Image Segmentation (Felzenszwalb and Huttenlocher)

Predicate for Segmentation



Predicate D determines whether there is a boundary for segmentation.

$$D(C_1, C_2) = \begin{cases} \text{true} & \text{if } Dif(C_1, C_2) > MInt(C_1, C_2) \\ \text{false} & \text{otherwise} \end{cases}$$

$$Dif(C_1, C_2) = \min_{v_i \in C_1, v_j \in C_2, (v_i, v_j) \in E} w(v_i, v_j).$$

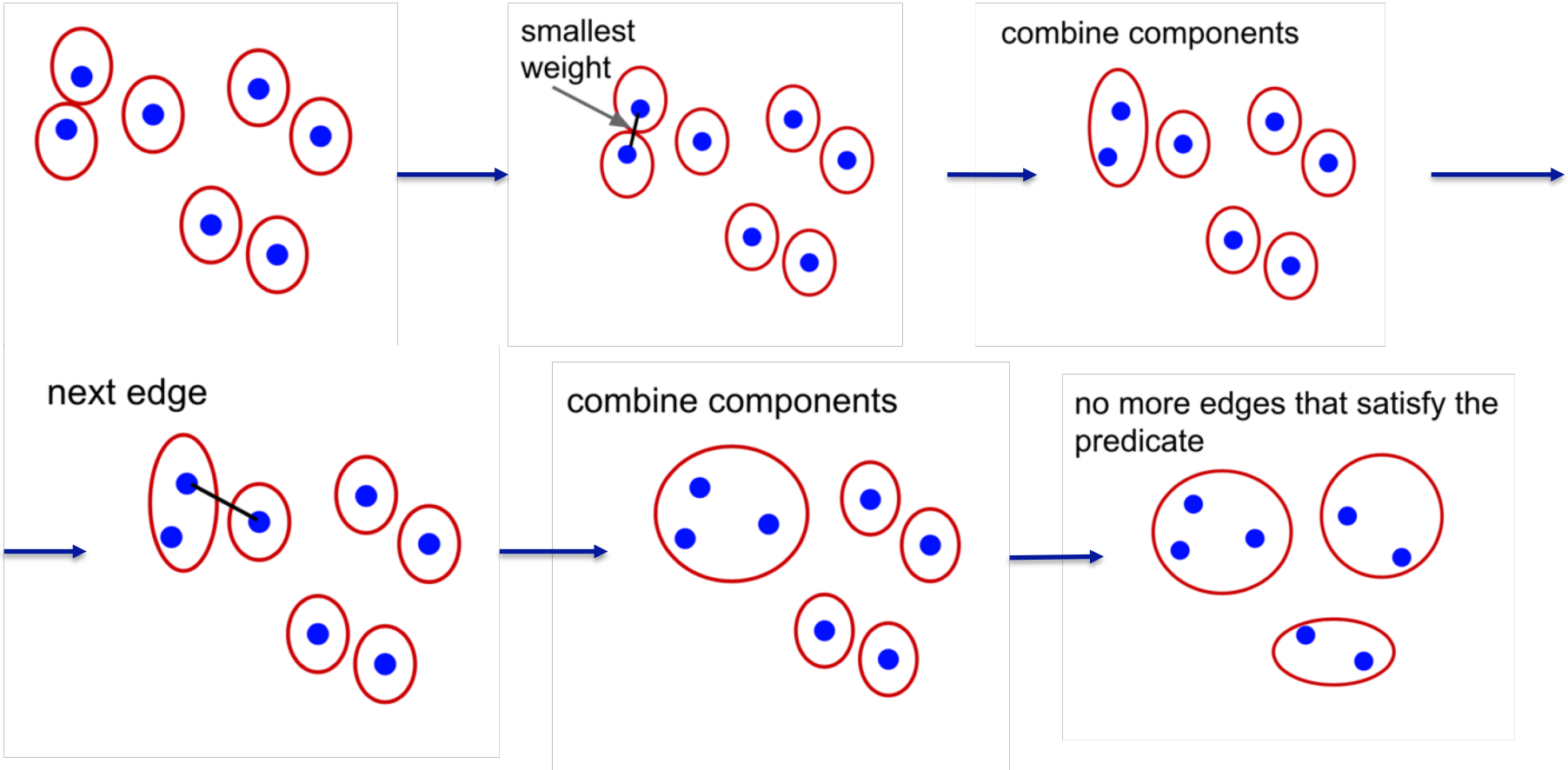
$$Int(C) = \max_{e \in MST(C,E)} w(e).$$

$$MInt(C_1, C_2) = \min(Int(C_1) + \tau(C_1),\ Int(C_2) + \tau(C_2)).$$

where $\tau(C) = k/|C|$

Efficient Graph-Based Image Segmentation (Felzenszwalb and Huttenlocher)

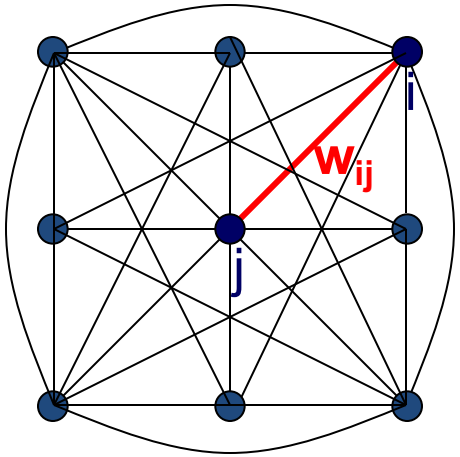
- 1. Sort E by non-decreasing edge weight
- 2. Start with each vertex is in its own component
- 3. Traverse the sorted edge set to exam the grouping criteria

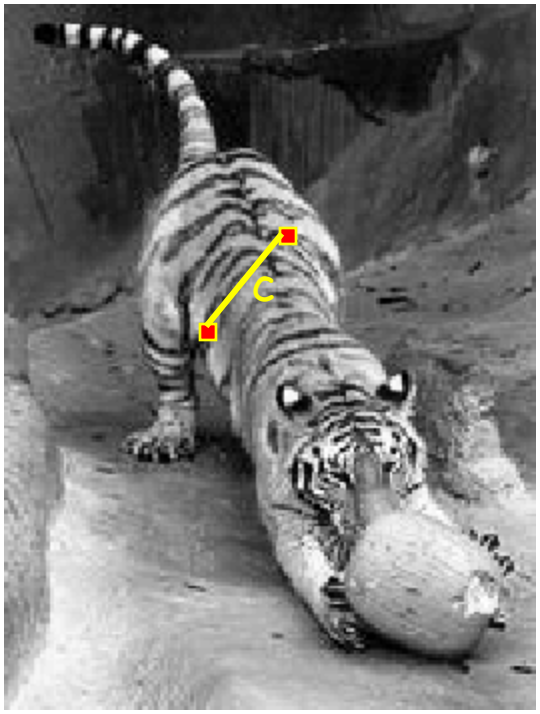


Efficient Graph-Based Image Segmentation (Felzenszwalb and Huttenlocher)

Normalized Cut

Images as graphs

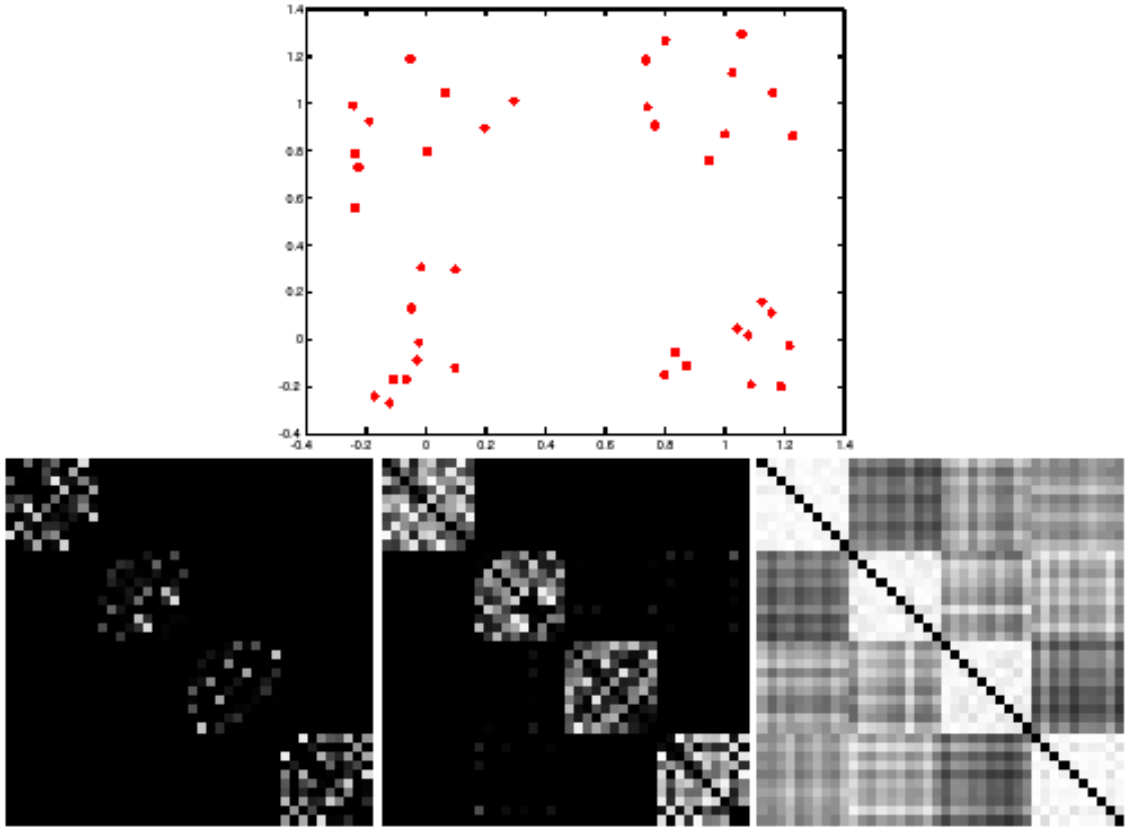




- *Fully-connected* graph
  - node for every pixel
  - link between *every* pair of pixels, **p**,**q**
  - similarity $\mathbf{w_{ij}}$ for each link

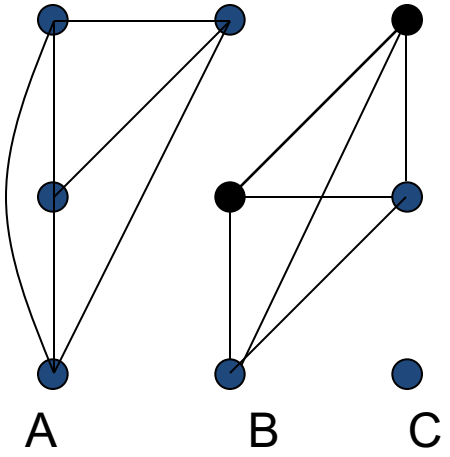
Source: Seitz

Similarity matrix



Increasing sigma

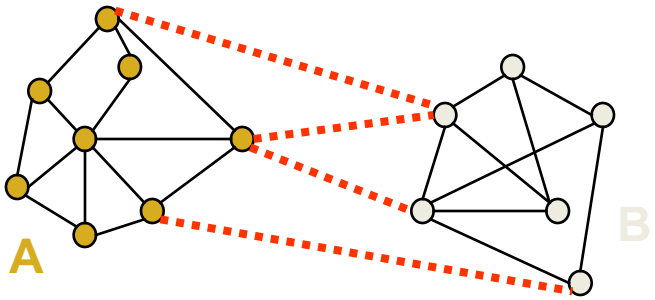
Segmentation by Graph Cuts



- Break Graph into Segments
  - Delete links that cross between segments
  - Easiest to break links that have low cost (low similarity)
    - similar pixels should be in the same segments
    - dissimilar pixels should be in different segments

Source: Seitz

Cuts in a graph



- Link Cut
  - set of links whose removal makes a graph disconnected
  - cost of a cut:

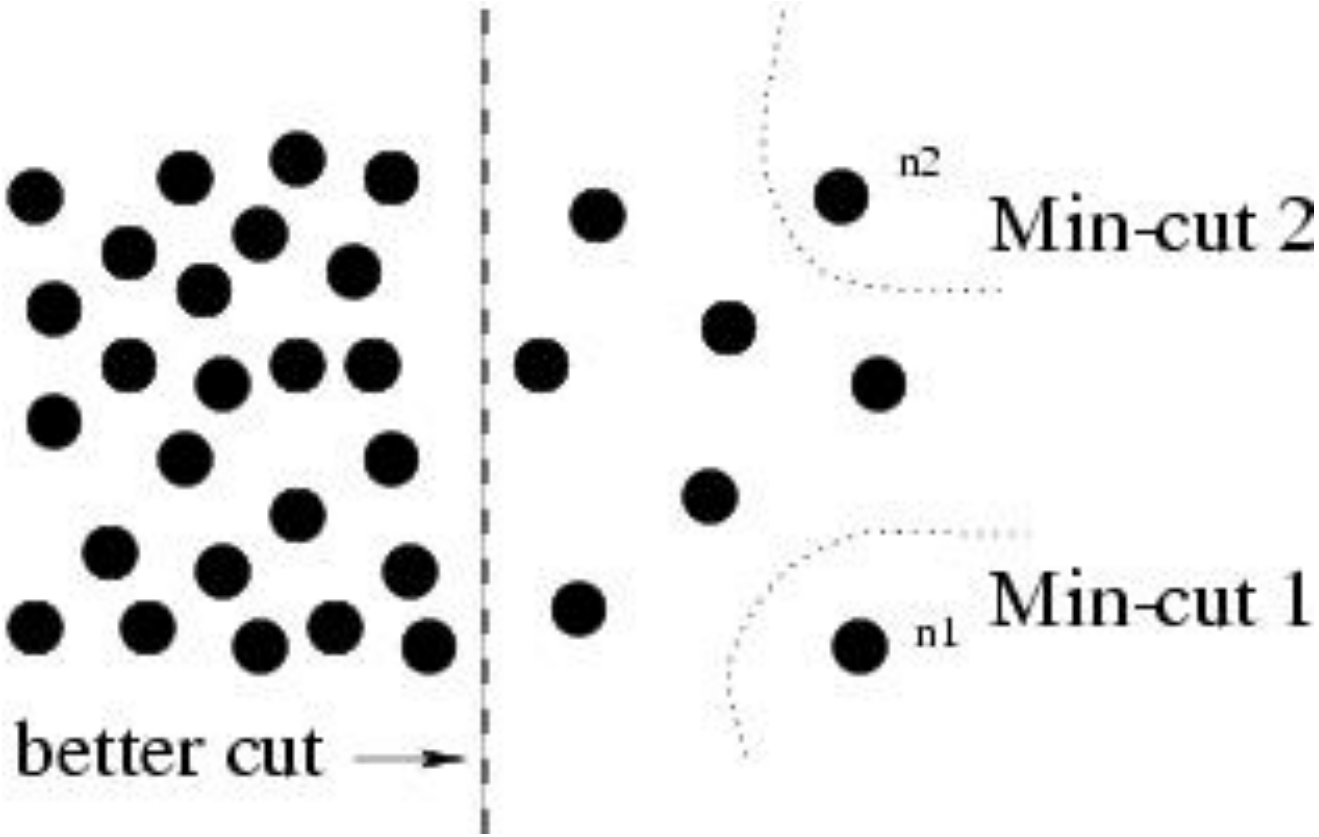
$$cut(A, B) = \sum_{p \in A, q \in B} c_{p,q}$$

One idea: Find minimum cut

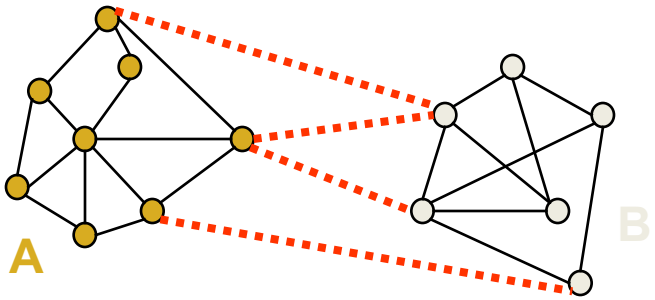
- gives you a segmentation
- fast algorithms exist for doing this

Source: Seitz

But min cut is not always the best cut...



Cuts in a graph



Normalized Cut

- a cut penalizes large segments
- fix by normalizing for size of segments

$$Ncut(A,B) = \frac{cut(A,B)}{volume(A)} + \frac{cut(A,B)}{volume(B)}$$

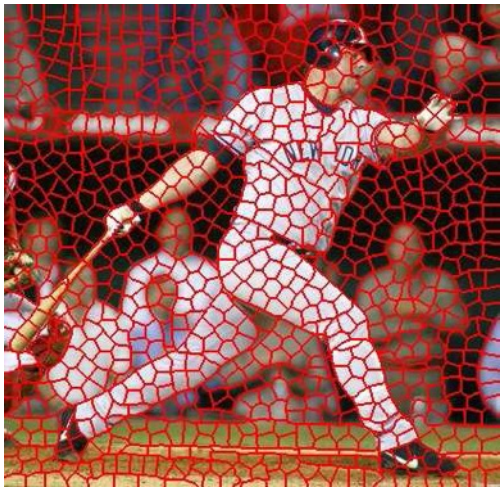
- volume(A) = sum of costs of all edges that touch A

Source: Seitz

Normalized cuts results

Normalized cuts: Pro and con

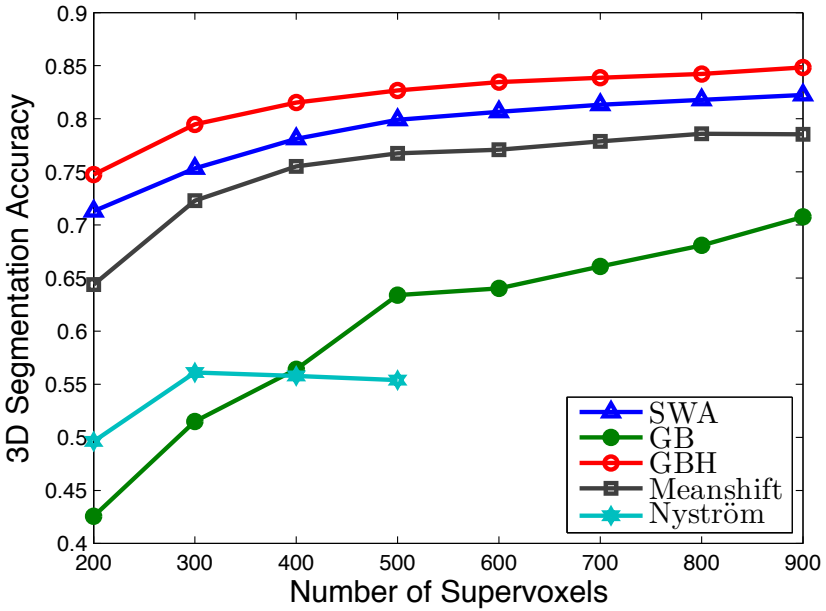
- Pros
  - Generic framework, can be used with many different features and affinity formulations
  - Provides regular segments
- Cons
  - Need to chose number of segments
  - High storage requirement and time complexity
  - Bias towards partitioning into equal segments
- Usage
  - Use for oversegmentation when you want regular segments

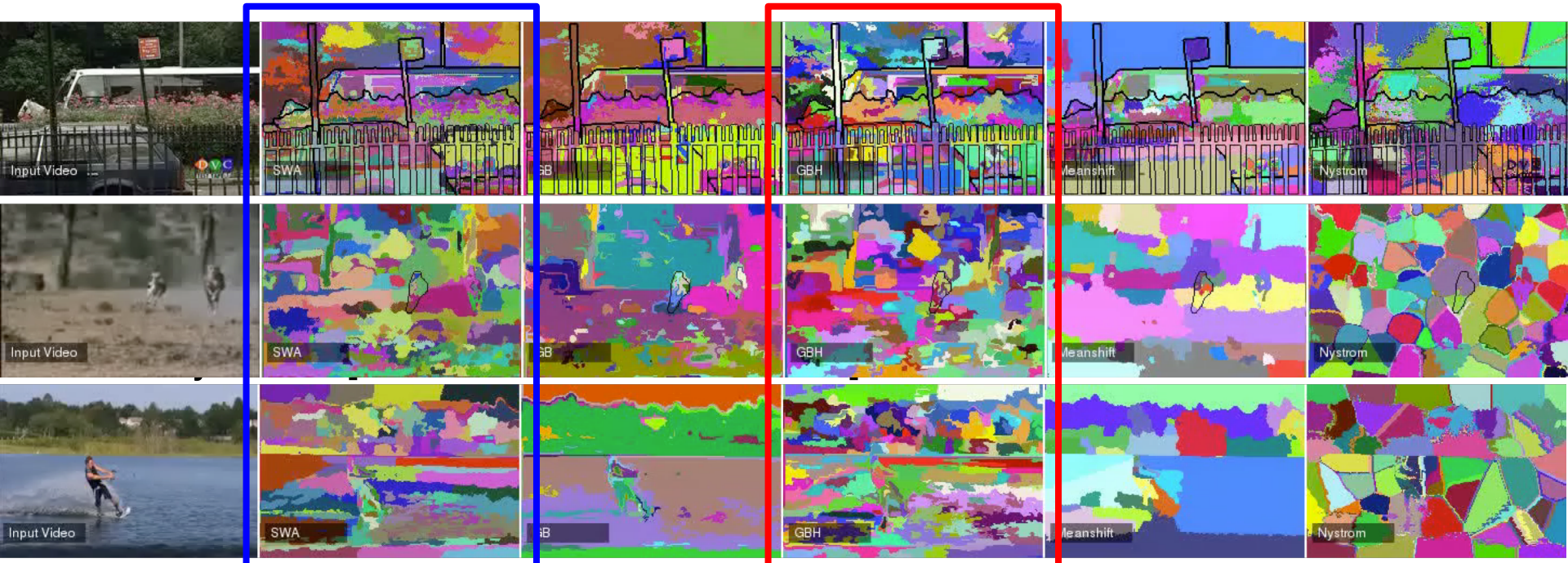


Bottom-Up Video Segmentation

Extending Image Methods to Videos

- **LIBSVX Key Finding:**
  - Hierarchical methods performed best in [Xu & Corso CVPR 2012].
- A set of properties are measured
  - Spatiotemporal uniformity.
  - Boundary preservation.
  - Motion consistency.





SWA GB GBH MeanShift Nyström

Streaming Hierarchical Video Segmentation

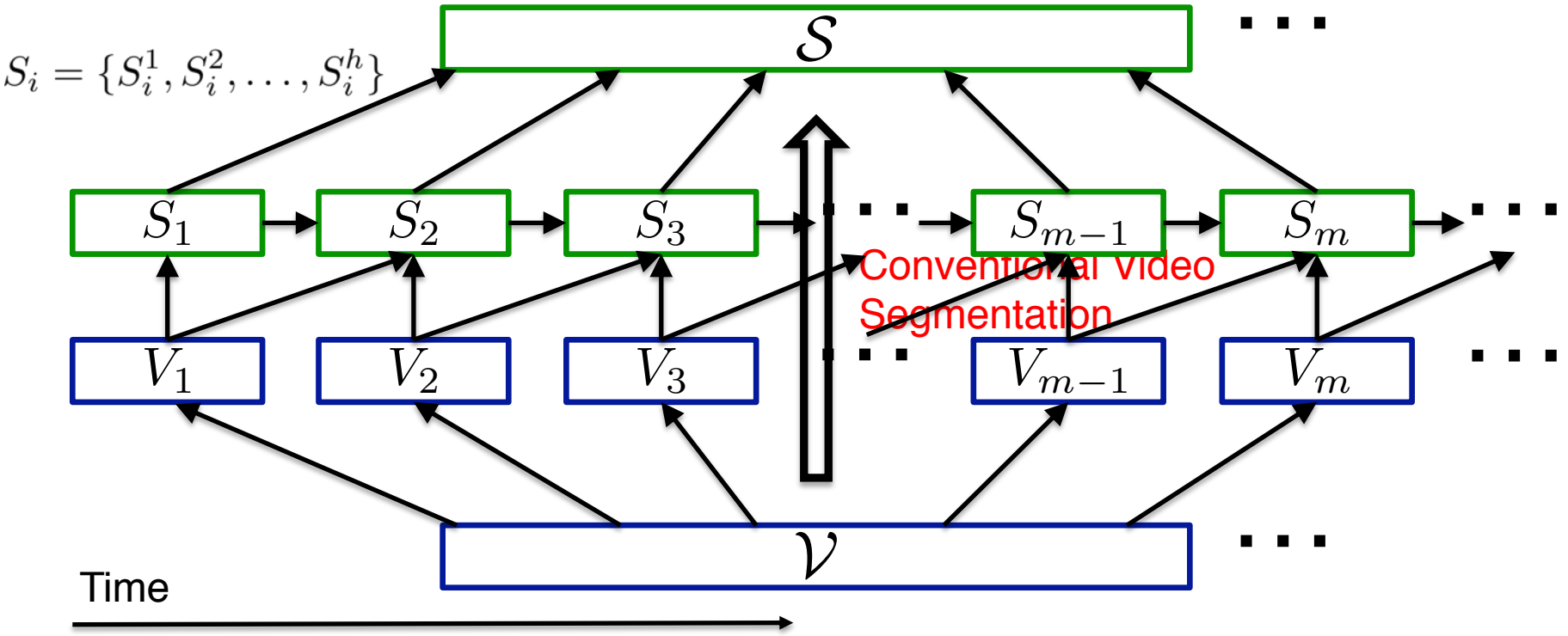
- Basic problem statement:
- Segmentation hierarchy

$$\mathcal{S} \doteq \{S^1, S^2, \ldots, S^h\}$$

- $S^i \doteq \{s_1, s_2, \ldots\}$ is one level of supervoxel segmentation.

Segmentation Video Input

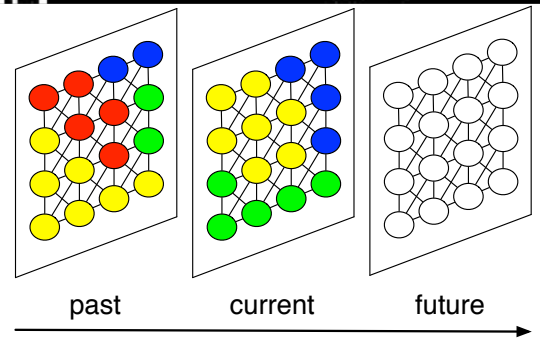
$$\mathcal{S}^* = \underset{\mathcal{S}}{\operatorname{argmin}}\, E(\mathcal{S}|\mathcal{V})$$



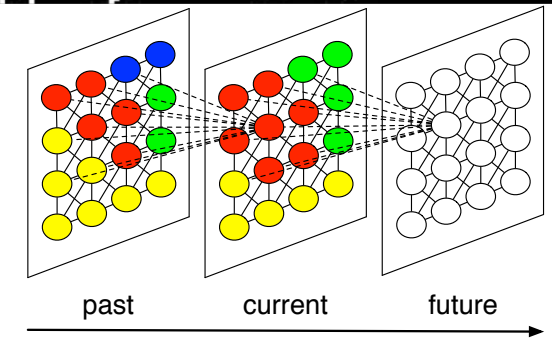
[Xu et al. ECCV 2012]

Why Streaming?

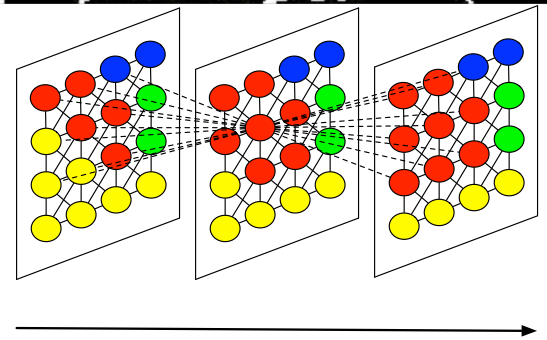
- Streaming is needed.
  - We can **bound memory needs** and **handle arbitrarily long** videos **with good consistency** of segmentation.



Frame-by-Frame
[Brendel and Todorovic ICCV 2009]
[Lee et al. CVPR 2011]



Streaming
[Paris ECCV 2008]
[Grundmann et al. CVPR 2010]
(Clip-based)



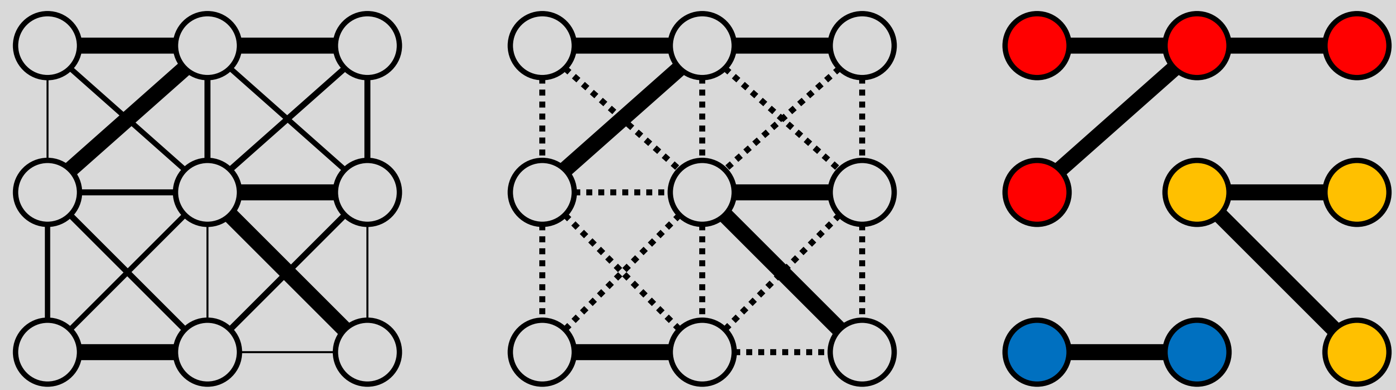
Full Video
[Paris and Durand CVPR 2007]
[Grundmann et al. CVPR 2010]
[Lezama et al. CVPR 2011]

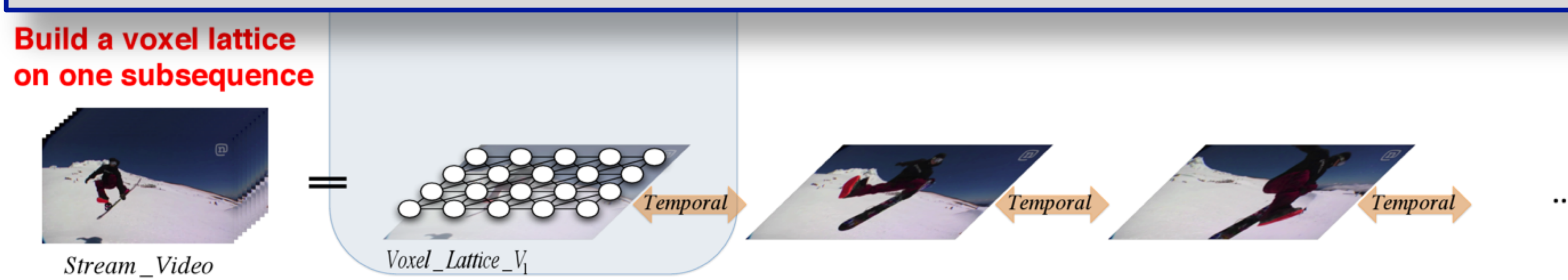
Streaming Hierarchical Video Segmentation

- Use the minimum spanning tree method of [Felzenszwalb and Huttenlocher IJCV'04]; our approximation framework is general.

$$E(S^1|\mathcal{V}) = \tau \sum_{s \in S^1} \sum_{e \in \mathrm{MST(s)}} w(e) + \sum_{s,t \in S^1} \min_{e \in \langle s,t \rangle} w(e)$$

Stage 3: [illegible] $E(S^1|\mathcal{V})$.



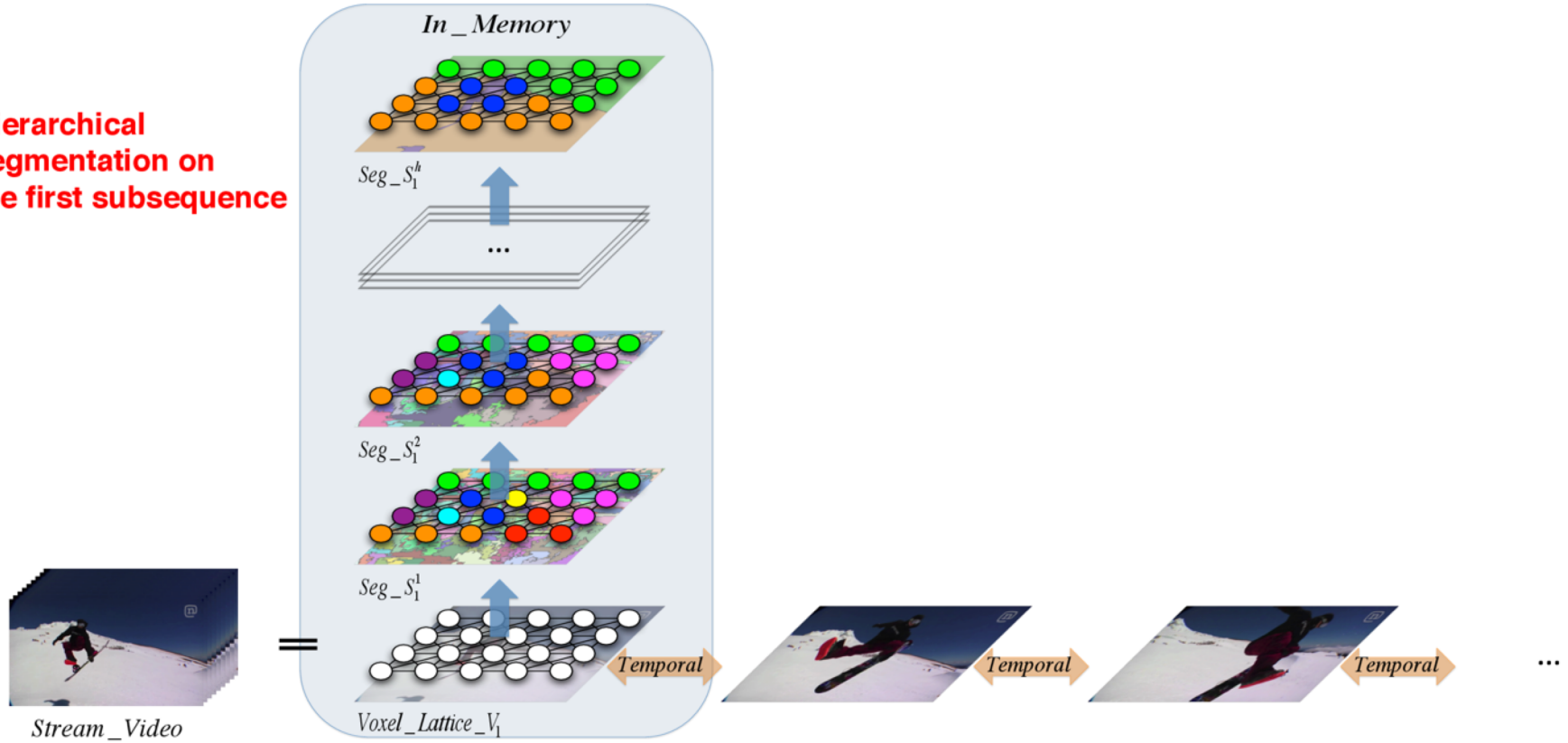


[Xu et al. ECCV 2012]

Streaming Hierarchical Video Segmentation

- Similarity between regions in the hierarchy is reevaluated with multiscale features.

Hierarchical segmentation on the first subsequence



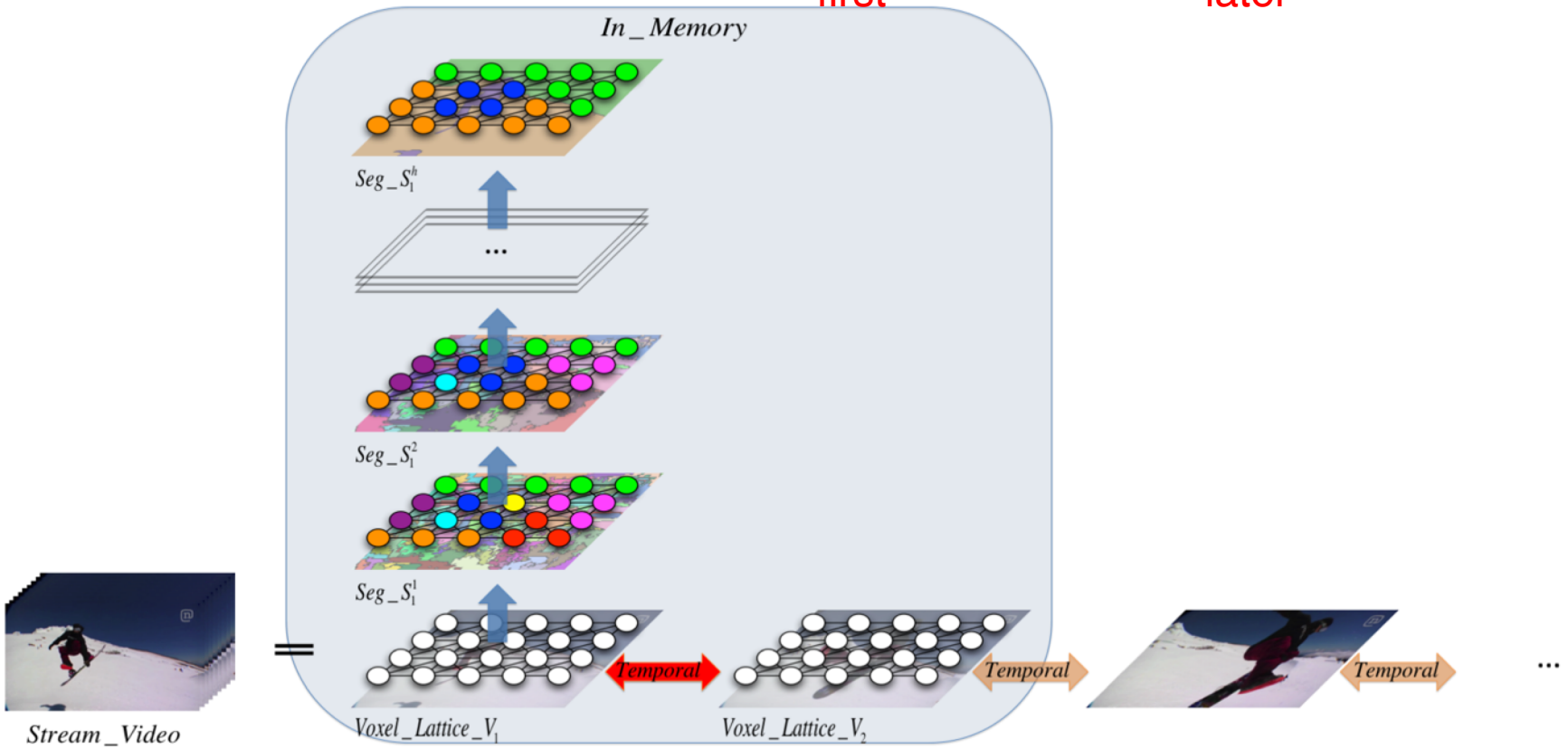
[Xu et al. ECCV 2012]

Streaming Hierarchical Video Segmentation

- Streaming Markov assumption.

$$\mathcal{S} = \{S_1, \cdots, S_m\} = \underset{S_1, S_2, \cdots, S_m}{\operatorname{argmin}} \left[E^1(S_1|V_1) + \sum_{i=2}^{m} E^1(S_i|V_i, S_{i-1}, V_{i-1}) \right]$$

first later



[Xu et al. ECCV 2012]

Streaming Hierarchical Video Segmentation

- Streaming & hierarchical Markov assumptions.

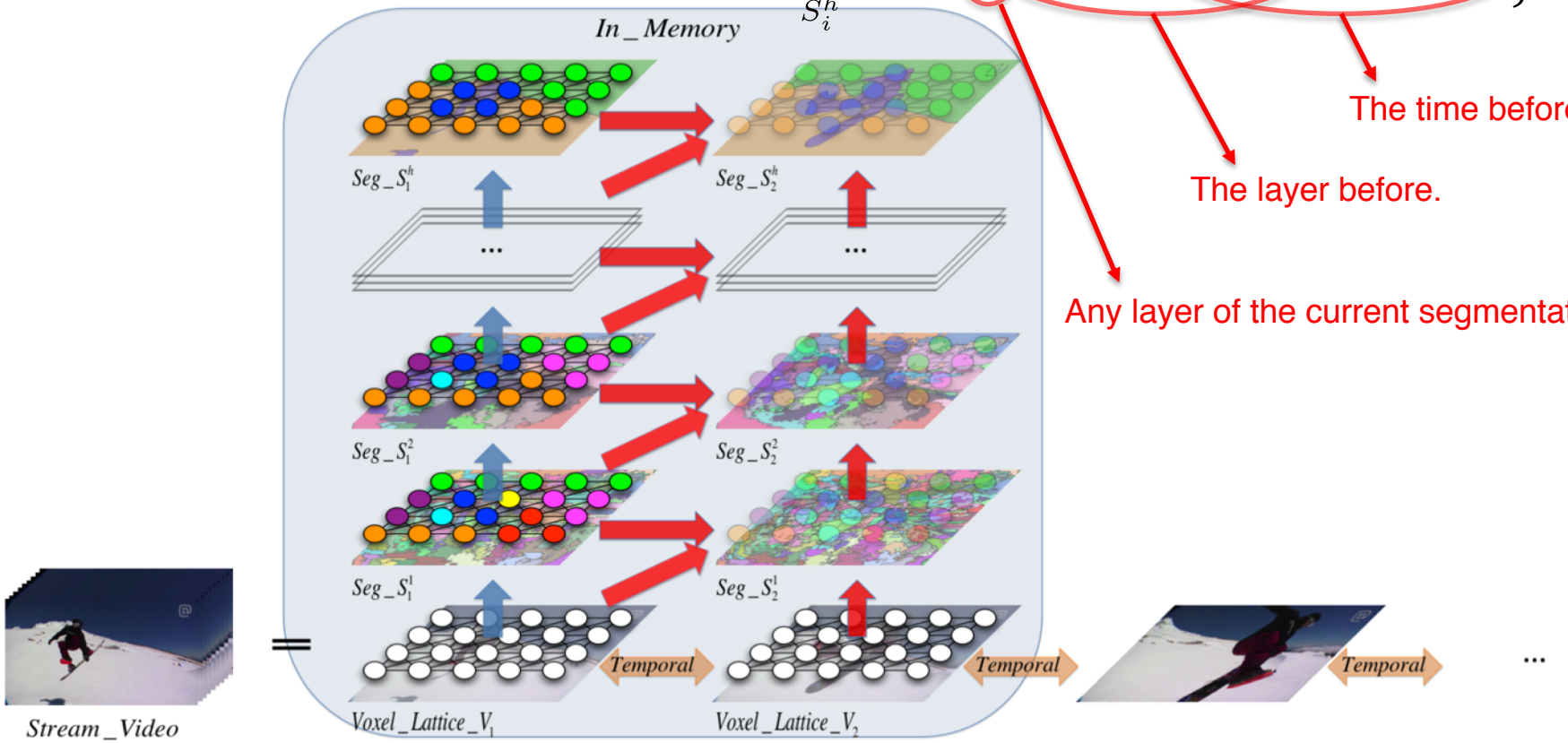
$$S_i = \underset{S_i}{\operatorname{argmin}}\, E^1(S_i|V_i, S_{i-1}, V_{i-1}) = \Big\{ \underset{S_i^2}{\operatorname{argmin}}\, E^2(S_i^2|V_i, S_i^1, S_{i-1}^1, S_{i-1}^2, V_{i-1}), \cdots,$$

$$\underset{S_i^h}{\operatorname{argmin}}\, E^2(S_i^h|V_i, S_i^{h-1}, S_{i-1}^{h-1}, S_{i-1}^h, V_{i-1}) \Big\}$$

The time before.

The layer before.

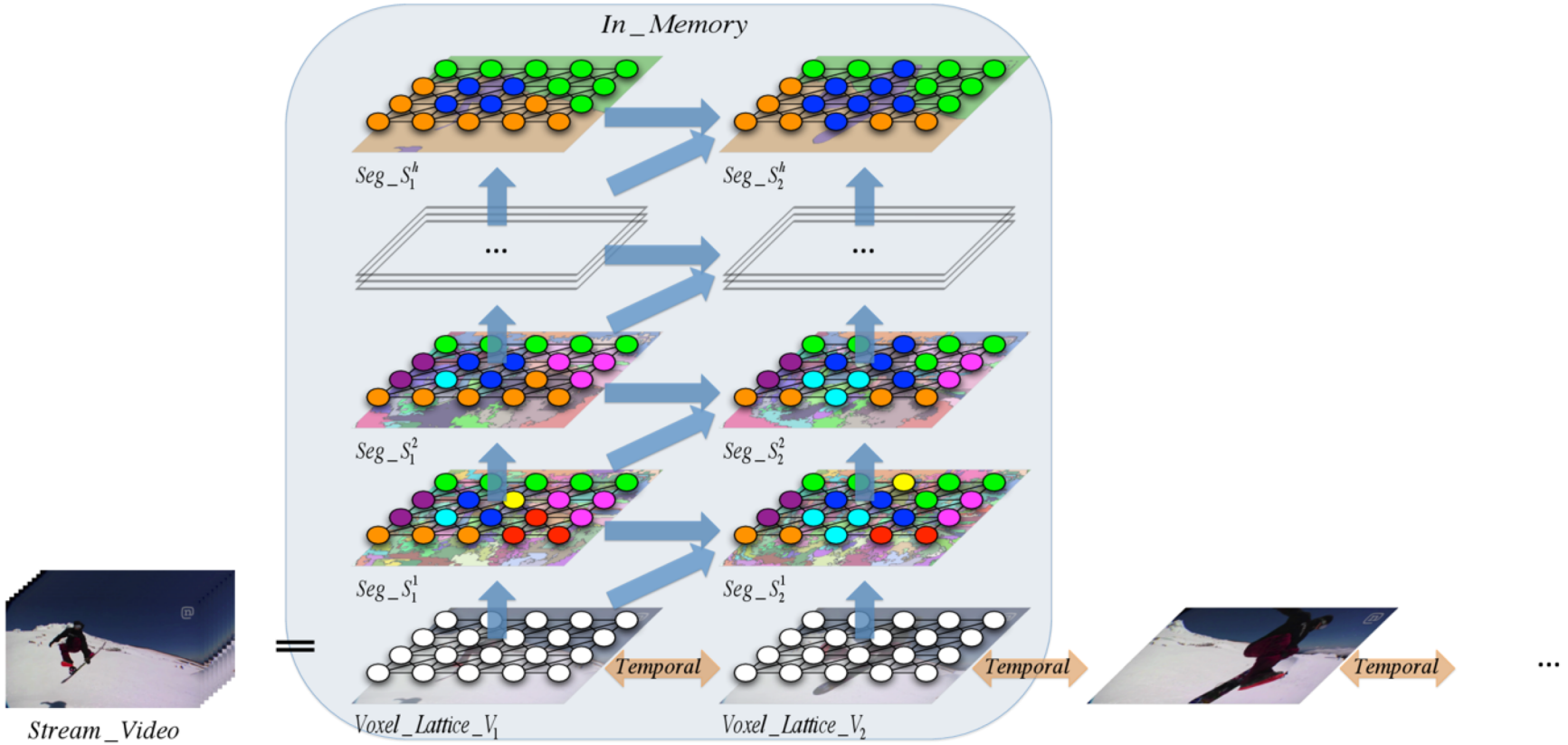
Any layer of the current segmentation.



[Xu et al. ECCV 2012]

Streaming Hierarchical Video Segmentation

- Finish the hierarchical segmentation at the current stream pointer time.

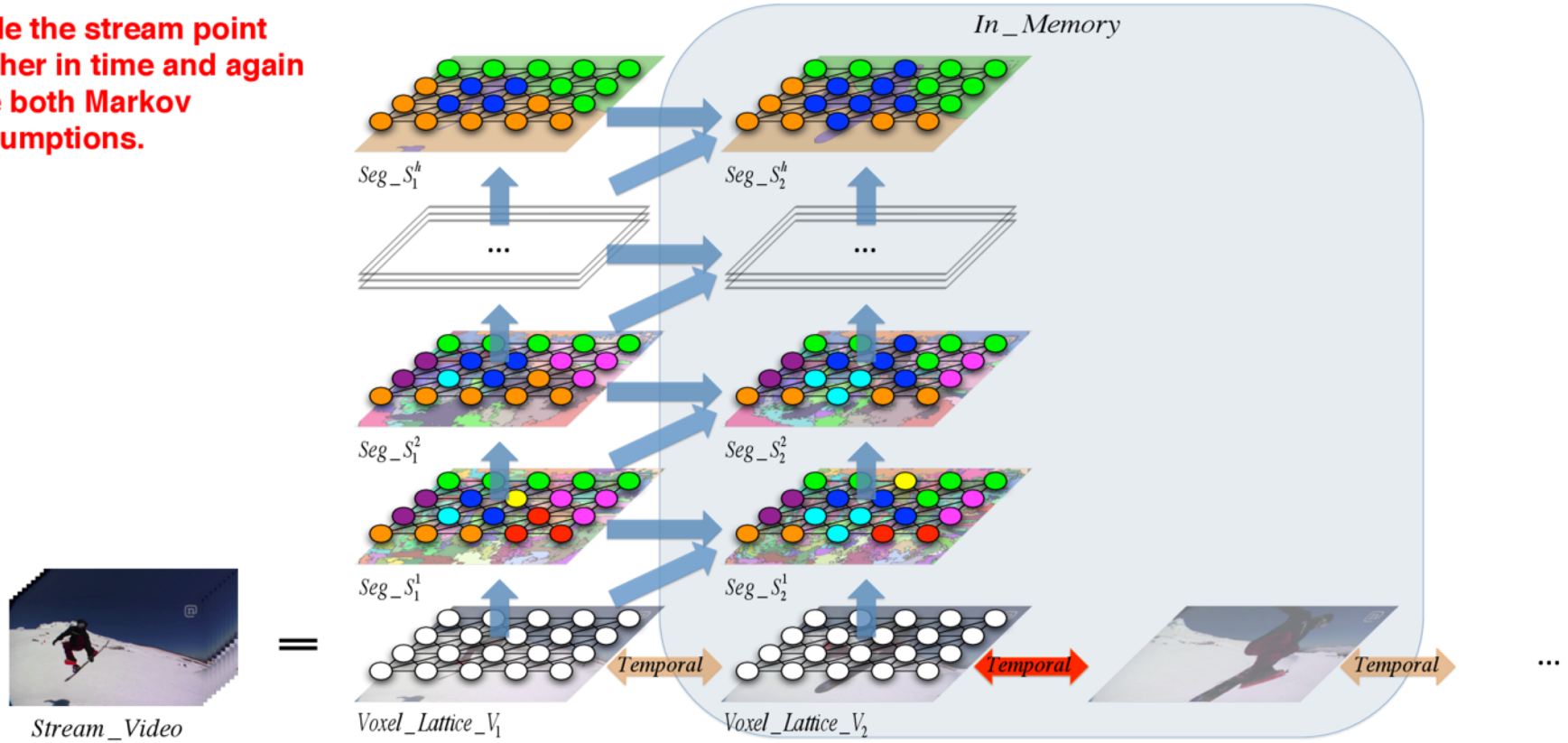


[Xu et al. ECCV 2012]

Streaming Hierarchical Video Segmentation

- Once finished with two subsequences, move the stream pointer forward.
- Offload the earlier subsequence from memory and load the next.

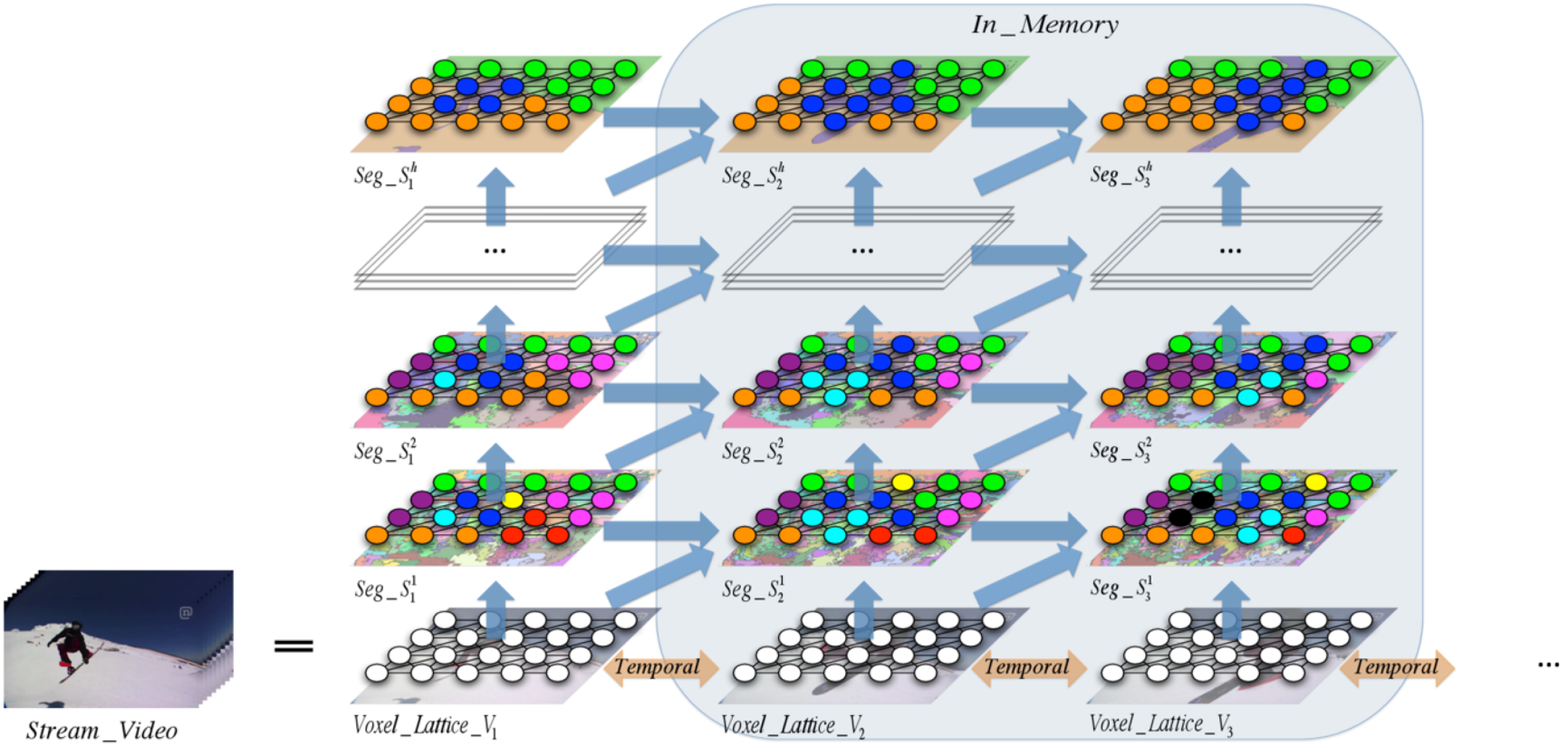
Slide the stream point further in time and again use both Markov assumptions.



[Xu et al. ECCV 2012]

Streaming Hierarchical Video Segmentation

- Segment again…



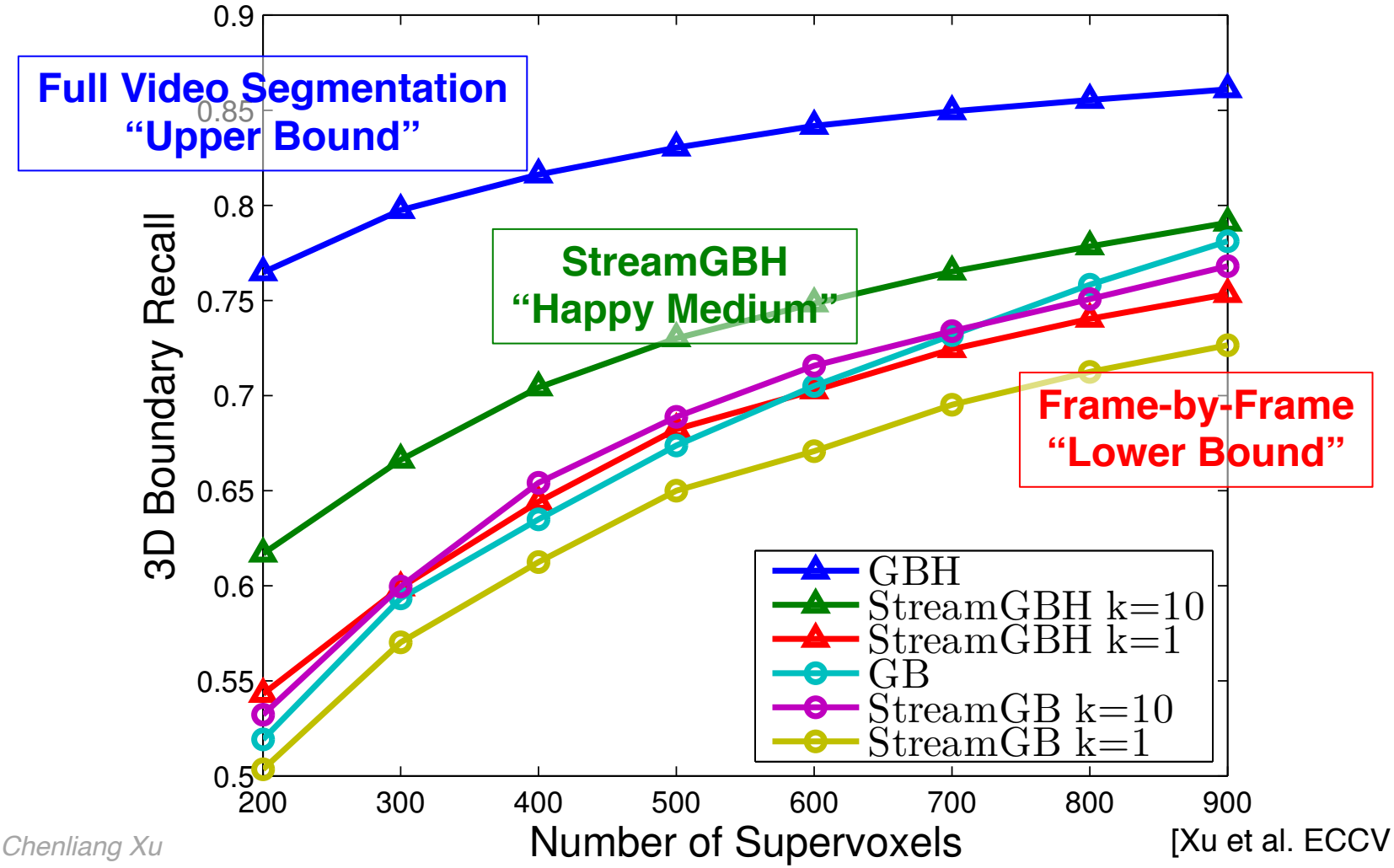
[Xu et al. ECCV 2012]

Example Segmentation Hierarchy

[Xu et al. ECCV 2012]

StreamGBH Quantitative Comparisons

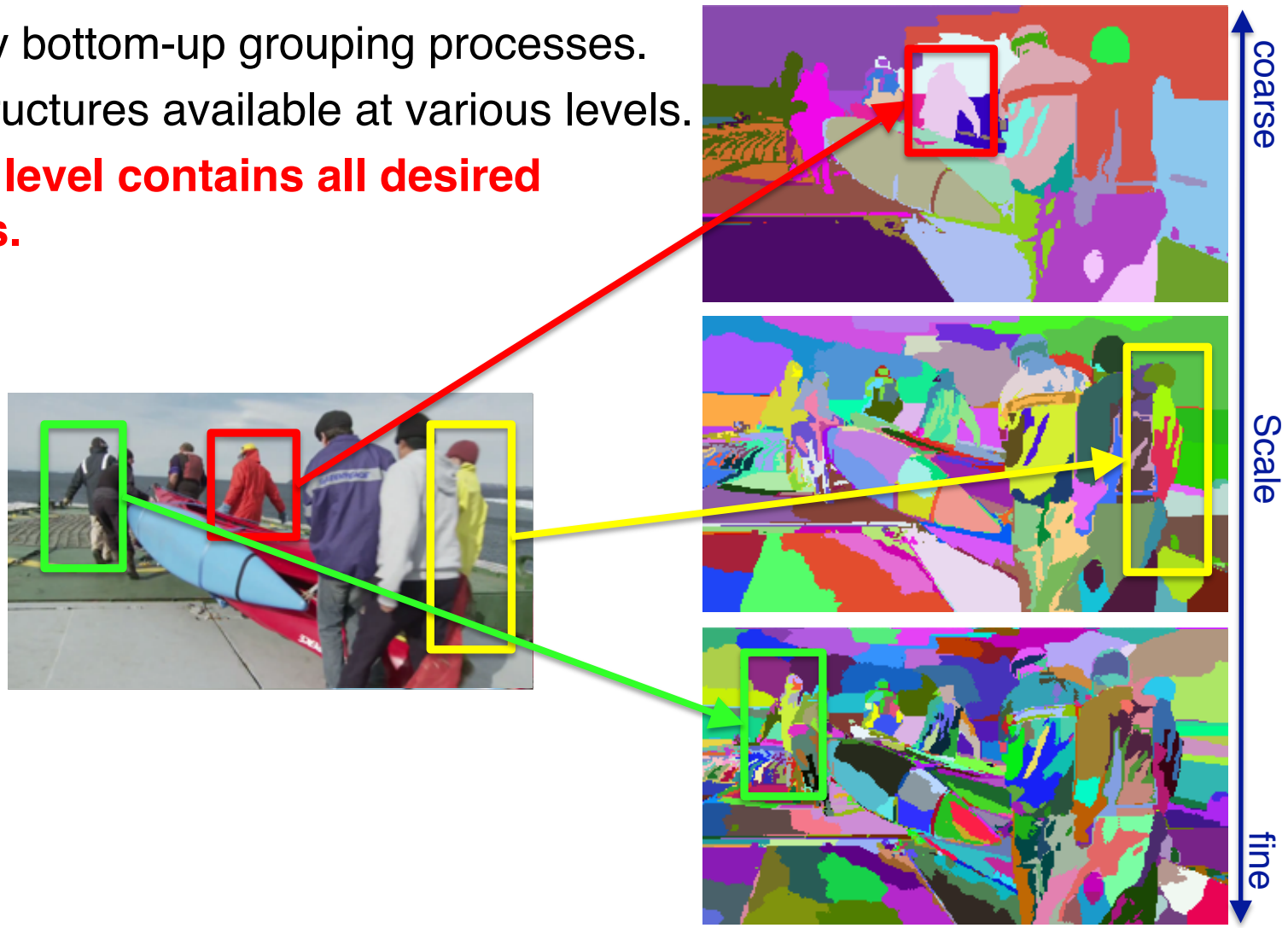
- Does StreamGBH balance between frame-to-frame methods and full-video methods?



[Xu et al. ECCV 2012]

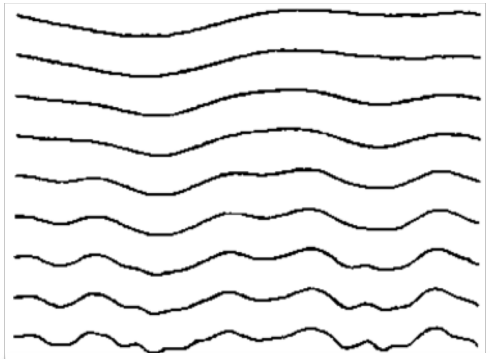
Semantic Scale Space

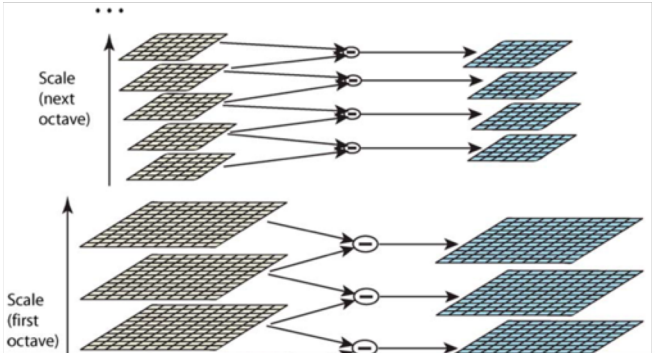
- Induced by bottom-up grouping processes.
- Various structures available at various levels.
- **No single level contains all desired structures.**

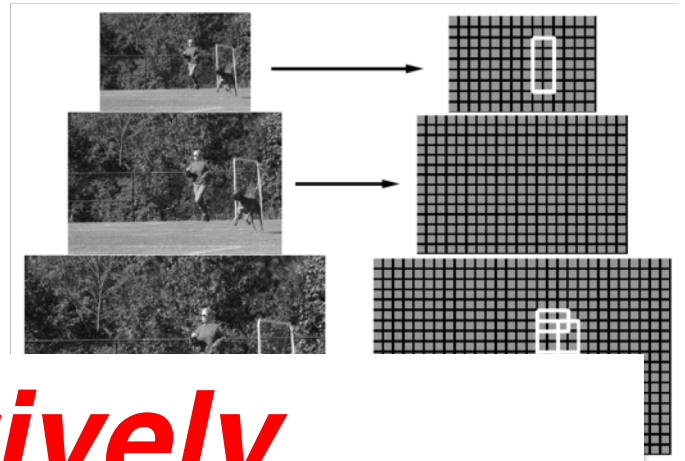


Scale — A Fundamental Problem in Computer Vision

- Exhaustive Search







Can we *adaptively* choose the scales?

Source: [Uijlings et al IJCV'13]. It is used in R-CNN object detection system [Girshick et al. CVPR'13].

Case Study:
Combining Bottom-Up and Top-Down

What action is happening? Who is performing the action? Where is the action happening?

- Pose them as a **video labeling problem** of **actors** and **actions**.

[Xu et al., CVPR 2015, CVPR 2016]

What action is happening? Who is performing the action? Where is the action happening?

- Pose them as a **video labeling problem** of **actors** and **actions**.

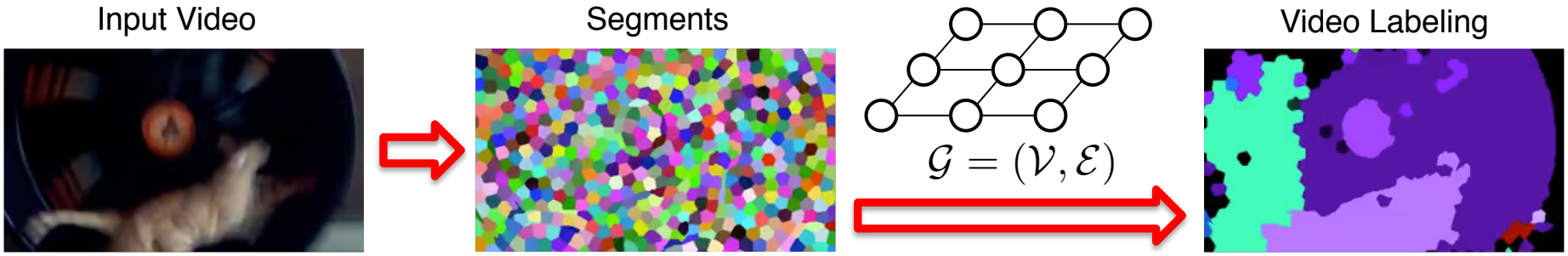
- Complementary to a collective of problems.

	Classification	Detection	Segmentation
Action	[Simonyan et al. NIPS'14] [Wang et al. IJCV'13] [Sadanand et al. CVPR'12]	[Weinzaepfel et al. ICCV'15] [Jain et al. CVPR'14] [Tian et al. CVPR'13]	[Lu et al. CVPR'15] [Jhuang et al. ICCV'13] [Ma et al. ICCV'13]
Object	[Krizhevsky et al. NIPS'12] [Lin et al. CVPR'11] [Deng et al. ECCV'10]	[Zhang et al. ECCV'14] [Girshick et al. CVPR'13] [Felzenszwalb et al. TPAMI'10]	[Fuxin et al. ICCV'13] [Lee et al. ICCV'11] [Brostow et al. ECCV'08]

adult, dog, cat, walking

[Xu et al., CVPR 2015, CVPR 2016]

Segment-Level Labeling CRF

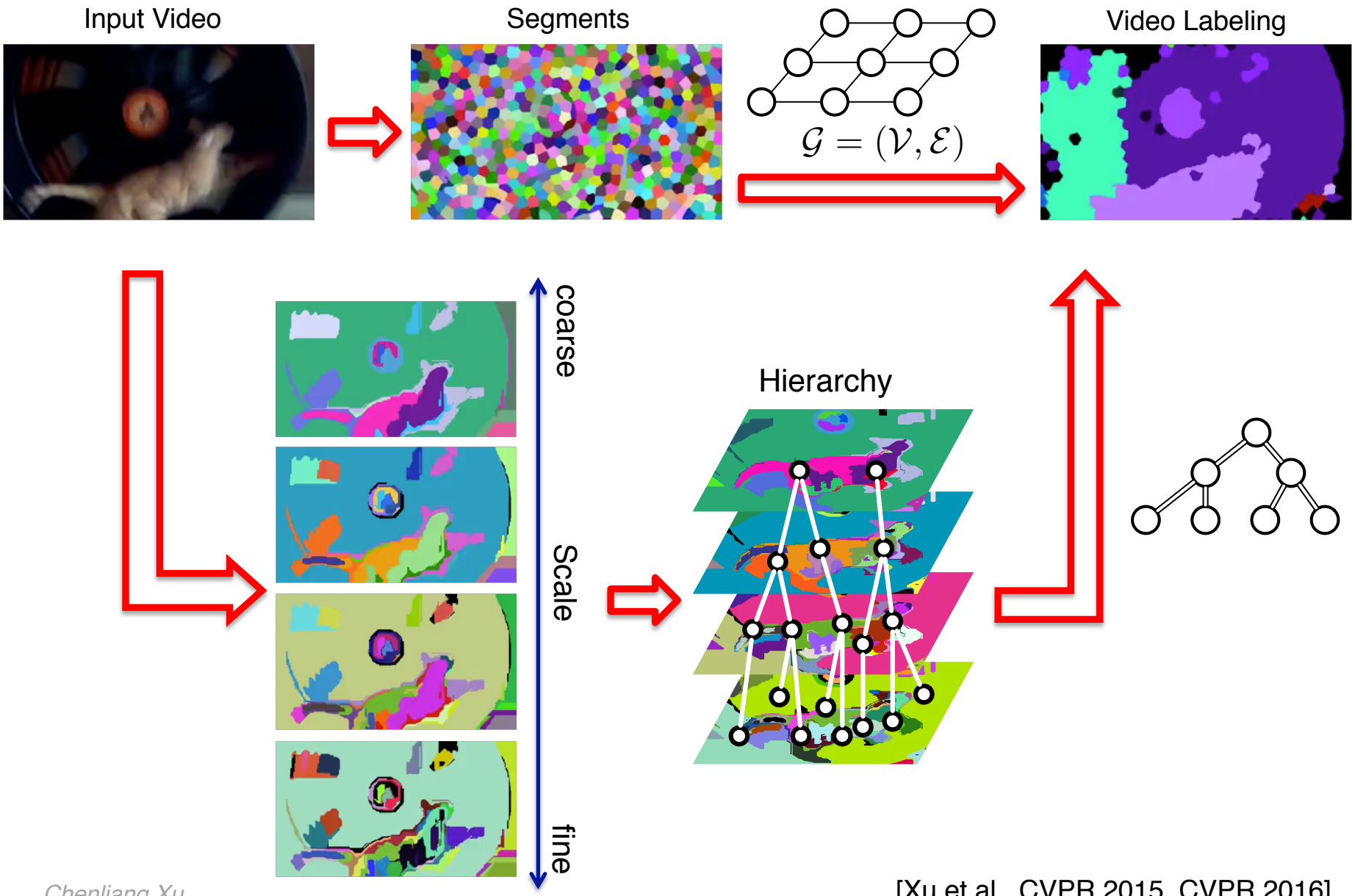


Local and unable to capture long-ranging interaction of video parts.

- CRF on a spatiotemporal graph for the video:
  - Nodes are supervoxels.
  - Edges connect neighbors sharing spatiotemporal boundaries.
- A set of features are computed:
  - Appearance: texton, sift, color, etc.
  - Motion: flow, dense trajectory, mbh, etc.

[Xu et al., CVPR 2015, CVPR 2016]

Our Approach to Overcome the Limitations

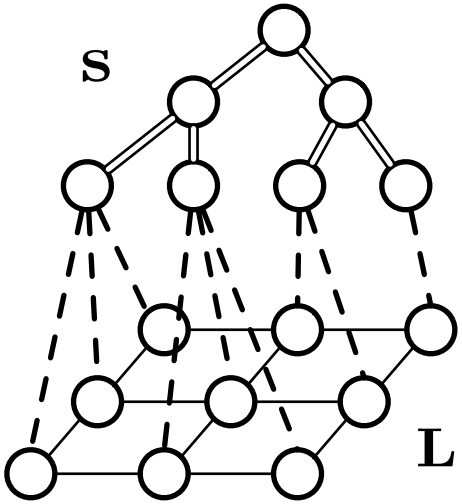


[Xu et al., CVPR 2015, CVPR 2016]

Grouping Process Model (GPM)

- A joint model of **scale selection** and **video labeling**.
- The overall objective function: $(\mathbf{L}^*, \mathbf{s}^*) = \arg\min_{\mathbf{L},\mathbf{s}} E(\mathbf{L}, \mathbf{s}|\mathcal{V}, \mathcal{T})$

Binary Labels
(1-selected, 0-not selected)



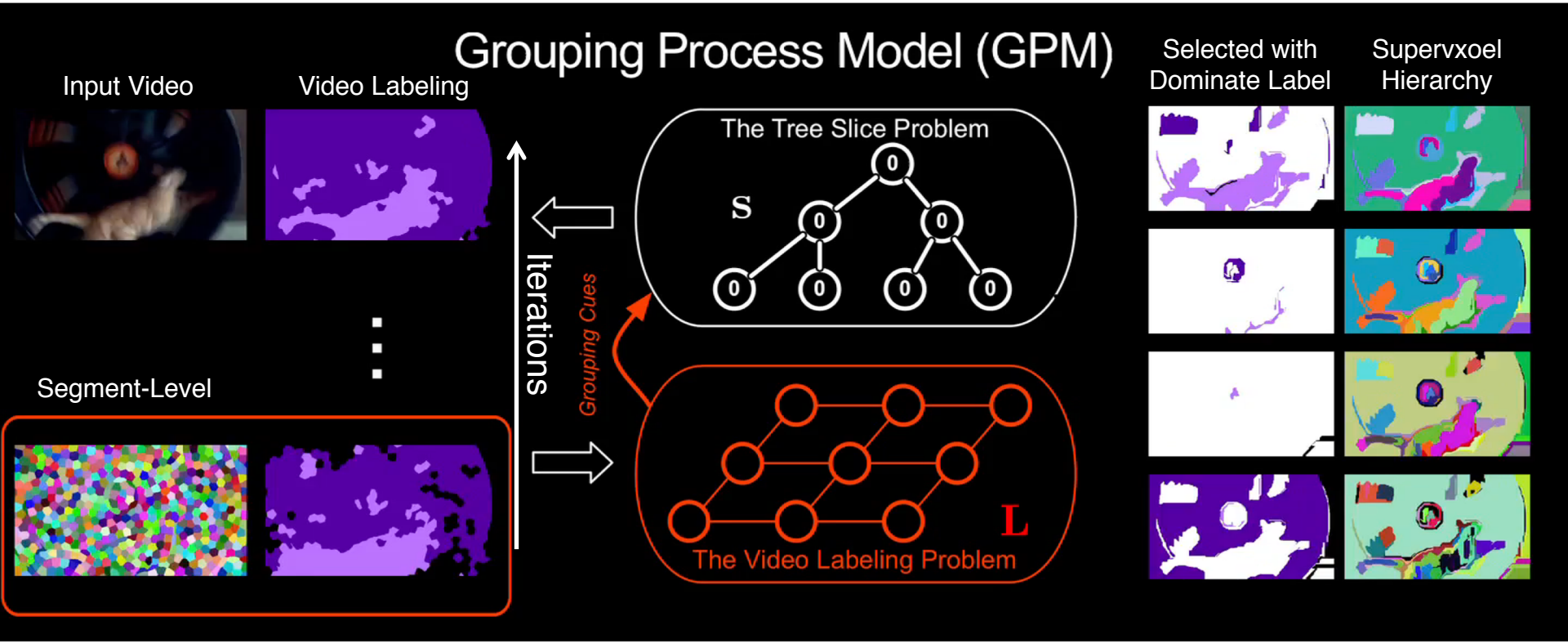
Actor Labels and Action Labels
(dog-crawling, bird-flying etc.)

Segment-level CRF | Tree Slice Constraint

$$E(\mathbf{L}, \mathbf{s}|\mathcal{V}, \mathcal{T}) = E^v(\mathbf{L}|\mathcal{V}) + E^h(\mathbf{s}|\mathcal{T}) + \sum_{t\in\mathcal{T}} (E^h(\mathbf{L}_t|s_t) + E^h(s_t|\mathbf{L}_t))$$

Grouping Cues | Selection Cues

[Xu et al., CVPR 2015, CVPR 2016]



The Video Labeling Problem

$$\mathbf{L}^* = \arg\min_{\mathbf{L}} E(\mathbf{L}|\mathbf{s}, \mathcal{V}, \mathcal{T})$$

$$= \arg\min_{\mathbf{L}} E^v(\mathbf{L}|\mathcal{V}) + \sum_{t \in \mathcal{T}} E^h(\mathbf{L}_t|s_t)$$

- Optimization depends on: $E^v(\mathbf{L}|\mathcal{V})$

✓ Solvable by graph-cuts multi-label inference.

The Tree Slice Problem

$$\mathbf{s}^* = \arg\min_{\mathbf{s}} E(\mathbf{s}|\mathbf{L}, \mathcal{V}, \mathcal{T})$$

$$= \arg\min_{\mathbf{s}} E^h(\mathbf{s}|\mathcal{T}) + \sum_{t \in \mathcal{T}} E^h(s_t|\mathbf{L}_t)$$

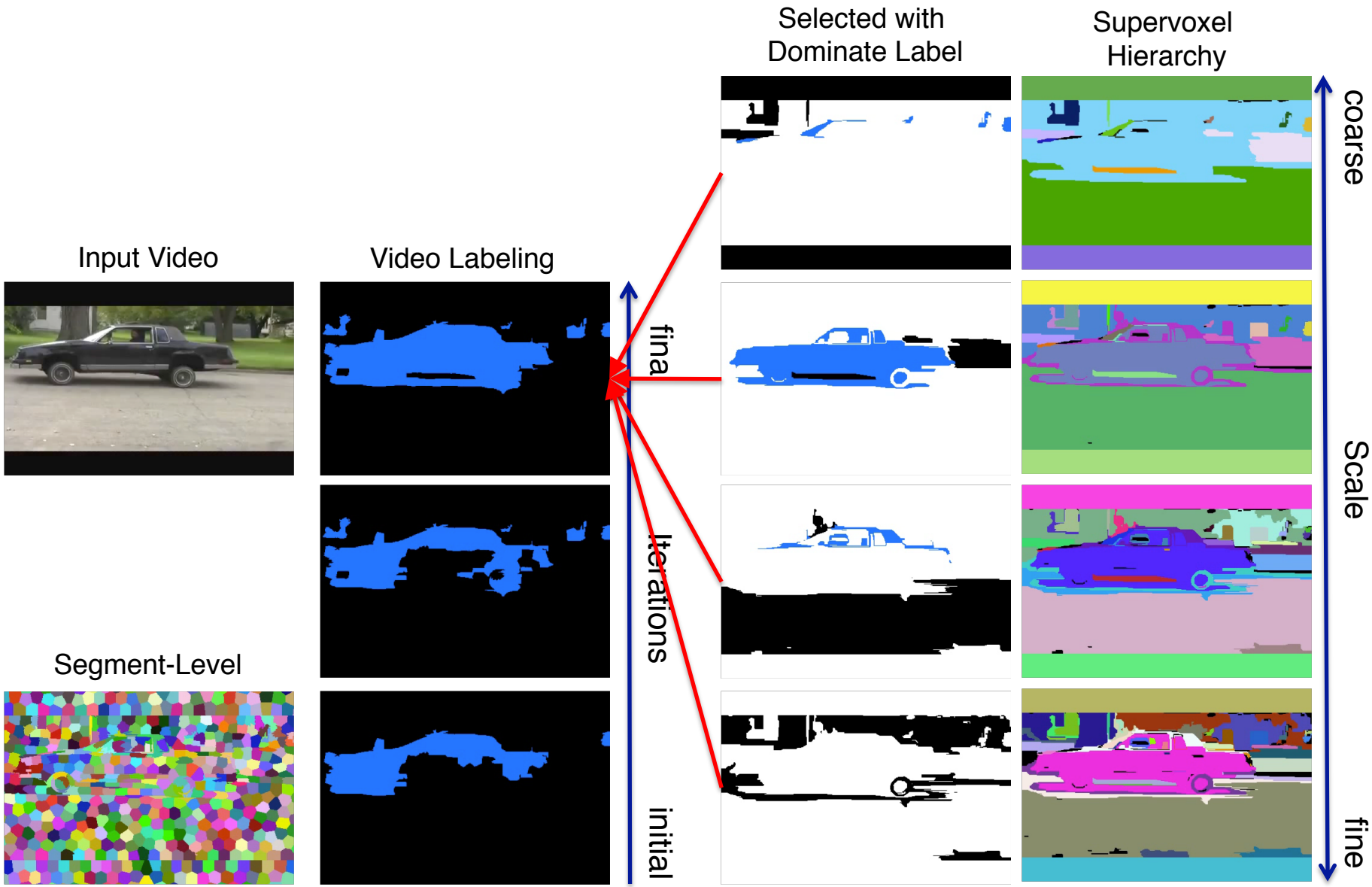
- This is a Binary LP.

$$\min \sum_{t \in \mathcal{T}} \alpha_t s_t \quad \text{s.t. } \mathcal{P}\mathbf{s} = \mathbf{1}_P \text{ and } \mathbf{s} = \{0,1\}^S$$

$$\alpha_t = \mathcal{H}(\mathbf{L}_t)|\mathbf{L}_t| + \theta_h$$

[Xu et al., CVPR 2015, CVPR 2016]

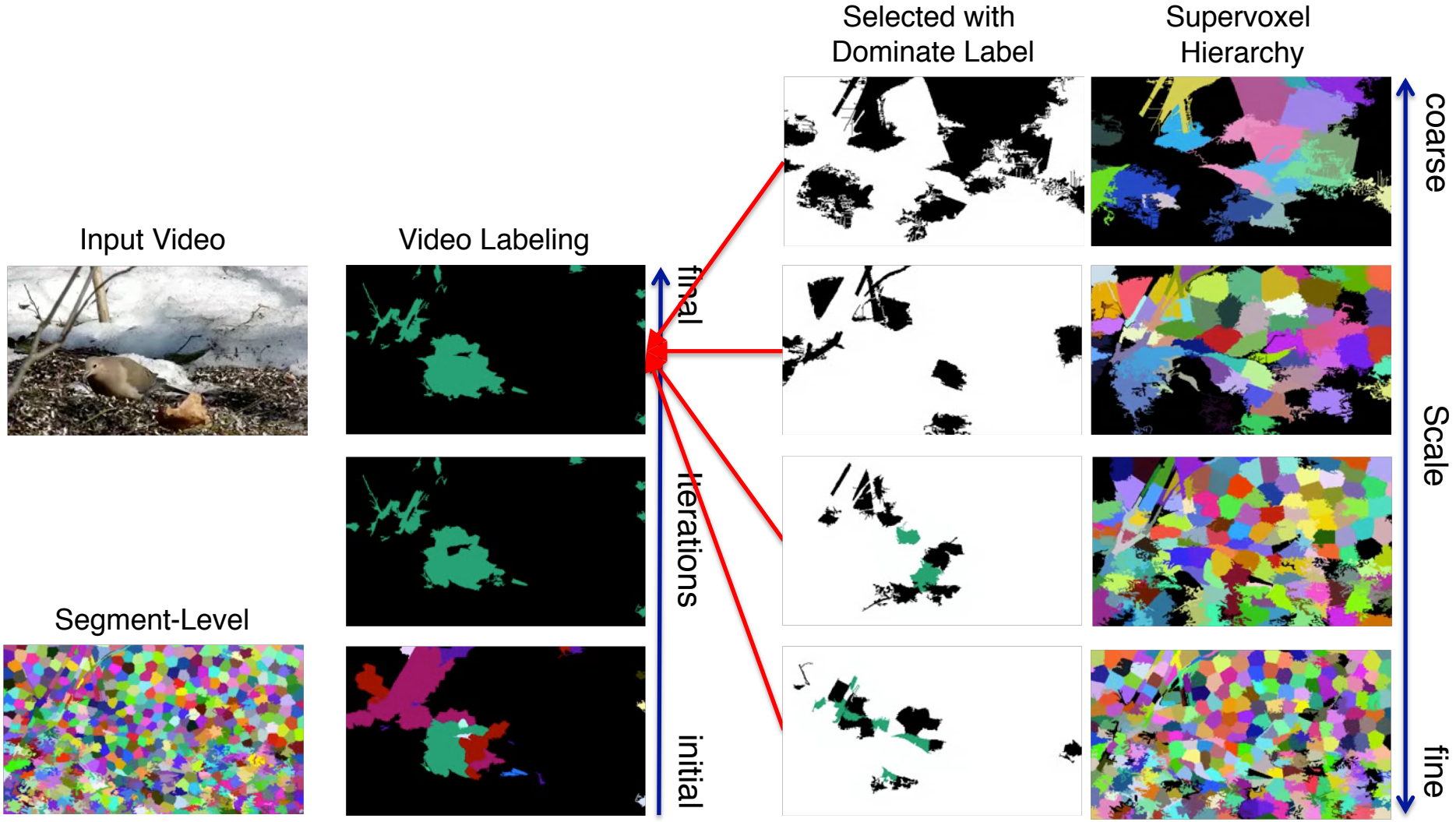
Results with GBH Hierarchy



[Xu et al., CVPR 2015, CVPR 2016]

Results with TSP Hierarchy

- Generalize to non-tree segmentation.



[Xu et al., CVPR 2015, CVPR 2016]

A2D: Actor-Action Dataset Statistics

actor \ action	climb	crawl	eat	fly	jump	roll	run	walk	none
adult	101	105	105		174	105	175	282	761
baby	104	106				107		113	36
ball				109	105	117			87
bird	99		105	106	102	107		112	26
car				102	107	104	120		99
cat	106		110		105	103	99	113	53
dog		109	107		104	104	110	176	46

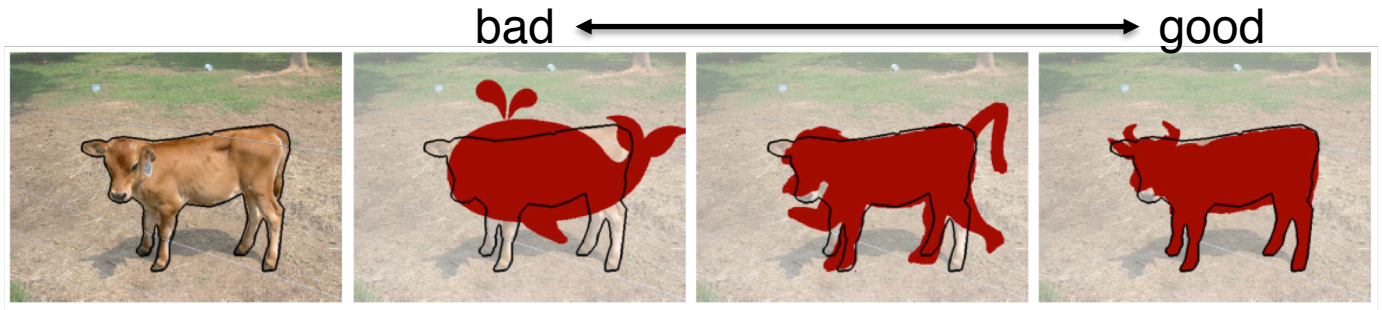
- The dataset consists of 3782 videos.
  - Average length: 136F; Minimum: 24F; Maximum: 332F.
  - One-third have more than one actor performing different actions.
  - We split the dataset into 80% training and 20% testing divided evenly over all actor-action tuples.
- The **first actor+action large dataset** in vision.

[Xu et al., CVPR 2015, CVPR 2016]


adult-climbing
adult-crawling
adult-eating
adult-jumping
adult-rolling
adult-running
adult-walking
baby-climbing
baby-crawling
baby-rolling
baby-walking
ball-flying
ball-jumping
ball-rolling
bird-climbing
bird-eating
bird-flying
bird-jumping
bird-rolling
bird-walking
car-flying
car-jumping
car-rolling
car-running
cat-climbing
cat-eating
cat-jumping
cat-rolling
cat-running
cat-walking
dog-crawling
dog-eating
dog-jumping
dog-rolling
dog-running
dog-walking
Chenliang

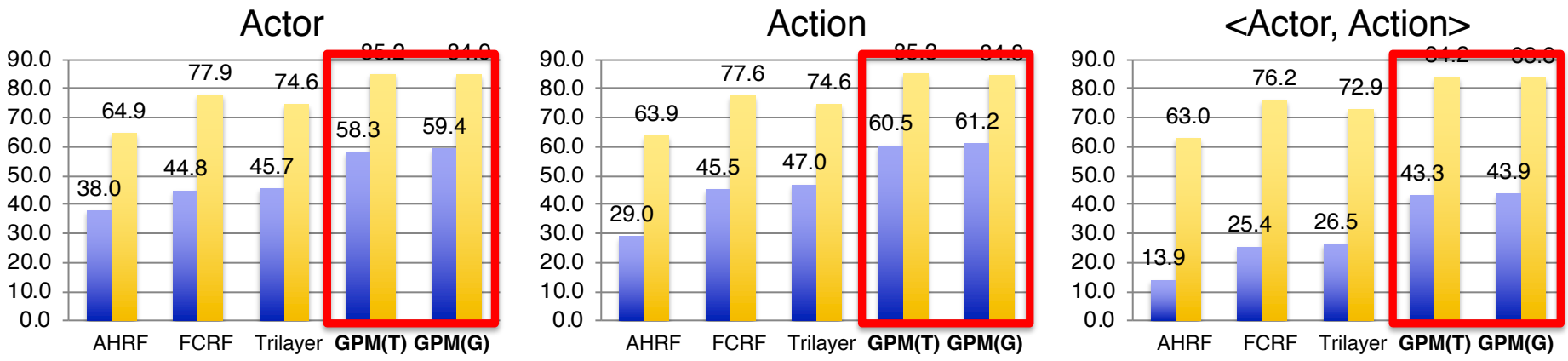
Experiments

- Evaluation metrics and tasks:



Pic Source: [Krahenbuhl and Koltun, ECCV'14]. We also care about instance class.

  - Three tasks: Actor, Action and <Actor, Action>.



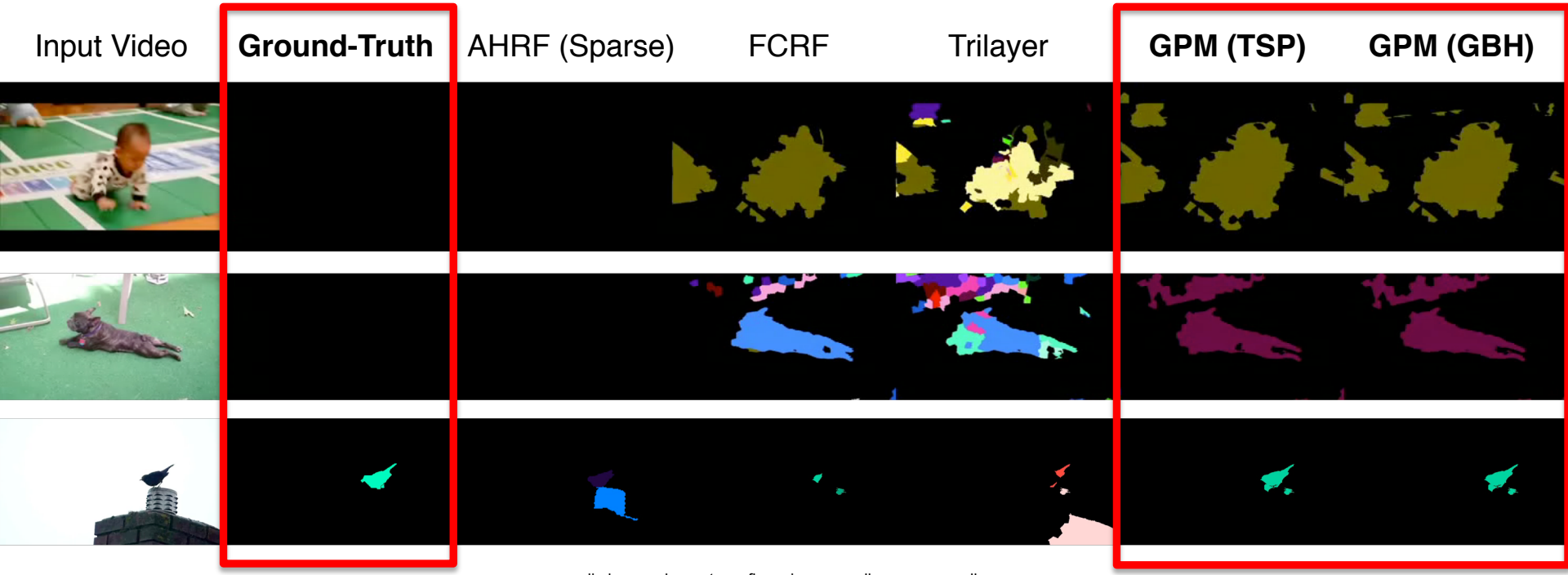
Blue: **Average Per-Class Accuracy**; Yellow: Global Pixel Accuracy

The higher the better.

[Xu et al., CVPR 2015, CVPR 2016]

Experiments

- Videos with single Actor-Acton label.

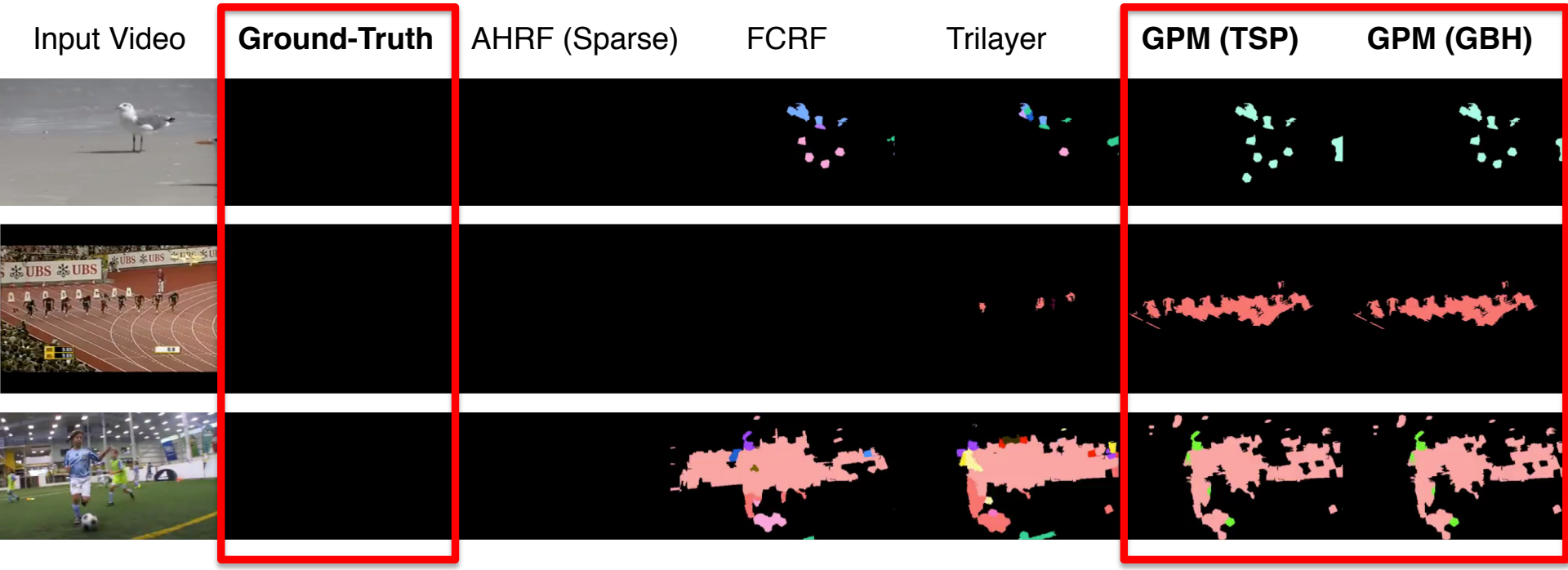


	climb	crawl	eat	fly	jump	roll	run	walk	none
adult	102	104	104		113	103	111	123	764
baby	104	103				106		109	54
ball				91	93	105			77
bird	99		102	103	102	107		109	30
car				103	108	102	111		82
cat	105		108		102	103	97	107	106
dog		106	107		102	103	106	168	128

[Xu et al., CVPR 2015, CVPR 2016]

Experiments

- Videos with multiple Actor-Action labels.



	climb	crawl	eat	fly	jump	roll	run	walk	none
adult	102	104	104		113	103	111	123	764
baby	104	103				106		109	54
ball				91	93	105			77
bird	99		102	103	102	107		109	30
car				103	108	102	111		82
cat	105		108		102	103	97	107	106
dog		106	107		102	103	106	168	128

[Xu et al., CVPR 2015, CVPR 2016]

Recap

- What are grouping problems in vision?

- Inspiration from human perception
  - Gestalt properties

- Bottom-up segmentation via clustering
  - Mode finding and mean shift: k-means, SLIC, mean-shift
  - Graph-based: minimum spanning forest, normalized cuts