Research at Google

Georgia Tech College of Computing

Graph-Based Hierarchical Video Segmentation

Matthias Grundmann,
Vivek Kwatra, Mei Han

Daniel Castro, Irfan Essa

Google Research

Georgia Institute of Technology

Video Segmentation

- Spatio-temporal regions: Group appearance and motion in space and time
- Application: Selecting regions ⇒ rapid annotation

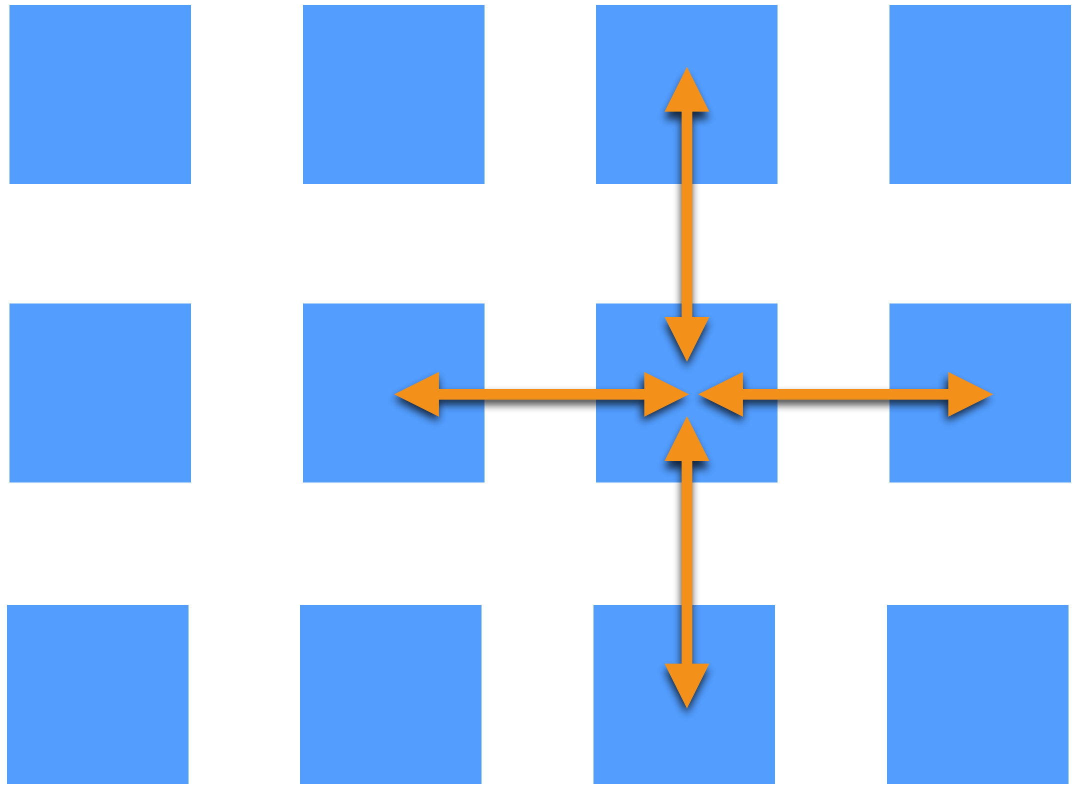
region color indicates region identity

Talk outline

- Graph-based segmentation in the video domain
- Segmentation approaches / agglomerative clustering
- Over-segmentation
- Hierarchical segmentation
- Based on [Grundmann et al. 2010]: Efficient graph-based hierarchical video segmentation
  with many improvements
- Streaming segmentation (next talk)

Graph-based segmentation

- Grid graph over image domain
- Connectedness: N4 or N8
- Affinity between pixels:
  - Color distance
  - Weighted with gradients
  - Take into account optical flow
  - From per pixel classifiers, etc.



Research at Google

Georgia Tech College of Computing

Extending to Video Domain

- Direct application of image-based algorithm per frame
- Lacking temporal coherence
- Unstable boundaries in time
    - Associating 2D regions will yield noisy outcome

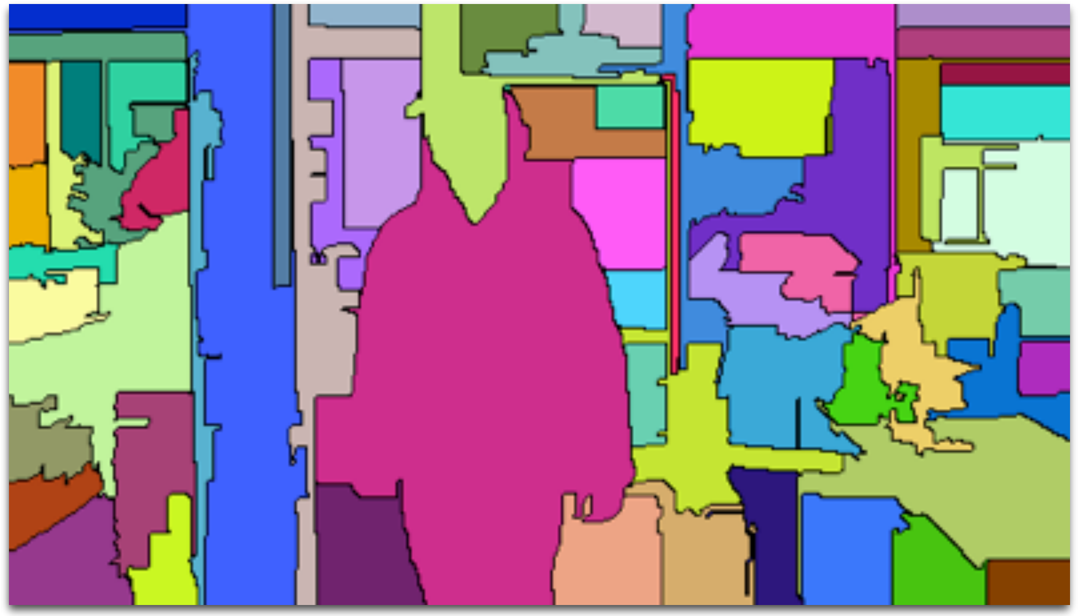
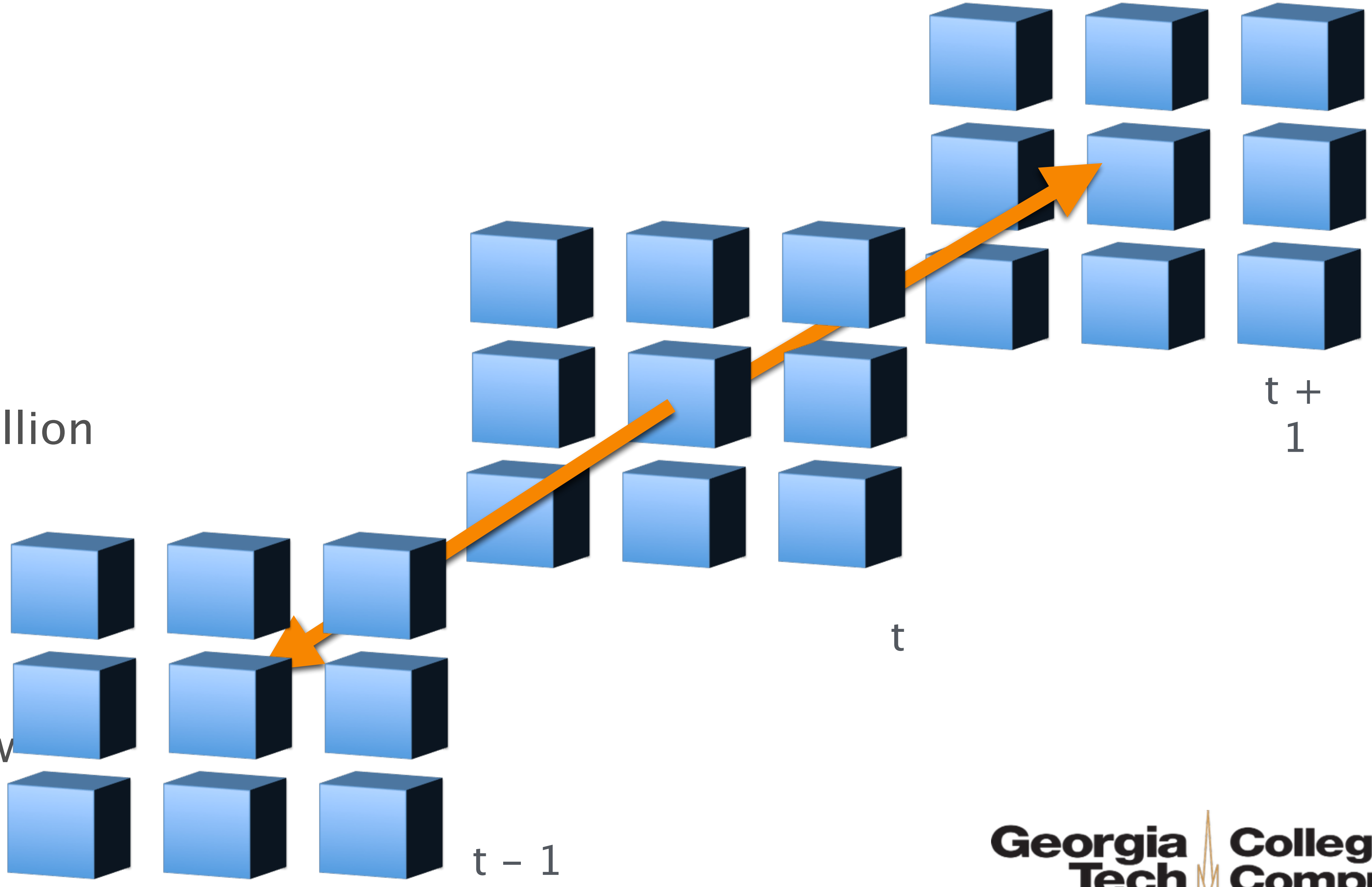


image segmentation
applied to each frame

Extending to Video Domain

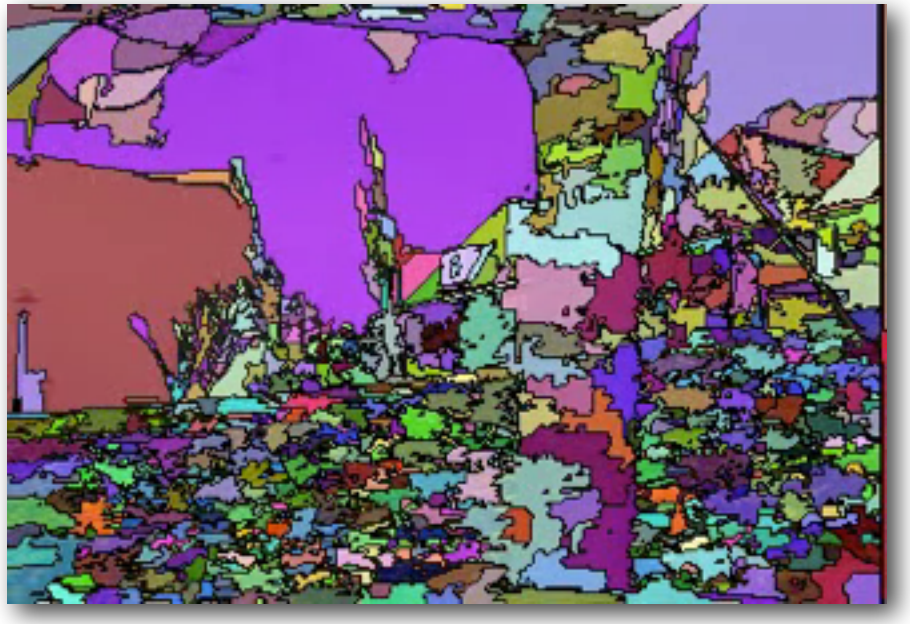
- Extend N8 graph in time: Spatio-Temporal volume
- Connect each pixel to also to its 9 neighbors in time (forward / backward)
- Connectedness: N26
  - 1 sec of 360p video: 90 million edges
  - vs. 1 million for image case
- How to connect?
  - Direct predecessor
  - Displaced along optical flow



Connection in time

- Direct predecessor can't model movements > 1 pixel
- Displace connection in time along dense optical flow

dense flow, hue encodes angle



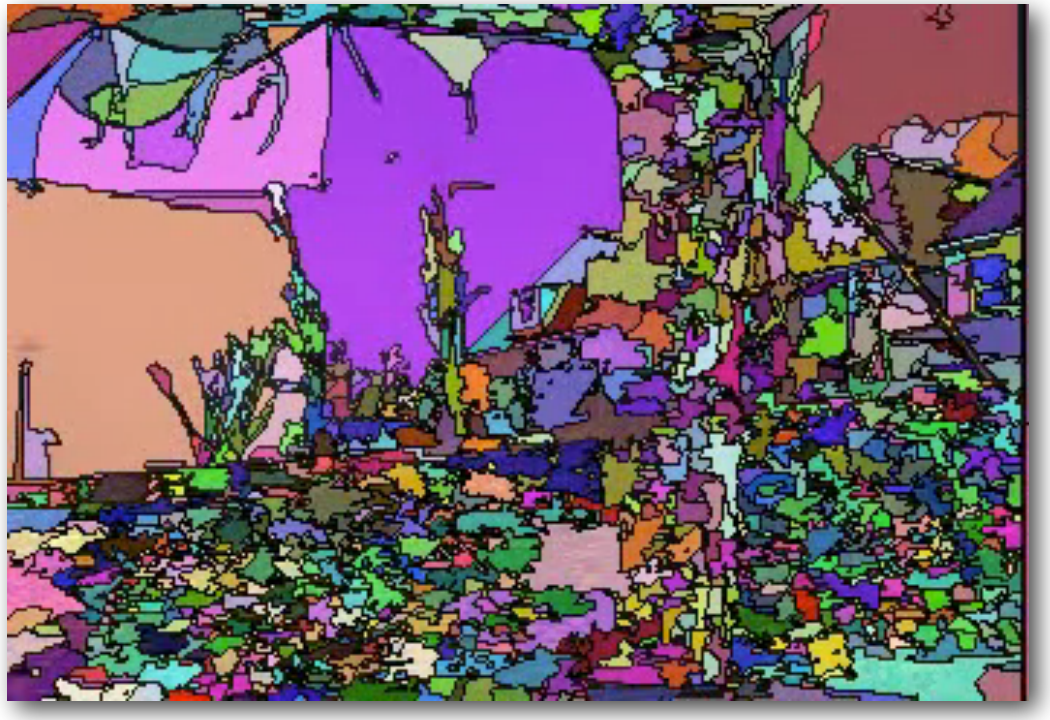
oversegmentation using
direct
predecessor in volume

Research at Google

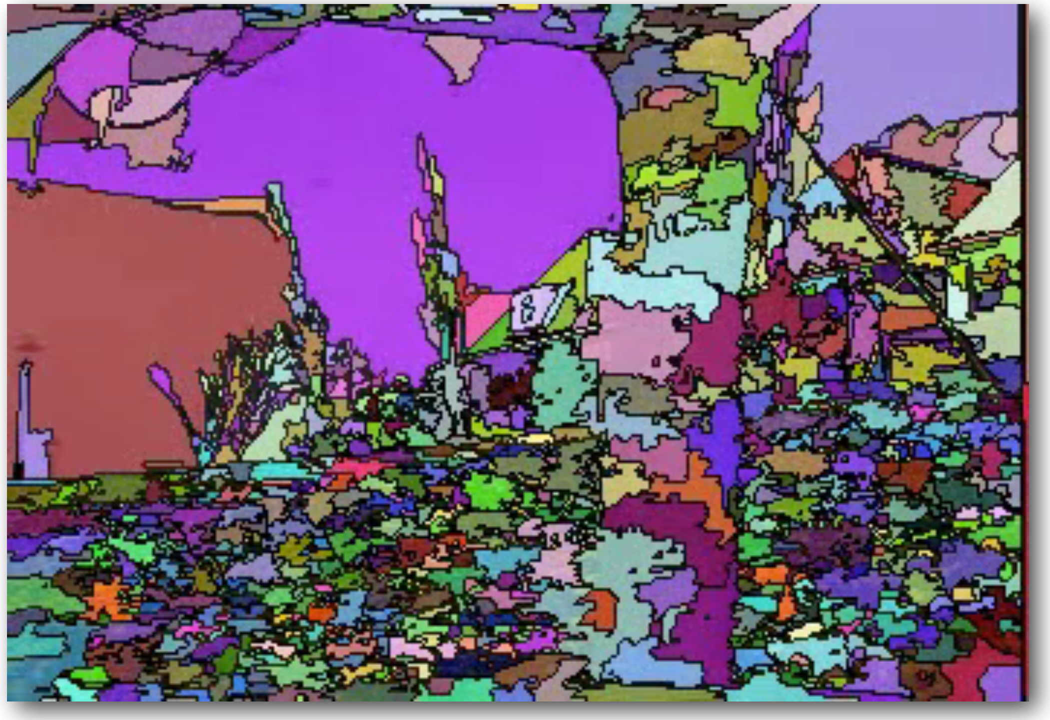
Georgia Tech College of Computing

Connection using dense optical flow

- Displace temporal connection along dense optical flow



oversegmentation using predecessor along dense flow



oversegmentation using direct predecessor in volume

Research at Google

Georgia Tech College of Computing

Connection using dense optical flow

- Displace temporal connection along dense optical flow

oversegmentation using predecessor along dense flow

oversegmentation using direct predecessor in volume

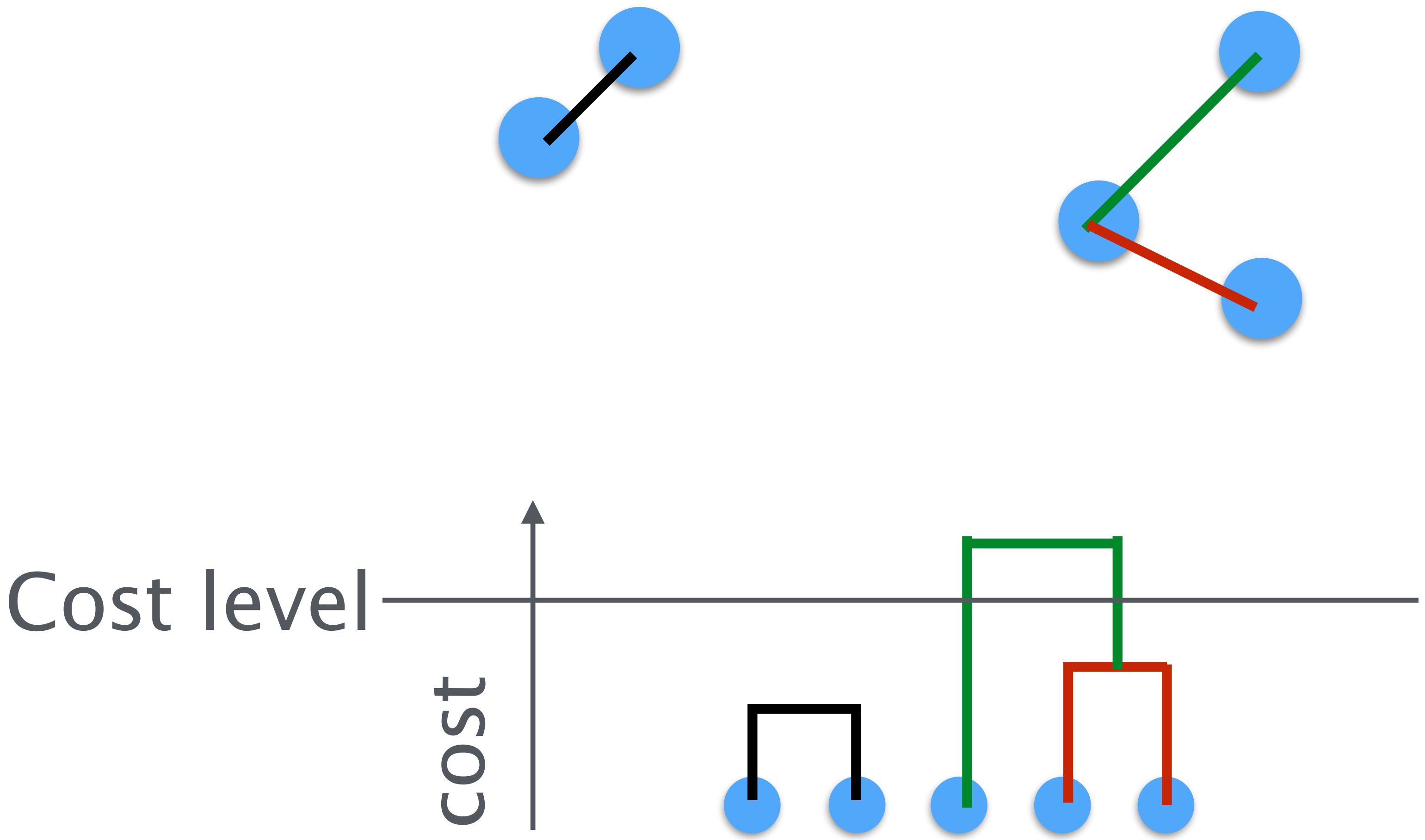
Why Graph-based segmentation

- Need:
  - Low-complexity segmentation algorithm
  - Algorithm that we can constrain (later: for streaming segmentation)
  - Initialization free (i.e. no prior user interaction or parameters, e.g. Snakes, GrabCut)

  - Mean-Shift [Comaniciu and Meer, 2002]
  - Normalized cuts [Shi and Malik, 1997]
  - k-Means, EM / Mixture of Gaussians [Bishop 2006]
  - SLIC [Achanta et al. 2012]
  - Watersheds
  - Turbo Pixels [Levinshtein et al. 2009]
  - Greedy Graph-Based [Felzenszwalb and Huttenlocher 2004]

Agglomerative clustering

- Simplest type of clustering:
- Put every item in a single cluster
- Define distance between clusters
- Iteratively merge the two closest one
- Merge sequence represented by dendrogram
- Segmentation result: Threshold at cost level (not necessarily uniform) or number of regions



Research at Google

Georgia Tech College of Computing

Agglomerative clustering

- How to define the cluster distance between cluster C_1 and C_2?
- Basically 3 types:
  - Single-link

$$\min_{a \in C_1, b \in C_2} ||d(a) - d(b)||$$

  - Complete-link

$$\max_{a \in C_1, b \in C_2} ||d(a) - d(b)||$$

  - Average-link
    (N = total number of summands)

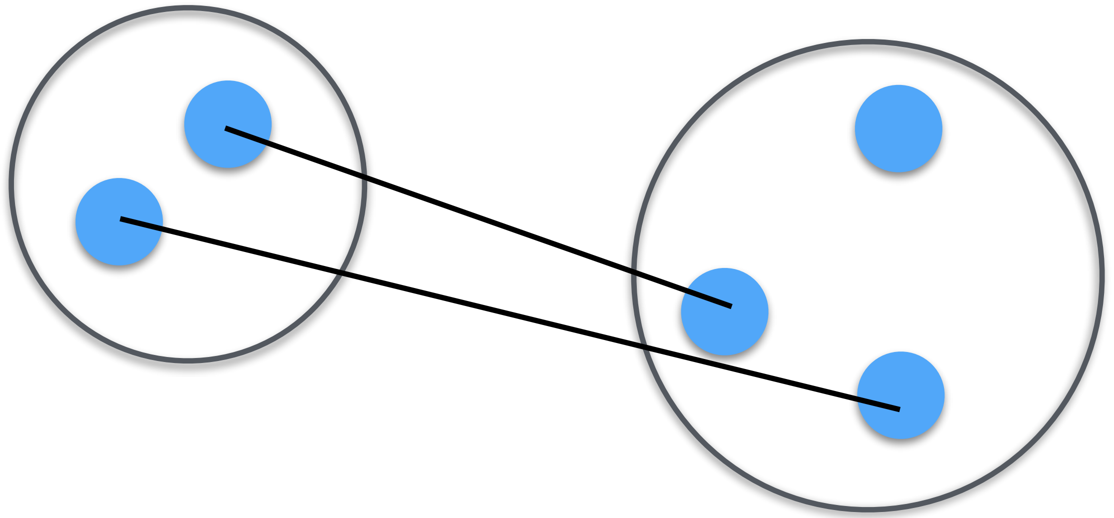
$$\frac{1}{N} \sum_{a \in C_1 \cup C_2} \sum_{b \neq a \in C_1 \cup C_2, b} ||d(a) - d(b)||$$

Research at Google

Georgia Tech College of Computing

Agglomerative clustering

- Single link:
  - Distance between closest two elements
- Complete link:
  - Distance between two furthest elements
- Average link:
  - Average distance between all elements (not drawn)
- Conclusion:
  - Only single link merges do not alter cluster distance!
  - 1 sec of 360p video: 90 million edges



Single link agglomerative clustering

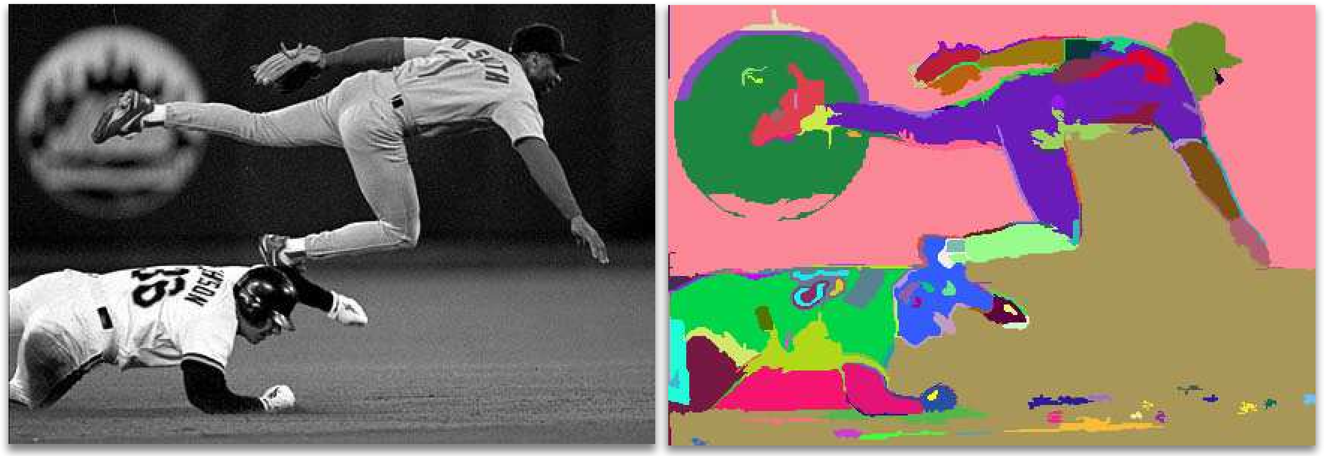
- Complexity:
  - Sort the edges between original nodes O(n log n)
  - Traverse in order O(n)
  - Merges via union–find / disjoint forest
    - Union by rank
    - Path compression
    - Total complexity: O(n α(n)), α : inv. Ackermann, α <= 5 in practice
  - Read result: O(n α(n))
- Total complexity: O(n log n)   Dominated by sort

Efficient graph based image segmentation

- [Felzenszwalb and Huttenlocher 2004]
- Single link agglomerative clustering
  - Cluster distance: Diff. pixel appearance
  - $\mathrm{Int}(C_i)$: last edge weight for each cluster (height from dendrogram)
  - Termination criteria:

$$\min_{a \in C_1, b \in C_2} ||d(a) - d(b)|| = \mathrm{Int}(C_1 \cup C_2) > \min(\mathrm{Int}(C_1) + \tau(C_1), \mathrm{Int}(C_2) + \tau(C_2))$$

  - $\tau(C)$ = constant / |C|



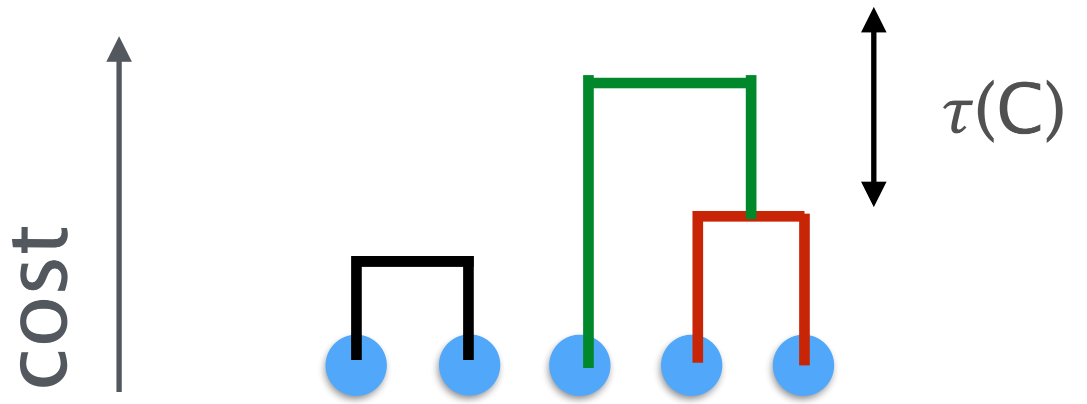
from [Felzenszwalb and Huttenlocher 2004]

Efficient graph based image segmentation

- Termination criteria

$$\mathrm{Int}(C_1 \cup C_2) >$$
$$\min(\mathrm{Int}(C_1) + \tau(C_1), \mathrm{Int}(C_2) + \tau(C_2))$$

- Int(C) : dendrogram height, τ(C) = constant / |C|
- Relative test, space decreases with region size



Efficient graph based image segmentation

- What to take away:
  - [Felzenszwalb and Huttenlocher 2004] is single link agglomerative clustering
  - “Local” termination criteria w.r.t. dendrogram spacing
  - Monotonic criteria: Once violated, the two clusters won’t be merged
  - **Also:** Any other monotonic criteria will do

Efficient graph based video segmentation

- Applying the "Local" termination criteria to video is problematic
  - τ(C) = constant / |C| decreases with region size
- For video:
  - In video region volume >> region area for images
  - Either increase constant (more segmentation errors)
  - Or: Have many small regions
- For practical implementations:
  - For large homogenous regions: $\tau(C) \rightarrow 0$
    ⇒ Regions are broken into small pieces
  - For textured regions: Additional merges required to achieve minimum region size

Research at Google

Georgia Tech College of Computing

Homogenous regions

$$\tau(C) \rightarrow 0$$

Research at Google

Georgia Tech College of Computing

Introducing additional merges

- Forced merges: Merge everything with edge weight < 1 intensity / compression level
- Regular merges: [Felzenszwalb and Huttenlocher 2004] local criteria
- Small region merges: also [Felzenszwalb and Huttenlocher 2004]

with forced merges

without forced merges

Research at Google

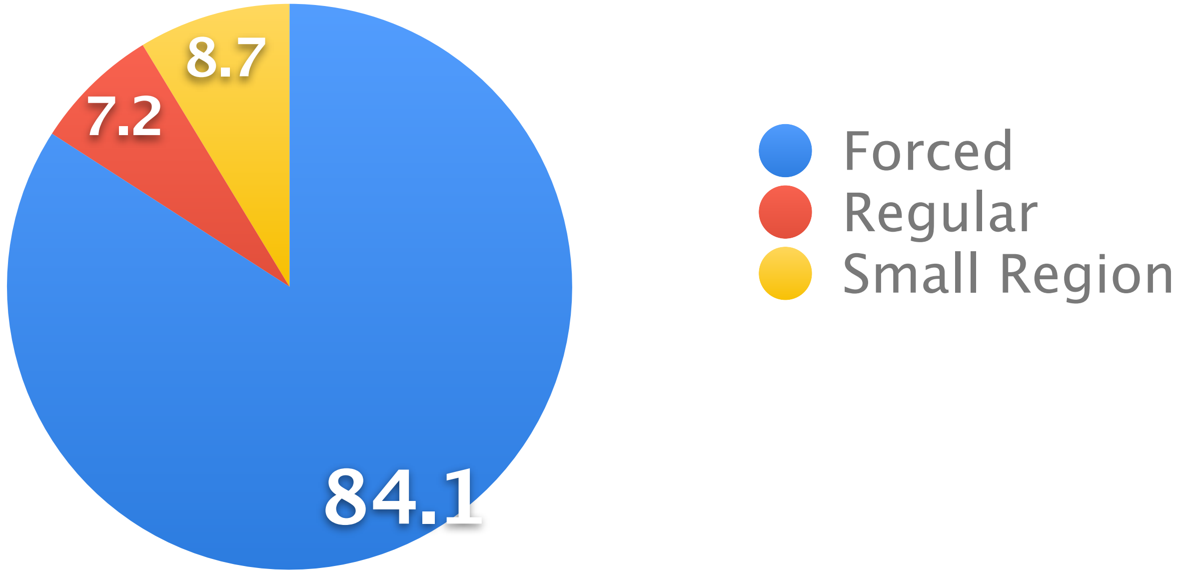
Results use new merge criteria, not [Felzenszwalb and Huttenlocher 2004]

Georgia Tech College of Computing

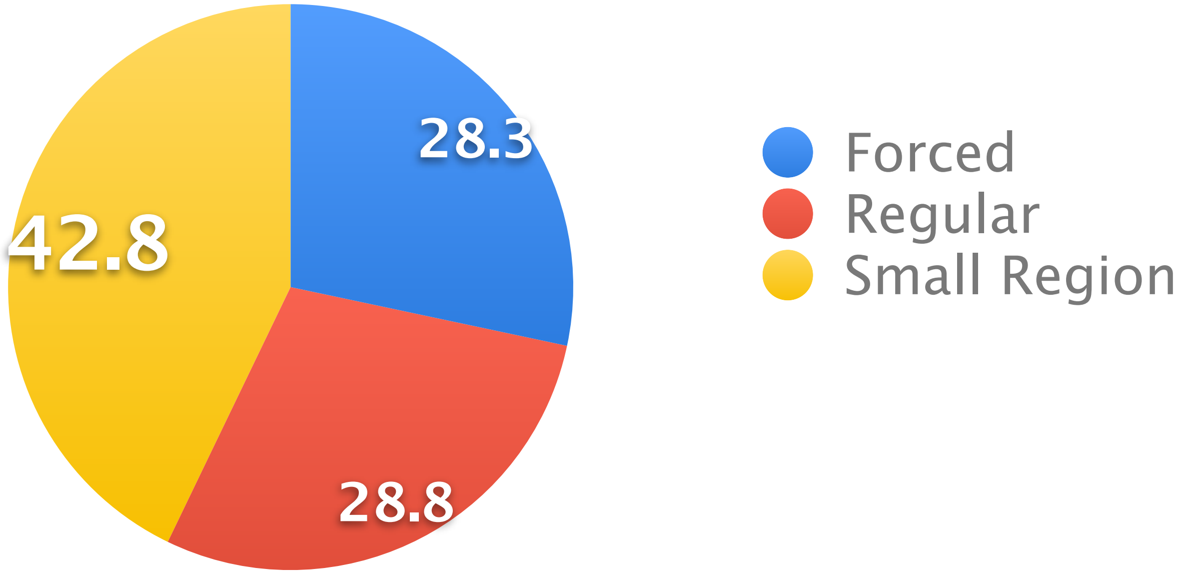
Merge percentages

- [Felzenszwalb and Huttenlocher 2004] with forced merges
- Regular merges account for less than 1/3 of all merges

Truck
(homogenous)



Flowergarden
(textured)



forced includes merges due to constraints

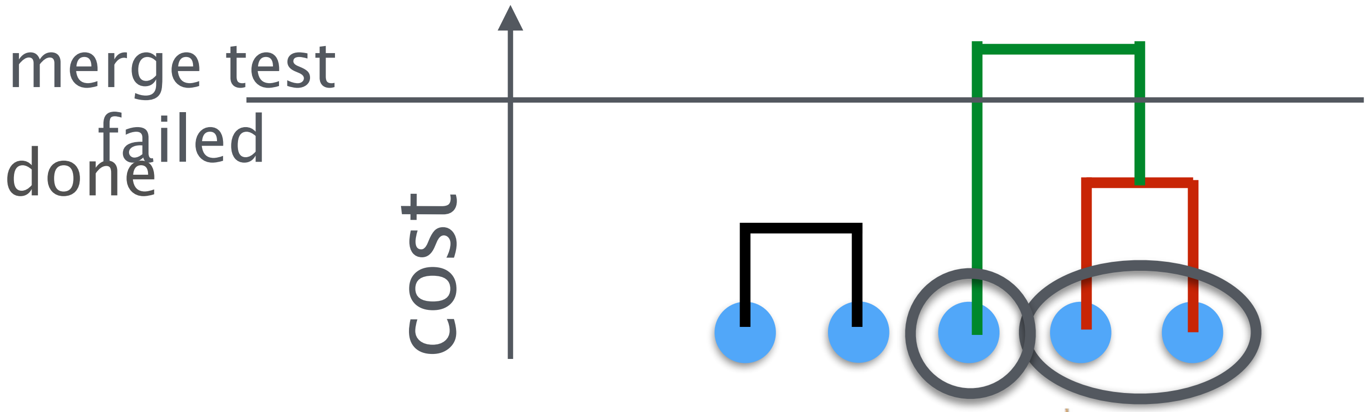
Talk outline

- Graph-based segmentation in the video domain
- Segmentation approaches / agglomerative clustering
- Over-segmentation
- Hierarchical segmentation

A new merge criteria

- Recall: **Any** monotonic criteria will do
- Need more regular merges, distance that accounts for compression levels
- Avoid “chaining” for single link clustering (small local edge weights can accumulate)

- Idea:
  - Build up local descriptors during merge process
  - Use edge and descriptor distance to determine if a merge should be performed
  - Incorporate small region merges
  - Monotonicity: If merge test fails, label regions as done

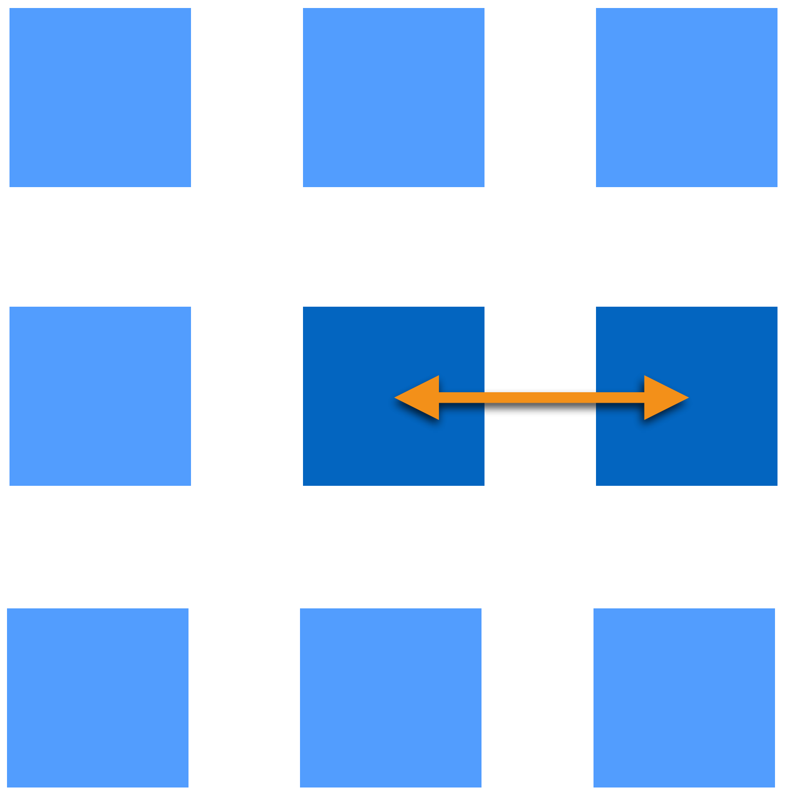


Research at Google

Georgia Tech College of Computing

Our new merge criteria

- Descriptor during merges:
  Mean color / Mean flow (any other possible)
- Merge regions if:
  - Edge weight < 1 intensity level and
    descriptor distance < 20%
    (allow for variability but control cutoff)
  - Edge weight >= 1 intensity level and
    descriptor distance < 5% intensity range
  - One of them is too small
- If violated: Flag as done (monotonicity!)



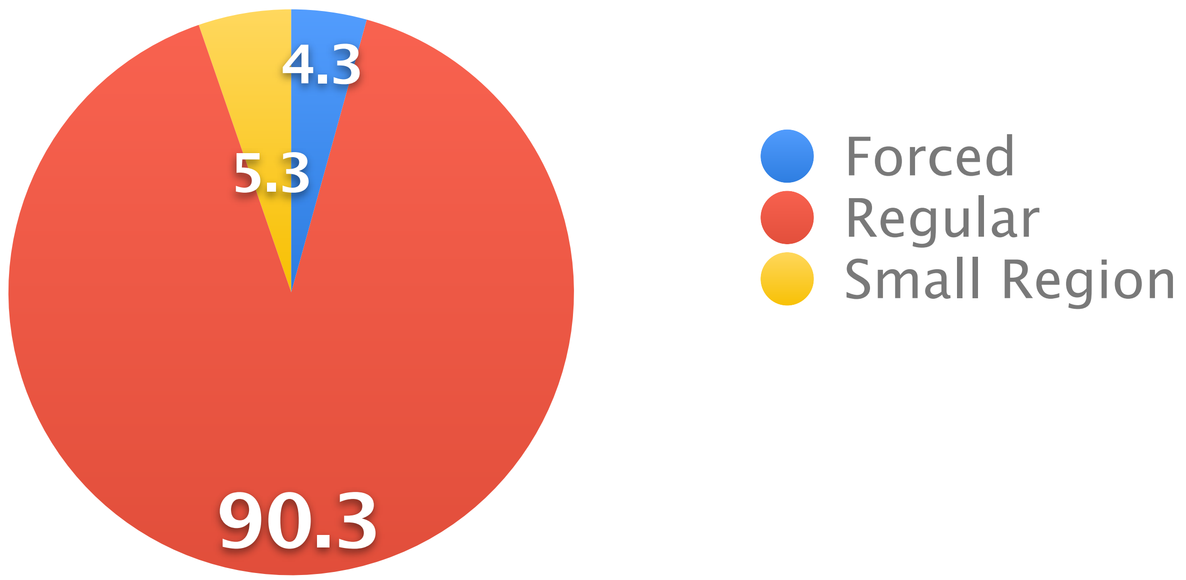
Research
at Google

Georgia Tech College of Computing

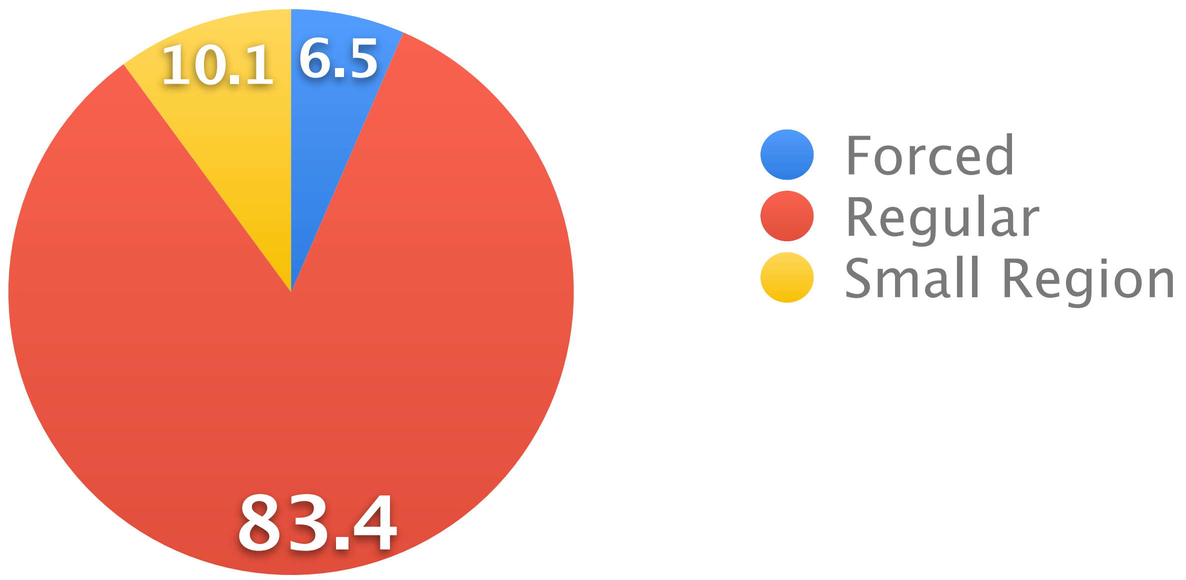
Merge percentages for new criteria

- Regular merges account for more than 80% of all merges!

Truck
(homogenous)



Flowergarden
(textured)



Research at Google

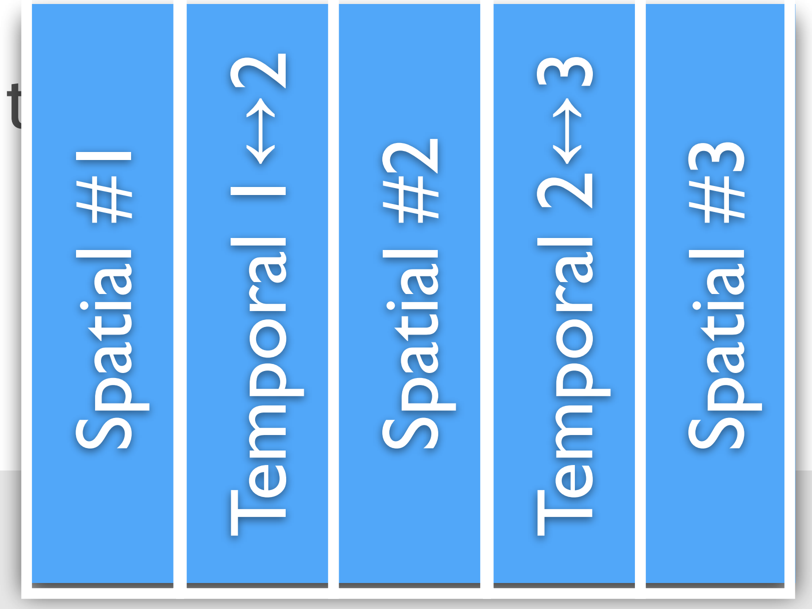
forced includes merges due to constraints

Georgia Tech College of Computing

Fast O(n) segmentation

- Single link agglomerative clustering: Total complexity: O(n log n)   Dominated by sort
- Idea: Skip the sort
    - Discretize edge weight domain into 2-4K buckets (bucket sort)
      L1 RGB color distance: 768 values
- **Complexity: O(n)** [no large multipliers, $\alpha(n) < 5$ for all practical values of N]
- Can we do better?
    - Observation: Edge evaluation is costly / Spatial and temporal edges are disjoint
- Bucket lists
    - For N frames use 2 * N – 1 list of 2K buckets
    - Create in parallel via on-demand threads!

31% faster!!



Parallel construction

Research at Google

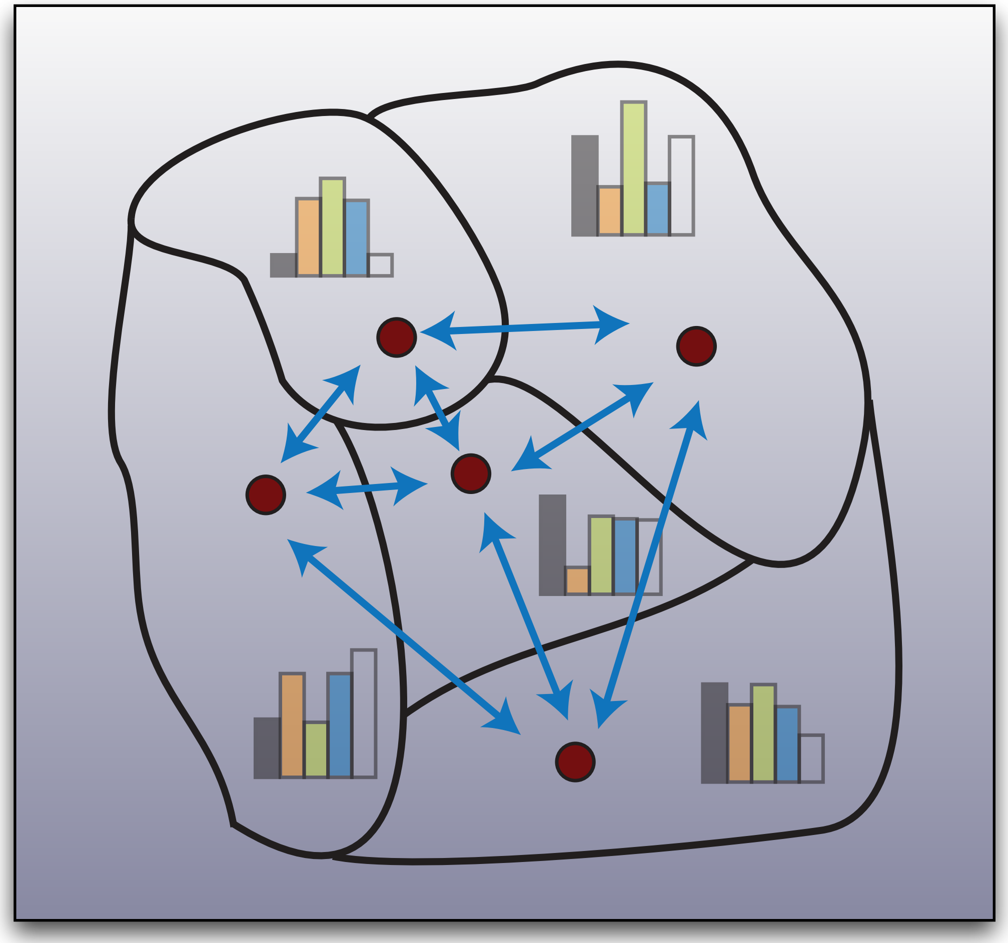
Georgia Tech College of Computing

Talk outline

- Graph-based segmentation in the video domain
- Segmentation approaches / agglomerative clustering
- Over-segmentation
- Hierarchical segmentation

Hierarchical graph-based segmentation

- Size of regions: Controlled by merge threshold between descriptors (earlier: τ(C))
- **Hierarchical segmentation**: Instead of tweaking thresholds
- Build spatio-temporal adjacency graph of regions from over-segmentation
- Edge weights based on similarity of region descriptors (Appearance, texture, motion)
- Segment regions in super-regions
- Repeat until: Minimum region number reached



Research at Google

Georgia Tech College of Computing

Hierarchical segmentation

- Descriptors (3):
  - LAB histogram (10 * 16 * 16) w/ interpolation
  - Flow histogram (20 angles)
  - Compare each via χ_2 distance
  - Region Size Penalizer (truncated ratio w.r.t. average region size)
  - Combine via Soft-OR distance times region penalizer γ :

$$\gamma[1 - \prod_i (1 - d_i)]$$

- Merge:
  - Merge each descriptor / histogram
- Important: Merge alters edge weights between clusters (like average-link clustering)

Research at Google

Georgia Tech College of Computing

Hierarchical segmentation

- Which algorithm to use?
- First version: Uses single link [Felzenszwalb and Huttenlocher 2004]
  - Note: Merges alter edge weights!
  - Only allows for one merge per region per hierarchy level
  - Bad control for number of merges per hierarchy level
- Current version: True average-link agglomerative clustering + specify percentage of merges
- Fast: O(n) bucket sort for edges
- At every merge (< n) : Update neighboring edges in parallel
- Total complexity: O(n * k), k < 100 for our purposes

Spatio-Temporal Over-Segmentation

original video

over-segmentation

Research at Google

Georgia Tech College of Computing

Hierarchical Segmentation

Over-segmentation

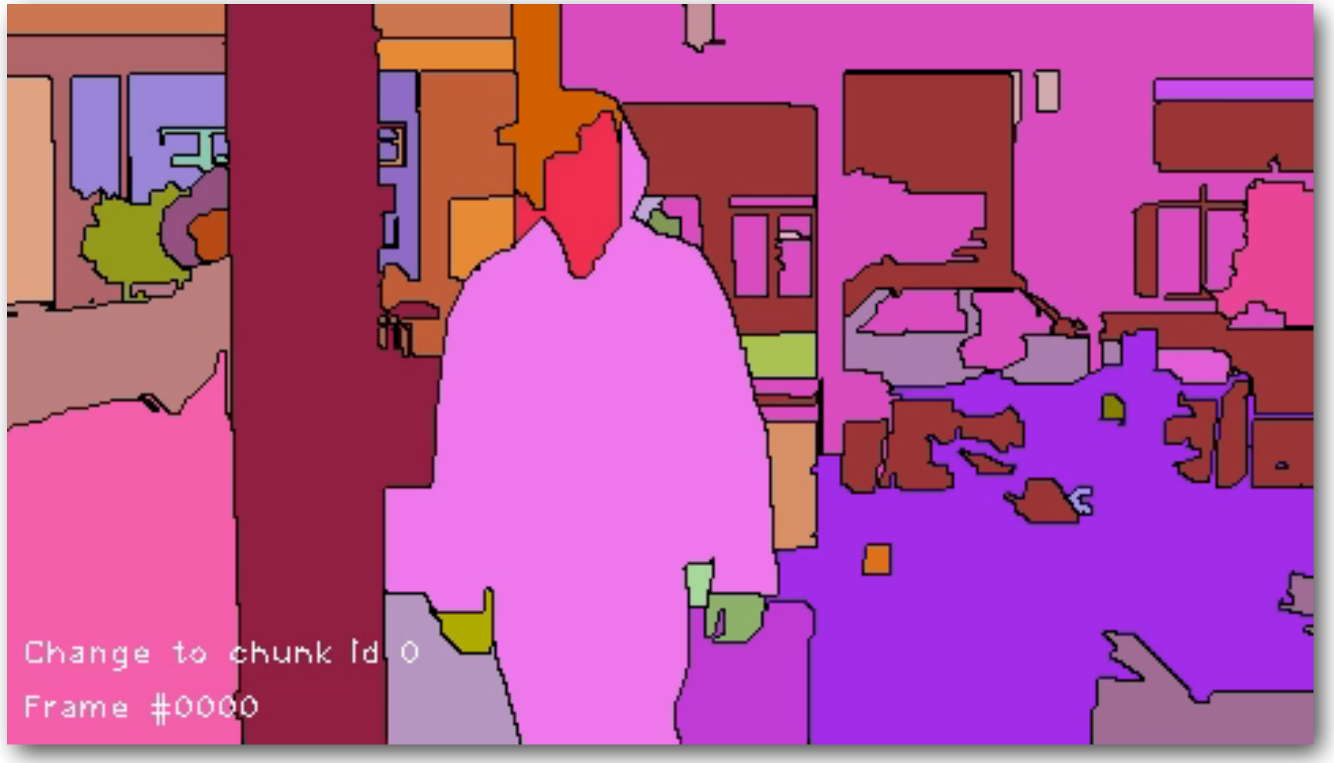
Hierarchy at 20%

Research at Google

Georgia Tech College of Computing

Hierarchical Segmentation

Hierarchy at 20%



Hierarchy at 50%

Benefits of hierarchical segmentation

Note: instability in over-segmentation (identities of region change [lights, window], boundaries are more unstable)

Hierarchical segmentation (shown at 50% of height of segmentation tree)

Over-segmentation only (manually tuned to give similar sized regions)

Research at Google

Georgia Tech College of Computing

Benefits of hierarchical segmentation

Hierarchical segmentation

Over-segmentation only
(manually tuned to give similar
sized regions)

Research
at Google

Georgia Tech College of Computing

Effect of flow as feature

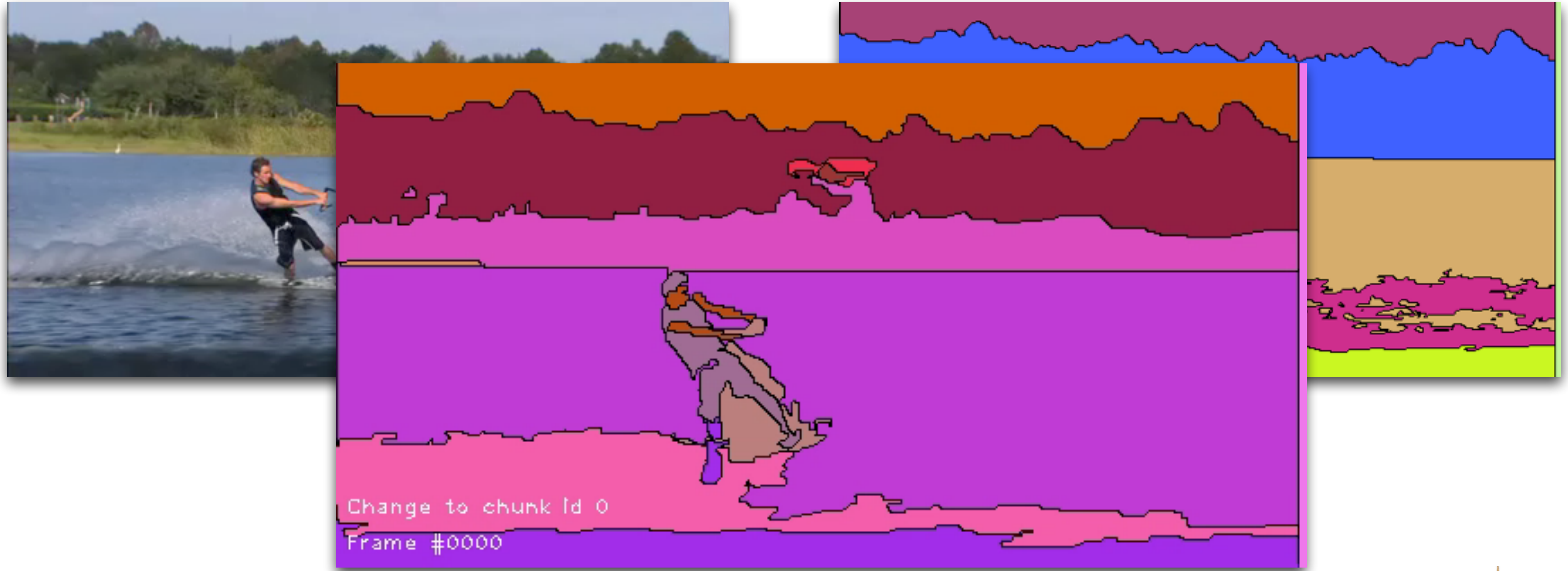
original

no flow

flow in hierarchical segmentation

flow in over-segmentation & flow in hierarchical segmentation

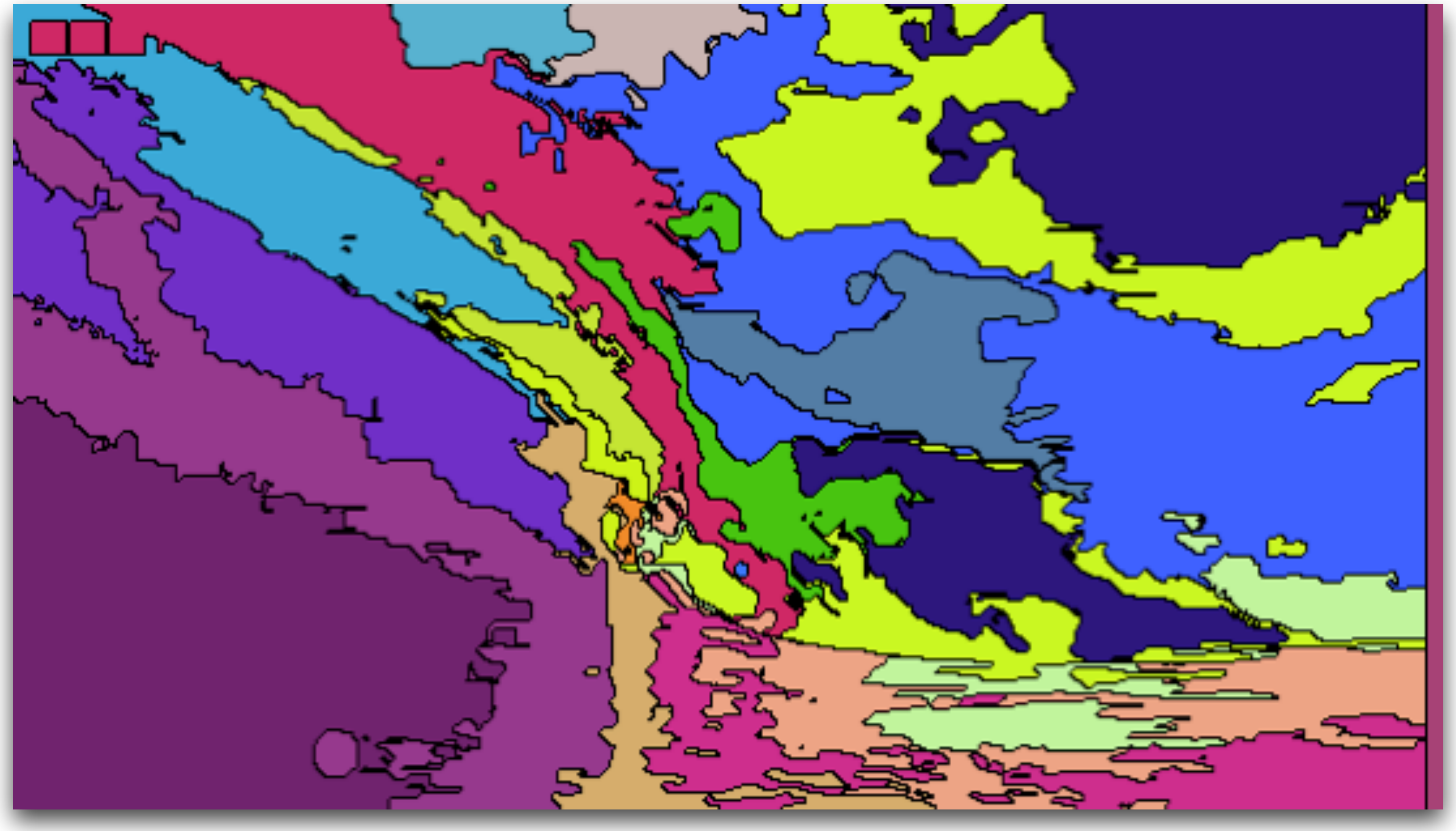
Results



Research at Google

Georgia Tech College of Computing

Results



Google Developers

Georgia Tech | College of Computing

Research at Google

Georgia Tech College of Computing

Applications of Video Segmentation

Matthias Grundmann, Vivek Kwatra, Mei Han

Daniel Castro, Irfan Essa

Google Research

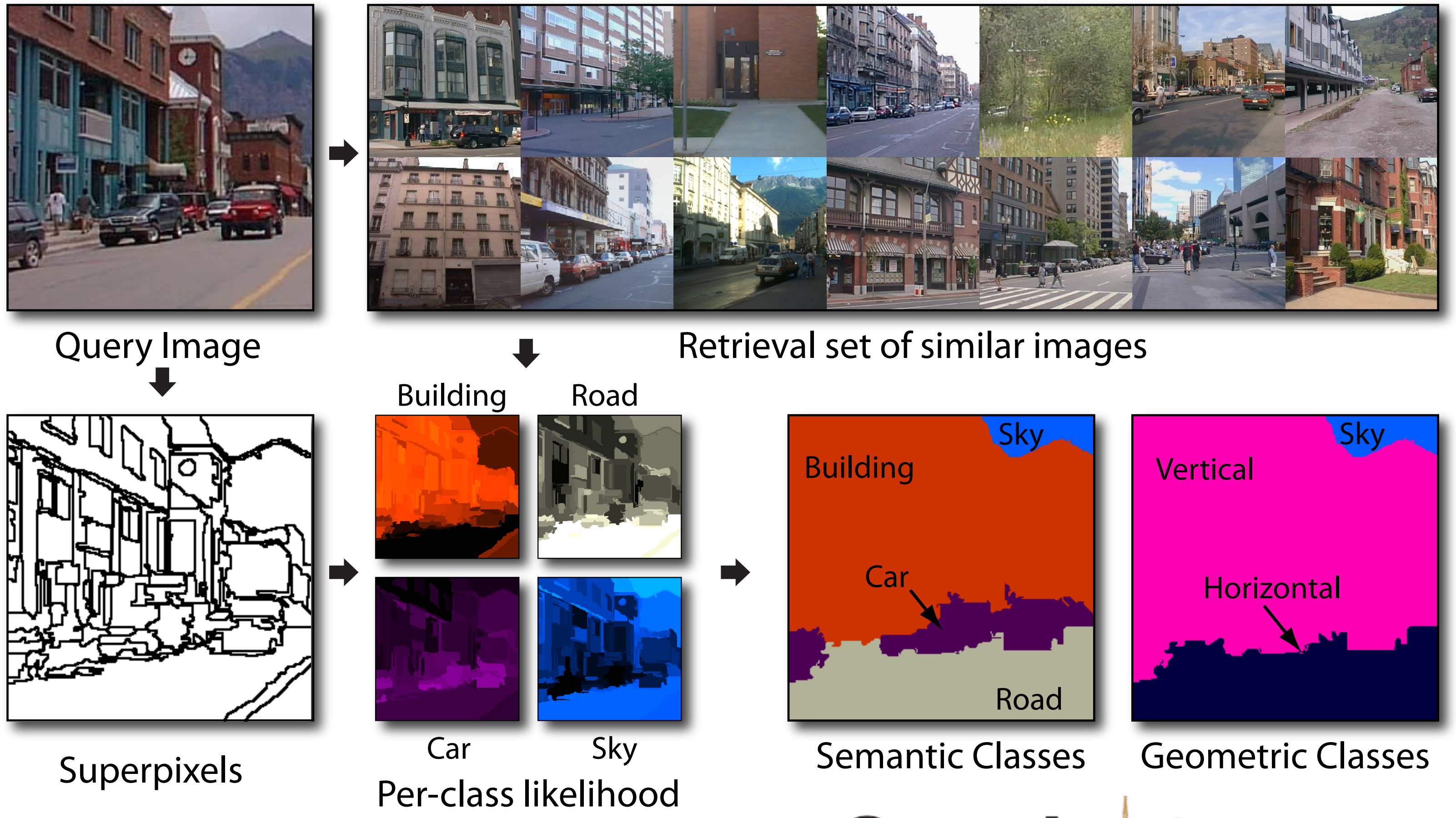
Georgia Institute of Technology

Talk outline

- Applications of video segmentation
  - Super-Parsing
  - Geometric context
  - Radiometric calibration for segmentation
  - Weakly supervised segmentation
- Online video segmentation and annotation
- Open source video segmentation

Super-Parsing

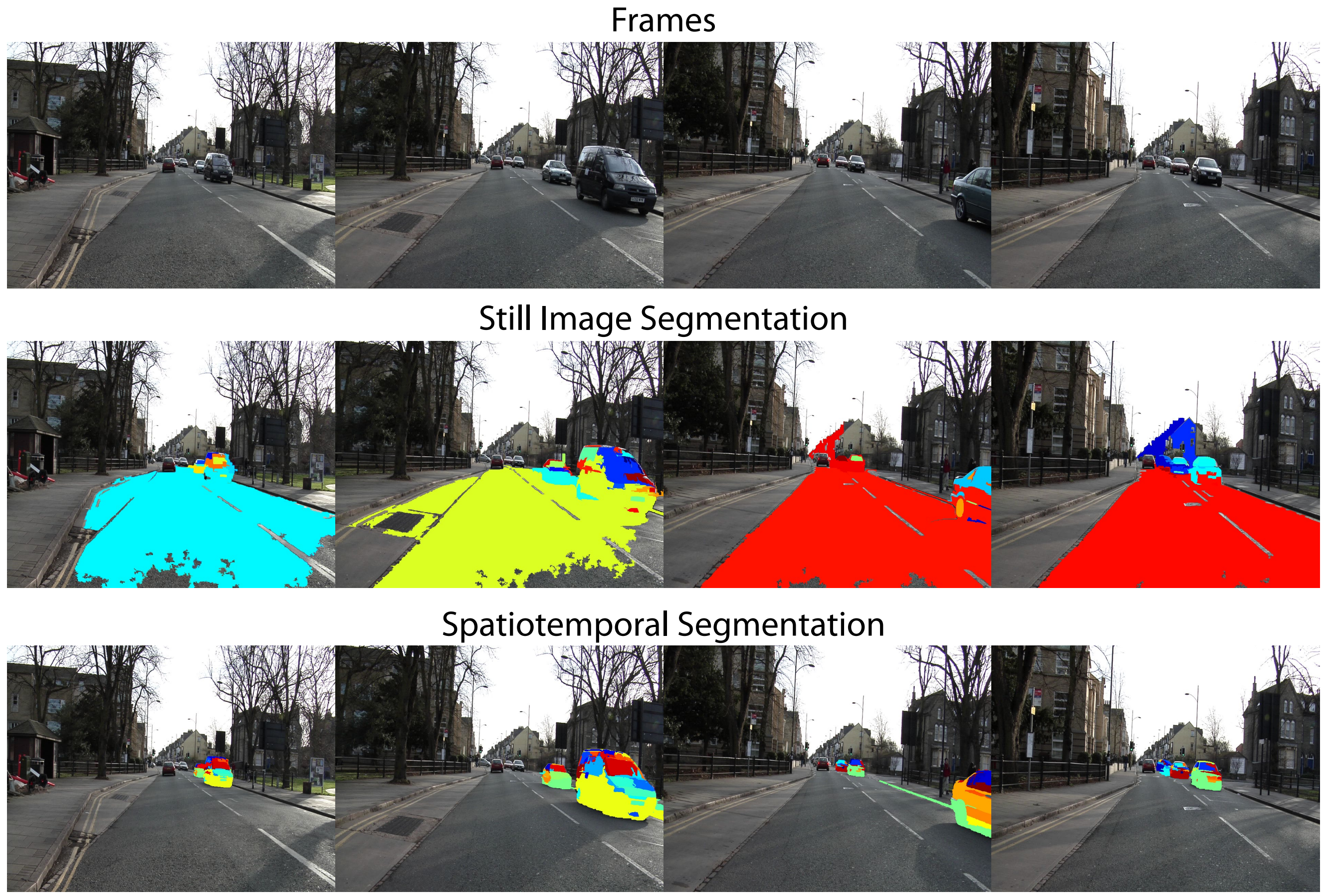
- [Joseph Tighe and Svetlana Lazebnik, 2012]:
  SuperParsing: Scalable Nonparametric Image Parsing with Superpixels
- Simplified description:
  - Extract super pixels from query image
  - Label transfer: From labeled super-pixel in training set
  - Smoothing via MRF



- http://www.cs.unc.edu/~jtighe/Papers/ECCV10/index.html

Super-Parsing

- [Joseph Tighe and Svetlana Lazebnik, 2012]:
  SuperParsing: Scalable Nonparametric Image Parsing with Superpixels
- Extended to video:
  - Super pixels → Spatio-Temporal regions
  - Aggregate prediction score over segments
  - MRF smoothing over super-voxels



Research at Google

Georgia Tech College of Computing

SuperParsing: Scalable Nonparametric Image Parsing with Superpixels

Joseph Tighe and Svetlana Lazebnik

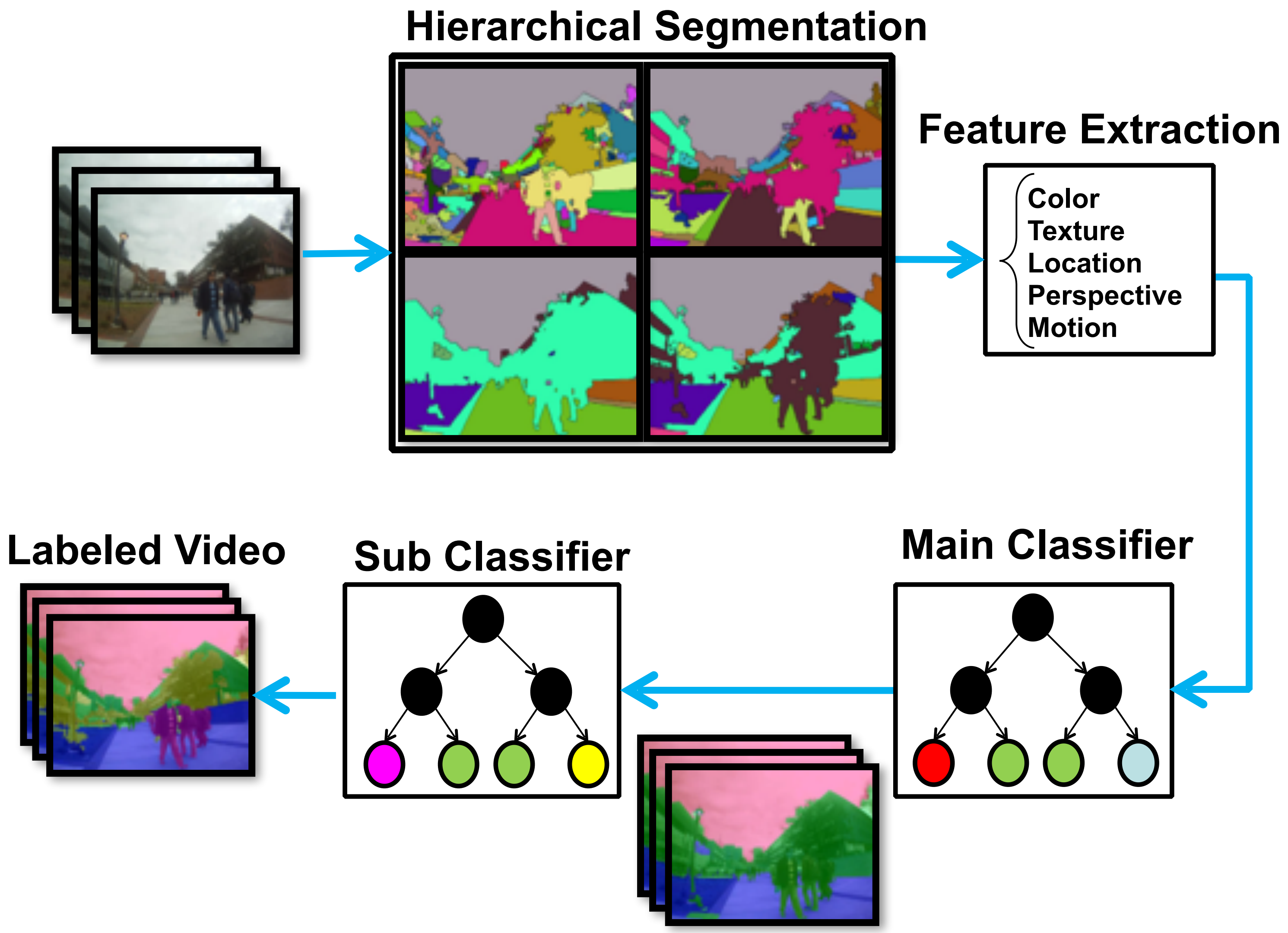
Geometric Context from Video

- Hoiem, Efros, Hebert, "Geometric Context from a Single Image", ICCV 2005
- Hussein, Grundmann, Essa, “Geometric Context from Video, CVPR 2013

http://www.cc.gatech.edu/cpl/projects/videogeometriccontext/

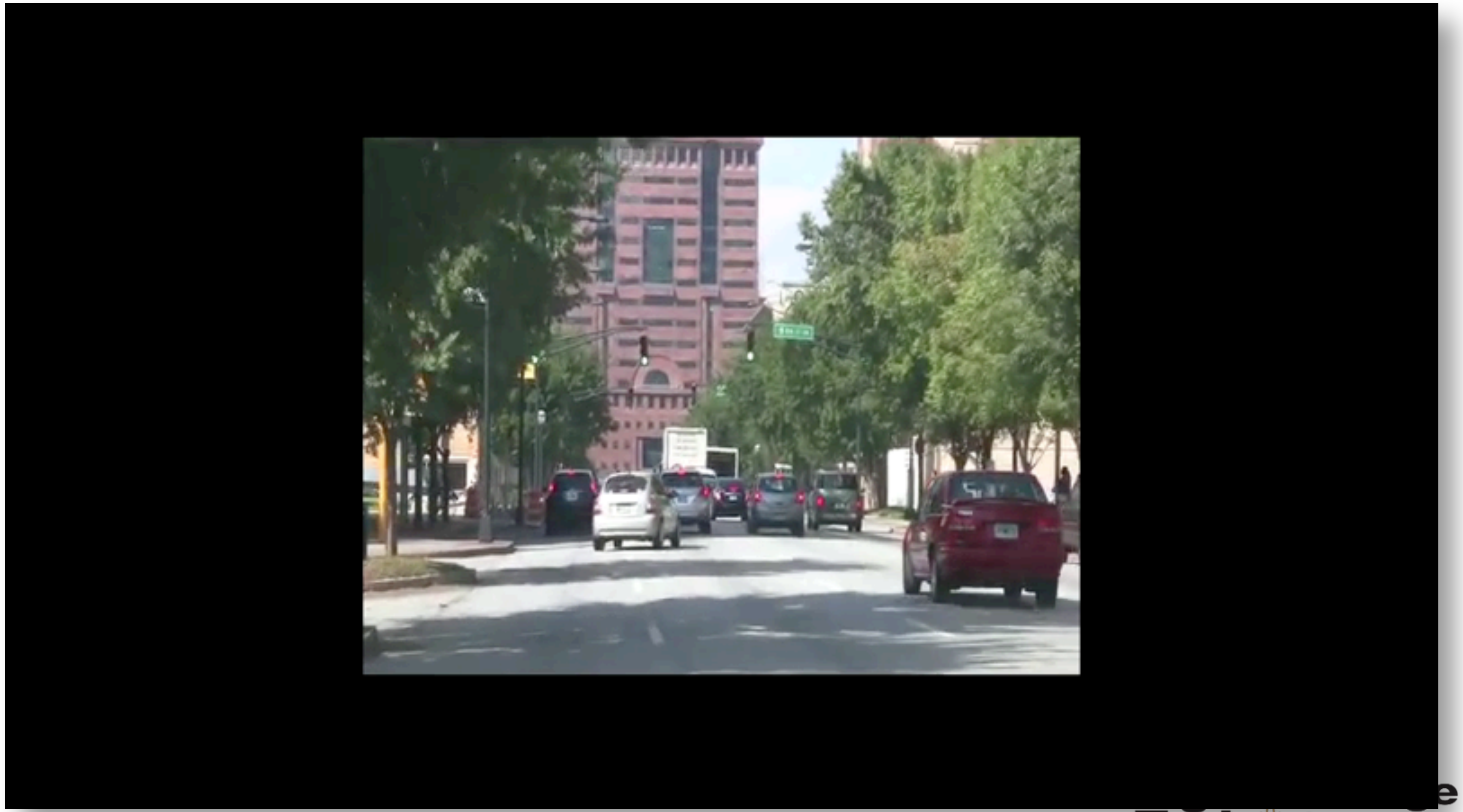
Geometric Context from Video

- Simplified description
  - Run video segmentation to yield super-regions
  - Extract spatio-temporal features
  - Train classifiers from features (from label dataset)
  - Different from super-parsing: Classifiers work directly on regions (not images)
  - Also: Aggregate prediction over hierarchy



Geometric Context from Video

Hussein, Grundmann, Essa, CVPR 2013



Research
at Google

e of
Tech Computing

Radiometric Calibration for Video Segmentation

Grundmann, Kang, Essa (ICCP 2013)

- Goal: Segmentation robust to gain changes in video
- Simplified description:
  - Radiometric calibration Colors → Irradiance
  - Segment in irradiance

Research at Google

Georgia Tech College of Computing

Pixels to Semantics (YouTube scale)

- G. Hartmann, M. Grundmann, J. Hoffman, D. Tsai, V. Kwatra, O. Madani, S. Vijayanarasimhan, I. Essa, J. Rehg, R. Sukthankar
Weakly Supervised Learning of Object Segmentations from Web-Scale Video
ECCV Workshop on Web-scale Vision and Social Media, 2012 (Best Paper)

Input video: dog

Result

Weakly supervised segmentation

- Simplified description
  - Stabilize and segment videos to yield good input (input: YouTube videos)
  - Yield spatio-temporal segments via Video Segmentation
  - Extract features from segments (appearance, motion, texture, shape etc.)
  - Weakly supervised: Training data only has (video, label) pairs (similar to MIL)
  - Learn model for each label by pooling over all extracted segments (MILBoost)
  - Evaluation: Manual annotation via online tool

Weakly Supervised Training Data

(video-level tags)

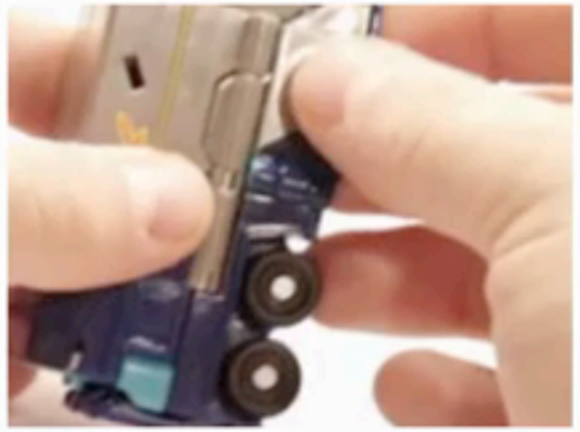
dog

bike

boat

horse

helicopter



transformers

card

robot

Talk outline

- Applications of video segmentation
- Online video segmentation and annotation
- Open source video segmentation

Research at Google

Georgia Tech College of Computing

Online video segmentation

- Goal:
  - Enable researchers / users to segment videos
- Initially launched on a single server in 2010 (limited resolution and length)
- In 2011: videosegmentation.com
  - Hosted on two machines with GPUs (for flow)
  - No limits on resolution or length (streaming)
  - One job at a time (HD video could stall queue for everyone)
  - REST API for terminal based usage
- Now:
  - Build fast, highly parallel cloud solution

Research at Google

Georgia Tech College of Computing

Fast online video segmentation

- Main ingredients:
  - Underlying segmentation algorithm O(n)
    - Parallelize over segmentation and hierarchical segmentation
  - Streaming segmentation
  - Run flow and both segmentations in a parallel pipeline
  - Resolution independence

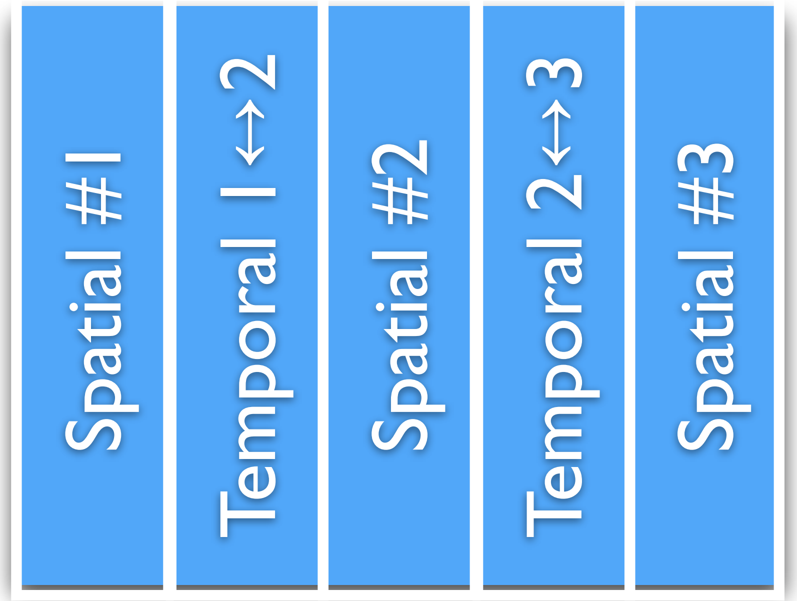
Research at Google

Georgia Tech College of Computing

Fast O(n) segmentation

- Use bucket sort: Discretize edge weight domain into 2-4K buckets (bucket sort)
  L1 RGB color distance: 768 values
- **Complexity: O(n)** [no large multipliers, α(n) < 5 for all practical values of N]
- Spatial and temporal edges are disjoint → Bucket lists:
  - For N frames use 2 * N – 1 list of 2K buckets
  - Create in parallel via on-demand threads! **31% faster!!**
- For hierarchical segmentation:
  - Evaluate region ↔ neighbor edges in parallel
  - Hash edges to weights for fast graph construction

Parallel construction



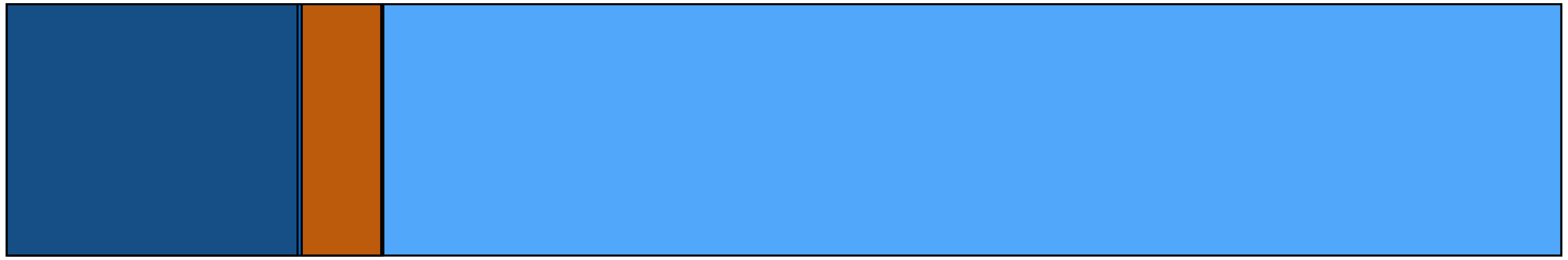
Research at Google

Georgia Tech College of Computing

Streaming video segmentation

- Clip-based with overlap
- Original implementation modified edge weights
- Modifying edge weights is bad!
  - Single-link clustering
  - Changes order of merges
  - If used with Felzenszwalb criteria prohibits merges

Video Volume: frame# →



Segment 30 frames

Output result

Constrain graph before segmentation using result of previous clip

Edge within a region
=> weight = 0
Edge across boundary
=>weight = ∞

Research at Google

Georgia Tech College of Computing

Constraint streaming video segmentation

- Instead of modifying edge weights:
  Add to each region a constraint id   (-1 for unconstrained)
- Proposed by several authors:
  - [Lezama et al. , 2011]: Track to the future
  - [C. Xu et al., 2012]: Streaming hierarchical video segmentation
  - videosegmentation.com, since early 2011
- Criteria:
  - Merge regions if constraints are equal (regardless of distance)
  - Never merge two different constraints
  - Constrains are sticky: Propagates to unconstrained nodes

- region_id
- size
- descriptor[]
- is_done
- constraint

Research at Google

Georgia Tech College of Computing

Resettable Constraints: Motivation

- Problem: Constraints over-constrain!
- Following example from
  [Joseph Tighe and Svetlana Lazebnik, 2012]:
  SuperParsing: Scalable Nonparametric Image
  Parsing with Superpixels
- Problem:
  Pixels in distance are grouped together
  (perspective = averaging)

  Cannot be broken apart!

Research at Google

Georgia Tech College of Computing

Resettable constraints: Observation

- Regions become increasingly larger
  - Over-constraint: Everything gets grouped together
  - Constraints are dominating the segmentation process
- [Joseph Tighe and Svetlana Lazebnik, 2012] p. 16 about videosegmentation.com:
  “we have found the segmentation results to be better if we run the videos through the system backwards”

segmented backwards

Resettable constraints

- New merge criteria supplies descriptor distance
- New split operation:
  - Split if: Same constraint but descriptor distance > 15%
  - Reset constraint of smaller region (if < 1/3) or both
- Requires addition of virtual nodes/edges for topological information (neighbors)

Research at Google

Georgia Tech College of Computing

Resettable constraints: Comparison

Constraint segmentation

Segmentation with resettable constraints

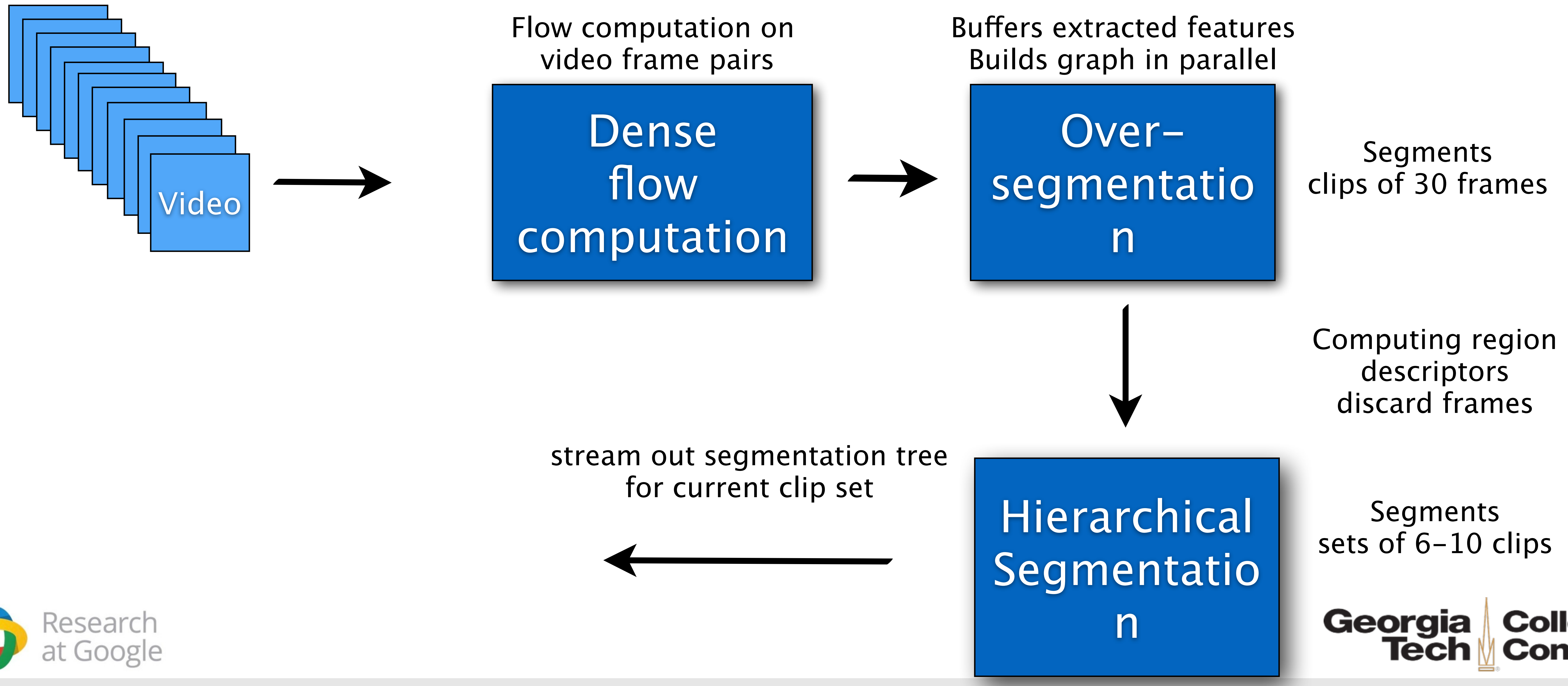
Research at Google

Georgia Tech College of Computing

Fast online video segmentation

- Main ingredients:
  - Underlying segmentation algorithm O(n)
  - Streaming segmentation
  - Run flow and both segmentations in a parallel pipeline
  - Resolution independence

Segmentation Pipeline



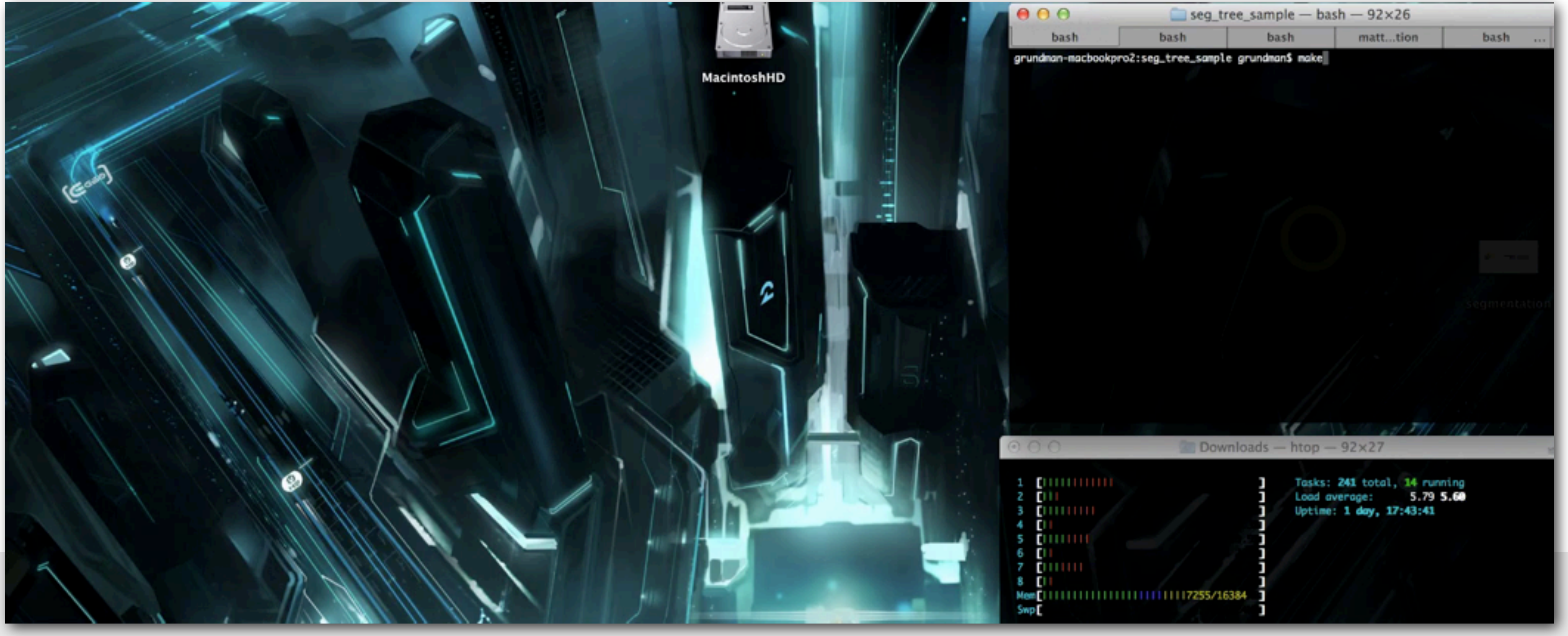
Research at Google

Georgia Tech College of Computing

Segmentation pipeline

Maximum parallelism

- Flow, Over-segmentation and Hierarchical segmentation can be run independently once input is available: 400% CPU sustained

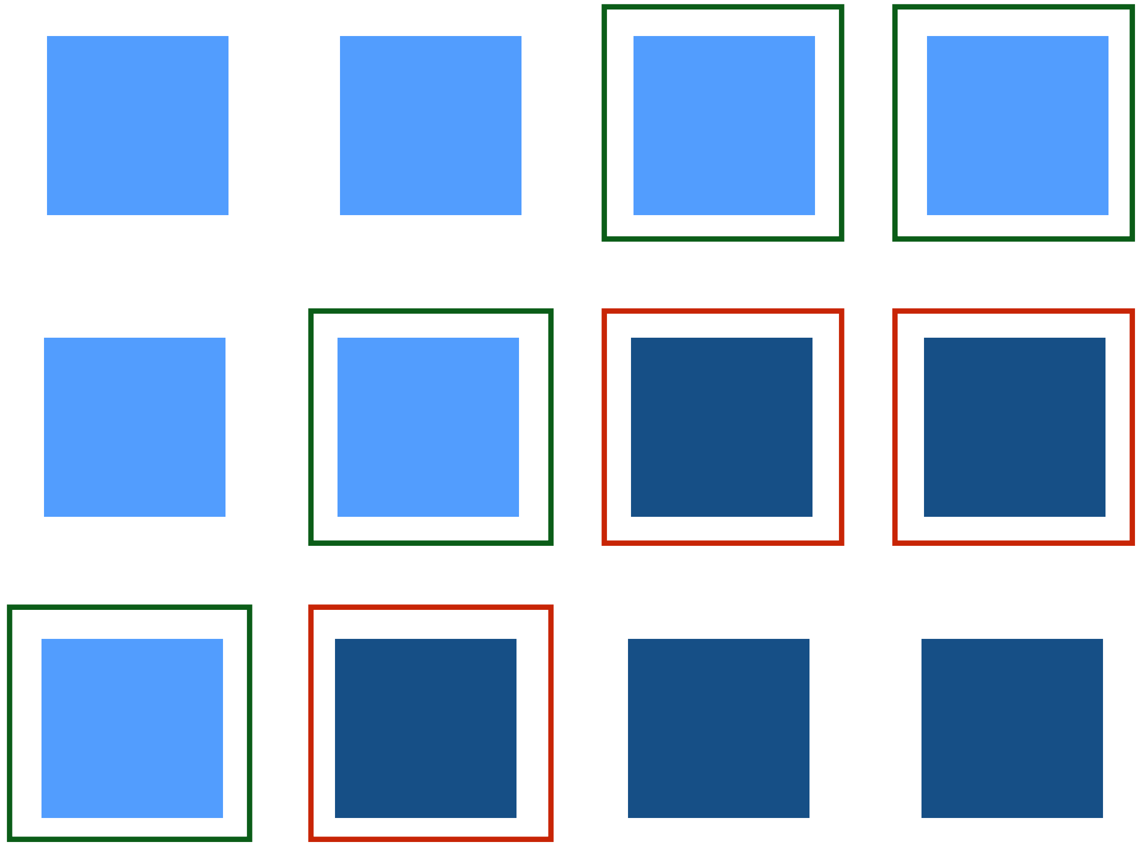


Segmentation resolution

- Problems with current approach:
  - Segmentation is always computed for a specific resolution (e.g. 360p, 720p, etc.)
  - Complexity of flow and over-segmentation grows with video resolution
  - Segmentation representation grows with resolution
    Rasterization: Set or scanline intervals (RLE encoding)
- Rasterization does not enable geometric transformations
  (w/o need for bilateral upsampling)

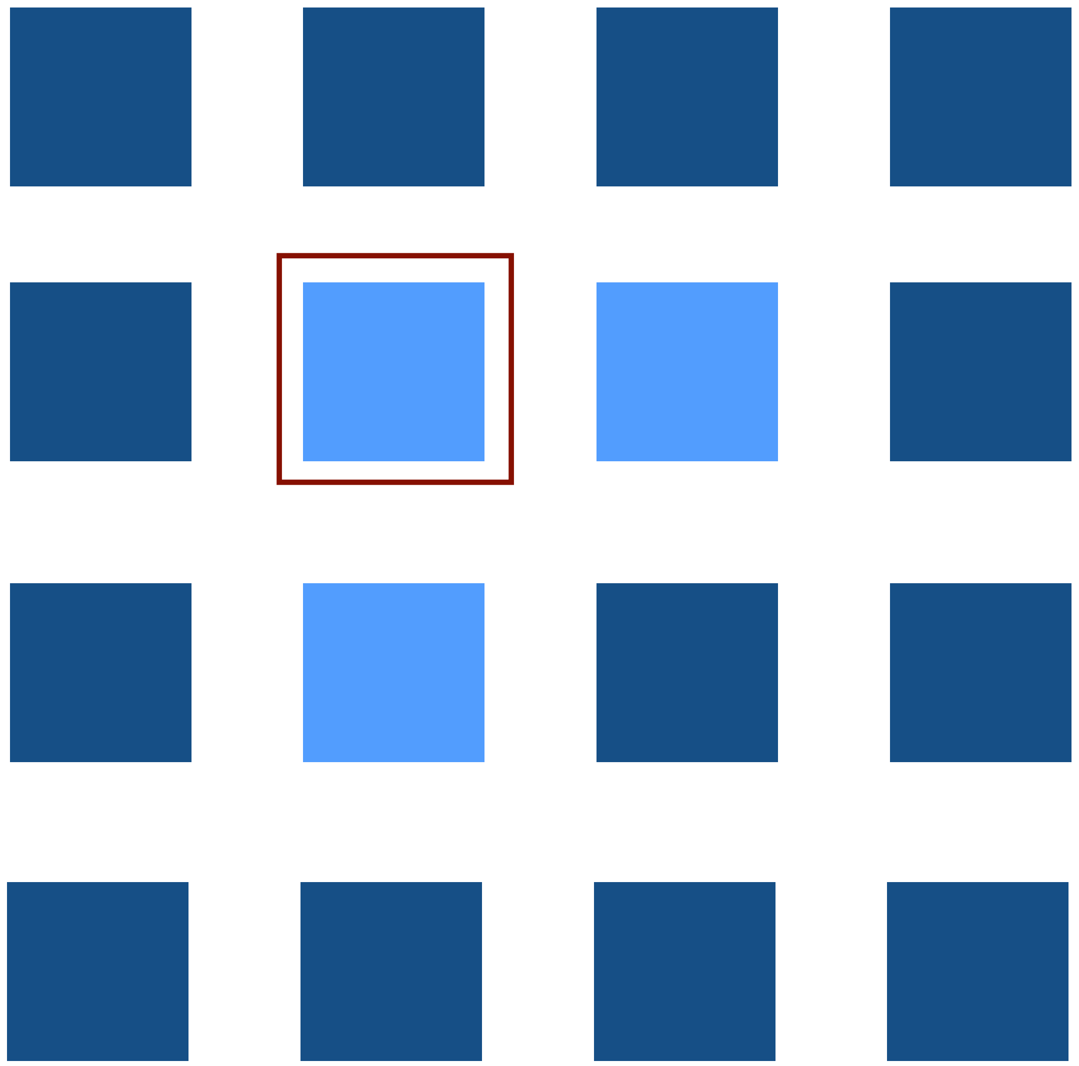
Compute segmentation boundaries

- Seems trivial at first:
  - Raster scan → Boundary pixel: current region is different from N4
  - Not "water tight" → double boundaries



Unique segmentation boundaries

- Based on:
  "A contour tracing algorithm that preserves common boundaries between regions"
  Yuh-Tay Liow, CVGIP: Image Understanding, 1991
- Idea: Similar to polygon rasterization (don't render bottom or rightmost boundary)
  - Assume N4 segmentation → N8 boundary
  - Start at most top-left pixel
  - 12 different configurations

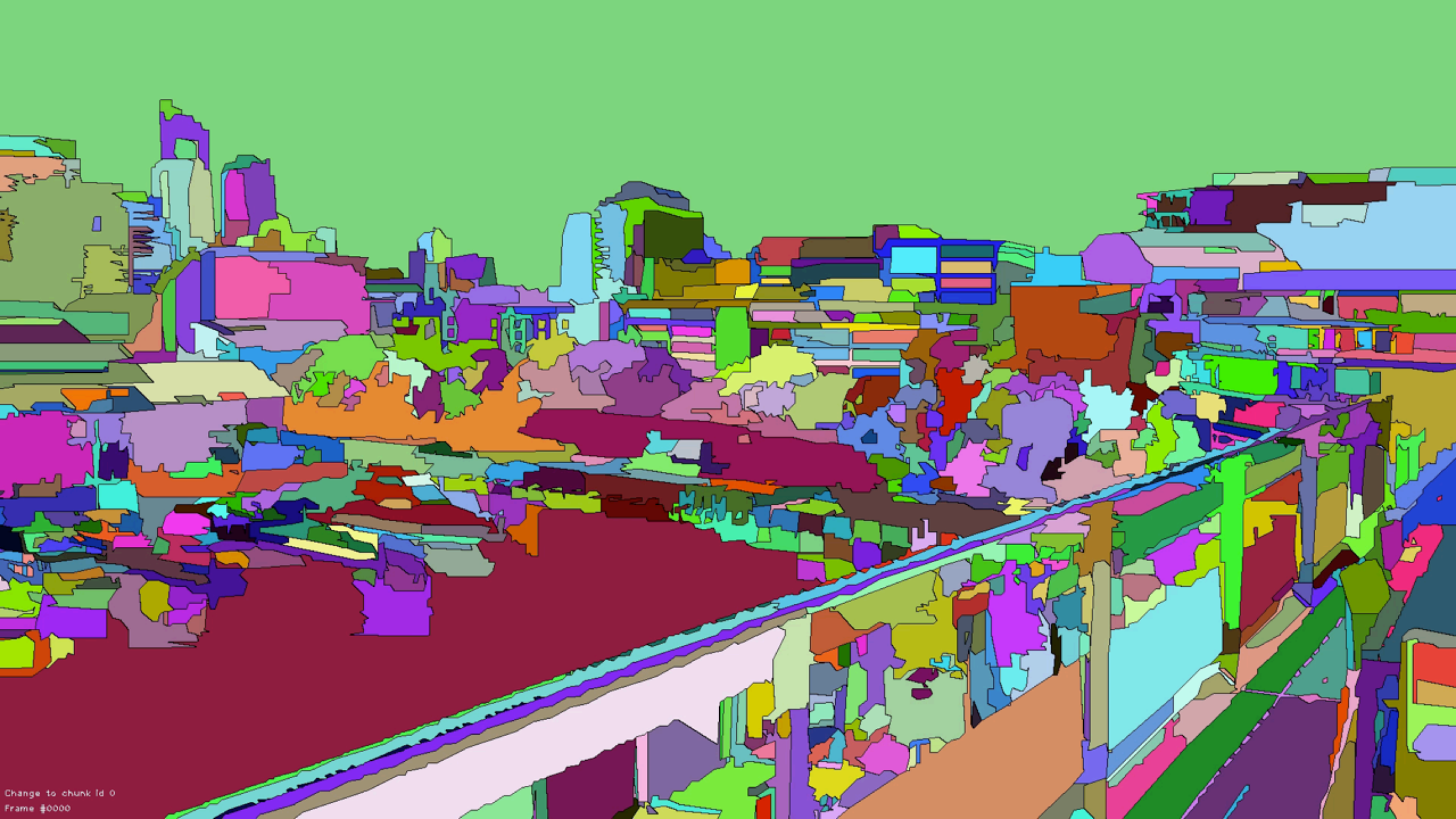


Research at Google

Georgia Tech College of Computing

Vector representation for Video Segmentation

- Extract boundaries for each component of a region (different from image case)
  - Trace each component
- Simplify boundaries via Ramer-Douglas-Peucker algorithm
  - Yields a water-tight polygon representation per region component
- Store coordinates into a vector mesh
  - **Geometrically transform mesh**
  - Rasterize if needed
- **Enables downscaling of input video and upscaling of result!**
  Segment 1080p! (~1 billion edges for 1s)


Change to chunk id 0
Frame #0000

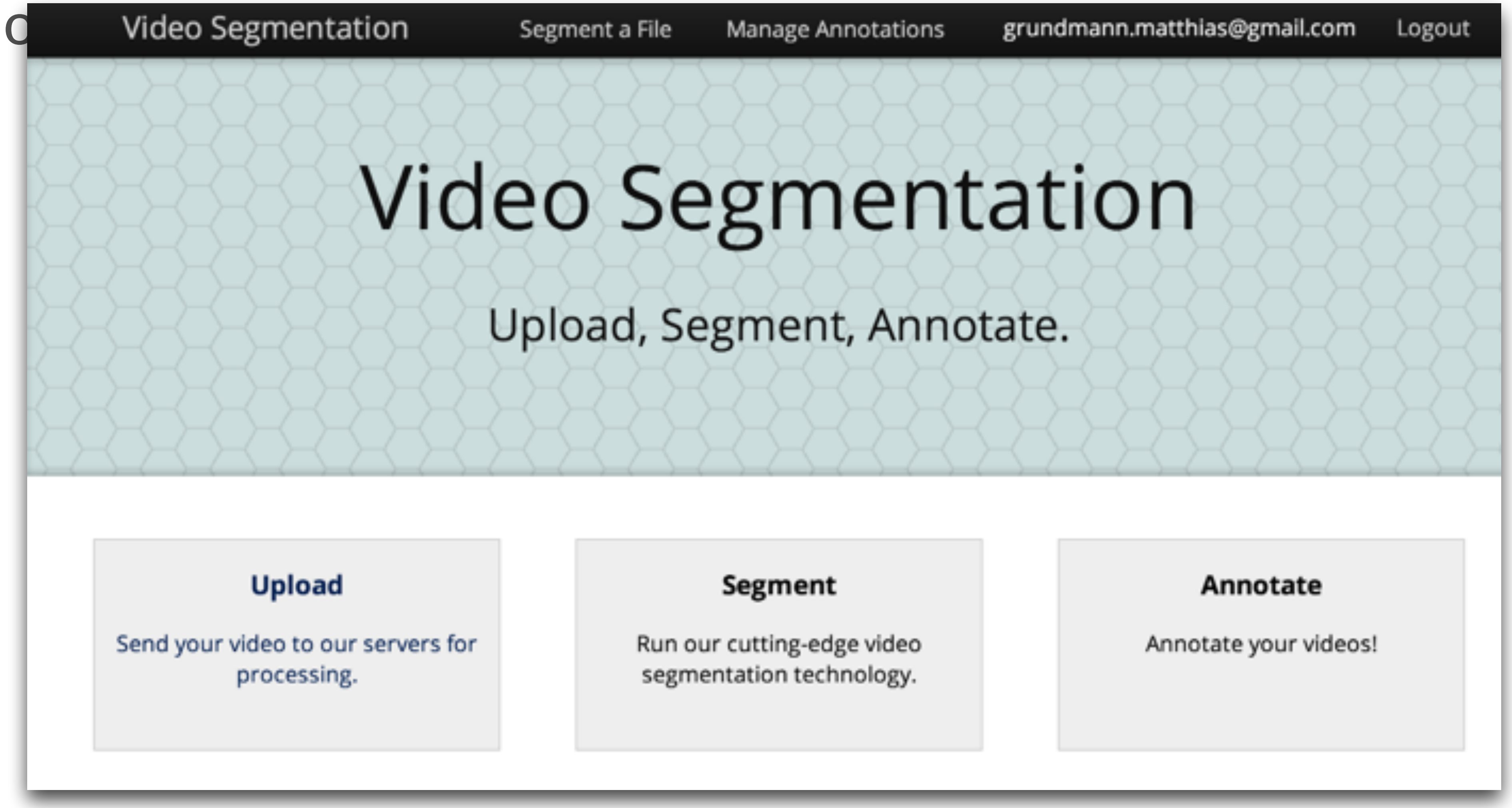
Fast online video segmentation

- Main ingredients:
  - Underlying segmentation algorithm O(n)
  - Streaming segmentation
  - Run flow and both segmentations in a parallel pipeline
  - Resolution independence

Video Annotation

Online Video Segmentation and Annotation

- End-to-end system for online video segmentation and annotation
- www.videosegmentation.com

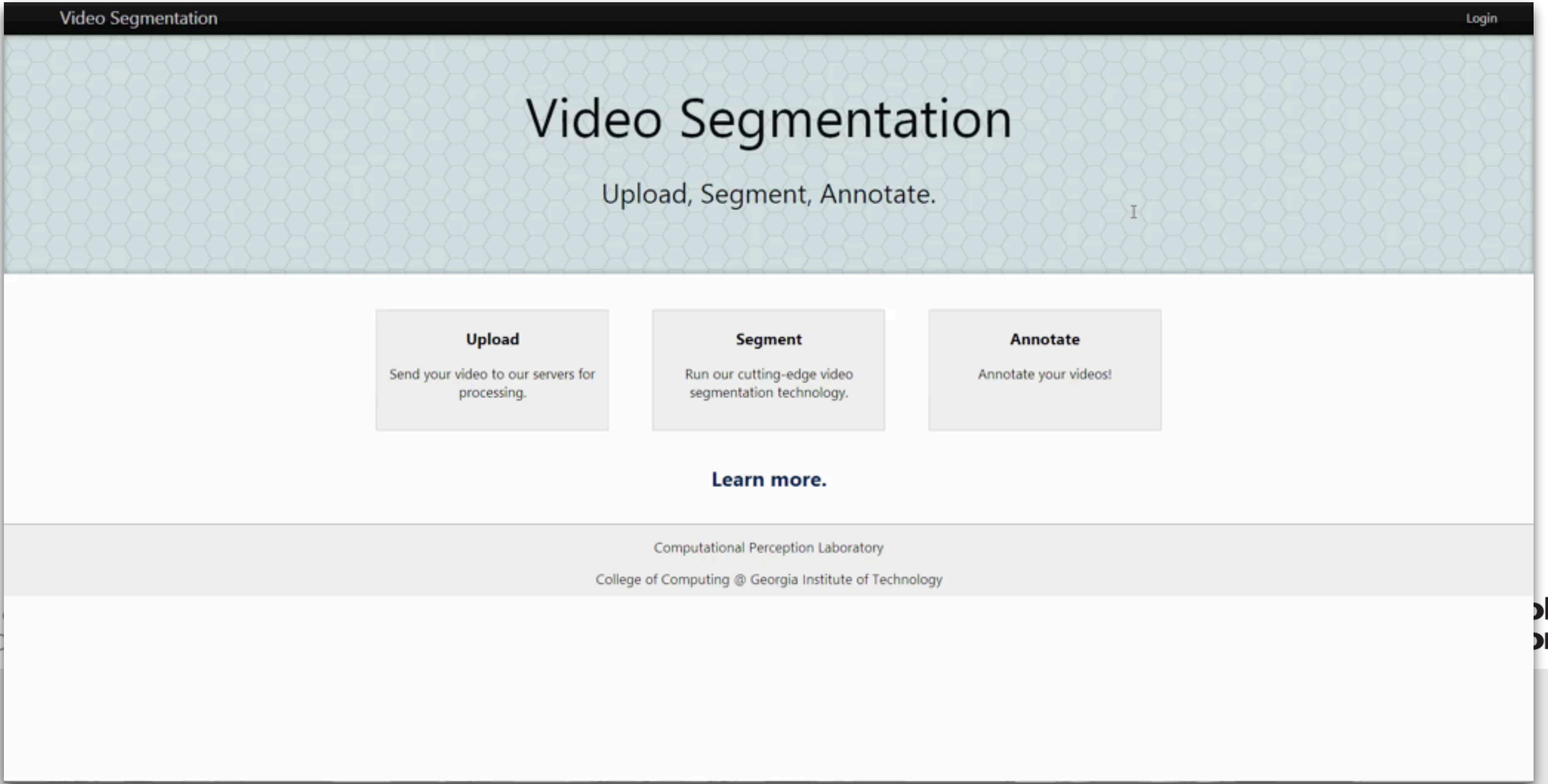


Research at Google

Georgia Tech College of Computing

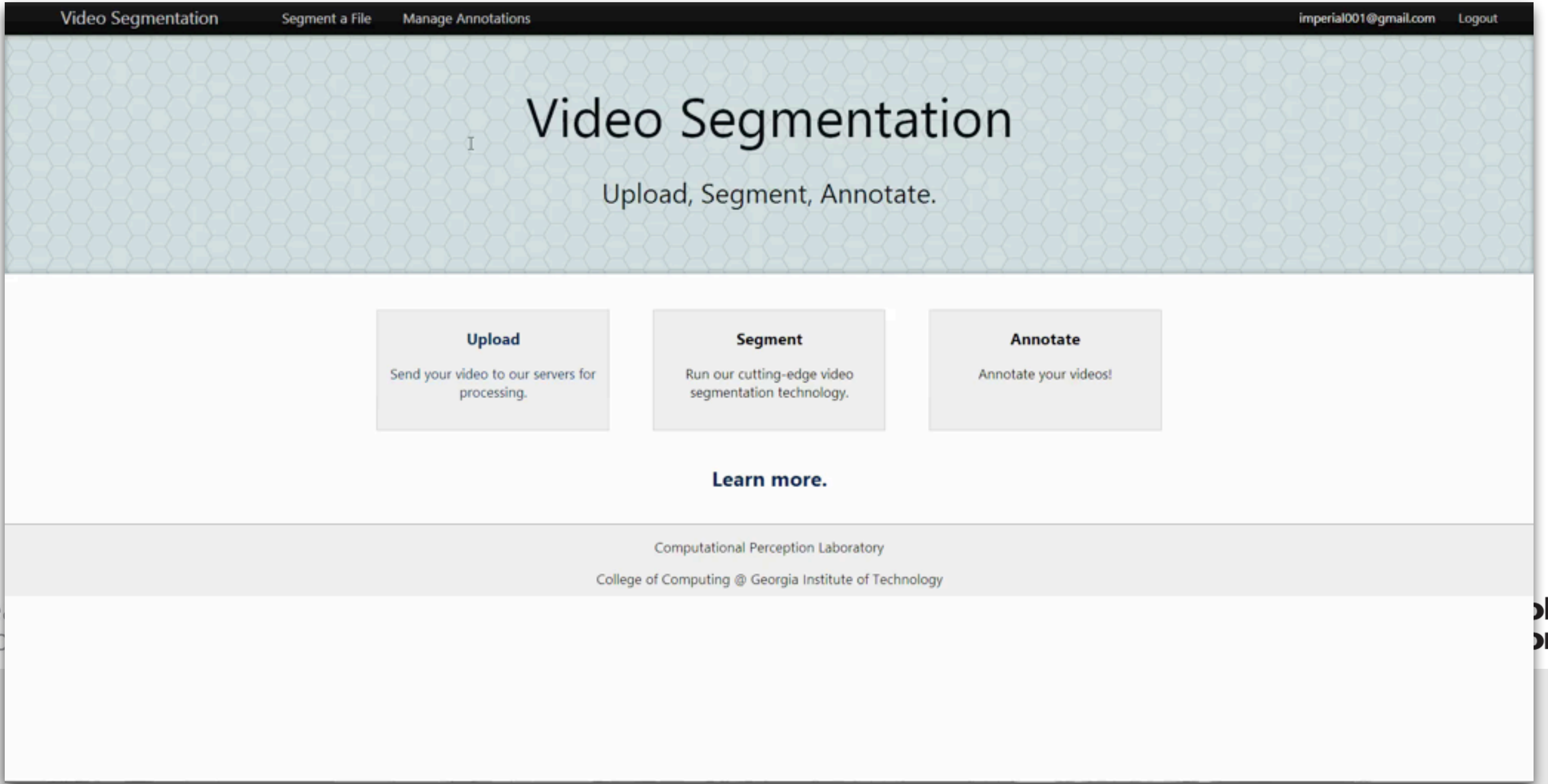
Online Video Segmentation and Annotation

Segment your videos



Online Video Segmentation and Annotation

Adjust options

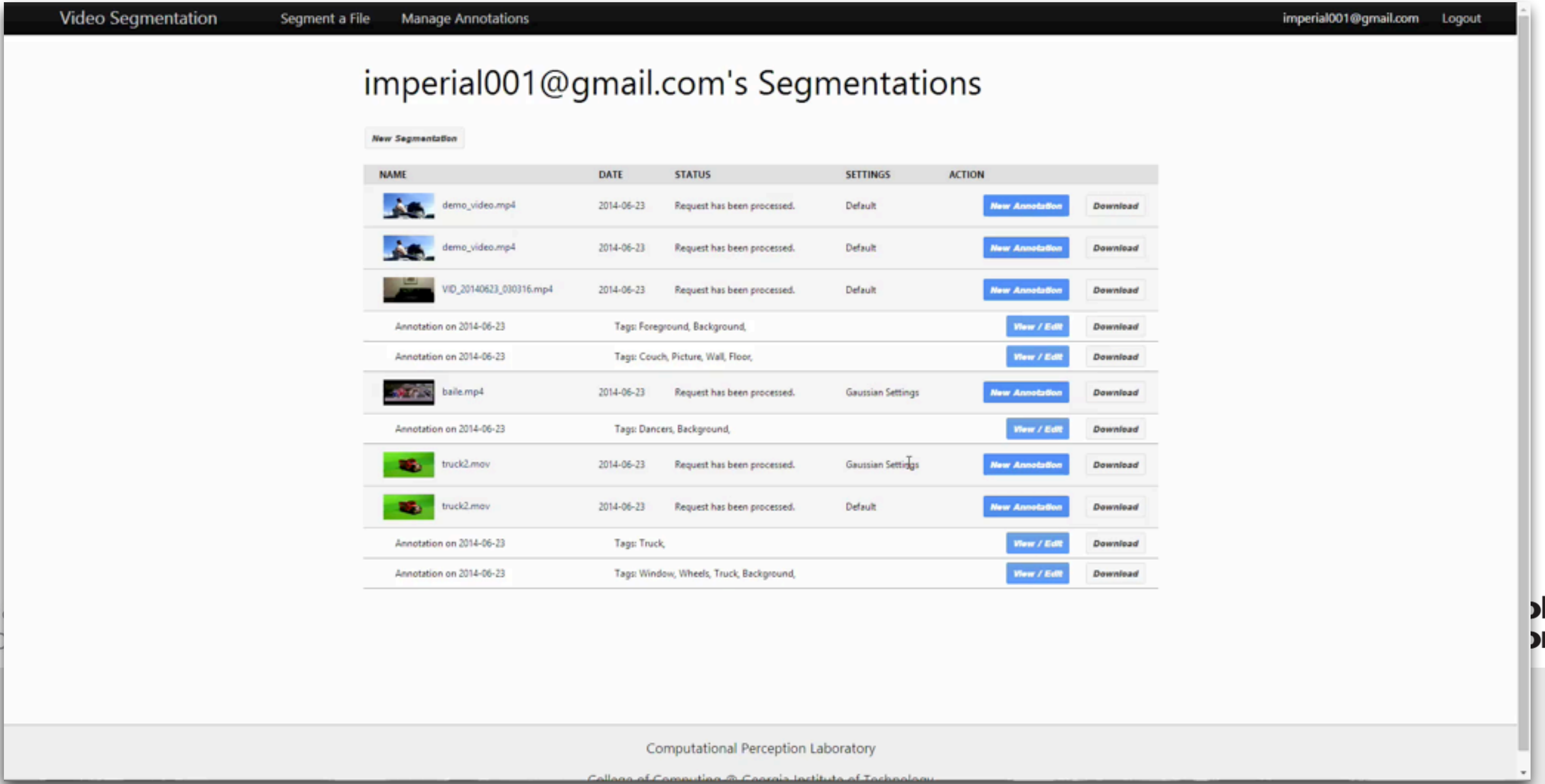


Rese
at Go

llege of
mputing

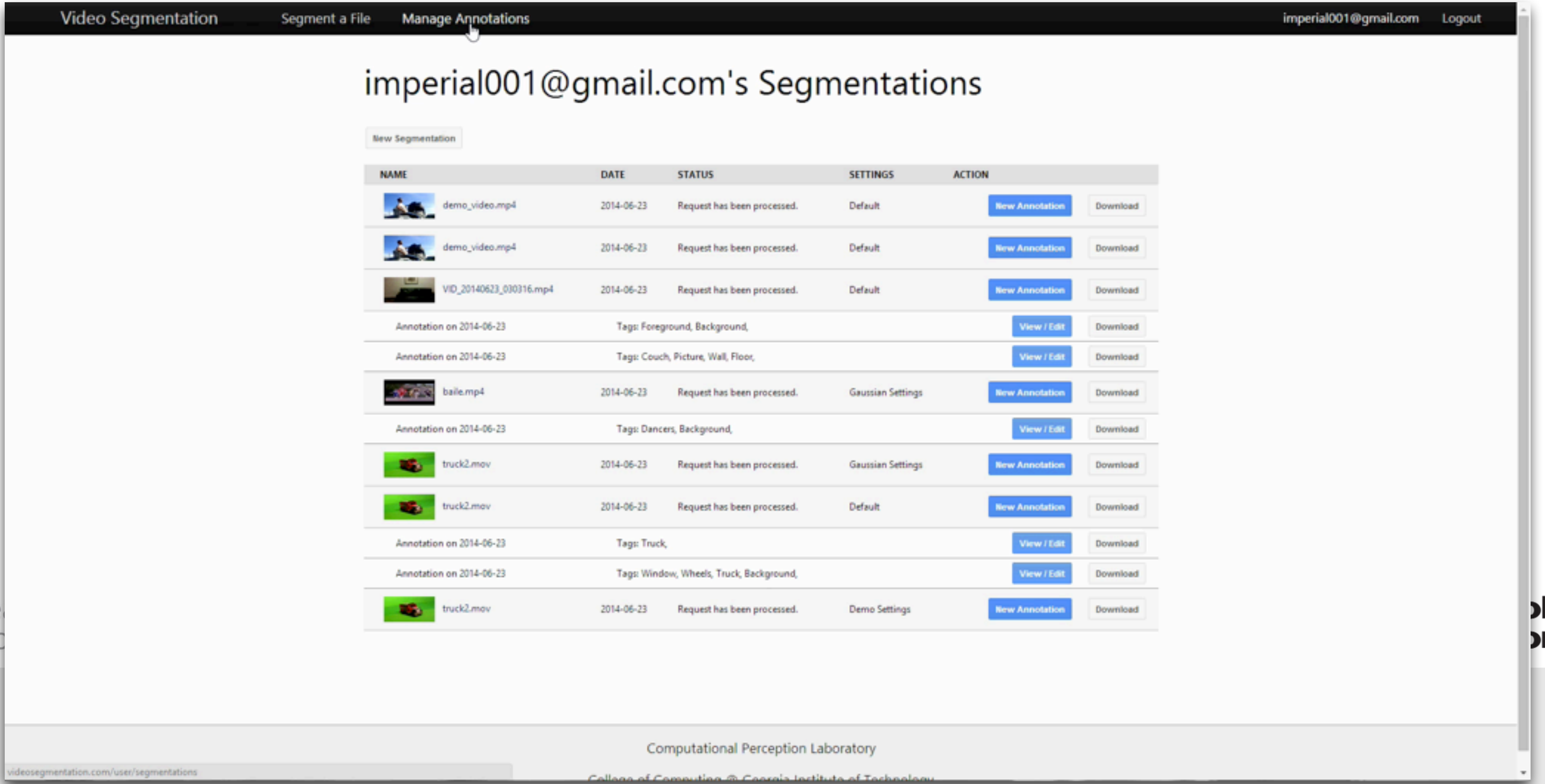
Online Video Segmentation and Annotation

Annotate!



Online Video Segmentation and Annotation

Download results

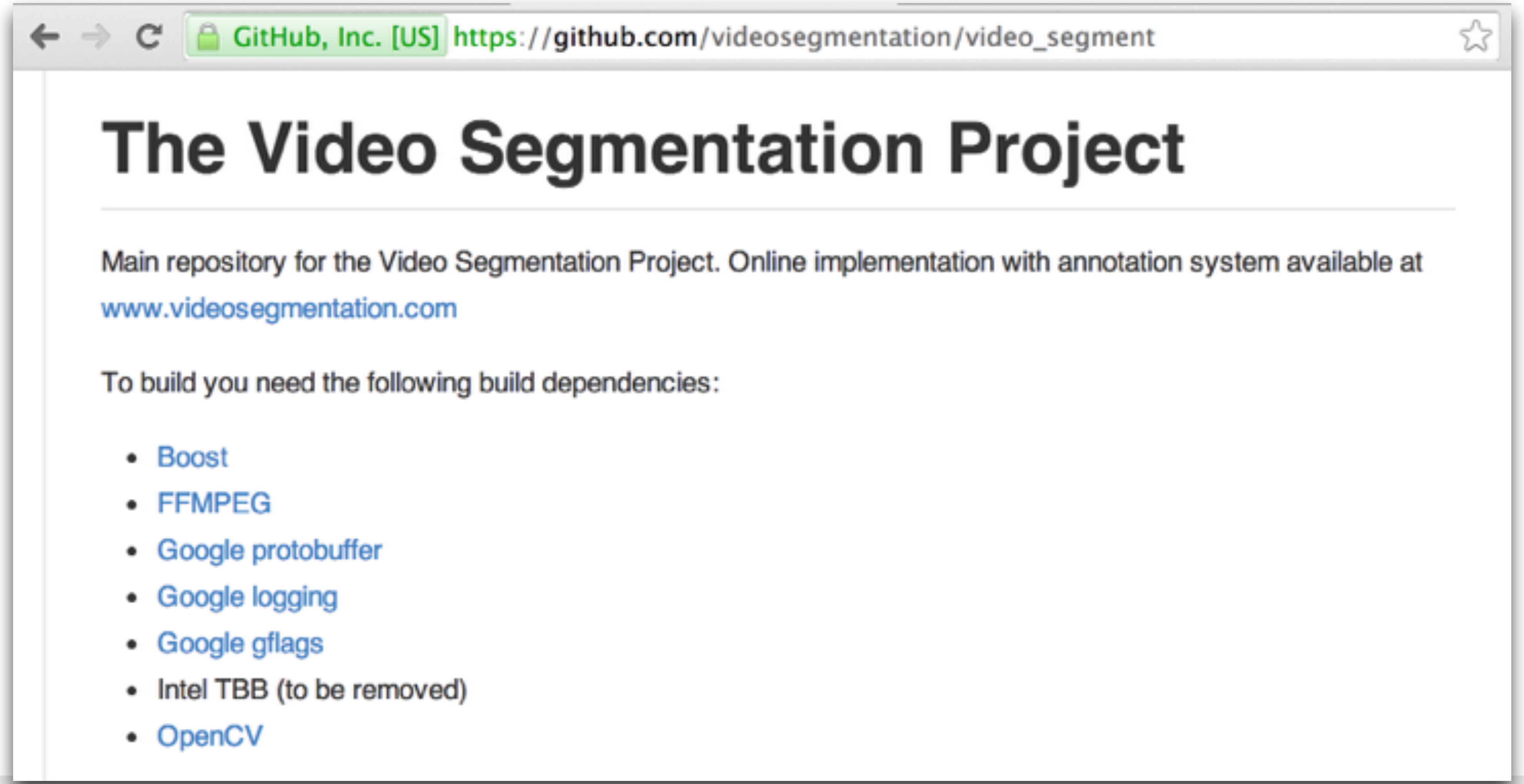


Talk outline

- Applications of video segmentation
- Online video segmentation and annotation
- Open source video segmentation

The Video Segmentation Project

- Open source implementation of everything shown today
  - https://github.com/videosegmentation/video_segment
  - BSD license
- Generic segmentation interfaces
  - Over segmentation:
    - Define pixel distance
    - region descriptors,
    - merge thresholds
  - Hierarchical segmentation:
    - Define region descriptors
    - distances



Research at Google

DenseSegmentation

Fully customizable features, distances and descriptors

C++

```
// Create generic distance for space and time (here L1, color only).
typedef SpatialCvMatDistance<ColorDiff3L1, ColorPixelDescriptor> SpatialCvMatDistance3L1;
typedef TemporalCvMatDistance<ColorDiff3L1> TemporalCvMatDistance3L1;

// Bundle spatial and temporal distances.
struct DistanceColorL1 : DistanceTraits<SpatialCvMatDistance3L1,TemporalCvMatDistance3L1>
{ };

// API callback to create dense segmentation graph.
virtual DenseSegGraphInterface* CreateDenseSegGraph(...) {
```


1.

RegionSegmentation

Fully customizable region descriptors and distances

C++

```
DescriptorExtractorList* extractors;       // Supplied by API
DescriptorUpdaterList* updaters;           // Supplied by API
shared_ptr<AppearanceExtractor> appearance_extractor(
    new AppearanceExtractor(options_.luminance_bins, options_.color_bins,
                            features[0]);    // Stores image.
extractors->push_back(appearance_extractor);
updaters->push_back(shared_ptr<NonMutableUpdater>(new NonMutableUpdater()));
shared_ptr<FlowExtractor> flow_extractor(
    new FlowExtractor(options_.flow_bins, features[1]));   // Stores flow images
extractors->push_back(flow_extractor);
updaters->push_back(shared_ptr<NonMutableUpdater>(new NonMutableUpdater()));
```


1.

Google Developers

Georgia Tech College of Computing