LIBSVX and Video Segmentation Evaluation

Chenliang Xu and Jason J. Corso

Computer Science and Engineering
SUNY at Buffalo

Electrical Engineering and Computer Science
University of Michigan

Talk Highlights

- LIBSVX Library Methods
  - Five offline supervoxel methods
  - One streaming supervoxel method (v2.0)
- Evaluation Benchmark
  - A set of 2D frame-by-frame & 3D volumetric metrics
  - Human independent metrics
- Flattening Algorithm: Uniform Entropy Slice (v3.0)
- Updates & Recognition (v3.1)

[Xu & Corso CVPR 2012]
[Xu, Xiong & Corso ECCV 2012]
[Xu, Whitt & Corso ICCV 2013]

LIBSVX: Supervoxel Methods

LIBSVX: Supervoxel Methods

- Offline Algorithms:
  - Graph-Based (GB)
    [Felzenszwalb & Huttenlocher IJCV 2004]
  - Graph-Based Hierarchical (GBH)
    [Grundmann et al. CVPR 2010]
  - Segmentation by Weighted Aggregation (SWA)
    [Sharon et al. CVPR 2001, NATURE 2006], [Corso et al. TMI 2008]
  - Nyström Normalized Cuts (NCut)
    [Fowlkes et al. TPAMI 2004] [Shi & Malik TPAMI 2000]
  - Mean Shift (External at the author's website)
    [Paris & Durand CVPR 2007]
- Streaming Algorithm:
  - Graph-Based Streaming Hierarchical (streamGBH)
    [Xu, Xiong & Corso ECCV 2012]

Data Format

- Input Video
  - Extract frames in png and ppm formats.
  - Organize in a folder as a sequence of images: %05d.png/ppm.
  - Example: ffmpeg -i video.mp4 input/%05d.ppm

- Output Segmentation
  - A sequence of color-coded images. Each supervoxel index or label is coded with a unique RGB color.
  - Use read_video_supervoxels.m to convert a folder of segmentation frames to a 3D matrix in Matlab, and use cvlbmap.m to sort the labels to 1:N.

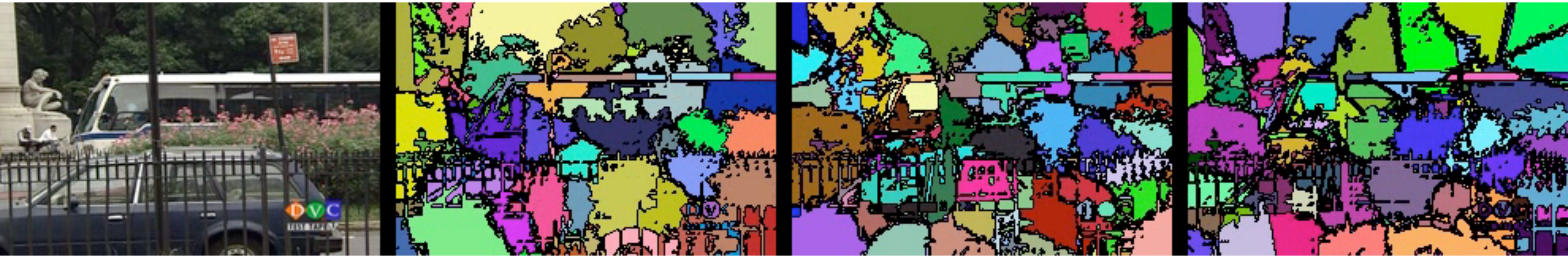
Nyström Normalized Cuts (NCut)

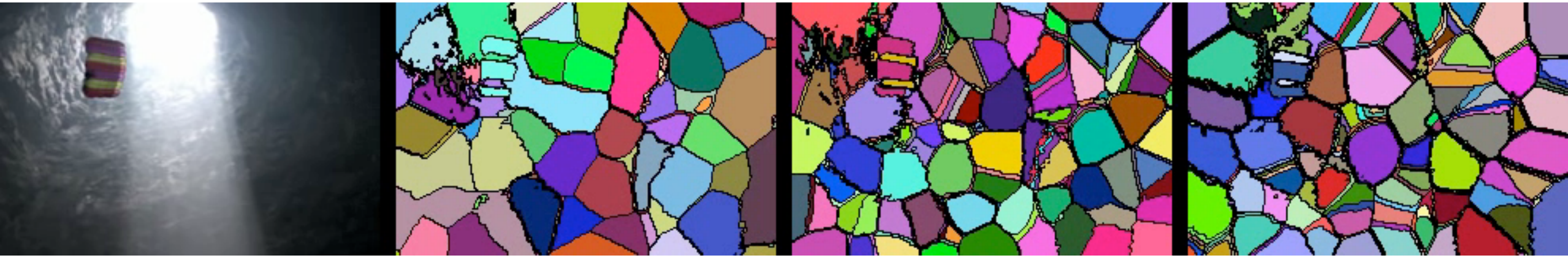
- Implement: MATLAB
- To Run: Nystrom_video(PathInput, PathOutput, numOfSvx, numOfSamples, numOfEvecs, sigmaE, sigmaLab, KNN)
  - **PathInput** & **PathOutput** – Paths to input and output directories.
  - **numOfSvx** – number of supervoxels
  - **numOfSamples** – number of sampled points in a video
  - **numOfEvecs**: number of eigenvectors
  - **sigmaE** – weight of Euclidean distance
  - **sigmaLab** – weight of Lab color distance
  - **KNN** – the option to use knn to assign final labels

[Fowlkes et al. TPAMI 2004] [Shi & Malik TPAMI 2000]

Nyström Normalized Cuts (NCut)

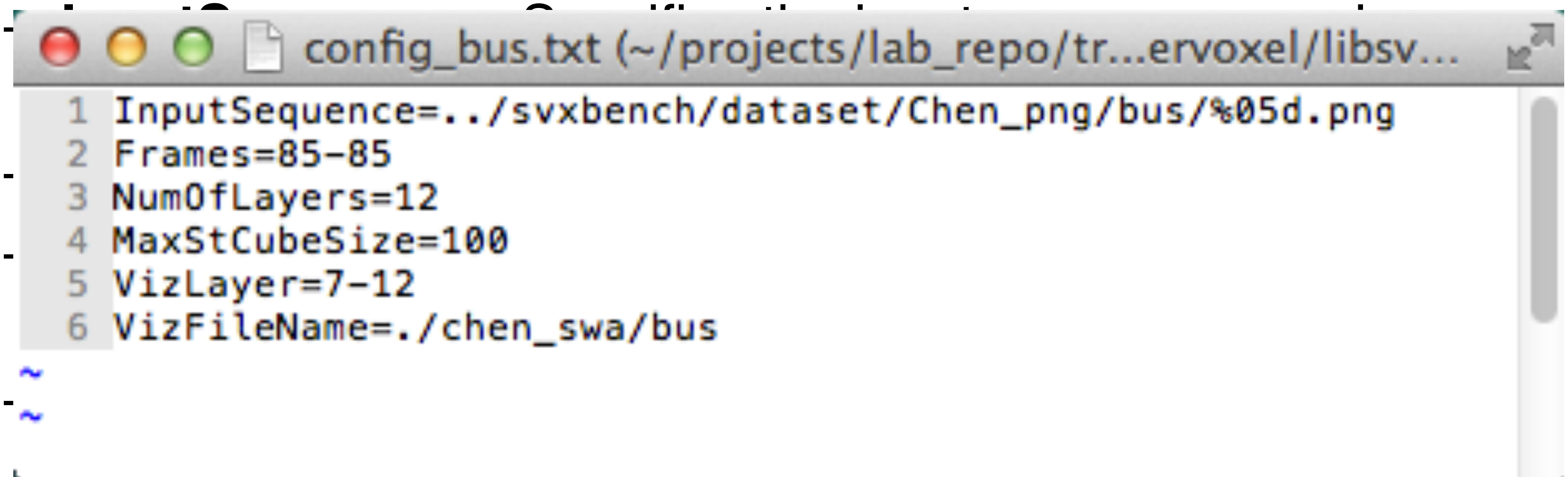
- Demo videos.





Segmentation by Weighted Aggregation (SWA)

- Implement: C/C++
- To Run: ./swa config.txt
- The following parameters are to be set in config.txt file.

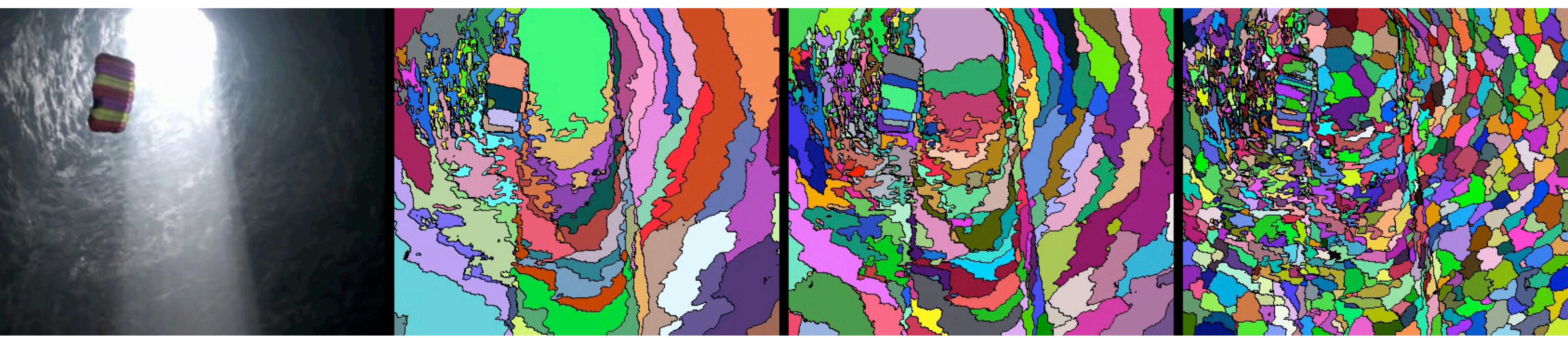


  – **VizLayer** – It specifies the layers of the hierarchy to be visualized.
  – **VizFileName** – The directory path for storing the visualization results.

[Sharon et al. CVPR 2001, NATURE 2006], [Corso et al. TMI 2008]

Segmentation by Weighted Aggregation (SWA)

- Demo videos.



Graph-Based (GB) and Hierarchical (GBH)

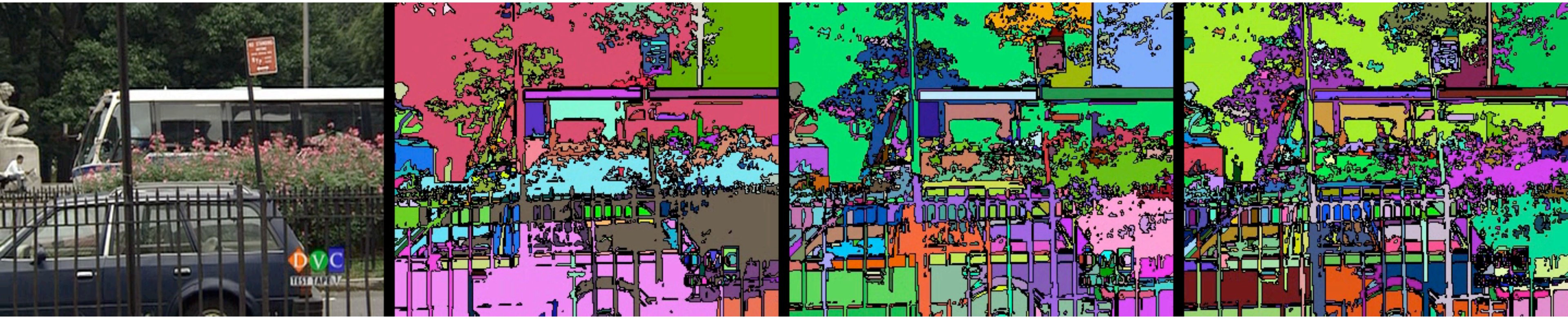
- Implement: C/C++
- To Run: ./gbh c c_reg min sigma hie_num input output
  - **c** – Governs the merging threshold of the two nodes in the minimum spanning tree during the oversegmentation stage. Bigger C means larger segments.
  - **c_reg** – Like c, it governs the merging of two nodes, but this is in the hierarchical levels whereas c is at the pixel level nodes.
  - **min** – Enforced minimum segment size for a whole supervoxel.
  - **sigma** – The variance of the Gaussian smoothing.
  - **hie_num** – The number of desired levels in the hierarchy + 1. If hie_num = 0, no hierarchy is created, which is GB.

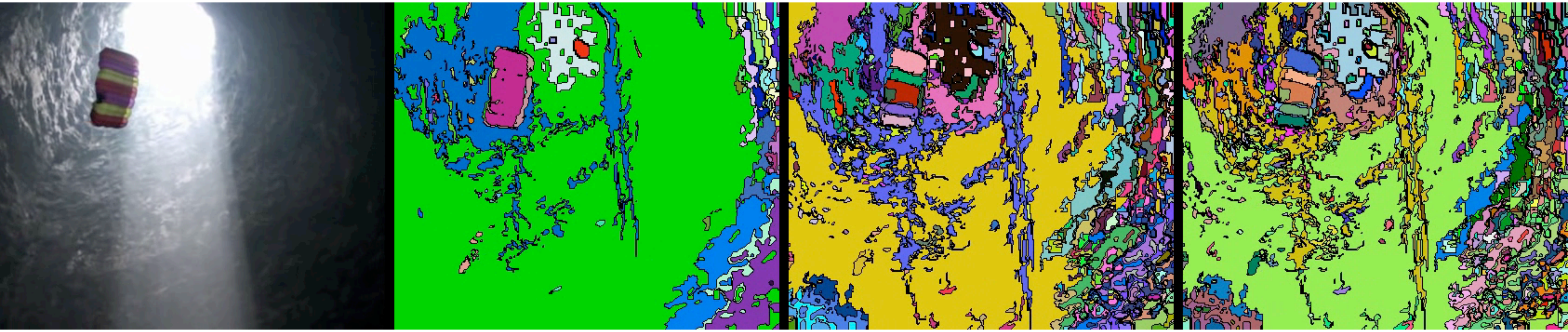
[Grundmann et al. CVPR 2010]

[Felzenszwalb & Huttenlocher IJCV 2004]

Graph-Based (GB)

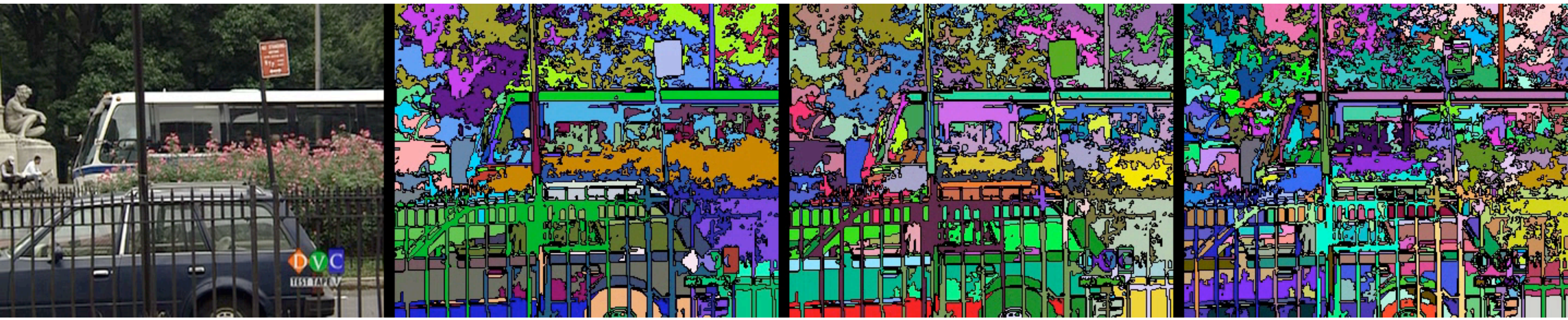
- Demo videos.

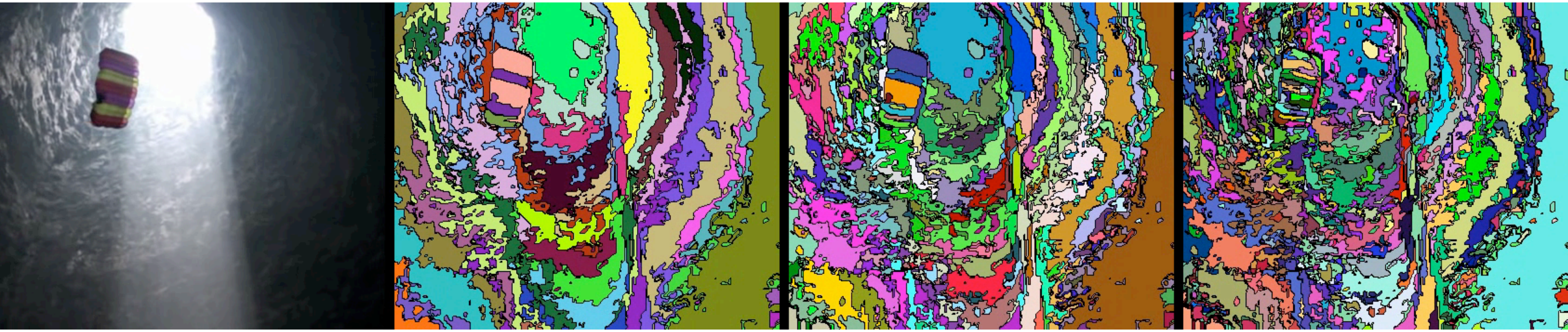




Graph-Based Hierarchical (GBH)

- Demo videos.





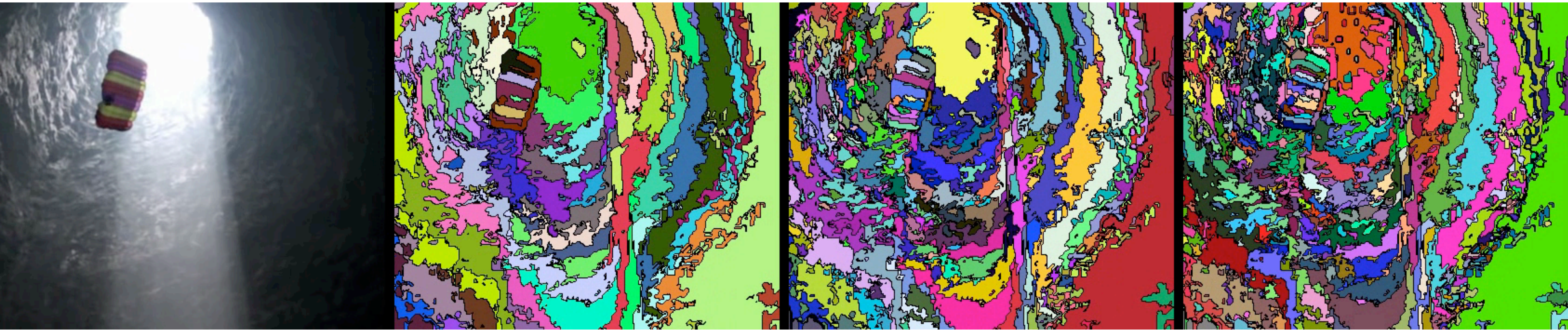
Graph-Based Streaming Hierarchical (streamGBH)

- Implement: C/C++
- To Run: ./gbh_stream c c_reg min sigma range hie_num input output
  - **range** – The number of frames to include in one subsequence (or clip). range = 1 means each frame is handled separately. range = number_of_frames means the whole video is handled at once. range = k is the typical way to run the streaming method.
  - Other parameters are just like in GBH.
- The program also streams the output of supervoxel segmentation for subsequences.

[Xu, Xiong & Corso ECCV 2012]

Graph-Based Streaming Hierarchical (streamGBH)

- Demo videos.



LIBSVX: Benchmark Evaluation

3D Undersegmentation Error

- It measures what fraction of the pixels exceed the boundary of the ground-truth segmentation.
- Ground-truth segmentation: $\mathcal{G} = \{g_1, g_2, \ldots, g_m\}$
- Supervoxels: $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$
- 3D Undersegmentation error:

$$UE(g_i) = \frac{\sum_{j=1}^{n} \mathrm{Vol}(s_j | s_j \cap g_i \neq \emptyset) - \mathrm{Vol}(g_i)}{\mathrm{Vol}(g_i)}$$

- where $\mathrm{Vol}(\cdot)$ denotes the number of the voxels that are inside a 3D volume.
- We take the average across all ground-truth segments, where they are equally weighted.

3D Segmentation Accuracy

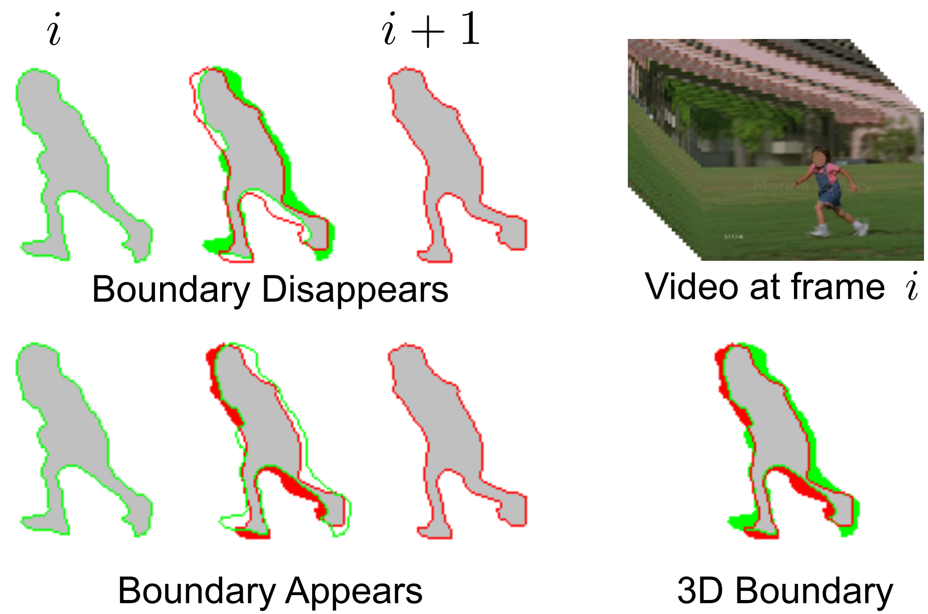
- It measures what fraction of a ground-truth segment is correctly segmented by the supervoxels: each supervoxel belongs to only one object or ground-truth segment.
- 3D Segmentation Accuracy:

$$SA(g_i) = \frac{\sum_{j=1}^{n} \text{Vol}(s_j)\mathbf{1}\{\text{Vol}(s_j \cap g_i) \geq \text{Vol}(s_j \cap \bar{g}_i)\}}{\text{Vol}(g_i)}$$

- where $\bar{g}_i = \mathcal{G} - g_i$ means all other segments.
- To evaluate the segmentation accuracy of a video, we again take the average of the segmentation accuracy score across all ground-truth segments.

3D Boundary Recall

- Within-frame boundary
  – Object boundary defined in 2D images.
- Between-frame boundary
  – A same object in adjacent two frames.



- The volumetric 3D boundary recall:

$$R = \frac{|S \cap G|}{|G|}$$

  – S and G are the 3D boundary maps for segments and GT.
  – $\bigcap$ solves a bipartite graph assignment between two boundary maps (often relaxed by 1 pixel offset).

Human-Independent Metrics

- **Explained Variation** considers the supervoxels as a compression method of a video.

Mean value of the voxels assigned to the supervoxel that contains x_i.

The global voxel mean.

$$R^2 = \frac{\sum_i (\mu_i - \mu)^2}{\sum_i (x_i - \mu)^2}$$

The actual video voxel value.

[Moore et al. CVPR 2008]

- **Supervoxel Mean Duration** measures the temporal extension of supervoxels. It measures the average length of supervoxels in a video.

Computational Cost

- **Computational Cost** in terms of time and peak memory consumption.

	GB	GBH	streamGBH	SWA	MeanShift	NCut
Time (s)	115	1166	1000	934	101	1198
Memory (GB)	6.9	9.4	1.6	19.9	3.8	20.9

- We report the computational cost of all methods for a typical video with 352 x 288 x 85 voxels. The experiment is done on a laptop featured with Intel Core i7-3740QM @ 2.70GHz and 32GB RAM running Linux.
- All methods are running in single thread except NCut running with 8 threads with resized spatial resolution to 240 x 160.

Evaluate your method with benchmark

- Dataset Setup
  - We include the Chen xiph.org dataset in the benchmark.
  - SegTrack and GaTech datasets can be downloaded and organized in a same file hierarchy.

- Preparing your results for use with the benchmark.
  - Compute supervoxels for a complete dataset.
  - Run with a varying set of parameters.
  - Segment each video with a distribution of supervoxel numbers varying from less than 200 to more than 900.
  - We use a linear interpolation scheme to compute the values for the curve. More samples leads to better estimate of the curve.

Evaluate your method with benchmark

- Let the root path of your results for Chen's data set be ROOT.

- The eight videos in the root path.
  - ROOT/bus
  - ROOT/container
  - ROOT/garden
  - ROOT/ice
  - ROOT/paris
  - ROOT/salesman
  - ROOT/soccer
  - ROOT/stefan

- The results for one video are put in different folders with the supervoxel number as the name in this example.
  - ROOT/bus/150
  - ROOT/bus/227
  - …
  - ROOT/bus/905

- The segmentation results of one video with one particular supervoxel number
  - ROOT/bus/150/00001.ppm
  - …
  - ROOT/bus/150/00085.ppm

Evaluate your method with benchmark

- Config and run EVALUATION.m

```
path_input_method = ROOT;          % path to your results of one data set
path_ppm = 'dataset/Chen_ppm';   % path to dataset
dataset = 1;       % 1 – Chen's xiph.org; 0 – SegTrack; 2 – GaTech
output_path = 'path_to_save_your_evaluation_results';
verbose = 1;       % option to show intermediate results
x_min = 200; x_max = 900;  % range of supervoxels generated by your method
```

- It generates both 2D frame-by-frame and 3D volumetric scores.

- To compare with scores by methods in the library, see XuCorso_CVPR2012_mat

Other Video Segmentation Evaluation

- A Unified Video Segmentation Benchmark [Galasso et al. ICCV 2013]
- Dataset: BVSD (100 videos) [Sundberg et al. CVPR 2011]
  - Labeled ground-truth frames at every 20^{th} frames.
  - Each is labeled by multiple human annotators.
  - Spatiotemporal coherence is preserved.
- Evaluation Metrics
  - Boundary Precision-Recall (BPR)
    - 2D image segment boundaries.
  - Volume Precision-Recall (VPR)
    - Treated as volumes at the ground-truth frames in a video.
    - Related to 3D undersegmentation error and 3D segmentation accuracy.

LIBSVX: Flattening Hierarchy

Prerequisites

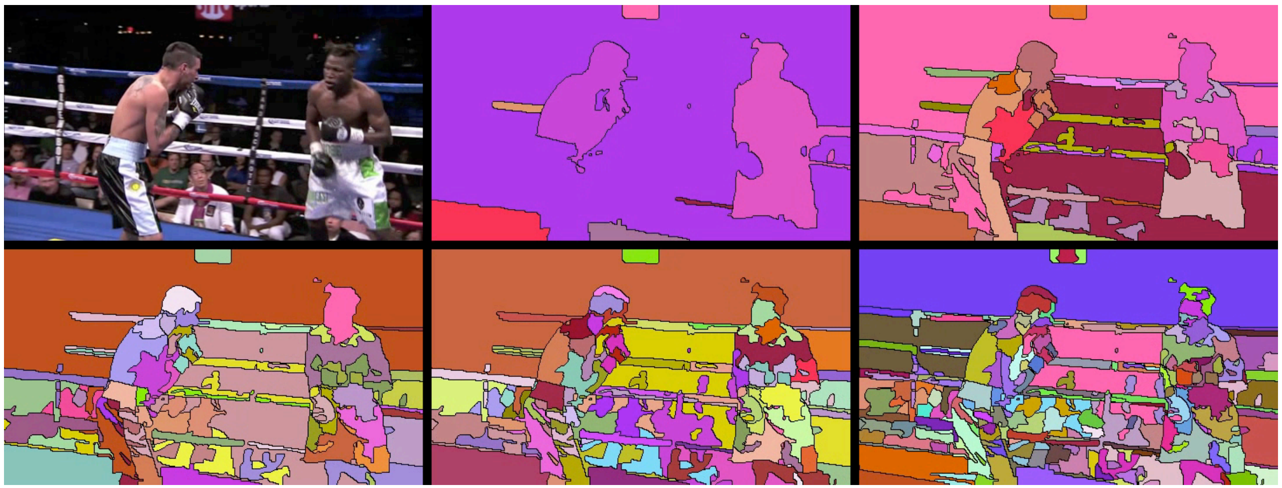
- Hierarchical Video Segmentation
  - LIBSVX: GBH or SWA (treeify)
  - GaTech web service: www.videosegmentation.com
- Solver for Binary QP
  - IBM ILOG CPLEX Optimization Studio V12.4+. http://www-03.ibm.com/software/products/en/ibmilogcpleoptistud/
- Feature Criteria
  - Ce Liu's Optical Flow Aug 1, 2011. http://people.csail.mit.edu/celiu/OpticalFlow/
  - Objectness V1.5. http://groups.inf.ed.ac.uk/calvin/objectness/
  - PFF DPM V4.01. http://www.cs.berkeley.edu/~rbg/latent/index.html

Usage

- To Run: ues(video_path, hie_path, output_path, hie_select_num, sigma, method, visflag)
  - **video_path**: path to raw video extracted png frames.
  - **hie_path**: path to hierarchical segmentation output.
  - **output_path**: folder to output flattening results.
  - **hie_select_num**: select a certain number of levels from the hierarchical segmentation as input to the flattening algorithm.
  - **sigma**: the weight between unary and binary.
  - **method**: feature criterion of motion-ness/object-ness/human-ness
  - **visflag**: option to output intermediate results.
- Treeify a Hierarchy
  - It modifies arbitrary supervoxel hierarchy to a tree structure, such as SWA. See treeify.m.
  - Enforce the supervoxel boundary agreement across levels in a hierarchy.

Demo: boxers

- Motion-ness



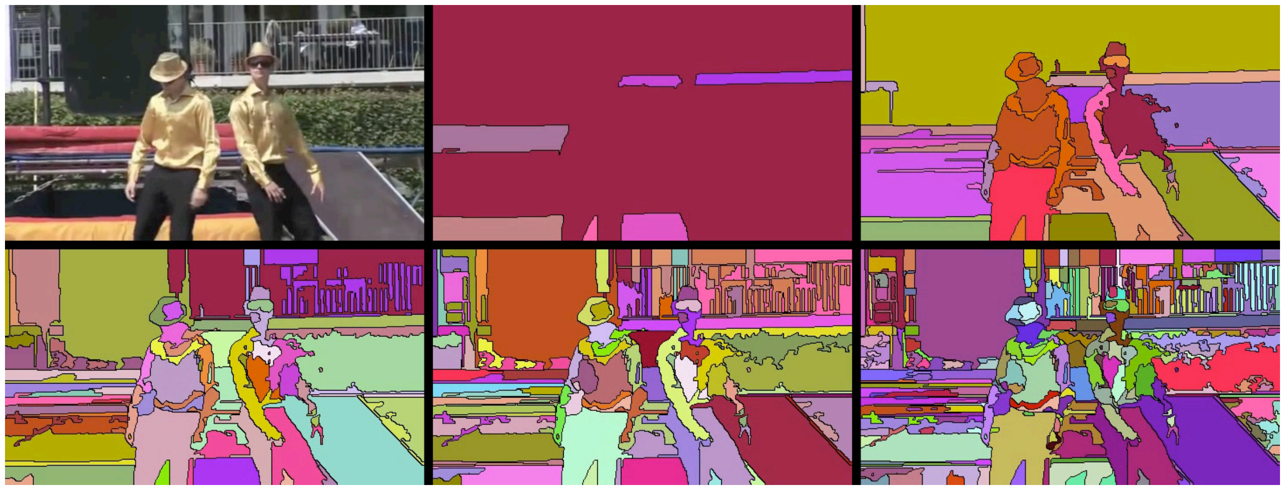
flow feature

selection

flattening

Demo: danceduo

- Motion-ness



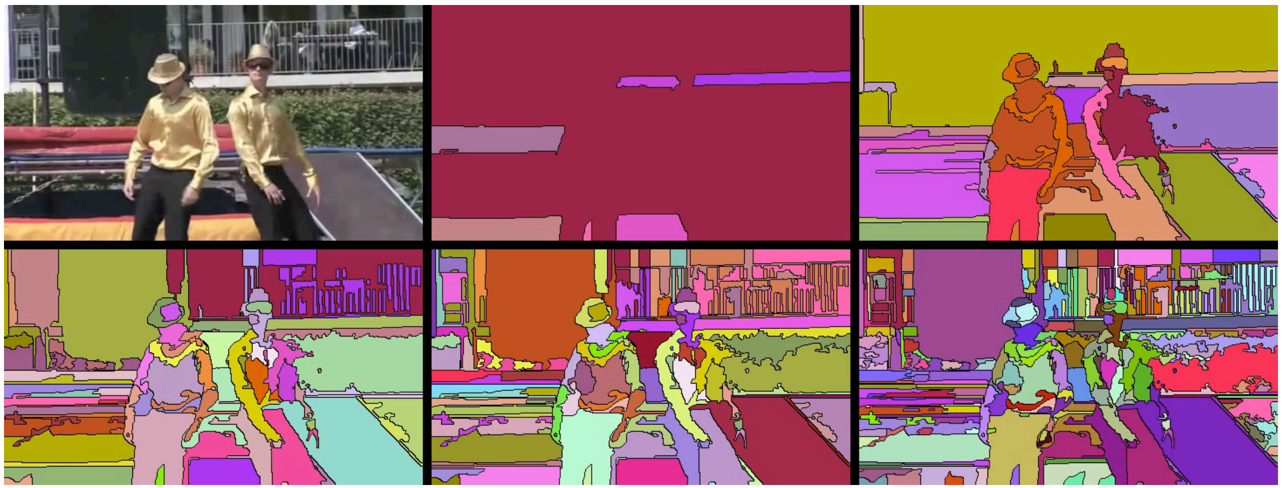
flow feature

selection

flattening

Demo: danceduo

- Object-ness



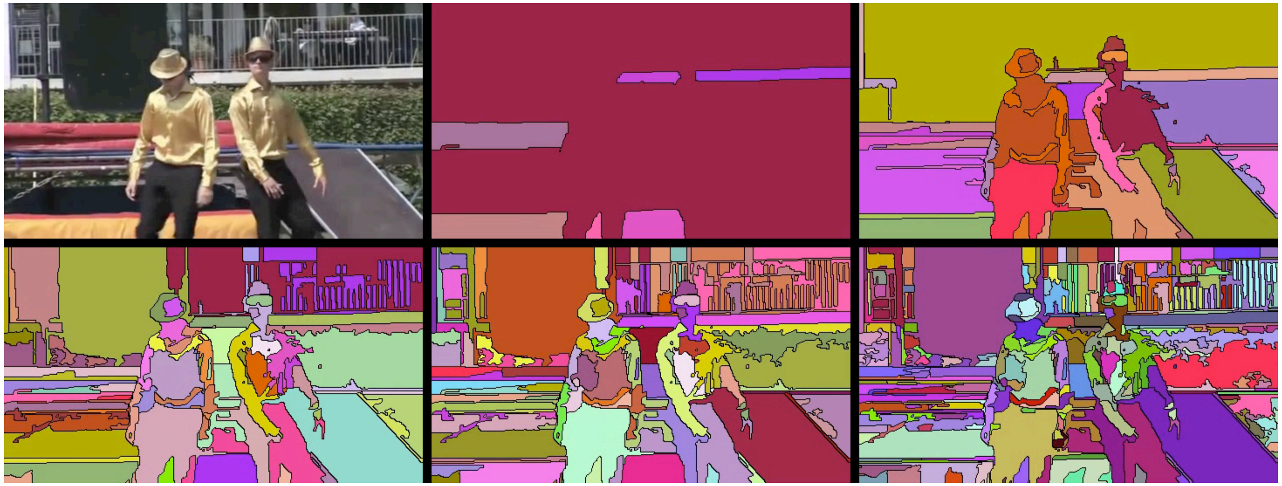
object-ness

selection

flattening

Demo: danceduo

- Human-ness



human detection

selection

flattening

LIBSVX v3.1: Updates & Recognition Task

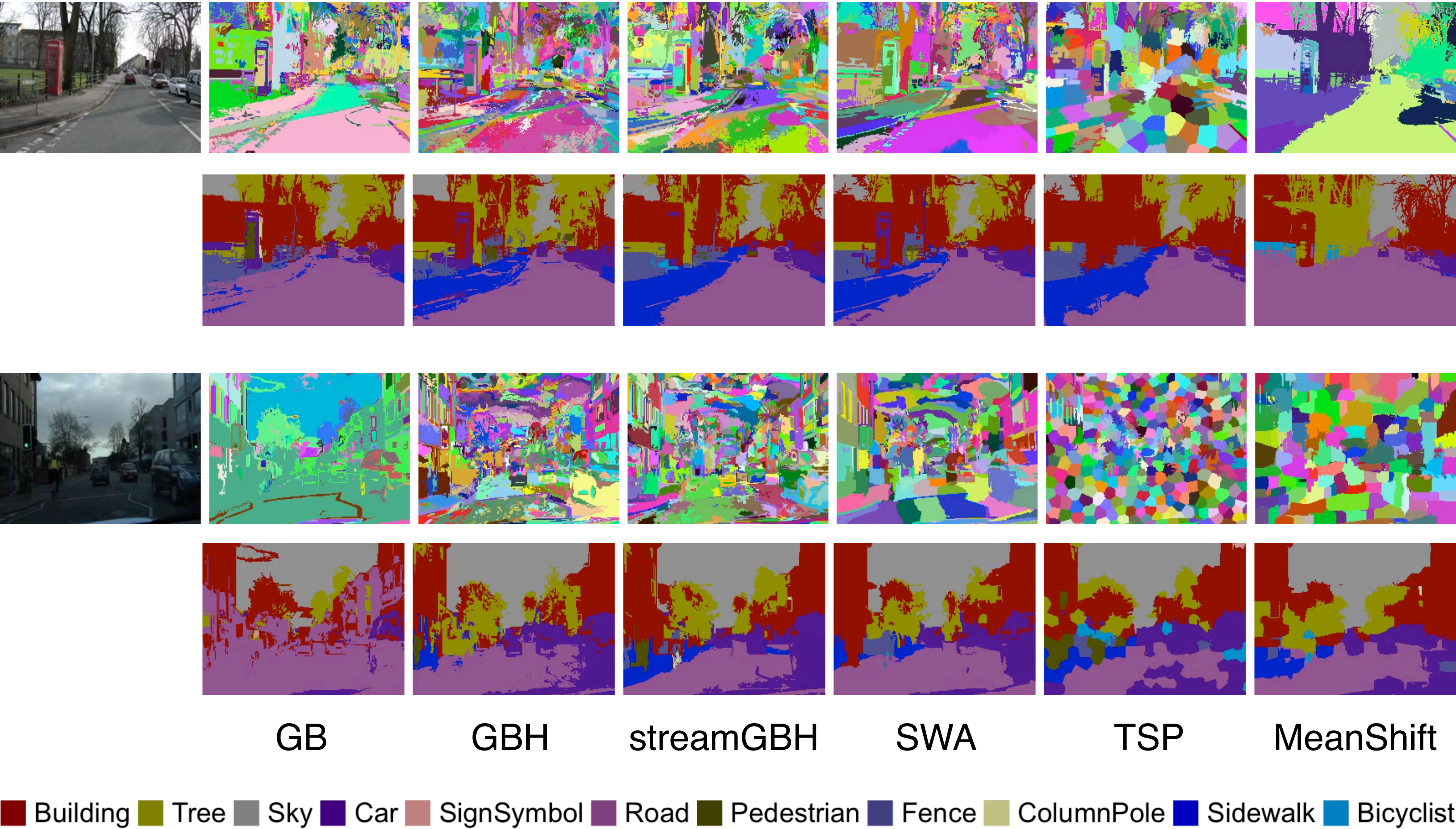
LIBSVX v3.1: Updates & Recognition Task

- New Datasets:
  - SegTrack v2 – Updated version of the SegTrack and provides frame-by-frame pixel-level multiple foreground objects labeling. It contains 14 video sequences. *[Li et al. ICCV 2013]*
  - BVDS – 100 videos with multiple human annotations by a sampling rate of every 20 frames. *[Galasso et al. ICCV 2013]*
  - CamVid – 18K frames with labeled 11 semantic object class labels at 1 Hz and in part 15 Hz. *[Brostow et al. ECCV 2008]*

- New Metrics:
  - Supervoxel Size Variation. *[Chang et al. CVPR 2013]*

LIBSVX v3.1: Updates & Recognition Task

- Recognition Task on CamVid dataset.

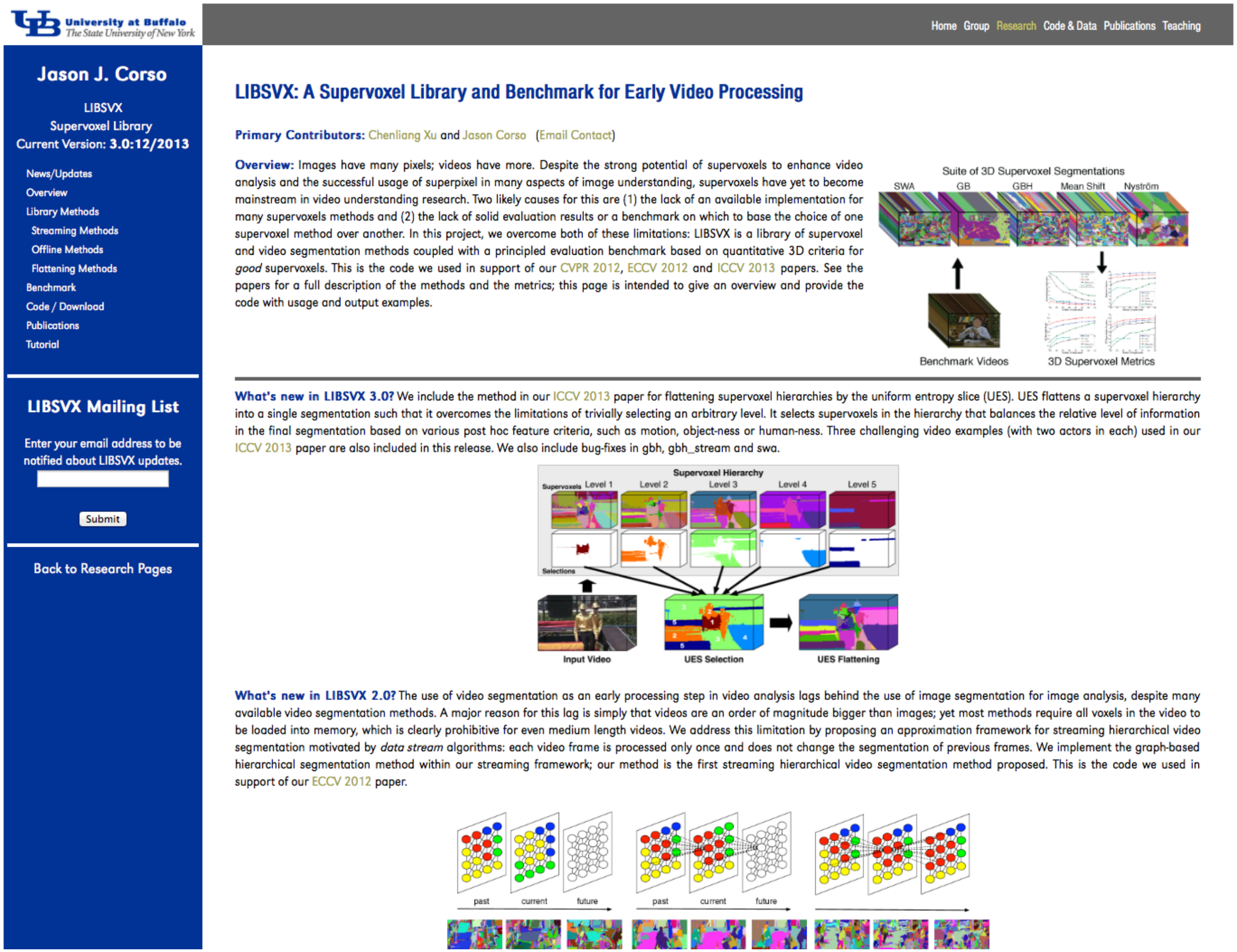


LIBSVX v3.1: Updates & Recognition Task

- Speed up the evaluation benchmark.
- Of course: Bug fix to methods!
- Planned release date: Fall 2014.

Thank you!

- Download LIBSVX: www.supervoxel.com



University at Buffalo
The State University of New York

Home Group Research Code & Data Publications Teaching

Jason J. Corso

LIBSVX
Supervoxel Library
Current Version: **3.0:12/2013**

- News/Updates
- Overview
- Library Methods
  - Streaming Methods
  - Offline Methods
  - Flattening Methods
- Benchmark
- Code / Download
- Publications
- Tutorial

LIBSVX Mailing List

Enter your email address to be notified about LIBSVX updates.

Submit

Back to Research Pages

LIBSVX: A Supervoxel Library and Benchmark for Early Video Processing

Primary Contributors: Chenliang Xu and Jason Corso (Email Contact)

Overview: Images have many pixels; videos have more. Despite the strong potential of supervoxels to enhance video analysis and the successful usage of superpixel in many aspects of image understanding, supervoxels have yet to become mainstream in video understanding research. Two likely causes for this are (1) the lack of an available implementation for many supervoxels methods and (2) the lack of solid evaluation results or a benchmark on which to base the choice of one supervoxel method over another. In this project, we overcome both of these limitations: LIBSVX is a library of supervoxel and video segmentation methods coupled with a principled evaluation benchmark based on quantitative 3D criteria for *good* supervoxels. This is the code we used in support of our CVPR 2012, ECCV 2012 and ICCV 2013 papers. See the papers for a full description of the methods and the metrics; this page is intended to give an overview and provide the code with usage and output examples.

Suite of 3D Supervoxel Segmentations
SWA GB GBH Mean Shift Nyström
Benchmark Videos
3D Supervoxel Metrics

What's new in LIBSVX 3.0? We include the method in our ICCV 2013 paper for flattening supervoxel hierarchies by the uniform entropy slice (UES). UES flattens a supervoxel hierarchy into a single segmentation such that it overcomes the limitations of trivially selecting an arbitrary level. It selects supervoxels in the hierarchy that balances the relative level of information in the final segmentation based on various post hoc feature criteria, such as motion, object-ness or human-ness. Three challenging video examples (with two actors in each) used in our ICCV 2013 paper are also included in this release. We also include bug-fixes in gbh, gbh_stream and swa.

Supervoxel Hierarchy
Supervoxels Level 1 Level 2 Level 3 Level 4 Level 5
Selections
Input Video UES Selection UES Flattening

What's new in LIBSVX 2.0? The use of video segmentation as an early processing step in video analysis lags behind the use of image segmentation for image analysis, despite many available video segmentation methods. A major reason for this lag is simply that videos are an order of magnitude bigger than images; yet most methods require all voxels in the video to be loaded into memory, which is clearly prohibitive for even medium length videos. We address this limitation by proposing an approximation framework for streaming hierarchical video segmentation motivated by *data stream* algorithms: each video frame is processed only once and does not change the segmentation of previous frames. We implement the graph-based hierarchical segmentation method within our streaming framework; our method is the first streaming hierarchical video segmentation method proposed. This is the code we used in support of our ECCV 2012 paper.

past current future past current future