Semantic (less) Motion and Video Segmentation

René Vidal
Johns Hopkins University
Talk Outline

- **Semantic-less Motion Segmentation** (Vidal et al., ECCV02, IJCV06; Vidal, Ma and Sastry CVPR03, PAMI05; Vidal and Sastry CVPR03; Vidal and Ma ECCV04, JMIV06; Vidal and Hartley, CVPR04; Tron and Vidal, CVPR07; Li et al. CVPR07; Goh and Vidal CVPR07; Vidal and Hartley, PAMI08; Vidal et al. IJCV08; Rao et al. CVPR 08, PAMI 09; Elhamifar and Vidal, CVPR 09)

- **Coarse-to-Fine Semantic Video Segmentation** (Jain et al. ICCV 2013)
Part I
Semantic-less Motion Segmentation

E. Elhamifar, A. Goh, R. Tron, S. Rao, R. Hartley, Y. Ma, S. Soatto, S. Sastry
René Vidal
Johns Hopkins University
2D Motion Segmentation Problem
Prior Work on 2D Motion Segmentation

• Cluster locally estimated models (Wang-Adelson '93-'94)

• Fit one dominant motion at a time (Irani-Peleg '92)

• Fit a mixture model (Jepson-Black'93, Ayer-Sawhney '95, Darrel-Pentland'95, Weiss-Adelson'96, Weiss'97, Torr-Szeliski-Anandan '99, Khan-Sha’01)

• Apply normalized cuts to motion profile (Shi-Malik ’98)
• Motion of a rigid-body lives in 3D affine subspace (Boult and Brown ’91, Tomasi and Kanade ’92)
  - \( P = \#\text{points} \)
  - \( F = \#\text{frames} \)

\[
W = M S^T
\]

\[
\begin{bmatrix}
\mathbf{x}_{11} & \cdots & \mathbf{x}_{1P} \\
\vdots & \ddots & \vdots \\
\mathbf{x}_{F1} & \cdots & \mathbf{x}_{FP}
\end{bmatrix}_{2F \times P} =
\begin{bmatrix}
A_1 \\
\vdots \\
A_F
\end{bmatrix}_{2F \times 4}
\begin{bmatrix}
\mathbf{X}_1 & \cdots & \mathbf{X}_P
\end{bmatrix}_{4 \times P}
\]
Prior Work on 3D Motion Segmentation

- **Iterative methods**
  - K-subspaces (Bradley-Mangasarian '00, Kambhatla-Leen '94, Tseng '00, Agarwal-Mustafa '04, Zhang et al. '09, Aldroubi et al. '09)

- **Probabilistic methods**
  - Mixtures of PPCA (Tipping-Bishop '99, Grubber-Weiss '04, Kanatani '04, Archambeau et al. '08, Chen '11)
  - Agglomerative Lossy Compression (Ma et al. '07, Rao et al. '08)
  - RANSAC (Leonardis et al. '02, Yang et al. '06, Haralik-Harpaz '07)

- **Algebraic methods**
  - Factorization (Boult-Brown '91, Costeira-Kanade '98, Gear '98, Kanatani et al. '01, Wu et al. '01)
  - Generalized PCA: (Shizawa-Maze '91, Vidal et al. '03 '04 '05, Huang et al. '05, Yang et al. '05, Derksen '07, Ma et al. '08, Ozay et al. '10)

- **Spectral clustering-based methods** (Zelnik-Manor '03, Yan-Pollefeys '06, Govindu '05, Agarwal et al. '05, Fan-Wu '06, Goh-Vidal '07, Chen-Lerman '08, Elhamifar-Vidal '09 '10, Lauer-Schnorr '09, Zhang et al. '10, Liu et al. '10, Favaro et al. '11, Candes '12)
How to Define a Good Subspace Affinity?

- **Spectral clustering**
  - Represent points as nodes in graph $G$
  - Connect points $i$ and $j$ with weight $c_{ij}$
  - Infer clusters from Laplacian of $G$

- **Good affinity matrix $C$ for subspaces?**
  - $c_{i,j} = \exp(-d^2(y_i, y_j))$
  - Points in the same subspace: $c_{ij} \neq 0$
  - Points in different subspaces: $c_{ij} = 0$

- **Challenge:** cannot define a pairwise affinity

- **Multiway affinity based on** $d+1$ or $d+2$ points (Chen-Lerman ’08)

- **Affinity based on** angles between local subspaces (Yan-Pollefeys ’06)
Sparse Subspace Clustering (SSC)

- Data in a union of subspaces are self-expressive
  \[ y_i = \sum_{j=1}^{N} c_{ji} y_j \implies y_j = Y c_i \implies Y = Y C \]

- Data in a union of subspaces admit a subspace-sparse representation

- The affinity can be constructed using L1 minimization
  \[ P_1 : \min ||c_i||_1 \; \text{s.t.} \; y_i = Y c_i, \; c_{ii} = 0 \]
Hopkins 155 motion segmentation database

- **Collected 155 sequences** (Tron-Vidal ’07)
  - 120 with 2 motions
  - 35 with 3 motions

- **Types of sequences**
  - **Checkerboard sequences**: mostly full dimensional and independent motions
  - **Traffic sequences**: mostly degenerate (linear, planar) and partially dependent motions
  - **Articulated sequences**: mostly full dimensional and partially dependent motions

- **Point correspondences**
  - In few cases, provided by Kanatani & Pollefeys
  - In most cases, extracted semi-automatically with OpenCV

Results on the Hopkins 155 database

- **2 motions, 120 sequences, 266 points, 30 frames**

<table>
<thead>
<tr>
<th></th>
<th>GPCA</th>
<th>LLMC</th>
<th>LSA</th>
<th>RANSAC</th>
<th>MSL</th>
<th>SCC</th>
<th>ALC</th>
<th>SSC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Checkerboard</strong></td>
<td>6.09</td>
<td>3.96</td>
<td>2.57</td>
<td>6.52</td>
<td>4.46</td>
<td>1.30</td>
<td>1.55</td>
<td><strong>1.12</strong></td>
</tr>
<tr>
<td><strong>Traffic</strong></td>
<td>1.41</td>
<td>3.53</td>
<td>5.43</td>
<td>2.55</td>
<td>2.23</td>
<td>1.07</td>
<td>1.59</td>
<td><strong>0.02</strong></td>
</tr>
<tr>
<td><strong>Articulated</strong></td>
<td>2.88</td>
<td>6.48</td>
<td>4.10</td>
<td>7.25</td>
<td>7.23</td>
<td>3.68</td>
<td>10.70</td>
<td><strong>0.62</strong></td>
</tr>
<tr>
<td><strong>All</strong></td>
<td>4.59</td>
<td>4.08</td>
<td>3.45</td>
<td>5.56</td>
<td>4.14</td>
<td>1.46</td>
<td>2.40</td>
<td><strong>0.82</strong></td>
</tr>
</tbody>
</table>

- **3 motions, 35 sequences, 398 points, 29 frames**

<table>
<thead>
<tr>
<th></th>
<th>GPCA</th>
<th>LLMC</th>
<th>LSA</th>
<th>RANSAC</th>
<th>MSL</th>
<th>SCC</th>
<th>ALC</th>
<th>SSC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Checkerboard</strong></td>
<td>31.95</td>
<td>8.48</td>
<td>5.80</td>
<td>25.78</td>
<td>10.38</td>
<td>5.68</td>
<td>5.20</td>
<td><strong>2.97</strong></td>
</tr>
<tr>
<td><strong>Traffic</strong></td>
<td>19.83</td>
<td>6.04</td>
<td>25.07</td>
<td>12.83</td>
<td>1.80</td>
<td>2.35</td>
<td>7.75</td>
<td><strong>0.58</strong></td>
</tr>
<tr>
<td><strong>Articulated</strong></td>
<td>16.85</td>
<td>9.38</td>
<td>7.25</td>
<td>21.38</td>
<td>2.71</td>
<td>10.94</td>
<td>21.08</td>
<td><strong>1.42</strong></td>
</tr>
<tr>
<td><strong>All</strong></td>
<td>28.66</td>
<td>8.04</td>
<td>9.73</td>
<td>22.94</td>
<td>8.23</td>
<td>5.31</td>
<td>6.69</td>
<td><strong>2.45</strong></td>
</tr>
</tbody>
</table>

- **All**

<table>
<thead>
<tr>
<th></th>
<th>GPCA</th>
<th>LLMC</th>
<th>LSA</th>
<th>RANSAC</th>
<th>MSL</th>
<th>SCC</th>
<th>ALC</th>
<th>LRR</th>
<th>LRSC</th>
<th>SSC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>All</strong></td>
<td>10.34</td>
<td>4.97</td>
<td>4.94</td>
<td>9.76</td>
<td>5.03</td>
<td>2.33</td>
<td>3.37</td>
<td>3.16</td>
<td>3.28</td>
<td><strong>1.24</strong></td>
</tr>
</tbody>
</table>
Dense 3D Motion Segmentation

- **BMS-26** (Brox-Malik’10)
  - 26 video sequences with pixel-accurate segmentation annotation of moving objects
  - 12 sequences are taken from the Hopkins 155 dataset

- **FBMS-59** (Ochs’14)

---

T. Brox, J. Malik. Object segmentation by long term analysis of point trajectories, ECCV 2010
P. Ochs, J. Malik, and T. Brox. Segmentation of moving objects by long term video analysis, PAMI 2014
Dense 3D Motion Segmentation

• Sparse trajectory clustering:
  – Spectral clustering based on pairwise motion affinities

• Dense segmentation
  – Variational approach based on color, texture, etc.

T. Brox, J. Malik Object segmentation by long term analysis of point trajectories, ECCV 2010
P. Ochs, J. Malik, and T. Brox. Segmentation of moving objects by long term video analysis, PAMI 2013
Future Vistas in 3D Motion Segmentation

- **Good progress in the last decades**
  - Sparse trajectories
  - Complete trajectories
  - Short videos
  - Affine cameras

- **Ongoing and future directions**
  - Dense trajectories
  - Incomplete and corrupted trajectories
  - Appearing and disappearing objects
  - Longer videos
  - Static objects
  - Deformable objects
  - Strong perspective effects

(Doretto’03, Chan’05, ’09, Ghoreyshi-Vidal’06)

Coarse-to-fine Semantic Video Segmentation Using Supervoxel Trees

Aastha Jain
LinkedIn

Shaunak Chatterjee
UC Berkeley

René Vidal
Johns Hopkins
Semantic Video Segmentation Problem

• Given a video sequence, assign a class label to each pixel

SUNY Dataset. Chen et al. Propagating multi-call pixel labels throughout video frames, WNYIPW 2010
Computational Challenges

\[ V = \text{number of supervoxels} \]
\[ L = \text{number of labels} \]
\[ O(L^V) \text{ possible segmentations} \]

- Existing energy minimization approaches trade-off accuracy for efficiency by finding an approximate solution
  - Graph cuts [Boykov et al. TPAMI01]
  - Belief propagation [Felzenszwalb-Huttenlocher IJCV06]
  - Hierarchical graph cuts [Kumar UIA09]

- While successful for many tasks in image segmentation, these approximate methods continue to be very slow for applications in video segmentation

- How to perform efficient semantic video segmentation?
Proposed Approach

• Observations
  – Real videos are spatially and temporally coherent
  – Set of coherent labelings is much smaller than the set of all labelings

• Approach
  – Construct a hierarchy of supervoxels
  – Propose a coarse-to-fine energy minimization strategy

• Advantages
  – Exact: it gives the same solution as minimizing over the finest graph
  – General: it can be used with any supervoxel hierarchy and any energy minimization algorithm to minimize any energy function
  – Efficient: it gives 2x-10x speedup for several datasets with varying degrees of spatio-temporal coherence
Energy Minimization Problem

object categories

\[ l \in \mathcal{L} = \{1, \ldots, L\} \]

labels:

\[ x_i \in \mathcal{L} \]

supervoxels

\[
E(x) = \lambda_U \sum_{v_i \in \mathcal{V}} \psi_i^U (x_i, V) + \lambda_P \sum_{e_{ij} \in \mathcal{E}} \psi_{ij}^P (x_i, x_j, V) + \lambda_H \sum_{c \in \mathcal{C}} \psi_c^H (x_c, V)
\]

- \( \psi_i^U (l, I) \): cost of assigning label \( l \) to supervoxel \( i \)
- \( \psi_{ij}^P (l_1, l_2, I) \): cost of assigning labels \( l_1 \) and \( l_2 \) to supervoxels \( i \) and \( j \)
- \( \psi_c^H (x_c, I) \): label consistency cost for clique \( c \in \mathcal{C} \)
Hierarchy of Supervoxels

- **Supervoxel Based Methods** [Xu and Corso CVPR12]
  - SWA [Sharon CVPR00], Graph Based [Felzenszwalb IJCV04], Hierarchical [Grundmann CVPR10], Mean Shift [Paris CVPR07], Nystom [Fowlkes TPAMI04]

![Hierarchy of Supervoxels](image)

Original image  | Level 5 (coarsest)  | Level 4
--- | --- | ---
Level 3  | Level 2  | Level 1 (finest)
Coarse-to-Fine Energy Minimization

Level 4

Level 3

Level 2

Level 1
Current = Level 4

Level 3

Mixed

Pure

Iteration 1

Next

Refine
Keep refining supervoxels with the mixed label until all supervoxels are pure
Exactness of the Coarse-to-Fine Solution

**Theorem.** If the coarse potentials in $E_{\mathcal{V}_{\text{curr}}}$ are lower bounds of their constituent exact potentials, the set of minimizers of the coarse-to-fine procedure (with algorithm $A$ in step 3) is the same as that of running algorithm $A$ at the finest level.

**Algorithm 1** Coarse-to-fine Inference Algorithm ($\mathcal{V}^{1:m}, \psi$)

1: \( \mathcal{V}_{\text{curr}} \leftarrow \mathcal{V}^m \)
2: \textbf{repeat}
3: \hspace{1em} Find \( x_{\mathcal{V}_{\text{curr}}} \) which minimizes \( E_{\mathcal{V}_{\text{curr}}} \)
4: \hspace{1em} \textbf{for all} \( v_{ij} \in \mathcal{V}_{\text{curr}} \) such that \( x_{ij} = L + 1 \) \textbf{do}
5: \hspace{2em} Refine \( v_{ij} \)
6: \hspace{2em} \( \mathcal{V}_{\text{curr}} \leftarrow \mathcal{V}_{\text{curr}} \cup \mathcal{R}(i, j, j - 1) \setminus v_{ij} \)
7: \hspace{1em} \textbf{end for}
8: \hspace{1em} \textbf{until} \( L + 1 \notin x_{\mathcal{V}_{\text{curr}}} \)
9: \hspace{1em} \textbf{return} \( x_{\mathcal{V}_{\text{curr}}} \)
Construction of the Coarse Potentials

• Consider the energy at the finest level (level 1)

\[ E(x) = \lambda_U \sum_{v_i \in V} \psi^U_i (x_i, V) + \lambda_P \sum_{e_{ij} \in E} \psi^P_{i,j} (x_i, x_j, V) + \lambda_H \sum_{c \in C} \psi^H_c (x_c, V) \]

• Unary cost for a coarse supervoxel at level \( j \)
  
  – Pure label: sum of the unary costs of constituent supervoxels at level 1
  
  – Mixed label: minimum cost over constituent supervoxels at level 1 subject to all the constituent supervoxels not getting the same label

• Pairwise cost
  
  – Pure label: sum of the pairwise costs of the edges connecting the constituent supervoxels
  
  – Mixed label: zero
Experiments: Datasets

• SUNY
  - 24 classes, 2 in each video, 70 training frames, 100 testing frames

• CamVid
  - 11 classes, 100 training frames, 100 testing frames
Experiments: Quantitative Results

• Time taken by the different inference algorithms (in minutes)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>CamVid1</th>
<th>CamVid2</th>
<th>CamVid3</th>
<th>CamVid4</th>
<th>CamVid5</th>
<th>SUNY Bus</th>
<th>SUNY Football</th>
<th>SUNY Ice</th>
</tr>
</thead>
<tbody>
<tr>
<td>GC Flat</td>
<td>130.1</td>
<td>137.3</td>
<td>117.6</td>
<td>145.1</td>
<td>140.1</td>
<td>35.3</td>
<td>25.0</td>
<td>32.7</td>
</tr>
<tr>
<td>Coarse-to-fine</td>
<td>32.7</td>
<td>40.9</td>
<td>27.3</td>
<td>43.8</td>
<td>29.4</td>
<td>6.5</td>
<td>2.3</td>
<td>5.3</td>
</tr>
<tr>
<td>BP Flat</td>
<td>256.0</td>
<td>270.1</td>
<td>258.3</td>
<td>307.0</td>
<td>319.2</td>
<td>50.3</td>
<td>34.7</td>
<td>50.9</td>
</tr>
<tr>
<td>Coarse-to-fine</td>
<td>50.5</td>
<td>79.1</td>
<td>61.5</td>
<td>107.7</td>
<td>90.5</td>
<td>9.3</td>
<td>4.1</td>
<td>8.3</td>
</tr>
</tbody>
</table>

• Computational speedup
  – CamVid: 3x-5x (2x-4x with time to compute hierarchy)
  – SUNY: 7x-10x (5x-6x with time to compute hierarchy)

• Percentage of time spent on bound computation
  – Graph cut: 40-50%
  – Belief propagation: 20-25%
Experiments: Qualitative Results

- Reduced problem size

![Figure 2](image)

Figure 2. Explored portions of the supervoxel tree. The blacked out portions in each superpixel level denotes the patch of superpixels which were never refined during inference. The top row shows results from the “football” video, the middle row from the “bus” video and the bottom row from the “ice” video (all from the SUNY dataset).

Table 1. Time taken by the different inference algorithms on different data sets (in minutes). The times reported for the coarse-to-fine case do not include supervoxel tree computation time. For the CamVid videos, the speedup is between 3x–5x, while for the SUNY videos the speedup is between 7x–10x. If we include the time of the on-demand refinement scheme discussed in Section 3.5, the overall speedup reduces to 2x–4x for CamVid and 5x–6x for the SUNY videos.
Experiments: Qualitative Results

- Segmentation accuracy versus number of refinement cycles

Figure 2. Explored portions of the supervoxel tree. The blacked out portions in each superpixel level denotes the patch of superpixels which were never refined during inference. The top row shows results from the "football" video, the middle row from the "bus" video and the bottom row from the "ice" video (all from the SUNY dataset).

Table 1. Time taken by the different inference algorithms on different data sets (in minutes). The times reported for the coarse-to-fine case do not include supervoxel tree computation time. For the CamVid videos, the speedup is between $3x$–$5x$, while for the SUNY videos the speedup is between $7x$–$10x$. If we include the time of the on-demand refinement scheme discussed in Section 3.5, the overall speedup reduces to $2x$–$4x$ for CamVid and $5x$–$6x$ for the SUNY videos.

Figure 3. Percentage of correctly classified supervoxels after every iteration of the coarse-to-fine belief propagation algorithm.
Discussion

• An exact, general and efficient coarse-to-fine energy minimization strategy for semantic video segmentation
  
  – It produces the same set of solutions as minimizing over the finest graph
  
  – It can be used with several energy minimization and hierarchy construction algorithms
  
  – It gives a 2x-10x speedup relative to flat algorithm

• Advances in energy minimization or hierarchy construction algorithms will only improve the efficiency of our framework
Thank You!

Vision Lab @ Johns Hopkins University
http://www.vision.jhu.edu

Center for Imaging Science @ Johns Hopkins University
http://www.cis.jhu.edu