Problem Formulation

- Problem: given a video sequence containing multiple moving objects, segment the video into multiple motions.
- Solution: track feature points in the video, cluster feature trajectories according to their motions, and estimate one motion model from each group.
- Challenges:
  - Need to estimate multiple motion models, without knowing which data correspond to which model.
  - Trajectories may be incomplete due to occlusions.
  - Trajectories may be corrupted due to tracking errors.
- Assumptions:
  - Multiple views $f = \{1, \ldots, F\}$, $F \geq 3$
  - Number of motions $n$ is known.
  - Affine camera model $x_{fp} = A X_{w}$, $p = \{1, \ldots, P\}$.
  - For each single motion $x_{fp} = [x_{1p}, \ldots, x_{xp}]$.

- Motion segmentation as a subspace clustering problem:
  - Trajectories $w_{ij}$ (the columns of $W$) for a single motion lie in a 2, 3 or 4-dimensional subspace of $\mathbb{R}^{2F}$.
  - Multiple motions generate multiple motion subspaces.
- Related issues:
  - Dimensionality reduction
  - Segmentation strategy
  - Missing data

Segmentation

Generalized Principal Component Analysis (GPCA)

- Idea: fit a polynomial $\varphi(w)$ to the union of subspaces and factorize $\varphi(w)$ to segment the individual subspaces.
- Algorithm:
  - Project the data onto $\mathbb{R}^t$ so that the motion subspaces become hyperplanes having normals $\{n_{ij}\}_{i,j}$.
  - Estimate $\varphi(w)$ linearly (via the Veronese map).
  - Apply spectral clustering to segment the data using the angle between normal vectors as a similarity measure.
- Performance: fast for $n = 2$, but does not scale well with $n$.

Agglomerative Lossy Coding (ALC)

- Idea: model union subspaces as a mixture of Gaussians and find segmentation that minimizes code length.
- Algorithm:
  - Initialize each point as a separate group.
  - Iteratively merge pairs of groups that maximally reduce the code length.
  - Stop when a local minimum is found.
- Performance: state-of-the-art classification results, with polynomial execution time.

Previous Work

Multi-Stage Learning (MSL)

- Idea: initialize using a factorization method, and refine the segmentation using a series of EM algorithms.
  - Factorize $W = UFV^\top$, and let $Q = VV^\top$.
  - Initialize $Q$s if $p$ if points and $f$ are in different groups.
  - Refine segmentation using EM assuming degenerate motions and then general motions.
- Performance: slow, provably correct only in the case of independent subspaces, however works well in practice.

Local Subspace Analysis (LSA)

- Idea: nearby trajectories likely belong to the same object.
  - Locally fit a subspace to each trajectory and its neighbors.
  - Use spectral clustering on the subspace angles.
- Performance: resilient to outliers, but has problems with dependent motions due to intersecting subspaces.

RANSAC

- Idea: repeatedly fit a model to a small random sample of points and keep the model with largest consensus.
  - Randomly sample $d$ vectors, fit a subspace to them, and compute the number of inliers to this subspace.
  - Repeat multiple times and keep the model with the most inliers.
- Performance: works well when number of parameters, number of motions, and percentage of outliers are small.

Dimensionality reduction

- Idea: project data from $\mathbb{R}^m$ to $\mathbb{R}^d$ to reduce computations.
- Theorem: the number and dimensions of the subspaces are preserved with probability 1 if $d > d$.
- How to choose $d$?
  - Minimum dimension ($d_{\min} = 5$).
  - Sum of the individual subspace dimensions ($d_{\sum} = 4n$).
  - Sparsity preserving projection $d_{\text{DP}} = \min d$ such that $d \geq 24 \log(2F/d)$ with $k = 4$.

Outliers

Agglomerative Lossy Coding (ALC)

- Outliers are tracked features that do not correspond to any particular motion.
- Coding length for outliers is high.
- ALC segments them in numerous small groups which are easily detected.

Corrupted Trajectories

- Idea: if only some of the entries in the trajectories are wrong, use $\ell^1$ optimization to correct them.
- Similar to $\ell^1$-based completion, but also the error vector $e$ is recovered.
- Use $\ell^1$ optimization to find $c$ and $e$.

Outlier Detection

Comparison on the Hopkins 155 dataset

Table 1: Distribution of the number of points and frames

<table>
<thead>
<tr>
<th>Two motions</th>
<th>Three motions</th>
</tr>
</thead>
<tbody>
<tr>
<td># Seq: Points</td>
<td>Frames</td>
</tr>
<tr>
<td>Checkboard</td>
<td>78</td>
</tr>
<tr>
<td>Traffic</td>
<td>31</td>
</tr>
<tr>
<td>Articulated</td>
<td>11</td>
</tr>
<tr>
<td>All</td>
<td>120</td>
</tr>
</tbody>
</table>

Figure 1: Sample images from some sequences in the dataset with tracked points superimposed.

Table 2: Average performance on Hopkins 155

<table>
<thead>
<tr>
<th>Two motions</th>
<th>MSL</th>
<th>LSA</th>
<th>GPCA</th>
<th>ALC</th>
<th>ALCsp</th>
<th>ALCsp†</th>
</tr>
</thead>
<tbody>
<tr>
<td>Checkboard</td>
<td>4.40%</td>
<td>2.57%</td>
<td>0.89%</td>
<td>2.60%</td>
<td>1.43%</td>
<td>1.43%</td>
</tr>
<tr>
<td>Traffic</td>
<td>5.23%</td>
<td>3.43%</td>
<td>2.83%</td>
<td>1.73%</td>
<td>2.83%</td>
<td>1.73%</td>
</tr>
<tr>
<td>Articulated</td>
<td>7.23%</td>
<td>4.10%</td>
<td>2.88%</td>
<td>6.90%</td>
<td>10.70%</td>
<td>10.70%</td>
</tr>
<tr>
<td>All Sequences</td>
<td>4.14%</td>
<td>3.45%</td>
<td>4.59%</td>
<td>3.03%</td>
<td>3.03%</td>
<td>3.03%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Three motions</th>
<th>MSL</th>
<th>LSA</th>
<th>GPCA</th>
<th>ALC</th>
<th>ALCsp</th>
<th>ALCsp†</th>
</tr>
</thead>
<tbody>
<tr>
<td>Checkboard</td>
<td>10.38%</td>
<td>5.86%</td>
<td>11.95%</td>
<td>6.78%</td>
<td>5.00%</td>
<td>5.00%</td>
</tr>
<tr>
<td>Traffic</td>
<td>1.80%</td>
<td>23.07%</td>
<td>19.83%</td>
<td>4.01%</td>
<td>8.86%</td>
<td>8.86%</td>
</tr>
<tr>
<td>Articulated</td>
<td>2.71%</td>
<td>7.25%</td>
<td>16.83%</td>
<td>21.08%</td>
<td>21.08%</td>
<td>21.08%</td>
</tr>
<tr>
<td>All Sequences</td>
<td>2.71%</td>
<td>7.25%</td>
<td>26.06%</td>
<td>6.20%</td>
<td>6.20%</td>
<td>6.20%</td>
</tr>
</tbody>
</table>

Outlier Detection

Table 3: Outlier detection rates (3 sequences)

<table>
<thead>
<tr>
<th>Two motions</th>
<th>MSL</th>
<th>LSA</th>
<th>GPCA</th>
<th>ALC</th>
<th>ALCsp</th>
<th>ALCsp†</th>
</tr>
</thead>
<tbody>
<tr>
<td># Seq: Points</td>
<td>Frames</td>
<td>Outlier %</td>
<td>ALC</td>
<td>ALCsp</td>
<td>ALCsp†</td>
<td></td>
</tr>
<tr>
<td>Checkboard</td>
<td>19.41%</td>
<td>* + ALC</td>
<td>4.15%</td>
<td>* + ALCsp</td>
<td>0.62%</td>
<td></td>
</tr>
<tr>
<td>Traffic</td>
<td>19.41%</td>
<td>* + ALC</td>
<td>4.15%</td>
<td>* + ALCsp</td>
<td>0.62%</td>
<td></td>
</tr>
<tr>
<td>Articulated</td>
<td>19.41%</td>
<td>* + ALC</td>
<td>4.15%</td>
<td>* + ALCsp</td>
<td>0.62%</td>
<td></td>
</tr>
</tbody>
</table>

Publicly available benchmark for 3-D motion segmentation algorithms

- Idea: find a low-rank ($r$) factorization of the data that minimizes the error on the non-missing entries.
- Algorithm:
  - Given estimate of $B$, find $A$ and then vice versa.
  - Orthonormalize columns of $B$.
  - Iterate until convergence.
- Disadvantages: disadvantage of union of subspaces is ignored.

Power Factorization

- Idea: find a $\ell^1$ based completion.
- Idea: data in subspaces is low-dimensional and self-descriptive.
- Idea: if only some of the entries in the trajectories are wrong, use $\ell^1$ optimization to correct them.
- Use $\ell^1$ optimization to find $c$ and $e$.
- $\ell^1$ and $\ell^0$ are equivalent if $c$ is sparse enough.
- Use $c$ to find the missing entries of $w_p$.