Tracking People

A presentation of Deva Ramanan’s “Finding and Tracking People from the Bottom Up” and “Strike a Pose: Tracking People by Finding Stylized Poses”

Tracking People: Context

Motion Capture

Surveillance

Video Data Mining

Human-Computer Interfaces
Tracking People: Context

Ideally a human tracker should:
• Provide 3-D articulated joint estimates
• Provide estimates of static link lengths/twists
• Work when camera(s) are moving
• Robustly handle occlusions and varying illumination conditions
• Require no prior initialization
• Require no modifications to the environment
• Permit natural clothing without the need for markers
• Run in real-time, depending on the proposed application

Tracking People: Context

Ramanan’s tracker addresses:
• Provide 3-D articulated joint estimates
• Provide estimates of static link lengths/twists
• Work when camera(s) are moving
• Robustly handle occlusions and varying illumination conditions
• Require no prior initialization
• Require no modifications to the environment
• Permit natural clothing without the need for markers
• Run in real-time, depending on the proposed application
• Other Approaches
  – Background Subtraction
  – Condensation
  – Explicit Motion Models

• Ramanan’s Approach
  – Build up nine-segment appearance model of people directly from video
  – Cluster segments over time
  – Prune segments with unlikely dynamics

Finding and Tracking: Overview

• Find likely segment candidates
• Opportunistically learn appearance of clusters across time
• Use appearance models to refine track in more difficult frames

Nine-segment puppet used as a person model
Finding and Tracking: Segment Detection

- Initial detectors do not detect, they learn appearance opportunistically
- Very simple parallel line detector
- Suffers from false positives, clustering and pruning techniques necessary

Finding and Tracking: Overview

- Find likely segment candidates
- Opportunistically learn appearance of clusters across time
- Use appearance models to refine track in more difficult frames
Finding and Tracking: Segment Clustering

Detected Segments

Project onto LAB-space

Finding and Tracking: Segment Clustering

Detected Segments

Feature Vectors to Cluster
Finding and Tracking: Segment Clustering

Detect Torsos
Across ‘n’ frames

Cluster in LAB-space
Via mean-shift

Prune clusters with unlikely dynamics
Finding and Tracking: Overview

- Find likely segment candidates
- Opportunistically learn appearance of clusters across time
- Use appearance models to refine track in more difficult frames

Finding and Tracking: Tracking using learned appearance

Detect new torsos using learned torso appearance models
Finding and Tracking: Tracking using learned appearance

Find arms and legs near the detected torsos

- Final track incorporates segment clustering and pruning techniques to detect multiple individuals
- Self initializes
- Works across occlusions
- Tracks a long sequence
- Tracks independent of activity
- Robust to drift
Finding and Tracking: Probabilistic Framework

• How can these high level heuristics be formalized and implemented?

– Capture the relationships with a probabilistic model and cast tracking as an inference problem
Finding and Tracking: Probabilistic Framework

Idealized segments are contained in set of all detected segments.

Segment appearance is constant.

Variables:
- \( C_{\text{seg}} \): Constant underlying appearance feature vector
- \( P_i^{\text{seg}} \): Position (and orientation) of segment in frame \( i \)
- \( \text{Im}_i^{\text{seg}} \): Collection of observed feature vectors for each image patch in frame \( i \)

Finding and Tracking: Probabilistic Framework

Person model contains kinematic constraints.

Full model for a torso and arm subset.

Variables:
- \( C_{\text{seg}} \): Constant underlying appearance feature vector
- \( P_i^{\text{seg}} \): Position (and orientation) of segment in frame \( i \)
- \( \text{Im}_i^{\text{seg}} \): Collection of observed feature vectors for each image patch in frame \( i \)
Finding and Tracking: Probabilistic Framework

- The algorithm performs an approximate inference on the kinematic dependency model using the above trees
  - a),b) appearance models of a segment
  - c) bounded velocity motion model
  - d) enforces kinematic constraints

Finding and Tracking: Algorithm Details

- The algorithm infers with a single pass through each link in the model
- Torso-arm example:
  - Infer (cluster) on a) to find similar torso segments
  - Infer on c) (torso) to extract dynamically valid segments
  - Infer on tree d) to find arms near the torso track.
  - Infer on tree b) to find similar arm segments across frames
- Overall order:
  - Torso ->Lower Arm ->Upper Arm
  - ->Lower Leg ->Upper Leg
Strike a Pose: Key ideas

- Learn appearance of segments by finding the entire kinematic model in canonical poses at once
- Concentrates efforts on easy poses
  - Easy to detect
  - Easy to learn appearance from
- High precision, low recall
- Extends to allow for quickly moving cameras

Strike a Pose: Detecting Poses

- Basic model is a tree pictorial structure, as described earlier, that decomposes a person model into a shape model and an appearance model:

\[
\Pr(P_1...P_n | \text{Im}) \propto \prod_{(i, j) \in E} \Pr(P_i | P_j) \prod_{i=1}^{n} \Pr(\text{Im}(P_i))
\]

For each segment
Strike a Pose: Detecting Poses

- Initial segment detection:
  - Compute edge image
  - Calculate chamfer distance transform and bin by orientation
  - Convolve with rectangle ‘segments’
  - Perform non-maximal suppression on convolution result

- Adding global constraints:
  - Left and right legs should look similar: add the disparity in leg appearance (in LAB space) to the negative log probability
  - Force left and right legs to be far apart: discard samples where legs points are within distance ‘d’ of each other

- Segmentation score:
  - Create test pools inside and outside detected segment - add number of misclassified pixels to get segmentation score
Strike a Pose: Tracking

- Now use segment appearance models to compute the image likelihood
- Increase angle interval bounds between segments to allow for all reasonable configurations

Results and Comparisons

- Finding and Tracking
  - Bottom-up appearance learning
  - High recall, low accuracy initial detector
  - Uses a (simple) motion model
  - Learning phase is not dependent on poses

- Strike a Pose
  - Top-down high level appearance learning
  - Low recall, high accuracy initial detector
  - No need for a motion model
  - Assumption of pose necessary for learning phase
Results and Comparisons

Finding and Tracking - High recall, low precision

Strike a Pose: High precision, low recall (initially)
Results and Comparisons

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Torso</th>
<th>Arm</th>
<th>Leg</th>
</tr>
</thead>
<tbody>
<tr>
<td>J. Jacks</td>
<td>94.6</td>
<td>87.4</td>
<td>91.8</td>
</tr>
<tr>
<td>Walk</td>
<td>99.5</td>
<td>84.3</td>
<td>68.2</td>
</tr>
<tr>
<td>Street Pass</td>
<td>91.2</td>
<td>57.4</td>
<td>38.7</td>
</tr>
<tr>
<td>Weave Run</td>
<td>90.3</td>
<td>21.2</td>
<td>61.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Torso</th>
<th>Arm</th>
<th>Leg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseball</td>
<td>98.4</td>
<td>93.75</td>
<td>95.3</td>
</tr>
<tr>
<td>Skating</td>
<td>99.2</td>
<td>77.5</td>
<td>97.6</td>
</tr>
<tr>
<td>Lola</td>
<td>95.6</td>
<td>89.8</td>
<td>92.6</td>
</tr>
<tr>
<td>Park</td>
<td>79.1</td>
<td>29.2</td>
<td>58.3</td>
</tr>
</tbody>
</table>

Discussion

- **Strengths**
  - Works in spite of camera motion
  - Robust to drift
  - Auto-initializing
- **Weaknesses**
  - Not applicable to realtime applications
  - Makes use of many heuristics (too many?)
  - May have problems dealing with lighting changes
Finding and Tracking: Extra videos

Strike a Pose: Extra videos