Optimal Importance Sampling for Human Motion Tracking using Persistent Low-level Image Features

Matthew Jenkins
jenkins@robots.ox.ac.uk

Ian Reid
ian@robots.ox.ac.uk

Active Vision Laboratory
Robotics Research Group
University of Oxford

5 July 2006
Outline

- Motivation
- Existing Human Motion Tracking Systems
- Our Approach
  - Using Low-level Features in Human Motion Tracking
  - Crossover Operator for Human Motion Tracking
  - System Overview
- Initial Results
- Conclusion & Future Work
Motivation: Monocular Human Motion Tracking

- **Challenges**
  - High dimensional (~30D)
  - Poorly-constrained
  - Extremely multi-modal

- **Currently available commercial systems**
  - Expensive – calibrated cameras, studio setting
  - Marker-based

- **Approaches using computer vision**
  - Model-driven - “tracking”
  - Image-driven - “detection”
  - More recently a combination of the two
Existing Human Motion Tracking Systems: Model-driven

- Top-down
  - Full human body model
  - Traditional “tracking” approach:
    - Predict from frame-to-frame using motion model to limit search space

- Problems
  - Requires initialisation
  - No recovery from failure
  - Human motion is complex and difficult to model

Deutscher & Reid [CVPR '01]
Existing Human Motion Tracking Systems: Model-driven

- Deutscher & Reid [CVPR '01]
  - Full 3D model
  - Calibrated cameras
  - Likelihood – region and edge based
  - Particle filter framework
  - Annealing & crossover operator
  - BUT Abandons the notion of a probabilistic tracker

- Sminchisescu & Triggs [CVPR '03]
  - Gaussian mixture model
  - Deterministic optimisation to find local likelihood modes
  - Still essentially particle-based
  - Importance function with kinematic jumps
Existing Human Motion Tracking Systems: Image-driven

- Systems essentially detectors
- Learn a mapping from low-level image features to human body pose (e.g. Agarwal & Triggs [PAMI '06])

Problems
- Harder to impose temporal constraints
- Difficult to generate "representative" training data

Taken from Agarwal & Triggs [PAMI '06]
The Best of Both Worlds?

- Recently an intermediate approach has become popular
  - Detect features in the images (usually learned human body part detectors)
  - Combine these parts using a high-level body model

- Examples
  - Bowden et al. [ECCV '06] use boosted part detectors, combining the parts with RANSAC
  - Sigal et al. [CVPR '04] used part detections combined using Belief Propagation

- Often predicated on “reliable” part detectors (e.g. faces) firing correctly
Our Approach

- We believe there is still a rôle for a traditional tracking approach in such cases
- Our framework is a particle filter with importance sampling
- However, even the most effective particle-based approaches waste many particles

- Why place particles where there are no edges?
- Use persistent low-level features to guide the particle placement
Our Approach

- We believe there is still a rôle for a traditional tracking approach in such cases.
- Our framework is a particle filter with importance sampling.
- However, even the most effective particle-based approaches waste many particles.

- Why place particles where there are no edges?
- Use persistent low-level features to guide the particle placement.
Using Low-level Features in Human Motion Tracking

- Interested in seeing how much we can gain from just using low-level features, which are always present in the scene

Bibby & Reid [ECCV '06] Fast Radial Blob Detector
Our Approach (cont'd...)

- The choice of importance function is key:
  - It is well-known and widely publicised (e.g. Doucet et al. [S&C'00]) that the *optimal importance function* is given by:

\[
g(\cdot | \cdot) = p(x_k | z_k, x_{k-1})
\]

- In practice, importance function should take our current measurement into account

- Difficult in the traditional CONDENSATION framework
  - Local measurements
  - Can't take measurements until we've already sampled
Our Approach (cont'd...)

We aim to approximate this optimal importance function, making use of low-level image features to judiciously guide the placement of particles

**Two Main Ideas**

- Constrained prior
  - use a tracked low-level feature to alter the prior to sample as the feature predicts
- Combine samples using a genetic algorithm-style crossover operator
  - combining information from the various low-level features
Constrained Prior

- We identify the modes of “constrained priors”
  - Non-linear optimisation
  - Constrained by low-level feature location
- Sampling function is a GMM approximating
- We generate samples from multiple constrained priors
- We later combine these samples to approximate the optimal importance function

\[ p(x_k|z_k^{(c)}, x_{k-1}) \]
Constrained Prior

- We identify the modes of “constrained priors”
  - Non-linear optimisation
  - Constrained by low-level feature location
- Sampling function is a GMM approximating

\[ p(x_k|z_k^{(c)}, x_{k-1}) \]

- We generate samples from multiple constrained priors
- We later combine these samples to approximate the optimal importance function
Constrained Prior

- We identify the modes of “constrained priors”
  - Non-linear optimisation
  - Constrained by low-level feature location
- Sampling function is a GMM approximating
  \[ p(x_k|z^{(c)}_k, x_{k-1}) \]
- We generate samples from multiple constrained priors
- We later combine these samples to approximate the optimal importance function
Crossover Operator for Human Motion Tracking

- Samples from various “constrained priors” combined using a “crossover operator”
- Combine high-likelihood parts of the state-space from various samples to locate higher-scoring particles
- We use a strict importance-sampling framework to maintain probabilistically correct tracking
Importance Function

If \( p(x) = \sum_i w_i N(\mu_i, \Sigma_i) \) and \( x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \Rightarrow \mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \)

then, two independent samples from constrained prior (1) and (2), have state partitioned probabilities given by:

\[
p^{(1)}(x_1) = \sum_i w_i N(x_1; \mu^{(1)}_i, \Sigma^{(1)}_{i,11}) \\
p^{(2)}(x_2) = \sum_i w_i N(x_2; \mu^{(2)}_i, \Sigma^{(2)}_{i,22})
\]

Thus, we can write the full importance function for Genetic Algorithm sampling as:

\[
g(x|\cdot) = p(n_1, n_2) p(c_1) p(c_2) p^{(c_1)}(x_{n_1}) p^{(c_2)}(x_{n_2})
\]

where \( n_1 \) & \( n_2 \) represent a partitioning of the state space, and \( c_1 \) & \( c_2 \) are the chosen low-level constraints.
System Overview

Human Body Tracker

Start

Pose Initialisation → Predict Pose with Motion Prior → Constrain Prior
→ Likelihood Weighting, Resampling

Constrain Prior

Associate Features with Limbs

Low-level Feature Tracker

Detect Low-level Image features → Match Features in New Frame
Initial Results

- 8 degree-of-freedom system, fixed centre point
- Constant-position motion-model

<table>
<thead>
<tr>
<th>Original</th>
<th>Particle Filter</th>
<th>Our System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low-level Harris corner detector</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
- Patch-matching for low-level feature tracking
Conclusions & Future Work

**Summary**

- Optimal Importance Function takes current measurement into account
- Combine multiple constrained priors using crossover operator to approximate optimal function
- Can define an appropriate importance function to retain probabilistic nature of tracking

**Future Work**

- Full system with multiple low-level trackers and separable likelihood for efficient sample combination
- Intelligent partitioning
- Mode clustering to increase efficiency
- Use high-level knowledge for 2D to 3D conversion