Real-Time Human Pose Recognition in Parts from Single Depth Images

Jamie Shotton et al.
Microsoft Research Cambridge & Xbox Incubation
CVPR 2011 Best Paper Award
Purpose

Compute 3D joint positions from a single depth image using Kinect

Key features: *(Why best paper)*
1) Single depth image, robust
2) 200 f/s on Xbox 360 GPU, fast
3) General people with general motions, general
A commercial used algorithm

1) The core algorithm in Kinect games
2) Has been included in the Microsoft Kinect SDK
Background: human motion

• Human motion recovery has been studied extensively in the past 20 years.
• Extracting 3D human poses from images (especially monocular images) is an extremely difficult task.

• Main difficulties:
  1) Depth 3D-2D Projection Ambiguities
Background: human motion

- Human motion recovery has been studied extensively in the past 20 years.
- Extracting 3D human poses from images (especially monocular images) is an extremely difficult task.

- Main difficulties:
  2) General poses

![General poses](image)
Background: human motion

- Human motion recovery has been studied extensively in the past 20 years.
- Extracting 3D human poses from images (especially monocular images) is an extremely difficult task.

- Main difficulties:
  3) Different body sizes
Background: human motion

- Human motion recovery has been studied extensively in the past 20 years.
- Extracting 3D human poses from images (especially monocular images) is an extremely difficult task.

- Main difficulties:
  4) Motion blur
Background: state-of-the-art
Background: state-of-the-art

Generative examples: Anneal Particle Filters

Background: state-of-the-art

Discriminative examples: Relevance Vector Regression

Assumption

An entertainment scenario
- indoor
- 1-5m from the Kinect
- dancing, kicking, running, et al.
Pipeline

Input depth image → Inferred body parts → 3D joint point proposals
Appearance ambiguity

Differences in local appearance within one body part
Motion capture data

- 500k frames in a few hundred sequences.
- Use a subset of 100k poses that no two poses are closer than 5cm.
Synthetic data

Base character models
Synthetic data

Example training images
Randomized decision forests

Randomized decision forests

Depth image features

\[ f_\theta(I, x) = d_I \left( x + \frac{u}{d_I(x)} \right) - d_I \left( x + \frac{v}{d_I(x)} \right) \]
Randomized decision forests

Depth image features

\[ f_\theta(I, x) = d_I \left( x + \frac{u}{d_I(x)} \right) - d_I \left( x + \frac{v}{d_I(x)} \right) \]
Randomized decision forests

Tree training

- The completely random approach
  Choose parameters randomly.

Randomized decision forests

Tree training

\[
\begin{align*}
    f_\theta(I, x) &= d_I \left( x + \frac{u}{d_I(x)} \right) - d_I \left( x + \frac{v}{d_I(x)} \right) \\
    \text{If } f_\theta(I, x) < \tau & \text{ go to left child} \\
    \text{otherwise } & \text{ go to right child}
\end{align*}
\]

- The greedy algorithm
  Best separate the given examples.

Randomized decision forests

Tree training

1. Randomly propose a set of splitting candidates \( \phi = (\theta, \tau) \) (feature parameters \( \theta \) and thresholds \( \tau \)).
2. Partition the set of examples \( Q = \{(I, x)\} \) into left and right subsets by each \( \phi \):
   \[
   Q_1(\phi) = \{ (I, x) \mid f_\theta(I, x) < \tau \} \quad \text{(3)}
   \]
   \[
   Q_r(\phi) = Q \setminus Q_1(\phi) \quad \text{(4)}
   \]
3. Compute the \( \phi \) giving the largest gain in information:
   \[
   \phi^* = \underset{\phi}{\operatorname{argmax}} G(\phi) \quad \text{(5)}
   \]
   \[
   G(\phi) = H(Q) - \sum_{s \in \{1, r\}} \frac{|Q_s(\phi)|}{|Q|} H(Q_s(\phi)) \quad \text{(6)}
   \]
   where Shannon entropy \( H(Q) \) is computed on the normalized histogram of body part labels \( l_I(x) \) for all \( (I, x) \in Q \).
4. If the largest gain \( G(\phi^*) \) is sufficient, and the depth in the tree is below a maximum, then recurse for left and right subsets \( Q_1(\phi^*) \) and \( Q_r(\phi^*) \).

Randomized decision forests

Tree training

3 trees, 20 deep, 300k training images per tree, 2000 training example pixels per image, 2000 candidate features, and 50 candidate thresholds per feature.

One day on a 1000 core cluster.

Joint position proposals

Inferred body parts \rightarrow 3D joint point proposals

A mean shift like method with a weighted Gaussian kernel

\[ f_c(\tilde{x}) \propto \sum_{i=1}^{N} w_{ic} \exp \left( - \frac{\| x - \tilde{x}_i \|^2}{b_c^2} \right) \]

\[ w_{ic} = P(c|I, x_i) \cdot d_I(x_i)^2 \]

Mean Shift Modality Analysis Example
Mean Shift Modality Analysis Example
Overall Speed

200 fps using simple clustering on the Xbox GPU
50 fps using mean shift on a modern 8 core desktop CPU
Experimental results
Conclusions

Contributions of this paper:

• A super real-time algorithm (200 f/s on Xbox GPU)

• A robust algorithm without using temporal coherence (tracking by detection)

• A synthetic training data generation method (depth image)

• A commercial used algorithm (it really works)
Thank you!