Human Skeleton Extraction
by Structured-light Sensors
Use kinect to detect and track humans. Get human position to avoid humans.
Humanoid Robot Control using Depth Camera

Halit Bener SUAY
Sonia CHERNOVA

WPI
Human Skeleton Extraction

Depth Image → Body Parts → Joint Position

IR Structured Light → Random Decision Forest → Mean Shift
Human Skeleton Extraction

Depth Image

Body Parts

Joint Position

IR Structured Light

Random Decision Forest

Mean Shift

Done
Human Skeleton Extraction

Depth Image

Body Parts

Joint Position

IR Structured Light

Random Decision Forest

Mean Shift
Human Skeleton Extraction

On Kinect v1

• Algorithms runs 5ms per frame on Xbox GPU
• Novelty: Intermediate body parts representation

Human Skeleton Extraction

Body parts inference procedures:
• Background-foreground segmentation
• Feature extraction
• Body-part classification
• Joint position estimation
**Background-Foreground Segmentation**

- Detect humans at the pixel level. For example, do a simple depth filtering.
Human Skeleton Extraction

Body parts inference procedures:

- Background-foreground segmentation
- Feature extraction
- Body-part classification
- Joint position estimation
Feature Extraction

• Depth image features

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

• \( d_I(x) \) is the depth at pixel \( x \) in image \( I \)
• \( \theta = (u, v) \) describe offsets \( u \) and \( v \)
• Each feature need only read at most 3 image pixels and perform at most 5 arithmetic operations
Human Skeleton Extraction

Body parts inference procedures:

- Background-foreground segmentation
- Feature extraction
- Body-part classification
- Joint position estimation
Body Part Classification

• Randomized Decision Forests

Randomized Decision Forests. A forest is an ensemble of trees.
Body Part Classification

• To understand a Randomized Decision Forest, we need to understand three concepts:
  • Decision tree
  • Forest
  • Randomized
Body Part Classification

- Decision Tree (an simple example)
Body Part Classification

• Decision tree

Each tree consists of split nodes (blue) and leaf nodes (green). The red arrows indicate the different paths that might be taken by different trees for a particular input.
Body Part Classification

• Train a Randomized Decision Forest
  A decision tree is like a pre-planned game of “twenty questions”

Twenty Questions: guess the person, place or thing in 20 questions or less!
Body Part Classification

• What kind of “questions” can the Kinect ask in its twenty questions?
  • Simplified version:
    • “Is the pixel at that offset in the background?”
  • Real version:
    • “how does the depth at that pixel compare to this pixel?”

\[ f_\theta(I, x) = d_I \left( x + \frac{u}{d_I(x)} \right) - d_I \left( x + \frac{v}{d_I(x)} \right) \]
Body Part Classification

• We want to choose as the next question the one that is most “useful”

• In practice, “useful” = information gain (which is derived from entropy)

• It is pointless for us to ask a question for which we already know the answer. In this situation, this information we get by asking the question would have very low entropy

• In our scenario, we need to decide a threshold of the feature value

\[ f_\theta(I, x) = d_I \left( x + \frac{u}{d_I(x)} \right) - d_I \left( x + \frac{v}{d_I(x)} \right) \]
Body Part Classification

• Classifier ensemble
  • Committees
    • Averaging the predictions of a set of individual models
    • E.g., Majority votes, weighted majority votes
  • Boosting
    • Classifiers trained in sequence
    • E.g., AdaBoost: subsequent decision makers are tweaked in favor of those data instances misclassified by previous decision makers.

Decision Forest Is
A Decision Tree Committee
Body Part Classification

• Randomized Decision Forests
  • Fast and effective multi-class classifier
  • Each split node consists of a feature and a threshold
  • At the leaf node in tree, given a learned $P_t(c|I, x)$
  • Final classification

\[ P(c|I, x) = \frac{1}{T} \sum_{t=1}^{T} P_t(c|I, x) \]
Body Part Classification

• Randomized:
  • Each tree train on different images
  • Each image pick 2000 example pixels
    (resolution of a depth image is 320x240)
Body Part Classification

• Randomized Decision Forest include three concepts:
  • Decision tree
  • Forest
  • Randomized
Body Part Classification

• Training the Random Decision Forests takes a lot of efforts (and data):

Learning the Kinect decision forest requires 24,000 CPU-hours, but takes only a day using hundreds of computers simultaneously.

“To keep the training times down we employ a distributed implementation. Training 3 trees to depth 20 from 1 million images takes about a day on a 1000 core cluster.”

—Shotton et al, CVPR(2011)

Where can we get those training data?
Body Part Classification

• Depth imaging
  • Simplifies the task of background subtraction
  • Most importantly: is easy to synthesize (generate the data as well as provide the ground truth)!!

![Diagram showing the process of taking real images, learning parameters, and generating lots of training data.](image)
**Body Part Classification**

- Body part labeling
  - 31 body parts
  - Distinct parts for left and right allow classifier to disambiguate the left and right sides of the body
Human Skeleton Extraction

Body parts inference procedures:
- Background-foreground segmentation
- Feature extraction
- Body-part classification
- Joint position estimation
Joint Position Estimation

• Joints are estimated using mean-shift (a fast mode-finding algorithm)
• Observed part center is offset by pre-estimated value
Joint Position Estimation

**Intuitive Description**

Objective: Find the densest region

Region of interest
Center of mass
Mean Shift vector
Joint Position Estimation

**Intuitive Description**

- **Objective**: Find the densest region

- **Region of interest**

- **Center of mass**

- **Mean Shift vector**
Joint Position Estimation

Intuitive Description

Objective: Find the densest region
Joint Position Estimation

**Intuitive Description**

**Objective**: Find the densest region
Joint Position Estimation

Intuitive Description

Objective: Find the densest region

Region of interest
Center of mass
Mean Shift vector
Joint Position Estimation

**Intuitive Description**

**Objective**: Find the densest region
Joint Position Estimation

**Intuitive Description**

**Objective**: Find the densest region
Joint Position Estimation

Results
Joint Position Estimation

Implementation

- 31 body parts
- 3 trees (depth 20)
- 300,000 training images per tree randomly selected from 1M training images
- 2,000 training example pixels per image
- 2,000 candidate features
- 50 candidate thresholds per feature
- Decision forest constructed in 1 day on 1,000 core cluster
Examples of skeletal human body models obtained from different devices. The OpenNI library tracks 15 joints; Kinect v1 SDK tracks 20 joints; Kinect v2 SDK tracks 25
Motion Capture Systems
Where to get skeleton data

1. Windows: Microsoft Kinect SDK
2. Linux: Robot Operating System (ROS)
3. Linux: OpenNI and Nite
Will robots hurt our jobs?
What (else) to perceive for HCR?
What to perceive?

- Humans
What to perceive?

• Humans
What to perceive?

• Human activities
What to perceive?

• Group behaviors
What to perceive?

• Human-object interaction
What to perceive?

• Objects
What to perceive?

• Scenes