Hand Pose Estimation

Matthew Krenik
Advisor: Fabrizio Pece
Agenda

- What is Hand Pose Estimation?
- Why does it matter?
- How does it work?
- What has been done?
What is Hand Pose Estimation?

- Estimate full Degree of Freedom (DOF) of a hand from depth images

- This is a tough problem, especially to perform in real time!
- Not to be confused with “hand shape estimation”
Why Does it Matter?

- More than just gestures
- Ideal for continuous input applications
- Links your hand dexterity into a computer model
- Will it redefine how we interact with computers??
Gaming
Design / Engineering
Robot Hand Control– Surgery? Industry?
Communication – Sign Language
How Does it Work?

- It's going to take some time to explain

- Starting from the ground up!
  - Decision trees
  - Ensemble techniques
  - Random forests
  - Body Pose estimation
  - Hand Pose Estimation

- Assumption is that everyone has a very basic idea of what machine learning is and does
Machine Learning

- **Goal:**
  - Given training data $T$ with entries $(x, y)$
  - Find a model that estimates $y$ for unseen $x$
  - This is called prediction

- **Quality Measurement:**
  - Minimize the probability of model prediction errors on future data

- **What are some models?**
  - Linear Regression
  - Support Vector Machines
  - Decision Trees!
Decision Trees

- Very intuitive
- Each node asks a question about a feature of the data
- Propagates through the tree depending on the answer to each question
- When algorithm gets to the end, the decision tree makes a classification
How to grow a tree from data?

- In what order do we ask the questions (test features)?
  - Each possible tree has an amount of entropy
  - Test out all possible questions for a node, and choose the one that reduces the entropy the most (largest information gain)

- How do nodes make decisions based on the features?
  - Same way!
  - Choose a decision boundary that gives the largest information gain
How to grow a tree from data?
Decision Trees: A Pretty Good Model!
Ensemble Learning

- Two competing methodologies:
  - Traditional: Build one really good model
  - Ensemble: Build many models and average the results

- Build a ton of “pretty good” models
- Combine them into one “pretty awesome” prediction!
- Important for individual models to not be correlated, otherwise there is a strong tendency to overfit
- So we add randomness!
Ensemble Techniques

- **Bootstrap Aggregation (Bagging)**
  - Take a random subsample from the training set $T$, with replacement
  - Train each model on a different subsample
  - Classification is the majority vote; Regression is the average

- **Random Forests: Multiple, randomized decision trees**
  1. Bagging
  2. Randomized Node Optimization: choose random set of questions
     - Number of questions affects the correlation of the trees
  3. Decision boundary of the decision trees: conic, linear, etc.
  4. Depth of the component decision trees
     - More depth means there will be more overfitting
Example: Different Trees
Example: Different Trees
Example: Different Trees

Training different trees in the forest

Testing different trees in the forest
Example: Random Decision Forest
Example: Multi-class Decision Trees
Example: Comparison to SVM Model
A quick look at body pose estimation

- Body Pose Estimation Pipeline
- Technology found in consumer devices, like the Kinect
- Very similar to hand pose estimation
Hand Pose Estimation Pipeline

1. Create a hierarchical skeleton model
2. Segment the hand into parts
3. Train RDFs and classify each pixel
4. Estimate the joints for each hand part
5. Connect the dots
What makes Hand Pose tough?

- Hand is much smaller than the body, but still has 22 DOF
- Self occlusion is very common and severe
- Can be rotated in any direction (body is always upright)
- Real depth data can be difficult to label
Some ideas..

- Restrict the viewing area of the hand
- One Advantage: Hands are fairly invariant among humans
- Train with synthetic data, rendered from 3D models
Train based on Synthetic Data

- Use 3D hand models to generate data
- Train the Random Decision Forests using this data
Hand Pose Estimation Pipeline

1. Create a hierarchical skeleton model
2. Segment the hand into parts
3. Train RDFs and classify each pixel
4. Estimate the joints for each hand part
5. Connect the dots
Pixel Classification

One Tree  Two Trees  Three Trees
Mean shift local mode finding

- Algorithm used to determine where the joints are
- Each pixel is given a weighted Gaussian kernel
- Weight is determined by class probability times depth
- Gradient ascent from many points finds the local maxima
- Highest local maxima determines the joint
- Threshold the scores to filter out non-visible joints
Joint Determination
Hand Pose Estimation Algorithm

Strengths
- Very fast
- Robust to fast movements and noise
- No initialization needed
- Can run on a GPU for interface applications or games

Issues
- Training must be done offline
- Number of images ~1-10M, takes 25-250 GB of data
- Number of operations is huge even with simple algorithm
Limitations of Single Layer RDF

- Difficult to generate every possible hand pose
- Dataset size is huge!
- Hard to capture the variation in the data set
- More variation $\rightarrow$ deeper trees $\rightarrow$ more RAM/memory

- Solution: Divide into sub problems and solve with separate RDFs
- Lower variation $\rightarrow$ lower complexity $\rightarrow$ less RAM/memory
Another Layer?

We must go deeper!
Multi-layered RDFs for Hand Pose

<table>
<thead>
<tr>
<th>Description</th>
<th>Layer Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Layer</td>
<td></td>
</tr>
<tr>
<td>Cluster Classification Layer</td>
<td></td>
</tr>
<tr>
<td>Cluster Layer</td>
<td></td>
</tr>
<tr>
<td>Pose Estimation Layer</td>
<td></td>
</tr>
<tr>
<td>Output Layer</td>
<td></td>
</tr>
</tbody>
</table>
Two Structures of Multi-layer RDFs

- Local Expert Network
  - Hand Shape Classification gives each pixel a label
  - Train local expert forests for each pixel label
  - Expert forest depends on pixel label; each pixel is classified

- Global Expert Network
  - Hand Shape Classification gives each pixel a label
  - The hand shape is determined by pixel voting
  - Train global expert forests for each pixel label
  - Expert forest depends on hand shape label; each pixel is classified
Local Expert Network
Global Expert Network

Diagram showing a network with nodes and edges connecting different levels.
Training a Multi-layer RDF

- Given the same data as before (hand shape not given)

1. Cluster the data
2. Train Hand Shape Classifier based on all clusters
3. Train each Pixel Classifier based on a specific cluster
Which is better? GEN or LEN

- Global Expert Networks average class distributions → More robust to noise

- Local Expert Networks use info from each pixel → Better at generalizing unseen data
Test: American Sign Language
Results

- Huge improvement over single-layer RDFs

<table>
<thead>
<tr>
<th>Method</th>
<th>Single-layered RDF</th>
<th>GEN</th>
<th>LEN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Per Pixel</td>
<td>68.0%</td>
<td>91.2%</td>
<td>90.9%</td>
</tr>
</tbody>
</table>
Results

- Remaining errors are concentrated on very similar poses
Summary

- What is Hand Pose Estimation?
  Determine the joint positions to fix all DOFs of the hand

- Why does it matter?
  Continuous Input Applications

- How does it work?
  Randomized Decision Forests

- What has been done?
  Add multiple layers for increased performance.
References

- [1] Keskin- Hand Pose Estimation and Hand Shape Classification Using Multi-layered Randomized Decision Forests
- [3] Qian- Realtime and Robust Hand Tracking from Depth
- [5] Oikonomidis - Evolutionary Quasi-random Search for Hand Articulations Tracking
- [7] Hilliges - Advanced topics in Gesture Recognition Part II
Questions?
Appendix: Getting Hand Shape from Hand Pose

- Hand shape is just shape information “fist”, “flat”, etc.
- Hand pose is specific joint angles for every DOF

- With hand pose, can use SVM to determine hand shape very robustly