Opening the Black Box:
Hierarchical Sampling Optimization for Estimating Human Hand Pose
Supplementary Material

Danhang Tang†‡
Jonathan Taylor†
Pushmeet Kohli†
Cem Keskin†
Tae-Kyun Kim‡
Jamie Shotton†

Microsoft Research†
Imperial College London‡

This document includes supplementary material to provide more implementation details for the ICCV 2015 Paper ‘Opening the Black Box: Hierarchical Sampling Optimization for Estimating Human Hand Pose’.

The kinematic Model

Fig. 1 illustrates the kinematic model as well as codenames for each joint. On top of that, Algorithm 1 defines the kinematic model in the form of pseudocode. This model is in fact a configuration for training the Hierarchical Sampling Forests.

Algorithm 1 Kinematic tree model (pseudocode in matlab style).

%Fields of a kinematic tree node:
%  Id
%  Input: input joint location (with codenames as in Fig. 1).
%  Output: predicted joints (with codenames as in Fig. 1).
%  Children: pointing to a set of kinematic tree node Ids.
%  Forest: a set of regression trees corresponding to this node.
%  →: the operator for accessing a field, e.g., k→Input accesses the ‘input’ field of k.

%The kinematic tree
KinematicTree = [
    struct('Id': 0, 'Input': [CoM], 'Output': [WR], 'Children': [1]),
    struct('Id': 1, 'Input': [WR], 'Output': [TR, IR, MR, RR, PR], 'Children': [2, 3, 4, 5, 6]),
    struct('Id': 2, 'Input': [TR], 'Output': [TM], 'Children': [7]),
    struct('Id': 3, 'Input': [IR], 'Output': [IM], 'Children': [8]),
    struct('Id': 4, 'Input': [MR], 'Output': [MM], 'Children': [9]),
    struct('Id': 5, 'Input': [RR], 'Output': [RM], 'Children': [10]),
    struct('Id': 6, 'Input': [PR], 'Output': [PM], 'Children': [11]),
    struct('Id': 7, 'Input': [TM], 'Output': [TT, TP], 'Children': []),
    struct('Id': 8, 'Input': [IM], 'Output': [IT, IP], 'Children': []),
    struct('Id': 9, 'Input': [MM], 'Output': [MT, MP], 'Children': []),
    struct('Id': 10, 'Input': [RM], 'Output': [RT, RP], 'Children': []),
    struct('Id': 11, 'Input': [PM], 'Output': [PT, PP], 'Children': [])
];

Training

Since there are 12 nodes in the kinematic model (see Algorithm 1), and each has a corresponding regression forest with 3 trees, in total we need to train 36 trees for an HSF. Algorithm 2 gives the training pseudocode for one tree w.r.t. one kinematic node $k$. When generating the training samples, input $p_k$ is the position of the joint indicated by $k \rightarrow \text{input}$. The

\footnotesize

1In the case of MSHD dataset, we need to train an HSF for each quaterion cluster.
output tuple \((\zeta_k, \theta_k)\) is calculated given \(k \rightarrow \text{output}\), as described in Section 3.3 of the paper. All notations are consistent with the paper if not specified.

**Algorithm 2 Training**

**Require:** A set of training samples \(S = \{(Z, \theta)\}\), where \(Z\) is a depth image and \(\theta\) is its parameter; A node \(k\) in the kinematic model; Maximum tree depth \(D\).

**Ensure:** A Hierarchical Sampling Tree \(t_k\).

1: **procedure** \(\text{Train}(S, k)\)
2: Construct a training set \(S_k = \{(Z, p_k, \zeta_k, \theta_k)\mid k, \theta\}\) \quad \triangleright \text{Generate the training set given } k \text{ and } \theta \text{ as in Sec. 3.3.}
3: Train a standard regression tree \(t_k\) with \(S_k\).
4: for all leaf node \(n \in t_k\) do
5: \(G = \text{Fit}(S_{kn})\). \quad \triangleright S_{kn} \text{ is the subset of } S_k \text{ that arrive to } n.
6: Store \(G\) with \(n\).
7: end for
8: end procedure

9: **function** \(\text{Fit}(S)\)
10: Fit a GMM \(G_{\zeta}\) to \(\{\zeta\mid (\zeta, \theta) \in S\}\) \quad \triangleright \text{Use the 3D offset } \zeta \text{ as a proxy to cluster sample tuples.}
11: for all component \(c_{\zeta} \in G_{\zeta}\) do
12: Fit a Gaussian \(c_\theta\) to \(\{\theta\mid (\zeta, \theta) \in c_{\zeta}\}\). \quad \triangleright \text{Generate the actual GMM with samples that are close enough.}
13: end for
14: Return \(G_\theta = \{c_\theta\}\).
15: end function

**Testing**

Algorithm 3 describes the process of testing, which is visualized in Fig. 3 of the paper.

**Algorithm 3 Testing**

**Require:** A segmented image \(Z\); the kinematic model \(K\).

**Ensure:** A full pose result \(\theta\).

1: **procedure** \(\text{Test}(Z)\)
2: Calculate \(p = \text{CoM}(Z)\). \quad \triangleright \text{Use center of mass as the starting point.}
3: Let \(k \leftarrow K[0]\)
4: for all \(i \leftarrow 1\) to \(N\) do
5: \(\theta = \text{DESCENDKINEMATICTREE}(Z, p, k)\). \quad \triangleright \text{Generate a full pose hypothesis } \theta.
6: Calculate the golden energy \(E_{Au}\) of \(\theta\).
7: end for
8: Return the best \(\theta\) with lowest energy.
9: end procedure

**function** \(\text{DESCENDKINEMATICTREE}(Z, p, k)\)
10: Randomly select a regression tree \(t\) from \(k \rightarrow \text{Forest}\). \quad \triangleright \text{‘\rightarrow’ retrieves a field.}
11: Descend \(t\) with \(p\), which returns a GMM \(G\).
12: Generate \(M\) samples from \(G\).
13: Choose the best sample as \((\zeta_k, \theta_k)\) with the silver energy \(E_{Ak}\). \quad \triangleright \text{Note that we need to calculate } \zeta_k \text{ from } \theta_k \text{ using the hand skeleton model.}
14: for all \(c \in k \rightarrow \text{Children}\) do
15: \(\theta \leftarrow \theta \cup \theta_{kc}, p_{kc} = p + \zeta_{kc}\).
16: \(\text{DESCENDKINEMATICTREE}(Z, p_{kc}, K[c])\).
17: end for
18: Return \(\theta\).
19: end function