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.

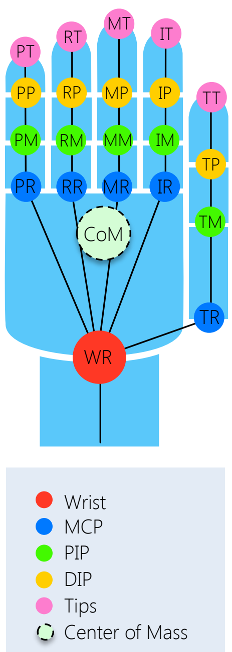

Figure 1: Kinematic model

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[1]. 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 $k \rightarrow \mathrm{input}$. The

[1] In the case of MSHD dataset, we need to train an HSF for each quaterion cluster.

output tuple (ζ_k, θ_k) is calculated given $k \rightarrow \text{ouput}$, 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 θ is its parameter; A node k in the kinematic model; Maximum tree depth D.
Ensure: A Hierarchical Sampling Tree t_k.

procedure TRAIN(S, k)
  Construct a training set $S_k = \{(Z, p_k, \zeta_k, \theta_k)|k, \theta\}$ ▷ Generate the training set given k and θ as in Sec. 3.3.
  Train a standard regression tree t_k with S_k.
  for all leaf node $n \in t_k$ **do**
    $G = \text{FIT}(S_{kn})$. ▷ S_{kn} is the subset of S_k that arrive to n.
    Store G with n.
  end for
end procedure

function FIT(S)
  Fit a GMM G_ζ to $\{\zeta|(\zeta, \theta) \in S\}$. ▷ Use the 3D offset ζ as a proxy to cluster sample tuples.
  for all component $c_\zeta \in G_\zeta$ **do**
    Fit a Gaussian c_θ to $\{\theta|(\zeta, \theta) \in c_\zeta\}$. ▷ Generate the actual GMM with samples that are close enough.
  end for
  Return $G_\theta = \{c_\theta\}$.
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 θ.

procedure TEST(Z)
  Caculate $p = \text{CoM}(Z)$. ▷ Use center of mass as the starting point.
  Let $k \leftarrow K[0]$
  for all $i \leftarrow 1$ to N **do**
    $\theta = \text{DESCENDKINEMATICTREE}(Z, p, k)$. ▷ Generate a full pose hypothesis θ.
    Calculate the golden energy E_{Au} of θ.
  end for
  Return the best θ with lowest energy.
end procedure

function DESCENDKINEMATICTREE(Z, p, k)
  Randomly select a regression tree t from $k \rightarrow$ Forest. ▷ '$\rightarrow$' retrieves a field.
  Descend t with p, which returns a GMM G.
  Generate M samples from G.
  Choose the best sample as (ζ_k, θ_k) with the silver energy E_{Ag}. ▷ Note that we need to calculate ζ_k from θ_k ▷ using the hand skeleton model.

  for all $c \in k \rightarrow$ Children **do**
    $\theta \leftarrow \theta \bigcup \theta_{kc}$, $p_{kc} = p + \zeta_{kc}$.
    $\text{DESCENDKINEMATICTREE}(Z, p_{kc}, K[c])$.
  end for
  Return θ.
end function