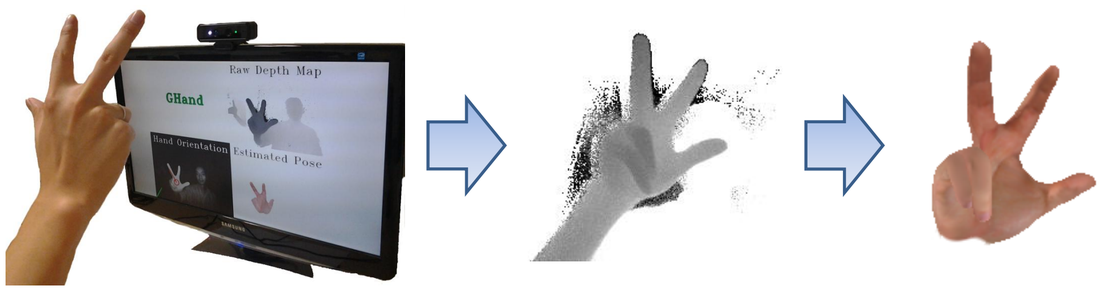
Real-time hand pose estimation from depth camera using GPU

Ashwin Nanjappa, Chi Xu, Li Cheng
Bioinformatics Institute, A*STAR, Singapore

Overview

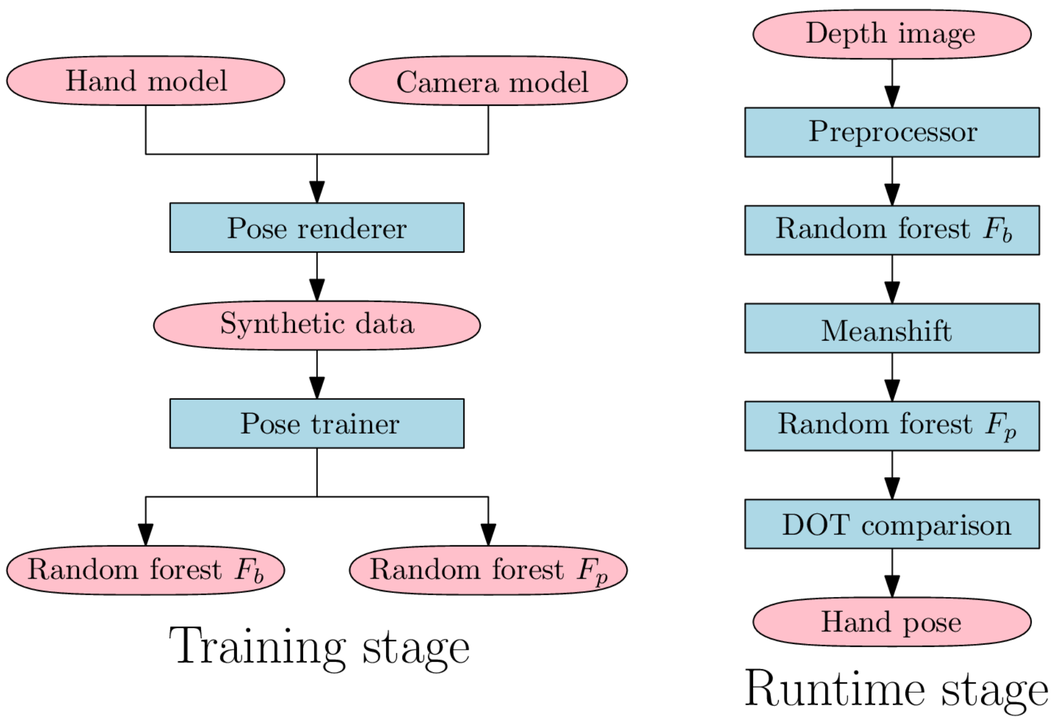


Depth camera, like in Xbox Kinect, is ubiquitous today.

Our **GHand** machine learning algorithm estimates 3D hand pose from a depth camera image using an unique two random forest approach:

- Forest F_b: Estimates 3D position and orientation of hand.
- Forest F_p : Estimates joint angles of our kinematic chain hand model.

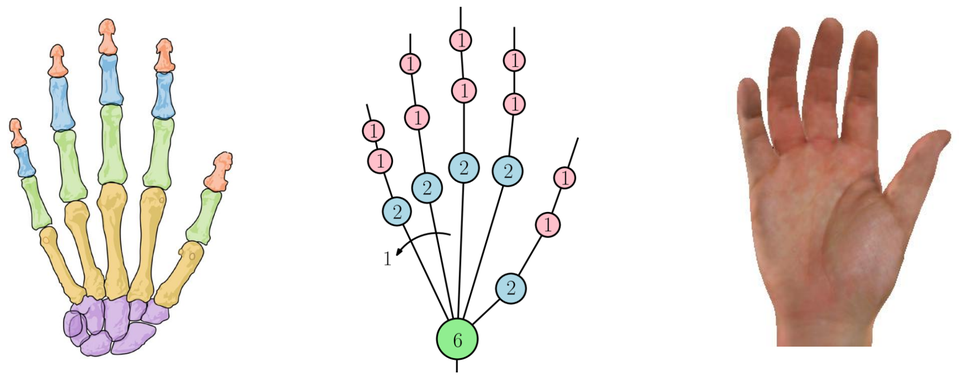
Our algorithm



All steps on GPU: Traversal of random forests, meanshift (to find mode of forest results) and DOT (to pick best matching pose)

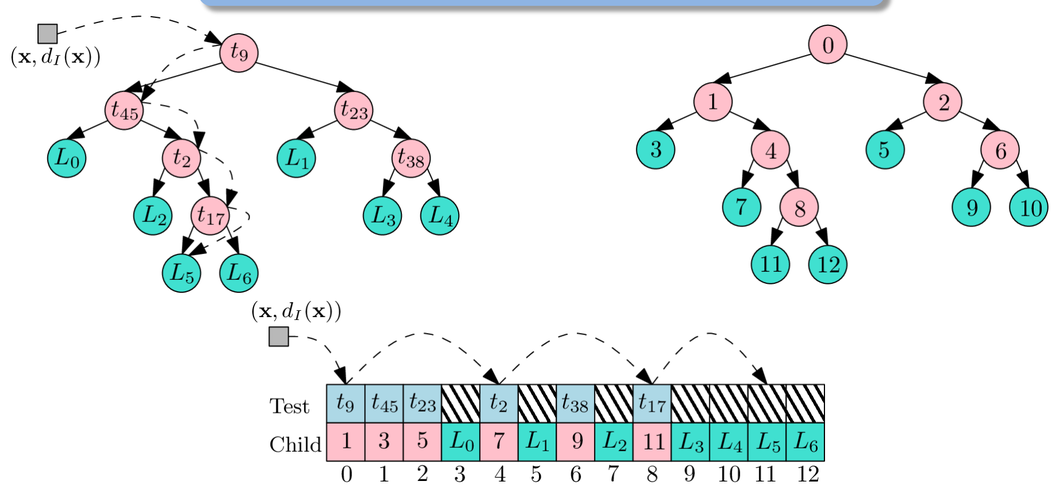
Performance: Our CUDA implementation runs real-time at 64FPS or more, a 4x speedup over CPU.

Kinematic chain hand model



- **Skeleton**: 15 joints and 20 bones
- **Degrees of freedom**: 6 DoF for hand base, 27 DoF in total
- **Mesh**: 70K triangles, for realistic synthetic data

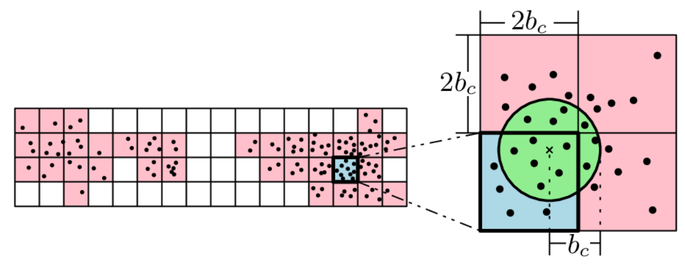
Random forests on GPU



Problem: Each pixel traverses trees and tests at each node. Gather results from leaves.

Solution:
- Efficient access: Trees of forest stored as contiguous arrays.
- Storage: Right and left child nodes stored together.
- Depth image stored as 2D texture for efficient access.
- Valid pixel locations are found and compacted. Explicitly cached for performance.

Mean shift on GPU



Mean shift to find mode of 6-dimensional points gathered from F_b.

Problem: Meanshift is iterative, hard to parallelize and finding points inside bandwidth is expensive.

Solution:
- Enclose points in uniform grid of cell size $2b_c$x$2b_c$x$2b_c$ (b_c is bandwidth).
- Make cell-to-point and point-to-cell map using cell index and GPU sort.
- Compute per-cell centroid and use as seeds for meanshift.
- For all seeds in parallel, compute next centroid by only accessing points in necessary neighbor cells.

DOT comparison on GPU

Aim: Compute similarity score between input and results from F_p.
DOT (Dominant Orientation Template) is similar to HOG, but has per-pixel parallelization.
- Transformation maps are first generated for image.
- Image mapped to 2D texture for efficient spatially coherent access.
- Bi-linear interpolation of texture units used to compute depth value of transformed pixel.
- Gradient computed efficiently using fast *intrinsic* functions.
- Atomic addition used for accumulation of results.

Results

Joint error: 30mm lesser than ICCV13[1]

Speedup: 4-5x faster than ICCV13[1]

[1] Xu, C., and Cheng, L. 2013. Efficient hand pose estimation from a single depth image. In ICCV.