Real Time Human Pose Estimation: Parallel Performance

Kenneth Marino, Advisor: Professor Yaser Sheikh

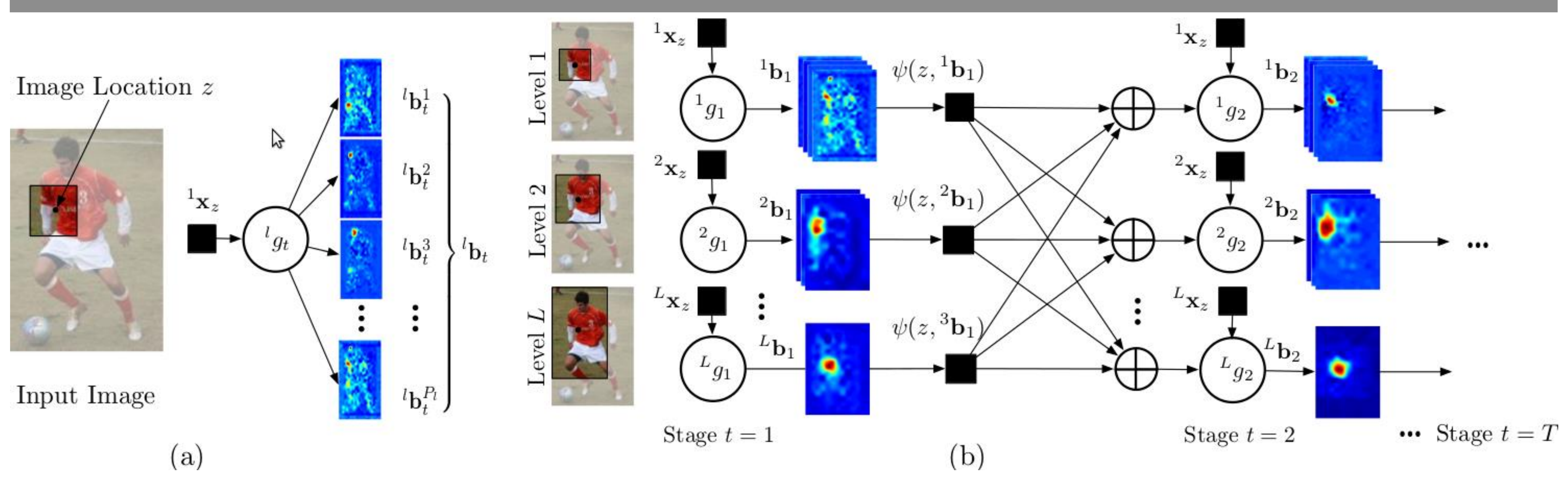
Problem

Pose Estimation
- Determining position of limbs and joints of people from still images
- Useful for human robot interaction, gesture recognition, and prediction

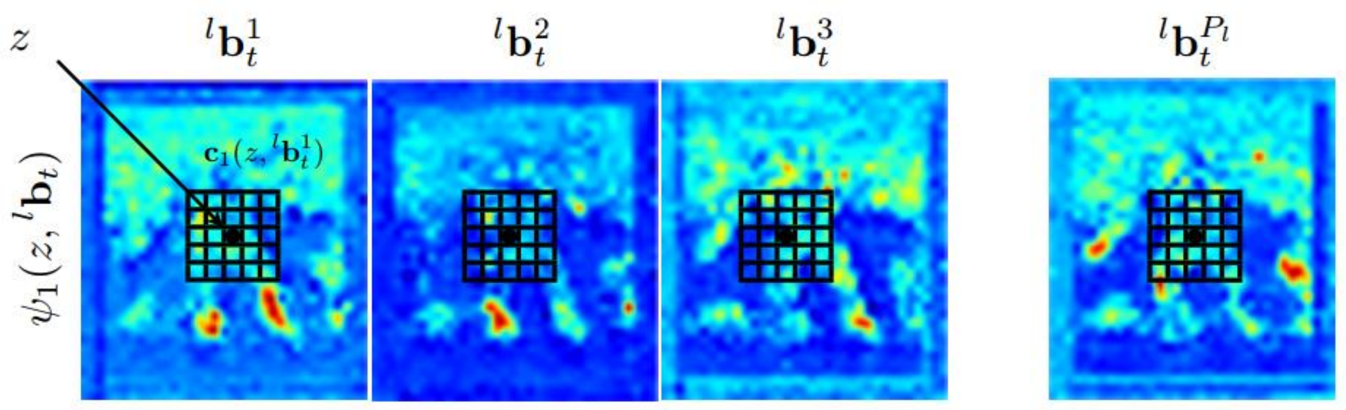
Challenges
- Kinect provides real-time pose estimation but requires IR sensor
- Current image-only pose estimators too slow for real-time application

Approach

Pose Machines



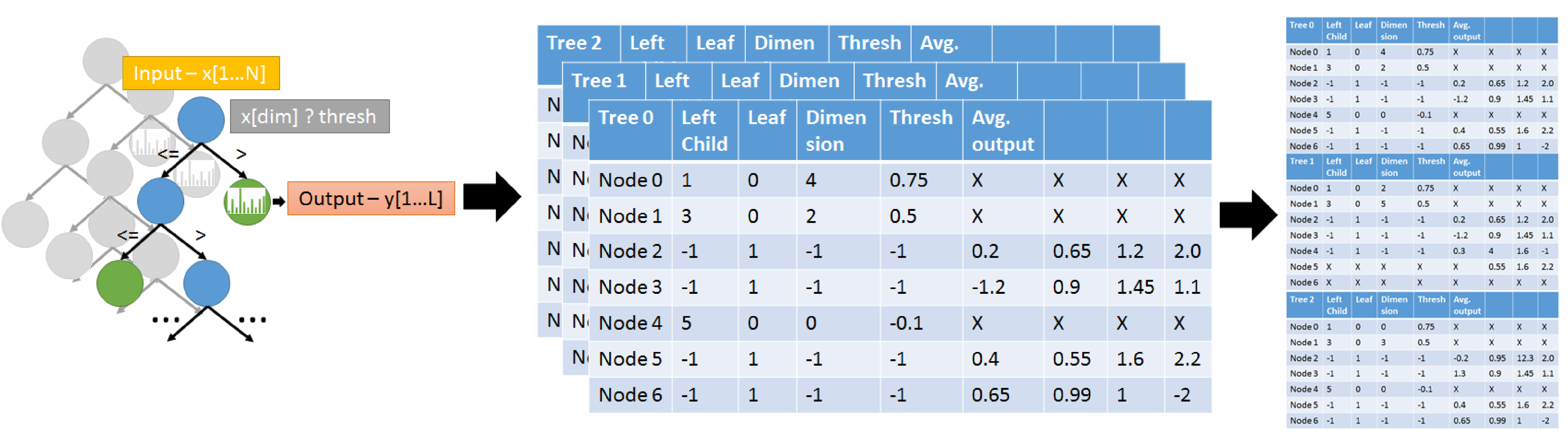
- Use multiple stages of part prediction with boosted random forests [1]
- Predict on multiple hierarchies of parts
- Stages and hierarchies pass information using context features



- Context Features extracted from patch around location *z*

GPU Parallelization - Testing

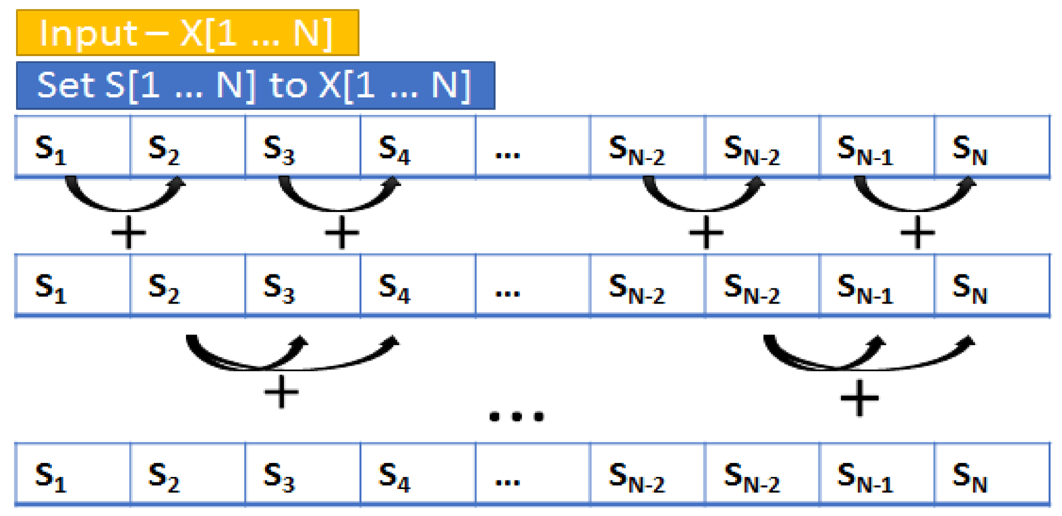
Random Forests
- Collection of trees trained on random subset of samples & features [2]
- Branching algorithm – each input navigates to leaf node of each tree to determine output distribution
- Used modification of method by Sharp (2008)
- Added parallelization over trees



Context Features
- Embarrassingly Parallel
- Minimized copies between GPU, CPU

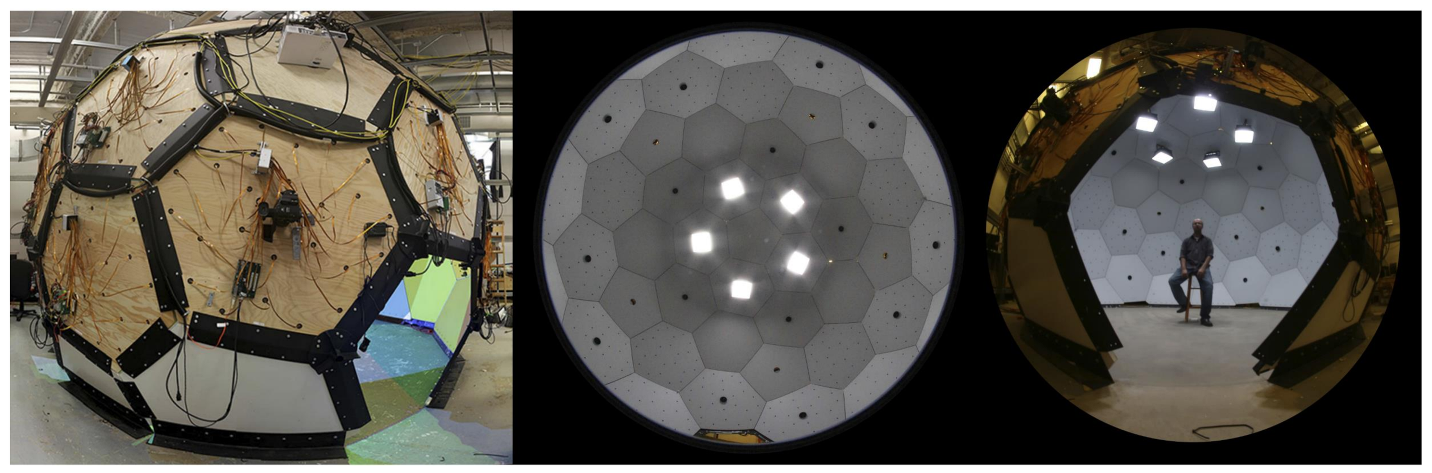
GPU Parallelization - Training

- Use GPU to accelerate training, train on more images
  - Current speed: projected 10 days for 100,000 images
- Most time used in sort and deciding splits
  - Main operation in deciding splits is calculating running sums to calculate gain
  - Sort is $\theta(NlogN)$
  - Gain calculation is $\theta(N)$
- Developed new algorithm for faster running sums



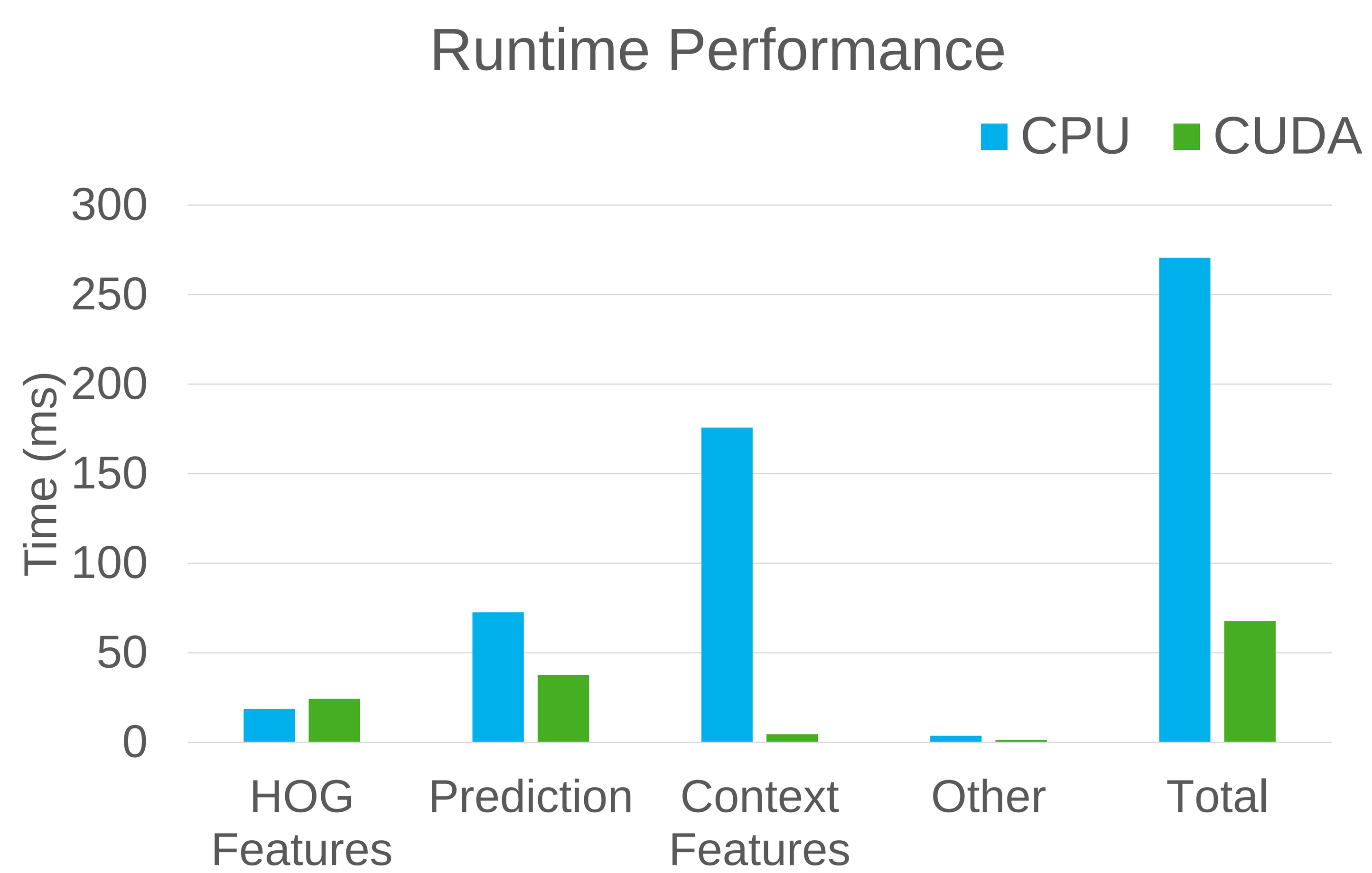
- Runtime $\theta(\frac{N}{k}logN)$ – k is number of parallel processes
- In practice k << N, so approximately $\theta(NlogN)$ runtime
- More precise than naïve sum

Massive Data
- Multiple viewpoints with Kinect annotation from Panoptic studio
- Large image dataset introduces time and space problems for training

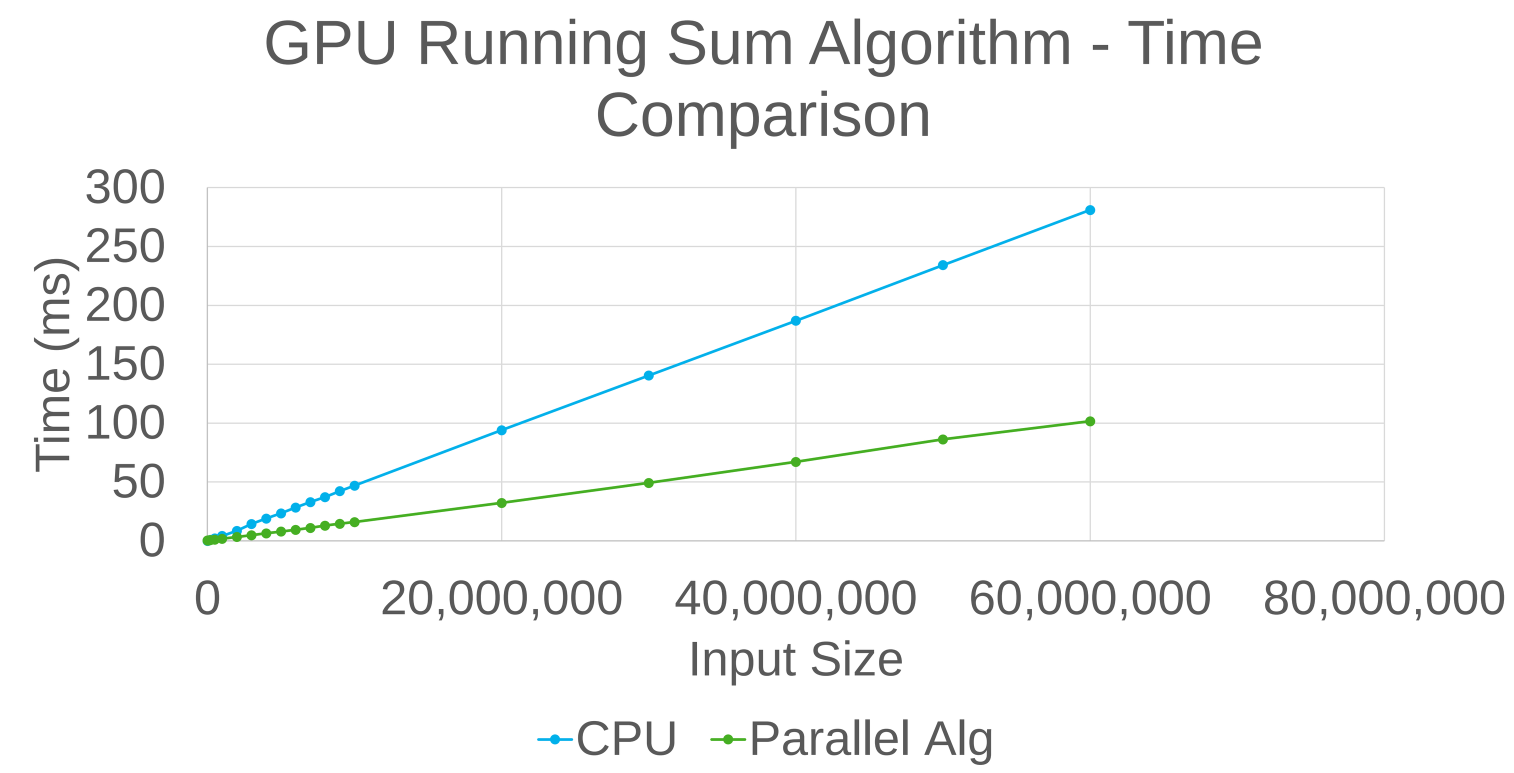


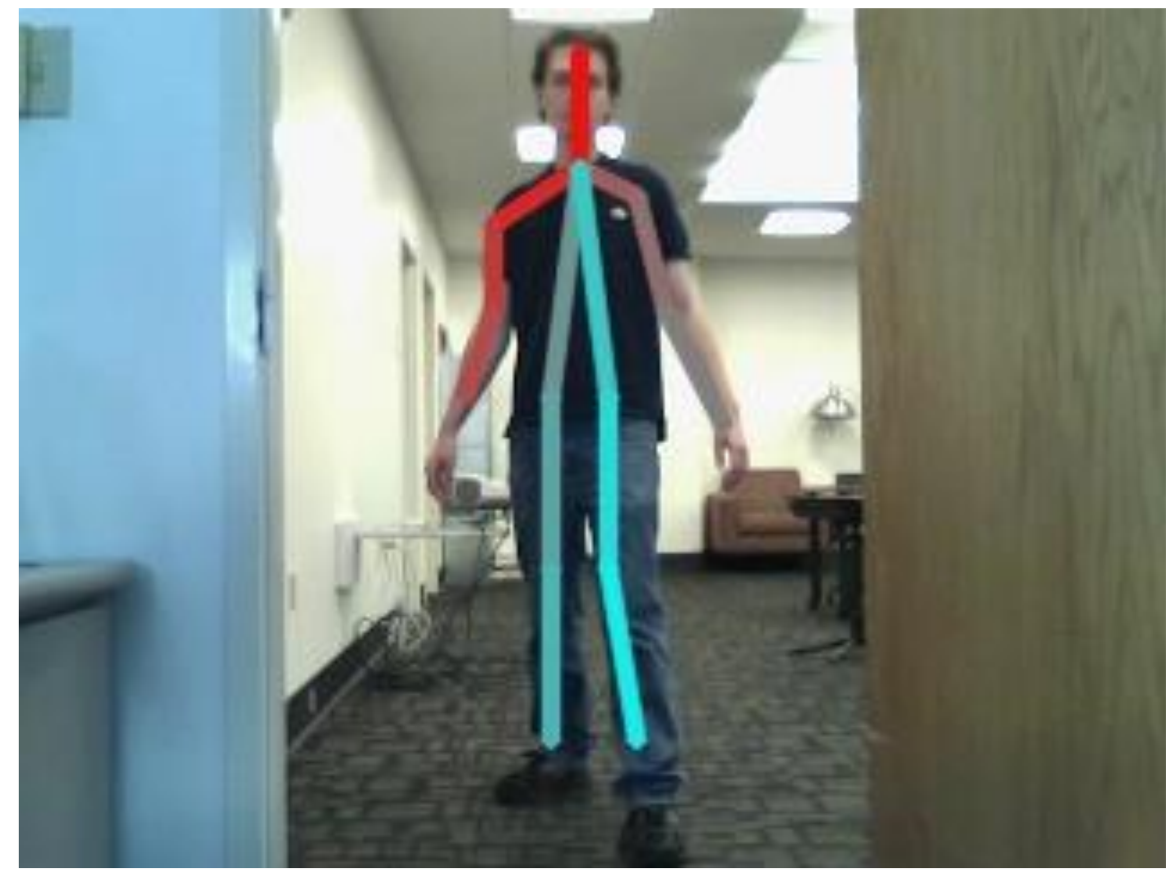
Results

4x Speed Improvement from CPU testing



3x Speed Improvement in Running Sum Algorithm





Example output of prediction

Conclusion
- Achieved real-time speeds
- Accuracy could still be improved
  - Larger training sets
  - More stages/hierarchies
  - Multiple scales
- Accuracy vs. time performance tradeoff
  - Most non-data accuracy improvements are linear time additions

Works Cited

[1] V. Ramakrishna, D. Munoz, M. Hebert, J. A. Bagnell, and Y. Sheikh, "Pose Machines: Articulated Pose Estimation via Inference Machines," *ECCV*, 2014.

[2] L. Breiman, "Random Forests," Machine Learning, 2001.

[3] T. Sharp, "Implementing Decision Forests on a GPU," *ECCV* 2008

Acknowledgements

- Thanks to Professor Yaser Sheikh and Varun Ramikirkrisna for their help and guidance this summer.