

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



- 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

	Tree	2 Left	Leaf	Dim	en Thre	sh Avg.						Tree 0 L	eft Le hild	af Dimer sion	n Threst	Avg. outp
Input – x[1N] x[dim] ? thresh Output – y[1L]												Node 0 1	0	4	0.75	X
	Т	ree 1 le	ft le	af Di	imen Tl	hresh 🗛	νσ					Node 1 3	0	2	0.5	X
	NI						Ъ .					Node 2 -	1 1	-1	-1	-1.2
	IN	Tree	left	Leaf	Dimen	Thresh	Δυσ					Node 4 5	0	0	-0.1	х
				Lear	·	mean	~¥6'					Node 5 -	1 1	-1	-1	0.4
	IN N	le la	Child		sion		output					Node 6	1 1	-1	-1	0.65
				•								Tree 1 L	eft Le bild	af Dimer	n Threst	Avg.
		Node 0	1	0	4	0.75	Х	Х	X	Х		Node 0 1	. 0	2	0.75	Х
			~	0	2	0.5	х	Х	Х	Х	•	Node1 3	0	5	0.5	х
		n Node T	3									Node 2 -	1 1	-1	-1	0.2
		Nede 2	1	1	-1	-1	0.2	0.65	1.2	2.0		Node 3 -	1 1	-1	-1	-1.2
		Node 2	-1							2.0		Node 4 -		-1	-1	0.3
		Nede 2	-1	1	-1	-1	-1.2	0.9	1.45	1.1		Node 5 X	x	x	x	x
		Node 3										Tree 2 L	eft Le	af Dime	n Thres	h Avg.
		L Nada 4	5	0	0	-0.1	Х	Х	Х	Х		0	hild	sion		outp
		Node 4										Node 0 1	. 0	0	0.75	x
	N	No.	1	4	-1	-1	0.4	0.55	1.6	2.2		Node1 3	0	3	0.5	X
		v Node 5	-1	T						2.2		Node 2 -	1 1	-1	-1	-0.2
		Nede C	1	4	1	1	0.05	0.00	1	2		Node 4 5	5 0	0	-0.1	X
		Node 6	-1	1	-1	-1	0.65	0.99	1	-2						

Context Features

- Embarrassingly Parallel
- Minimized copies between GPU, CPU



- 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 $\theta(\frac{N}{\nu} \log N) k$ 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



x x x x x x

 0.65
 1.2
 2.6

 0.9
 1.45
 1.1

 X
 X
 X

 0.55
 1.6
 2.7

 0.99
 1
 -2

 X
 X
 X

 X
 X
 X

 0.65
 1.2
 2.0

 0.9
 1.45
 1.1

 4
 1.6
 -1

 0.55
 1.6
 2.2

 X
 X
 X

x x x x x x

0.95 12.3 2.0 0.9 1.45 1

х х х

Node 5 -1 1 -1 -1 0.4 0.55 1.6 2 Node 6 -1 1 -1 -1 0.65 0.99 1 -2

Results

4x Speed Improvement from CPU testing **Runtime Performance**





3x Speed Improvement in Running Sum Algorithm

GPU Running Sum Algorithm - Time Comparison



40,000,000 Input Size



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.



80,000,000

