3D Skeletal Tracking on Azure Kinect

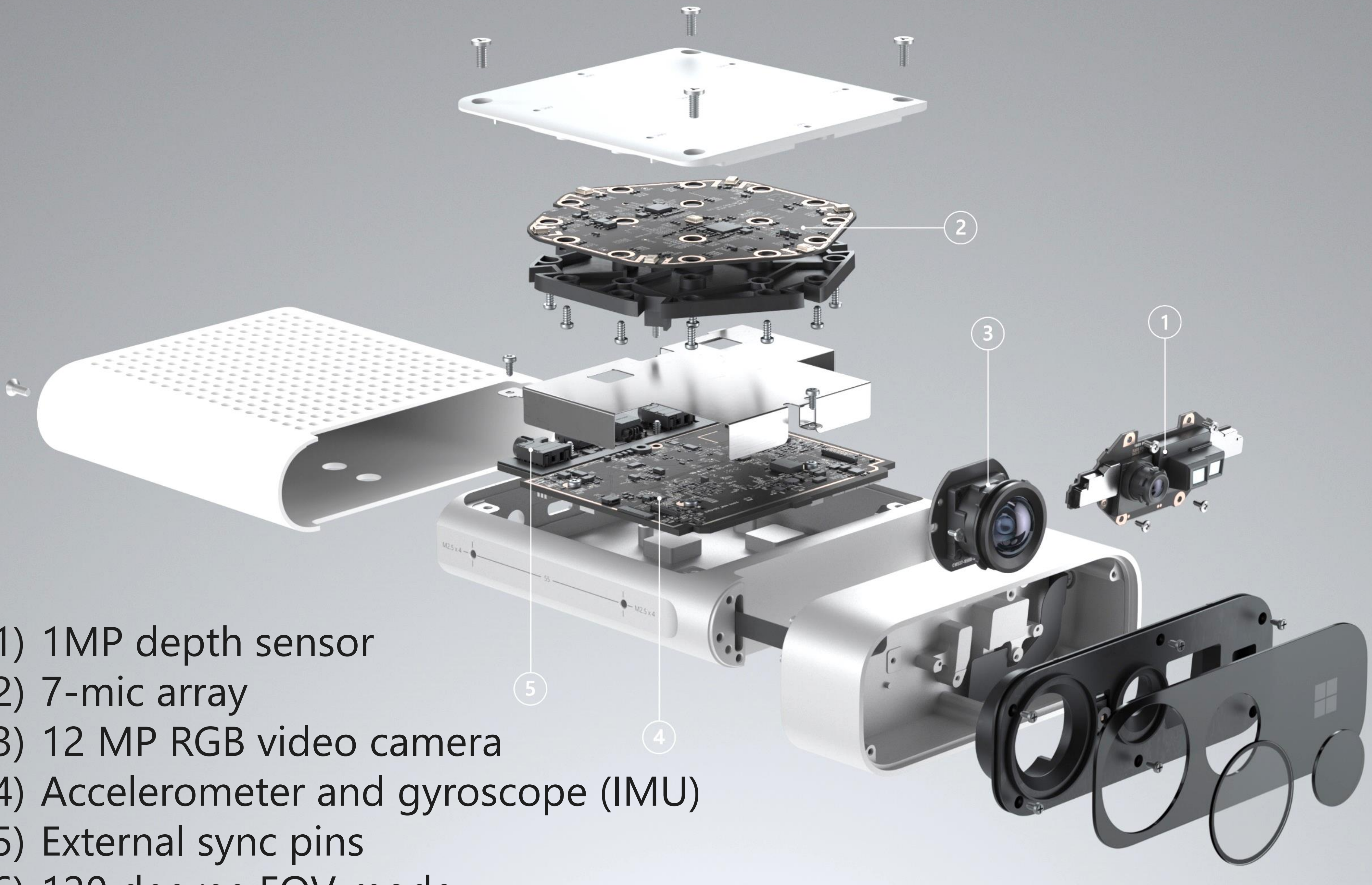
--Azure Kinect Body Tracking SDK

Zicheng Liu, Principal Research Manager
Microsoft

Azure Kinect DK

Build computer vision and speech models using a developer kit with advanced AI sensors

- Get started with a range of SDKs, including an open-source Sensor SDK.
- Experiment with multiple modes and mounting options.
- Add cognitive services and manage connected PCs with easy Azure integration.



(1) 1MP depth sensor
(2) 7-mic array
(3) 12 MP RGB video camera
(4) Accelerometer and gyroscope (IMU)
(5) External sync pins
(6) 120 degree FOV mode

Use Cases

Analyzed over 900 IWANTKINECT survey responses for body tracking applications. Thank you!

Three clear winners

- Kinematic analysis
- Human understanding
- Human interaction

Large focus on these use cases in training and validating model

Kinematic Analysis

Posture analysis

Rehabilitation

Fitness

Patient monitoring

Fall detection

Sports instruction

Hack for Good

Gigi’s Playhouse

AI-Based Physical Therapy for Down Syndrome

Microsoft

Human Understanding

Shopper behavior understanding

Person detection and counting

Person tracking

Smart spaces interaction

Human Interaction

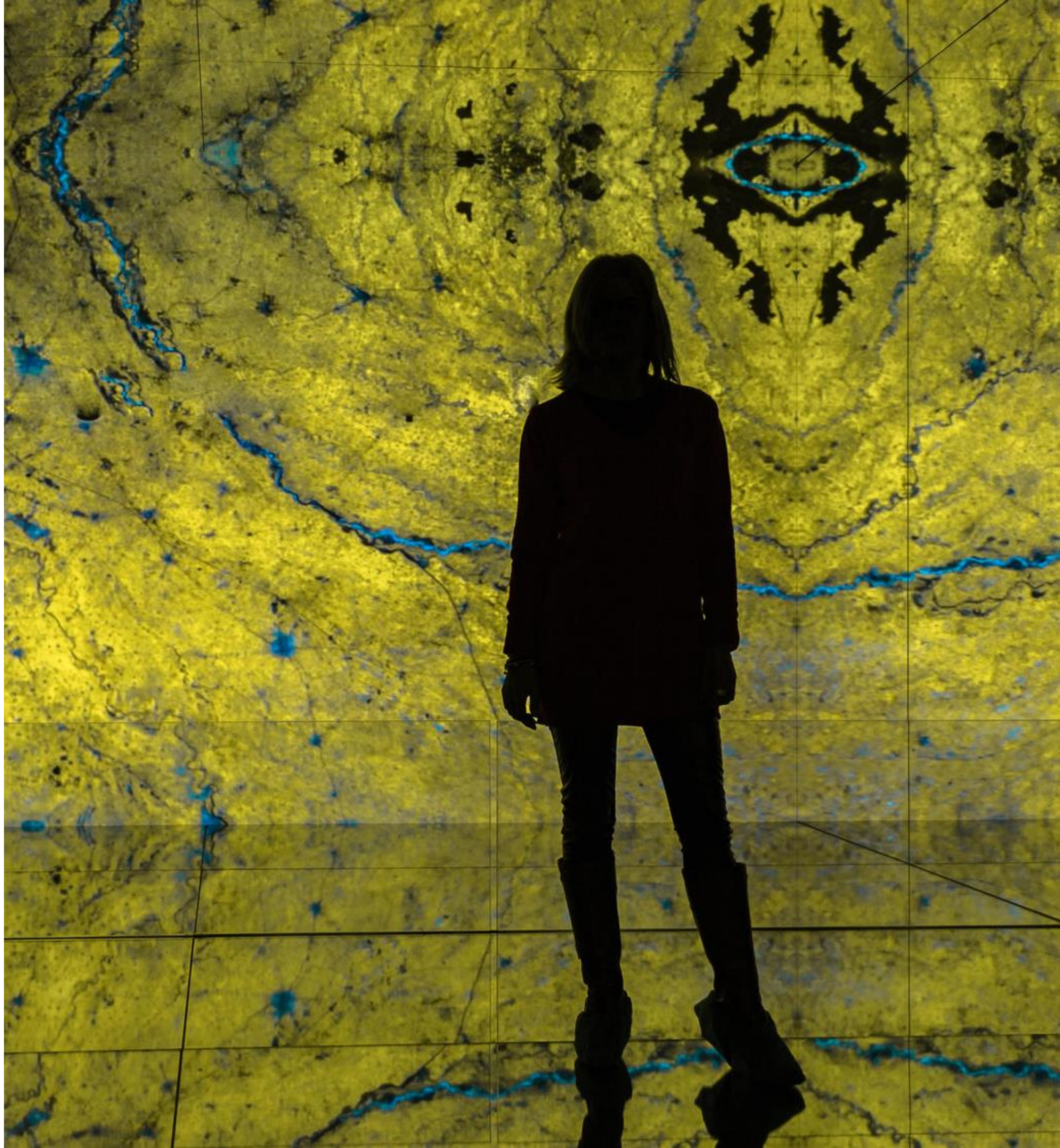
Information signs and video walls

Interactive art and performance

Interactive (museum) exhibits

Customer sizing and fitting

Machine safety



Overview of Body Tracking SDK

Designed from the ground up for Azure Kinect DK

- Instance segmentation map
- 3D joint positions per person
- Unique IDs to track temporally

Improved performance over Kinect for Windows v2

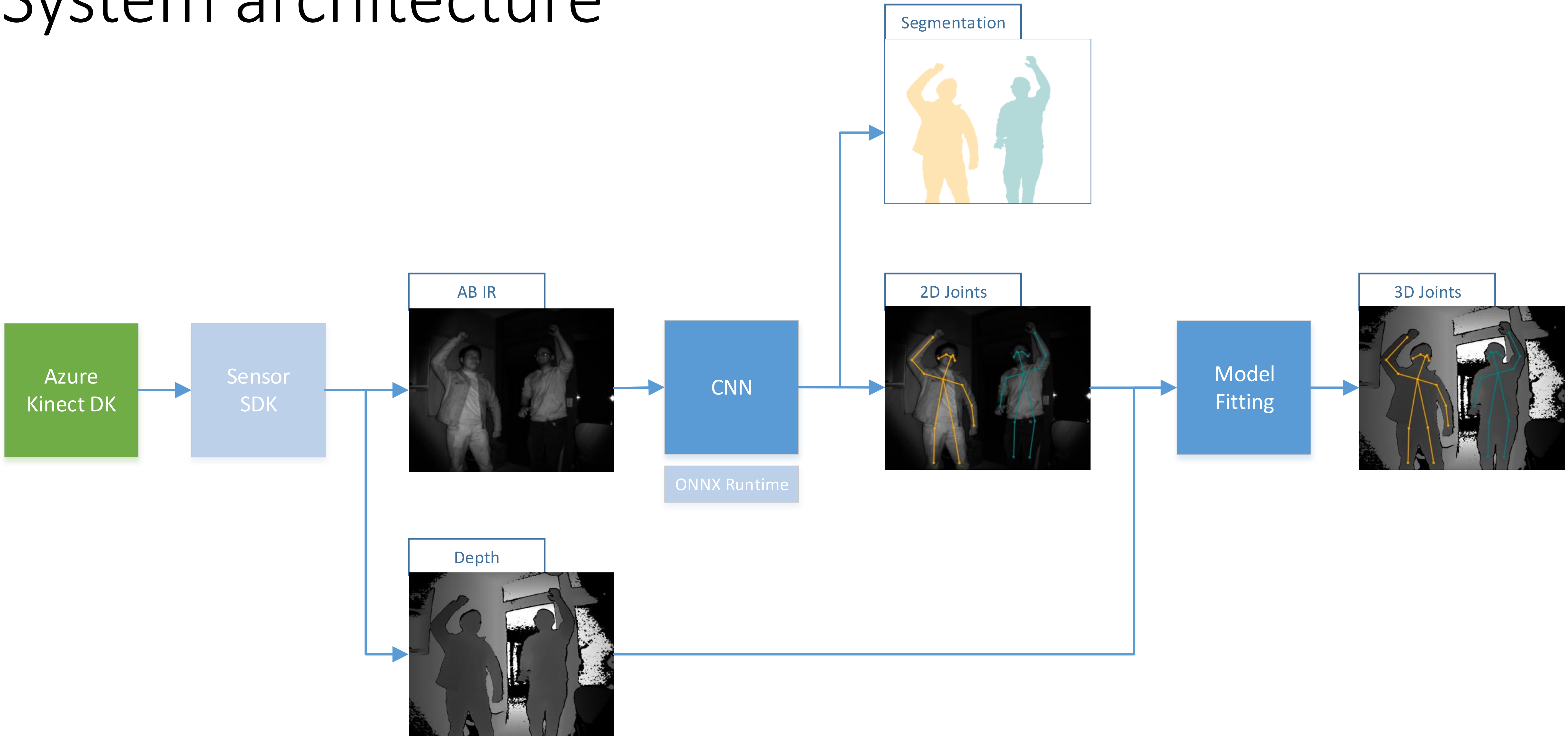
- Anatomically (28+ land marks / joints) more accurate skeleton
- Higher joint accuracy and precision
- Improved robustness e.g. side view, bending, lying

Cross platform development

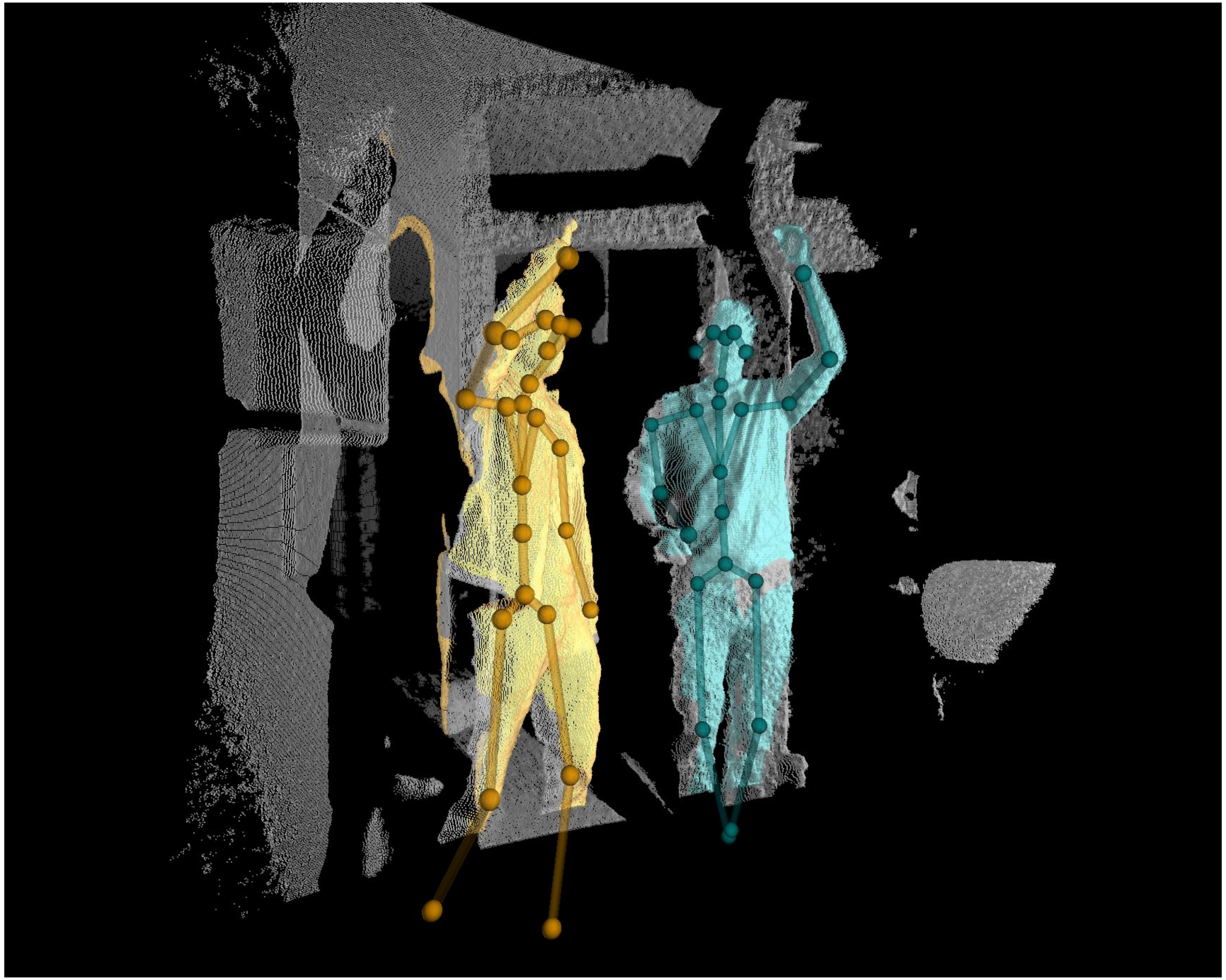
- Windows with Linux in preview
- C/C++ and C# (coming later)

ONNX runtime with support for NVIDIA 1070 (or better) hardware acceleration

System architecture



3D Skeletons



Why 3D

- Calculating joint angles is not possible to do correctly in 2D
- Understanding of whether a joint is coincident with another 3D object
- Accurate scale estimation for user size/height

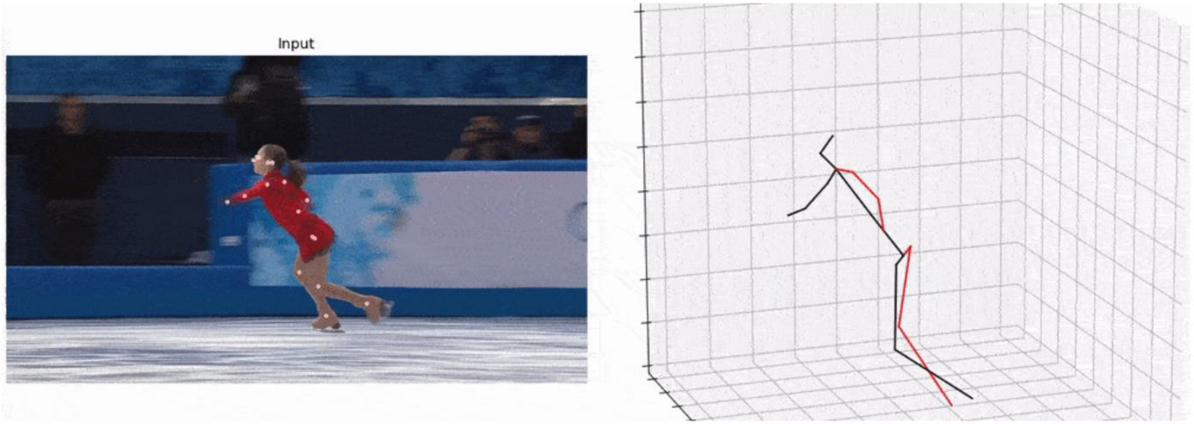
CNN: 2D pose estimation from IR

Human Pose Estimation

- Top-Down
  - ✓Person detector + Single-person pose estimation
  - ✓Person detection errors

Top-Down vs. Bottom-Up

- Bottom-Up
  - ✓Directly inferring the poses of multiple people in an image
  - ✓Unknown number of people that can occur at any position or scale
- 2D => 3D
  - ✓Ongoing research
  - ✓Single-person based 2D-to-3D conversion
  - ✓Depth/scale is not deterministic

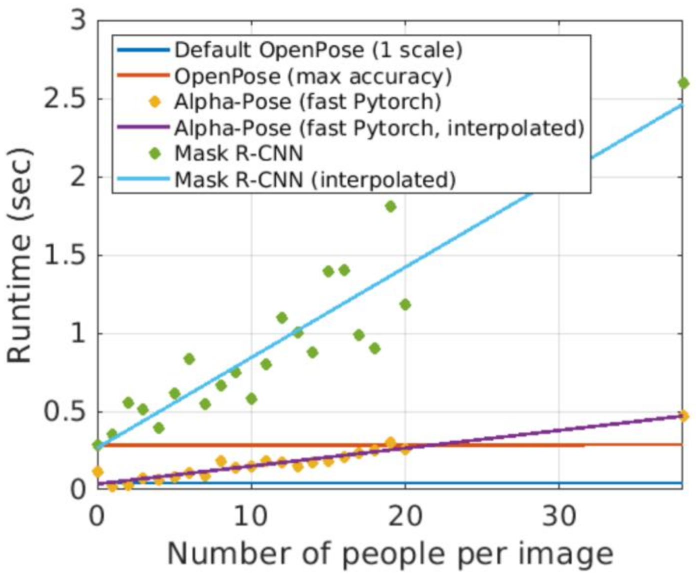


❖ https://github.com/facebookresearch/VideoPose3D

Challenge

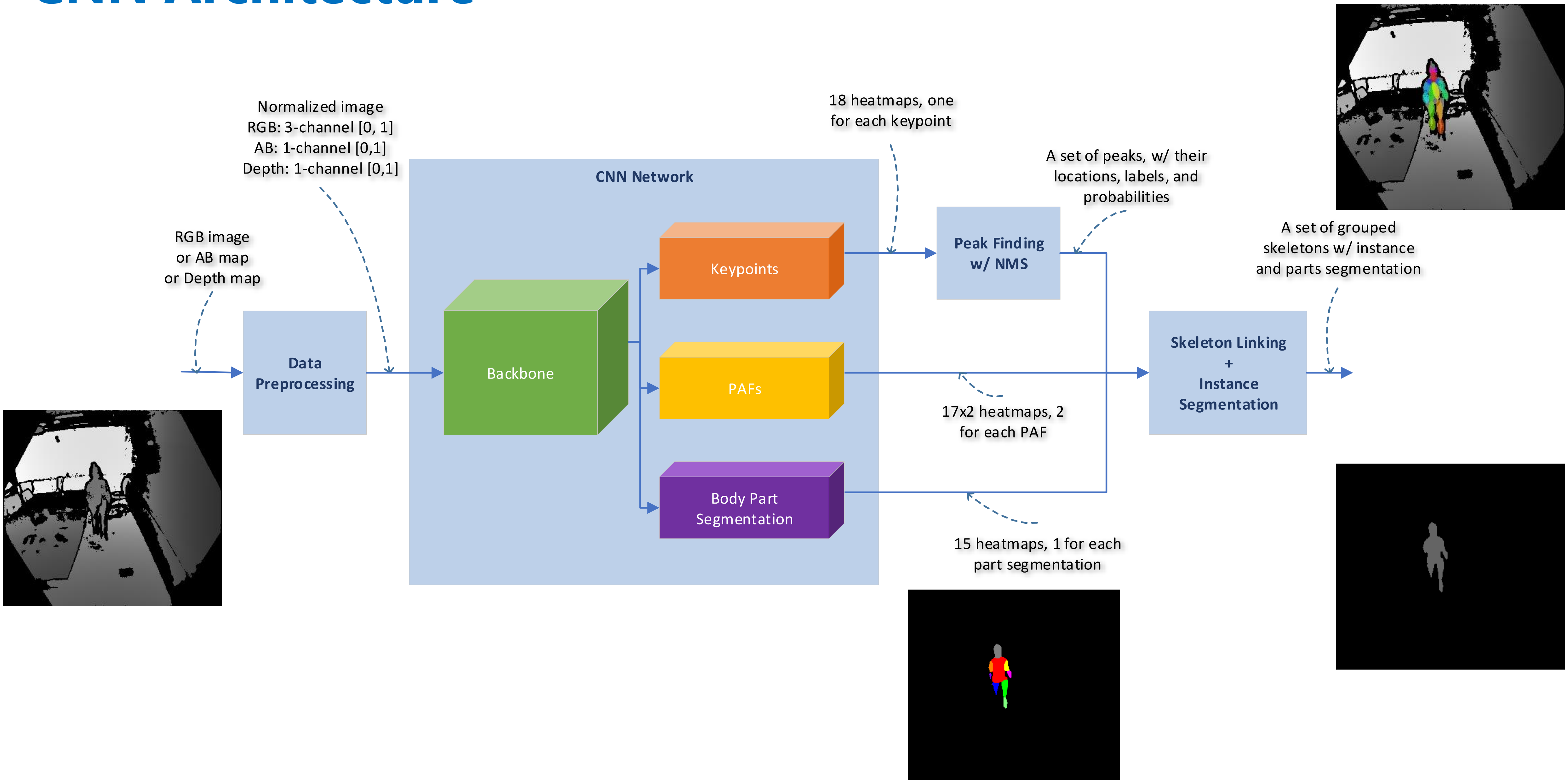
- Accuracy vs. Speed
  - ✓Trade-off for low-end GPUs
- RGB vs. AB/Depth
  - ✓No available dataset like MSCOCO for AB/Depth
- Real vs. Synthetic
  - ✓The reality gap
- Additional output
  - ✓Instance segmentation
  - ✓Pose estimation for hands and feet

❖ https://github.com/CMU-Perceptual-Computing-Lab/openpose
❖ http://cocodataset.org/#keypoints-leaderboard

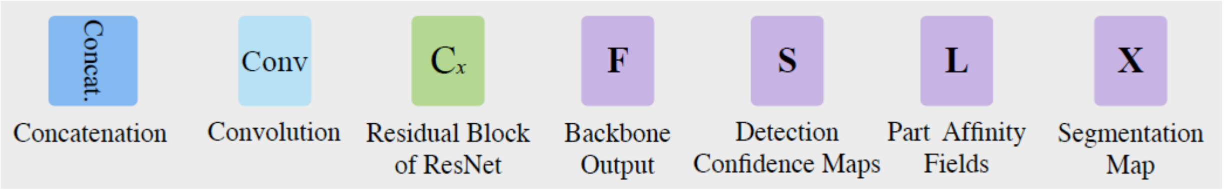


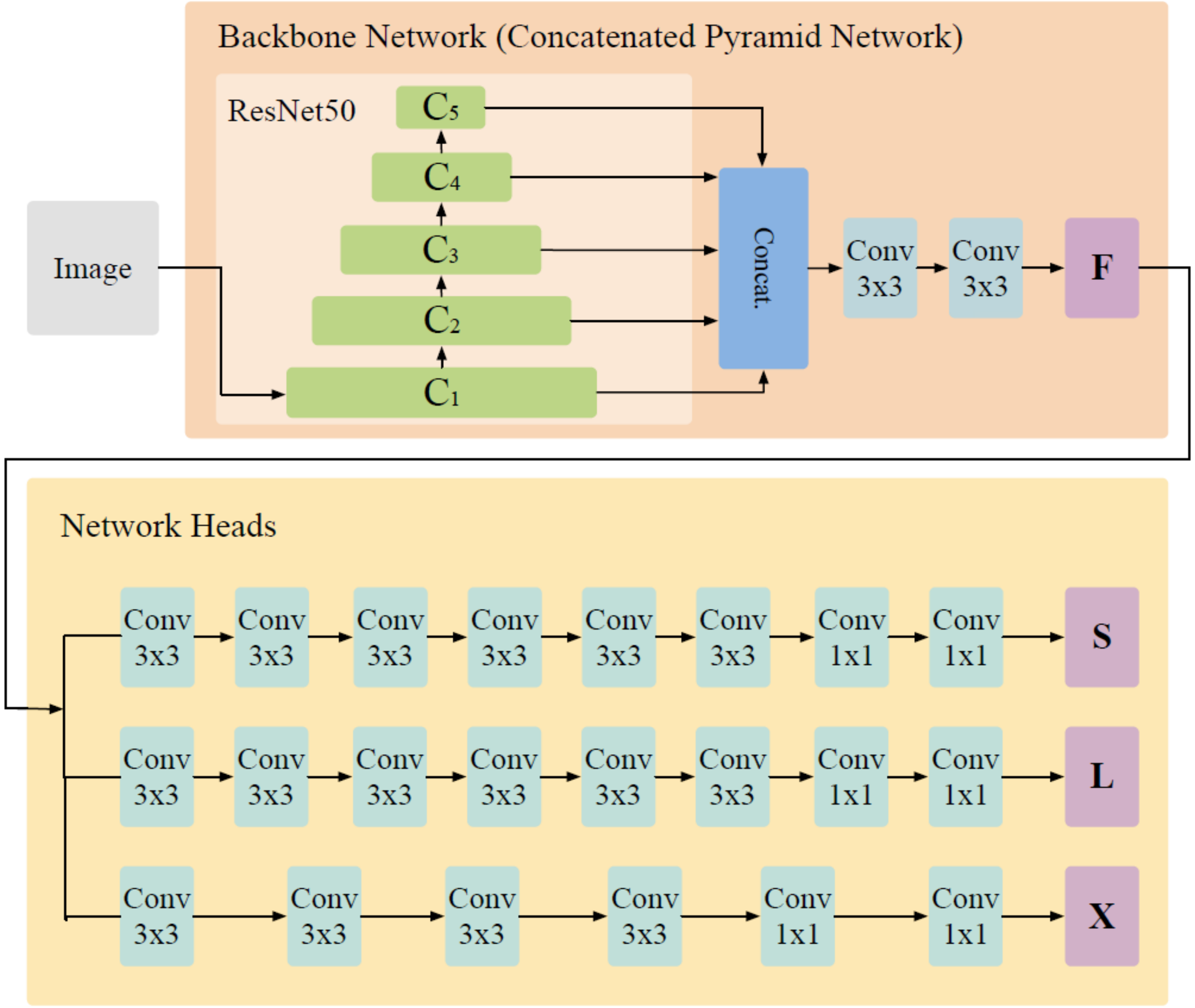
	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR	AR^{50}	AR^{75}	AR^{M}	AR^{L}	date
Megvii (Face++)	0.781	0.941	0.859	0.745	0.833	0.831	0.967	0.898	0.793	0.882	2018-09-09
MSRA	0.765	0.924	0.840	0.730	0.827	0.815	0.958	0.882	0.774	0.872	2018-09-09
The Sea Monsters	0.759	0.921	0.830	0.717	0.821	0.804	0.951	0.867	0.758	0.867	2018-09-09
KPLab	0.751	0.918	0.824	0.715	0.812	0.809	0.954	0.871	0.766	0.869	2018-09-09
DGDBQ	0.749	0.916	0.820	0.710	0.808	0.806	0.952	0.868	0.758	0.872	2018-09-09
ByteDance-SEU	0.742	0.918	0.819	0.706	0.802	0.801	0.953	0.866	0.757	0.860	2018-09-09
fadivugibs	0.740	0.913	0.815	0.706	0.801	0.802	0.952	0.867	0.757	0.864	2018-09-09
SNU CVLAB	0.738	0.907	0.810	0.705	0.800	0.792	0.947	0.855	0.750	0.850	2018-09-09
Megvii (Face++)	0.730	0.917	0.809	0.695	0.781	0.790	0.951	0.859	0.748	0.846	2017-10-29
bangbangren	0.728	0.894	0.796	0.686	0.800	0.787	0.941	0.848	0.736	0.856	2017-10-29
jd_y	0.724	0.906	0.797	0.686	0.791	0.791	0.948	0.854	0.741	0.858	2018-09-09
oks	0.720	0.903	0.797	0.676	0.784	0.771	0.939	0.840	0.725	0.835	2017-10-29
ByteCV	0.717	0.906	0.792	0.686	0.772	0.770	0.944	0.836	0.728	0.830	2018-09-09
Fast-20-FPS	0.717	0.888	0.783	0.674	0.780	0.774	0.928	0.830	0.724	0.841	2018-09-09
Raven-DL	0.713	0.901	0.780	0.673	0.773	0.761	0.932	0.819	0.713	0.827	2018-09-09
G-RMI	0.710	0.879	0.777	0.690	0.752	0.758	0.912	0.819	0.714	0.820	2017-10-29
METU	0.705	0.877	0.772	0.661	0.773	0.749	0.909	0.807	0.701	0.815	2018-09-09
TFMAN	0.702	0.892	0.770	0.656	0.763	0.747	0.914	0.806	0.693	0.821	2018-09-09
FAIR Mask R-CNN	0.692	0.904	0.760	0.649	0.763	0.752	0.937	0.811	0.703	0.818	2017-10-29
SJTU	0.688	0.875	0.759	0.646	0.751	0.736	0.910	0.798	0.689	0.802	2017-10-29
Huya	0.654	0.870	0.717	0.609	0.722	0.700	0.900	0.755	0.648	0.771	2018-09-09
iie-samsung-pose	0.636	0.852	0.698	0.582	0.713	0.688	0.885	0.741	0.626	0.774	2017-10-29
CMU-Pose	0.618	0.849	0.675	0.571	0.682	0.665	0.872	0.718	0.606	0.746	2016-09-16
G-RMI_2016	0.605	0.822	0.662	0.576	0.666	0.662	0.866	0.714	0.619	0.722	2016-09-16
DL-61	0.544	0.753	0.509	0.583	0.543	0.708	0.827	0.692	0.753	0.768	2016-09-16

CNN Architecture

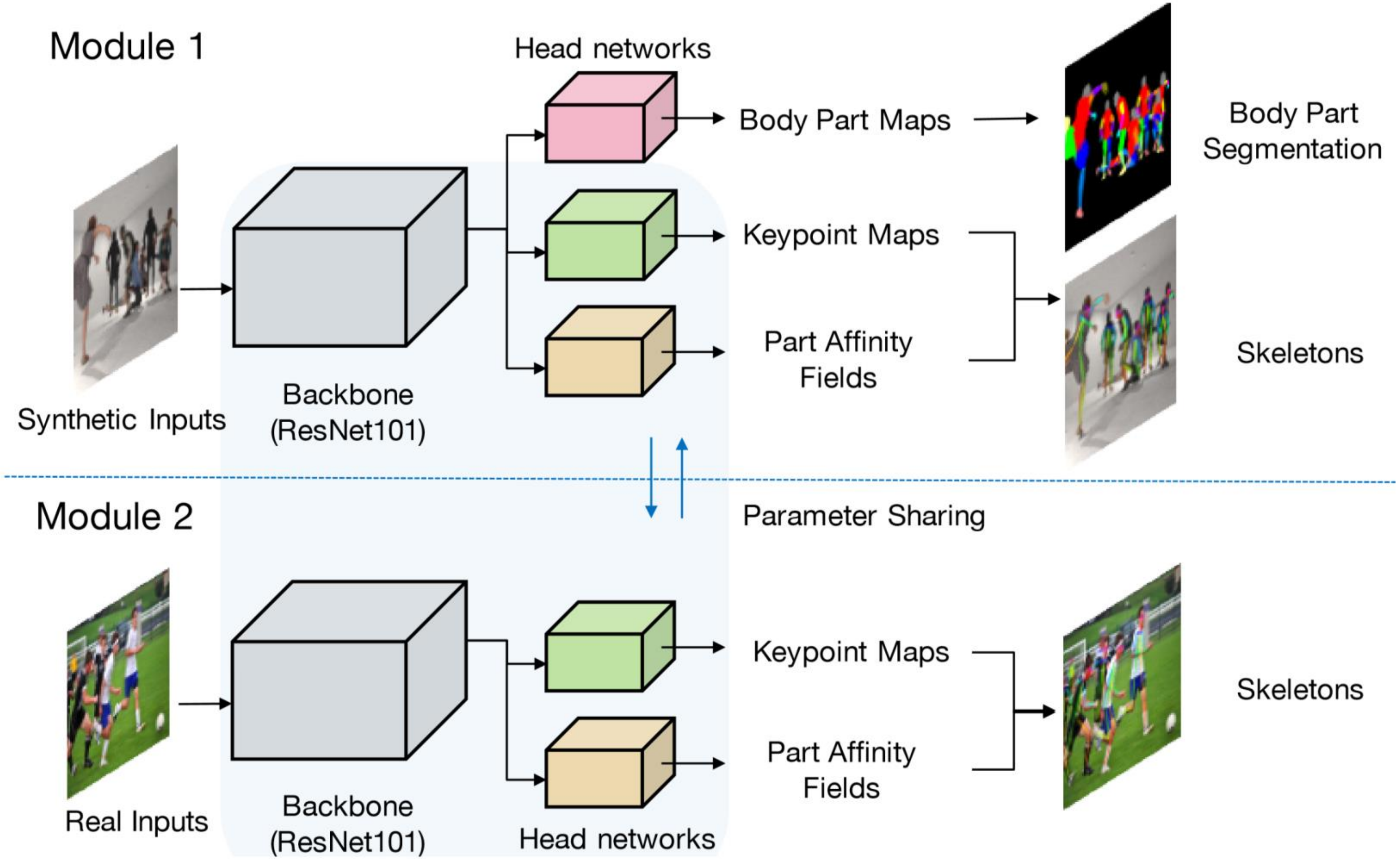


CNN Architecture





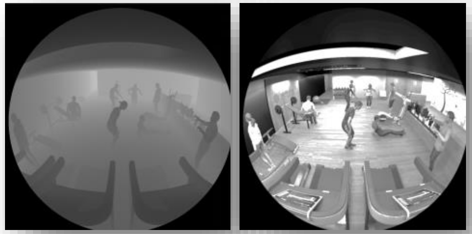
CNN Training



$$L = L_{pose}(D_r^{pose}) + L_{pose}(D_s^{pose}) + L_{part}(D_s^{part})$$

Synthetics Data Strategy

*Synthetic Data used for training



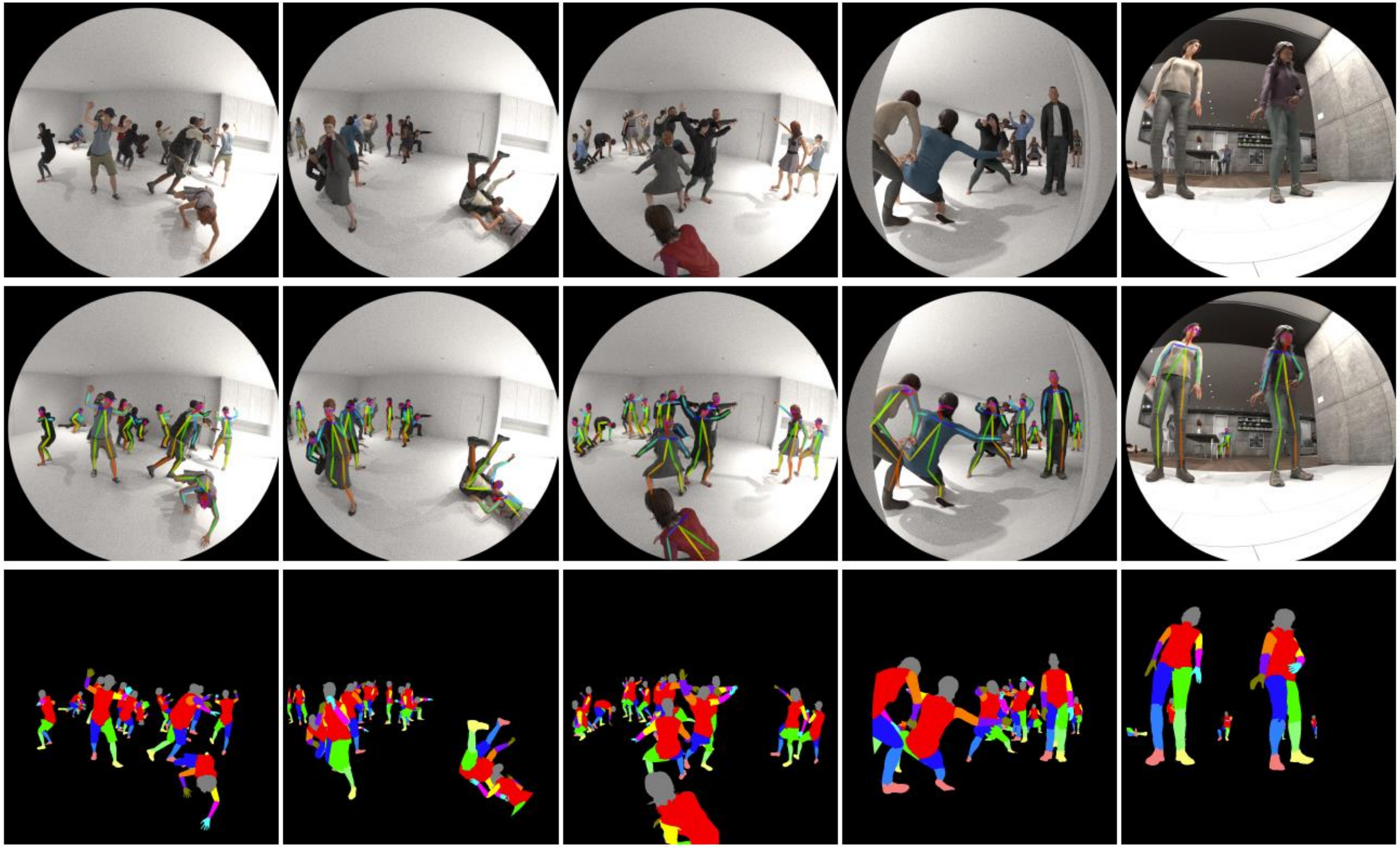
Sensor

Human diversity

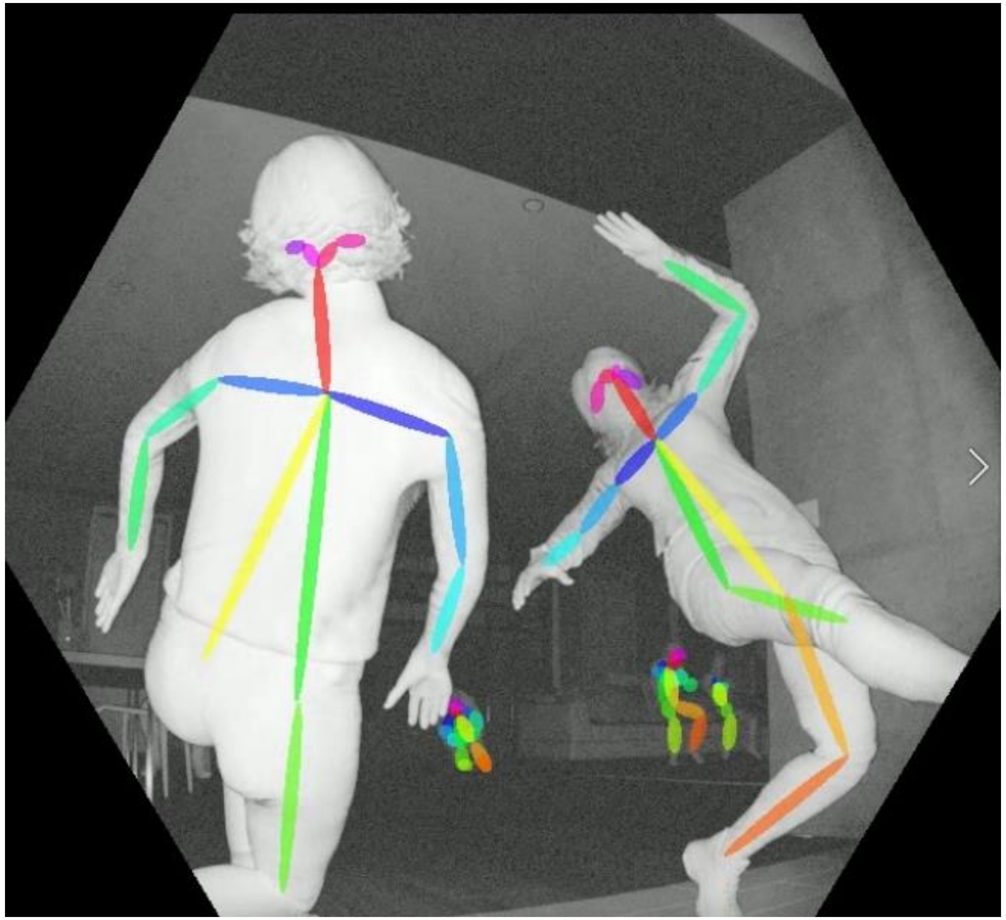
Environments

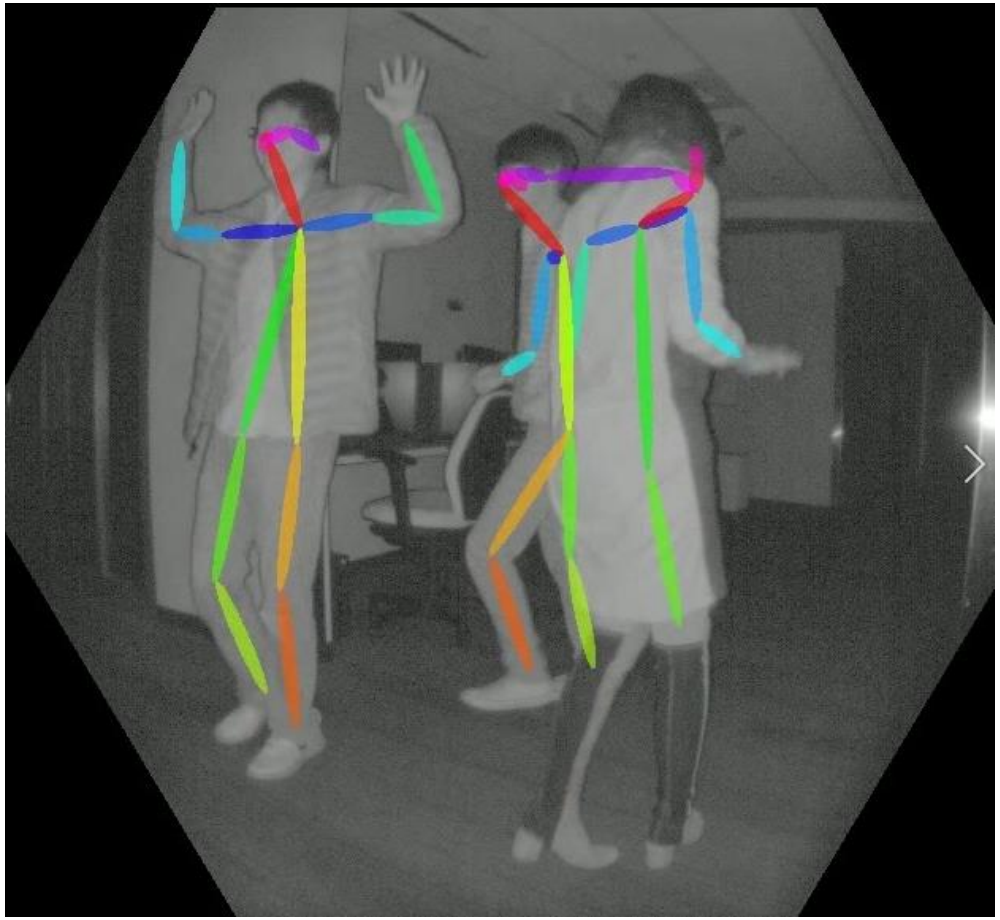
Labeling

Synthetic Data



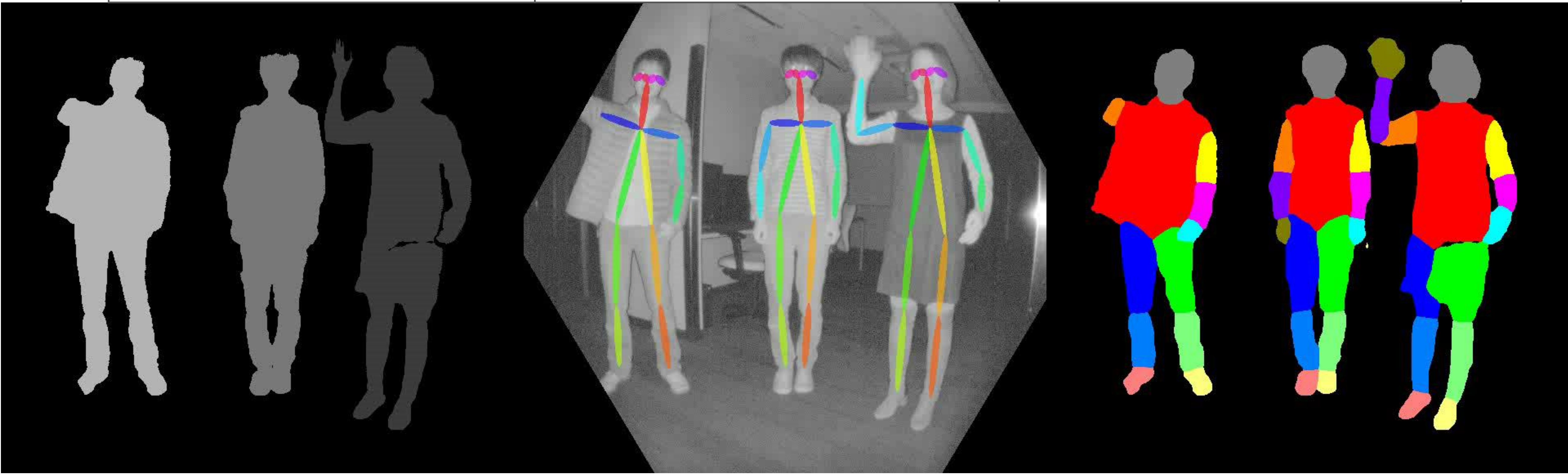
Reality Gap between Real and Synthetics





Results on Real AB Input

Instance Segmentation	Skeleton	Body Part Segmentation

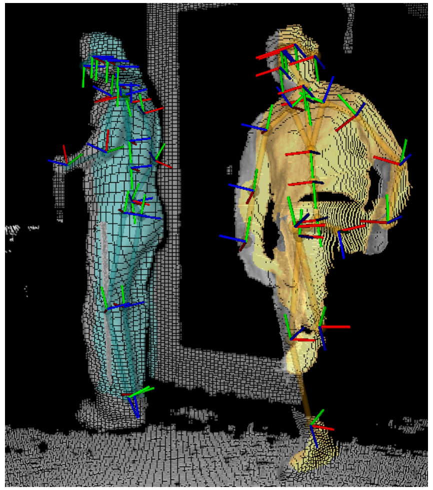


Results on Real RGB Input

Live Skeleton Tracking on iPhone

3D Model Fitting Using Depth Map

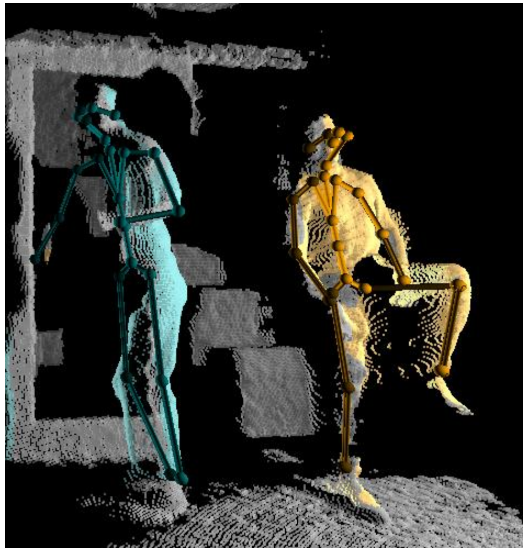
Model Fitting



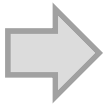
<u>Input</u>
- AB frame
  - Linked 2D DNN keypoints
- Depth frame

<u>Output</u>
- 3D Joint Locations
- Joint Orientation
- Temporal Identity

Side views:



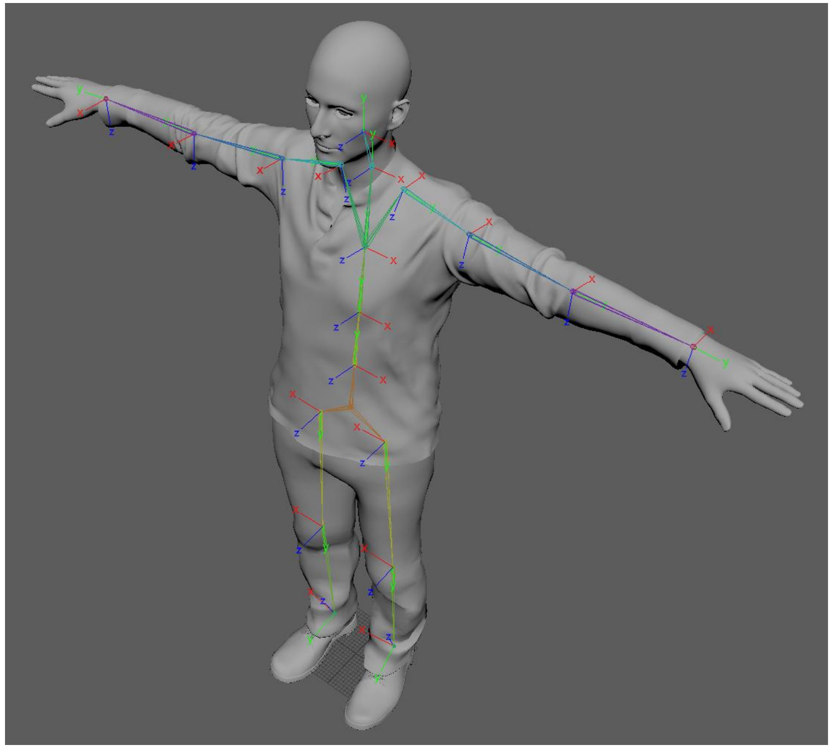
Model Fitting - Challenges

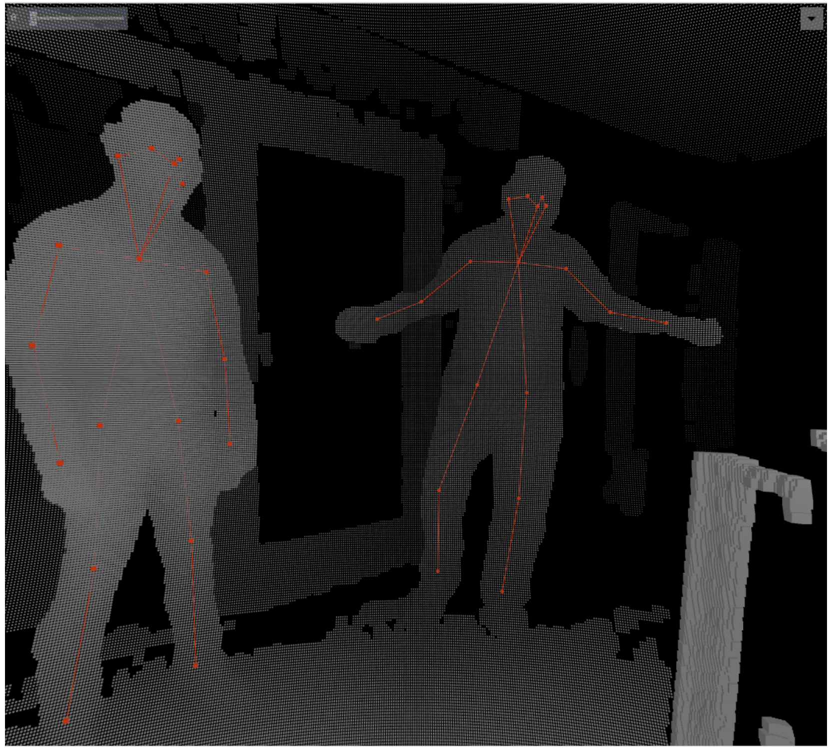


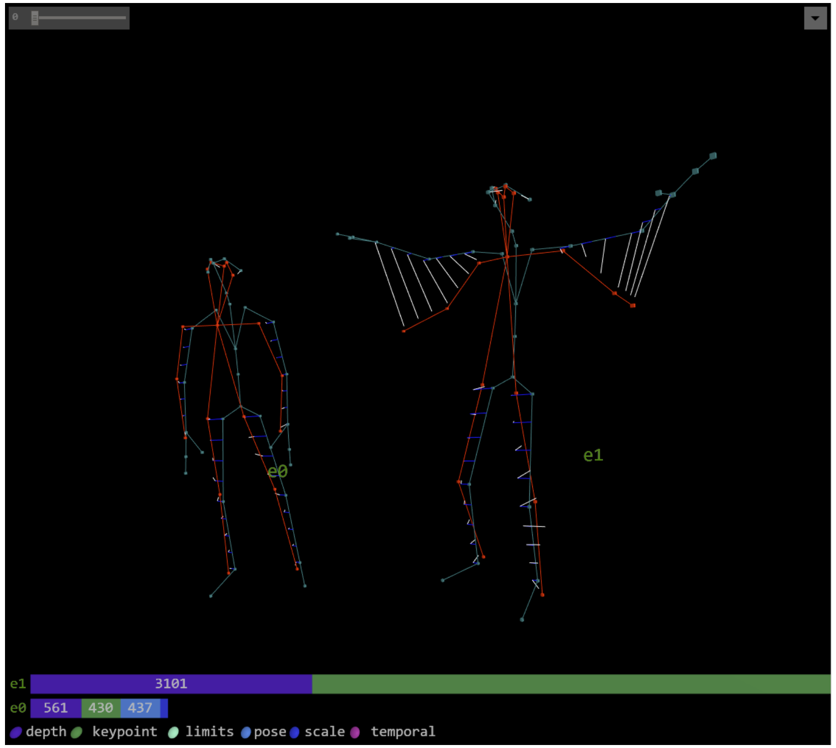
Side view

- **Easy Case**
  - Frontal view, un-occluded
- **Challenging Cases**
  - Unreliable depth
    - Dark clothes (IR absorbing)
    - FOV cut-off
  - Partial view of the person
    - Self-occlusions (e.g. side view)
    - People occluding other people

Model Fitting – Skeleton Based Tracking







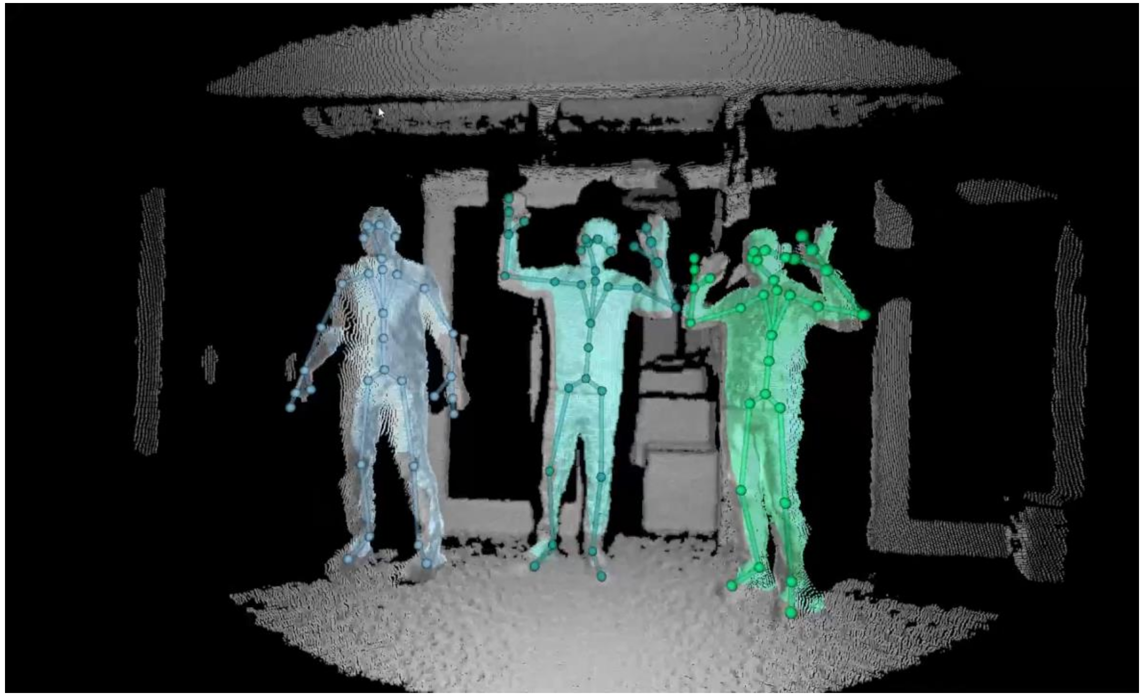
Kinematic Model
- Joint angles
- Scaling factor
- Global rigid transform

Input
- Depth image
- Linked DNN keypoints in 2D
  (from AB image)

Energy Data Terms
- 2D keypoint reprojection
- 3D surface depth displacement

Energy Regularization Terms
- Anatomical joint limits
- Pose prior regularization
- Scale prior regularization
- Temporal coherency

Model Fitting – Results



Demo

Model Fitting – Results

Runtime Speed

Hardware	CPU	GPU	Depth Speed (ms)	DNN Speed (ms)	Model Fitting Speed[1 person] (ms)	SDK Framerate (FPS)
Z440	Xeon(R) CPU E5-1660 v4 @ 3.20GHz 3.20 GHz	GTX 1080Ti	3.0	19.2	2.9	50
Z420	Xeon(R) CPU E5-1620 0 @ 3.60GHz 3.60 GHz	GTX 1070	4.0	30.2	3.3	30
Surface Book	I7-8650U CPU @ 1.90GHz 2.11 GHz	GTX 1060M	6.2	47.1	3.6	17

Summary

- Azure Kinect Body Tracking SDK
  - DNN based algorithm
  - Using synthetic data
  - Handling challenging poses and camera angles
- Beta release in Windows and Linux: https://docs.microsoft.com/en-us/azure/kinect-dk/sensor-sdk-download

THANKS!

Acknowledgement to the dev team of the AKBT SDK

Contact: Zicheng Liu
zliu@microsoft.com