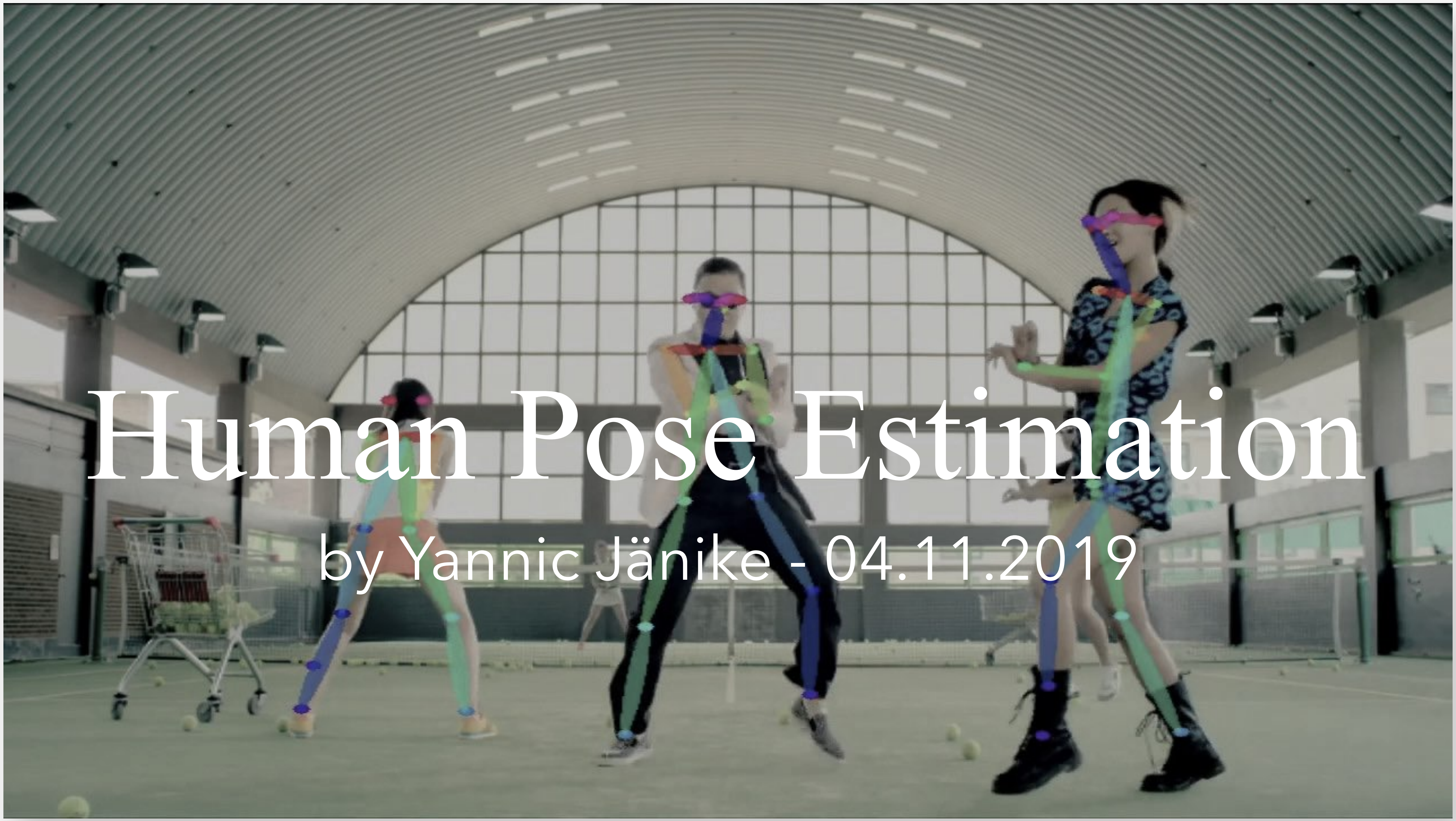


Human Pose Estimation

by Yannic Jänike - 04.11.2019

https://www.youtube.com/watch?v=mxKIUO_tjcg

Human Pose Estimation

1. What is Human Pose Estimation

2. OpenPose Pipeline

3. Bottom Up or Top Down Approach

https://www.youtube.com/watch?v=mxKlUO_tjcg

What is Human Pose Estimation (HPE)?

Pose Estimation is predicting the body part or joint positions of a person from an image or a video.

Where are we in terms of solving the problem of human pose Estimation?

link

Multi Person Human Pose Estimation - Cao et al. (2018)

Real Time Human Pose Estimation on your smartphone or Laptop: **or** *https://storage.googleapis.com/tfjs-models/demos/posenet/camera.html*

Why is this interesting for Intelligent Robotics?

Care/service robots:

- detecting falls
- bad posture

Autonomous Driving:

- intentions of pedestrians

Interaction between humans involves a lot **non verbal cues**

- understanding the direction of a arm showing something
- „give me that object!" with a pointed finger
- Robotic task learning from watching humans performing that task

The different types of HPE

How many persons?

What is our input?

What is the output?

How do we define our model?

Single vs Multi Person HPE

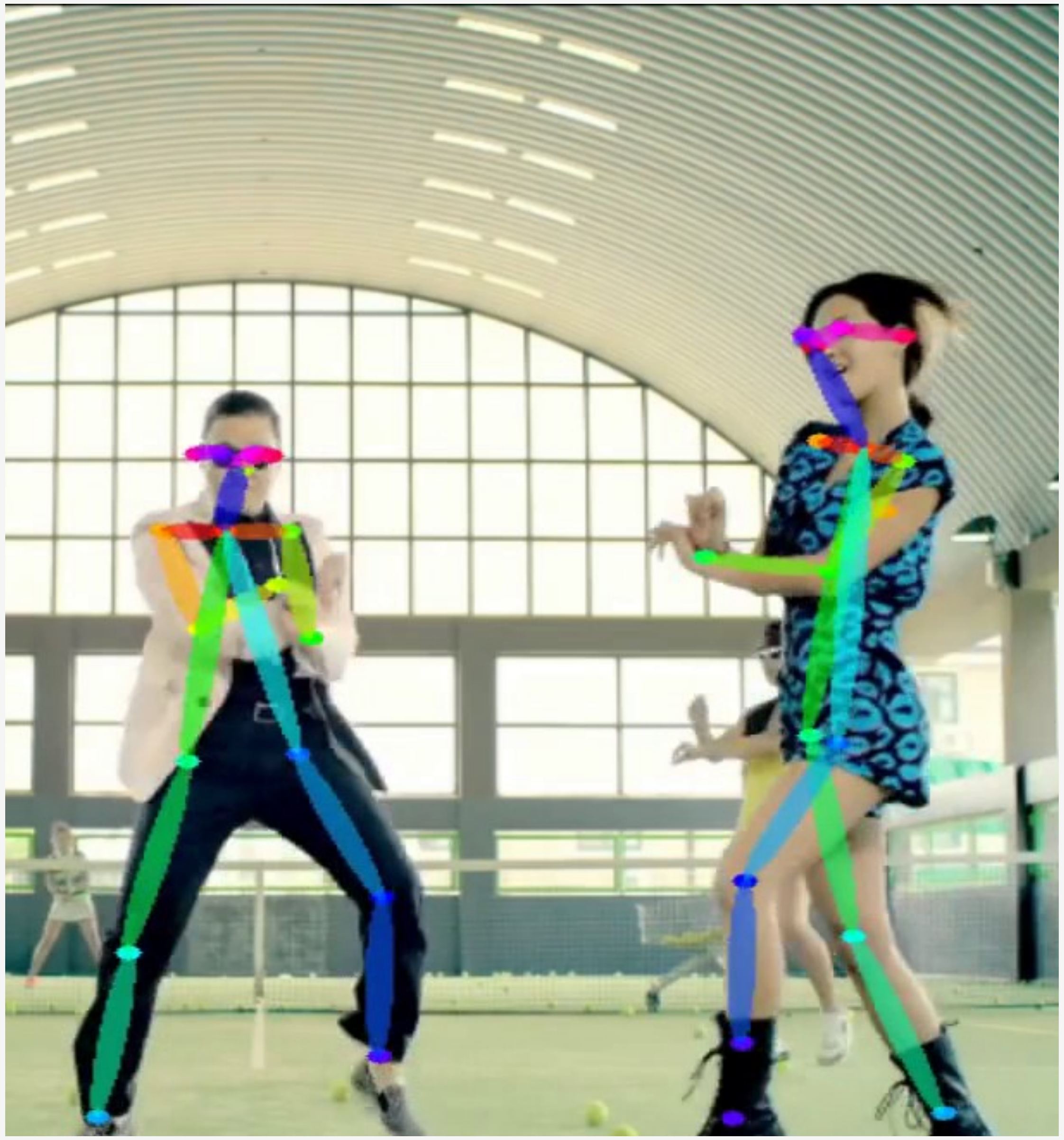
(SPPE vs MPPE)

Single Person:

- Only one is in the input

Multi Person:

- Arbitrary number of people in the input
- Alogrithms need to differentiate between humans



Multi Person Pose Estimation
from: https://www.youtube.com/watch?v=mxKIUO_tjcg

Input Modality

Techniques Used:

- RGB Images
- Depth (Time of flight) Images
- Infrared (IR) Images

Depth image (top) vs IR image (bottom)
http://www.norrislabs.com/images/depth.png
https://i.ytimg.com/vi/w6-b5Bpr1iY/hqdefault.jpg

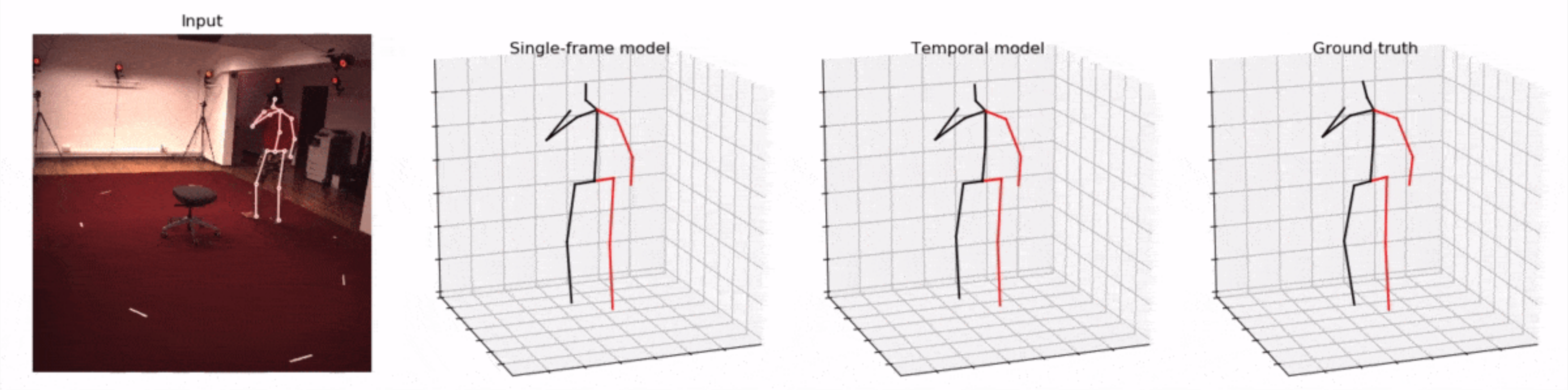
Static Images vs Video

Static:

- computationally less demanding
- Less accurate
- inconsistency problems

Video - frame by frame or with temporal information :

- consecutive frames share huge portion of information -> *temporal dependency*
- computational more demanding



Single-frame model vs temporal model - Pavllo et al. (2018)

<u>*link*</u>

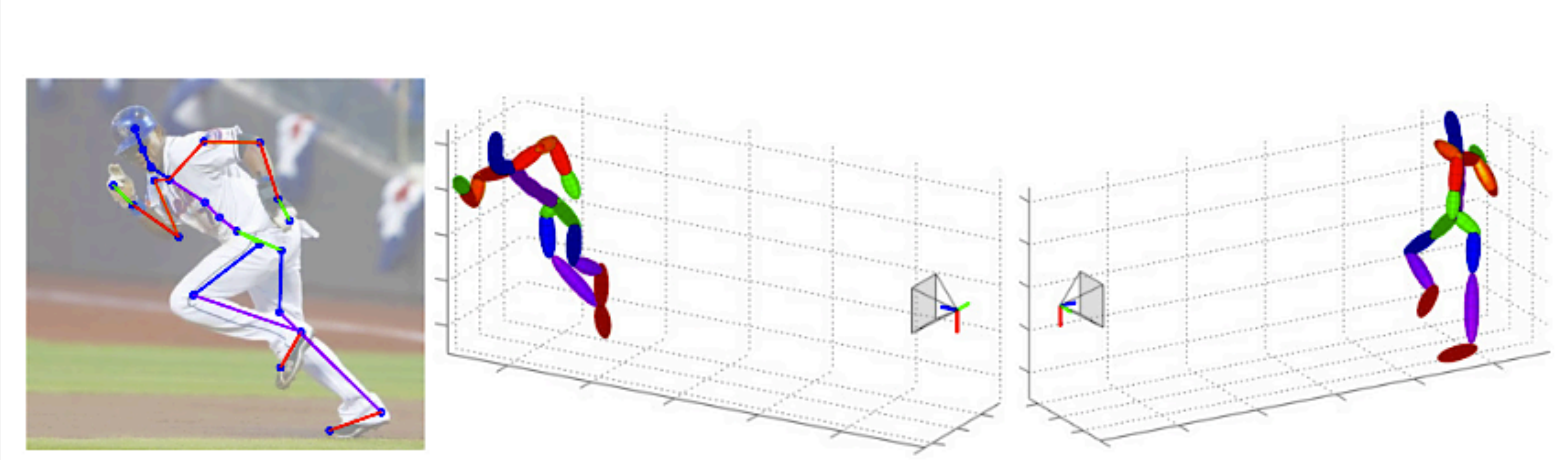
2D vs 3D Output Model

2D

- location of body joint in the image
- in terms of pixel values

3D

-three dimensional spatial arrangement of all body joints

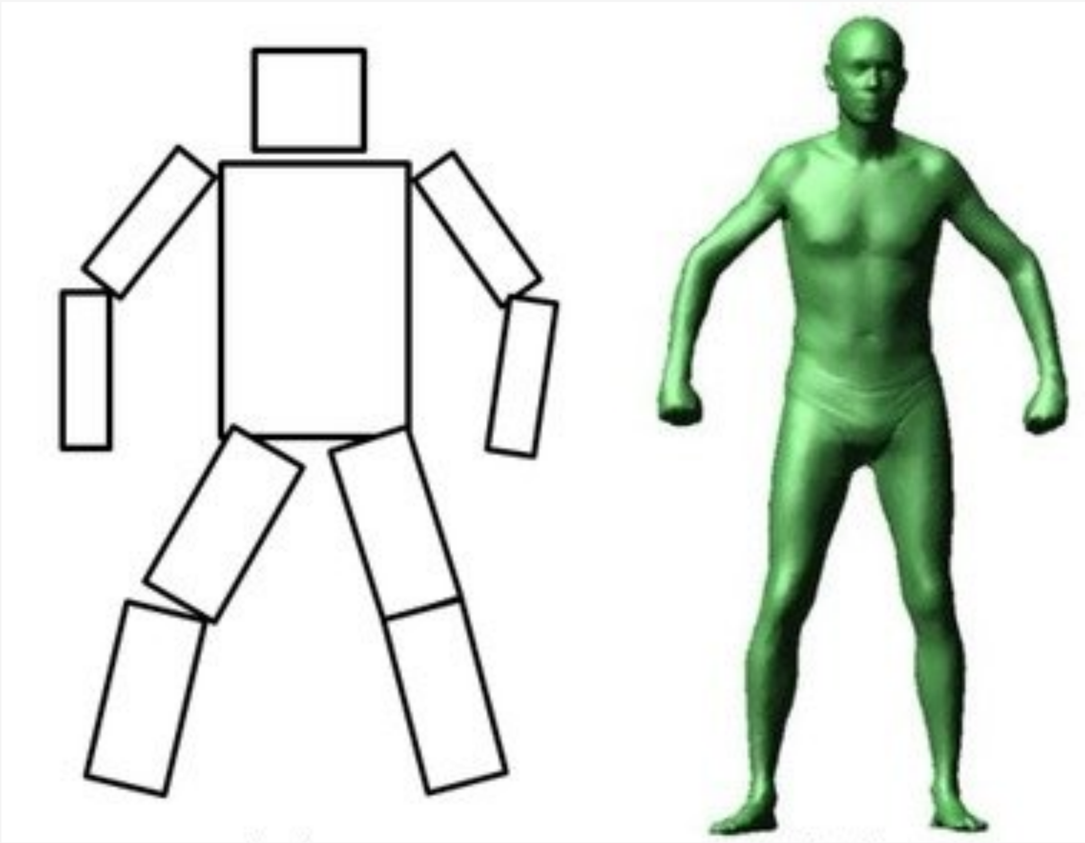


2D (left) vs 3D (middel and right) output model - Chen et al. (2017)

Body Model

Must be defined beforehand!

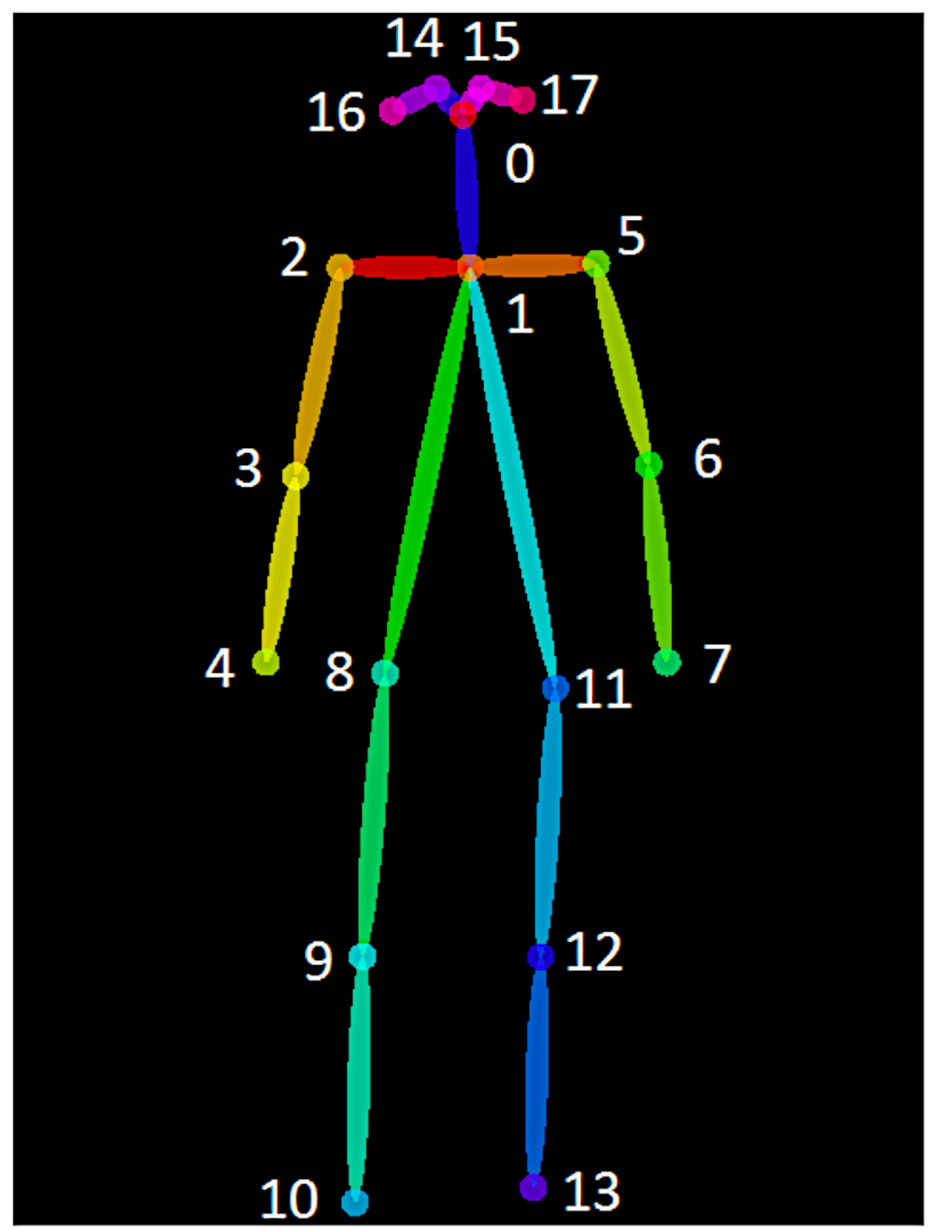
- N-joint rigid kinematic skeleton model
- highly detailed mash models
- shape-based body model (primitive, used in early HPE)



Shape (left) vs mash (right) model
https://www.mdpi.com/1424-8220/16/12/1966

N-joint rigid kinematic skeleton model

- representation as a graph
- each vertex V = joint
- edges can encode constraints



N-joint model
https://nanonets.com/blog/content/images/2019/04/Screen-Shot-2019-04-11-at-5.17.56-PM.png

Bottom Up vs. Top Down

Detect all joints from multiple persons in the frame

assemble human body pose estimation(s) from detected joints

Detect all humans in the frame

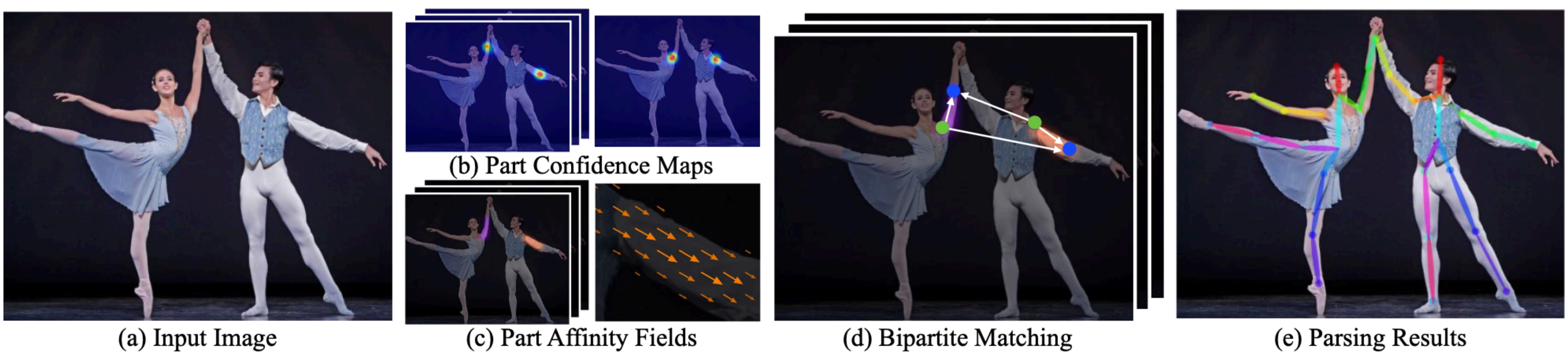
On each cut out, perform human pose estimation

OpenPose: Realtime Multi-Person 2D PoseEstimation using Part Affinity Fields

Zhe Cao, Student Member, IEEE, Gines Hidalgo, Student Member, IEEE, Tomas Simon, Shih-En Wei, and Yaser Sheikh
(Submitted on 18 Dec 2018 (v1), last revised 30 May 2019 (this version, v2))

How Many Persons?	**Multiple Person**
What is our input?	RGB Images
	Video
What is the output?	**2D Model**
How do we define our model?	N-joint

OpenPose: Realtime Multi-Person 2D PoseEstimation using Part Affinity Fields

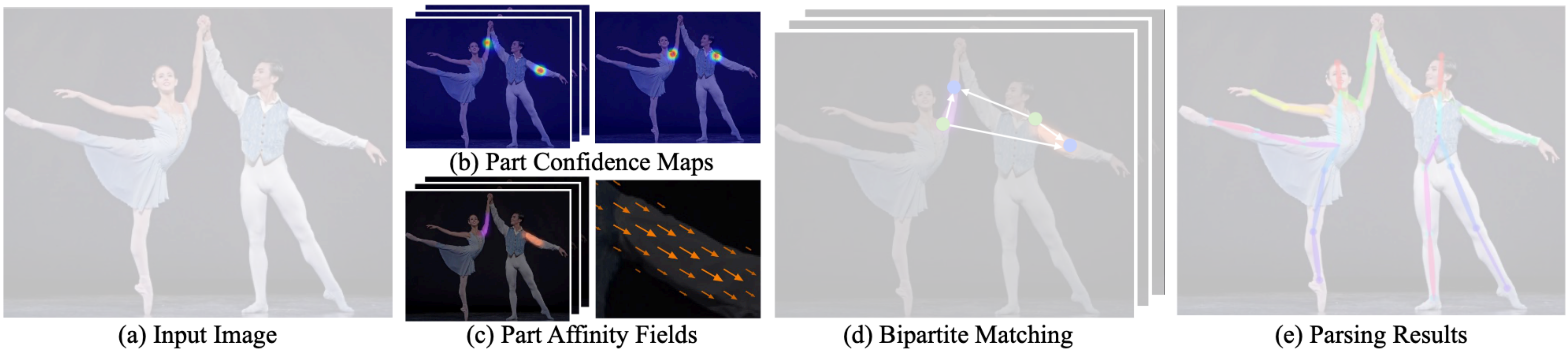


Human Pose Estimation Pipeline - Chao et al. (2018)

Pipeline:

- (b) Part Confidence Maps (PCM)
- (c) Part Affinity Fields (PAF)
- (d) Bipartite Matching
- (e) Parsing Results

OpenPose: Realtime Multi-Person 2D PoseEstimation using Part Affinity Fields

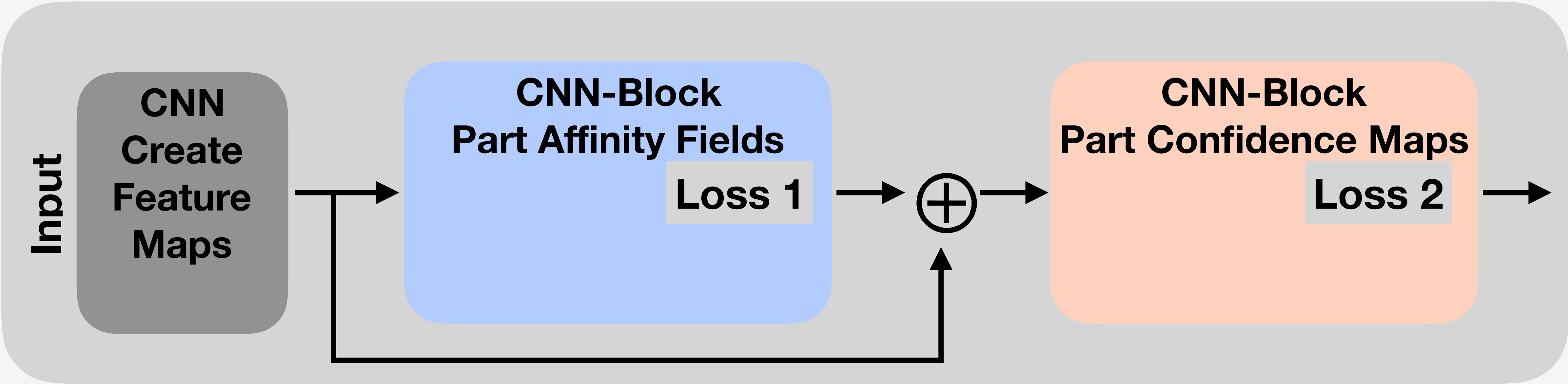


Human Pose Estimation Pipeline - Chao et al. (2018)

Pipeline:

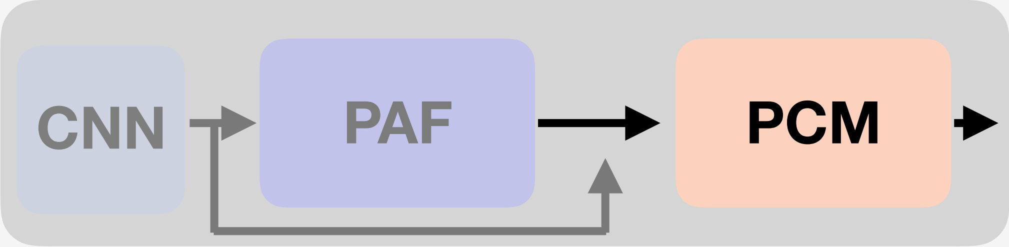
- (b) Part Confidence Maps (PCM)
- (c) Part Affinity Fields (PAF)
- (d) Bipartite Matching
- (e) Parsing Results

Network Architecture

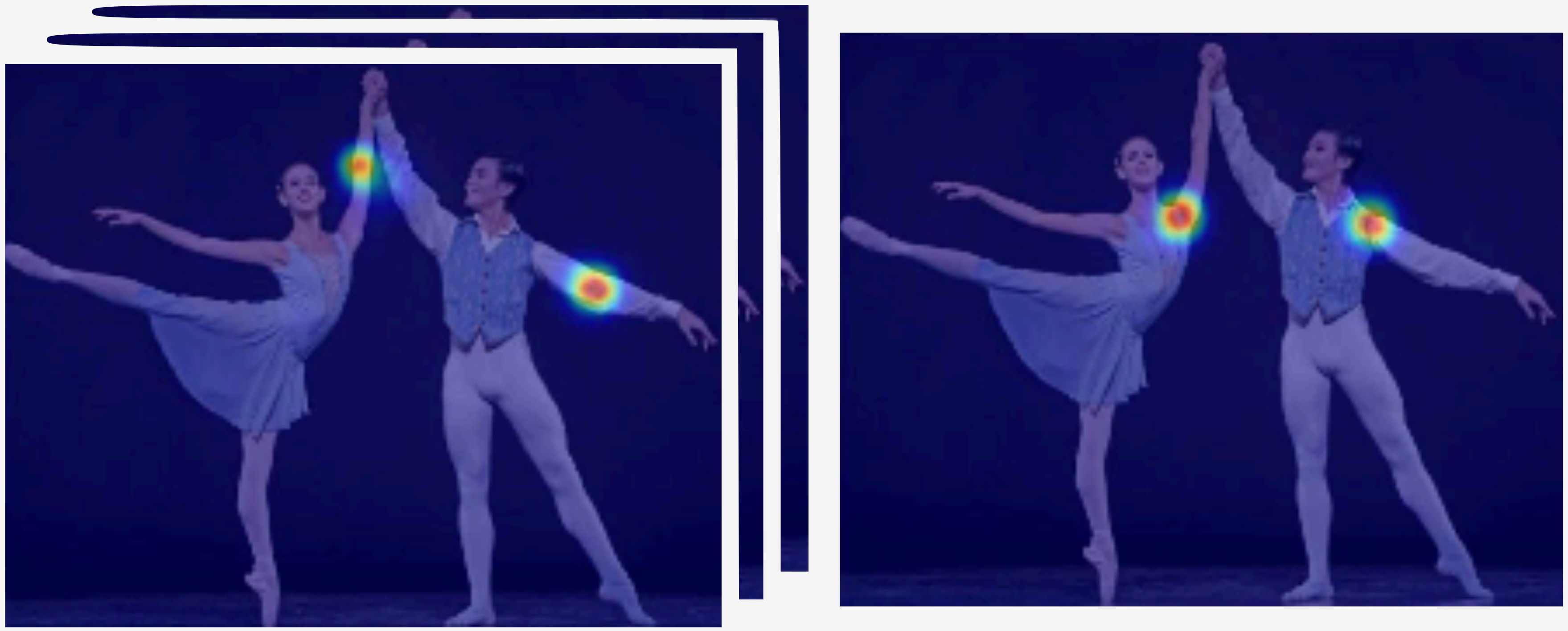


Architecture of the Neural Networks - Adapted from Chao et al. (2018)

- iterative prediction
- **intermediate supervision**
  - Loss calculation after each Block (compared to groundtruth)
- Concatenation of Feature Maps and Part Affinity Fields
- PCM is trained on latests update of PAF

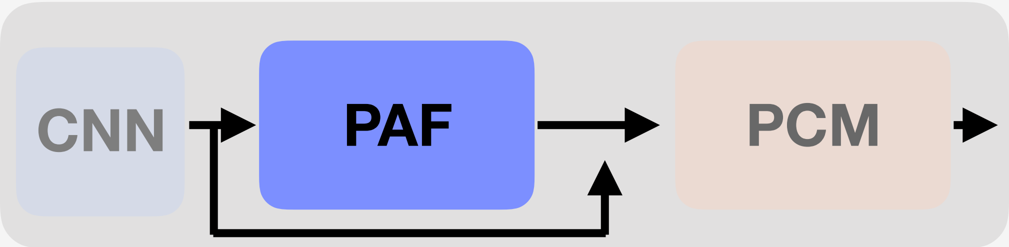


Part Confidence Maps



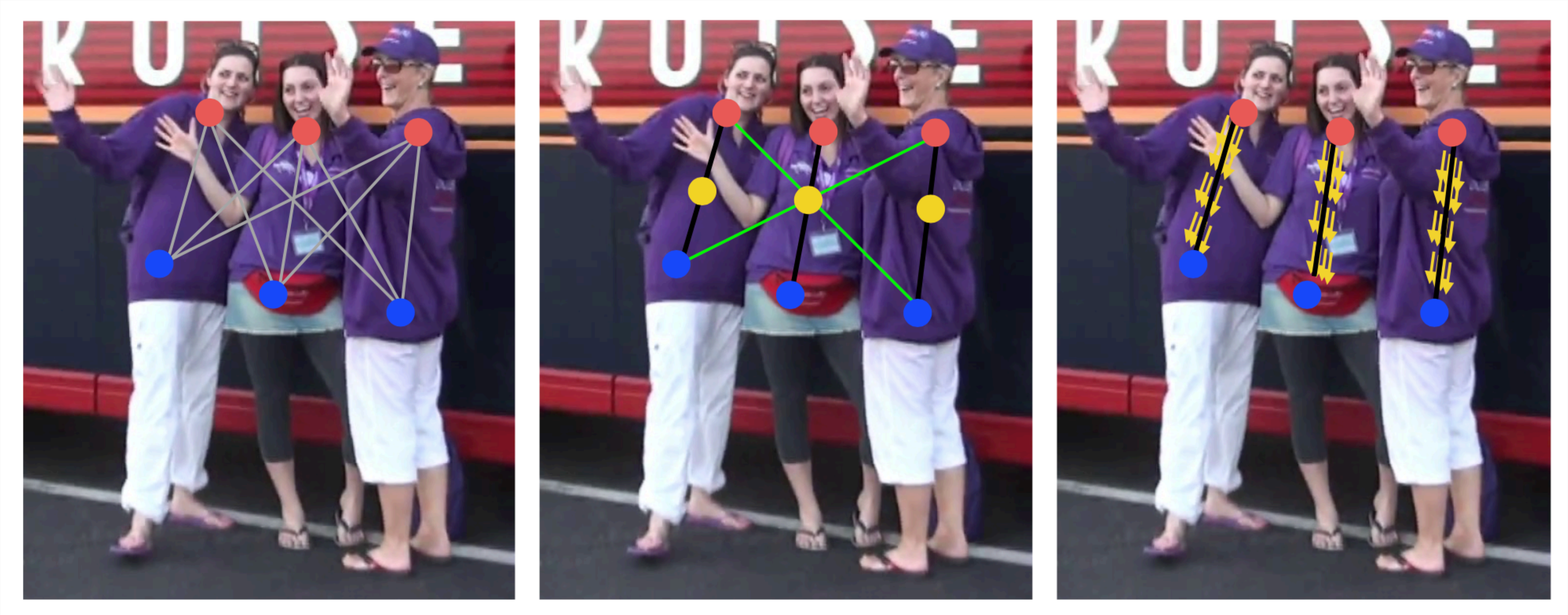
Part Confidence Maps - Chao et al. (2018)

- all of different joints are **detected separately**
- CNN predicts a set of 2D confidence maps
- joint locations are **Gaussian peaks** on a map



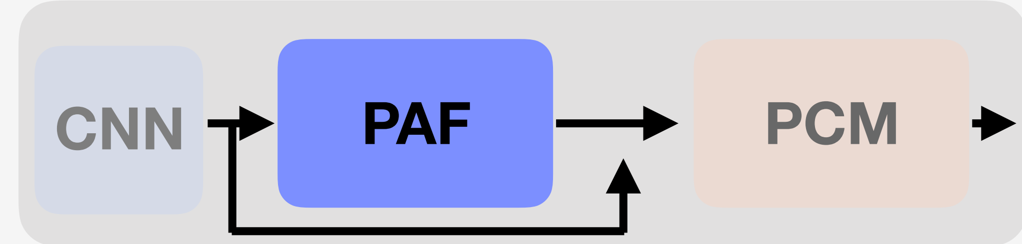
Part Affinity Fields

We have the set of detected body parts. How do we assemble possibly multiple persons?



Part Confidence Maps - Chao et al. (2018)

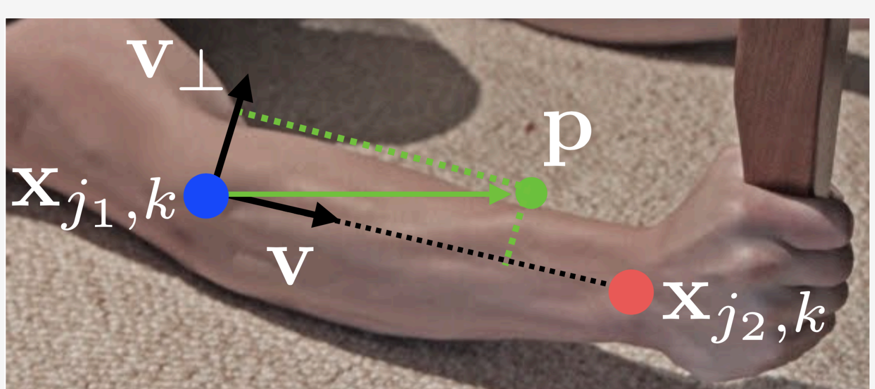
? **Middel Points?** **Part Affinity Fields!**



Part Affinity Fields

Part Confidence Maps - Chao et al. (2018)

- 2D vector field for each limb (connection between the two joints)
- preserve both **location** and **orientation** information
- color encodes angle and vector size encodes likelihood

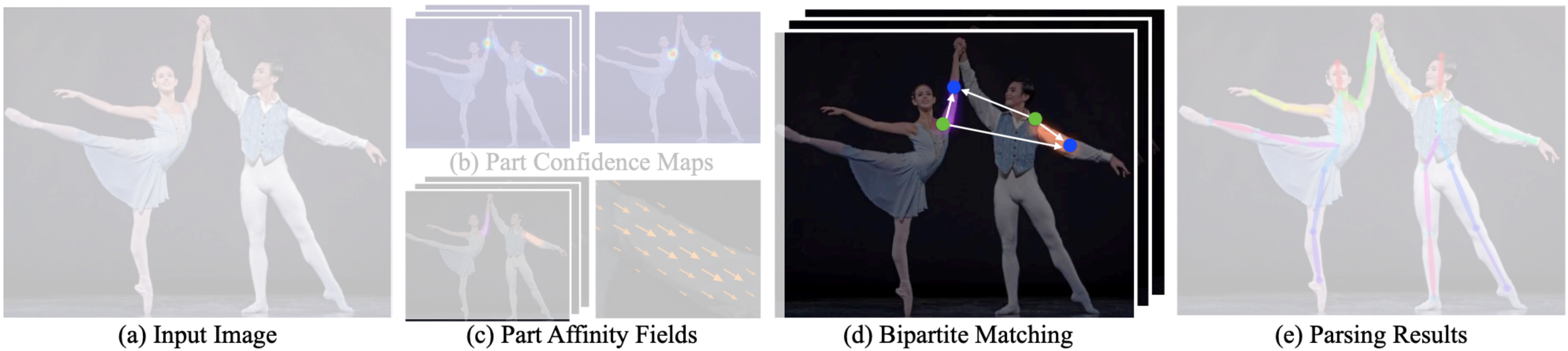


vetor connecting joints - Chao et al. (2018)

joint one of person k   joint two of person k

$$\begin{cases} \text{if } \mathrm{p} \text{ is on limb, } \mathrm{p} \text{ is a vector pointing from } \mathrm{j}_1 \text{ to } \mathrm{j}_2 \\ \text{else } \mathrm{p} = 0 \end{cases}$$

OpenPose: Realtime Multi-Person 2D PoseEstimation using Part Affinity Fields



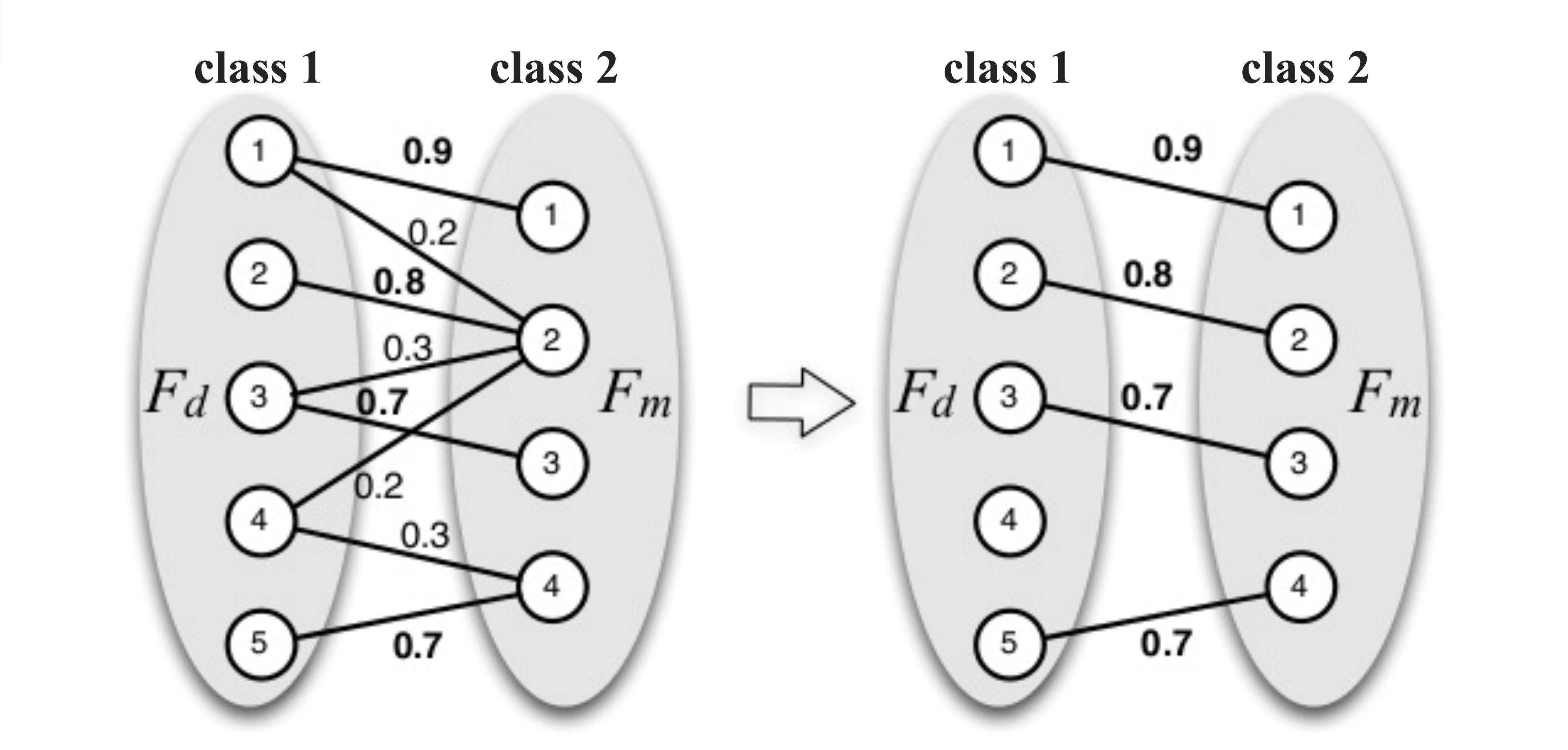
Human Pose Estimation Pipeline - Chao et al. (2018)

Pipeline:

- (b) Part Confidence Maps (PCM)
- (c) Part Affinity Fields (PAF)
- (d) Bipartite Matching
- (e) Parsing Results

Bipartite Matching

- No two points from class 1 can have connection to same point in class 2
- can be solved using the Hungarian Algorithm

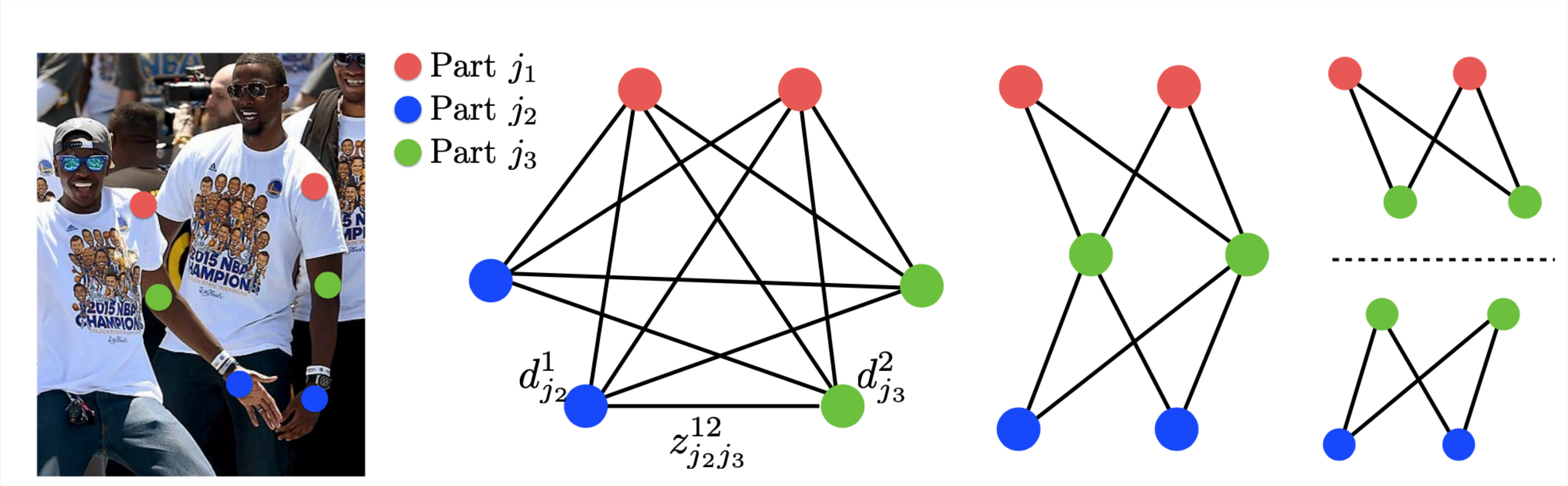


https://image.slidesharecdn.com/defense-150722070628-lva1-app6892/95/phd-dissertation-defense-april-2015-30-638.jpg?cb=1437548981

Bipartite Matching

Finding the optimal joint connections corresponds to a K-dimensional matching problem.

- reduce NP-Hard problem into smaller sub problems

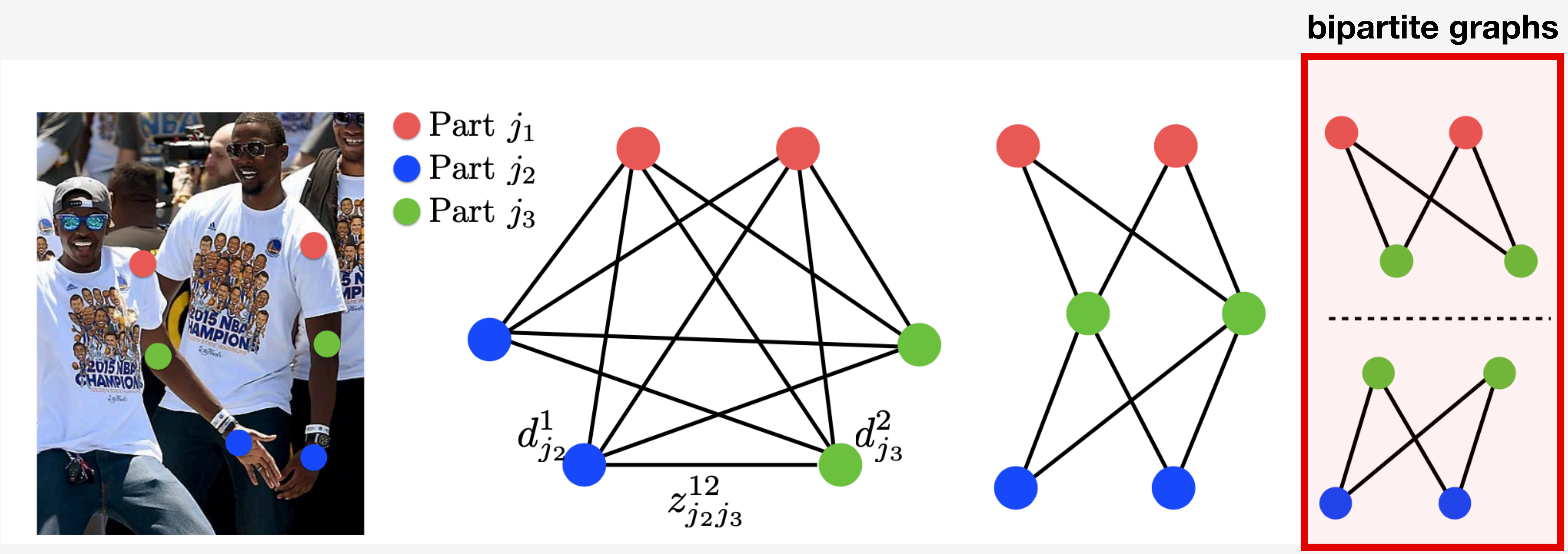


Graph Matching - Chao et al. (2018)

Bipartite Matching

Finding the optimal parse corresponds to a K-dimensional matching problem.
This is known to be NP-Hard.

- reduce NP-Hard problem into smaller sub problems
- from limb candidates, full-body poses are computed
- weights on edges are the Integral of the PAFs



Graph Matching - Chao et al. (2018)

Results & Discussion

Benchmark Datasets:

- MPII human multi-person dataset
- COCO key point challenge dataset

Measurement:

- **mean Average Precision** (mAP) of all body parts
- **average inference**/optimization **time** per image in seconds

Results & Discussion - MPII

Method	Hea	Sho	Elb	Wri	Hip	Kne	Ank	**mAP**	s/image
				Full testing set					
DeeperCut [2]	78.4	72.5	60.2	51.0	57.2	52.0	45.4	59.5	485
Iqbal et al. [41]	58.4	53.9	44.5	35.0	42.2	36.7	31.1	43.1	10
Levinko et al. [71]	89.8	85.2	71.8	59.6	71.1	63.0	53.5	70.6	-
ArtTrack [47]	88.8	87.0	75.9	64.9	74.2	68.8	60.5	74.3	**0.005**
Fang et al. [6]	88.4	86.5	78.6	**70.4**	74.4	**73.0**	**65.8**	76.7	-
Newell et al. [48]	**92.1**	89.3	78.9	69.8	76.2	71.6	64.7	77.5	-
Fieraru et al. [72]	91.8	**89.5**	**80.4**	69.6	**77.3**	71.7	65.5	**78.0**	-
Ours (one scale)	89.0	84.9	74.9	64.2	71.0	65.6	58.1	72.5	0.005
Ours	91.2	87.6	77.7	66.8	75.4	68.9	61.7	75.6	**0.005**

Results on the MPII dataset - Chao et al. (2018)

- Outperforms previous state of the art (DeeperCut) by **13% mAP**
- inference time is **6 order of magnitude less**
- **PAFs** are effective for feature representation

Results & Discussion - MPII

	Method	Hea	Sho	Elb	Wri	Hip	Kne	Ank	**mAP**	s/image
	Full testing set									
	DeeperCut [2]	78.4	72.5	60.2	51.0	57.2	52.0	45.4	59.5	485
	Iqbal et al. [41]	58.4	53.9	44.5	35.0	42.2	36.7	31.1	43.1	10
	Levinko et al. [71]	89.8	85.2	71.8	59.6	71.1	63.0	53.5	70.6	-
	ArtTrack [47]	88.8	87.0	75.9	64.9	74.2	68.8	60.5	74.3	**0.005**
	Fang et al. [6]	88.4	86.5	78.6	**70.4**	74.4	**73.0**	**65.8**	76.7	-
	Newell et al. [48]	**92.1**	89.3	78.9	69.8	76.2	71.6	64.7	77.5	-
top-down	Fieraru et al. [72]	91.8	**89.5**	**80.4**	69.6	**77.3**	71.7	65.5	**78.0**	-
bottom-up	Ours (one scale)	89.0	84.9	74.9	64.2	71.0	65.6	58.1	72.5	0.005
bottom-up	Ours	91.2	87.6	77.7	66.8	75.4	68.9	61.7	75.6	**0.005**

Results on the MPII dataset - Chao et al. (2018)

- Top-down approach outperforms bottom-up
- MPII is only images, not videos

Fieraru et al.:

Three Modules: - human candidate detector
- single-person pose estimator (Cascade pyramide network)
- human pose tracker

Results & Discussion - COCO

Team	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}
Top-Down Approaches					
Megvii [43]	78.1	94.1	85.9	74.5	83.3
MRSA [44]	76.5	92.4	84.0	73.0	82.7
The Sea Monsters*	75.9	92.1	83.0	71.7	82.1
Alpha-Pose [6]	71.0	87.9	77.7	69.0	75.2
Mask R-CNN [5]	69.2	90.4	76.0	64.9	76.3

Team	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}
Bottom-Up Approaches					
METU [50]	70.5	87.7	77.2	66.1	77.3
TFMAN*	70.2	89.2	77.0	65.6	76.3
PersonLab [49]	68.7	89.0	75.4	64.1	75.5
Associative Emb. [48]	65.5	86.8	72.3	60.6	72.6
Ours	64.2	86.2	70.1	61.0	68.8
Ours [3]	61.8	84.9	67.5	57.1	68.2

Results on the MS COCO dataset, Top-Down (left) and Bottom-Up (right) - Chao et al. (2018)

- Top-down approach outperforms bottom-up

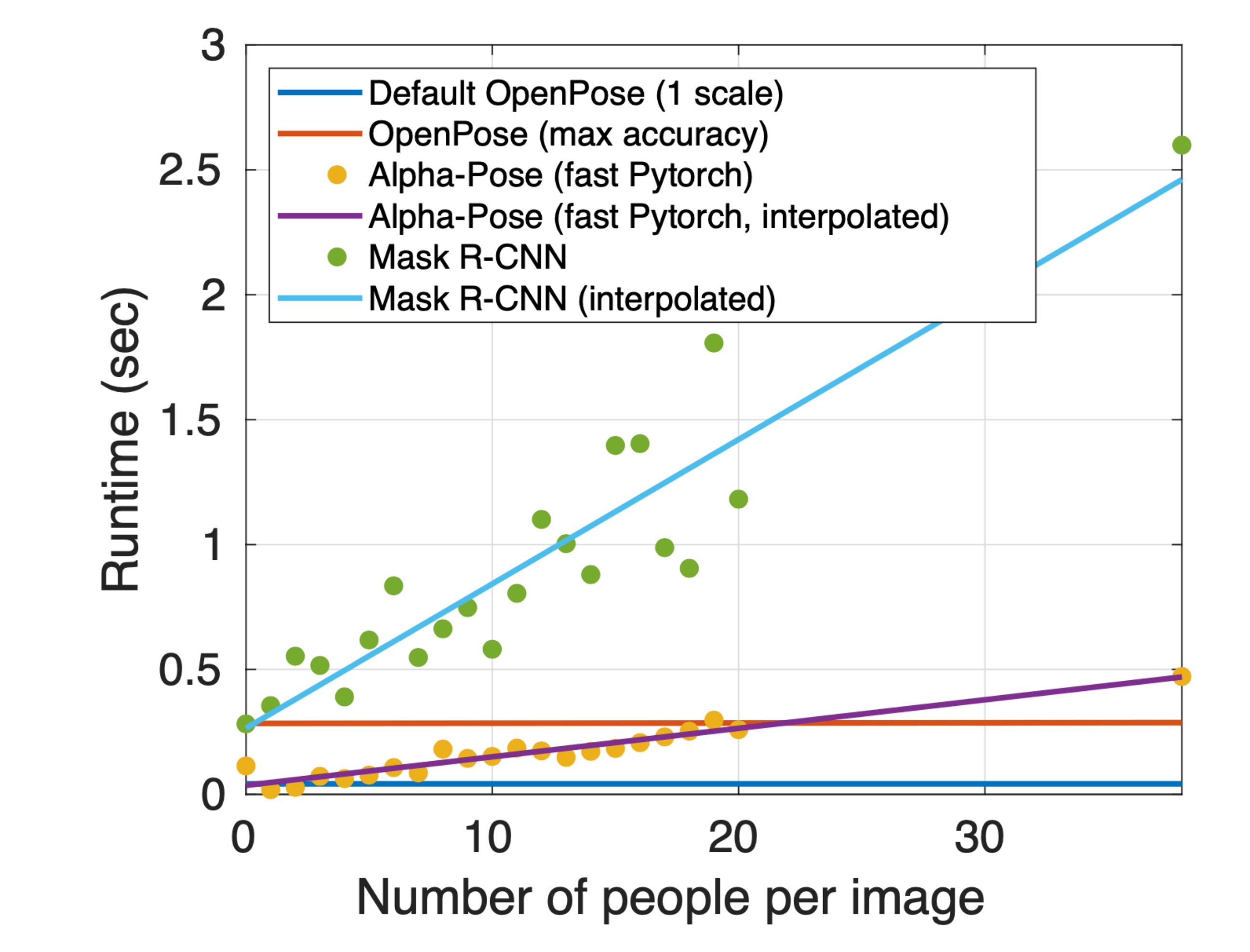
Why not always take top-down approach?

- Crowded groups bring problems for human candidate detector

  Problems in this stage can’t be solved later on

- running time tends to grow with the number of people

Results & Discussion



Inference time comparison between HPE libraries
- Chao et al. (2018)

OpenPose

- no correlation between number of people and runtime

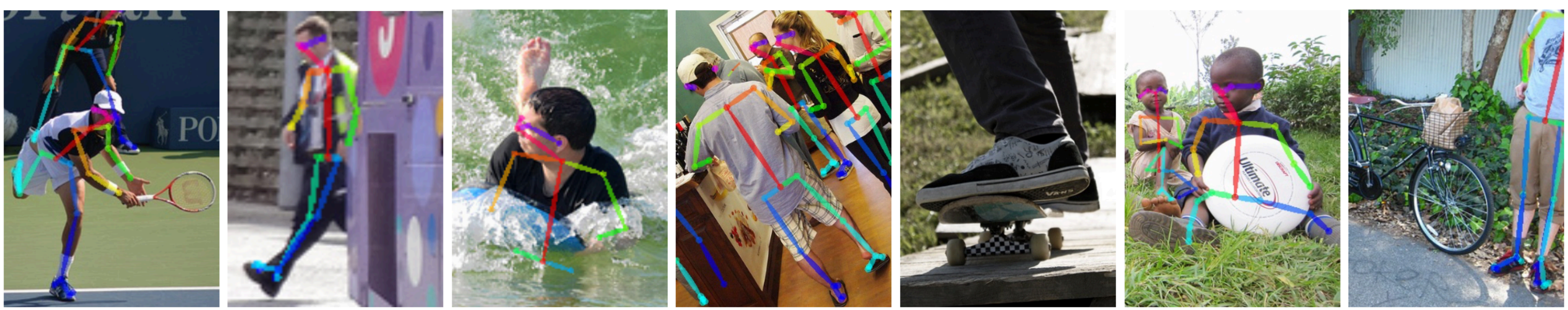
Other (Alpha-Pose, Mask R-CNN)

- correlation between number of people and runtime

Common Failure Cases

(a) (b) (c) (d) (e) (f)

Fig. 15: Common failure cases: (a) rare pose or appearance, (b) missing or false parts detection, (c) overlapping parts, i.e., part detections shared by two persons, (d) wrong connection associating parts from two persons, (e-f): false positives on statues or animals.



(a) (b) (c) (d) (e) (f) (g)

Fig. 16: Common foot failure cases: (a) foot or leg occluded by the body, (b) foot or leg occluded by another object, (c) foot visible but leg occluded, (d) shoe and foot not aligned, (e): false negatives when foot visible but rest of the body occluded, (f): soles of their feet are usually not detected (rare in training), (g): swap between right and left body parts.

Common failure cases - Chao et al. (2018)

Conclusion

- bottom-up or top-down?

  Depends on the use case
- **real-time** method for Multi-Person 2D Pose Estimation
- **Part Confidence Maps** to detect joints
- **Part Affinity Fields** to represent connections between joints
- **greedy** approach for matching problem

Thank you!

Real Time Human Pose Estimation on your smartphone or Laptop:

https://storage.googleapis.com/tfjs-models/demos/posenet/camera.html

References

Pavllo, Dario, et al. "3D human pose estimation in video with temporal convolutions and semi-supervised training." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019.

Chen, Ching-Hang, and Deva Ramanan. "3d human pose estimation= 2d pose estimation+ matching." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017.

Cao, Zhe, et al. "OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields." arXiv preprint arXiv:1812.08008 (2018).