3D Human Pose Estimation with GCN and TCN

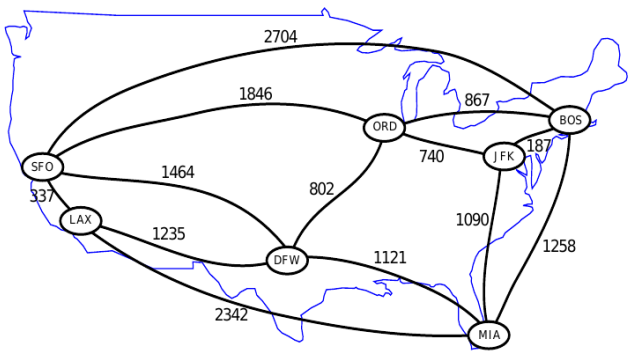
김기훈
Vision and Display System Lab.
Sogang University

Contents

- Preliminaries
  - Graph
  - Human Pose Estimation
  - Graph Convolutional Network
  - Temporal Convolutional Network
- Graph and Temporal Convolutional Networks for 3D Multi-person Pose Estimation in Monocular Videos
- Conclusion

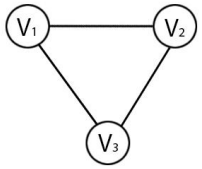
Preliminaries (Graph)

- What is a Graph?
  - $G = \{V, E\}$
  - Set of vertices and edges
  - Represents relationship between objects
- Types of Graph
  - Directed graph
  - Undirected graph
  - Weighted graph

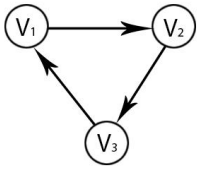


Weighted Graph

Undirected Graph

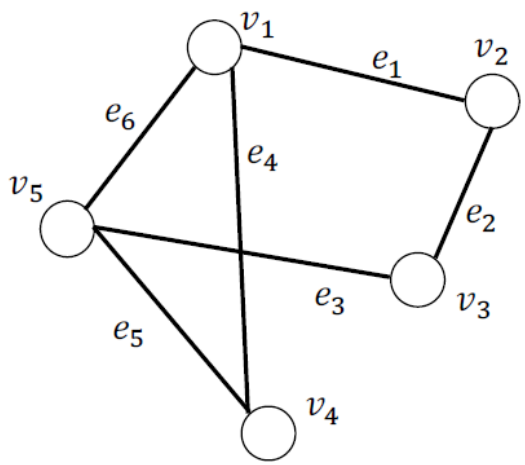


Directed Graph



Preliminaries (Graph cont.)

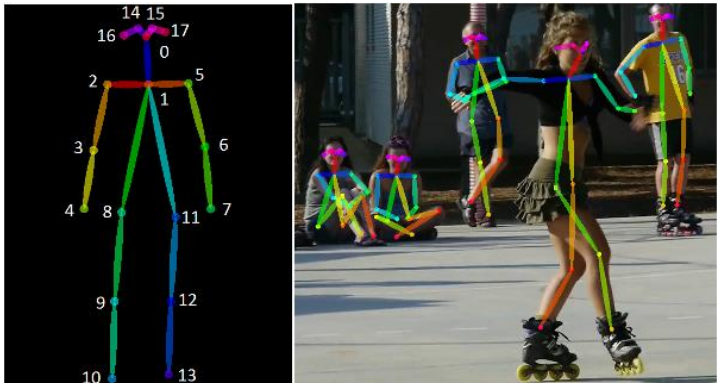
- Graph Representations
  - Adjacency matrix



$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Preliminaries (Human Pose Estimation)

- What is Human Pose Estimation?
  - "Estimating the configuration of human body parts from input data(images and videos)" [1]
  - "Aims to locate the human body parts and build human body representation" [1]

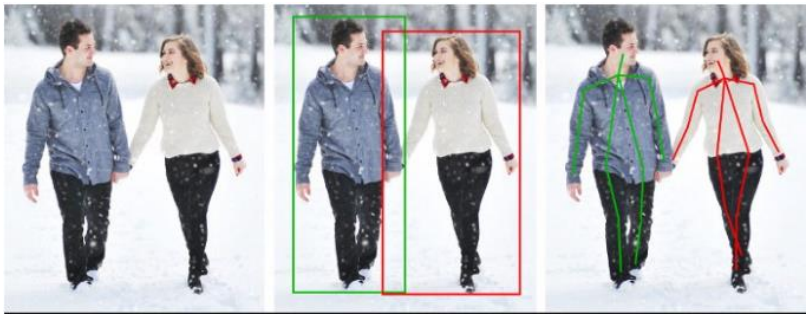


- Application
  - Activity recognition
  - Motion capture
  - AR/VR
  - Training robots

https://medium.com/beyondminds/an-overview-of-human-pose-estimation-with-deep-learning-d49eb656739b

Preliminaries (Human Pose Estimation cont.)

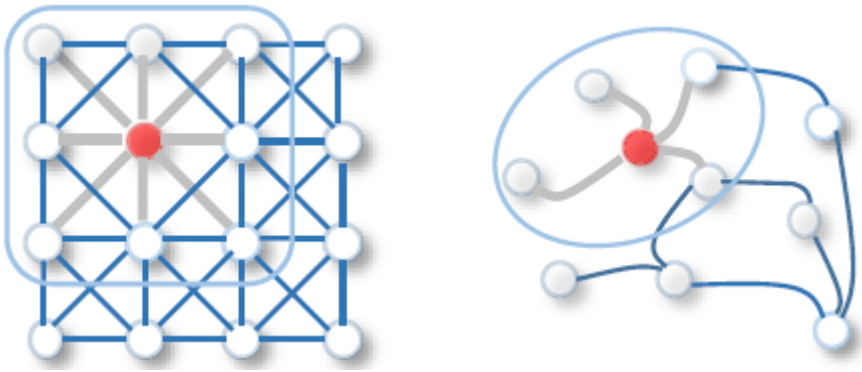
- Taxonomy
  - **Top-down Approach**
    - Detect human first



  - Bottom-up Approach
    - Detect joints first

Preliminaries (Graph Convolution)

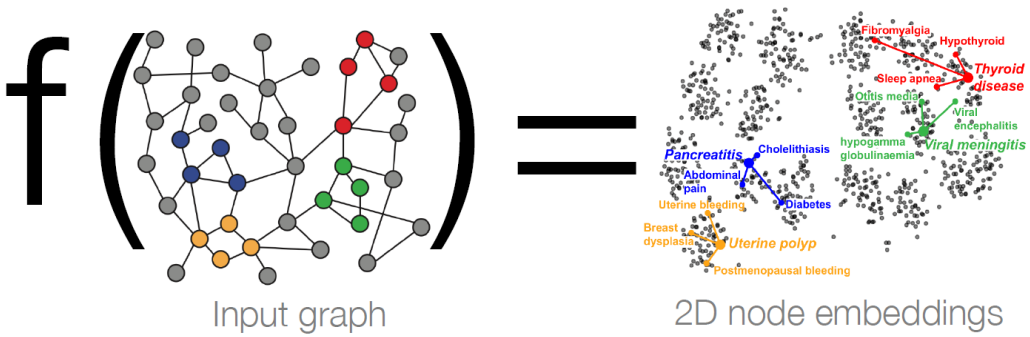
- Convolution on graphs
  - Grid data → Generalized data



https://towardsdatascience.com/understanding-graph-convolutional-networks-for-node-classiication-a2bffdb7aba7b

Preliminaries (Graph Convolution)

- How to learn mapping function f?



- Neural networks

$$H^{[i+1]} = \sigma(W^{[i]} H^{[i]} + b^{[i]})$$

feature representation at layer i+1

activation function

weights at layer i

feature representation at layer i

bias at layer i

Graph neural network

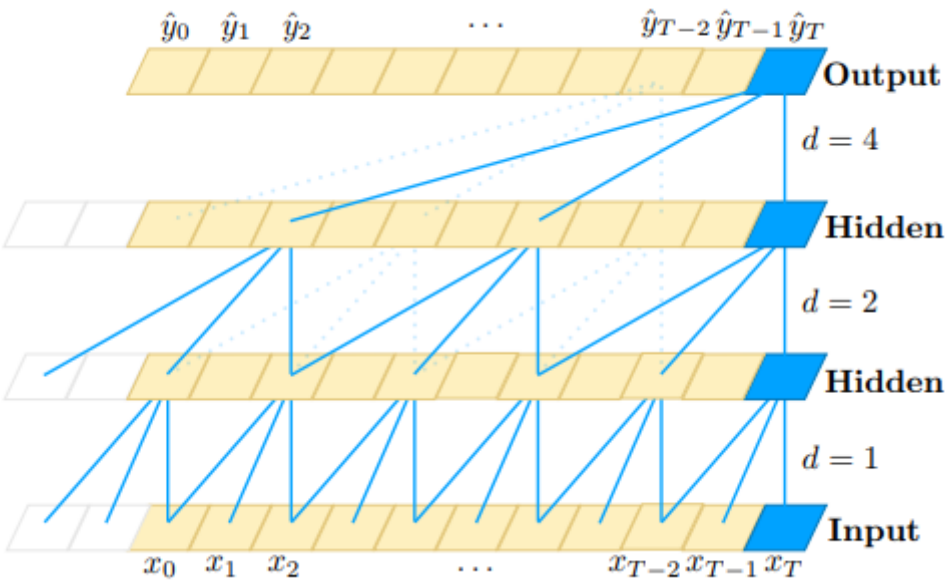
$$H^{[i+1]} = \sigma(W^{[i]} H^{[i]} A^{*})$$

https://towardsdatascience.com/understanding-graph-convolutional-networks-for-node-classiication-a2bffdb7aba7b

Images from CS224W

Preliminaries (Temporal Convolution)

- Characteristics
  - Take sequence of any length and map it to output sequence of same length
    - Achieved by zero-padding
  - Convolution architecture are causal (No information leakage)



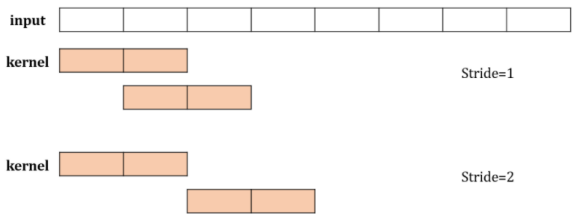
Preliminaries (Temporal Convolution)

- 1D convolution(PyTorch)

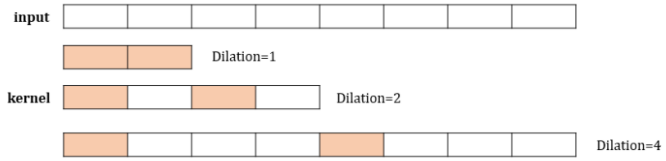
```
import torch.nn as nn

conv1 = nn.Conv1d(in_channels, out_channels, kernel_size, stride, padding, dilation, groups, bias, padding_mode)
```

Parameters

- **in_channels** (*int*) – Number of channels in the input image
- **out_channels** (*int*) – Number of channels produced by the convolution
- **kernel_size** (*int or tuple*) – Size of the convolving kernel
- **stride** (*int or tuple, optional*) – Stride of the convolution. Default: 1
- **padding** (*int, tuple or str, optional*) – Padding added to both sides of the input. Default: 0
- **padding_mode** (*string, optional*) – `'zeros'`, `'reflect'`, `'replicate'` or `'circular'`. Default: `'zeros'`
- **dilation** (*int or tuple, optional*) – Spacing between kernel elements. Default: 1
- **groups** (*int, optional*) – Number of blocked connections from input channels to output channels. Default: 1
- **bias** (*bool, optional*) – If `True`, adds a learnable bias to the output. Default: `True`



Stride



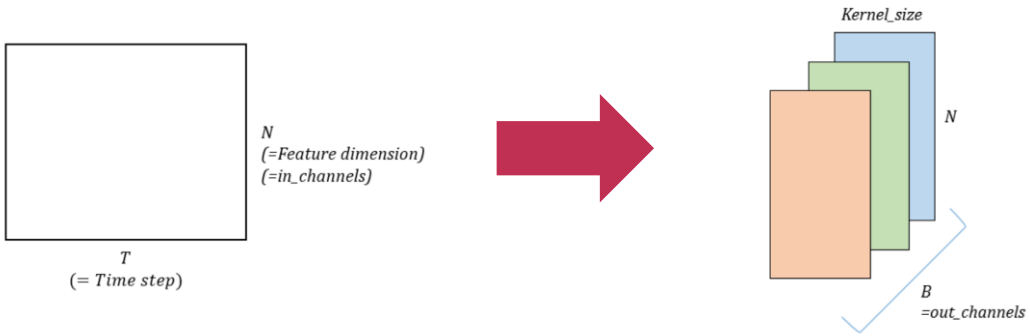
Dilation

Preliminaries (Temporal Convolution)

- 1D convolution(PyTorch)

```
import torch.nn as nn

conv1 = nn.Conv1d(in_channels, out_channels, kernel_size, stride, padding, dilation, groups, bias, padding_mode)
```

- Operation



Paper

- Title : “Graph and Temporal Convolutional Networks for 3D Multi-person Pose Estimation in Monocular Videos” Cheng, et al. (AAAI 2021) [2]
- One of the series of papers written by same authors and group
- Motivation
  - Missing Information
    - Occlusion
    - Out-of-frame target person
  - Contribution
    - GCN
      - Human joint GCN (Directed graph)
      - Bone connection (Human bone GCN)
    - TCN
      - Joint-TCN
      - Velocity-TCN
      - Root-TCN

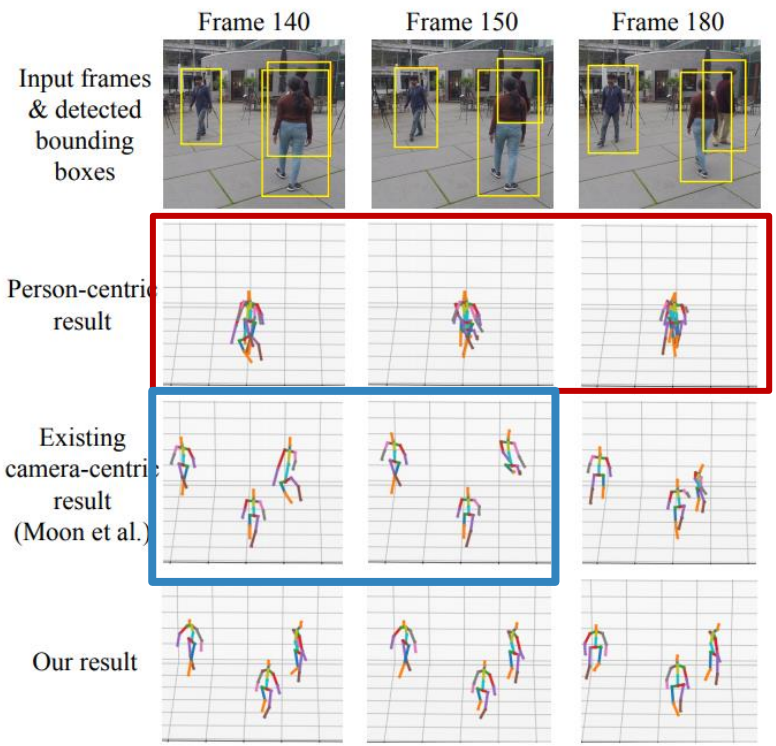
GnTCN

arXiv 2012.11806v3 | Ranked #2 3D Multi-Person Pose Estimation (absolute) on MuPoTS-3D
Ranked #2 3D Multi-Person Pose Estimation (root-relative) on MuPoTS-3D
State of the Art Root Joint Localization on Human3.6M

Introduction

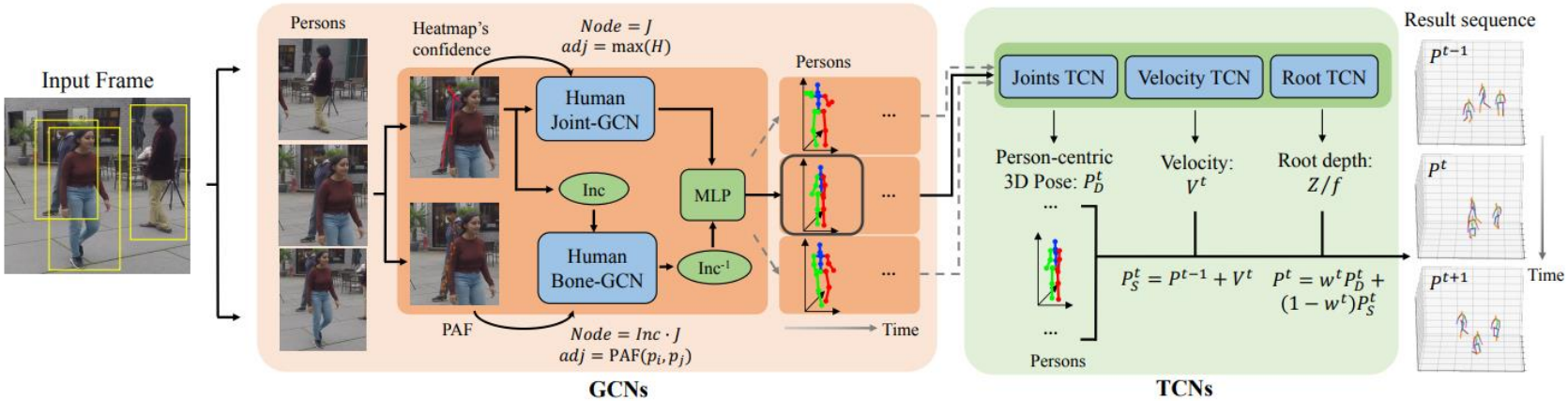
- Top-down approach
  - Problem statement: “Person centric coordinate”

- Multi-person
  - Missing Information
    - Occlusion
    - Out-of-frame
    - Inaccurate person detection



Method

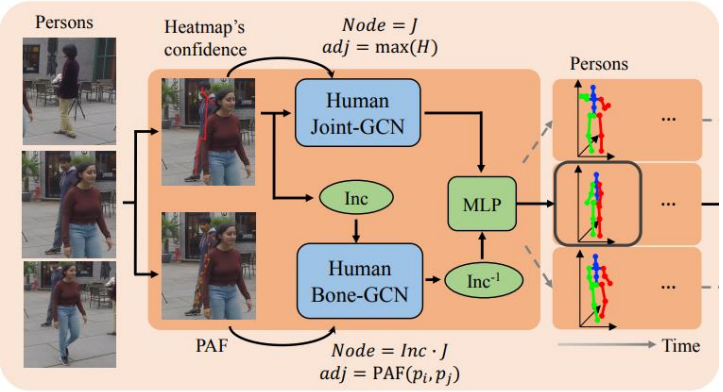
- Overview



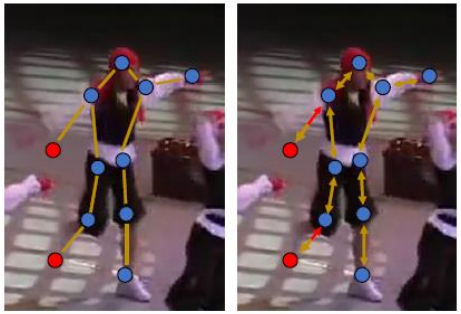
- Input: 2D pose from 2D pose estimator (HR-Net[3])
- Output: 3D pose estimation (camera-coordinate)

GCN

- Common
  - Directed graph
    - Higher confidence propagate more information
  - Frame-by-frame basis
  - Vertices: 2D joints
- Joint-GCN
  - Edge weight: Confidence scores of joints
- Bone GCN
  - Edge weight: Confidence scores of PAF(Part Affinity Field)



GCNs



GCN(cont.)

- Adjacent matrix
  - H: Heatmap
  - order(i, j): distance(# of hops) from i to j
    → Impose more weights to close vertices

$$\mathbb{A}_{i,j} = \begin{cases} max(H_i)e^{-order(i,j)}(i \neq j) \\ max(H_i)(i = j) \end{cases},$$

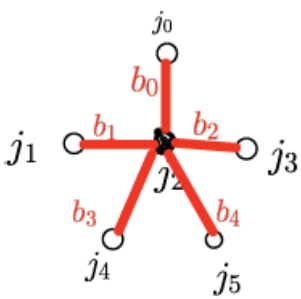
Adjacency Matrix

- Propagation
  - F: feature transformation function
  - W: learnable parameter
  - h_i: ith layer

$$h_i = \sigma(F(h_{i-1})W_i)$$

GCN (Bone-GCN)

- Exploit bone information
  - Bone information only considers human joints
    - Ex



Structure 1

Joints

Bones

$$\begin{bmatrix} 1 & & -1 & & & \\ & 1 & -1 & & & \\ & & -1 & 1 & & \\ & & -1 & & 1 & \\ & & -1 & & & 1 \end{bmatrix}$$

$\mathbb{I}_n$: Incident matrix

(x, y)

Joints

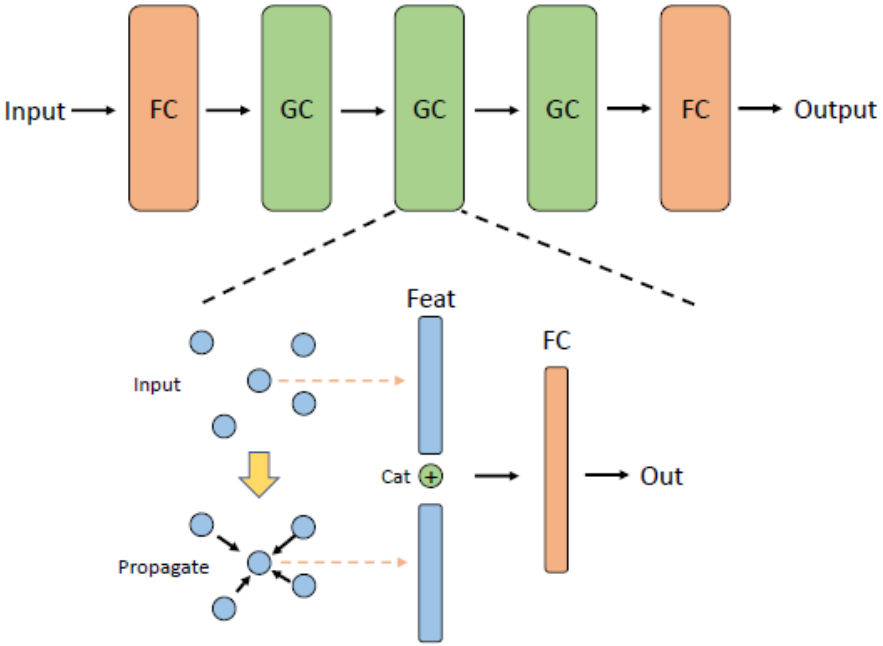
$$\begin{bmatrix} 4 & 6 \\ 4 & 3 \\ \dots & \dots \end{bmatrix}$$

$\mathbb{J}$: *Joint matrix*

- PAF(Part Affinity Field) [4]
  - 2D vector for each limb

GCN (Implementation)

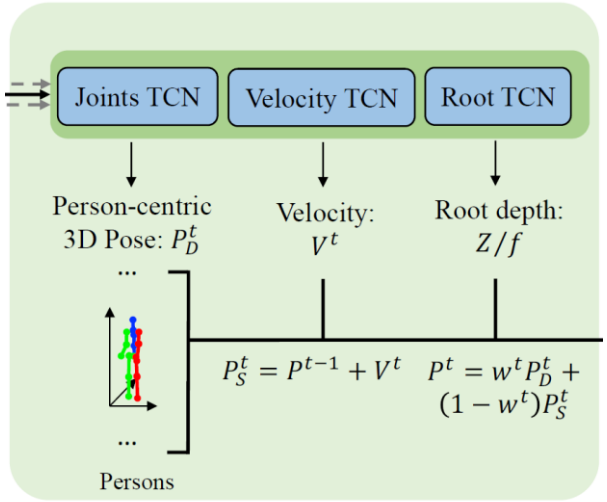
- 2 Branches
  - Joint-GCN
  - Bone-GCN
  - Directed Graph
- Structure
  - # output channels = 512
  - 3 GC(Graph Convolution) layers in each branch



TCN (Temporal Convolutional Network)

- TCNs
  - Joint TCN
  - Velocity TCN
  - Root TCN

  - Same architecture

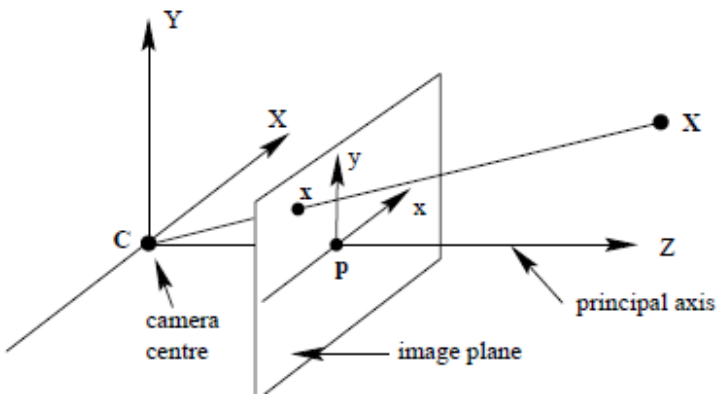


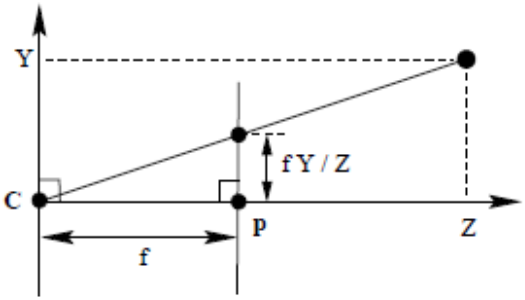
TCNs

Root-TCN

- Weak Perspective

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 1/Z \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \qquad X = \frac{Z}{f}(x - c_x) \quad Y = \frac{Z}{f}(y - c_y)$$





Root-TCN

- Estimates Z/f
  - Regression problem → Classification problem
    - Divide depth into N discrete ranges
    - Root-TCN outputs vector with probabilities $\{x_1, \dots, x_N\}$

$$[\frac{Z}{f}]^t = \textit{Soft-argmax}\,(\,f_R(\,p^{t-n:t+n}, c^{t-n:t+n}, s^{t-n:t+n}))$$

p: normalized pose
c: person's center
s: scale factor

Joint-TCN

- Input
  - Sequence of consecutive 3D pose
  - Utilizes temporal information and interpolate
- Output
  - Refined 3D pose (P_D)

Velocity-TCN

- Input
  - 3D joints and their velocities
- Output
  - Velocity of all joints

$$V^t = (v_x^t, v_y^t, v_z^t) = \mathrm{TCN}_v(p^{t-n:t-1}, V^{t-n:t-1})$$

Velocity-TCN

$$P_S^t = P^{t-1} + V^t$$

Estimated pose using velocity

Joint-TCN & Velocity-TCN

- Joint-TCN
  - Interpolate past and future
  - Connection between past and future frames
- Velocity-TCN
  - Predict from the past
  - Handles motion drift

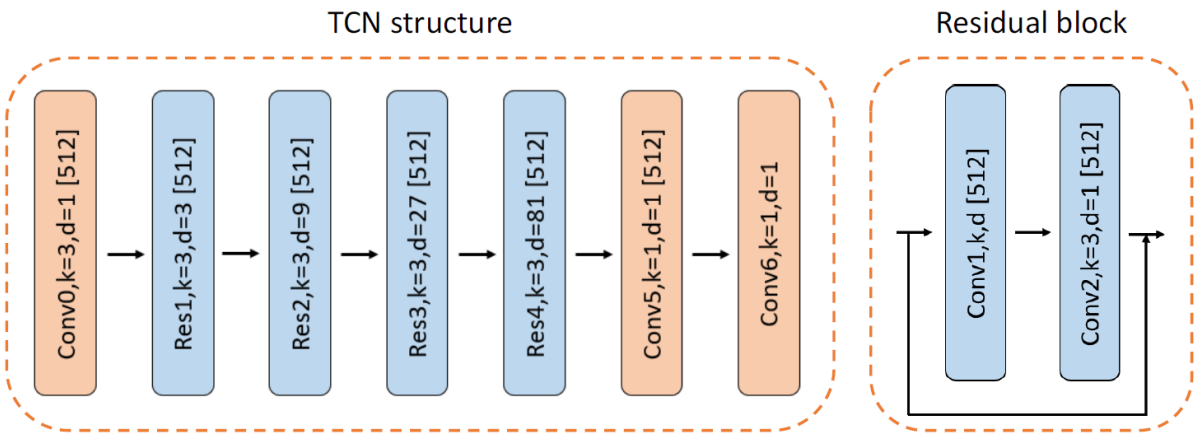
- Weighted Average
  - Occlusion
    - Joint: Heatmaps with less than 0.5 confidence
    - Pose: Less than 30% non-occluded joints
  - Weighting factor
    - Close to occlusion boundary → P_D^t
    - Far from occlusion boundary → P_S^t

$$P^t = w^t P_D^t + (1 - w^t) P_S^t$$

$$w_v^t = e^{-min(t - T_{occ}^{start}, T_{occ}^{end} - t)}$$

TCN (Implementation)

- Structure
  - 4 residual blocks (dilation rate of 3,9,17,81)
    - Consist of 2 convolutional layers
  - Temporal window length = 243

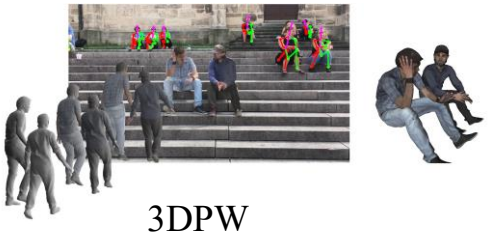


Experiments

- Datasets
  - Human3.6M
    - 3.6 million 3D “single” person
    - Captured by MoCap system
  - MuPoTS-3D
    - 3D multi-person testing set
    - Captured by multi-view markerless capture
  - 3DPW(3D Poses in the Wild)
    - Outdoor multi-persion dataset
    - Used for testing

Human3.6M

MuPoTS-3D



3DPW

Experiments

- Metric
  - Person-centric
    - MPJPE(↓) (Mean Per Joint Position Error)
    - PA-MPJPE (↓)
    - PCK (Percentage of Keypoints)
    - AUC_{rel}
  - Camera-centric
    - AP_{25}^{root}
      - Average precision of 3D human root location (correct with in 25cm)
    - PCK_{abs}
      - 3DPCK without root alignment

Experiments

- Ablation Study
  - GCN and TCN matters!
    - Especially TCN
  - Baseline
    - Joint-TCN: time window 1
    - Root-TCN: time window 1

Method	AP_{25}^{root}	AUC_{rel}	PCK	PCK_{abs}
Baseline	24.1	32.9	74.4	29.8
Baseline (GT box)	28.5	34.2	78.9	31.2
Baseline + GCNs	35.4	39.7	83.2	35.1
Baseline + TCNs	38.4	43.1	85.3	38.7
Full model	**45.2**	**48.9**	**87.5**	**45.7**

Table 1: Ablation study on MuPoTS-3D dataset. Best in bold, second best underlined.

Method	AP_{25}^{root}	AUC_{rel}	PCK	PCK_{abs}
Joint* GCN	24.1	27.3	73.1	25.6
Joint GCN	28.5	30.1	76.8	29.0
Joint + Bone* GCN	28.4	31.9	78.1	29.7
Joint + Bone GCN	33.4	37.9	82.6	34.3
Joint + Bone + Aug.	35.4	39.7	83.2	35.1
Joint TCN	43.1	45.8	86.2	42.6
Joint + Velocity	**45.2**	**48.9**	**87.5**	**45.7**

Table 2: Ablation study on our proposed Joint and Bone GCNs and TCNs. * stands for the GCN structure with undirected graph. We keep the GCN as the best one (joint + bone + aug.) to perform an ablation study on TCN.

Experiments

- Multi-person 3D dataset
  - Trained only on Human3.6M

Group	Method	PCK	PCK_{abs}
Person-centric	Mehta et al. (2018)	65.0	n/a
	Rogez et al., (2019)	70.6	n/a
	Cheng et al. (2019)	74.6	n/a
	Cheng et al. (2020)	80.5	n/a
Camera-centric	Moon et al. (2019)	82.5	31.8
	Lin et al. (2020)	83.7	35.2
	Zhen et al. (2020)	80.5	38.7
	Li et al. (2020)	82.0	43.8
	Our method	**87.5**	**45.7**

Table 3: Quantitative evaluation on multi-person 3D dataset, MuPoTS-3D. Best in bold, second best underlined.

- Single-person 3D dataset
  - Comparable with SOTA models

Group	Method	MPJPE	PA-MPJPE
Person-centric	Hossain et al., (2018)	51.9	42.0
	Wandt et al., (2019)*	50.9	38.2
	Pavllo et al., (2019)	46.8	36.5
	Cheng et al., (2019)	42.9	32.8
	Kocabas et al., (2020)	65.6	41.4
	Kolotouros et al. (2019)	41.1	n/a
Camera-centric	Moon et al., (2019)	54.4	35.2
	Zhen et al., (2020)	54.1	n/a
	Li et al., (2020)	48.6	30.5
	Ours	**40.9**	**30.4**

Table 5: Quantitative evaluation on Human3.6M for normalized and camera-centric 3D human pose estimation. * denotes ground-truth 2D labels are used. Best in bold, second best underlined.

Experiments

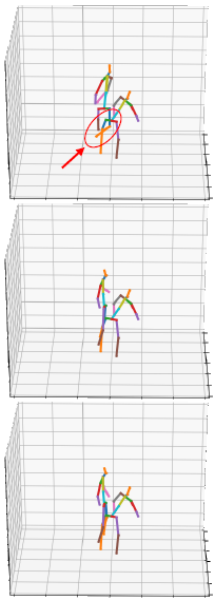
- 3DPW dataset
  - Ground-truth: 3D mesh models
  - Subset
    - To verify robustness of the model

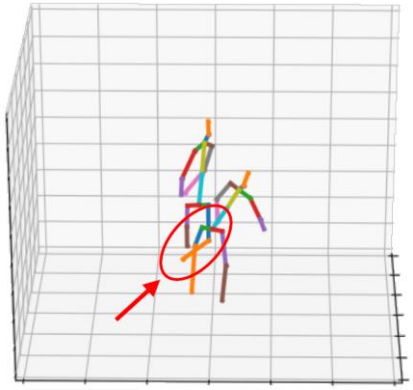
Dataset	Method	PA-MPJPE	δ
Original	Dabral et al. (2018)	92.2	n/a
	Doersch et al. (2019)	74.7	n/a
	Kanazawa et al. (2019)	72.6	n/a
	Cheng et al. (2020)	71.8	n/a
	Sun et al. (2019b)	69.5	n/a
	Kolotouros et al. (2019)*	59.2	n/a
	Kocabas et al., (2020)*	**51.9**	n/a
	Our method	64.2	n/a
Subset	Cheng et al. (2020)	96.1	+24.1
	Sun et al. (2019b)	94.1	+24.6
	Kolotouros et al. (2019)*	88.9	+29.7
	Kocabas et al., (2020)*	**82.5**	+30.6
	Our method	85.7	**+21.5**

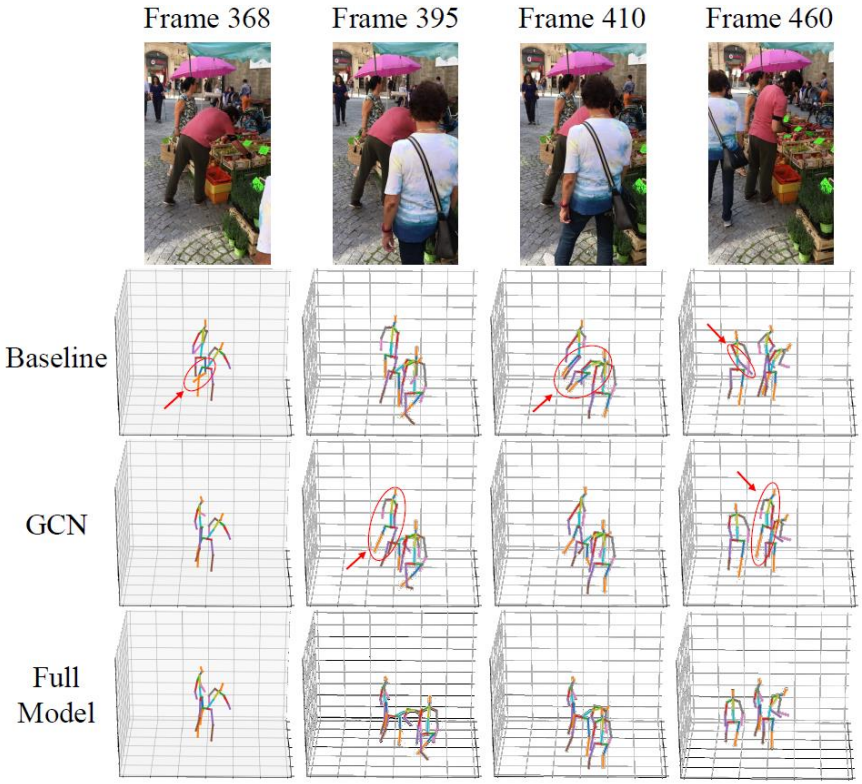
Table 4: Quantitative evaluation using PA-MPJPE in millimeter on original 3DPW test set and its occlusion subset. * denotes extra 3D datasets were used in training. Best in bold, second best underlined.

Experiments

- Qualitative Results
  - Baseline from ablation study

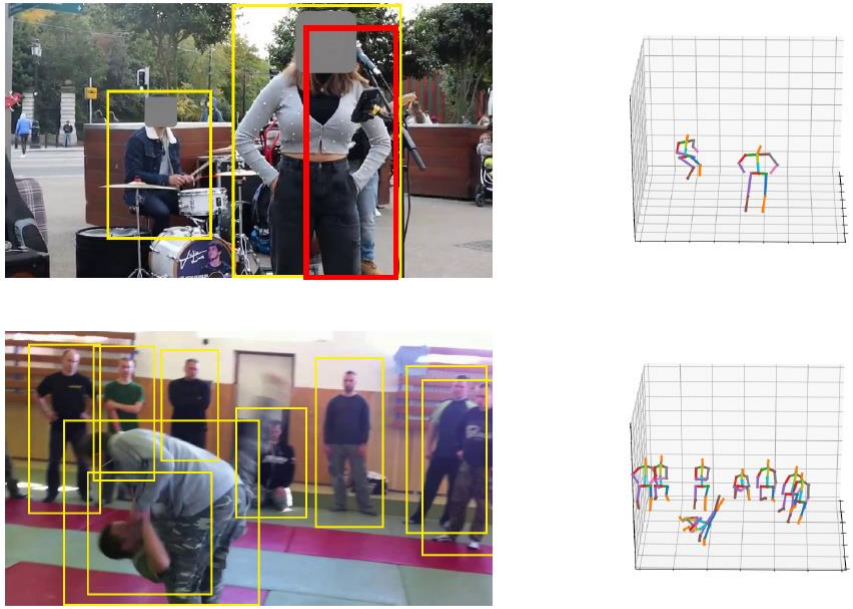






Experiments

- Failure cases
  - False human detection
    - Missing, duplicate
    - Problem of top-down approach
  - Rare poses
    - Trained only on Human3.6M



References

[1] Ce Zheng, Wenhan Wu, Taojiannan Yang, Sijie Zhu, Chen Chen, Ruixu Liu, Ju Shen, Nasser Kehtarnavaz, & Mubarak Shah. (2021). Deep Learning-Based Human Pose Estimation: A Survey.

[2] Yu Cheng, Bo Wang, Bo Yang, & Robby T. Tan. (2021). Graph and Temporal Convolutional Networks for 3D Multi-person Pose Estimation in Monocular Videos.

[3] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, & Bin Xiao. (2020). Deep High-Resolution Representation Learning for Visual Recognition.

[4] Zhe Cao, Tomas Simon, Shih-En Wei, & Yaser Sheikh. (2017). Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields.

+ Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. (2021). A Comprehensive Survey on Graph Neural Networks. IEEE Transactions on Neural Networks and Learning Systems, 32(1), 4–24.

Thank you!