Supplementary Material for Multi-person Articulated Tracking with Spatial and Temporal Embeddings

Sheng Jin[1,2] Wentao Liu[1] Wanli Ouyang[3,4] Chen Qian [1]
[1] SenseTime Research [2] Tsinghua University [3] The University of Sydney
[4] SenseTime Computer Vision Research Group, Australia
[1]{jinsheng, qianchen}@sensetime.com, liuwtwinter@gmail.com [3] wanli.ouyang@sydney.edu.au

1. Model Architecture

1.1. SpatialNet

The model architecture of SpatialNet resembles [9], which consists of a four-stage stacked hourglass module [10] with an input resolution of 512 × 512 and an output size of 128 × 128. The model architecture of SpatialNet is illustrated in Figure 1. SpatialNet processes a single frame at a time. Given an image, it first extracts 256-dimensional low-level feature maps by a few convolution layers. The extracted feature maps are then processed by 4-stage stacked hourglass to obtain higher level semantic features. Finally, SpatialNet produces auxiliary ordinal maps, body heatmaps, Keypoint Embedding (KE), and Spatial Vector Fields (SVF) simultaneously.

1.2. TemporalNet HE branch

TemporalNet consists of HE branch and TIE branch, it shares low-level feature extraction layers with SpatialNet. As illustrated in Figure 2, our **HE branch** is based on [11], which exploits both high-level and mid-level features. Given low-level features from SpatialNet as the input, HE branch first extracts region-based features by RoI-Align [2]. A few resnet blocks [3] are then used to obtain higher level semantic features. We use Global Average Pooling (GAP) [7] to fuse the features. We follow [11] to combine both high-level semantic information and discriminative mid-level semantic features. We apply GAP on res5a and res5b and concatenate the pooled feature vectors. We obtain the 1024D mid-level feature by a fully connected layer (fc). Finally, both high-level and mid-level semantic features are concatenated to produce the final 3072-dimensional Human Embedding (HE).

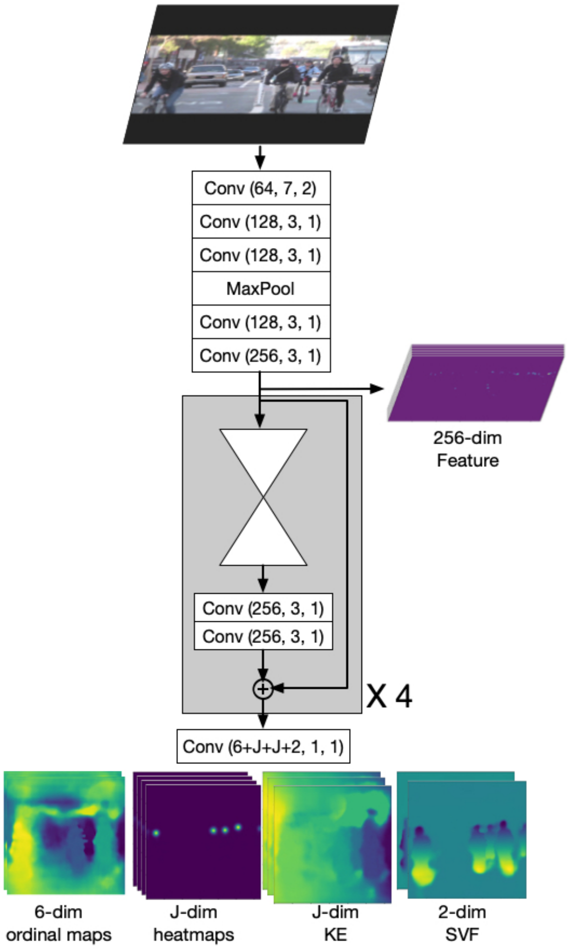


Figure 1. Model architecture of SpatialNet. Conv(c,k,s) means convolution block with the output channel of c, the kernel size of k, and the stride of s. The activation layer is ReLU.

1.3. TemporalNet TIE branch

As illustrated in Figure 3, we use one stage hourglass module as the backbone of **TIE branch**. We first concatenate low-level feature maps, body heatmaps and SVF from both $(t-1)$-th frame and t-th frame. The concatenated features are input to a 1×1 convolution layer to reduce dimensions. One stage hourglass module is used to extract higher-level features. Finally, after a few convolution lay-

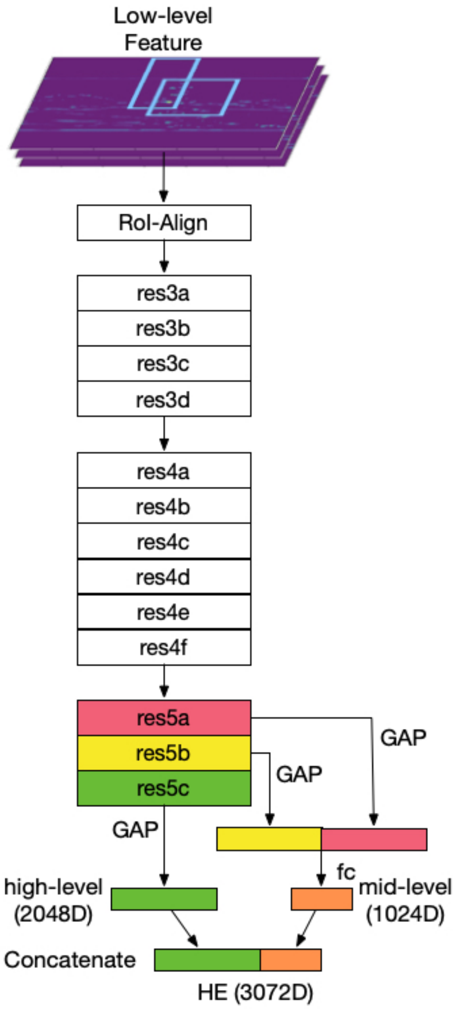


Figure 2. Architecture of TemporalNet HE branch. This branch is based on [11], which augments resnet50 [4] by fusing both high-level and mid-level features. res means the residual block in resnet50. GAP means global average pooling [7].

ers, we obtain the predicted bi-directional temporal vector fields (TVF), which are denoted as $\hat{\mathcal{T}}$ and $\hat{\mathcal{T}}'$ respectively.

2. Training Details

When evaluating on ICCV'17 PoseTrack Challenge Dataset [5], we follow the common settings [5] to train the keypoint model on MSCOCO [8] first, then finetune on the mixture of MPII Human Pose Dataset [1] and PoseTrack Challenge Dataset [5]. Training was performed on four NVIDIA GeForce GTX Titan GPUs in our experiments.

2.1. SpatialNet

In order to reduce overfitting, we follow [9] to perform data augmentation. We randomize hue ([0.8,1.2]) and saturation ([0.5,1.5]), then adjust brightness ([0.7,1.3]) and contrast ([0.5,1.5]) accordingly. We also randomly rotate images between -30 and 30 degrees, horizontally flip the images, and scale between 0.75 and 1.25.

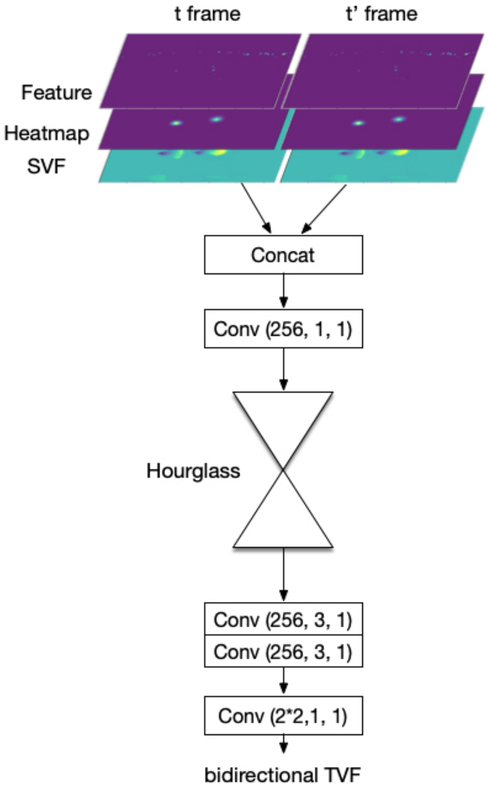


Figure 3. Architecture of TemporalNet TIE branch. Conv(c,k,s) means convolution block with the output channel of c, the kernel size of k, and the stride of s. The activation layer is ReLU.

SpatialNet simultaneously produces ordinal maps, body heatmaps, KE and SIE. Therefore, the total loss consists of L_{det}, L_{KE}, L_{aux} and L_{SIE}, with their weights 1,1e-3,1e-4 and 1e-4. We use Adam [6] to train our model and set the batch size to 16. The initial learning rate is set to 2×10^{-4} and reduced to 1×10^{-5} after 250K iterations. Then we fine-tune SpatialNet with PGG included for 40 epochs. The learning rate is set to 1×10^{-5}. The weight of PGG grouping loss L_{PGG} is simply set to 1e-4. In practice, we have found the iteration number $R = 1$ is sufficient, and more iterations do not lead to much gain.

2.2. TemporalNet

Since SpatialNet and TemporalNet share the same feature extraction layers, it is straightforward to train TemporalNet in an end-to-end manner. In practice, however, we simply fix SpatialNet and train TemporalNet for another 40 epochs with the initial learning rate of 2×10^{-4}. Note that the HE branch uses the publicly available ImageNet pre-trained resnet50 model for initialization.

To reduce overfitting, we randomly select a pair of images I^t and $I^{t'}$ from a range-5 temporal window ($\|t - t'\|_1 \leq 5$) in a video clip as input. The same augmentation procedure is applied to I^t and $I^{t'}$ simultaneously. We use Adam [6] to train our model and set the batch size to 16.

3. Qualitative Analysis

In this section, we visualize more qualitative results and also show some failure cases. We use different colors for different track ids.

Figure 4 demonstrates the qualitative evaluation results of our approach in handling challenges including close proximity, camera movement, fast motion and shot changes. Figure 4(a) shows the ability to handle crowded scenes. We observe that our method produces smooth tracing results. Figure 4(b) demonstrates that our method is able to distinguish persons in highly cluttered scenes with frequent interactions among targets. Figure 4(c)(d) shows that our model is able to deal with camera movement and fast body motion. Typical geometric-based tracking methods are prone to fail in fast camera movement and body motion, since the assumption that keypoint locations change smoothly does not hold. By exploiting appearance features of Human Embedding (HE), our method is more robust to such challenges. Moreover, as illustrated in Figure 4(e), videos sometimes contain shot changes (marked with the 'green' box). At the shot change, the smoothness between frames is not guaranteed. Note that in the green box, the relation order among targets is disrupted. However, our appearance augmented tracker correctly matches the targets.

3.1. Failure Cases

Our method is robust to crowding, body motion, and camera movement in general. However, we also observe some failure cases for pose estimation and pose tracking, as shown in Figure 6. We observe that videos may contain strong motion blur due to the fast body and/or camera motion as shown in Figure 6(a). Such problems can be mitigated by leveraging temporal information from adjacent video frames. Other challenges such as rare human poses (b) and large pose occlusion (c) may be solved by adding sufficient training data. Multi-scale spatial pyramid will cope with the challenging small-scale human poses. We will leave these modifications in future work. One failure case of our model for pose tracking is illustrated in Figure 6(e), where the man colored in green is totally occluded, and he is assigned to a new track ID (cyan) afterward. Since our tracking algorithm only exploits short-term temporal information without long-range temporal consistency, the track id will drift with severe occlusion. We will investigate long-term dependencies in future work.

References

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 2

[2] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. *arXiv preprint arXiv:1703.06870*, 2017. 1

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 1

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[5] U. Iqbal, A. Milan, M. Andriluka, E. Ensafutdinov, L. Pishchulin, J. Gall, and Schiele B. PoseTrack: A benchmark for human pose estimation and tracking. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[6] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2, 3

[7] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013. 1, 2

[8] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, 2014. 2

[9] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *Advances in Neural Information Processing Systems (NIPS)*, 2017. 1, 2

[10] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision (ECCV)*, 2016. 1

[11] Qian Yu, Xiaobin Chang, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. The devil is in the middle: Exploiting mid-level representations for cross-domain instance matching. *arXiv preprint arXiv:1711.08106*, 2017. 1, 2

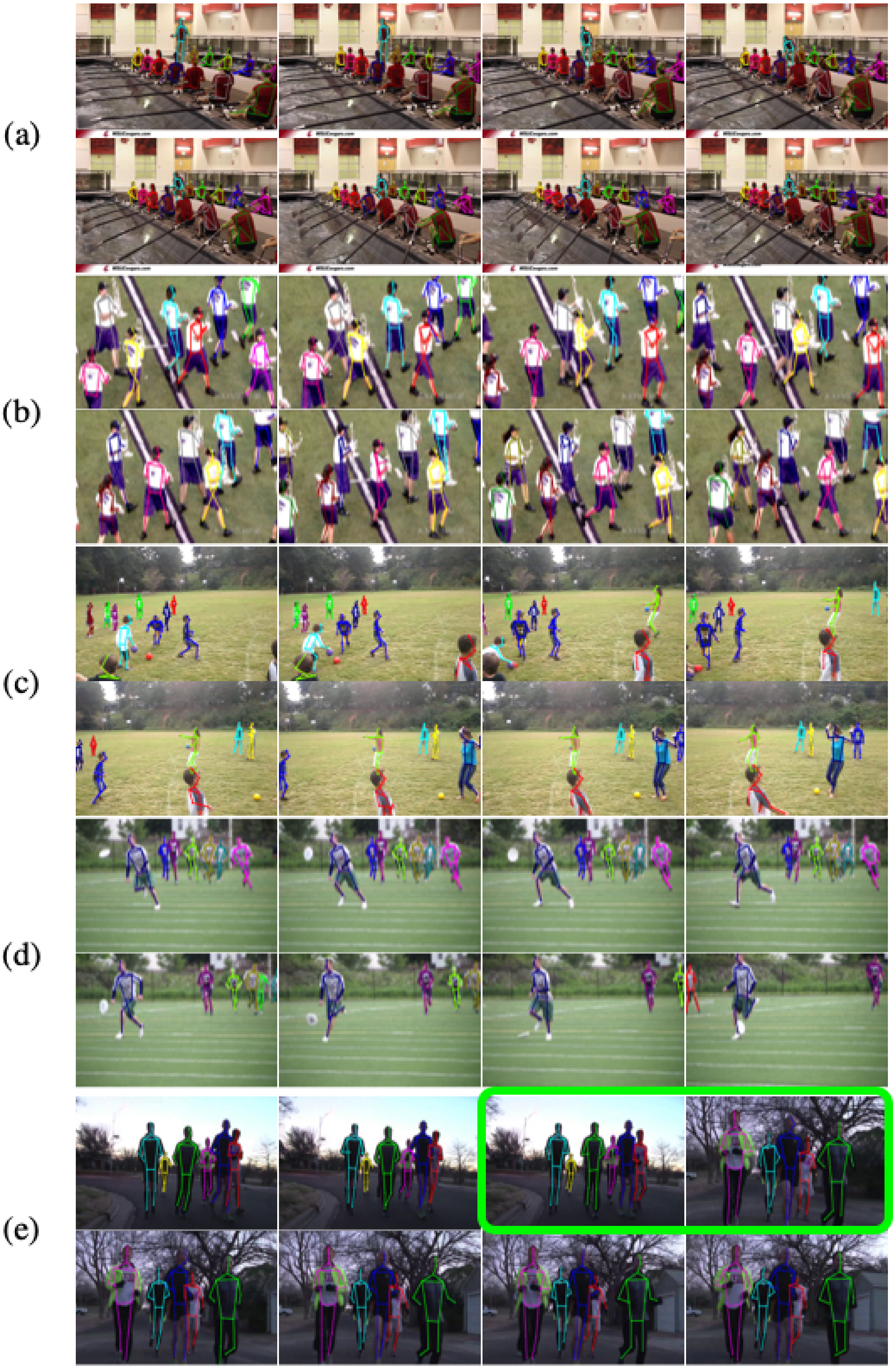


Figure 4. Qualitative evaluations for pose tracking. We show eight frames for each video sequence. Poses are color-coded by predicted track id. (a) crowding (b) multiple-person interaction (c) camera movement (d) fast body motion (e) camera shot change marked with a 'green' box. (Best viewed in color).

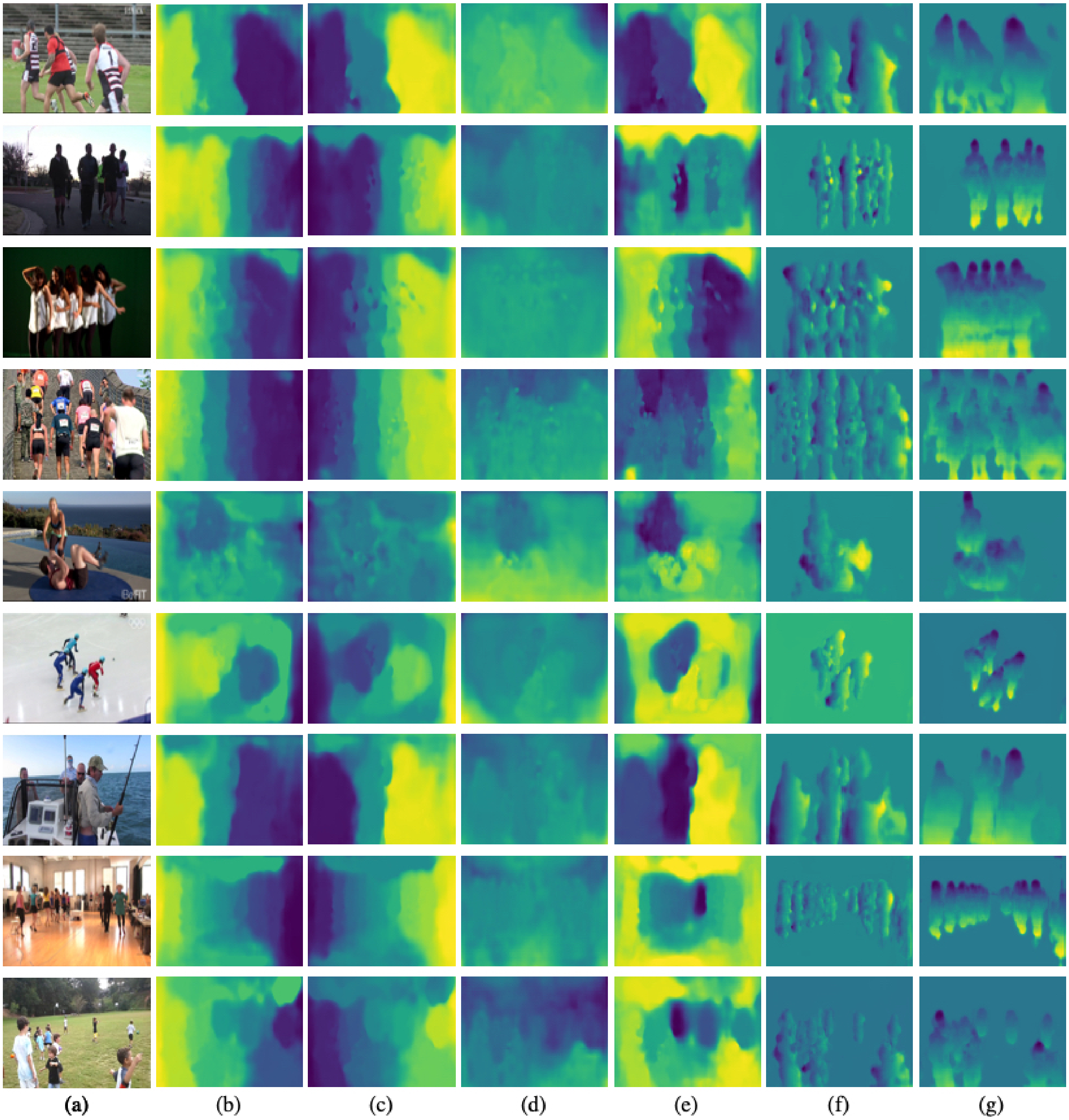
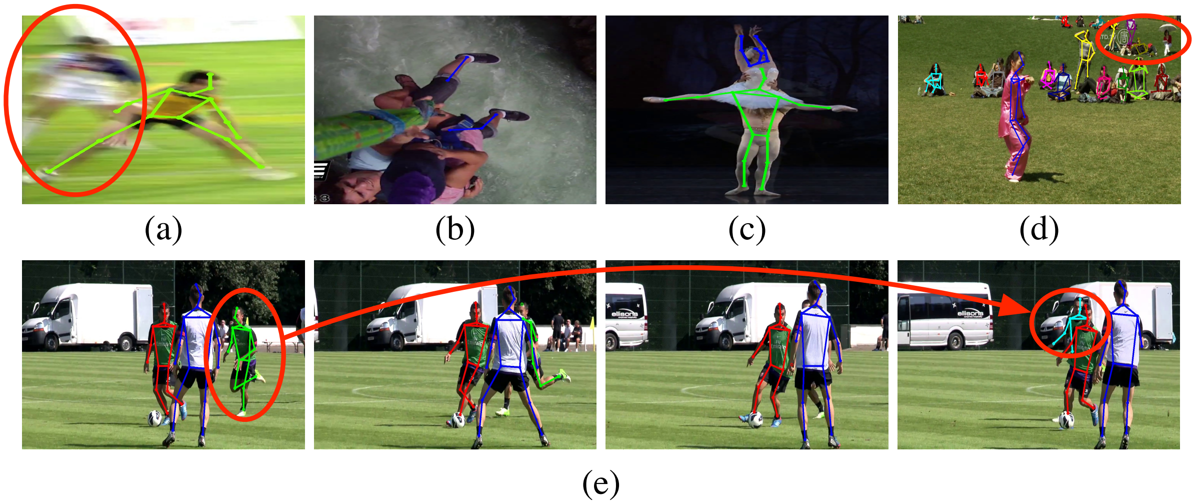


Figure 5. More visualization results of spatial embeddings. (a) Input image. (b) The average KE. (c)(d)(e) Predicted 'left-to-right', 'top-to-bottom' and 'far-to-near' geometric-relation maps. We use colors to indicate the predicted orders, where the brighter color means the higher ordinal value. (f)(g) The spatial vector fields of x-axis and y-axis respectively. The bright color means positive offset relative to the human center, while dark color means negative.



(a) (b) (c) (d)

(e)

Figure 6. Failure cases for pose estimation and pose tracking. (a) motion blur (b) rare poses (c) extreme occlusion (d) small scale poses (e) track id drifting (green to cyan) for lack of long-range temporal consistency.