An Efficient PointLSTM for Point Clouds Based Gesture Recognition Supplementary Material

Yuecong Min[1,2], Yanxiao Zhang[1,2], Xiujuan Chai[3], Xilin Chen[1,2]
[1]Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS),
Institute of Computing Technology, CAS, Beijing, 100190, China
[2]University of Chinese Academy of Sciences, Beijing, 100049, China
[3]Agricultural Information Institute, Chinese Academy of Agricultural Sciences, Beijing, 100081, China
{yuecong.min,yanxiao.zhang}@vipl.ict.ac.cn, chaixiujuan@caas.cn, xlchen@ict.ac.cn

A. Overview

In this supplementary material, we provide details that are not shown in the main paper. We first present the details of the proposed network architectures in Sec. B (main paper Sec. 3.3) and pre-processing details in Sec. C (Sec. 4.2). Then we discuss the inference speed with different point cloud scales in Sec. D (Sec. 4.2). At last, we analyze some failure cases and show the confusion matrices of baseline and proposed methods in Sec. E (Sec. 4.2 & 4.3).

B. Details on Baseline Architectures (Sec. 3.4)

This model is a modified version of FlickerNet [2]. To evaluate the effects of the PointLSTM, we divided the network into five stages: the first stage extracts intra-frame features using spatial grouping, and the second to fourth stages extract inter-frame features by spatio-temporal grouping. The fifth stage is to gather information from all timesteps. Table 1 shows the entire network structure.

At each inter-frame stage, point-wise shared fully connected, batch normalization, and ReLU layers are adopted to update each point's feature vector and get position embedding. Then the two branch features are added together and propagated to the next stage with the original coordinates.

Table 1. More details about the baseline architecture. The first stage extracts intra-frame features using general grouping, and the second to fourth stages are inter-frame features. Point-wise shared Conv1x1 (1×1 Convolution-BN-ReLU) is used at each stage.

	Layer	Output Size
Stage 1	Grouping	(32, 4, 128, 16)
	Conv1x1	(32, 32, 128, 16)
	Conv1x1	(32, 64, 128, 16)
	Max-pooling	(32, 64, 128, 1)
Stage 2	STGrouping	(32, 132, 128, 24)
	STSampling	(32, 132, 64, 24)
	Position Embedding	(32, 128, 64, 24)
	Conv1x1	(32, 128, 64, 24)
	Add	(32, 128, 64, 24)
	Max-pooling	(32, 128, 64, 1)
Stage 3	STGrouping	(32, 260, 64, 48)
	STSampling	(32, 260, 32, 48)
	Position Embedding	(32, 256, 32, 48)
	Conv1x1	(32, 256, 32, 48)
	Add	(32, 256, 32, 48)
	Max-pooling	(32, 256, 32, 1)
Stage 4	STGrouping	(32, 516, 32, 12)
	STSampling	(32, 516, 8, 12)
	Position Embedding	(32, 512, 8, 12)
	Conv1x1	(32, 512, 8, 12)
	Add	(32, 512, 8, 12)
	Max-pooling	(32, 512, 8, 1)
Stage 5	Conv1x1	(32, 1024, 8, 1)
	Max-pooling	1024
Classifier	FC	classes

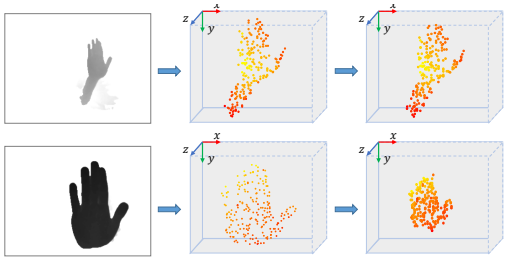
C. Details on Data Pre-Processing (Sec. 4.1)

The original NVGesture, SHREC'17, and MSR Action 3D datasets do not provide point cloud sequences. However, most gesture data are collected by short-range depth cameras. The sizes of hand region in the video change along with the distance between hand and camera, which will make the recognition system confused. Therefore, we transform the pixels of depth videos from the image coordinate system to the world coordinate system. Let $(x_{i,2D}, y_{i,2D})$, $(x_{i,3D}, y_{i,3D}, z_{i,3D})$ be the location of point i in the image and the world coordinate system, the inverse perspec-

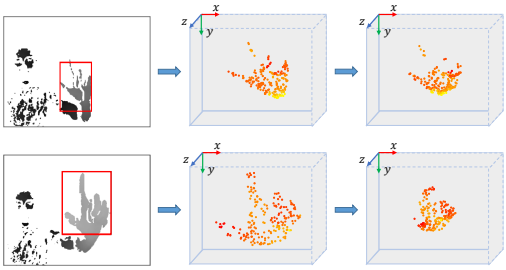
tive transformation can be represented by

$$
\begin{aligned}
x_{i,3D} &= \frac{x_{i,2D} - c_x}{f_x} I_{(x_{i,2D}, y_{i,2D})} \\
y_{i,3D} &= \frac{y_{i,2D} - c_y}{f_y} I_{(x_{i,2D}, y_{i,2D})} \\
z_{i,3D} &= I_{(x_{i,2D}, y_{i,2D})}
\end{aligned} \tag{1}
$$

where (f_x, f_y, c_x, c_y) are intrinsic parameter of a depth camera. We use the default values of each depth device: (224.502, 230.494, 160.000, 120.000) for SoftKinetic DS325 and (463.889, 463.889, 320.000, 240.000) for RealSense SR300.



(a) NVGesture



(b) SHREC'17

Figure 1. Examples of point sampling process on two Datasets (a) NVGesture, (b) SHREC'17. Each row contains an original depth map (left), a sampled point cloud (middle), and a transformed point cloud (right).

Moreover, the Z values of the background are almost zeros. In order to reduce the unnecessary computation for background points, we use the bounding boxes [1] provided in SHREC'17, and use the Otsu [3] threshold to remove the background for the other two datasets and then sample points from hand regions. Fig. 1 shows the point-sampling process.

D. Details on Inference Time (Sec. 4.2)

Sec. 4.2 has introduced the model size and inference time of different architectural designs. Here we show the inference speed of the model with different point cloud scales in Table 2. For this evaluation, we sample 32 frames and keep the same number of points for each frame by #points.

Table 2. Inference time of PointLSTM and baseline with different point cloud scales and batch sizes.

#Points	32	32	64	64	128	128	256
Batch size	1	8	1	4	1	2	1
PointLSTM (ms)	24	54	28	53	34	56	60
Baseline (ms)	10	38	13	38	22	39	41

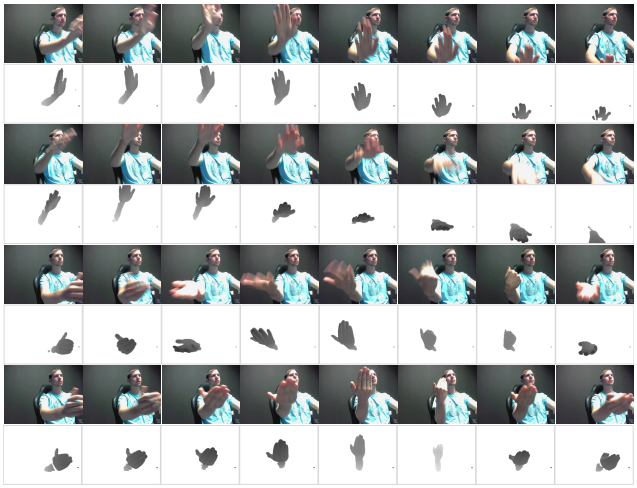


Figure 2. Examples of NVGesture Dataset. Gestures from top to down are "Move hand down", "Push hand down", "Pull hand in" and "Call someone". Each pair has similar postures and motion trajectories with small differences.

The inference time is measured on a single Tesla P100 GPU with Pytorch [4].

E. Details on Result Analysis (Sec. 4.2 & 4.3)

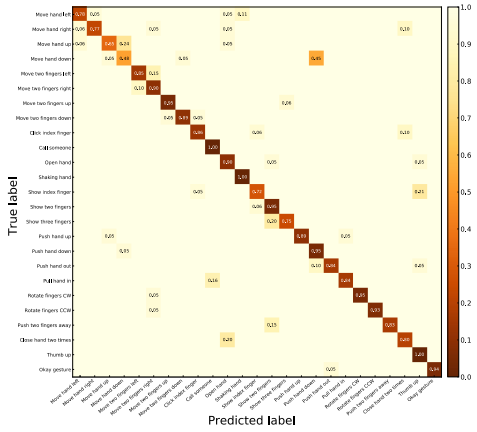
Confusion matrix comparison on NVGesture dataset for baseline and PointLSTM are shown in Fig. 4. We can find that PointLSTM can clearly distinguish gestures with similar postures yet different motion directions, such as "Move hand up" and "Move hand down". However, some gesture samples are very similar and hard to distinguish. As shown in Fig. 2, both "Move hand down" and "Push hand down" have similar postures and motion trajectories with only small hand orientation differences. Besides, many noisy postures and motions are also occur in videos: "Move hand up" gesture is shown in the preparation stage of both "Move hand down" and "Push hand down" gestures, and "Thumb up" gesture is shown in the "Pull hand in" gesture. Therefore, the quality of gesture localization is one of the essential factors to further improve the recognition results.

PointLSTM-middle with direct grouping shows better results than aligned grouping on SHREC'17. Thus we analyze failure cases and visualize several typical examples in Fig. 3. Examples show PointLSTM with aligned grouping operation tends to predict motion-relevant gestures. We believe this is mainly because such roughly alignment will

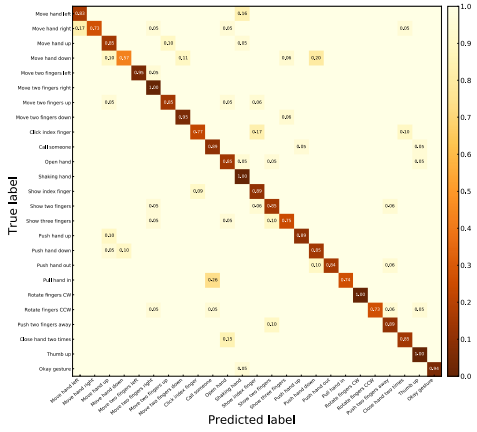
Figure 3. Several failure cases of PointLSTM-middle with aligned grouping on SHREC'17, which is more sensitive to the displacements.

enhance the motion cues and weaken the spatial correlation of hand shapes, especially when the gesture is performed with noisy motion. Inaccurate detections also lead to unstable alignment, as shown in the last row of Fig. 3. Thus, we believe a more accurate alignment method can further improve the recognition performance.

Confusion matrix comparison on SHREC'17 for baseline and PointLSTM is shown in Fig. 5. Most of the classes that required long-range dependencies achieve higher accuracy than 90%, such as "Swipe" and "Rotation" relevant classes, and show significant improvements compared with baseline. We can see from Fig. 5(a) that the baseline method is hard to distinguish between "Swipe Down" and "Tap" due to the lack of long-term information. PointLSTM (Fig. 5(b)) can better recognize these two classess, and improve the discriminative power of both short-range and long-range gestures. We can draw the same conclusion from the Confusion Matrices on MSR Action3D dataset shown in Fig. 6.



(a) Baseline



(b) PointLSTM

Figure 4. Confusion matrix comparison on NVGesture dataset.

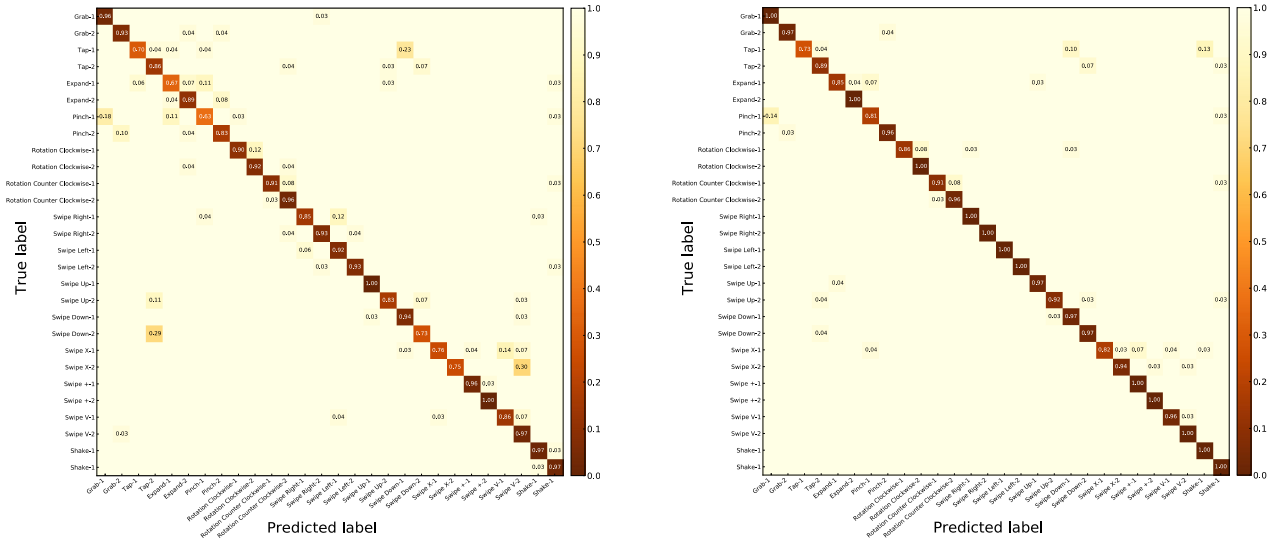
References

[1] Quentin De Smedt, Hazem Wannous, Jean-Philippe Vandeborre, Joris Guerry, Bertrand Le Saux, and David Filliat. Shrec'17 track: 3d hand gesture recognition using a depth and skeletal dataset. In *In Eurographics Workshop on 3D Object Retrieval*, 2017. 2

[2] Yuecong Min, Xiujuan Chai, Lei Zhao, and Xilin Chen. Flickernet: Adaptive 3d gesture recognition from sparse point clouds. In *British Machine Vision Conference*, 2019. 1

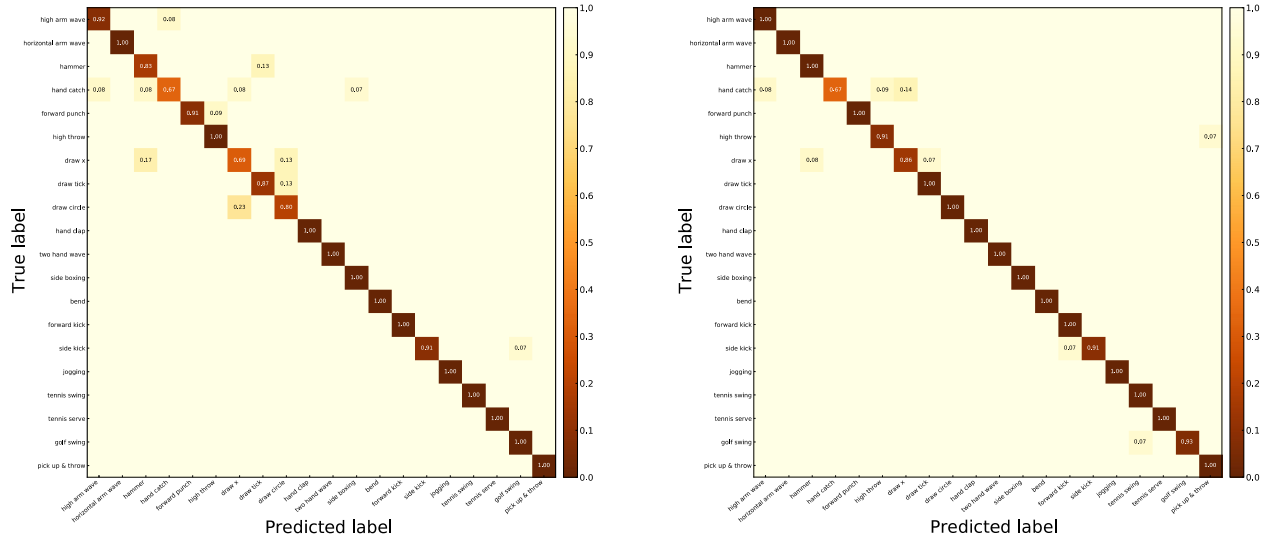
[3] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979. 2

[4] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *Advances in Neural Information Processing Systems Autodiff Workshop (NIPSW)*, 2017. 2



(a) Baseline (b) PointLSTM

Figure 5. Confusion matrix comparison on SHREC'17 dataset.



(a) Baseline (b) PointLSTM

Figure 6. Confusion matrix comparison on MSR Action3D dataset.