FlickerNet: Adaptive 3D Gesture Recognition from Sparse Point Clouds

Yuecong Min[1,2]
yuecong.min@vipl.ict.ac.cn
Xiujuan Chai[3]
chaixiujuan@caas.cn
Lei Zhao[4]
zhaolei8@huawei.com
Xilin Chen[1,2]
xlchen@ict.ac.cn

[1] Key Lab of Intelligent Information Processing of Chinese Academy of Sciences(CAS), Inst. 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, Key Lab of Agricultural Big Data, Ministry of Agriculture, Beijing, 100081, China

[4] HUAWEI Technologies CO. , LTD., Beijing, 100095, China

Abstract

Recent studies on gesture recognition use deep convolutional neural networks (CNNs) to extract spatio-temporal features from individual frames or short video clips. However, extracting features frame-by-frame will bring a lot of redundant and ambiguous gesture information. Inspired by the flicker fusion phenomena, we propose a simple but efficient network, called FlickerNet, to recognize gesture from a sequence of sparse point clouds sampled from depth videos. Different from the existing CNN-based methods, FlickerNet can adaptively recognize hand postures and hand motions from the flicker of gestures: the point clouds of the stable hand postures and the sparse point-cloud motion for fast hand motions. Notably, FlickerNet significantly outperforms the previous state-of-the-art approaches on two challenging datasets with much higher computational efficiency.

1 Introduction

Visual gesture recognition [8, 15, 22] is a promising field and provides natural interfaces for human-computer interaction. However, it's still a challenging problem due to the various hand postures, tiny finger motions, and large-scale body motions in computer vision.

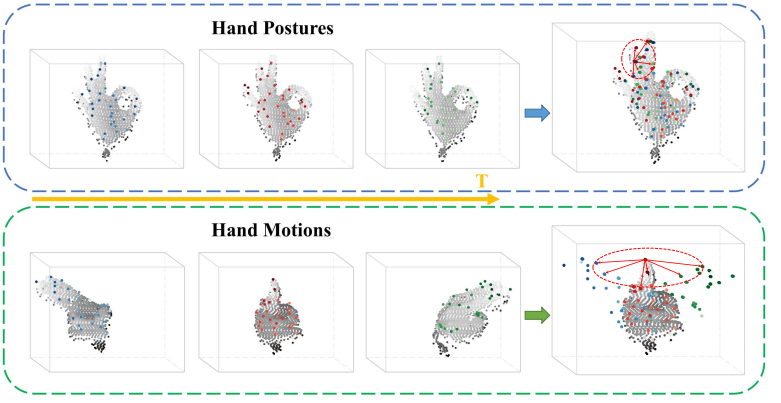


Figure 1: Three consecutive images of significant hand postures and that of hand motions with sampled points are shown in the figure. Top: there is no observable difference among frames; thus accumulated point clouds can represent hand postures. Bottom: sparse point-cloud motion can also adequately describe the hand motions. Red circles are typical examples of K Nearest Neighbor of cluster centroids, which show that sparse point clouds can adaptively represent postures and motions information.

Inspired by the successful application of Convolutional Neural Networks (CNNs) in image classification, previous studies focused on modeling the spatial and temporal information, which is also crucial in action recognition. Simonyan and Zisserman [27] proposed a two-stream network to explicitly fuse the appearance information from individual RGB frames and the motion information from optical flow. Ji *et al.* [11] designed a 3D-CNN to extract features simultaneously from spatial and temporal dimensions. To capture the long-term temporal cues in videos, Shi *et al.* [35] developed a convolutional LSTM to preserve spatial and temporal information for modeling long-range dependencies.

Different from action recognition, the purpose of gesture recognition is to understand the information conveyed through postures and motions, especially from hands and arms [6]. To convey information clearly, hand motions are relatively slow when hand postures play an essential role. Meanwhile, hand motions are the cues to recognize gestures when hands are too small to recognize or handshapes are blurred caused by fast-moving. Thus the recognition of gestures can be decoupled into two components: significant hand postures recognition in the spatial domain and hand motions recognition in the temporal domain.

Furthermore, hand postures and hand motions play different roles in gestures. As shown in Figure 1, the most significant hand postures appear with slow hand motions and the consecutive frames can be almost identical. When hands move fast, hand postures change rapidly, and hand motions may play a more active role in the recognition process. Although CNN-based methods have made significant advancements, they still have to process video clips as a whole or frame-by-frame, which extract redundant features when hands move slowly and can be easily affected by ambiguous gestures when hands move fast.

Different from CNNs, human recognize gestures by perceiving a continuous stream of information rather than a set of discrete images. Inspired by the flicker fusion phenomena [26], we propose a sparse point cloud-based method for real-time gesture recognition. We use depth data because it's more sensitive to distance changes and more robust to illumination and background changes than RGB data, which is crucial for gesture recognition.

A sequence of sparse point clouds is sampled from depth video to represent gestures. As shown in Figure 1, when significant hand postures appear, hands keep relatively static, and postures become steady with accumulated point clouds. As for hands moving rapidly, sparse point-cloud motions can effectively describe the hand motions. To adaptively learn hand postures and motions information from such a sparse point clouds representation, FlickerNet, a simple but efficient network is proposed. FlickerNet can extract features from point clouds by abstracting information from the multi-scale local spatio-temporal region. We evaluated FlickerNet on two publicly available datasets, which are NVIDIA Dynamic Hand Gesture Dataset (nvGesture) [16] and Common Japanese gestures datasets [20]. Our experiments show that the proposed network outperforms all of the previous state-of-the-art approaches on two public datasets with real-time speed on a single TITAN XP. Our main contributions are summarized as follows:

- We propose a new sparse point clouds representation for 3D gestures that can better describe the hand postures and motions from the redundant gesture information.
- A simple but efficient FlickerNet is designed to adaptively extract features for hand postures and motion trajectories from a sequence of unregistered sparse point clouds.

2 Related Work

Vision-based gesture recognition has been extensively studied over recent decades and efficiently capturing spatio-temporal information is the main challenge of gesture recognition [5, 15]. In the early stage, handcrafted features have been widely used for gesture recognition, such as histogram of oriented gradients (HOG) [7], hidden Markov model (HMM) [28] and covariance matrices [32].

With the success of deep learning in image recognition, temporal information can be captured either by extending 2D approaches to the spatio-temporal domain [3, 29] or by using other input modalities instead of frames for the network [1, 27, 34]. Several recent studies [25, 30] factorize the network to learn spatial and temporal events separately. Because of the flexible of point clouds, our work adaptively recognizes spatial and temporal events without explicit distinguishing.

Compared to action recognition, gesture recognition has lower information densities. Models are expected to implicitly extract posture and motion features from a large amount of information, which can be a difficult task. In some previous studies [13, 17, 33], hand detection [4, 10] or pose estimation [2] are used to locate hand region and eliminate noisy background. Multiple modalities [16, 31] are used to avoid overfitting to modal-specific representations and different multi-modal fusion strategies[14, 17] are used to improve performance, which leads to unacceptable inference time for practical uses.

Different from image data, the point cloud is an important type of geometric data structure with the irregular format. PointNet [23, 24] is a pioneering work that operates on unordered points set directly and has shown success in object classification and semantic segmentation. Ge *et al*. [8, 9] proposed networks to learn 3D hand articulations from the 3D point cloud and Kingkan *et al*. [12] directly fused consecutive point cloud frames on the point level without registration and applied PointNet with attention modules to recognize human gesture from consecutive point cloud frames. However, the point clouds of a gesture isn't an unordered point sets, previous works didn't take the temporal evolution of the point cloud into consideration. Efficiently spatio-temporal feature extraction from point clouds is remained to be solved and the approach proposed in this work is a novel attempt.

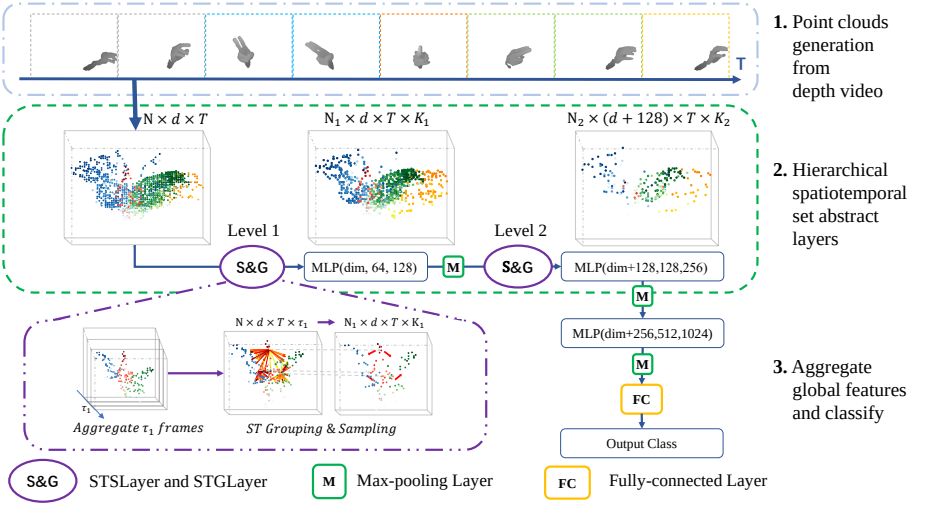


Figure 2: The overall framework of our FlickerNet. (1) A sequence of sparse point clouds is generated from the depth video, (2) spatio-temporal features are extracted from the abstraction layers, (3) aggregates global features across the temporal axis and classifies. A single level spatio-temporal sampling and grouping layer is visualized in the bottom left: for each timestep, point clouds within a short temporal period τ are combined in the same spatial space, and a spatial grouping operation builds local regions for cluster centroids, which are then fed into MLPs to adaptively extract hand postures and motions features. More details are given in the supplementary material.

3 Methdology

The framework of our approach is shown in Figure 2. Given a depth video, our method first downsamples and converts it into a sequence of sparse point clouds. Then the sparse point clouds pass through two spatio-temporal set abstraction levels, and local features are aggregated to recognize gestures. In the rest of this section, we first briefly review PointNet++ that closely relates to the proposed method, and then present network architecture and spatio-temporal set abstraction layer.

3.1 Review of PointNet++

PointNet++ [24] is a powerful model to extract hierarchical features from unordered point sets. Set abstraction levels is proposed to capture local context at different scales. Given an unordered point set $\mathcal{X} = \{x_i \in \mathbb{R}^n | i = 1, 2, \cdots, N\}$, a set abstraction level can be defined as a set function $f : \mathcal{X} \rightarrow \mathcal{Y}$ that maps the given set to a sparser subset $\mathcal{Y} = \{y_i \in \mathbb{R}^m | i = 1, 2, \cdots, M, m > n, M < N\}$ with higher dimension, which can be written as:

$$y_i = f(x_{p_i}) = \gamma \left(\max_{j \in \mathcal{N}(p_i), j=1,\cdots,N} \{h(g(x_{p_i}, x_j))\} \right), \quad (1)$$

where γ and h usually are multi-layer perceptron (MLP) networks, and g is a group operation to construct local regions based on the spatial neighborhoods $\mathcal{N}(p_i)$ of the centroids x_{p_i}.

3.2 Spatio-temporal Set Abstraction Level.

Different from classification and segmentation task, regarding a gesture as an unordered point set will lose temporal information, so the original set abstraction is no longer suitable for gesture recognition. Therefore, we propose a spatio-temporal set abstraction level which takes a sequence of point clouds $\{x_{i,t}|i=1,2,\cdots,N;t=1,2,\cdots,T\}$ as input, and then extracts spatio-temporal features from a local spatial region within a short period τ, using the following Eq 2:

$$y_{i,t} = f(x_{p_i,t}) = \gamma\left(\max_{(j,t+\Delta t)\in\mathcal{N}(p_i,t),j=1,\cdots,n}\{h(g(x_{p_i,t},x_{j,t+\Delta t}))\}\right), \tag{2}$$

where γ and h usually are multi-layer perceptron (MLP) networks, and g is a group operation to construct local regions based on the spatio-temporal neighborhoods $\mathcal{N}(p_i,t)$ of the centroids $x_{p_i,t}$ within a short period τ $(-\tau/2<\Delta t<\tau/2)$. Different from previous methods [12, 23, 24], the proposed method keeps the temporal information throughout the set abstraction. Besides, the temporal evolutions between frames are captured for recognition.

To be more specific, a spatio-temporal set abstraction level contains three key layers: Spatio-temporal Sampling Layer (STSLayer), Spatio-temporal Grouping Layer (STGLayer), and PointNet Layer. It takes a $(T,N,(d+C))$ matrix per gesture as input from $T\times N$ points with d-dim coordinates and C-dim point features and outputs a $(T,\alpha N,(d+C'))$ matrix with sampling fraction α at each timestep with d-dim coordinates and new C'-dim point features summarizing local spatio-temporal information.

STSLayer. The STSLayer is designed to collect local spatio-temporal information by sampling point cloud features at each moment, which can reduce the computation cost and recognize gestures at different scales. A subset of αN points is selected at every timestep as cluster centroids.

STGLayer. The proposed STGLayer can learn more fine-grained hand postures from denser point clouds when hands keep relatively static, and more global motion information when hands move fast. Given the cluster centroids matrix $(T,\alpha N,(d+C))$, the STGLayer can construct local region sets by finding K neighboring points around the centroids within a short period τ. The output is $T\times N'$ groups of points, and the points coordinates of each group are translated to a relative coordinates to the centroid point: $\tilde{x}_{i,t}^{(j)} = x_{i,t}^{(j)} - \hat{x}^{(j)}$ for $i=1,2,\cdots,K$ and $j=1,2,\cdots,d$ where $\hat{x}$ is the coordinate of the centroid.

PointNet Layer. Spatio-temporal features need to be abstracted from groups of point sets. A simplified version of PointNet[23] with fewer layers and fewer channels are used as the basic block to avoid overfitting and the features of each local region are aggregated and transformed to a higher dimensional feature space C'.

3.3 Implementation Details

Sparse Point Cloud Generation. As the major ROIs of gestures, hand region is usually in the foreground of gesture videos, which can be segmented by depth information and use the Otsu threshold [19] to remove the background. We uniformly sample T frames from a depth video and N unordered points from the hand region of each frame. Each point in sparse point clouds is represented by both spatial and temporal features. We only use the (x,y,z,t) coordinate as our point's input channels for simplicity and efficiency.

STSLayer and STGLayer. The exact way this sampling strategy is implemented is a design choice and will be uniform sampling in our case with considering both efficiency and

reproducibility. Experiments about it can be found in the supplementary material. As for STGLayer, we conduct experiments with two strategies. The first is the original strategy in PointNet++ [24] that grouping points in spatial without explicitly modeling sequential temporal structure. With the second strategy, STGLayer groups point clouds in spatial space within a short temporal period τ, and each point set contains both local appearance and motion information (see Figure 2 bottom left). To capture both short-term and long-range changes, we apply K nearest neighbor(kNN) search to find a fixed number of neighboring points (we use k=64 empirically).

Network Configuration. A modified PointNet is used as a basic block of our network. The layer configuration is schematically shown in Figure 2. All MLP layers are implemented as 1×1 Convolution and followed by the rectification (ReLU) activation function and Batch Normalization layer. Max-pooling layers are used to aggregate information from local regions. The baseline method is the same network architecture with the original group layer, which regards gesture recognition as 3D object classification.

Training and Inference. Following common practice, we uniformly sample T frames along its temporal axis and N points are generated for each frame (in default we use T=32 and N=64). We train all the models from scratch for 40 epochs with a mini-batch size of 8. SGD with optimizer momentum is used with learning rate 10^{-2} and divided by 10 at epoch 25 and 35. The default sampling fractions of two set abstraction layer are 0.5 and 0.25. To increase variability in the training examples, we apply the following data augmentation steps to each video: random scale($\pm$20%), random rotate($\pm 15^{\circ}$) and random input dropout(20%). During inference, data pass through the networks without augmentation, and the model of last epoch is used to evaluate.

4 Experiments

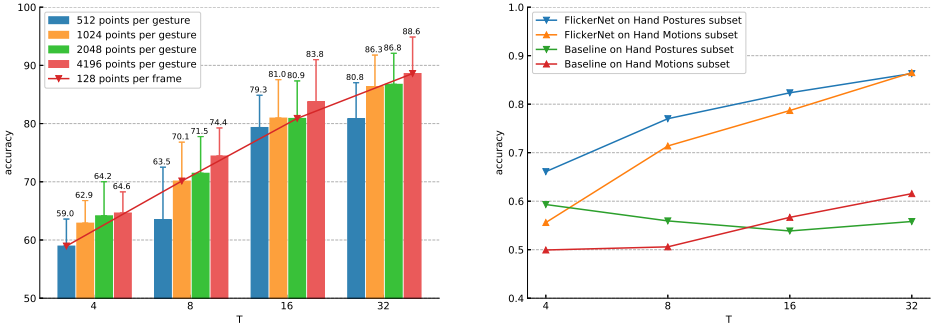
We perform comprehensive studies on the challenging nvGesture Dataset to investigate the advantages of the proposed method. To verify the generalize ability of our model, the experiments on UBPG Dataset are also reported.

4.1 Experiments on nvGesture

Dataset. nvGesture dataset [16] is a challenging dataset with 25 gesture classes proposed for human-computer interfaces in cars. The dataset is recorded with multiple sensors and viewpoints in a car simulator from 20 subjects. A total of 1532 weakly-segmented dynamic hand gestures videos are split by subject into 1050 training (about 70%) and 482 test (about 30%). Each video contains only one gesture. As the validation set is not provided, we split the training set into six groups, two subjects in each, to run 6-fold cross-validation and report average accuracy $\pm$ standard deviation in accuracy between groups. For our main results, we report accuracy on the test set.

Experiments on the number of frames and the density of points. Here we show performance changes of FlickerNet with regard to the number of frames T as well as the number of points for each frame N. We experiment with point clouds size $T \times N$ ranging from 512 to 4096 due to the limitation of GPU resources, and the results have been shown in Figure 3(a). From the figure, we can get the following conclusions:

1) Both N and T play active roles in gesture recognition: increasing N results in a 4.5–10.9% performance gain for different values of T and increasing T results in a 29.6% per-



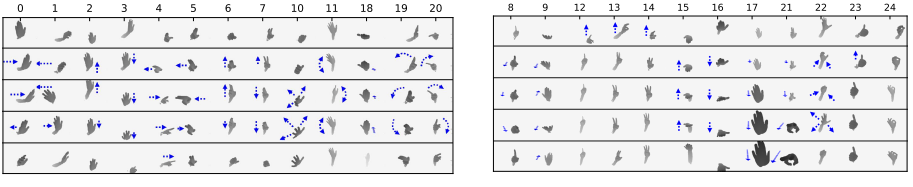
(a) (b)

Figure 3: (a) Comparisons between different combinations of the total the number of frames T and the number of points for each frame N on the validation set. Different colors represent different numbers of sampled points for gestures ($T \times N$). Red line shows the performance of different T with $N = 128$. (b) The different performance of FlickerNet and Baseline on two subsets of nvGesture Dataset with different T (number of frames).

formance gain with $N = 128$.

2) T plays a more crucial role than N with same point clouds size, the performance of 4096 points on ($T \times N = 32 \times 128$) is 24.0% better than on ($T \times N = 4 \times 1024$).

The above observations show that the FlickerNet can effectively capture gesture information from sparse point clouds. It's worth mentioning that even with 512 points ($T \times N = 32 \times 16$), the average accuracy can still reach 80.8% which is higher than some recent offline classification results for depth modality on the nvGesture benchmark.

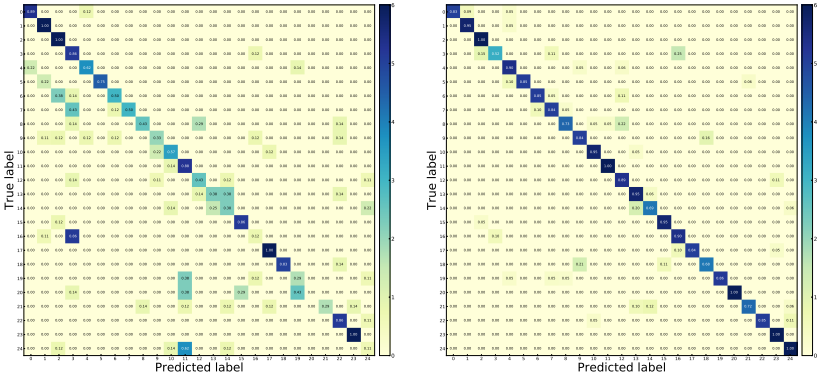


(a) (b)

Figure 4: Twenty-five dynamic hand gesture classes of the nvGesture dataset are roughly split into two categories: (a) Motion-dominant hand gestures (b) Posture-dominant hand gestures, which are used to explore the effects of sparse point clouds input.

Experiments on adaptive postures and motions information capturing. A key assumption for FlickerNet is that it can adaptively learn hand postures and hand motions from accumulated sparse point clouds. To further verify our assumptions, we subjectively split the nvGesture dataset into two subsets (Figure 4). The first subset is more related to the directions and trajectories of hand motions and the second subset is more related to the hand posture changes. Based on our assumption, FlickerNet should outperform baseline on motion-dominant hand gestures subset due to its ability to model temporal changes, and keep performance on posture-dominant hand gestures subset as the number of frames increases with accumulated sparse point clouds.

We count the average recall of our trained method on these two subsets and compare with baseline method mentioned in Section 3.3. In Figure 3(b), without explicit modeling temporal information, the baseline is better at posture-dominant hand gestures with fewer frames and the performance on these two subsets show the opposite trend when T changes.



(a) (b)

Figure 5: Confusion Matrix on the validation set with $T = 32$ and $N = 128$. (a) Baseline. (b) Proposed Method.

On the contrary, the proposed method has a much better performance on both subsets and the performance on the posture-dominant hand gestures is better than on the other one, which is beyond our expectations. These results show that the proposed spatio-temporal set abstraction layers are more robust to ambiguous gestures and the FlickerNet can effectively extract features for hand postures and motions from accumulated sparse point clouds.

To further explore the intrinsic reasons, we show the confusion matrix of several confusing classes in Figure 5. The confusing classes are easily classified by FlickerNet, which indicates that our model can effectively capture temporal evolution information. We provide more visual results in the supplementary material.

Temporal Period		Accuracy
$\tau_1 = 1$	$\tau_2 = 1$	84.78 ± 2.71
$\tau_1 = 1$	$\tau_2 = 3$	85.46 ± 4.26
$\tau_1 = 3$	$\tau_2 = 3$	87.83 ± 5.66
$\tau_1 = 3$	$\tau_2 = 5$	$\mathbf{88.15 \pm 5.04}$
$\tau_1 = 5$	$\tau_2 = 5$	86.76 ± 7.26

Table 1: Varying values of τ, the temporal period of spatio-tempora grouping layer on the validation set.

	Baseline	FlickerNet
Total	286.8	**12.9**
Set abstraction	284.2	**10.9**
Convolutional layers	2.6	**2.0**

Table 2: Comparison of inference time (ms) with point clouds input.

Experiments on the temporal period for grouping. Large temporal period may lead to indistinct point clouds. To further explore the effect of the temporal period, we try varying temporal periods τ_1 and τ_2 at two spatio-temporal set abstraction levels. As shown in Table 1, both small and large temporal strides will hurt performance. So we choose (3,5) as our temporal period default values.

Comparison with SOTA results. Table 3 shows the comparisons with state-of-the-art results in nvGesture. Our method achieves an accuracy of 86.3 with 3.7 GFLOPs, a 1.9% increase over the past best result and approaches human recognition result of 88.4% on RGB video. We also record forward time with a batch size 1 using pytorch 1.0.1 [21] with a single TITAN XP. By using preprocessed point clouds data as input, the inference time of FlickerNet is over 20 times faster than the baseline (Table 2), which is mainly contributed by spatio-temporal set abstraction level.

Model	Input	Accuracy	FLOPs
HOG+HOG^2 [18]	depth video	36.3%	-
SNV [36]	depth video	70.7%	-
C3D [29]	depth video	78.8%	38.5 G
FOANet [17]	depth video	73.7%	-
R3DCNN [16]	depth video	80.3%	37.8 G
PreRNN [37]	depth video	84.4%	38.5 G
Baseline [24]	sparse point clouds	63.9%	**3.7 G**
Ours	sparse point clouds	**86.3**%	**3.7 G**
Human [16]	color	88.4%	-

Table 3: Results on the nvGesture Dataset.

4.2 Experiments on UBPG Dataset

Dataset. Common Japanese gestures datasets [20] is a common Japanese gestures dataset collected by Kinect sensor. The dataset contains 115926 point cloud frames (3235 video samples) in 10 gesture classes. These gestures are performed by 5 subjects and each subject repeating the gestures at least 30 times. The dataset contains only the upper part of the body and is hard to get hand shape annotations, so we use the whole point clouds as inputs and show the robustness of the proposed FlickerNet.

Experiments Settings. There are two standard evaluation protocols for this dataset. In the first setting [12] (setting-1), only subject1 is used, and 24861 frames for training, 22748 frames for validation and 9776 frames for testing. In the second partition [20] (setting-2), for each gesture class, 70% samples(64420) are used for training and the rest(28758) for testing. Besides, cross-subject evaluation (setting-3) is used as the third setting to evaluate the generalization ability of proposed methods, two of the subjects with 60% samples are used to train and the rest to testing.

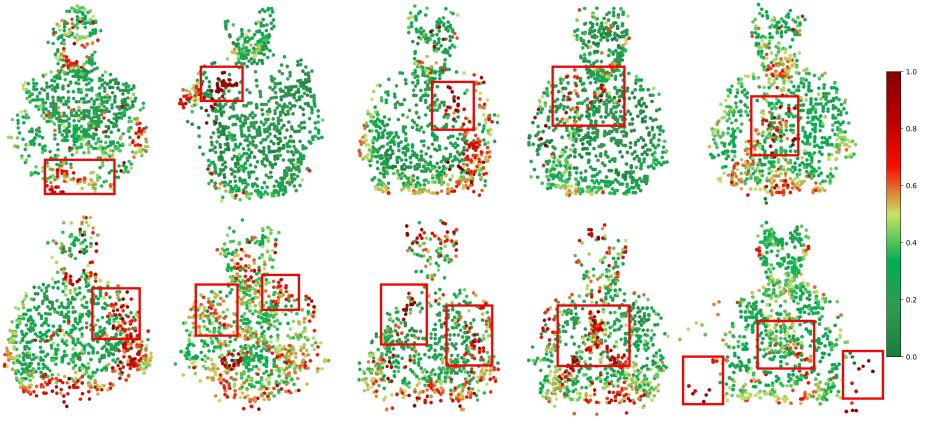


Figure 6: Point Activation Cloud: the mid-level features are mapped back to each point to generate point activation clouds (PACs). The PACs highlights the discriminative parts of point clouds.

RoIs of FlickerNet. Different from nvGesture Dataset, in this section, we use the whole point clouds as inputs without any preprocessing or attention modules. To verify FlickerNet

truly learns gesture features from hand regions, we calculate point activation cloud (PAC) from the activation of FlickerNet, which exposes the implicit attention of the network. As shown in Figure 6, points of hand regions with higher responses, which shows FlickerNet can adaptively focus on hand regions.

For a given sparse point clouds, let $f^{(k)}(j,i)$ represent the kth activation of point i which around the centroid point j before the second max-pooling layer. And each point values of PAC is given by:

$$PAC_i = \max_{i \in N(j), j} \max_{k} f^{(k)}(j,i), \tag{3}$$

Model	Input	Setting-1	Setting-2	Setting-3
Random forest	voxels	-	67.6	-
3DCNN [20]	voxels	-	84.4	-
PointNet [23]	points, 1 frames	91.8	-	-
PANet [12]	points, 1 frames	92.5	-	-
PANet [12]	points, 4 frames	94.2	-	-
Baseline [24]	sparse points, 16 frames	90.2	89.8	61.9
Ours	sparse points, 16 frames	**96.4**	**95.3**	**77.9**

Table 4: Results on the common Japanese gestures datasets.

Comparison with SOTA results. In Table 4, the proposed method is compared with previous results on UBPG dataset and our results are better than current results in both two standard settings. Different from the nvGesture dataset, the performance of baseline is much higher, the main reason may be fewer classes and less similarity between classes.

5 Conclusions

In this work, we propose FlickerNet, a simple but efficient network for 3D gesture recognition. FlickerNet naturally takes a sequence of sparse point clouds from hand regions as inputs and experiments show that FlickerNet can adaptively extract features of hand postures and motions from accumulated sparse point clouds. And the proposed spatio-temporal set abstraction levels can dramatically improve grouping speed. As we have shown, the proposed method can achieve state-of-the-art accuracy with high efficiency. In the future, it's worthwhile thinking how to build an end-to-end architecture which can recognize gestures and actions without pre-processing steps such as hand region detection and segmentation.

References

[1] H. Bilen, B. Fernando, E. Gavves, A. Vedaldi, and S. Gould. Dynamic image networks for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3034–3042, 2016.

[2] Z. Cao, G. Hidalgo, T. Simon, S. E. Wei, and Y. Sheikh. OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. *arXiv preprint arXiv:1812.08008*, 2018.

[3] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6299–6308, 2017.

[4] X. Chai, Z. Liu, F. Yin, Z. Liu, and X. Chen. Two streams recurrent neural networks for large-scale continuous gesture recognition. In *Proceedings of the IEEE International Conference on Pattern Recognition (ICPR)*, pages 31–36. IEEE, 2016.

[5] H. Cheng, L. Yang, and Z. Liu. Survey on 3d hand gesture recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(9):1659–1673, 2016.

[6] H. Cooper, B. Holt, and R. Bowden. Sign language recognition. In *Visual Analysis of Humans*, pages 539–562. Springer, 2011.

[7] W. T. Freeman and M. Roth. Orientation histograms for hand gesture recognition. In *International workshop on automatic face and gesture recognition (FG)*, volume 12, pages 296–301, 1995.

[8] L. Ge, Y. Cai, J. Weng, and J. Yuan. Hand pointnet: 3d hand pose estimation using point sets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8417–8426, 2018.

[9] L. Ge, Z. Ren, and J. Yuan. Point-to-point regression pointnet for 3d hand pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 475–491, 2018.

[10] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision (ICCV)*, pages 1440–1448, 2015.

[11] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1): 221–231, 2013.

[12] C. Kingkan, J. Owoyemi, and K. Hashimoto. Point attention network for gesture recognition using point cloud data. In *Proceedings of the British Machine Vision Conference (BMVC) (pp. 1–13)*, pages 475–491, 2018.

[13] C. Lin, J. Wan, Y. Liang, and S.Z. Li. Large-scale isolated gesture recognition using a refined fused model based on masked res-c3d network and skeleton lstm. In *Proceedings of the IEEE International Conference on Automatic Face & Gesture Recognition (FG)*, pages 52–58. IEEE, 2018.

[14] Q. Miao, Y. Li, W. Ouyang, Z. Ma, X. Xu, W. Shi, and X. Cao. Multimodal gesture recognition based on the resc3d network. In *Proceedings of the IEEE International Conference on Computer Vision Workshops (CVPRW)*, pages 3047–3055, 2017.

[15] S. Mitra and T. Acharya. Gesture recognition: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(3):311–324, 2007.

[16] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz. Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4207–4215, 2016.

[17] P. Narayana, J. R. Beveridge, and B. A. Draper. Gesture recognition: Focus on the hands. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[18] E. Ohn-Bar and M. M. Trivedi. Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations. *Proceedings of the IEEE transactions on intelligent transportation systems*, 15(6):2368–2377, 2014.

[19] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979. doi: 10.1109/TSMC.1979.4310076.

[20] J. Owoyemi and K. Hashimoto. Spatiotemporal learning of dynamic gestures from 3d point cloud data. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–5. IEEE, 2018.

[21] 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.

[22] V. I. Pavlovic, R. Sharma, and T. S. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (7):677–695, 1997.

[23] C.R. Qi, H. Su, K. Mo, and L. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 652–660, 2017.

[24] C.R. Qi, L. Y, H. Su, and L. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems (NIPS)*, pages 5099–5108, 2017.

[25] Z. Qiu, T. Yao, and T. Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 5533–5541, 2017.

[26] E. Simonson and J. Brozek. Flicker fusion frequency: background and applications. *Physiological reviews*, 32(3):349–378, 1952.

[27] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems (NIPS)*, pages 568–576, 2014.

[28] T. Starner, J. Weaver, and A. Pentland. Real-time american sign language recognition using desk and wearable computer based video. *IEEE Transactions on pattern analysis and machine intelligence*, 20(12):1371–1375, 1998.

[29] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision (ICCV)*, pages 4489–4497, 2015.

[30] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6450–6459, 2018.

[31] J. Wan, Y. Zhao, S. Zhou, I. Guyon, S. Escalera, and S. Z. Li. Chalearn looking at people rgb-d isolated and continuous datasets for gesture recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 56–64, 2016.

[32] H. Wang, X. Chai, X. Hong, G. Zhao, and X. Chen. Isolated sign language recognition with grassmann covariance matrices. *ACM Transactions on Accessible Computing (TACCESS)*, 8(4):14, 2016.

[33] H. Wang, P. Wang, Z. Song, and W. Li. Large-scale multimodal gesture recognition using heterogeneous networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3129–3137, 2017.

[34] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *Proceedings of the IEEE Conference on European conference on computer vision (ECCV)*, pages 20–36. Springer, 2016.

[35] S. Xingjian, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems (NIPS)*, pages 802–810, 2015.

[36] X. Yang and Y. Tian. Super normal vector for activity recognition using depth sequences. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 804–811, 2014.

[37] X. Yang, P. Molchanov, and J. Kautz. Making convolutional networks recurrent for visual sequence learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6469–6478, 2018.