LPSNet: A Novel Log Path Signature Feature based Hand Gesture Recognition Framework

Chenyang Li, Xin Zhang* and Lianwen Jin
School of Electronic and Information Engineering
South China University of Technology
Guangzhou, P. R. China
eexinzhang@scut.edu.cn

Abstract

*Hand gesture recognition is gaining more attentions because it's a natural and intuitive mode of human computer interaction. Hand gesture recognition still faces great challenges for the real-world applications due to the gesture variance and individual difference. In this paper, we propose the LPSNet, an end-to-end deep neural network based hand gesture recognition framework with novel log path signature features. We pioneer a robust feature, path signature (PS) and its compressed version, log path signature (LPS) to extract effective feature of hand gestures. Also, we present a new method based on PS and LPS to effectively combine RGB and depth videos. Further, we propose a statistical method, DropFrame, to enlarge the data set and increase its diversity. By testing on a well-known public dataset, Sheffield Kinect Gesture (SKIG), our method achieves classification rate as **96.7%** (only use RGB videos) and **98.7%** (combining RGB and Depth videos), which is the best result comparing with state-of-the-art methods.*

1. Introduction

With the development of intelligent devices (*e.g.*, AR, VR and smart-home devices), gesture interaction is attracting more and more attention because it is more closely related to human expression instinct and habits. Many popular products (*e.g.*, Microsoft HoloLens, Facebook Oculus and DJI MAVIC) involve hand gestures as one of the interaction tools. However, these products are limited to specific usage scenarios, in which only a small amount of gestures can be used or a specific touch needs to be held. Therefore, a more universal and robust hand gesture recognition algorithm is still in pressing need.

Hand gestures can be divided as two categories: static hand gesture and dynamic hand gestures. Static hand gesture recognition can be treated as the well-studied object recognition problem. For the dynamic hand gesture recognition, recently, few convolutional neural network (CNN) based methods have achieved promising results [16, 18]. However, for the real-world application, we still face great challenges. Firstly, the hand gesture has high variance due to different users and various application environments. For example, any specific gesture performed by different people can be in different speeds, shapes, hand poses and background surroundings. Although deep neural network can learn effective features of the hand gesture, it requires large diversified training data for generalization. Researchers choose to introduce complex model like R3DCNN [16], RNN [18], but the high computational cost prevents them from the wide application. By adding the optical flow, the result is getting better with the same network [16]. Hence, extracting essential and representable features may be the possible solution. Secondly, with the improvement of hardware, we can now obtain the color and depth information easily and multi-stream and multi-resource framework has shown promising results. Nevertheless, methods of combining these information are relatively simple so that we need a more effective way to make them fully complementary.

In this paper, we present an end-to-end hand gesture recognition framework with novel log path signature feature, named LPSNet. The contributions are as follows: i) to the best of our knowledge, we are the first to introduce the path signature (PS) [2] concept into the hand gesture recognition problem; ii) further, considering the unique characteristic of hand gestures, we employ the log path signature (LPS) [21], an extended, compressed and sparse version of PS; iii) to simulate various hand gestures with different speeds and trajectories, we propose an statistical data enhancement method, DropFrame; iv) we propose an effective method to combine RGB and depth information based on PS and LPS. By testing on a public dataset and comparing with state-of-the-art methods, we achieve the best recognition accuracy on both RGB and RGB-D sequences.

2. Related work

Hand gesture recognition methods can be mainly divided into two types: selected feature based methods and deep learning (DL) based methods. As in the first category, many selected features have been used to character hand shape, physical structure and motion features. [4] extracts four different sets of feature descriptors including the distances of the fingertips from the hand center and from the palm plane, the curvature of the hand contour and the geometry of the palm region. To detect the positions of hand parts, they firstly employ skin-based segmentation and a circle fitting algorithm and 3D plane fitting for palm detection, but the skin-based method is not very robust especially when the intensity of light changes violently. [15] designs a local descriptor called a 3D Motion Scale-Invariant Feature Transform (3D MoSIFT), which detects the interesting points and consequently encodes the motion information to describe the interesting points based on RGB and depth information. 3D MoSIFT characters the motion features well but it requires some complex prophase calculation and tends to have high dimension.

In recent years, a trend toward using deep neural network to extract feature is growing. In [24], a two-stream CNN is introduced, which guides the network to learn from both spatial and temporal inputs. Later, [5] further discusses where is the best position to fuse two streams. They demonstrate that the two-stream architecture and the multi-frame dense optical flow are useful in action recognition tasks. In the context of hand gesture recognition, a multi-stream Recurrent Neural Network (MRNN) to combine the RGB and depth sequences is presented in [18]. They carefully design the network to fuse multiple temporal modalities using multiple streams of recurrent neural networks with Long Short-Term Memory cells (LSTM-RNNs). [16]proposes a Recurrent 3D Convolutional Neural Network (R3DCNN) to utilize multi-modal information. They use a pre-trained 3D-CNN for local spatio-temporal feature extracting, a recurrent layer for global temporal modelling, a softmax layer for probabilities predicting, and finally a connectionist temporal classification for class labels predicting. Deep neural network methods often require a well-designed structure, a pre-trained CNN and a careful designed training strategy for RNN.

Although DL based algorithms have achieved good results, their abilities to describe hand gesture characteristics are still limited, especially hand shape under different light intensities and hand motion with varying speeds. Therefore, we propose to introduce a discriminative feature PS and its extended and compressed version LPS as the hand gesture feature descriptors, which will greatly reduce the structure and training requirements for the subsequent classification network. PS was first proposed in [2] in the form of noncommutative formal power series. After that PS was used to solve differential equations driven by rough paths [6, 13]. More recently, PS has been successfully applied in sound compression [14], online handwriting recognition [7, 10, 26, 27] and human action recognition [28] for its remarkable performance in extracting information contained in a finite path. In this work, by considering every dynamic hand gesture as a finite trajectory, we apply PS and LPS to obtain essential features for the final classification.

3. Path signature and log path signature

In this section, we will briefly introduce the mathematical definition of PS and its extended version, LPS. This section is mainly referred to [3].

3.1. Path signature

Before giving the definition of PS, we first introduce the iterated integral of path. Assume a path $P : [t_1, t_2] \to \mathbb{R}^d$, where $[t_1, t_2]$ is a time interval. The coordinate paths are denoted by $(P_t^1, ..., P_t^d)$, where each $P^i : [t_1, t_2] \to \mathbb{R}$ is a real-value path. For any single index $i \in \{1, ..., d\}$, define the quantity:

$$S(P)^i_{t_1,t} = \int_{t_1<a<t} dP^i_a = P^i_t - P^i_{t_1} \tag{1}$$

which is the increment of the *i*-th coordinate of path P at time $t \in [t_1, t_2]$. Note that $S(P)^i_{t_1,t} : [t_1, t_2] \to \mathbb{R}$ is itself a real-value path.

Further, for any pair $i, j \in \{1, ..., d\}$, define the *double-iterated integral* as:

$$\begin{aligned} S(P)^{i,j}_{t_1,t} &= \int_{t_1<a<t} S(P)^i_{t_1,a} dP^j_a \\ &= \int_{t_1<b<a<t} dP^i_b dP^j_a \end{aligned} \tag{2}$$

in which $S(P)^i_{t_1,a}$ is given by Eq. 1. The *double-iterated integral* represents path curvature. Emphasis again that $S(P)^i_{t_1,a}$ and P^j_a are simply real-valued paths, so the Eq. 2 is a special case of the path integral. $S(P)^{i,j}_{t_1,t} : [t_1, t_2] \to \mathbb{R}$ is itself a real-value path.

Continue recursively, for any positive integer $k \geq 1$ and the collection of indexes $i_1, ..., i_k \in \{1, ..., d\}$, define:

$$S(P)^{i_1,...,i_k}_{t_1,t} = \int_{t_1<a<t} S(P)^{i_1,...,i_{k-1}}_{t_1,a} dP^{i_k}_a \tag{3}$$

As before, because $S(P)^{i_1,...,i_{k-1}}_{t_1,a}$ and $P^{i_k}_a$ are real-valued paths, the Eq. 3 is defined as a path integral, and $S(P)^{i_1,...,i_k}_{t_1,t} : [t_1, t_2] \to \mathbb{R}$ is itself a real-valued path. The Eq. 3 can be equivalently written as:

$$S(P)^{i_1,...,i_k}_{t_1,t} = \int_{t_1<a_k<t} ... \int_{t_1<a_1<a_2} dP^{i_1}_{a_1} ... dP^{i_k}_{a_k} \tag{4}$$

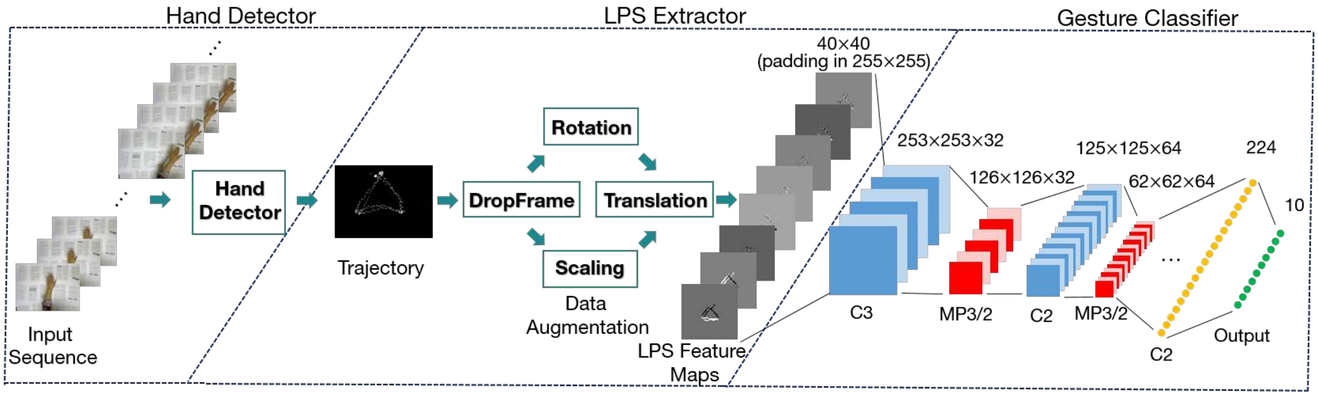

Figure 1. Flowchart Overview of the LPSNet for the gesture recognition .

The real number $S(P)^{i_1,...,i_k}_{t_1,t_2}$ is called the *k-fold iterated integral* of P long indexes $i_1, ..., i_k$.

The definition of PS is given as follows. The signature of a path $P : [t_1, t_2] \to \mathbb{R}^d$, denoted by $S(P)_{t_1,t_2}$, is the collection (infinite series) of all the iterated integrals of P. $S(P)_{t_1,t_2}$ can be expressed as a sequence of real numbers:

$$\begin{aligned} S(P)_{t_1,t_2} =& (1, S(P)^1_{t_1,t_2}, ..., S(P)^d_{t_1,t_2}, \\ & S(P)^{1,1}_{t_1,t_2}, S(P)^{1,2}_{t_1,t_2}, ...) \end{aligned} \tag{5}$$

The superscripts run along the set of all multi-indexes:

$$I = \{(i_1, ..., i_k) | k \geq 1, i_1, ..., i_k \in \{1, ..., d\}\} \tag{6}$$

The k-th level PS is the collection (finite series) of all the *k-fold iterated integral* of path P. For example, the *1*-th level PS of path P is $(S(P)^1_{t_1,t_2}, ..., S(P)^d_{t_1,t_2})$, and the *2*-th level PS of path P is $(S(P)^{1,1}_{t_1,t_2}, S(P)^{1,2}_{t_1,t_2}, ..., S(P)^{d,d}_{t_1,t_2})$. Note that the *0*-th level PS of path P is equal to 1 by convention.

In practice, we often truncate the $S(P)_{t_1,t_2}$ at level m to enture the dimension of the PS feature in a reasonable range. The dimension of $S(P)_{t_1,t_2}$ truncated at level m is calculated through:

$$M = 1 + d + \cdots + d^m \tag{7}$$

We empirically set m to 1,2,3 and 4 in our experiments because the PS of a higher level typically characterizes more trivial details of a path and do not lead to further improvement.

3.2. Log path signature

The LPS can be obtained from PS by taking the formal logarithm of PS in algebra of formal power series. It is an extended, compressed and sparse version of PS.

According to Chen's identity [23], there is another way to describe the PS of P truncated at level m by a formal power series where $e_1, ..., e_d$ are d formal indeterminates and the coefficient of each monomial $e_{i_1} \cdots e_{i_k}$ is $S(P)^{i_1,...,i_k}_{t_1,t_2}$. For simplicity, we choose the same symbol $S(P)_{t_1,t_2}$ to express this representation:

$$S(P)_{t_1,t_2} = \sum_{k=0}^{m} \sum_{i_1,...,i_k \in 1,...,d} S(P)^{i_1,...,i_k}_{t_1,t_2} e_{i_1} \cdots e_{i_k} \tag{8}$$

Still, the *0*-th level of the PS $S(P)^0_{t_1,t_2} = 1$.

For a power series:

$$s = \sum_{k=0}^{\infty} \sum_{i_1,...,i_k \in 1,...,d} \lambda_{i_1,...,i_k} e_{i_1} \cdots e_{i_k} \tag{9}$$

Assume that $\lambda_0 > 0$, then its logarithm can be defined as the power series:

$$\log s = \log \lambda_0 + \sum_{n \geq 1} \frac{(-1)^n}{n} (1 - \frac{s}{\lambda_0})^{\otimes n} \tag{10}$$

where the product $\otimes$ between monomials is defined by joining together multi-indexes:

$$e_{i_1} \cdots e_{i_k} \otimes e_{j_1} \cdots e_{j_k} = e_{i_1} \cdots e_{i_k} e_{j_1} \cdots e_{j_k} \tag{11}$$

The definition of LPS is given as follows. The LPS of a path $P : [t_1, t_2] \to \mathbb{R}^d$, is defined as the formal power series $\log S(P)_{t_1,t_2}$. We can calculate it by substituting $s = S(P)_{t_1,t_2}$ into Eq. 10:

$$\log S(P)_{t_1,t_2} = \sum_{n \geq 1} \frac{(-1)^n}{n} (1 - S(P)_{t_1,t_2})^{\otimes n} \tag{12}$$

For two formal power series s_1 and s_2, the their Lie bracket [23] is given by:

$$[s_1, s_2] = s_1 \otimes s_2 - s_2 \otimes s_1 \tag{13}$$

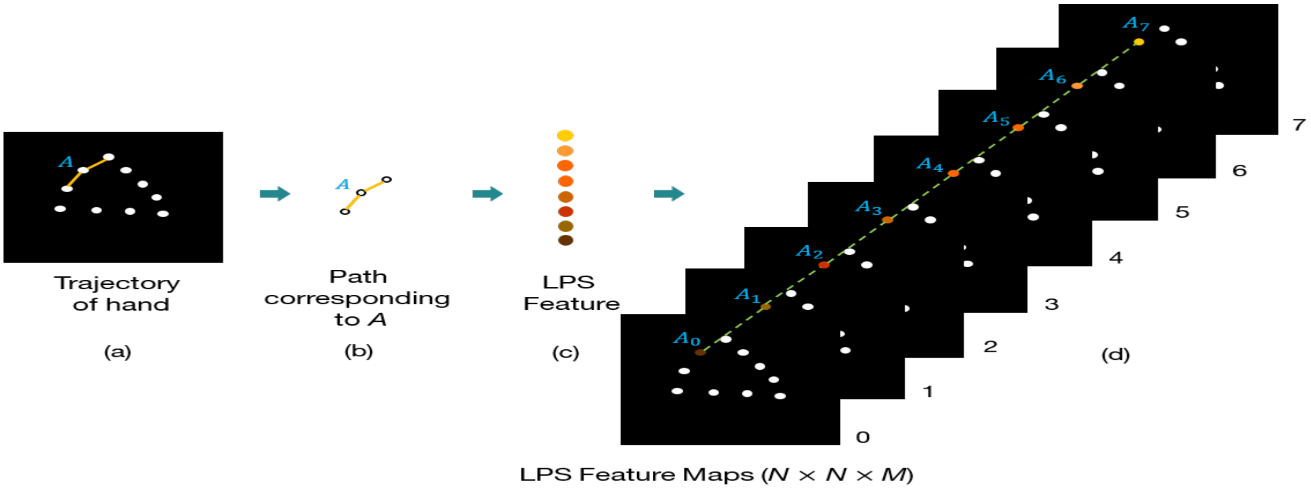

Figure 2. The process of calculating and storing LPS feature for one point (A) of the hand trajectory.

A direct computation shows that the first few terms of the LPS are:

$$\log S(P)_{t_1,t_2} = \sum_{i=1}^{d} S(P)^i_{t_1,t_2} e_i + \sum_{1\le i<j\le d} \frac{1}{2}(S(P)^{i,j}_{t_1,t_2} - S(P)^{j,i}_{t_1,t_2})[e_i, e_j] + \cdots \tag{14}$$

For the same number of levels of the PS, the LPS contains the same information in fewer numbers. A formula for the number of each level of the LPS is mentioned in [25].

4. Proposed method

Our proposed LPSNet consists of three major components: hand detector, LPS feature extractor and gesture classifier, as shown in Fig. 1. Firstly, we employ the Faster R-CNN [22] to return hand positions. Next, given the hand gesture trajectory, we propose a DropFrame method, which is followed by rotation, scaling and translation to increase diversity of training data (all the data augmentation methods are only utilized during training). Then, we compute the LPS feature maps based on the extracted hand gesture trajectory. Finally, the CNN-based network is used to provide the classification result.

4.1. Hand detector

The hand detector provides the bounding box of the hand position accurately regardless of illumination, background and hand pose change. After detection, we gain the trajectory of every gesture video (we use the left-top point of the bounding box as the trajectory in our experiment). Note that Faster-RCNN can be replaced by any other detection algorithms such as SSD [12], YOLO [19, 20] and so on. We manually label the hand bounding box of few sequences from the dataset for the training, which can be downloaded from `http://www.hcii-lab.net/data/`.

4.2. Data augmentation

Deep learning model usually requires sufficient and diverse training data to ensure its generalization capability. Therefore, beside a mix of affine transformations (*i.e.*, rotation, scaling and translation), inspired by DropStroke [26], we propose a data-augmentation method named DropFrame, which randomly omits some frames form the sequences in each training batch and generates a set of new hand trajectory sequences. Beside increasing the number of training samples, DropFrame can also simulate various hand gestures with different speeds and trajectories, which enhances the generalization. For instance, if some frames among an interval are selected to be dropped, the hand speed will become faster.

Assume that there are N frames in a hand sequences and the DropFrame rate is r, so the number of dropped out frames n is equal to $floor(N \times r)$. Consequently, the number of new possible hand sequences is:

$$C^n_N = \frac{N!}{n!(N-n)!} \tag{15}$$

Furthermore, the number of all the generated hand sequences is $\sum_{i=1}^{n} C^i_N$. The larger the DropFrame rate, the more hand sequences will be generated.

4.3. Log path signature (LPS) feature maps generation

After data augmentation, we have many different trajectories of the same hand gesture. Here we will discuss how to

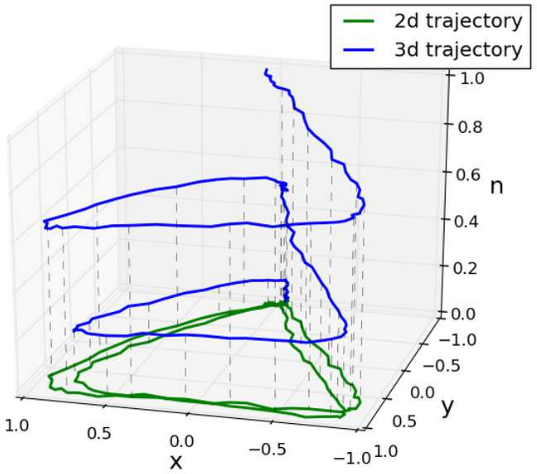

Figure 3. Time dimension is added as a third dimension (use the normalized coordinates).

represent the obtained trajectory and how to compute corresponding LPS feature maps. It is the essential step specially designed for the hand gesture. Note that the method to generate PS feature is similar with that for LPS feature, just replace the LPS with PS. We call the framework PSNet correspondingly.

Two-dimensional trajectory representation (2*d*). In the hand gesture sequence, the hand position of every frame can be expressed as two-dimensional coordinate (x, y), where x and y are the normalized spatial coordinates in the image. By connecting every position, we can obtain the hand gesture trajectory, as shown in Fig. 2(a).

Three-dimensional trajectory representation (3*d*). The LPS of 2D trajectory (LPS_2d) can uniquely characterize the path if it contains no part that exactly retraces itself [1, 8]. However, in hand trajectories, some local parts do retrace themselves due to slight hand shaking. For instance, path $\{(0, 0), (0.3, 0.3), (0.4, 0.5), (0.3, 0.3)\}$ has the same LPS with path $\{(0, 0), (0.3, 0.3)\}$ since the local part $\{(0.3, 0.3), (0.4, 0.5), (0.3, 0.3)\}$ retraces itself. Under this circumstance, the LPS_2d is unable to express a path uniquely.

To handle this problem, we redefine (x, y) as (x, y, n), where n is a monotone time dimension of the hand gesture sequence (n=0,1,2,3,...). For example, $(0.3, 0.5, 0)$ means it is the first frame of the gesture sequence and the hand position is normalized x, y coordinates 0.3, 0.5. This concept is demonstrated in Fig. 3.

Four-dimensional trajectory representation (4*d*). The LPS_3d can characterize the path uniquely because the 3D trajectories contain no part that exactly retraces itself. Nevertheless, the LPS_3d cannot express the vertical movement clearly. For example, when a part of a 3d trajectory is $\{(0.3, 0.3, 0), (0.3, 0.3, 1), (0.3, 0.3, 2)\}$, we actually cannot determine the hand is stationary or is moving vertically.

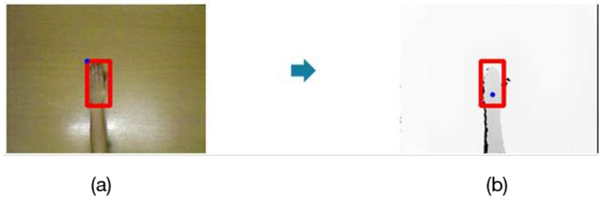

Figure 4. Illustration of how to generate d in the four-dimensional coordinate (x, y, d, n).

In order to describe the vertical movement , we propose a method to combine the depth information with the RGB information. As shown in Fig. 4(a), we get the bounding box from the hand detector and use the left-top point as the track point of the hand. Then we draw a bounding box at the same position on the corresponding depth frame and use the pixel value of a fixed point in this bounding box as the depth information of the hand (as Fig. 4(b)). Consequently, we turn the 3d expression (x, y, n) into the 4d one (x, y, d, n), where d is the depth information calculated by the above method.

LPS feature maps computation. Theoretical speaking, we can directly compute its LPS features according to the definition and then exrpess them with a vector but this method could miss important details and neighboring relationship. Hence, we propose to compute segment-wise LPS feature, and re-organize them as a series of feature maps in terms of the PS order. We will introduce a general method to generate an $N \times N \times M$ LPS feature maps ($N \times N$ denotes the shape of feature map in Fig. 2 and Fig. 3, M represents the dimension of feature) for both LPS_2d, LPS_3d and LPS_4d. Note that M is determined by trajectory dimension d and truncated level m.

The detail about how to construct an $N \times N \times M$ input for hand gesture classifier is as follows: i) initialize an array of size $N \times N \times M$ to all zeros; ii) calculate the LPS feature corresponding to each point of the hand trajectory. For instance, take w points before and after point A in Fig. 2 to generate a path P_A for A (w is 1 in Fig. 2(b) and 4 in our experiments), then calculate the LPS feature of P_A as Fig. 2(c); iii) put the LPS feature of A in the corresponding position of the $N \times N \times M$ cube as A_0-A_7 in Fig. 2(d).

In Fig. 5, we visualize the feature maps of LPS_2d truncated at level 4 of ten gestures in SKIG [11]. Each row includes the feature maps corresponding to two hand categories. And each column is a one dimension of the truncated LPS feature, which represents special information (*e.g.*, displacement in x-axis, displacement in y-axis). The way to calculate the number of columns per k is mentioned in

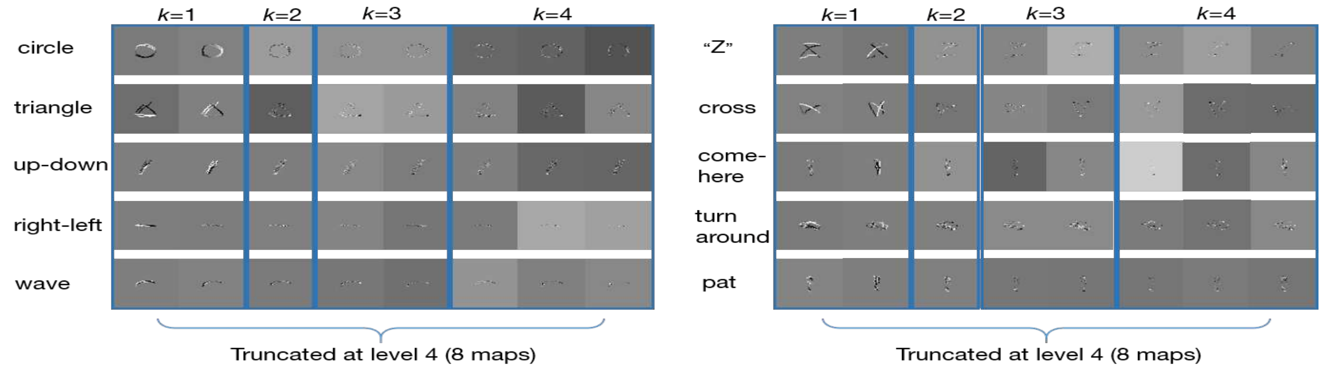

Figure 5. Visualization the feature maps of LPS_2d truncated at level 4 (randomly selecting ten samples from ten categories of SKIG [11] respectively).

Section 3. When the trajectories of hand move in a horizontal direction (such as circle, triangle, right-left, wave, Z and cross), we can distinguish them well. However, when the hand trajectories mainly move in the vertical direction (*e.g.*, up-down, comehere, turnaround and pat), we need to use the higher level LPS feature or even depth information to distinguish them.

We want to emphasize again that LPSNet can be easily turned into PSNet only by replacing the LPS mentioned above by PS, More comparisons and discussions will be presented in Section 5.

4.4. Hand gesture classifier

Finally, we train a deep convolutional neural network (DCNN) based on LPS feature maps. As shown in Fig. 1 (right part), our classification network contains eight layers with weights, among which the first seven layers are convolutional layers and the last one is fully-connected layer. The output of the last fully-connected layer is fed to a n-way softmax. The first six convolutional layers are followed by max-pooling layers which are carried out over a 3×3 pixel window with a stride of 2 pixels. The size of the convolutional filter is 3×3 for the first layer and 2×2 for the others, with a constant stride of 1 pixel. Rectified linear units (ReLUs) [17] are used as activation functions for neurons in all convolutional layers and fully-connected layers. We set the number of convolutional filter kernels to 32 for the first layer and increase it by an increment of 32 after each max-pooling. We render the hand trajectory as a 40×40 bitmap embedded in a 255×255 image that is initialized to zero, which is more convenient for data augmentation (*e.g.*, rotation, scaling and translation). In summary, the architecture of our classification network is $M \times 255 \times 255$-$32C3$-$MP3/2$-$64C2$-$MP3/2$-$96C2$-$MP3/2$-$128C2$-$MP3/2$-$160C2$-$MP3/2$-$192C2$-$MP3/2$-$224C2$-$output$.

In order to handle different training samples, we also propose a small network. We make the following changes to the network mentioned above: i) reduce 2 convolutional layers to the network; ii) changing the render size from 40×40 to 20×20. For convenient, we call the first network Network_1 and the small one Network_2.

To sum up, our proposed framework—LPSNet effectively combines color and depth videos basing on LPS feature. It can not only characterize the path discriminatively and avoid the interference of background, but also dramatically reduce the dimension of the input feature.

5. Experimental results and discussion

5.1. Hand gesture dataset and network settings

The SKIG [11] dataset is organized to evaluate and advance the state-of-the-art dynamic hand gesture recognition under challenging conditions (with variable lighting conditions, different poses and multiple subjects). The dataset contains 2160 hand gesture sequences (1080 RGB sequences and 1080 corresponding depth sequences) collected from 6 subjects (people). It collects 10 categories of hand gestures in total: circle(clockwise), triangle(anti-clockwise), up-down, right-left, wave, Z, cross, comehere, turnaround and pat, as shown in Fig. 6. All types of gestures are performed with three different hand postures: fist, index and flat. The sequences are recorded under 3 different backgrounds (*i.e.*, wooden board, white plain paper and paper with characters) and 2 illumination conditions (*i.e.*, strong light and poor light) to increase the diversity. Consequently, each subject performs 10(categories)×3(poses)×3(backgrounds)×2(illumination) ×2(RGB and depth)=360 gesture sequences.

We perform our experiments on a PC with a GTX Titan X GPU. The training mini-batch size is set to 100 for both Network_1 and Network_2. For Network_1, the DropOut [9] rate of the DropOut layers after seven convolutional layers are calculated through $0.5 \times \frac{i}{l}$, where l=6 and

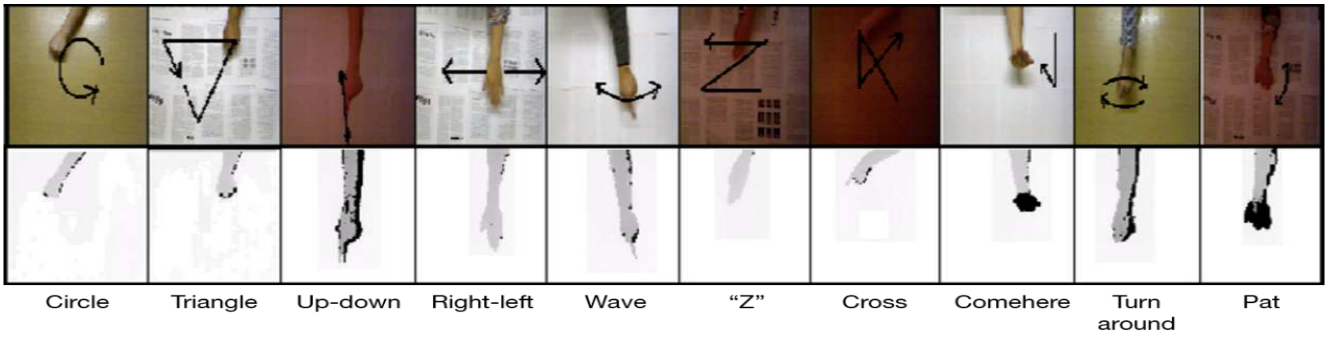

Figure 6. Some example samples of SKIG [11] dataset.

i=0,1,2,3,4,5,6 respectively for the first to seventh convolutional layers. And for Network_2, the l=4 and i=0,1,2,3,4. We set the DropFrame rate to 0.08, *i.e.*, r=0.08. To compare with other algorithms, we adopt three-fold cross-validation on SKIG (*i.e.*, four subjects for training and the other two for testing).

5.2. Evaluation

As we have discussed in Section 4.3, the hand gesture can be represented as 2D, 3D and 4D trajectories.

The dimensionality of trajectory representation. We firstly compare the 2d and 3d representation, which are denoted by (L)PS_2d and (L)PS_3d respectively in Fig. 7. The X-axis indicates different truncated levels of PS and LPS features and Y-axis expresses recognition accuracy. Our experimental results show that 3d representation is more beneficial to the recognition accuracy. It demonstrates that the (L)PS_3d is more discriminative than the (L)PS_d2 due to the ability of uniquely characterizing the paths containing some local parts that retrace themselves. We use the Network_2 as classifier for (L)PS_2d and (L)PS_3d.

Combining RGB and depth videos. We utilize the method mentioned in Section 4.3 to generate 4d track points. Because adding the depth information brings feature dimension increasing, we use Network_1 as classifier. The experimental results are shown in Fig. 7 as LPS_4d. The results show that the recognition accuracy improves significantly for all truncated levels when adding the depth information (97.59%, 98.06%, and 98.30% for truncated levels 2, 3, and 4 respectively), which demonstrates the ability of LPS_4d to effectively characterize the hand movement, especially vertical movement.

5.3. Comparison of path signature (PS) and log path signature (LPS) feature

As shown in Fig. 7, PS and LPS has comparable performance when the dimension of trajectories and truncated levels are low. When the truncated level goes high, results of both LPS and PS are getting better. But, LPS shows obvious superior performance than PS. Specifically, PS_3d is 95.28% and LPS_3d is 96.20% at the truncated level 4. It is because that LPS is more compressed and refined than PS, which is beneficial for the learning process. In other words, as the dimension of feature increases, there is redundancy in PS feature. For example, at truncated level 4, the dimensionality of LPS_3d feature is 32 while that of PS_3d has 121. In this experiment, we utilize Network_2.

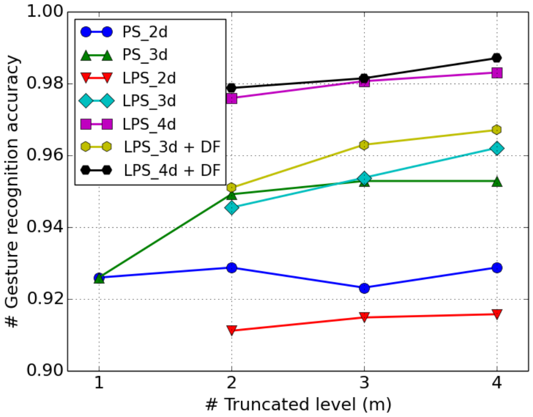

Figure 7. Comprehensive comparison of the trajectory dimensionality, PS v.s. LPS and the DropFrame method.

5.4. Evaluation and comparison of DropFrame (DF) method

We evaluate the proposed DropFrame method on LPS_3d and LPS_4d with DropFrame rate 0.08. We use Network_1 here for the reason that the training data increased dramatically after using DropFrame. The results are shown in Figure 7 as LPS_3d+DF and LPS_4d+DF. The recognition accuracies improve after adding DropFrame and the best recognition accuracies (**96.7%** for RGB and **98.7%** for RGB+depth) are observed for LPS_3d+DF and LPS_4d+DF truncated at level 4.

The results indicate that the DropFrame method do successfully simulate various hand gestures with different speeds and trajectories and dramatically increase the train samples diversity. Thus, the generalization of the network is enhanced.

5.5. Comparison with state-of-the-art methods

Confusion matrix. Table 1 and Table 2 show the confusion matrices of our proposed LPSNet (Table 1 is for LPS_3d+DF and Table 2 is for LPS_4d+DF, both are truncated at level 4). Each row of the confusion matrix represents one category of hand gesture. We use three-fold cross-validation so the sum of each row is 108. From confusion matrices, we found that LPS_3d+DF network often confused c (up-down), h (comehere), i (turnaround) and j (pat). Since these four kinds of gestures move mainly in the vertical direction, it is difficult to distinguish them by only using RGB sequences. Therefore, when we added depth information to our network (as LPS_4d+DF), the recognition accuracy improved significantly. It demonstrates again that LPS_4d can effectively characterize the vertical movement.

We report gesture recognition results and compare them with several state-of-the-art methods in Table 3. Our proposed LPSNet outperforms other methods in both RGB and RGB-D sequences. When only using RGB sequences, our LSPNet achieves a recognition rate of **96.7%**, leading to 5.1% higher classification rate than MRNN [18]. This could be because the LPS feature maps are more robust to background interference and individual difference, relative to original RGB inputs. When using RGB-D sequences, our LPSNet achieves **98.7%**, even outperforming the methods additionally using optical flow information (*e.g.*, M-RNN [18] and R3DCNN [16]). This result illustrates that LPS feature maps have included more motion information than optical flow. To the best of our knowledge, this accuracy represents the state-of-the-art performance for SKIG dataset.

6. Conclusion

In this paper, we proposed a novel feature and model named LPSNet for the dynamic gesture recognition. We firstly leverage PS feature and its compressed version LPS feature into the gesture recognition task and introduced an effective method to combine RGB and depth information based on PS and LPS. Further, we introduce DropFrame to generate a set of new hand gesture samples by simulating various hand gestures with different speeds and trajectories. Incorporated with LPS feature and the DropFrame method, the performance improves significantly. Our method achieves a promising recognition rate of **96.7%** (only use RGB video) and **98.7%** (RGB and depth videos) on SKIG dataset, which outperforms the existing methods. We believe the LPS feature can be extended to the tasks that involve the analysis of finite paths and the DropFrame method can be used when data sequences are limited. Our future work will include more explorations on LPS feature and network architecture, using more challenging databases.

a	b	c	d	e	f	g	h	i	j	
107	0	0	0	0	1	0	0	0	0	**a(circle)**
0	108	0	0	0	0	0	0	0	0	**b(triangle)**
0	0	98	1	1	0	0	3	1	4	**c(up-down)**
0	0	3	105	0	0	0	0	0	0	**d(right-left)**
0	0	2	0	106	0	0	0	0	0	**e(wave)**
0	0	0	0	1	107	0	0	0	0	**f(Z)**
0	0	0	0	0	0	107	1	0	0	**g(cross)**
0	0	3	0	0	0	0	103	0	2	**h(comehere)**
0	0	2	0	0	0	0	1	103	2	**i(turnaround)**
0	0	6	0	0	0	0	0	1	101	**j(pat)**

Table 1. The confusion matrix for LPS_3d+DF.

a	b	c	d	e	f	g	h	i	j	
108	0	0	0	0	0	0	0	0	0	**a(circle)**
0	108	0	0	0	0	0	0	0	0	**b(triangle)**
0	0	103	1	0	0	0	1	0	3	**c(up-down)**
0	0	0	106	1	0	0	0	1	0	**d(right-left)**
0	0	0	1	107	0	0	0	0	0	**e(wave)**
0	0	0	0	0	108	0	0	0	0	**f(Z)**
0	0	0	0	0	0	108	0	0	0	**g(cross)**
0	0	0	0	0	0	0	108	0	0	**h(comehere)**
0	0	0	0	0	0	0	1	106	1	**i(turnaround)**
0	0	1	0	0	0	0	0	3	104	**j(pat)**

Table 2. The confusion matrix for LPS_4d+DF.

Method	RGB	Depth	Optical flow	Recognition accuracy(%)
RGGP [11]	√			84.6
MRNN [18]	√			91.6
Our LPSNet	√			**96.7**
RGGP [11]	√	√		88.7
R3DCNN [16]	√	√		97.7
MRNN [18]	√	√	√	97.8
R3DCNN [16]	√	√	√	98.6
Our LPSNet	√	√		**98.7**

Table 3. Our approach compared with state-of-the-art approaches.

7. Acknowledgement

This research is supported in part by the MSRA Research Collaboration Funds (FY16-RES-THEME-075), Fundamental Research Funds for Central Universities of China(2017MS050), GDSTP (2016A010101014, 2015B010101004, 2015B010130003, 2015B010131004, 2014A020208112, 2017A010101027), NSFC (61472144, 61673182), National Key Research & Development Plan of China (2016YFB1001405), GZSTP(201607010227, 201707010160).

References

[1] H. Boedihardjo, X. Geng, T. Lyons, and D. Yang. The signature of a rough path: uniqueness. *Advances in Mathematics*, 293:720–737, 2016.

[2] K.-T. Chen. Integration of paths–a faithful representation of paths by noncommutative formal power series. *Transactions of the American Mathematical Society*, 89(2):395–407, 1958.

[3] I. Chevyrev and A. Kormilitzin. A primer on the signature method in machine learning. *arXiv preprint arXiv:1603.03788*, 2016.

[4] F. Dominio, M. Donadeo, and P. Zanuttigh. Combining multiple depth-based descriptors for hand gesture recognition. *Pattern Recognition Letters*, 50:101–111, 2014.

[5] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1933–1941, 2016.

[6] A. Garrido. System control and rough paths (oxford mathematical monographs), 2010.

[7] B. Graham. Sparse arrays of signatures for online character recognition. *Computer Science*, 135(3410):89–90, 2013.

[8] B. Hambly and T. Lyons. Uniqueness for the signature of a path of bounded variation and the reduced path group. *Annals of Mathematics*, pages 109–167, 2010.

[9] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *Computer Science*, 3(4):pgs. 212–223, 2012.

[10] S. Lai, L. Jin, and W. Yang. Toward high-performance online hccr: A cnn approach with dropdistortion, path signature and spatial stochastic max-pooling. *Pattern Recognition Letters*, 89:60–66, 2017.

[11] L. Liu and L. Shao. Learning discriminative representations from rgb-d video data. In *IJCAI*, volume 4, page 8, 2013.

[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[13] T. Lyons, M. Caruana, and T. Lévy. Differential equations driven by rough paths. *Ecole dEté de Probabilités de Saint-Flour*, 34:1–93, 2004.

[14] T. J. Lyons and N. Sidorova. Sound compression: a rough path approach. In *Proceedings of the 4th international symposium on Information and communication technologies*, pages 223–228. Trinity College Dublin, 2005.

[15] Y. Ming, Q. Ruan, and A. G. Hauptmann. Activity recognition from rgb-d camera with 3d local spatio-temporal features. In *Multimedia and Expo (ICME), 2012 IEEE International Conference on*, pages 344–349. IEEE, 2012.

[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*, pages 4207–4215, 2016.

[17] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[18] N. Nishida and H. Nakayama. Multimodal gesture recognition using multi-stream recurrent neural network. In *Pacific-Rim Symposium on Image and Video Technology*, pages 682–694. Springer, 2015.

[19] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[20] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016.

[21] J. Reizenstein. Calculation of iterated-integral signatures and log signatures. Technical report, Centre for Complexity Science, University of Warwick, 2016.

[22] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[23] C. Reutenauer. Free lie algebras. *Handbook of algebra*, 3:887–903, 2003.

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

[25] Wikipedia. *Necklace polynomial*. 2015. `https://en.wikipedia.org/wiki/Necklace_polynomial`.

[26] W. Yang, L. Jin, and M. Liu. Chinese character-level writer identification using path signature feature, dropstroke and deep cnn. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 546–550. IEEE, 2015.

[27] W. Yang, L. Jin, Z. Xie, and Z. Feng. Improved deep convolutional neural network for online handwritten chinese character recognition using domain-specific knowledge. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 551–555. IEEE, 2015.

[28] W. Yang, T. Lyons, H. Ni, C. Schmid, L. Jin, and J. Chang. Leveraging the path signature for skeleton-based human action recognition. 2017.