Point based Gesture Recognition Techniques

Varun Sharma
Department Computer Science
Liverpool John Moores University
Liverpool, UK

Hoshang Kolivand
School of Computer Science and Maths
Liverpool John Moores University
Liverpool, UK
h.kolivand@ljmu.ac.uk

*Shiva Asadianfam**
Faculty of Electrical & Computer Engineering, Qom University of Technology, Qom, Iran
Department of Computer Engineering
Islamic Azad University Qom Branch, Qom, Iran
Sh_asadianfam@yahoo.com

Dhiya Al-Jumeily
School of Computer Science and Maths
Liverpool John Moores University
Liverpool, UK

Manoj Jayabalan
School of Computer Science and Maths
Liverpool John Moores University
Liverpool, UK

***Abstract*—Gesture recognition is a computing process that attempts to recognize and interpret human gestures through the use of mathematical algorithms. In this paper, we describe Point Based Gesture Recognition and Point Clouds nearest neighbors and sampling. Also, we explore these techniques with previous studies.**

Keywords— *Point based gesture recognition, Point Clouds nearest neighbors and sampling, Gesture recognition.*

I. INTRODUCTION

In this century, backed up by fast changing and improved information technologies and communication systems, new systems are being developed which helps provide better user experience and replicate real world scenarios. One such field which is continuously changing, and getter better is gesture recognition. New methodologies are being developed to enhance the recognition and Point Cloud is an effort towards same. With launch of cheap depth sensors like Kinect, Intel RealSense etc. lot of time and investment has been put to derive an algorithm which can help solving the gesture recognition, use of Point Cloud is a step towards it. However, before the point clouds be used , it is important to assess the quality of point clouds [1].

Point clouds are datasets that represent objects or space. These points represent the X, Y, and Z geometric coordinates of a single point on an underlying sampled surface. Point clouds are a means of collating many single spatial measurements into a dataset that can then represent a whole. When colour information is present, the point cloud becomes 4D. Point clouds are most generated using 3D laser scanners and LiDAR (light detection and ranging) technology and techniques. Here, each point represents a single laser scan measurement. These scans are then stitched together, creating a complete capture of a scene, using a process called 'registration'. Conversely, point clouds can be synthetically generated from a computer program.

Point clouds have information hidden such that they can be used precisely describe the geometric structure and help in finding the distance between object surfaces, these informational elements provide necessary details which are useful in gesture recognition. Since point cloud is not a visual rather geometric representation of an object, it given immense possibility to build a concept which can be used for all types of gesture recognition, however our focus in this study is limited to hand gestures. 3D representation of an object help extracts both motion and structure feature and gives a possibility to derive methodologies which can help tune the processing. Below Figure 1, is an example of how point clouds looks with four different gestures [2].

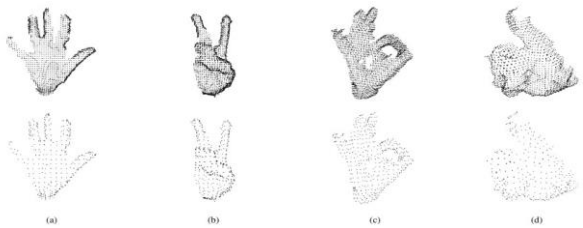

Figure 1. Gesture point clouds: (a) Open hand pose. (b) Peace pose. (c) Ok pose. (d) Like pose [2]

This paper is structured as follows. In Section 2, the related studies on point based gesture recognition are described in detail. In Section 3, we describe point clouds nearest neighbors and sampling. Finally, the general conclusion of this paper is explained in section 4.

Point clouds nearest neighbors and sampling

II. POINT BASED GESTURE RECOGNITION

Above section, we learnt use of Graph based methodologies in gesture recognition, idea was very clear and was to make use of joints in human body and derive methods which are efficient. Different other techniques were then applied on graph to make the recognition generic.

Another technique that came into existence was use of points, meaning representing the body part (or whole body depending on use case) as points and then deriving methodologies.

Li et al., [3], 3D points were used from the depth maps to classify the action. During this time there was no public

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

dataset (benchmark) available and hence a dataset having 20 actions was collected for study. In total 7 subjects performed each action 3 times to create a dataset of 4020 action samples with depth map of 640X480. The depth image was first down sampled to reduce computational cost, the sampled 3D points were then allocated to XY, YZ and XZ projections. For all conducted experiments, , training samples were clustered using the Non-Euclidean Relational Fuzzy (NERF) C-Means and the dissimilarity between two depth maps was calculated as the Harsdorf distance between the two sets of the sampled 3D points as shown in Figure 1 [3].

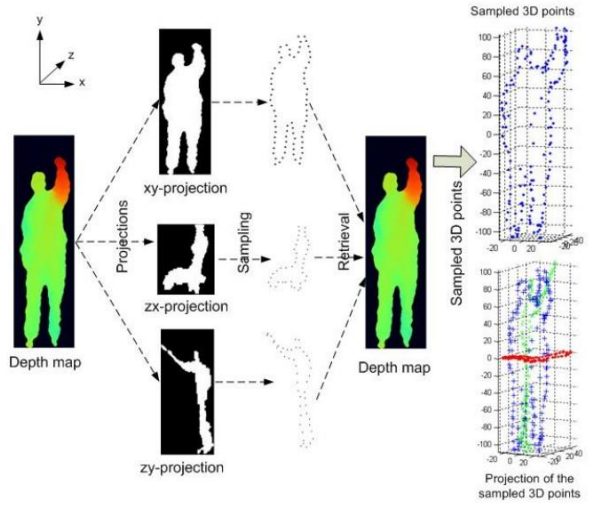

Figure 1. Bag of 3D points from depth image[3]

To evaluate the methodology, the datasets were divided into 3 parts, details of datasets and accuracy are listed in Table 1.

Table 1. NERF based accuracy matrix

		Train-Test split – 1/3-2/3
AS1	Simple action of one kind	89.5%
AS2	Simple action of another kind	89.0%
AS3	Complex actions	96.3%
Overall		91.6%

This study also observed differences in accuracies when different quadrant of images was removed (images was divided in 4 quadrants), it was concluded that part which had least significance is removed then accuracies increases and vice-versa, this study hence paved a path to concentrate on points which are significant for that recognition. It was found that 3D points offer a lot, and more research were required to get to better and generic methods.

Vemulapalli et al. [4], a new methodology which represented action as a curve in the lie group was proposed. As there are difficulties in classifying the curves in lie group, mapping of curves in lie group was done to the lie algebra, which is nothing but a vector space and hence easy to model for classification. The vectors were then used with DTW, Fourier temporal pyramid representation and linear SVM. Below Figure 2 shows curve in the lie group and Figure 3 shows methodology used [4].

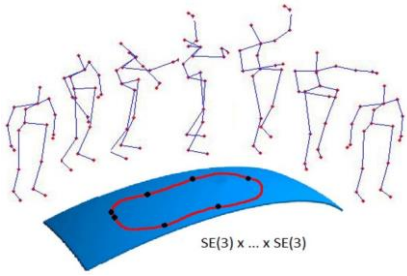

Figure 2. Representation of an action (skeletal sequence) as a curve in the Lie group SE(3) ×. .. × SE(3) [4]

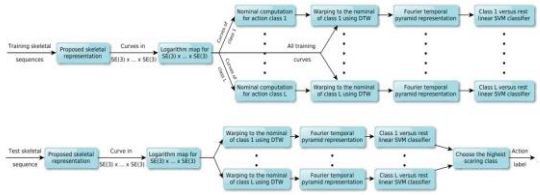

Figure 3. Training and Testing pipeline using Lie Group[4]

The model was evaluated on many datasets and found to be effective, below Table 2 shows the accuracies.

Table 2. Lie group-based accuracy matrix

Datasets	Accuracies
MSR-Action3D	92.46
UTKinect	97.08
Florence3D	90.88

Though the model results were good, but usage was limited due to following reasons:

- Dynamic identification of body part used for action recognition was unavailable
- Actions performed were by a single person and hence accuracies cannot be generalized.

Many research were being done during these years, focused on cloud based points, taking the studies further, in 2018 [5], Hand PointNet was proposed. This methodology was different from other CNN based (methods using CNN had issue containing the space and time complexities which can grow cubically) from the fact that this method directly processed the 3D point cloud which was formed from the visible hand surface for pose regression. To improve the

efficiency of the model, neighboring finger points were also considered and used in the modelling and referred to as fingertip refinement network. Below Figure 4 details out the architecture used in Hand PointNet[5].

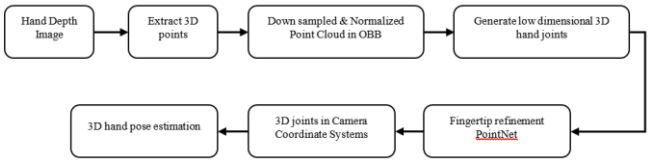

Figure 4. Hand PointNet architecture

OBB – It helps solve the large variation in hand orientation, by transforming the original hand point cloud into a canonical coordinate, making global orientation consistent.

Fingertip refinement PointNet – Takes k-nearest neighboring points of fingertip location and outputs refined 3D fingertip locations, with this we can

- Decrease fingertip estimation error
- KNN will not change even if there is deviation of fingertip location from ground truth.

The methodology was compared against many other methods, and it was found that PointNet was superior to other as shown in Figure 5.

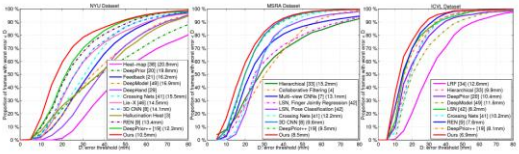

Figure 5. Comparison with other techniques (Our stands for PointNet) [5]

Point cloud at this stage has become one of most studied and many methodologies were proposed. In continuation on how point clouds can further be utilized, in 2019[6], Meteornet was proposed. Specialty of this proposed method was that it directly consumes the dynamic sequences of point clouds and learns both local and global features and this learning was utilized to solve problems like classification, segmentation etc.

MeteorNet is based on a novel neural network module called Meteor module, specialty of Meteor is that it takes point clouds and learn features of each point from it by aggregating the spatiotemporal neighborhoods. Design is such that it can learn from previous features by stacking modules over one another. Finally, the stacked modules can capture information from larger neighborhood. However, problem with this method was its inability to properly determine the spatiotemporal neighborhoods while the object is in motion. To address this, under MeteorNet two methods were proposed, shown in Figure 6[6].

- Direct grouping – Directly increases the grouping radius over time
- Chained grouping – Tracks the motion of the object and uses offline estimated scene flow to construct the neighbourhood.

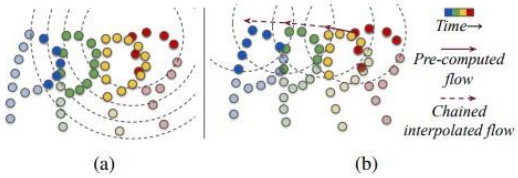

Figure 6. (a) direct grouping; (b) chained-flow grouping[6]

MeteorNet evaluation was done on many datasets, and below Table 3 shows the accuracy achieved on classification dataset (limited due to scope of this study)

Table 3. Classification accuracy on MSRAction3D (%)

# Of Frames	Accuracy
4	78.11
8	81.14
12	86.53
16	88.21
24	88.50

Another methodology, FlowNet3D [7] based on deep neural network was proposed which is capable of learning flow of scene using point clouds in an end-to-end fashion. The architecture is such that it can learn hierarchical feature from point clouds and flow embeddings representing point motions simultaneously.

FlowNet3D utilizes two consecutive flow frames let's call then point cloud 1 and point cloud 2, the network inside FlowNet3D is capable of estimating a translation flow vector for every single point in point cloud 1 (with help of point cloud 2) thereby indicating the motion(flow) between two frames as shown in Figure 7 [7].

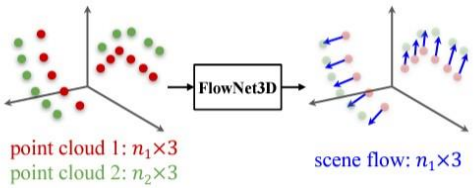

Figure 7. End-to-end scene flow estimation from point clouds[7]

FlowNet3D has 3 building blocks as explained below on overall how it works.

- Point Feature Learning – Since point clouds are irregular and order less, traditional convolutions are not effective and hence a new architecture PointNet++ [8] was used which is capable of learning hierarchical features.
- Point Mixture – Mixing of two-point clouds is not an easy task due to viewpoint shift and occlusions. Idea was based on finding displacement between two points at time t and t+1 and using weights to achieve the end state.

- Flow Refinement – Here the flow embedding found using Point Mixture are up-sampled to the original points and helps in propagating the points features so that they can be learnt.

To evaluate the model, end point error (EPE = average distance between estimated flow vector and ground truth) and flow estimation accuracy (ACC = portion of estimated flow vector which are below EPE threshold) was used.

Results when evaluated on FlyingThings3D were as follows:

- EPE = 0.1694
- ACC (<5%) = 25.37%
- ACC (<10%) = 57.85%

There were multiple research using point cloud and various different methods were applied, another such techniques which utilizes point cloud based on LSTM was proposed in 2020 [9]. In this methodology idea was to propagate the information from past to future while retaining the spatial structure. This method combines past state information of neighboring points with current state and with the help of weight-shared LSTM. One of problems of all methods described was that they were unable to capture long term relationships and were basically short term. Ideally a point at time t-1 will have representation at time t but, it won't be easy to find the exact point and that is where this method has been useful. It used the neighboring points to find the state.

PointLSTM was found effective on datasets like SHREC'17 and NVGestures. On SHREC'17, on 14 gesture it reported max of 95.9 % and on 28 gestures, 94.7%. On NVGesture it was found to be 87.9% accurate.

Another LSTM based approach , Two Stream LSTM [10] was proposed which focused on learning salient spatial features using CNN and finally using LSTM to map the temporal relationship. Proposed techniques showed that the output of both layers combined has greater recognition ability than using either stream alone. The results showed that a fully connected layer output can be used as a tool to direct the LSTM to the important parts of the convolutional feature sequence. The proposed mechanism achieved 94.6 %, 99.1%, 69.0 % accuracy on UCF Sport , UCF11 and jHMDB datasets respectively.

Another study which also based on the fact that the single-stream models are not adequate for capturing both fine-grained local posture variations and global hand movements was proposed [11]. The idea was to decouple learning of local and global features using a dual-stream model, the features were then fused to a LSTM layer where temporal learning was done. Through leveraging the complementary benefits of raw point clouds and BPS-based representations, proposed framework explicitly learns global position and local posture features as shown in Figure 8 . The approach was computationally more efficient, but the accuracy could have been increased by adding an extra module for low-level spatiotemporal feature extraction.

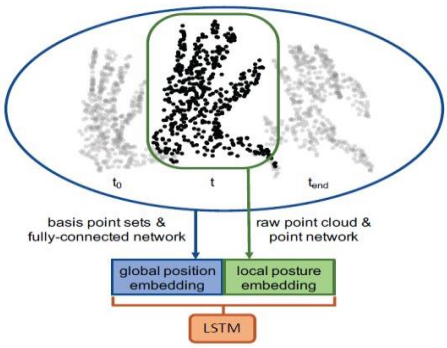

Figure 8. 2S model to capture both global position and local posture features[11]

The method was found to be highly effective on datasets like SHREC'17 and achieved 96.1 and 95.2 % accuracy on 14 and 28 Gestures respectively.

III. Point Clouds Nearest Neighbors and Sampling

Efficiently finding nearest neighbor in point cloud is an important task, as we have studies in other research papers that information about a point can be derived using its neighbors, however more the effective method is less the noise that will get propagated from one frame to another. Another aspect is to sample the 3D point clouds, as the data is huge we also need to make sure that we do not oversample and same time simplify the 3D point cloud data.

Representing the point clouds is an important task and a study was done in 2010, [12] and a simplified way to represent 3D point clouds were proposed. Idea was to integrate both the feature parameters and unform spherical sampling. The first thing we do is establish parameters comprised of the average of neighboring point's average distance, angle formed between point and neighboring points and point curvature, another step is then is to define the feature threshold by calculating density of 3D points, this helps in differentiating between feature and non-feature points. Uniform sampling is then applied over the non-feature points. Figure 9 shows feature and non-feature points. Spherical parameterization techniques can be used for sampling[12].

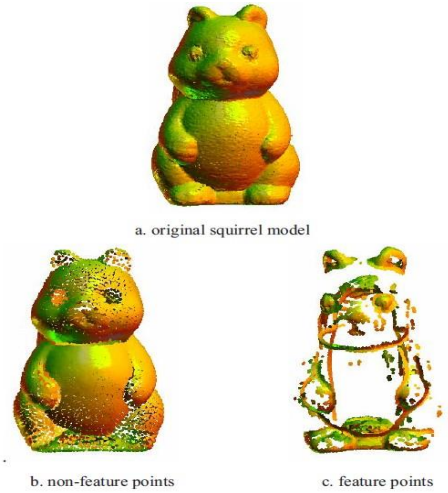

Figure 9. Feature and Non-Feature points[12]

The benefit achieved by this method was that it shows ways to reduce the 3D point cloud data and same time retaining the sharp information without loss.

Continuing with the previous studies on how to effectively create optimized 3D points was proposed in 2011 using k nearest neighbor [13, 14]. 3D point clouds need to be processed so as to construct high level information beginning from cartesian coordinates. The process of processing point clouds usually entails removing residual noise, adjusting the sampling rate, estimating the points' normal, and/or tessellation. Various operations in the cloud require computing the k-nearest neighborhoods (KNN) of each point. Angelo and Giaccari proposed knn search based on new data structure using typical space division approach. It was comprised of two basic steps :

1. Data Structure creation – Helps reducing computational cost
2. Nearest-neighbour search

With success in using KNN on point clouds another approach, an improved method based on dimension reduction and sorting was proposed for the K-nearest neighbors algorithm [14]. PCA was introduced to analyze the spatial distribution of point clouds data main directions. In the next step, turn the main directions so they match the X, Y, and Z coordinates, sort the point cloud data in the three coordinate axes, and find the position of the query point. Next the distance between the query point and its neighbors by extracting neighbor points in proportion to the three sorted point cloud data sets. In order to determine the nearest neighbors, sort the distance by the first k points found. The proposed algorithm helped in reducing point to point distance calculation time efficiently. The KNN algorithm was improved, resulting in a major time savings when solving point cloud normal vectors, reconstructing surfaces, and similar operations.

Exploring feature of neighboring point was possible based on studies done, however the existing methods suffered from ambiguous feature, mainly junction regional points. An approach using segmentation scores of neighboring points was proposed in 2019 [15]. The module proposes a method for improving 3D point cloud segmentation scores by utilizing attention-based refinement, which can be easily integrated with existing networks. This study allowed possible to focus on a particular region to extract required feature which can be used for modeling.

IV. CONCLUSION

The general definition of gesture recognition is the ability of a computer to understand gestures and execute commands based on those gestures. In this paper, we describe Point Based Gesture Recognition and Point Clouds nearest neighbors and sampling. Also, we explore these techniques with previous studies.

REFERENCES

1. Alexiou, E. and T. Ebrahimi. *Point cloud quality assessment metric based on angular similarity*. in *2018 IEEE International Conference on Multimedia and Expo (ICME)*. 2018. IEEE.
2. Apostol, B., C.R. Mihalache, and V. Manta. *Using spin images for hand gesture recognition in 3D point clouds*. in *2014 18th International Conference on System Theory, Control and Computing (ICSTCC)*. 2014. IEEE.
3. Li, W., Z. Zhang, and Z. Liu. *Action recognition based on a bag of 3d points*. in *2010 IEEE computer society conference on computer vision and pattern recognition-workshops*. 2010. IEEE.
4. Vemulapalli, R., F. Arrate, and R. Chellappa. *Human action recognition by representing 3d skeletons as points in a lie group*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014.
5. Ge, L., et al. *Hand pointnet: 3d hand pose estimation using point sets*. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.
6. Liu, X., M. Yan, and J. Bohg. *Meteornet: Deep learning on dynamic 3d point cloud sequences*. in *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019.
7. Liu, X., C.R. Qi, and L.J. Guibas. *Flownet3d: Learning scene flow in 3d point clouds*. in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019.
8. Qi, C.R., et al., *Pointnet++: Deep hierarchical feature learning on point sets in a metric space.* Advances in neural information processing systems, 2017. **30**.
9. Min, Y., et al. *An efficient pointlstm for point clouds based gesture recognition*. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.
10. Gammulle, H., et al. *Two stream lstm: A deep fusion framework for human action recognition*. in *2017 IEEE winter conference on applications of computer vision (WACV)*. 2017. IEEE.
11. Bigalke, A. and M.P. Heinrich. *Fusing Posture and Position Representations for Point Cloud-Based Hand Gesture Recognition*. in *2021 International Conference on 3D Vision (3DV)*. 2021. IEEE.

12. Wang, L., J. Chen, and B. Yuan. *Simplified representation for 3D point cloud data*. in *IEEE 10th International Conference on Signal Processing Proceedings*. 2010. IEEE.
13. Di Angelo, L. and L. Giaccari, *An efficient algorithm for the nearest neighbourhood search for point clouds.* International Journal of Computer Science Issues (IJCSI), 2011. **8**(5): p. 1.
14. Li, D. and A. Wang. *Improved KNN algorithm for scattered point cloud*. in *2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*. 2017. IEEE.
15. Zhao, C., et al. *Pooling scores of neighboring points for improved 3D point cloud segmentation*. in *2019 IEEE International Conference on Image Processing (ICIP)*. 2019. IEEE.