A machine learning approach for control in Augmented Reality

Junxiao Shen[a,b]

[a] *1Department of Engineering, University of Cambridge, UK*
[b] *js2283@cam.ac.uk*

Keyword: Human-Computer Interaction, Deep Learning, Machine Learning, Gesture Text Input, Augmented Reality, finger segmentation

Abstract: Augmented Reality (AR) has attracted much popularity across the industry, however, due to the expensive hardware and unintuitive control. It has not been as popular as mobile phones among normal users. A lot of research has been put into the development of the control system in AR. Machine learning technique can be explored here to be applied to this control system.

1. Introduction

Deep learning has been developed tremendously on different fields due to its high efficiency and accuracy. [1]. Both speech recoginition[2] and image recognition [3] has had great improvement by using deep learning. The reason of this breakthroughn is due to the dramactic incress of the data and an explosion of the computational power, therefore, neural networks can be used extensively and efficiently. Within augmented reality control, there are many problems to be addressed from a high level such as an efficient and intuitive user interface to low level like image segmentation and registering. The machine learning method is a strong and convincing way to boost the performance of the control in AR because numerous data is been transmitted between the environment and the hardware. Therefore, different machine learning methods can tackle various problems. natural language processing[4] [5], speech recognition and syhthesis[6] and computer vision [7] are all very important and can be explored to make the control system better. Therefore, in this research proposal, there are three questions need to be specified. Firstly, how to seamlessly do text input in AR, Secondly, how to do image segmentation of the figures so that the sensor can track different parts of the hand. Each question requires different approaches and will be tackled individually.

1.1. Text input

Text input is the most effective way to interact with computer systems so far and different methods are used in different devices. For example, QWERTY keyboard has been the most popular text input keyboard in personal computers whereas T9 predictive text was originally invented for users in the mobile phones considering the small user interface, and later on, after the screen gets bigger and smartphones came onto the market, QWERTY keyboard started to become a very effective method for text input in smart mobile phones.
In Augmented Reality and in the past few decades, keyboards and mice have played an important role in human-computer interaction. However, due to the rapid development of hardware and software, a new HCI method is needed. Due to the inaccuracy for fingers to do tap actions, the swipe method is now considered to be a more effective and more intuitive way to input texts in AR. However, in QWERTY keyboard, due to the offset between the actual movement track and the sensed movement track of the fingers, there will be a difference between the distribution of the likelihood of the word given the gesture and the actual word we intended to input.
Therefore, I came up with two ideas s to tackle this problem. One is to use T9 keyboard so to decrease the offset and the other is to use a machine learning method to minimise the offset in the QWERTY keyboard. The advantage to having a T9 keyboard is that it only has 9 blocks so the offset might be small enough compared to the actual scale of the gesture tracking. However, the difficulty is that there is not a good enough gesture model for the T9 keyboard to have a good prediction on the word based on the very limited gestures since there are only nine blocks. Furthermore, it is not a common text board for users now and it is against the intuition of users. The advantage of using QWERTY keyboard is that it has an established mature gesture model together with the language model to give a good posterior distribution over the word. Therefore, considering the pros and cons of two different designs of the text board, we decided to use the QWERTY keyboard. Due to the hardware

inability to accurately sense the trace of the figure, there is a big displacement between the trace of the figure tips we see in the user interface in HoloLens and the actual data input of the trace. Therefore, a recognition model to map the trace to word or sentences should be invariant of the starting position of the keyboard. Therefore, Deep learning can be used to train this recognition model. Innovated from speech recognition, since speech is a 1D temporal data and the trace is a temporal and spatial data, therefore, the network models and state-of-art algorithms for speech recognition can be used to innovate us to design and train the neural networks for the trace recognition.

1.2. Finger Segmentation

It is well known that vision-based gesture recognition technology is an important part of human-computer interaction. There are two types of gestures: hand gesture and body gesture. These gestures can also be divided into static gestures and dynamic gestures. Gesture recognition can be of great importance in AR control and thus figure segmentation becomes a problem that needed to be tackled. Hand gesture recognition can be used in many human computer interactions such as TV control [8]. Figure 1 shows the workflow of the hand gesture recognition. Among the different components, Finer and palm segmentation remains a challenging and interesting area since the performance can be greatly affected by the environment and the movement of the hand. This makes it more difficult to realise real time and efficient hand segmentation in AR. Many different classifiers such as SVM (support vector machine), HMM (hidden markov model), CRF(conditional random field) can be trained to discriminate hand gestures. However, again, it still can not meet the standard to do a smooth recognition in AR. Therefore, deeper neural network can be used to explore the underlying features and may lead to a more accurate and even real time finger segmentation.

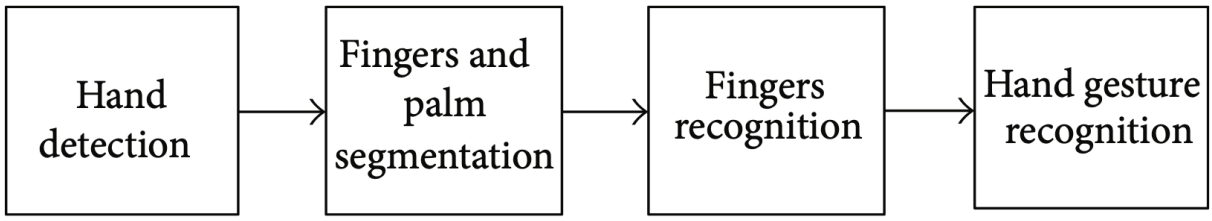

Figure 1: Hand gesture recognition work flow

2. Research Plan

2.1. Text input

To use deep learning, numerous data should put into the neural network. However, it is not possible to collect this amount of gestures. Therefore, we need to synthesis artificial data on our own. Furthermore, to have the right feeling of the gestures, an experiment to study the user habit for gesture input should be used to investigate how the user will input the gestures. Figure 2 shows the overall plan. it is a component diagram and it is used to visualise and help us to have a better idea of the working process.

2.1.1. First stage

Firstly. the OptiTrack system which is a motion capture system that can accurately track the coordinates of the markers including HMD (Head Mounted Display), fingertips. The reason why we use OptiTrack is that the HoloLens inbuilt sensor is not good due to the huge latency and displacement of the actual figure and the data whereas OptiTrack is the world's most accurate and wide-area VR trackers. This leads to an ultra-low latency and better smooth tracking for HMD and fingertips. Then the data will be transferred to pc by NatNet which is an SDK for streaming motion tracking data across networks. Then the data received on the pc will be transformed to the keyboard frame since the original data is in the world frame. The trace will be compared with the previous setpoints, once it approaches the region of the points, the pc will send a query with a world to the HoloLens and it will output the word for the user.

2.1.2. Second stage

We will thus use the data we collected from the users' experiment to synthesis the data together with the data for the gaze collected from the experiment since we also believe that the trace of the gaze can be relevant to the final output as well. And the synthesised data will be from used to train the neural network and the trained model will thus replace the recognition component in Figure 2. The reason to synthesis the data is that unlike speech recognition, the recorded labelled data can be accessed quite easily nowadays since people can pay others to read and label whereas in our project. The dataset is limited due to /the lack of funding to hire a lot of people to generate the traces and due to the short time frame considering it is an only one-year project.

2.1.3. Third Stage

We will do another users experiment of the established system and test the results and compare them with the state-of-art text input system. The user experience will still be carried out in the OptiTrack system.

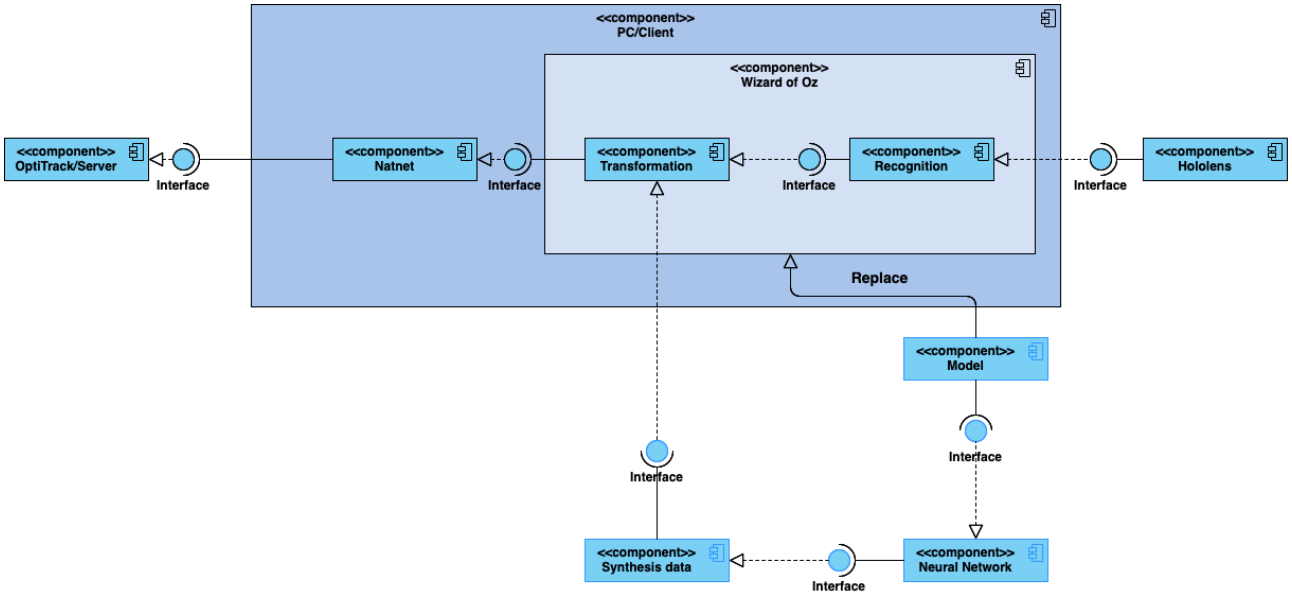

Figure 2: Component diagram for the development

2.2. Detailed development

2.2.1. User experiment design

We will use the OptiTrack to track the coordinates of the keyboard, figure tips and the HMD in the world frame. Natnnet is used to get the data from OptiTrack which is the server to send the data to the client which is the PC to run the script. A wizard of oz experiment will be designed so to mimic the perfect software for gesture input. Two data kinds will be used in the future to train the neural network, they are the trace of the figure tip and the trace of the gaze both in the keyboard frame. Firstly, a transformation software should be written to transform the coordinates of the HMD and fingertips from world frame to keyboard frame. Secondly, a recognition script should be written to recognise whether the figure tip has hit the region of the previous setpoints if they are within the region in the right order, then a word will be sent to the HoloLens. We will see whether we should use 3D data of the fingertips and gaze straight away or project the 3D data onto the keyboard to make it 2D data since theoretically the data on the direction that is perpendicular to the keyboard should be independent of the output world but realistically speaking, this might not be the case. This will be explored later.

2.2.2. Data synthesis

We will generate data based on the data we obtained from the user experiments. There are many methods that i can use to generate the data. There are many existing generative models, one of the most popular ones is Generative Adversarial Networks[9] which use a generator produces training samples from the vector of random noise and use discriminator to classify whether a training sample is real or produced by the generator. However, this model is mainly used in computer vision but this can still be of usefulness to our project. More relevant generative models in Text to Speech can be very important as well such as WaveNet[10]. As can be seen from the workflow diagram in Figure 3, a quality check needs to be conducted to see whether the synthetic data is similar

to real data. Different methods can be used such as calculating the KL divergence like TSNE (T-Distributed stochastic neighbour embedding) which learn low-dimensional embedding of feature vector by minimising KL between similarities in both spaces[11]. Classifier approach can also be used so that if it fails to classify the real data and synthetic data then it is a good indicator.

2.2.3. Design and Train neural network

Different neural networks can be used. I would start from Recurrent Neural Networks (RNN) first since the trace of fingertips is a 3d spatial and temporal signal and this can be in analogy to speech recognition. Furthermore, we could also try to combine CNN and RNN to see whether it will lead to better results. Transfer learning can also be used since the fine-tuning train ml model with synthetic data in first iterations and the with real data in further iterations. We should also be careful that we shouldn't use synthetic data in a test set since it can have a corresponding pair of real data which may lead to a dangerous result.

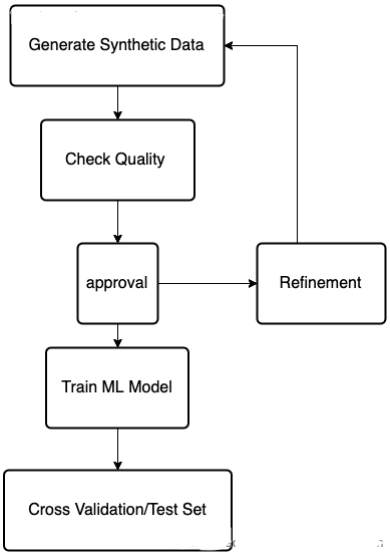

Figure 3: work flow for second and third stage

2.2.4. Risk Management

There are many risks comes from users, technologies and equipment. However, the biggest challenge and the risk is to generate the data and train the neural network due to several reasons.

- the data is unlike images and speech, the 3D or 2D spatial and temporal data will be not intuitive to us and having a good intuition and understanding of the data will be very important towards the training of the model and designing of the model.
- Mapping 3D or 2D signal to text is a very innovative field so not too much work has been done before and it is not sure if the previous networks for computer vision and speech would be particularly useful.
- The quality control step is very important and we might getting synthesised data that is not similar to the real data.

3. Hand detection and Finger segmentation

In the field of existing visual-based human hand detection, there are mainly feature detection methods, template matching methods, image difference methods, and the like. In the hand detection method, most of the hand skin color[12][13][14], palm texture, and hand shape [15][16][**?**] are used as detection features. Due to the complex background (the picture contains a large number of skin-like areas), the illumination transformation, the complex shape of the human hand, and the occlusion interference, there is no particularly stable and mature detection method for the hand. With the development of depth cameras (Kinect sensors, Xtion sensors provided by ASUS, etc.), depth information is widely used in hand detection, and the application of depth information improves the hand detection rate of the manual detection system, but There are still problems that make it difficult to distinguish between the palm of your hand and the type of hand. In the case of occlusion in the hand,

rapid movement of the hand, and mutual contact between the hand and the hand, the hand and the face, the human hand detection system still has many deficiencies. In recent years, deep learning has been widely used in the field of object detection. The use of deep convolution networks in hand detection systems [**?**][**?**] has improved the accuracy and robustness of hand detection. However, the focus of research on hand detection in video streaming is mainly image detection. The temporal and spatial correlation of the hand is not fully utilised, and the detection caused by rapid movement, occlusion and new hand is not well solved. The way to solve this may be to use a Hypernet network adds the multi-scale feature extraction module to the object detection deep convolutional neural network Faster R-CNN, which improves the network's ability to detect small objects. The HyperNet network consists of three parts: a multi-scale feature extraction module, a region generation module, and an object detection module. The multi-scale feature extraction module extracts the image features through the convolutions network, and normalizes the features extracted by different convolutional layers to the same scale through Max pooling down-sampling and Deconv upsampling, and then uses local response normalization processing and connected together to form Multi-scale features (Hyper features).

4. Conclusion

The first problem which is to use gesture for text input takes a lot of space in this proposal is because this is a very innovative method. Therefore, I need to be very clear about the question statement and the clear overall plan for this problem. For the second problem, what I want to do is to improve the state-of-art finger segmentation algorithms.

References

[1] *Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436 EP –.*
URL `https://doi.org/10.1038/nature14539`

[2] *G. E. Dahl, D. Yu, L. Deng, A. Acero, Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition, IEEE Transactions on Audio, Speech, and Language Processing 20 (1) (2012) 30–42. doi:10.1109/TASL.2011.2134090.*

[3] *A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: F. Pereira, C. J. C. Burges, L. Bottou, K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 25, Curran Associates, Inc., 2012, pp. 1097–1105.*
URL `http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf`

[4] *M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 2227–2237. doi:10.18653/v1/N18-1202.*
URL `https://www.aclweb.org/anthology/N18-1202`

[5] *J. Howard, S. Ruder, Universal language model fine-tuning for text classification, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 328–339. doi:10.18653/v1/P18-1031.*
URL `https://www.aclweb.org/anthology/P18-1031`

[6] *W. Xiong, L. Wu, F. Alleva, J. Droppo, X. Huang, A. Stolcke, The microsoft 2017 conversational speech recognition system, in: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, pp. 5934–5938. doi:10.1109/ICASSP.2018.8461870.*

[7] *M. D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), Computer Vision – ECCV 2014, Springer International Publishing, Cham, 2014, pp. 818–833.*

[8] *C. Keskin, F. Kıraç, Y. E. Kara, L. Akarun, Real time hand pose estimation using depth sensors, in: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), 2011, pp. 1228–1234. doi:10.1109/ICCVW.2011.6130391.*

[9] *I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative Adversarial Networks, arXiv e-prints (2014) arXiv:1406.2661arXiv:1406.2661.*

[10] *A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, K. Kavukcuoglu, WaveNet: A Generative Model for Raw Audio, arXiv e-prints (2016) arXiv:1609.03499arXiv:1609.03499.*

[11] *L. van der Maaten, G. Hinton, Visualizing data using t-SNE, Journal of Machine Learning Research 9 (2008) 2579–2605.*
URL `http://www.jmlr.org/papers/v9/vandermaaten08a.html`

[12] *C. Li, K. M. Kitani, Pixel-level hand detection in ego-centric videos, in: 2013 IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3570–3577. doi:10.1109/CVPR.2013.458.*

[13] *S. Bilal, R. Akmeliawati, M. J. Salami, A. A. Shafie, Dynamic approach for real-time skin detection, J. Real-Time Image Process. 10 (2) (2015) 371–385. doi:10.1007/s11554-012-0305-2.*
URL `http://dx.doi.org/10.1007/s11554-012-0305-2`

[14] *M. Aziz, J. Niu, X. Zhao, J. Li, K. Wang, Using novel shape, color and texture descriptors for human hand detection, 2014, pp. 150–157. doi:10.1109/IBCAST.2014.6778138.*

[15] *A. Kumar, D. Zhang, Personal recognition using hand shape and texture, Image Processing, IEEE Transactions on 15 (2006) 2454 – 2461. doi:10.1109/TIP.2006.875214.*

[16] *M. Bhuyan, K. F. MacDorman, M. K. Kar, D. R. Neog, B. C. Lovell, P. Gadde, Hand pose recognition from monocular images by geometrical and texture analysis, Journal of Visual Languages Computing 28 (2015) 39 – 55. doi:https://doi.org/10.1016/j.jvlc.2014.12.001.*
URL `http://www.sciencedirect.com/science/article/pii/S1045926X14001566`