2016 International Conference on Virtual Reality and Visualization

3D Hand Gesture Coding for Sign Language Learning

Yi Ji, Chunping Liu, Shengrong Gong
School of Compute Science and Techonology
Soochow University
Suzhou, China
{jiyi, cpliu, shrgong}@suda.edu.cn

Weidong Chen
Suzhou Keda Technology Co. Ltd
Suzhou, China
wdchen@kedacom.com

***Abstract*—Hand gesture and sign language are important in facilitating non-verbal communication between human beings. This paper proposes a new approach to code 3d hand gestures for presentation and recognition. The coding is based on 3D hand and finger structure extracted from Leap Motion Controller (LMC). From the 3D coordinates tracked by device, shapes, positions or distances between finger tips are coded and normalized according to a set of rules. This coding process can be completed manually or semi-automatically. The recognition stage is automatic.**

In experiments, our system applied automatic coding and recognition on a subset of Chinese sign language, which is a special set of meaningful hand gestures. Integrated in a Unity application, the method can recognize learned gestures quickly and accurately. For unknown gestures, the system also provided interactive learning method to further extend recognizable alphabet signs.

Keywords—Hand Gesture Coding; Leap Motion, Sign Language, Interactive Learning

I. INTRODUCTION

Non-verbal language and communication is widely employed for both deaf-mute and normal people. For example, through hand signs, information can be conveyed visually and quickly between two friends without a common language. How to interpret these signs automatically and accurately, raises an interesting issue for computer vision researchers. Nowadays, various studies tried various sensors to build recognition systems

Traditional sign recognition is normally based on special-designed gloves or digital video cameras [1]. Obviously, data gloves are not practical and economical for wide use. Though for most 2D video based recognition systems, lighting, occlusion and surrounding environment creates difficulties in efficient handling. Recently, depth sensors such as Leap Motion [2], Microsoft Kinect [3], or ToF (time of flight) cameras, are employed to capture 3D movements. In this paper, we will make use of Leap Motion Controller (LMC) to track hands and interpret language signs.

As a 3D hand tracker device, LMC (Fig.1) is used as a cheap and easy-use 3D sensor to track the hand and provide the information of finger's joints and their movements. Building from standard tracking and for better VR Visulalizer, we exploit Leap Motion Core Unity to smooth the integration of device and Augmented Reality scenes.

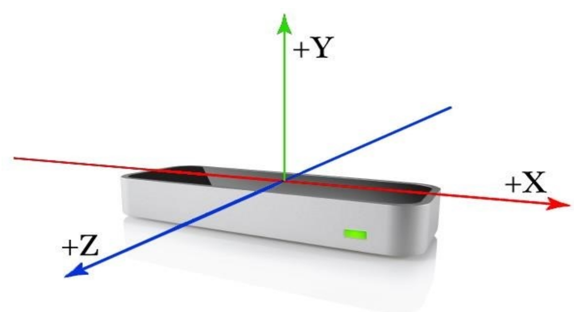

Fig. 1. Leap Motion device and its coordinates

II. FINGER POSE CODING

LMC device tracks hands in camera's field of view and maintains the inner models for them. In this model, all fingers contain 4 bones from base to tip. As thumb does not have a base metacarpal bone, 0 length bone is inserted at that location (Fig. 2). From Leap Motion frames, 3D positions, velocities and other information are detailed about these detected fingers.

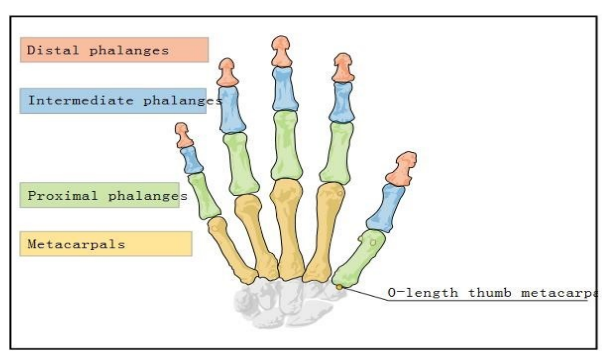

Fig. 2. Bones of the human hand

From valid tracking data, we derive the 3D positions of each joint for all fingers and encode to two bits to show finger status

978-1-5090-5188-5/16 $31.00 © 2016 IEEE
DOI 10.1109/ICVRV.2016.74

as in TABLE 1. The longitudinal arches (the rays formed by the finger bones and their associated metacarpal bones) is evaluated as full raised, half raised, clenched (folded) or crossed with neighbor finger. These four status are encoded as {00, 01, 00, 10} for one finger.

The first three status are easy to identify. For the fourth status to differentiate the crossed fingers, we will consider two cases. Firstly, for the first four fingers other than thumb, the encoder will consider the phalanges of fingers as segments. If they are intersected on palm plane (use the palm normal vector to project), both neighbor fingers are labelled as crossed. Secondly, for thumb, we only identify the other three status as it is separated from palm and can cross with any other finger.

To identify other non-crossed status, the stretch angle between proximal phalange and intermediate phalange intersect (except for thumb, where we observe the angle between intermediate phalange and distal phalange) is computed. In practice, assume that the 3D vectors for proximal phalanges as P and intermediate phalanges as I, the spread angle θ is given by

$$\cos\theta = \frac{P \cdot I}{|P| \cdot |I|} \tag{1}$$

The spread angle of joint is classified as summarized in Table I. This finger status is encoded for each finger. Their combination of shapes and orientation will visually transmit sign pattern to convey meaning in communication.

TABLE I. CODES FOR FINGER STATUS

$\theta \geq 0.8\pi$	11	Full raised
$0.4\pi < \theta < 0.8\pi$	01	Half raised
$\theta \leq 0.4\pi$	00	Clenched
Intersected	10	Crossed with neighbor finger

To differentiate various gestures, other than finger status, it should be known whether fingers are parted from each other. To encode this status, the angle between proximal phalange can be computed to capture two status: parted or not as in Eq.2. In coding system, we defined 0 as parted and 1 as not parted. In the LMC models, we have five fingers and four spaces between them to label. Suppose the 2D vectors J, K projected on palm plane for proximal phalanges of neighbor fingers, the spread angle δ between fingers is obtained similarly as in Eq.1. The status is decided as:

$$\text{status} = \begin{cases} 1, \cos\delta < T \\ 0, \cos\delta \geq T \end{cases} \tag{2}$$

where T is the threshold for the spread angle between fingers. On experiments, we use $\pi/12$ to differentiate the parted finger and close together ones. On the sample frame in Fig 3, there are two spaces between thumb and first finger, second finger and third finger, the correct code is 0101.

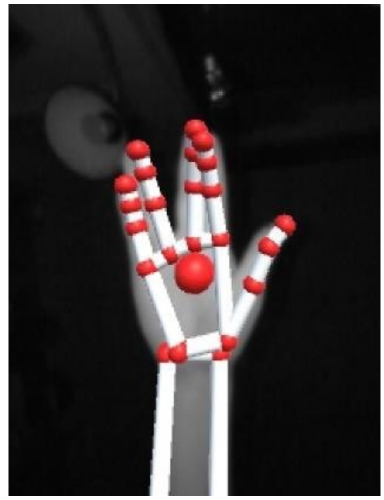

Fig. 3. This sign is coded as 0101 to show the two gaps between thumb and forefinger, middle finger and ring finger.

III. GESTURE CODING AND RECOGNITION

Instead of audible voice, sign language employs hand gestures to convey visual meaningful message. It is a major mean to benefit Hearing impaired individuals. Researchers employ various technologies to build the automatic conversion between sign languages and spoken languages [4] [5]. Other than using distance measure [5] [7], support vector machine [2] or decision tree [6], we apply 3D finger pose coding in Sec.2 to further encode the postures (Fig.4) for recognition.

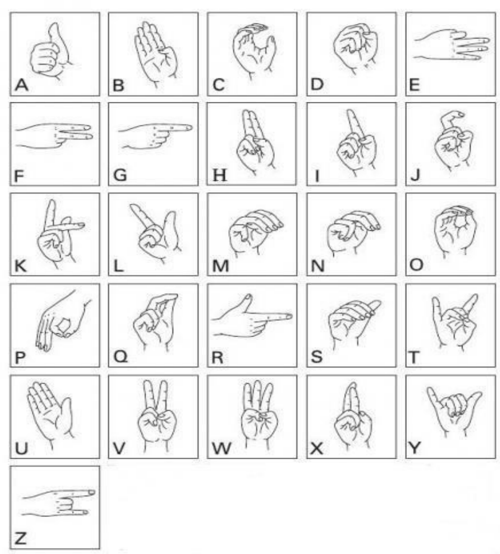

Fig. 4. Signs of the letters in Chinese Sign Language (CSL).

The next step is straight forward, for each posture in Fig. 3, a 14 digital code is given. 10 digitals is encoded as the sequence of pose coding for thumb, forefinger, middle finger, ring finger and pinky finger and 4 digitals to encode the existence of gaps between fingers. For example, two similar letters 'H' and 'V' are encoded as '00111100000101' and '001111000000' respectively. The full alphabet is encoded and saved manually according to Fig.4. This coding will be further integrated within the Unity plugins and used in applications.

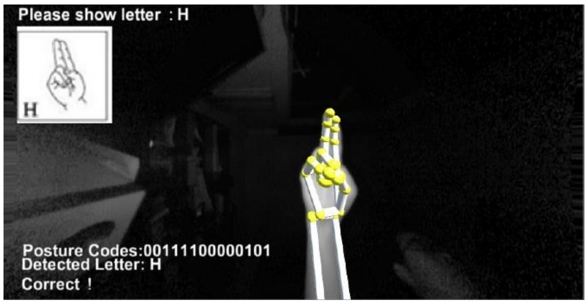

Fig. 5 Automatically Identification of sign H.

For application, integrated with Unity 3D and Leap Motion Core asset, our system provides a primitive prototype to learn and practice sign language. The human signer is given the sign definition image on the frame. The gesture is randomly selected from the visual vocabulary in Fig 4. Signer will follow the instructor to show the sign. Leap motion will help to figure out the information from the detected skeleton. In this process, we collected the hand direction, finger joint positions and palm normal from the device.

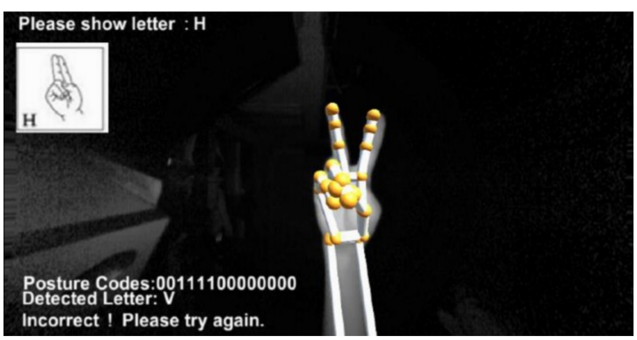

Fig. 6. Incorrect sign for H.

Based on collected data, the hand posture will be translated to the gesture coding with 14 binary digitals as in sec III. After comparison to pre-defined sign coding, the hamming distance between each pair of binary strings is calculated. In the other words, the hamming distance *d (x, y)* between two vectors x, y $\in$ F(14) is the number of coefficients in which they differ, where F(14) is a finite field with 14 elements in this system. As a system built for teaching and learning standard sign gestures, only identical binary strings are be considered as the right sign.

A negative sample is showed in Fig. 6. Similar gesture but with different codes is noticed and the system will tell the signer to correct and do more practice for more accurate visual patterns

Because our coding system is very sparse, there are numerous posture signs not in pre-defined sign vocabulary. After scanning and comparison with all known binary strings, if no matching is found, the detected gesture will be labelled as unknown. The feature will provide flexible vocabulary for the next section.

IV. Interactive Learning for New Gestures

Following the discussion in the last part of previous section, it is obvious that this coded vocabulary for gestures are very flexible and extendable. In most of works, while features are feed into traditional classifiers (SVM [2], KNN [3], and decision tree [6]) or training Multilayer perceptron Neural Networks (MLP) [8], the addition of new signs will require the change of learning structure and forceful re-training. To avoid this, in this section, we will present the addition of new signs in current code dictionary. This semi-automatic interactive procession is easy and user-friendly.

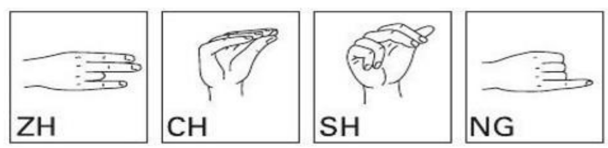

Fig. 7. Some voiceless/voiced pairs in CSL.

If signer plans to add some language-specific signs (for example, the phonemes in Fig. 7, which corresponding to pinyin in Chinese) into the vocabulary, he/she can initiate a procession to insert signs as in Fig.8.

Fig. 8. Insertion of new sign for phoneme 'SH'

Through the input field, signer can define the name of expected sign. Then, user can perform the sign, its posture is coded according to sec II and showed on the screen. If the codes are correct for the new posture of hand, signer can confirm the encoder and save for further usage. If the identifiable coding is not satisfying, signer can press 'Retry' to perform more accurate gesture. This semi-automatic process is straightforward and smooth. When saving the sign, the camera captures the image of hand as the instructor picture in learning. By this function, the sparse coding method can provide the flexibility and extendibility for various new gestures.

V. CONCLUSION

Sensor-based recognition systems have been developed in many applications to identify the sign language automatically. In this pilot work, we proposed to exploit a coding system to encode the fingers and hand gesture. This visual alphabets are based on 3D finger features extracted by LMC. The 3D coordinate positions are translated by simple rules to meaningful codes by fingers' shape and their correlation. This middle-level codes smooth the gap between various random gestures and meaningful sign language. User can take advantage of this easy-use prototype to learn sign language. The system can help people to improve their performance of any other signs in an automatic way.

As the proposed method provides a very sparse coding for any random gestures, the system contributes an extendable visual vocabulary. Users can insert more signs to the vocabulary. This feature can be compared with most applications [2] [8] which implement fixed export classifiers as SVM or MLP and non-extendable. To deal with these limitations, in this system signer can further increase the number of sign data to establish a rich vocabulary. This promising application can help the deaf or normal people to learn and practice sign language, even own the potential of automatic translator.

For future work, we plan to further enrich the coding system by adding more information encapsulated in sign language, such as orientation of palm, signs using both hands, even the facial expressions. The dynamic traces of hands [9] are also interesting to help in sentence interpreting. While under the developing, it is still in prototype stage. We hope this technology can impact the area of sign language education and interpreting

ACKNOWLEDGMENT

This work was partially supported by National Natural Science Foundation of China (NSFC Grant No. 61170124, 61272258, 6130129, 61272005, 61572085), Provincial Natural Science Foundation of Jiangsu (Grant No. BK20151254, BK20151260) and Collaborative Innovation Center of Novel Software Technology and Industrialization.

REFERENCES

[1] Y Quan. "Chinese sign language recognition based on video sequence appearance modeling". 5th IEEE Conference on Industrial Electronics and Applications (ICIEA), 2010, pp 1537– 1542

[2] C-H Chuan, R. Regina, C. Guardino, "American Sign Language Recognition Using Leap Motion Sensor". Machine Learning and Application (ICMLA), 13th International Conference on, Detroit, MI, 2014, pp 541-544.

[3] C. Dong, M.C. Leu, Z. Yin, "American Sign Language Alphabet Recognition Using Microsoft Kinect". Computer Vision and Pattern Recognition Workshops, 2015 IEEE Conference on, Boston, MA, 2015, pp 44-52.

[4] V. Tiwan, V,Anand, A.G. Keskar,V.R. Satpute, "Sign Language recognition through Kinect based depth images and neural network". Advances in Computing, Communications and Informatics (ICACCI), 2015 International Conferences on, Kochi. 2015. Pp. 194-198

[5] C. Sun, T. Zhang, BK Bao, C.S Xu, T. Mei."Discriminative Exemplar Coding for Sign Language Recognition with Kinect". IEEE Transactions on Cybernetics. 2013, Vol.43 (5). Pp.1418-1428.

[6] D. Yao, M. Jiang, A. Abulizi. X. You, "Decision-tree-based Algorithm for 3D Sign Classification". Proceedings of ICSP2014, Hangzhou, China, 2014, pp.1200-1204.

[7] R.B Napari, G. Kharat, "Real Time Human Pose Recognition Using Leap Motion Sensor', IEEE International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN). Kolkata, 2015, pp.323-328.

[8] A.S. Elons, M, Ahmed, H, Shedid and M.R. Tolba, "Arabic Sign Language Recognition Using Leap Motion Sensor", Computer Engineering & Systems (ICCES), 19th International Conference on, 2014, pp. 368-373.

[9] R. Yang, S. Sarkar, B. Leoding, "Handling Movement Epenthesis and Hand Segmentation Ambiguities in Continuous Sign Language Recognition Using Nested Dynamic", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010 Vol. 32, Issue 3, 462-477.