Journal of the Korea Academia-Industrial cooperation Society
Vol. 14, No. 10 pp. 5169-5175, 2013

http://dx.doi.org/10.5762/KAIS.2013.14.10.5169

A Framework for 3D Hand Gesture Design and Modeling

Doo-Young Kwon[1*]

[1]**Department of New Media, Korean German Institute of Technology**

삼차원 핸드 제스쳐 디자인 및 모델링 프레임워크

권두영[1*]

[1]한독미디어대학원대학교 뉴미디어학부

Abstract We present a framework for 3D hand gesture design and modeling. We adapted two different pattern matching techniques, Dynamic Time Warping (DTW) and Hidden Markov Models (HMMs), to support the registration and evaluation of 3D hand gestures as well as their recognition. One key ingredient of our framework is a concept for the convenient gesture design and registration using HMMs. DTW is used to recognize hand gestures with a limited training data, and evaluate how the performed gesture is similar to its template gesture. We facilitate the use of visual sensors and body sensors for capturing both locative and inertial gesture information. In our experimental evaluation, we designed 18 example hand gestures and analyzed the performance of recognition methods and gesture features under various conditions. We discuss the variability between users in gesture performance.

요 약 본 논문에서는 삼차원 핸드 제스쳐 디자인 및 모델링을 위한 프레임워크를 기술한다. 동작 인식, 평가, 등록을 지원하기위해 동적시간정합(Dynamic Time Warping, 이하 DTW)과 은닉마코브모델 (Hidden Markov Mode, 이하 HMM)을 활용 하였다. HMM은 제스쳐 인식에 활용되며 또한 제스쳐 디자인과 등록 과정에 활용된다. DTW은 HMM 훈련 데이터가 부족한 경우 제스쳐 인식에 활용되고, 수행된 동작이 기준 동작의 차이를 평가하는 데에 활용된다. 동작 움직임에 나타나는 위치 정보와 관성 정보를 모두 획득하기 위해 바디센서와 시각센서를 혼합하여 동작을 감지하였다. 18개의 예제 손동작을 디자인하고 다양한 상황에서 제안된 기법을 테스트하였다. 또한 제스쳐 수행 시 나타나는 사용자간 다양성에 대해 토론한다.

Key Words : 3D Hand Gesture, Gesture Design, Gesture Evaluation, Gesture Recognition

1. Introduction

The recent advance of sensing and display technologies has been transforming our living and working environment to a window connecting the physical and the virtual world. This new computational environment beyond desktops encourages the use of 3D hand gestures for more natural and intuitive human computer interaction. A wide range of 3D hand gestures from simple to complex has been designed and demonstrated in various applications including virtual reality, smart environments, game interface design, and digital art performance.

Our research goal is to improve the growth of available 3D hand gesture vocabulary by supporting people to easily design and learn gestures, and use optimal ones appropriate for their preference and physical condition. In this paper, we propose our approach to develop a design framework for 3D hand gestures by combining different sensors and putting emphasis onto the extensibility of the

This research is supported by Ministry of Culture, Sports and Tourism(MCST) and Korea Creative Content Agency(KOCCA) in the Culture Technology(CT) Research & Developement Program 2011

*Corresponding Author : Doo-Young Kwon(Korean German Institute of Technology)
Tel: +82-2-6393-3225 email: dykwon@kgit.ac.kr

Received September 16, 2013 Revised (1st September 30, 2013, 2nd October 7, 2013) Accepted October 10, 2013

model.

Using the proposed framework, users can acquire a wide range of gesture information from approximate to detail. A wearable input device is designed to support the easy integration of different body sensors and robust positional tracking with visual sensors. Our gesture model is designed to support the registration and evaluation of gestures as well as their recognition. We extended the previously introduced gesture unit, motion chunk that decomposes a 3D hand gesture into a set of postures and gestures[1]. The explicit distinction of postures and dynamic gestures within the HMM model facilitates the design of new gestures in a flexible and convenient way. We use the DTW technique to recognize gestures with a limited training data and also evaluate the performed gestures comparing to the templates. Therefore, users can use newly designed gestures without a large training dataset, and improve their performance during the practical use.

2. Related Work

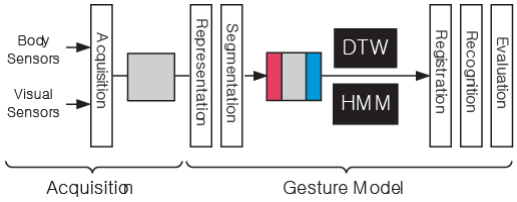
This section summarizes previous approaches for 3D hand gesture modeling. 3D hand gestures have been acquired using different sensor technologies. Electromagnetic sensors have been widely used to acquire six degrees of freedoms (6-DOFs). However, magnetic tracker devices are expensive and it is difficult to increase the number of tracking points. Therefore, electromagnetic sensors have mostly been used for the development of 3D input devices in VR.

Visual sensors, such as cameras, have been used to acquire 3D gestures along with a set of algorithms to derive visual features from the acquired camera images [2]. The main advantage of visual sensors is the lack of bulky and disturbing sensing devices. Users can freely move around inside the visible areas of the installed cameras. However, this approach relies on the accurate detection of relevant features which is a challenging task under varying illumination and background conditions. An alternative approach is to use passive or active markers. The VisionWand [3] was designed with two distinct colors for each end of the wand to provide more robust tracking with visual sensors. The XWand [4] utilizes Infrared LEDs. Occlusion often prevents continuous tracking of a desired body part from a single view. Very often, multiple cameras are used to partly solve the occlusion problem by taking images from different view angles.

Another popular approach to acquire 3D hand gestures is to use body sensors that are attached to the body or to a hand-held input device, and measure accurate movements directly from the body. With the advance of Micro-Electro-Mechanical Systems (MEMS) technology, body sensors are getting more advanced in terms of size, accuracy, and communication and can be embedded into wearable objects such as wrist watches, belts, and clothes [5]. 3motion [6] was developed as a 3D hand gesture input device with general-purpose software development kit. Soapbox [7] was introduced as a light, matchbox-sized device for a 3D gesture input. The device contains a set of basic hardware components namely processors, pre-defined sensors, and wireless and wired data communications. Ubi-Finger [8] was proposed for a gesture interface in smart environment. Perng et al. [9] developed a glove equipped with six 2-axis accelerometers on the finger tips and back of the hand to capture the movement of hands. Using this glove, they developed a text-editor application to type a letter of the alphabet using hand gestures.

3. Overview

Fig. 1 shows an overview of the proposed framework which consists of two main components (acquisition and gesture model). During acquisition (Section 4), 3D hand gestures are acquired through body sensors and visual sensors. The acquired data is segmented and represented with a combination of postures and gestures (Section 5.1). Using DTW and HMMs, the gesture model operates in three phases: design and registration (Section 5.2) to design a novel gesture and to add it to the system, evaluation (Section 5.3) to measure the quality of the input gesture, and recognition (Section 5.4) to identify the type of the un known input gesture. The following sections describe each component in detail.

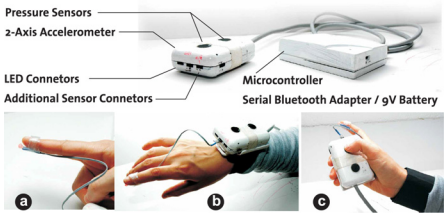


[Fig. 1] Overview of the framework

4. Acquisition

In our framework, 3D hand gestures can be acquired from body sensors (e.g. accelerometers) or visual sensors (cameras). Combining different sensors we intend to make features more expressive and to disambiguate recognition. A wearable input device (Fig. 2) is designed to help users integrate different body sensors. The device can be worn on the wrist (Fig. 2-b) like a wrist watch or hold in a hand (Fig. 2-c) like a cellular phone. By default the device is equipped with one 2D-axis accelerometer inside and two pressure sensors attached on the top surface of the case. Using external sensor connectors, users can easily connect other types of body sensors like bend sensors or digital compasses. The device provides LED connectors. With additional extension wires, users can connect LEDs and different colors and attach them to body parts such as fingers (Fig. 2-a), elbows, and shoulders.

Bright color LEDs enable faster and more robust tracking of multiple 3D positions using visual sensors. Their focal brightness provides relatively robust tracking results even for small-scale movements in indoor environments. To compute the 3D position of the interest, we employ conventional triangulation from a pair of calibrated cameras [10,11]. We use accelerometers that precisely measure the tilt, movement, and vibration of individual body parts.

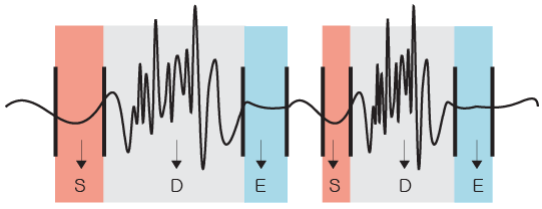


[Fig. 2] The wearable input device

5. The Gesture Model

5.1 Segmentation and Representation

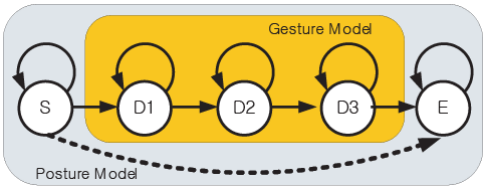
The obtained gesture signals are processed to find the start and end point of a gesture using a simple sliding window technique. We compute a standard deviation of the samples in the window (typically of size 20) which slides along the signal with a sampling rate of 30Hz. We assume that a gesture starts with a preceding start posture if the standard deviation is above the starting threshold, and subsequently a gesture ends with a following end posture if the standard deviation is below the ending threshold. After segmentation, the segmented signal is represented based on the structure of motion chunk [1] as shown in Figure 3. This motion chunk is used as the core representation of our gesture model and serves as a basis for gesture design, registration, evaluation, and recognition.



[Fig. 3] The structure of a motion chunk: start-static chunk S, dynamic chunk D, and end-static chunk E.

5.2 Gesture Design and Registration

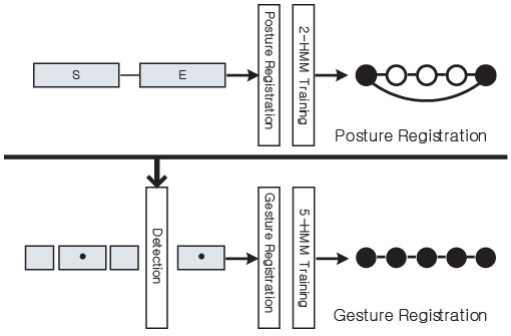
A user designs an individual 3D hand gesture following the structure of motion chunk (i.e. design first start posture and an end posture, and in-between gesture connecting two postures subsequently). According to this design sequence, each 3D hand gesture is modeled as a single HMM(λ) [12] with five states as illustrated in Figure 4. The first start and end state are equivalent to the start-static chunk and the end-static chunk respectively. The three in-between states are used for dynamic chunk features only. For static chunks they are skipped by directly connecting a start state to an end state. We call the resulting two state HMM a posture model and the complete five state HMM a gesture model.



[Fig. 4] The topology of the HMM model.

Once postures are designed, the two state posture model can be trained separately from the gesture model. In our framework, this pre-trained posture model is used to detect input training gestures automatically. We generalized this process with two separate interactive steps: a posture registration and a gesture registration as illustrated in Figure 5. During posture registration, users provide the start posture and the end posture for a certain time (2 or 3 seconds) by pressing the upper and lower pressure buttons of the device (Fig. 2-c) respectively. The two types of posture data (O_s, O_E) are used to adjust the parameters of the two-state posture HMM model respectively.

Once the posture model is trained, the system employs it to automatically discriminate training gestures for the full 5 state gesture HMM model from arbitrary input gestures such as recovery gestures or rest gestures. The detection is accomplished if $P(O_s, O_E|\lambda)$ is above a certain threshold (typically 90%). This approach guides users to easily design 3D hand gestures, and simplifies the user's effort to manually segment and detect training gestures.



[Fig. 5] Overview of the gesture registration process.

5.3 Gesture Evaluation

The gesture evaluation measures the similarity between the actual gesture and a reference gesture. The result (e.g. a numerical score) can for instance be used to improve user performance or to correct wrong gestures as presented in our previous work [1]. Similar to the practical motion training process [6], the evaluation consists of both posture evaluation and gesture evaluation. Three distinct scores are computed for the start static chunk, the dynamic chunk, and the end static chunk respectively. We use Dynamic Time Warping (DTW) that supports non-linear time alignment differences between an input gesture and a template gesture [13]. We also applied the Derivative Dynamic Time Warping (DDTW) technique [14] for a more natural alignment.

5.4 Gesture Recognition

The gesture recognition identifies the gesture template that most closely matches the input gesture. We designed a HMM recognizer and a DTW recognizer. The HMM recognizer is used when a certain amount of training data (typically 20) is available to parameterize and condition the model. It accommodates the probabilistic nature of the signal efficiently. During the training phase, an HMM λ_n is built for each gesture G_n. Then, for each unknown gesture, the model computes the likelihoods for all possible models $P(O|\lambda_n), 1 \le n \le N$ and selects the gesture $G_n\hat{}$ with the highest model likelihood.

The DTW recognizer as a non-parametric technique employs the original gesture frames directly for gesture recognition. It works even in cases where only one training dataset is available so that newly designed gesture can be recognized without a large training dataset. The DTW recognizer identifies the type of input gesture by selecting the template that minimizes the overall distance to the input gesture. We provide two different types of DTW recognizers depending on the number of templates: a single template DTW (SDTW) and a multiple-template DTW (MDTW). The MDTW improves the recognition rate by accommodating the variations between multiple templates even though it can be computationally more expensive. In practice, three templates are sufficient in our tests.

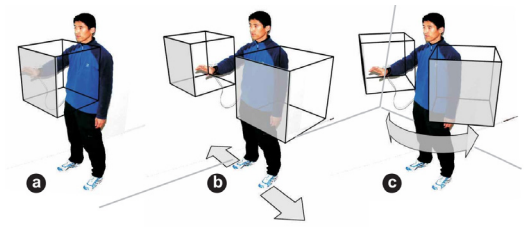
6. Experimental Evaluation

6.1 Process

We conducted a preliminary evaluation to test our framework, and analyze issues in designing and learning 3D hand gestures. We designed 18 hand gestures with three style groups for 3D hand gestures: a planar-style, a curved style, and a twisted-style. If the trajectory of gesture is on the 2D plane, we call it a planar-style. On the hand, the curved gesture is performed by drawing 3D curve. The twisted style is for the gestures performed with twisted hand-palm.

We hired two subjects (male and female) individually and asked to provide twenty training data. They wore the proposed wearable input device with the LED ring on the index finger as illustrated in Figure 2. 2-dimensional accelerometer data was used for body sensor features and the relative 3D positions (rx,ry,rz) of the index finger tip were used as the visual feature. Our experimental setup with two cameras provides the active volume (about 3×3×3 in meter) regarding shift, and to the maximum rotation angle (60 °).

Two other independent test data sets for translated (shifted) position and rotated position were acquired and utilized to test the invariance of the recognition, as illustrated in Figure 6. We used leave-one-out (LOO) cross validation to compute the recognition rates. During acquisition, subjects were requested to randomly change their positions in short time intervals to create more realistic situations. This added some additional variation to their gesture performances.



[Fig. 6] The three different user positions: (a) same (initial), (b) shifted, and (c) rotated.

6.2 Results

Table 1 shows the result of testing gesture features at different user positions with five state HMM (5SHMM). Overall, the combined visual and body features (VB) performs best and achieves the highest recognition rates in all three user positions. As expected, the body-only features (B) outperform the visual-only features (V) in the rotated-position, reaching about 15.9% reduction in the error rate. The visual sensor features perform better for shifted positions. We also compared DTW recognizers (SDTW and MDTW) with the HMM recognizer. Even though the HMM recognizer is still better, the result of the DTW recognizers is also good considering the required amount of training data (1 for SDTW and 3 for MDTW).

To analyze the performance variability between two subjects, we compared a user-dependent model (D) and a user- independent model (I) in terms of three different gesture styles. As Table 2 shows, while the recognition rates of the user-dependent model are over 90%, the recognition rates of the user-independent model is below 50% due to the difference in the gesture performance between users. In the user-independent model, the recognition rate of the curved-style gestures are far inperior to the others. Two subjects spontaneously turned their hand in different ways because the diagrams for a curved-style (Fig. 7-b) do not indicate the hand face (palm-down and palm-up) and the rotational direction of the hand.

[Table 1] Recognition rates of three user positions with different gesture features.

User Position	same	shifted	rotated	overall
V-5SHMM	96.0%	88.2%	60.0%	81.4%
B-5SHMM	94.5%	85.2%	75.9%	85.2%
VB-5SHMM	95.4%	93.1%	86.3%	91.6%
VB-SDTW	89.2%	86.7%	78.2%	84.7%
VB-MDTW	91.4%	89.3%	85.6%	88.7%

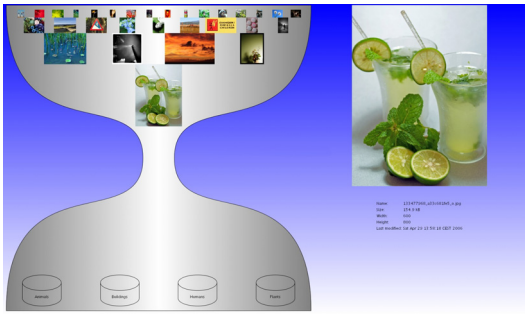
[Table 2] Recognition rates of three gesture style groups with the user-dependent (D) and the user-independent (I) model.

Gesture Type	planar	curved	twisted	overall
VB-5SHMM (D)	90.8%	97.2%	98.2%	95.4%
VB-5SHMM(I)	69.8%	20.5%	60.7%	50.3%

7. Prototype Application

To make our framework more generic, we continuously strive to separate application-related policy issues from technical issues such as acquisition and gesture model. In this section, we briefly introduce a prototype application implemented during the development of our framework.

This prototype application (Fig. 7) was developed to explore the creation and use of 3D hand gestures on a large screen display where the orientation of the user is rather fixed. In this application, users can register their own 3D hand gestures and perform the task of sorting digital photos using registered gestures. The sandglass metaphor (Fig. 7) was invented to accomplish image sorting tasks only using a set of 3D hand gesture commands.



[Fig. 7] The Sandglass Interaction Graphical User Interface (GUI).

The sand-glass metaphor is used in the sense that all images fall from the top of the sandglass to the bottom in a sequential manner. At the small gap at the center, users can move each photo to destination folders by simply performing 3D hand gestures. This allows the users to focus on the task of sorting photos without relying on the use of icons and menus which might be difficult to operate on a large screen display. In addition to gesture recognition, the application provides online gesture registration and evaluation using our framework. During the gesture registration (typically when the system starts), users are asked to perform a single gesture for each command to setup their own gesture templates.

Once this short initialization process is finished, the application can recognize new input gestures by comparing them to the same-user gesture sets, and evaluate the performance in relation to reference gesture sets trained by another user for instruction. When the gesture is recognized, its associated command is executed. Obviously, if users know how to perform the required 3D hand gestures, the gesture registration and evaluation process can be skipped. The DTW recognizer can be switched to the HMM recognizer if enough training data is available.

8. Conclusion and Future Work

In this paper, we presented a versatile framework to acquire, design and recognize 3D hand gestures using a wearable input device. It is intended to support application developers and end-users in easily exploring the full advantages of 3D hand gestures for human computer interaction. As we pursue the use of more various 3D hand gestures for HCI, we will further develop our framework with respect to gesture features and gesture recognition. For additional gesture features, we intend to test other body sensors such as digital compasses and gyro sensors using mWire, and also investigate the Kinect sensor which have been extensively studied in computer vision research.

On the other hand, we currently extend our gesture model to efficiently handle activity level gestures which consist of multiple sub-gestures. The use of motion chunk in our approach can provide a convenient and efficient method for designing activity level gestures reducing the computational complexity. We also intend to handle highly dynamic gestures for some specific game applications like sparring and sports. In general, the recognition of such gestures is a non- trivial task because the silent boundaries are no longer evident owing co-articulation with different speed and power.

References

[1] D. Y. Kwon and M. Gross. Combining body sensors and visual sensors for motion training. In Proceedings of ACM SIGCHI ACE'05, pages 94–101. 2005. DOI: http://dx.doi.org/10.1145/1178477.1178490

[2] S. T., Auxier, J., Ashbrook: The gesture pendant: A

self-illuminating, wearable, infrared computer vision system for home automation control and medical monitoring. In Proceedings of ISWC 2000.
DOI: http://dx.doi.org/10.1109/ISWC.2000.888469

[3] X. Cas, R. Balakrishnan: Visionwand: interaction techniques for large displays using a passive wand tracked in 3d. In Proceedings of UIST '03, pp. 173-182. 2003.
DOI: http://dx.doi.org/10.1145/964696.964716

[4] A. Wilson., S. Shafer.: Xwand: Ui for intelligent spaces. In Proceedings of ACM CHI'03, pp. 545-522. 2003.
DOI: http://dx.doi.org/10.1145/642611.642706

[5] J. Rekimoto: Gesturewrist and gesturepad: Unobtrusive wearable interaction devices. In Proceedings of the ISWC '01, p. 21. 2001.
DOI: http://dx.doi.org/10.1109/ISWC.2001.962092

[6] P. Keir, J. Payne, J. Elgoyhen, M. Horner, M. Naef, and P. Anderson. Gesture-recognition with non-referenced tracking. In Proceedings of the 3D User Interfaces (3DUI'06), 2006.
DOI: http://dx.doi.org/10.1109/VR.2006.64

[7] E. Tuulari and A. Ylisaukko-oja. Soapbox: A platform for ubiquitous computing research and applications. In Proceedings of Pervasive '02, pages 125-138, 2002.
DOI: http://dx.doi.org/10.1007/3-540-45866-2_11

[8] K. Tsukada and M. Yasamura. Ubi-finger: Gesture input device for mobile use. In Proceedings of APCHI '02, pages 388-400, 2002.

[9] J.K. Perng, B. Fisher, and S. Hollar et al. Acceleration sensing glove. In Proceedings of The Third International Symposium on Wearable Computers, pages 178-179, 1999.
DOI: http://dx.doi.org/10.1109/ISWC.1999.806717

[10] Z.Zhang.. Flexible camera calibration by viewing a plane from unknown orientations. In Proceedings of the 7th International Conference on Computer Vision 1999, pages 662-673, 1999
DOI: http://dx.doi.org/10.1109/ICCV.1999.791289

[11] OpenSource Computer Vision Library. Intel Corp. , http://www.intel.com.

[12] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In Proceedings of the IEEE, volume 77, pages 257-286, Feburuary 1989.
DOI: http://dx.doi.org/10.1109/5.18626

[13] M. H. Ko, G. West, S. Venkatesh, and M. Kumar. Online context recognition in multisensor systems using dynamic time warping. In Proceedings of ISSNIP '05, 2005.

[14] E. Keogh and M. Pazzani. Derivative dynamic time warping. In Proceedings in First SIAM International Conference on Data Mining, 2001.
DOI: http://dx.doi.org/10.1109/ISSNIP.2005.1595593

Doo-Young Kwon [Regular member]

- Feb. 2004 : University of Washington, Dept. of Architecture, MS
- July. 2007 : Swiss Federal Institute of Technology, Dept. of Computer Science, PhD
- Feb. 2009 ~ Current : KGIT. Dept. of New Media, Associate Professor

<Research Interests>
New Media Art & Design, HCI, Mixed Reality