

A Framework for 3D Spatial Gesture Design and Modeling Using a Wearable Input Device

Doo Young Kwon and Markus Gross
Computer Graphics Laboratory, ETH Zürich, Switzerland
dkwon, grossm@inf.ethz.ch

Abstract

We present a framework for 3D spatial gesture design and modeling. A wearable input device that facilitates the use of visual sensors and body sensors is proposed for gesture acquisition. We adapted two different pattern matching techniques, Dynamic Time Warping (DTW) and Hidden Markov Models (HMMs), to support the registration and evaluation of 3D spatial 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 gestures with a limited training data, and evaluate how the performed gesture is similar to its template gesture. In our experimental evaluation, we designed 18 example gestures and analyzed the performance of recognition methods and gesture features under various conditions. We discuss the variability between users in gesture performance.

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 spatial gestures for more natural and intuitive human computer interaction. A wide range of 3D spatial 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 spatial 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 spatial gestures by combining different sensors and putting emphasis onto the extensibility of the 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 spatial gesture into a set of postures and gestures [3]. 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. Overview

Figure 1 shows an overview of the proposed framework which consists of two main components (acquisition and gesture model). During acquisition (Section 4), 3D spatial gestures are acquired through body sensors and visual sensors. The acquired data is segmented and represented with a combination of postures and gestures (Section 4.1). Using DTW and HMMs, the gesture model operates in three phases: *design and registration* (Section 4.2) to design a novel gesture and to add it to the system, *evaluation* (Section 4.3) to measure the quality of the input gesture, and *recognition* (Section 4.4) to identify the type of the unknown input gesture. The following sections describe each component in detail.

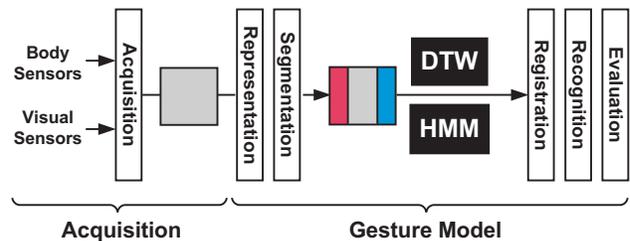


Figure 1. Overview of the framework.

3. Acquisition

In our framework, 3D spatial 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 (Figure 2) is designed to help users integrate different body sensors. The device can be worn on the wrist (Figure 2-b) like a wrist watch or hold in a hand (Figure 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 (Figure 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 [6], [4]. We use accelerometers that precisely measure the tilt, movement, and vibration of individual body parts.

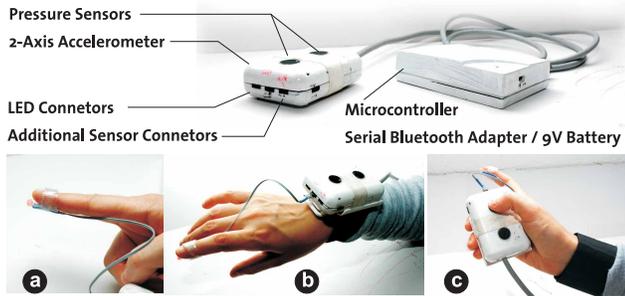


Figure 2. The wearable input device.

4. The Gesture Model

4.1. 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* [3] 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.

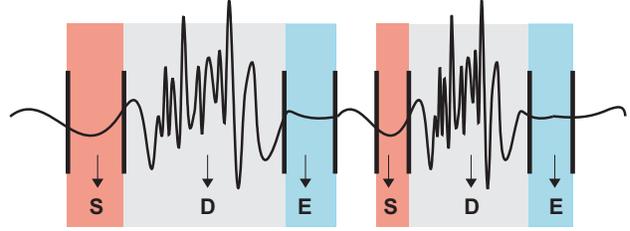


Figure 3. The structure of a motion chunk: start-static chunk S , dynamic chunk D , and end-static chunk E .

4.2. Gesture Design and Registration using HMM

A user designs an individual 3D spatial 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 spatial gesture is modeled as a single HMM(λ) [5] 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*.

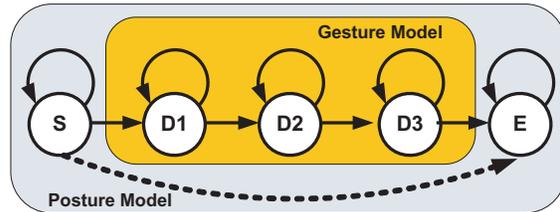


Figure 4. The topology of the HMM model.

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 (Figure 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 spatial gestures, and simplifies the user’s effort to manually segment and detect training gestures.

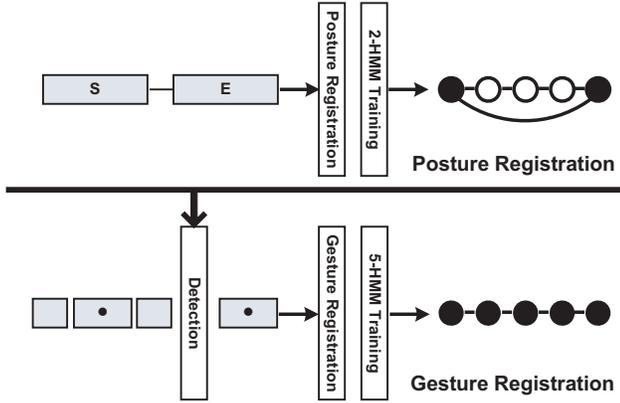


Figure 5. Overview of the gesture registration process.

4.3. Gesture Evaluation using DTW

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 [3]. Similar to the practical motion training process [3], 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 [2]. We also applied the Derivative Dynamic Time Warping (DDTW) technique [1] for a more natural alignment.

4.4. Gesture Recognition using HMM/DTW

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 $P(O|\lambda_n)$, $1 \leq n \leq N$ and selects the gesture $G_{\hat{n}}$ with the highest model likelihood.

The DTW recognizer as a non-parametric technique employs the original gesture frames directly for gesture recog-

nition. 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.

5. Experimental Evaluation

5.1. Process

We conducted a preliminary evaluation to test our framework, and analyze issues in designing and learning 3D spatial gestures. We designed 18 gestures with three style groups for 3D spatial gestures: a planar-style, a curved-style, and a twisted-style, and represented with our unique gesture diagram as shown in Figure 7.

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 $3 \times 3 \times 3$ in meter) regarding shift, and to the maximum rotation angle (60°).

Two other independent test datasets 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.

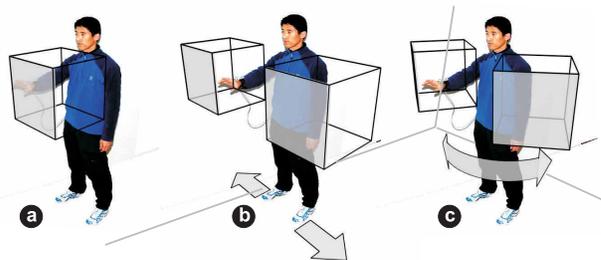


Figure 6. The three different user positions: (a) same (initial), (b) shifted, and (c) rotated.

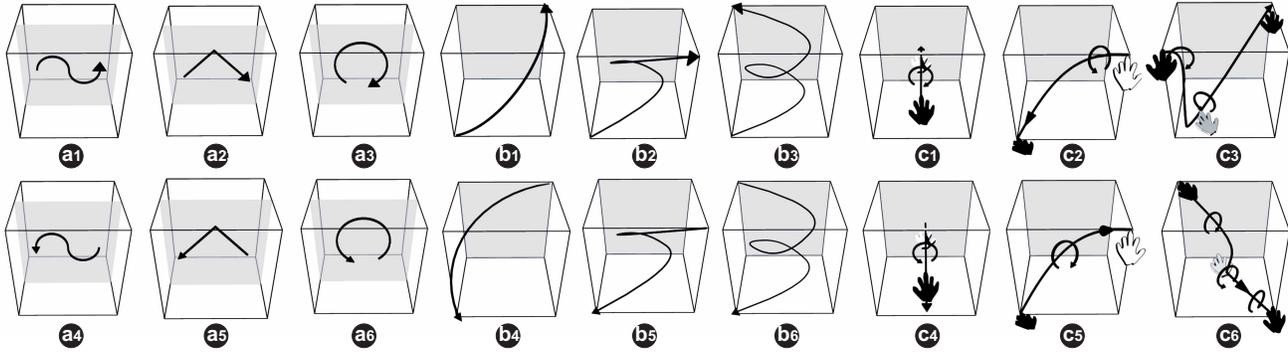


Figure 7. The 18 gesture diagrams with three style groups: (a) planar, (b) curved, and (c) twisted. The line indicates the trajectory of the gesture and the end of the gesture is presented as an arrow. The hand symbol uses black to indicate palm-down and white for palm-up.

5.2. 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 inferior to the others. Two subjects spontaneously turned their hand in different ways because the diagrams for a curved-style (Figure 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%

6. Conclusion

In this paper, we presented a versatile framework to acquire, design and recognize 3D spatial gestures using a wearable input device. It is intended to support application developers and end-users in easily exploring the full advantages of 3D spatial gestures for human computer interaction.

Acknowledgments. This work is carried out in the context of the blue-c-II project, funded by ETH grant No. 0-21020-04 as an internal poly-project.

References

- [1] E. Keogh and M. Pazzani. Derivative dynamic time warping. In *Proceedings in First SIAM International Conference on Data Mining*, 2001.
- [2] 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.
- [3] 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. ACM Press, 2005.
- [4] OpenSource Computer Vision Library. Intel Corp., <http://www.intel.com>.
- [5] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77, pages 257–286, February 1989.
- [6] 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.