Feature Processing and Modeling for 6D Motion Gesture Recognition

Mingyu Chen, Ghassan AlRegib, *Senior Member, IEEE,* and Biing-Hwang Juang, *Fellow, IEEE*

***Abstract*—A 6D motion gesture is represented by a 3D spatial trajectory and augmented by another three dimensions of orientation. Using different tracking technologies, the motion can be tracked explicitly with the position and orientation or implicitly with the acceleration and angular speed. In this work, we address the problem of motion gesture recognition for command-and-control applications. Our main contribution is to investigate the relative effectiveness of various feature dimensions for motion gesture recognition in both user-dependent and user-independent cases. We introduce a statistical feature-based classifier as the baseline and propose an HMM-based recognizer, which offers more flexibility in feature selection and achieves better performance in recognition accuracy than the baseline system. Our motion gesture database which contains both explicit and implicit motion information allows us to compare the recognition performance of different tracking signals on a common ground. This study also gives an insight into the attainable recognition rate with different tracking devices, which is valuable for the system designer to choose the proper tracking technology.**

***Index Terms*—Gesture Recognition, Motion Gesture, 6D Motion Tracking**

I. Introduction

A GESTURE is a meaningful body movement expressed by a subject. The expression may be realized by moving the user's body, face, hands, and/or fingers. It can convey the user's intention to communicate or interact with the environment. In human-computer interactions, gestures can form a complementary modality beyond traditional input devices. Motion gestures, rendered as part of the motion control, also empower the user to interact with a device or an implement in a more natural and intuitive way.

We can consider gestures as a special case of a sign language. Natural gestures are free-form and can occur in any order whereas a sign language is normally linguistically structured and has a defined grammar. Gesture recognition is practically identical to isolated sign recognition, but may not have the contextual or grammatical constraints of a sign language. In this work, we focus on the recognition of gestures without associating them with meanings or interpretations because they can be culture-specific. Even with the same definition, the gesture performed by different individuals can vary dynamically in shape and duration, which makes user-independent gesture recognition very challenging.

M. Chen, G. AlRegib, and B-H. Juang are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332 USA.

As in sign languages, gestures can consist of static postures, dynamic motions, or both. However, we only focus on gestures defined by dynamic motions, i.e., motion gestures. To be more precise, we refine the definition of gestures as distinguishable hand movements in a free space without regard to finger/body motion or posture. Our main interest is in motion command and control, to which issues with posture are traditionally considered distant.

With the emerging concept of the natural user interface (NUI), new interface modes are designed and meant to support truly natural human motions in a free space. It is convenient and more intuitive to use 3D motions to interact with a 3D user interface. We can expand the gesture vocabulary without the 2D constraint. Motion information beyond a 2D trajectory, such as depth and orientation, provides additional insight into the motion gesture and can possibly improve the accuracy and robustness of gesture recognition. Tracking an object in space actually requires six dimensions: three for translation and three for rotation. Therefore, a 6D motion gesture is represented by a 3D spatial trajectory and augmented by another three dimensions of orientation. In our case, the motion gesture is composed only of the position and orientation of the hand or the handheld device.

There are several technologies for motion tracking, each with its own characteristics in terms of sampling rate, latency, resolution, and accuracy [1]. Nowadays, optical sensing and inertial sensing are the most popular. Optical sensing tracks the position in a global reference frame. If we construct a rigid body with at least four tracking points, the optical sensing can also estimate the global orientation. In contrast, inertial sensing measures the acceleration and angular speed in the device-wise coordinates, which depict an implicit 6D motion. Theoretically, it is possible to reconstruct the spatial trajectory from the implicit 6D data with an inertial navigation algorithm. Drifting and error propagation would deteriorate the reconstruction over time and make inertial sensing less accurate than the explicit 6D motion from optical tracking. However, the implicit motion signal still contains enough kinematic cues for describing and recognizing the motion gesture.

Depending on the tracking technology in use, the scope of supported control motion varies. For a conventional pointing device, such as a mouse or a trackpad, the control motion is limited to 2D trajectories on a plane, which also forms the basis of many current motion gesture interface devices. Confining gestures to a 2D plane can be unnatural, and converting 3D gestures into 2D ones by projection degrades the discrimination performance due to loss of information.

A motion gesture can be viewed as a spatio-temporal pattern composed of different tracking results, including the spatial trajectory with or without the orientation information. Depending on the tracking technology, these tracking results can be explicit, implicit or both as described above. In addition to the motion tracking, we also need to know the starting and ending points of a gesture both in time and in space. A common scheme for temporal segmentation is push-to-gesture, which allows the user to specify when the gesture starts and ends. The "push" may be triggered by a physical button or other explicit signals, e.g., voice. The alternative method of continuous gesture recognition is gesture spotting, which tries to locate meaningful patterns from a stream of motion. Gesture spotting is beyond the scope of this paper, and we focus only on the recognition of 6D motion gestures.

There exists no published gesture dataset that has both a large gesture set and comprehensive motion information. Thus, we first build a 6D motion gesture database (6DMG) which contains both explicit and implicit 6D motion data in a set of 20 gestures. We are interested in understanding which type of tracking signals and features help to describe the motion gesture. 6DMG makes it possible to compare the recognition performance over different tracking signals on a common ground.

Our main contribution is to investigate the relative effectiveness of various features derived from different tracking signals for motion gesture recognition. Similar to the case of speech recognition, it is desirable that the recognition system accommodates user-specific customization, but it is also very important to achieve robust user-independent recognition. Both user-dependent and user-independent cases are addressed. We recognize the motion gesture by treating it as a static pattern or by considering the time series nature of the motion. In the former approach, we extract a corresponding fixed length feature set from various tracking signals. These features are either geometric or algebraic and barely contain any temporal or ordering information. Thus, we propose a temporal extension to the feature set to include the temporal characteristics of a motion gesture. Benchmark recognition results are then obtained by applying a simple linear classifier on the extracted features. The second approach represents the motion gesture as a sequence of feature vectors (observations) derived from various tracking signals and uses hidden Markov models (HMM) for recognition. The HMM structure is chosen with reasonable physical meanings. We also propose a feature normalization procedure and prove its effectiveness in achieving "scale" invariance especially in the user-independent case. Our study gives an insight into the attainable recognition rate with different tracking devices. Overall, the statistical feature-based linear classifier can achieve 85.2% and 93.5% accuracy with implicit and explicit 6D data. The HMM-based recognizer has higher recognition rates, 91.9% and 96.9% respectively.

The paper is organized as follows. In the next section, we describe the background of the research, including motion tracking technologies and previous studies on gesture recognition. In Section III, we present the 6D motion gesture database. Section IV is devoted to the detailed description of feature extraction and gesture recognition. The experiments and results are given in Section V, and Section VI concludes this paper.

II. Background and Related Works

The implementation of a gesture-control interface contains two key components: motion tracking and gesture recognition. In Figure 1, a general system diagram is shown, where the tracking technology, corresponding motion data, and the recognition kernel vary upon implementation.

A. Motion Tracking

We have to capture the motion before performing gesture recognition. Motion capture devices are essentially the input device for a motion-based user interface. Compared to planar pointing devices, a 3D input device usually has a higher level of tracking noise and is subject to hand tremor if held in space. Here, we categorize the tracking technologies into two types: vision-based and tracker-based.

Vision-based techniques provide more natural and unencumbered interaction. An ideal vision-based system poses no requirement on the user to wear markers, gloves, or long sleeves. Based on monocular images or videos, we can extract the projected 2D trajectory and possibly the orientation. A depth camera usually estimates the depth in a rougher scale than the fixed image plane. With the help of stereo or multi-view cameras, it is possible to track a full 3D motion. Xbox 360 Kinect demonstrates the capability of tracking a human body in 3D, but it also has limitations on tracking subtle hand movements like wrist twisting. Moreover, the accuracy and the robustness of vision-based systems are affected by many factors such as self-occlusion, varying illumination and lighting, and interference with existing objects or backgrounds.

On the other hand, tracker-based techniques achieve more precise motion tracking at the expense of requiring the user to wear certain equipment, which may encumber the interaction. A motion tracking system most often derives estimate of motion information from magnetic, acoustic, inertial, or optical sensors. Each approach has advantages and limitations. In [1], it is argued that there is no silver bullet that satisfies the needs in every circumstance and application. Because we are using the motion tracking system as the input device to motion-based user interfaces, the tracker is strongly desirable to be small and wireless to minimize the user's efforts for interaction. Among these tracking technologies, optical sensing and inertial sensing better suit our needs.

Optical sensors track either active or reflective markers and provide accurate motion tracking results at a relatively high speed. A primary constraint of all optical systems is that there must be a clear line of sight between the tracking targets and the optical sensor, and at least two pairs of the tracker-sensor relationship are needed for valid triangulation to determine the position of one tracker. More thorough study of the characteristics of spatio-temporal signals acquired by optical motion tracking can be found in [2], [3].

Here, we refer the inertial sensors to the MEMS (micro-electronic mechanical systems) accelerometers and gyroscopes in chip form. Comparing to optical motion tracking, inertial

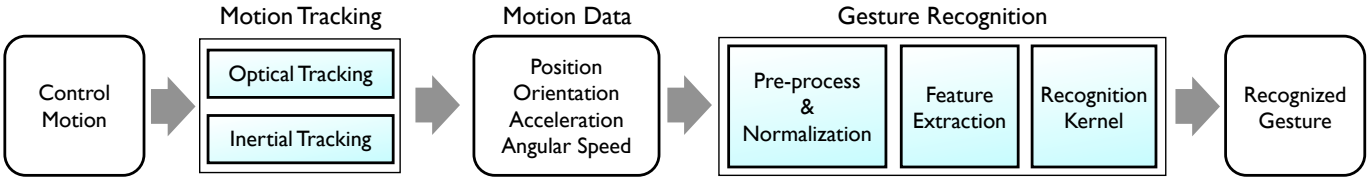

Fig. 1. The system diagram for gesture control

sensors have smaller latency and a much higher sampling rate. The accuracy of inertial sensing has been studied in static, quasi-static, and dynamic cases [4]. The accelerometers measure the accelerations in the device-wise coordinates, and the gyroscope measures the angular speeds in yaw, pitch, and roll. The orientation, as a measurement, is actually accumulated from the angular speeds based on a global reference frame which can be determined by measuring the direction of gravity and the magnetic north if magnetic sensors are equipped. We can also reconstruct the motion trajectory by integrating the accelerations along the corresponding orientation, but the accuracy is not reliable due to the drifting issue and error propagation over time.

In order to accurately track motions with as minimum system noise as possible, a hybrid framework of optical sensing and inertial sensing should satisfy the need. The former measures the position, and the latter estimates the orientation of the tracking device. In addition to the explicit position and orientation, there are actually six extra dimensions from the acceleration and angular speed, which can also infer the kinematic properties of the motion gesture.

B. Gesture Recognition

Motion gesture recognition has been an active research topic for years. Although there has been a significant amount of work on recognizing gestures with either explicit or implicit motion information, thorough study and comparison of the full six dimensions are lacking.

Here, we temporarily relax the definition of a gesture as a finite characteristic motion made in 2D or 3D space using a suitable input device. We also generalize gesture recognition as a spatio-temporal pattern recognition problem, which may include the sign language recognition and online handwriting recognition. Unlike speech signals, gestures lack a standardized "vocabulary", but there are still several widely accepted basic motions, e.g., swiping motions. Including allography, the rendering of uni-stroke hand writing characters still follows a fixed definition with no room for user customization. High level linguistic contraints can help the recognition of concatenated characters in a word or sentence. In contrast, motion gestures usually don't concatenate in general and have no contextual information.

Major approaches for analyzing spatial and temporal patterns include Dynamic Time Warping (DTW) [5], Neural Networks (NNs) [6], Hidden Markov Models (HMMs) [7]–[9], data-driven template matching [10], [11], and statistical feature-based classifiers [12], [13]. In general, the reported recognition rates are above 90%. Since these results are obtained with different data sets and various experimental settings, a direct comparison of the performance achieved by these techniques is not meaningful.

When designing a recognizer, a trade-off is usually made between personalization and generality. The two extreme cases are user-dependent and user-independent gesture recognition. Customized gestures are usually personal and are only considered in the user-dependent case, which has no generality issue. Even with a predefined gesture vocabulary, robust user-independent gesture recognition can be very challenging due to the large variations among different users.

The DTW is an effective algorithm based on dynamic programming to match a pair of time sequences that contain temporal variability (i.e., stretching or compressing in time). It is an important component in template-based pattern recognition and classification. The issue of general statistical variability in the observation is not explicitly addressed by the matching algorithm and a template-based system is usually used in user-dependent applications, e.g., personalized gesture recognition, where such variability is limited. DTW can be useful for personalized gesture recognition, where a large set of training samples is hard to collect. When the range of variations increases, e.g., in the user-independent case, the need for explicit statistical modeling of the variability becomes crucial for the sake of performance and computational load.

The $1 recognizer [10] and its variants [11] are also based on template matching. Unlike DTW which relies on dynamic programming, these algorithms process the trajectory with re-sampling, rotation, and scaling and then match the point-paths with the reference templates. These recognizers are simple to implement, computationally inexpensive, and require only a few training samples to function properly. However, for user-independent recognition, a significant amount of templates are needed to cover the range of variations and hence the performance of such approaches is often degraded.

The Rubine classifier [12] is a popular feature-based statistical classifier. It captures geometric or algebraic properties of a 2D gesture for recognition. The planar trajectory is converted into a fixed length feature set and recognized by a linear classifier. Hoffman [13] extends Rubine's feature set to work on the implicit 6D motion, i.e., the acceleration and angular speed. Note that the feature extraction actually treats a gesture more like a static path regardless of the temporal (ordering) information, which may cause confusion between mirroring gesture pairs. In this work, we propose an extension to incorporate the temporal information, and use the feature-based statistical classifier as the baseline for performance comparison.

The HMM is efficient at modeling a time series with spatial

TABLE I
THE GESTURE LIST OF 6DMG, INCLUDING THE GESTURE NAME, DURATION (NUMBER OF SAMPLES), THE MAXIMUM TO MINIMUM RATIO OF THE NORMALIZATION SCALE

Gesture Name	Sample # avg. (std.)	max/min norm. ratio P	O	V	W	A
SwipeRight	51.9 (20.7)	8.7	5.9	14.5	12.4	24.1
SwipeLeft	51.6 (20.4)	6.4	5.0	30.3	10.8	27.4
SwipeUp	44.6 (15.5)	5.0	5.2	27.0	11.6	19.2
SwipeDown	47.2 (16.7)	4.2	7.2	46.1	21.1	29.9
SwipeUpright	45.2 (16.9)	5.4	5.6	17.7	16.6	31.7
SwipeUpleft	44.9 (17.5)	5.5	3.9	14.6	17.0	36.2
SwipeDnright	46.5 (18.8)	6.1	8.1	18.8	15.2	20.9
SwipeDnleft	47.5 (19.0)	7.2	5.2	26.2	37.8	40.3
PokeRight	70.9 (23.0)	4.2	4.5	15.4	11.2	16.5
PokeLeft	74.5 (25.1)	4.4	4.7	16.0	18.4	19.3
PokeUp	72.3 (23.4)	4.9	4.4	23.5	13.0	11.1
PokeDown	71.0 (24.9)	5.0	5.5	18.1	17.4	20.4
Vshape	71.6 (23.7)	4.4	4.9	9.7	16.0	11.4
Xshape	99.3 (28.0)	4.0	4.2	9.1	11.7	13.8
CirHorClk	104.3 (27.0)	3.5	4.2	9.0	7.3	9.1
CirHorCclk	103.1 (30.0)	3.6	4.0	9.0	10.6	7.2
CirVerClk	108.3 (33.0)	3.8	5.8	13.4	8.9	19.2
CirVerCclk	102.4 (32.0)	3.8	6.5	8.4	8.6	13.6
TwistClk	63.2 (18.9)	8.8	3.3	8.6	4.2	6.5
TwistCclk	64.5 (18.9)	11.4	2.7	6.7	4.0	10.0

and temporal variations, and has been successfully applied to gesture recognition [7], [8], sign language recognition [14], [15], and online handwriting recognition [9]. Depending on the tracking technology in use, the features (observations) for the HMMs vary, including the position, moving direction, acceleration, etc. The raw sensor signals may need proper normalization or quantization to handle the variations of gestures, especially in the user-independent case. In our implementation of the HMM-based recognizer, we also propose a normalization procedure specifically for the explicit and implicit motion data.

III. 6DMG: 6D MOTION GESTURE DATABASE

In our hybrid motion tracking system, WorldViz PPT-X4 is used as the optical tracking system, which tracks the positions of infrared dots at 60 Hz. We use the Wii Remote Plus (Wiimote) for the inertial measurement of the acceleration and angular speed. The orientation of the tracking device is then computed from the inertial measurement. We fuse the accelerations, i.e., the indication of gravity, to calibrate the orientation in pitch and roll. Because Wiimote is not equipped with magnetometers, we don't have automatic calibration in yaw. We solve the drifting issue in yaw by manually aligning the controller to the global coordinates and resetting the orientation to identity quaternion periodically. Our experimental results show that the orientation estimation is stable enough for the duration of several gesture recording [16]. We mount an infrared LED at the front of the Wiimote, which works as a position tracker for PPT-X4. Overall, the tracking device provides both explicit and implicit 6D spatio-temporal information sampled at 60 Hz, including the position, orientation, acceleration, and angular speed. For gesture recording, we used a push-to-gesture scheme, which allows the user to explicitly segment the uni-stroke motion gesture, human/user error notwithstanding. We consider the imprecise segmentation as part of the variation of the gesture data.

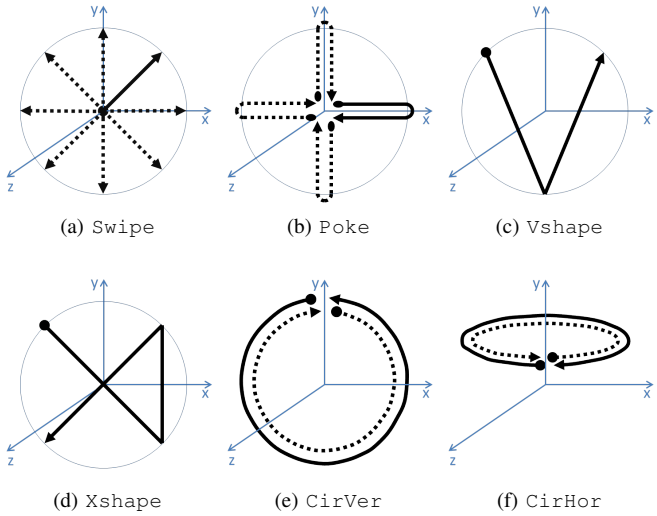

Fig. 2. Selected gestures from the 6DMG database

We define a total of 20 gestures, including swiping motions in eight directions (see Fig. 2a), poke gestures that swipe rapidly forth and back in four directions (Fig. 2b), v-shape, x-shape, clockwise and counter clockwise circles in both vertical and horizontal planes (Fig. 2c-2f), and wrist twisting (roll). Complicated gesture patterns go against the original purpose of intuitiveness and are less favored in the gesture design space [17]. Thus, we decide to make our gesture set simple but still general enough to work in a variety of virtual and augmented reality applications. There are no mirror gestures, which means the direction and rotation are the same for both right- and left-handed users. The names and statistics of the 20 gestures are listed in Table I, and we will explain the columns of normalization ratio later.

We recruited 28 participants (21 right-handed and 7 left-handed, 22 male and 6 female, ranging in age of 15 to 33) for recording. All the participants are undergraduate or graduate students in a university, and at least have gaming experience with Wiimote. Every subject recorded each distinct gesture 10 times, and the 6DMG database has 5600 gesture samples in total. While recording, we advised the subject to perform the gesture in a consistent way, but we did not strictly constrain the gripping posture, the gesture articulation style and speed. Variations of the same gesture between individuals are expected, which make the user-independent recognition challenging. Space limitations preclude the implementation and recording details of 6DMG. The interested reader is referred to [16][1].

IV. GESTURE RECOGNITION

A. Gesture Definitions

It is very important to understand what defines a motion gesture before we do the recognition. In most cases, it is the spatial trajectory that matters. This basically holds true not

[1]The 6DMG database, technical paper, gesture viewer, loader, and exporter are available at http://www.ece.gatech.edu/6DMG

only for our gesture set but also for other existing gestures with 3D spatial or gaming interactions. Exception exists when the spatial trajectory contains little or no deterministic information. For example, the wrist twisting gesture is better described by the change in orientation or the angular speed than the position because the spatial trajectory of wrist twisting is small and varies upon the gripping posture. Therefore, the features for gesture recognition should be primarily extracted from the spatial trajectory and also supplemented with orientation information.

In general, people recognize a gesture by the path spanned by the motion regardless of its speed and scale. Therefore, the recognizer should not be affected by the speed or scale unless fast/slow or big/small motions have different meanings in the gesture set. This is very unlikely to happen especially in user-independent systems because the definition of fast/slow or big/small motions can be vague and different among users.

Let $A^o = [a_x, a_y, a_z]^\top$ denote the device-wise accelerations and $W^o = [w_y, w_p, w_r]^\top$ denote the angular speeds in yaw, pitch, and roll, respectively. From the position data, we can derive P^o and V^o, where $P^o = [p_x, p_y, p_z]^\top$ denotes the positions offset by the starting position, and $V^o = [\Delta p_x, \Delta p_y, \Delta p_z]^\top$ is the rate of change in position. In 6DMG, the orientation is represented in quaternion, $O^o = [q_w, q_x, q_y, q_z]^\top$. Although it is easier to interpret and visualize Euler angles, an Euler representation suffers from gimbal lock and discontinuity when the angle wraps around, and it is numerically less stable near a singularity. The notations above represent the time sequences of a gesture in corresponding coordinates, e.g., $A^o = [a_x(i), a_y(i), a_z(i)]^\top, i = 1, 2, ...N$, where N is the number of samples in a gesture.

B. Feature Extraction and Recognition

In this section, we give the implementation details of two approaches of gesture recognition. The statistical feature-based linear classifier [18] has the advantage of fast and easy implementation with reasonable performance, and it works as our baseline. The hidden Markov model-based classifier [19] is based on a more sophisticated statistical model framework and utilizes the time series nature of motion signals to achieve better performance, especially in the user-independent case.

1) Statistical Feature-Based Linear Classifier: Rubine's feature set was originally designed for 2D trajectories using a mouse or a stylus [12]. Hoffman et al. [13] adapted Rubine's feature set to the 3D domain with an underlying assumption to treat the acceleration and angular speed as position in a 3D space. After a close look at the signals of our inertial sensors, we figure that the "trajectory" in the acceleration space is very jerky and far from the geometric concept that Rubine's feature set was originally designed for. Therefore, we used a running average with a span of five points to smooth the acceleration and angular speed before feature extraction. The tracking results of the position and orientation are much smoother, so filtering is unnecessary.

We first introduce the feature set derived from the spatial trajectory. For simplicity, we use the notation $[x, y, z]^\top$ for the spatial trajectory, which can be either the explicit $[p_x, p_y, p_z]^\top$ or the implicit $[a_x, a_y, a_z]^\top$, and x_i denotes the i_{th} sample in a gesture. The first feature f_1 is the gesture duration. The following features f_{2-13} are the maximum, minimum, mean, and median values of x, y, and z, respectively. f_{14} is the diagonal length of the bounding volume. We first define the step distance and angles in XY and XZ planes:

$$\Delta x_i = x_i - x_{i-1} \quad \Delta y_i = y_i - y_{i-1} \quad \Delta z_i = z_i - z_{i-1}$$
$$d_i = \sqrt{\Delta x_i^2 + \Delta y_i^2 + \Delta z_i^2}$$
$$\theta_i = \arctan\left(\frac{\Delta x_i \Delta y_{i+1} - \Delta x_{i+1} \Delta y_i}{\Delta x_i \Delta x_{i+1} + \Delta y_i \Delta y_{i+1}}\right)$$
$$\gamma_i = \arctan\left(\frac{\Delta x_i \Delta z_{i+1} - \Delta x_{i+1} \Delta z_i}{\Delta x_i \Delta x_{i+1} + \Delta z_i \Delta z_{i+1}}\right)$$

We can derive f_{15-29} relating the angles in XY and XZ planes or the traveled distance as follows,

$$f_{15} = (x_3 - x_1)/\sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2}$$
$$f_{16} = (y_3 - y_1)/\sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2}$$
$$f_{17} = (z_3 - z_1)/\sqrt{(x_3 - x_1)^2 + (z_3 - z_1)^2}$$
$$f_{18} = (x_N - x_1)/\sqrt{(x_N - x_1)^2 + (y_N - y_1)^2}$$
$$f_{19} = (y_N - y_1)/\sqrt{(x_N - x_1)^2 + (y_N - y_1)^2}$$
$$f_{20} = (z_N - z_1)/\sqrt{(x_N - x_1)^2 + (z_N - z_1)^2}$$
$$f_{21} = \sum_{i=2}^{N-1} \theta_i \quad f_{22} = \sum_{i=2}^{N-1} |\theta_i| \quad f_{23} = \sum_{i=2}^{N-1} \theta_i^2$$
$$f_{24} = \sum_{i=2}^{N-1} \gamma_i \quad f_{25} = \sum_{i=2}^{N-1} |\gamma_i| \quad f_{26} = \sum_{i=2}^{N-1} \gamma_i^2$$
$$f_{27} = \sum_{i=2}^{N-1} d_i \quad f_{28} = \max d_i^2$$
$$f_{29} = \sqrt{(x_N - x_1)^2 + (y_N - y_1)^2 + (z_N - z_1)^2}$$

f_{15-16} are the sine and cosine of the starting angle in the XY (vertical) plane, and f_{17} is the sine of the staring angle in the XZ (horizontal) plane. The third sample is chosen empirically based on the average duration of our gesture data to derive the starting angle. f_{18-19} are the sine and cosine of the angle from the first to last point in the XY plane, and f_{20} is the sine of the angle from the first to last point in the XZ plane. After that, f_{21-26} are the total angle traversed, the absolute value and the squared value of that angle in the XY and XZ planes respectively. The last three features f_{27-29} are the total traveled distance, the maximum squared step distance, and the Euclidean distance between the first and the last point.

As for the rotational trajectory, the features for W^o and O^o are slightly different. The angular speed introduces another 12 features, f_{30-41}: the maximum, minimum, mean, and median values of w_y, w_p, and w_r, respectively. For the orientation, we define f_{30-45} as the maximum, minimum, median, and mean values of q_w, q_x, q_y, and q_z.

These features barely contain any temporal information. After a few test runs, we discovered that Hoffman's feature

set leads to confusion between some pairs of gestures like `PokeRight` and `PokeLeft`, `PokeUp` and `PokeDown`, and `CirHorClk` and `CirHorCclk`. Obviously, the time series nature of a motion gesture is crucial for distinction here. We introduce extra features to incorporate the temporal information: the mean values of the first half, the second half, and the center one third of $[a_x, a_y, a_z]^\top$, the mean values of the first half of $[w_x, w_y, w_z]^\top$, and the mean values of the first and the second half of $[p_x, p_y, p_z]^\top$. These features are designed to describe the motion in different time windows at a very coarse time scale. The window selection is empirical and also depends on how complicated the gesture is defined. In general, the time derivative physical quantity usually requires temporal features at a finer scale, which explains why we use more temporal features to describe A^o than P^o and W^o than O^o.

After converting a motion gesture g into a feature vector f, we use a linear classifier for the initial investigation. Associated with each gesture class is a linear evaluation function defined as follows:

$$v_c = w_{c0} + \sum_{i=1}^{F} w_{ci} f_i, 0 \leq c < C, \tag{1}$$

where F is the number of features and C is the total number of classes. The classification of g is the class index c that maximizes v_c. Please refer to [12] for the details on training the weights w_c. The feature-based statistical classifier is presented here as a baseline, which is proven to be simple yet effective in [18]. However, with a full consideration of the spatio-temporal nature of motion gestures, the recognition task can be significantly improved.

2) Hidden Markov Model-Based Classifier: Depending on the available tracking signal, we can use a corresponding set of features (observations) with kinematic meanings for the HMMs, including the position, velocity, acceleration, orientation, and angular speed.

With these features, we represent a motion gesture as a spatio-temporal pattern. Each underlying state in the HMMs actually has a particular kinematic meaning and describes a subset of this pattern, i.e., a segment of the motion. Because the motion gesture is an order-constrained time-evolving signal, the left-right HMM topology is found to be suitable and widely used [7]–[9], [14]. Although we can estimate the transition between hidden states, the physical motion transition is often blurred. If we carefully select the number of states, it is unlikely to skip a segment of the continuous motion when rendering a gesture. Thus, we do not consider skip-transition in the HMM topology.

In order to make the recognizer scale and speed invariant, proper feature normalization is very important. We use the upper case letters without superscript to denote the normalized feature and explain the corresponding normalization procedure as follows. Normalization of P^o, V^o, and W^o is straightforwardly accomplished by uniform linear scaling, i.e., $P = s_p P^o$, $V = s_v V^o$, and $W = s_w W^o$, where the scaling factors are computed as:

$$s_p = \frac{1}{\max[d_x, d_y, d_z]}, \begin{cases} d_x &= \max(p_x) - \min(p_x) \\ d_y &= \max(p_y) - \min(p_y) \\ d_z &= \max(p_z) - \min(p_z) \end{cases} \tag{2}$$

$$s_v = \frac{1}{\max(\|V^o(i)\|)}, i = 1, 2, ...N \tag{3}$$

$$s_w = \frac{1}{\max[\max(|w_y|), \max(|w_p|), \max(|w_r|)]} \tag{4}$$

The scaling factor is determined according to the physical meaning of the normalization target.

The device-wise acceleration A^o is actually a mixture of the acceleration of the gravity and the motion, which provides a very rough estimate of the partial orientation, i.e., pitch and roll. We cannot rescale A^o directly because of the gravity. In [9], the gravitational acceleration is compensated by subtracting the mean of A^o based on the assumption that the sensor heading is constant over the time of one recording. Apparently this does not work in our case because the heading keeps changing during gesture articulation. We have to use extra information, i.e., the orientation, to remove the gravitational acceleration. Given O^o, we first convert the device-wise acceleration to the global coordinates and subtract the constant gravity g as shown in Equation 5.

$$[0, A^g(i)] = \vec{q}_i * \vec{a}_i * \vec{q}_i^{-1} - \vec{g}, \tag{5}$$

where $\vec{q}_i = O^o(i)$, $\vec{a}_i = [0, A^g(i)]$, $\vec{g} = [0, 0, 1, 0]^\top$, and $*$ denotes quaternion multiplication. We then linearly scale A^g to obtain the normalized $A = s_a A^g$, where

$$s_a = \frac{1}{\max(\|A^g(i)\|)}, i = 1, 2, ..N. \tag{6}$$

The normalization of O^o is the most tricky one because the quaternion cannot be "scaled" directly. First, we need to offset (rotate) O^o by the starting orientation so that the first orientation becomes a unit quaternion and then convert the quaternion into the axis-angle representation as

$$[\cos\frac{\alpha_i}{2}, \vec{r}_i \sin\frac{\alpha_i}{2}] = O^o(i) * O^o(1)^{-1}, \tag{7}$$

where α_i is the angle rotated about the axis $\vec{r}_i$ by the right-hand rule. Even though the absolute orientation may provide extra information to distinguish gestures, it is only consistent and usable within a single user. For example, `SwipeRight` rendered by different users can be from the center to right with a rotation angle of 30 degrees or from left to right with an angle of 120 degrees. The concept here is to normalize α and keep the axis untouched, i.e., scale the rotation amount without changing the rotation direction. However, there is one important limitation of the orientation representation: the rotation direction (or angle) is not unique. For example, rotating 0.5π and -1.5π around the same axis $\vec{r}$ have different rotation directions but result in an identical orientation. This is exactly the physical interpretation of $\vec{q}$ and $-\vec{q}$: they are different quaternions but represent the same orientation. Given an orientation, we actually don't know the true rotation amount and the direction to normalize. The ambiguity cannot be resolved unless we keep track of the evolving orientation.

In 6DMG, the orientation starts at identity quaternion when the tracker is pointing forward and facing upright. We update the orientation with a delta rotation at every sampling instant to ensure its continuity. In other words, we are able to track the evolving orientation, and the quaternion represents the true rotation direction and angle in the range from 0 to 2π. The ambiguity still arises when the angle exceeds 2π, but fortunately this never happens in our gesture data. We then scale the rotation angle and compute the normalized orientation as follows,

$$O(i) = [\cos\frac{s_\alpha \alpha_i}{2}, \vec{r}_i \sin\frac{s_\alpha \alpha_i}{2}] \quad (8)$$

$$s_\alpha = \frac{\alpha_{max}}{\max(\alpha_i)}, i = 1, 2, ...N. \quad (9)$$

α_{max} indicates the maximum rotation angle after normalization and is determined empirically. We analyze the distribution of the maximum rotation angles of all gestures in 6DMG. The median, mean, and standard deviation are 0.47π, 0.48π, and 0.18π respectively. Thus, $\alpha_{max} = 0.5\pi$ is considered a reasonable choice here. We actually tested the recognition with two values for α_{max}: 0.5π and π. The performance shows almost no difference, and $\alpha_{max} = \pi/2$ gives an insignificantly better accuracy ($< 0.1\%$).

Given data of the same gesture, the ratio of the maximum over the minimum scaling factor of the normalized features can be a good indicator for "scale" variations. The ratio of each normalized feature set for every gesture in the user-independent case is listed in Table I. In general, the ratios of P and O are much smaller than those of the time-derivative features A, V, and W. The only exception is the ratio of P for twisting gestures, in which we already expect the spatial trajectory is not deterministic. In the user-dependent case, the ratios for all features basically fall under 3, which mean limited variation. Therefore, the normalization process should be more helpful for the user-independent case due to its huge in-class variations as shown in Table I. Note that feature normalization is no elixir. The concept of scale invariance reduces the in-class variations, but it may backfire if the variations between classes are reduced too much at the same time. For example, P^o of `TwistClk` and `TwistCclk` tends to be small in scale, which is an important clue to distinguish them. After normalization, their non-deterministic spatial trajectories are scaled up and may cause confusion with other gestures. In such case, P may be less discriminative than P^o.

V. Experiment Setup and Results

In this section, the evaluation mainly focuses on the HMM-based recognizer, including the effectiveness of normalization and different combinations of feature sets in both user-dependent and user-independent cases. We will also compare the performance with our baseline, the statistical feature-based classifier.

For the HMM-based recognizer, we use the Hidden Markov Model Toolkit (HTK)[2] for modeling, training, and testing. The experiments are done with HMMs of 4, 6, and 8 states with one single Gaussian component per state (in the general mixture density form). The number of states is chosen based on the statistics of the duration of gestures in 6DMG. Although different topologies can be specified per gesture according to its complexity, we use the same topology for all gestures for generality.

For the user-dependent recognition experiment, we form the training set with five tokens randomly drawn from each gesture of a single user, and use the remaining five tokens for testing. We repeat the experiment 50 times for each of the 21 right-handed users for cross validation. For the user-independent case, the training set is formed from the gesture data of randomly selected five right-handed users. We then test with the gestures of the remaining 16 right-handed (UI.R) and 7 left-handed users (UI.L) respectively. This is equivalent to training the recognizer in advance with only right-handers and having new right- or left-handed users simply come in and use the system. The experiment is repeated 200 times to calculate the average recognition rate. We use the same initial seed to randomize the combination of selected training samples so that the results are reproducible and comparable across different feature sets. The size of the training set is chosen intentionally in order to provide an estimate of achievable performance with limited training data, which is motivated by the fact that it is time consuming to collect a lot of gesture data for general users. We separate the right- and left-handed testing sets in order to investigate the handedness on gesture recognition.

A. Evaluation of Normalization

First, we investigate the effectiveness of the normalization of the five basic features (P , V, O, A, and W) in both user-dependent (UD) and user-independent (UI) cases. The average recognition rates from HMMs of 4 states and 8 states are listed in Table II. The UI.R in Table II shows the results of the right-handed testing sets. In the user-dependent case, the normalization only slightly improves the performance (and decreases in A), but it has significant impact in the user-independent case. This actually confirms our hypothesis that the normalization helps when large in-class variations exist. Note that in the normalization of A^o we use O^o to remove the gravity, which means we also remove the embedded partial rotation information. Thus, the recognition rate of A is slightly worse than A^o in the user-dependent case.

In Table I, it is shown that the time-derivative features V, W, and A have higher variation ratios in the user-independent case than features P and O. Thus, they benefit more from the normalization process than P and O as shown in Table II. The performance of P even slightly falls behind P^o in the eight-state HMM case, where the ambiguity caused by the normalized `TwistClk` and `TwistCclk` outweighs the gain of in-class variation reduction. Note that the ratio of O only takes into account the "scaling" of rotation angles and doesn't include the variation of the absolute orientation, i.e., the staring orientation. Therefore, the large improvement of O (from 64.4% to 85.3%) is partially contributed by the orientation offset (+15.0%) and the rotation angle normalization (+5.9%)

[2]The Hidden Markov Model Toolkit (HTK) is available at http://htk.eng.cam.ac.uk/

TABLE II
THE RECOGNITION RATES WITH AND WITHOUT NORMALIZATION OF USER-DEPENDENT (UD) AND USER-INDEPENDENT (UI.R) CASES

		P^o	P	O^o	O	V^o	V	W^o	W	A^o	A
UD	(4 states 1 GMM)	95.88	96.23	97.45	97.51	97.84	97.88	96.78	97.26	97.58	97.03
UD	(8 states 1 GMM)	97.57	97.83	98.54	98.76	98.20	98.54	97.71	98.10	98.54	98.40
UI.R	(4 states 1 GMM)	85.13	87.38	64.42	85.32	73.64	87.65	63.75	75.40	62.15	80.33
UI.R	(8 states 1 GMM)	88.72	88.62	72.55	88.88	82.05	91.31	69.83	80.79	71.51	88.58

with 4 states 1 Gaussian mixture HMMs. Increasing the number of states also gradually improves the performance, but the gain is less prominent for the features that already achieve high accuracy.

Second, we compare the discriminative power of these five basic feature sets. In the user-dependent case, V achieves the highest accuracy (98.5%), and surprisingly P is the lowest (97.8%). In the user-independent case, V still performs the best (91.3%), but W becomes the worst (80.8%). Based on the results, even though the motion gestures are mostly defined by the spatial trajectory, each of the five basic feature sets is effective to a certain degree to distinguish the motion gesture.

B. Evaluation of the Combined Feature Sets

Now we evaluate the recognition performance with different combinations of these basic features, including the explicit spatial 3D (PV), implicit 6D (AW), pure inertial (AWO), explicit 6D (PVO), and complete 6D ($PVOW$ and $PVAOW$). The normalization of A^o actually requires the orientation O^o. Therefore, AWO can be considered the full feature set when only inertial sensors are available, which can be the case in a smartphone. These combinations of features correspond to the possible available tracking signals. The average recognition rates are plotted in Figure 3. After putting together the position and orientation information, either the implicit or explicit 6D feature sets achieves over 99% accuracy in the user-dependent case. In the user-independent case, the performance of the implicit 6D feature set degrades more than the explicit 6D (PO and PVO), and the complete 6D feature sets still have the best accuracy. Adding the orientation to the implicit 6D leads to significant improvement, and AWO should be the best feature set for a pure inertial tracking system based on our findings.

When we combine feature sets of different kinematic meanings, we tie more "constraints" on each HMM state and make it more discriminative. However, the improvement becomes marginal at certain level. In general, adding more HMM states can better capture the time series nature of motion signals and helps to model the motion gesture, but it may suffer from the overfitting problem especially with the combined feature sets.

C. Adaptation to Stripped-Down Motion Tracking

In addition to the combined feature sets above, we also investigate the recognition with limited motion information that reflects a special case of the tracking system in practice: 2D optical tracking when only one camera is used. Let $\bar{P}$ and $\bar{V}$ denote the 2D projection of P and V onto the image plane. For simplicity, we directly truncate the z (depth) component to form $\bar{P}$ and $\bar{V}$. This is very close to placing the camera in the front center of the user with negligible perspective projection. We can derive three new feature sets, $\bar{P}\bar{V}$, $\bar{P}\bar{V}OW$ and $\bar{P}\bar{V}OWA$. We repeat the experiment with these new feature sets and compare the results in Table III. The results of AW and AWO are also listed for comparison. Note that we only show the results of the right-handed testing sets in the user-independent case (UI.R).

In our gesture set, only `HorCirClk` and `HorCirCclk` are not defined on the vertical (XY) plane, but their 2D projections are still distinguishable from other gestures. In some cases, the z dimension is less meaningful and sustains large variations among different subjects. For example, the z increases monotonically if `SwipeRight` is rendered from center to right, but it may decrease then increase if rendered from left to right. Thus, losing the z dimension should not affect the recognition much in our gesture set. In Table III, it is shown that $\bar{P}\bar{V}$-related feature sets performs slightly worse than their full spatial 3D counterparts in general. The only exception is that $\bar{P}\bar{V}$ outperforms PV in the user-independent case. When we only rely on the spatial trajectories, the variation in z can be misleading and degrades the performance. However, the z dimension is still essential if the gesture definition covers the whole 3D space.

D. Leave-One-Out Cross Validation

It is interesting that the accuracy of the left-handed testing set in Figure 3(c) is higher than that of the right-handed testing set in Figure 3(b), even though the recognizer is trained with right-handed data. This may have resulted from the unbalanced size of testing sets (16 right-handed versus 7 left-handed). Based on the results, we assume that our HMM-based recognition of motion gestures is handedness-independent. To verify this assumption, we choose PV, AWO, and $PVOWA$ as the representative feature sets and run leave-one-out cross validation on our whole database.

We have shown that more HMM states can better model the motion gesture and improve the performance in previous experiments. With leave-one-out cross validation, we can have the largest training set from 6DMG, i.e., 270 samples per gesture (27 users$\times$10 trials per gesture), which allows HTK to train more complicated HMMs. Therefore, we further investigate the modeling capability of multiple Gaussian mixtures per state upon our original experiment setting. The average recognition rates of leave-one-out cross validation are shown in Table IV. We make the best result of each column bold. The training processes of AWO with 4 states 3 GMM, 6 states 2 GMM, and 8 states 2 GMM partially fail in HTK, so we mark their results with round brackets. To solve this problem, we have to increase the training data, which is not

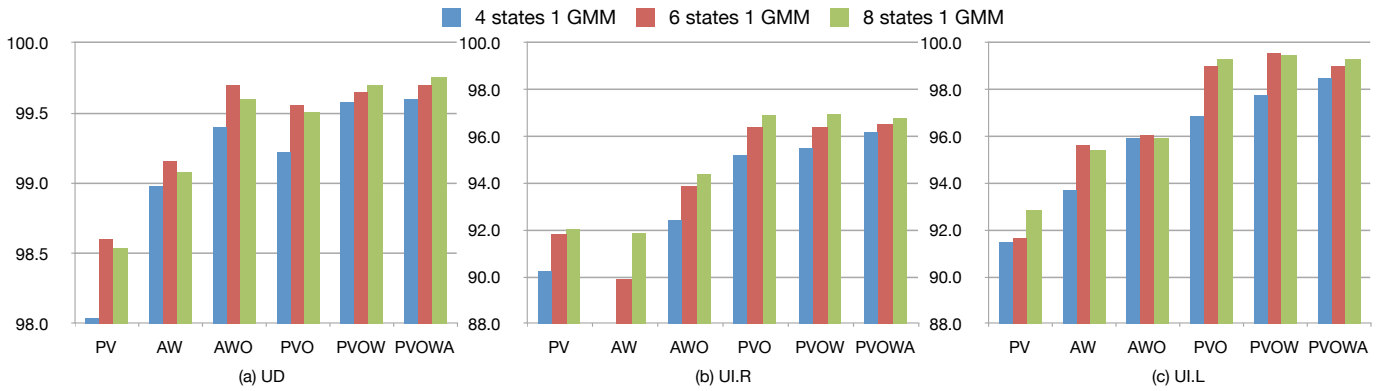

Fig. 3. The recognition rates of combined feature sets. (a) UD: user-dependent case. (b) UI.R: user-independent case on right-handed users. (c) UI.L: user-independent case on left-handed users.

TABLE III
THE RECOGNITION RATES OF COMBINED FEATURE SETS OF USER-DEPENDENT (UD) AND USER-INDEPENDENT (UI.R) CASES

		PV	$\bar{P}\bar{V}$	$PVOW$	$\bar{P}\bar{V}OW$	$PVOWA$	$\bar{P}\bar{V}OWA$	AW	AWO
UD	(4 states 1 GMM)	98.04	97.04	99.58	99.43	99.60	99.47	98.98	99.40
UD	(8 states 1 GMM)	98.54	98.19	99.70	99.73	99.76	99.76	99.08	99.60
UI.R	(4 states 1 GMM)	90.24	92.22	95.52	94.81	96.19	95.80	87.30	92.41
UI.R	(8 states 1 GMM)	92.05	94.46	96.96	96.17	96.80	96.69	91.86	94.38

possible for our current setting. In Table IV, it is shown that the pure inertial feature set AWO outperforms the optical (spatial)-only feature set PV, and the complete feature set $PVOWA$ achieves the best performance. Motion information beyond a spatial trajectory indeed provides additional insight to the motion gesture and improves the performance.

After we separate the leave-one-out testing set by handedness, the left-handed group still has higher average and smaller standard deviation of the recognition rate than the right-handed group for all HMM settings. For example, the average (and standard deviation) of the 21 right-handers and 7 left-handers are 97.76% (3.25%) and 99.57% (0.56%), respectively, with 4 states, 1 GMM , and $PVOWA$. This confirms our assumption that the HMM-based recognizer is handedness-independent, and the performance difference results from the intrinsic variations in our database.

The confusion matrix of the leave-one-out cross validation with 8 states, 1 GMM, and $PVOWA$ is shown in Figure 4. The confusion between g01 (`SwipeRight`), g05 (`SwipeUpright`), and g07 (`SwipeDnright`) reflects the fact that some users tend to render the diagonal gestures very close to `SwipeRight`. Similarly, confusion arises between g02 (`SwipeLeft`), g06 (`SwipeUpleft`), and g08 (`SwipeDnleft`). With the feature set PV, we also notice the confusion between g19 (`TwistClk`) and g20 (`TwistCclk`), e.g., 88.6% and 85.7% accuracy with 8 states and 1GMM, which can be solved by introducing the orientation-based features.

Compared with Table III, we can see that more training data significantly improve the performance for the same HMM structure, i.e. 4, 6, 8 states with single Gaussian mixture. In general, using more states in HMM still improves the recognition rate, but the gain becomes less prominent as can be seen in Table IV. On the other hand, using more Gaussian mixtures

	Recognition Result																			
	g01	g02	g03	g04	g05	g06	g07	g08	g09	g10	g11	g12	g13	g14	g15	g16	g17	g18	g19	g20	acc (%)
g01	267				2		6		4				1								95.4
g02		251				13		14							2						89.6
g03			274		1	4					1										97.9
g04				271								3	4	2							96.8
g05					275		1								4						98.2
g06		21				247		2							10						88.2
g07	3			1			266						7	3							95.0
g08		2						278													99.3
g09	1								276						3						98.6
g10		1								276					1	2					98.6
g11											280										100.0
g12				2								277		1							98.9
g13													280								100.0
g14														280							100.0
g15															268	1	11				95.7
g16															1	275		4			98.2
g17															3		277				98.9
g18														1				279			99.6
g19																1			279		99.6
g20																				280	100.0

Fig. 4. The confusion matrix of leave-one-out cross validation with 8 states, 1 GMM, and $PVOWA$, where g01 to g20 are the gestures from top to bottom in Table I

per state improves the performance when the HMM topology is very simple, i.e., 4 states with single Gaussian mixture. The time series nature of motion gestures is better captured by more states in HMM. Even single Gaussian mixture works well enough to model the probability distribution within each state. When considering the HMM structures of the same number of total Gaussian mixtures, using more states instead of more mixtures per state tends to be a better strategy.

We have shown the effect of HMM structures on motion gesture recognition. The optimal number of states and mixtures per state actually depends on the gesture set. Therefore, fine-tuning the optimal HMM structure should be done on a case-by-case basis and is beyond the scope of this paper.

TABLE IV
THE RECOGNITION RATES OF LEAVE-ONE-OUT CROSS VALIDATION OVER DIFFERENT HMM STRUCTURES

States	GMM	PV	AWO	$PVOWA$
4	1	94.55	96.66	98.21
4	2	95.34	97.36	98.27
4	3	95.63	(93.50)	98.13
6	1	95.80	**97.38**	98.39
6	2	95.52	(93.90)	98.29
8	1	95.73	**97.38**	**98.48**
8	2	**96.05**	(97.09)	98.25

TABLE V
THE COMPARISON BETWEEN THE STATISTICAL FEATURE-BASED LINEAR CLASSIFIER AND THE HMM-BASED RECOGNIZER

	Implicit 6D		Explicit 6D	
	Linear	HMM	Linear	HMM
UD	98.80	99.08	99.59	99.51
UI.R	85.24	91.86	93.51	96.93
UI.L	78.58	95.43	96.99	99.29

E. Comparison with the Baseline

We use the statistical feature-based classifier as the baseline for performance comparison. Due to the defined statistical features, not all combinations of tracking signals are available. The recognition results are obtained with features extracted from either implicit or explicit 6D motion data. In the HMM case, the best corresponding feature sets derived from implicit and explicit 6D are AW and PVO. The comparison between the statistical feature-based linear classifier and the HMM-based recognizer with 8 states and single Gaussian mixture is shown in Table V. In the user-dependent (UD) case, the performance is almost the same. In the right-handed user-independent (UI.R) case, the HMM-based recognizer outperforms by 6.6% for implicit 6D and 3.4% for explicit 6D in the absolute recognition rate.

In the user-independent case with left-handed testing set (UI.L), the HMM-based recognizer still achieves better performance. In general, the left-handed testing set yields higher accuracy than the right-handed set, which probably results from the unbalanced size of testing sets. Note that the linear classifier with implicit 6D features particularly has much worse performance on the left-handed set than the right-handed one. In such case, we postulate that the handedness makes a difference to a certain level for the implicit statistical features.

VI. CONCLUSION

Gestures can be a natural and intuitive way for interaction, and we are especially interested in motion gestures rendered by the hand or handheld device in free space without regard to the posture, finger or body movements. With different tracking technologies, the affordable motion information varies, which can be the position, orientation, acceleration, and angular speed. Although motion gestures are usually defined by the spatial trajectory, other kinematic properties still contain information to distinguish the gestures. Our 6D motion gesture database contains 20 distinct gestures totaling 5600 gesture samples performed by 28 subjects. It records comprehensive motion data, including the position, orientation, acceleration, and angular speed. Thus, 6DMG can be used as a common ground to compare the recognition performance of different tracking signals and methods.

We compare the effectiveness of various features derived from different tracking signals in both user-dependent and user-independent cases. We present two approaches for recognition: the statistical feature-based linear classifier as a simple baseline and the HMM-based recognizer that takes account of the spatio-temporal nature of gesture signals.

In the user-dependent case, both approaches work well with either implicit or explicit 6D data. The user-independent case is more challenging due to the large in-class variations between users. In light of the inherent variation in scale and speed across users, these two factors should be minimized as the differentiating feature in the definition of any gesture. For the HMM-based recognizer, we propose a normalization procedure to alleviate this problem and prove its effectiveness. Unfortunately, some of the statistical features prevent us from applying the same normalization concept, and we let the statistical nature take its course to handle the in-class variations. Overall, the statistical feature-based linear classifier can achieve 85.2% and 93.5% accuracy with implicit and explicit 6D data. The HMM-based recognizer has higher recognition rates, 91.9% and 96.9% respectively. In addition to better performance, the HMM-based recognizer also works with more flexible feature combinations and in general keeps the accuracy above 96%, which means flexibility in choosing the tracking technologies. Based on our results, motion gesture recognition benefits from the complete 6D motion information. Robust motion gesture recognition is achievable even for the challenging user-independent case.

REFERENCES

[1] G. Welch and E. Foxlin, "Motion tracking: no silver bullet, but a respectable arsenal," *Computer Graphics and Applications, IEEE*, vol. 22, no. 6, pp. 24 – 38, Nov 2002.

[2] R. Teather, A. Pavlovych, W. Stuerzlinger, and I. MacKenzie, "Effects of tracking technology, latency, and spatial jitter on object movement," *Proceedings of IEEE Symposium on 3D User Interfaces*, vol. 9, pp. 43–50, 2009.

[3] M. Chen, G. AlRegib, and B.-H. Juang, "Characteristics of spatio-temporal signals acquired by optical motion tracking," in *Signal Processing (ICSP), 2010 IEEE 10th International Conference on*, oct. 2010, pp. 1205 –1208.

[4] A. Godwin, M. Agnew, and J. Stevenson, "Accuracy of inertial motion sensors in static, quasistatic, and complex dynamic motion," *Journal of Biomechanical Engineering*, vol. 131, no. 11, p. 114501, 2009.

[5] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "uwave: Accelerometer-based personalized gesture recognition and its applications," *Pervasive and Mobile Computing*, vol. 5, no. 6, pp. 657 – 675, 2009, perCom 2009.

[6] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS - PART C*, vol. 37, no. 3, pp. 311–324, 2007.

[7] H.-K. Lee and J. H. Kim, "An hmm-based threshold model approach for gesture recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, pp. 961–973, Oct. 1999.

[8] J. Mäntyjärvi, J. Kela, P. Korpipää, and S. Kallio, "Enabling fast and effortless customisation in accelerometer based gesture interaction," in *Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia*, ser. MUM '04, 2004, pp. 25–31.

[9] C. Amma, D. Gehrig, and T. Schultz, "Airwriting recognition using wearable motion sensors," in *Proc. of the 1st Augmented Human Intl. Conf.*, ser. AH '10, 2010, pp. 10:1–10:8.

[10] J. O. Wobbrock, A. D. Wilson, and Y. Li, "Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes," in *Proc. of UIST '07*, 2007, pp. 159–168.
[11] S. Kratz and M. Rohs, "Protractor3d: a closed-form solution to rotation-invariant 3d gestures," in *Proceedings of the 16th international conference on Intelligent user interfaces*, ser. IUI '11, 2011, pp. 371–374.
[12] D. Rubine, "Specifying gestures by example," *SIGGRAPH Comput. Graph.*, vol. 25, pp. 329–337, Jul. 1991.
[13] M. Hoffman, P. Varcholik, and J. LaViola, "Breaking the status quo: Improving 3d gesture recognition with spatially convenient input devices," in *Virtual Reality Conference (VR10)*, Mar. 2010, pp. 59 –66.
[14] T. Starner, J. Weaver, and A. Pentland, "Real-time american sign language recognition using desk and wearable computer based video," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 12, pp. 1371 –1375, dec 1998.
[15] V. Pitsikalis, S. Theodorakis, C. Vogler, and P. Maragos, "Advances in phonetics-based sub-unit modeling for transcription alignment and sign language recognition," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*, june 2011, pp. 1 –6.
[16] M. Chen, G. AlRegib, and B.-H. Juang, "6dmg: A new 6d motion gesture database," in *Proceedings of the third annual ACM conference on Multimedia systems*, ser. MMSys '12, 2012.
[17] J. Ruiz, Y. Li, and E. Lank, "User-defined motion gestures for mobile interaction," in *Proceedings of the 29th international conference on Human factors in computing systems*, ser. CHI '11. ACM, 2011.
[18] M. Chen, G. AlRegib, and B.-H. Juang, "A new 6d motion gesture database and the benchmark results of feature-based statistical recognition," in *Proceedings of the first IEEE conference on Emerging Signal Processing Applications*, ser. ESPA '12, 2012.
[19] ——, "6d motion gesture recognition using spatio-temporal features," in *ICASSP*. IEEE, 2011.

Mingyu Chen received the B.S. degree in electrical engineering from the National Taiwan University, Taipei, Taiwan, in 2005, and the M.S. degree in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, in 2009. He is currently pursuing the Ph.D. degree in electrical and computer engineering, Georgia Institute of Technology.

Since July 2009, he has been with the Center for Signal and Image Processing (CSIP), Georgia Institute of Technology, as a Research Assistant. His research interests include motion tracking, motion recognition, and motion-based human-computer interaction.

Mr. Chen was a recipient of the Studying Abroad Scholarship of Ministry of Education, Republic of China (Taiwan), in 2011.

Ghassan AlRegib is currently Associate Professor at the School of Electrical and Computer Engineering at the Georgia Institute of Technology in Atlanta, GA, USA. His research group is working on projects related to image and video processing and communications, immersive communications, collaborative systems, quality of images and videos, and 3D video processing. Prof. AlRegib is a Senior Member of IEEE. Prof. AlRegib received the ECE Outstanding Graduate Teaching Award in spring 2001 and both the Center for Signal and Image Processing (CSIP) Research Award and the CSIP Service Award in spring 2003. In 2008, he received the ECE Outstanding Junior Faculty Member Award at Georgia Tech. Prof. AlRegib served as the chair of the Special Sessions Program at the IEEE International Conference on Image Processing (ICIP) in 2006. He served as the Area Editor for Columns and Forums in the IEEE Signal Processing Magazine, January 2009 January 2012. He also served as the Associate Editor for the IEEE Signal Processing Magazine (SPM), 2007-2009. He was the Track Chair in the IEEE International Conference on Multimedia and Expo (ICME) in 2011 and the co-chair of the IEEE MMTC Interest Group on 3D Rendering, Processing, and Communications, 2010-present. Prof. AlRegib is a member of the Editorial Board of the Wireless Networks Journal (WiNET), 2009-present. Prof. AlRegib co-founded the ICST International Conference on Immersive Communications (IMMERSCOM) and served as the Chair of the first event in 2007. Since then, Prof. AlRegib serves as the Steering Committee co-Chair of IMMERSCOM. Prof. AlRegib is the founding Editor-in-Chief (EiC) of the ICST Transactions on Immersive Communications to be inaugurated in late 2012. He is also the Chair of the Speech and Video Processing Track at Asilomar 2012.

Biing-Hwang (Fred) Juang is currently the Motorola Foundation Chair Professor and a Georgia Research Alliance Eminent Scholar at Georgia Institute of Technology. He is also enlisted as Honorary Chair Professor at several renowned universities. He received a Ph.D. degree from University of California, Santa Barbara. He conducted research at Speech Communications Research Laboratory (SCRL) and Signal Technology, Inc. (STI) on a number of Government-sponsored research projects. Notable accomplishments during the period include development of vector quantization for voice applications, voice coders at extremely low bit rates, 800 bps and around 300 bps, and robust vocoders for use in satellite communications. He subsequently joined the Acoustics Research Department of Bell Laboratories in 1982, working in the area of speech enhancement, coding and recognition. Prof. Juang became Director of Acoustics and Speech Research at Bell Labs in 1996, and Director of Multimedia Technologies Research at Avaya Labs (a spin-off of Bell Labs) in 2001. He joined Georgia Tech in 2002. Prof. Juang has published extensively, including the book Fundamentals of Speech Recognition, co-authored with L.R. Rabiner, and holds nearly two dozen patents. He received the Technical Achievement Award from the IEEE Signal Processing Society in 1998 for contributions to the field of speech processing and communications and the Third Millennium Medal from the IEEE in 2000. He also received two Best Senior Paper Awards, in 1993 and 1994 respectively, and a Best Paper Awards in 1994, from the IEEE Signal Processing Society. He served as the Editor-in-Chief of the IEEE Transactions on Speech and Audio Processing from 1996 to 2002, and Chair of the IEEE SP Society Fellow Evaluation Committee from 2002 to 2004. He was elected an IEEE Fellow (1991), a Bell Labs Fellow (1999), a member of the US National Academy of Engineering (2004), and an Academician of the Academia Sinica (2006).