A NEW 6D MOTION GESTURE DATABASE AND THE BENCHMARK RESULTS OF FEATURE-BASED STATISTICAL RECOGNITION

Mingyu Chen, Ghassan AlRegib, and Biing-Hwang Juang

School of Electrical and Computer Engineering, Georgia Institute of Technology
Atlanta, Georgia 30332, U.S.A.
{mingyu, alregib, juang}@gatech.edu

ABSTRACT

A motion gesture can be represented by a 3D spatial trajectory and maybe augmented by additional 3 dimensions of orientation. Depending on the tracking technology in use, the 6D motion gesture can be tracked explicitly with the position and orientation or implicitly with the acceleration and angular speed. In this work, we first present a motion gesture database which contains both explicit and implicit 6D motion information. This database allows us to compare the recognition performance over different tracking signals on a common ground. Our main contribution is to investigate the relative effectiveness of various feature dimensions in motion gesture recognition. Using a simple and primitive recognizer, we evaluate the recognition results of both explicit and implicit motion data. In our experiments, both user dependent and user independent cases are addressed. We also propose two general techniques to improve the recognition accuracy: smoothing and the temporal extension. Our pilot study produces benchmark results that give an insight into the attainable recognition accuracy with different tracking devices.

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

1. INTRODUCTION

With the development of tracking technologies, motion-based control and motion gestures are gaining popularity and forming a complementary modality in human-computer interactions beyond the traditional devices. The control motion of conventional pointing devices, such as mouse and trackpad, is limited to trajectories on a plane, which also form the basis of many current motion gesture interface devices. As these new interface modes are meant to support truly natural human computer interactions, they must be designed with 3D in mind. Motion information beyond a 2D trajectory, such as depth and orientation, may provide additional insight into the motion gesture, expand the "vocabulary" of gesture, and improve the accuracy and robustness of gesture recognition. With the help of tracking technologies, we are able to capture the hand motions in space. Therefore, a motion gesture is represented by a 3D spatial trajectory and maybe augmented by the additional three dimensions of orientation, forming what we shall call a 6D motion gesture. In this work, we focus on 6D motion gestures realized by a hand or a handheld device.

There are several technologies for 6D motion tracking, each with its own characteristics in terms of sampling rate, latency, resolution, and accuracy. Among them, optical sensing and inertial sensing are the most popular. The optical sensing usually tracks the explicit 6D motion, i.e., the position and orientation in a global reference frame. The inertial sensing actually measures the accelerations and angular speeds in the device-wise coordinates, which depict the implicit 6D motion. It is possible to infer the displacement in position and orientation through integration, although not as accurate as the explicit 6D motion from optical tracking.

Depending on the tracking technology in use, the motion gesture is represented in different dimensions of tracking results, including the spatial trajectory with or without the three dimensions for orientation. The spatial trajectory can be either 3D or its 2D projection, and the tracking results can be explicit or implicit as described above. Motion gestures can be viewed as spatio-temporal patterns of different dimensions, and the recognition is widely done with hidden Markov models [1, 2, 3]. Other approaches for gesture recognition include dynamic time warping [4], data-driven template matching [5, 6], and feature-based statistical classifiers [7, 8]. The reported recognition rates are above 90% in general. However, it is hard to compare the performance because the results are obtained for different datasets and in various experimental settings.

In this work, we first present a 6D motion gesture database (6DMG) which contains both explicit and implicit 6D motion information. We are interested in understanding which type of tracking signals and resulting 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 feature dimensions in motion gesture recognition. We use a simple and primitive linear classifier to evaluate the resulting recognition rates of both explicit and implicit motion data. Similar to the case of speech recognition, it is desirable that the recognition system accommodates user-specific adaptation or customization, but it is also very important to achieve robust user-independent recognition. Both user-dependent and user-independent cases are addressed. We further show the gain in the recognition rate when including the temporal characteristics of motion gestures. Our pilot study produces benchmark results that give an insight into the attainable recognition accuracy with different tracking devices.

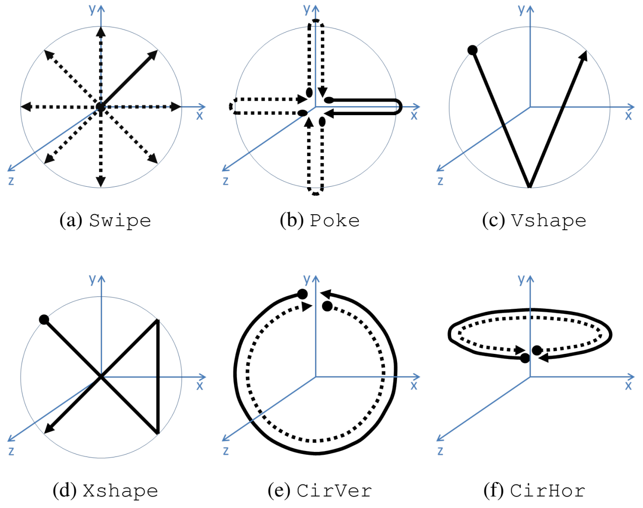
The 6D motion gesture database is presented in the next section. We describe the feature set extracted from different tracking signals in Section 3. The experiments and results are shown in Section 4. Finally, Section 5 concludes the paper.

2. 6DMG: 6D MOTION GESTURE DATABASE

We use a hybrid framework of optical and inertial sensing for motion tracking. Thus, the recorded data contain comprehensive spatio-temporal information sampled at 60 Hz, including position, orientation, acceleration, and angular speed. We use WorldViz PPT-X4 as

Table 1: The gesture list of 6DMG

Name	Duration (ms) Avg. (std.)	Name	Duration (ms) Avg. (std.)
`SwipeRight`	866.2 (345.2)	`PokeUp`	1206.4 (389.9)
`SwipeLeft`	861.1 (340.7)	`PokeDown`	1183.6 (415.6)
`SwipeUp`	743.6 (258.7)	`Vshape`	1193.5 (394.5)
`SwipeDown`	787.9 (277.8)	`Xshape`	1655.2 (466.1)
`SwipeUpright`	754.5 (282.7)	`CirHorClk`	1738.4 (449.2)
`SwipeUpleft`	748.9 (291.4)	`CirHorCclk`	1719.9 (500.5)
`SwipeDnright`	777.1 (313.9)	`CirVerClk`	1806.5 (549.3)
`SwipeDnleft`	792.4 (317.7)	`CirVerCclk`	1707.7 (532.1)
`PokeRight`	1181.6 (383.3)	`TwistClk`	1054.8 (315.5)
`PokeLeft`	1242.4 (418.4)	`TwistCclk`	1075.9 (315.3)



(a) `Swipe` (b) `Poke` (c) `Vshape`

(d) `Xshape` (e) `CirVer` (f) `CirHor`

Fig. 1: Selected gestures in 6DMG

the optical tracking system and the Wii Remote Plus (Wiimote) as the inertial measurement unit. The Wiimotes B button is used for the push-to-gesture scheme so that the user explicitly segments the uni-stroke motion gesture. We consider the imprecise segmentation as part of the variation of the gesture data.

Swiping motions in eight directions as shown in Figure 1a are viewed as the basic elements to form other complex gestures . We also define a group of poke gestures that swipe rapidly forth and back in four directions (see Figure 1b). Other commonly used motion gestures such as circle, cross, v-shape (Figure 1c-1f), and roll are also included. The names and durations of the 20 gestures are listed in Table 1. There are no "mirror" gestures, which means the direction and rotation are the same for both right and left handed users.

We recruited 28 participants (21 right-handed and 7 left-handed, 22 male and 6 female) for recording. Every tester was asked to repeat each distinct gesture for 10 times. There are in total 5600 gesture samples in the 6DMG database. When recording, we did not strictly constrain the gripping posture, the gesture articulation style and the speed. Variations of the same gesture between individuals are expected, and recording motion gestures from different users ensures the in-class variability of 6DMG. Table 1 shows the variation in gesture articulation speeds. Space limitations preclude the implementation and recording details of 6DMG. The interested reader is referred to our technical report[1].

3. FEATURE EXTRACTION

3.1. The Baseline

Rubine's feature set was originally designed for 2D trajectories using the mouse or stylus [7]. With an underlying assumption to treat the acceleration and angular speed data as position information in a 3D space, Hoffman et al. [8] adapted Rubine's feature set to the 3D domain of the implicit 6D motion data.

Let $[a_x, a_y, a_z]$ denote the accelerations and $[w_x, w_y, w_z]$ denote the angular speeds in yaw, pitch, and roll respectively. In Hoffman's set, the first feature f_1 is the gesture duration in milliseconds. The following features f_{2-13} are the maximum, minimum, mean, and median values of a_x, a_y, and a_z. f_{14-16} are the sine and cosine of the starting angle in the XY (vertical) plane and the sine of the starting angle in the XZ (horizontal) plane. f_{17-19} are the sine and cosine of the angle from the first to last point in the XY plane and the sine of the angle from the first to last point in the XZ plane. After that, f_{20-25} are the total angle traversed, the absolute value and squared value of that angle in the XY and XZ planes respectively. The last four features f_{26-29} for accelerations are the diagonal length of the bounding volume, the Euclidean distance between the first and the last point, the total traveled distance, and the maximum squared delta acceleration. The angular speeds introduce another 12 features, f_{30-41}: the maximum, minimum, mean, and median values of w_x, w_y, and w_z. For detailed implementation, please refer to Rubine's work [7].

3.2. Extension to Explicit 6D

Our database also provides explicit position and orientation data. Note that the orientation is represented in quaternion, which can be spherical linearly interpolated without gimbal lock. Although it is easier to interpret or visualize Euler angles, an Euler representation suffers from discontinuity when the angle wraps around, and it is numerically less stable near a singularity.

Let $[p_x, p_y, p_z]$ denote the positions offset by the starting position, and $[q_w, q_x, q_y, q_z]$ denote the quaternion of the orientation. It is very straightforward to extend the feature set above to real position and orientation data. For f_{1-29}, simply replace $[a_x, a_y, a_z]$ with $[p_x, p_y, p_z]$. For the orientation part, we define f_{30-45} as the maximum, minimum, median, and mean values of q_w, q_x, q_y, and q_z. Thus, 29 features are used to describe the motion gestures with only positions, and 45 features are used when we include the orientation.

3.3. Smoothing

Compared to planar pointing devices, 3D input devices generally have higher tracking noise, and are subject to hand tremor if held in space. In our hybrid tracking framework, the raw readings from the MEMS inertial sensors are even noisier than the measurements from the optical tracking system. This could have mainly resulted from the characteristics of the tracking hardware in use. In [3], the accelerometer and gyroscope data of the hand motion from an Analog Devices ADIS16364 Inertial Measurement Unit are comparably smooth at 819.2 Hz. The minimum-jerk model also suggests that a skilled motion is characterized by a decrease of jerk magnitude [9].

[1] The 6DMG database, technical report, and accompanying programs are available at `http://www.ece.gatech.edu/6DMG`.

After a close look at the signals of our inertial sensors, we figure that the assumption of taking accelerations and angular speeds as spatial trajectories is weak. The "trajectory" in the acceleration space is very jerky and far from the geometric concept that Rubine's feature set is originally designed for. Therefore, the jitter will mess up the angle-related features f_{14-25} and makes them less discriminative. An intuitive remedy is to process the data with a running average before feature extraction. Note that the smoothing is mainly effective to angle-related features of accelerations. In implementation, we use a running average with a span of 5 points. At the signal processing perspective, the smoothness in time domain suggests that the energy distribution of human motions mainly concentrates in the low frequency band. The time derivative operator actually boosts the spectrum proportional to the frequency and magnifies the unwanted parts. Taking the running average works as a low-pass filter to retain the signature of the gesture.

3.4. Incorporating the Temporal Information

These statistical features in Hoffman's set are either geometric or algebraic. They treat the gesture as a static trajectory. The features take no account of the ordering of angles in a trajectory and barely contain any temporal information. Only f_{14-21} carry directional information. Given a symmetric trajectory, swapping the start and end points still forms the same path. In such circumstances, only the directional features (f_{14-21}) can be discriminative.

Currently our 3D tracking device produces noisier data than a working 2D input device; as a result, the angle related features are not particularly reliable. Gesture motions in space are also harder to articulate precisely with a clean segmentation than those constrained on a 2D plane. For example, early or late start of the push-to-gesture scheme can mess up the starting angle. Overshoot can cause problems for the start-to-end angle. If the start and end of the input gesture are not clearly delineated, the directional features become less reliable.

After a few test runs using the linear classifier, 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`. Apparently, the time series properties of a motion gesture are crucial to discriminate them. We introduce extra features to incorporate temporal information into the modified feature set: the mean values of the first half, second half, and the center one third of $[a_x, a_y, a_z]$, the mean values of the first half of $[w_x, w_y, w_z]$, and the mean values of the first and second halves of $[p_x, p_y, p_z]$. These features are able to describe the motions in different time windows at a very coarse scale. The selection of temporal windows depends on how fine (or coarse) the scale we need to distinguish the gesture. Taking time derivative usually means that we need temporal features at a finer scale. Therefore, we use more temporal features to describe accelerations than positions and angular speeds than orientation. It also depends on how complicated the gesture motions are defined. Our temporal features are determined empirically. They are simple yet effective.

4. EXPERIMENT SETUP AND RESULTS

After converting a motion gesture g into a feature vector $\mathbf{f}$, we use a linear classifier for recognition. Associated with each gesture class is a linear evaluation function over the features as follows,

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

where F is the number of features, and C is the total number of classes. The classification of g is the c that maximizes v_c. Refer to [7] for the details of training the weights w_c. The threshold for rejection depends on the feature set in use and requires empirical tuning. In our case, all the gesture samples are rendered intentionally with labels. Thus, it's reasonable not to consider the case of rejecting gestures. We conducted experiments on both user dependent and user independent cases. The relationship between right-handed and left-handed gestures is also investigated. In every experiment, we evaluate the average error rate of the baseline as well as the cases with smoothing and/or the temporal extension over implicit and explicit motion data. All 20 gestures are used to evaluate the recognition performance. We use the same initial seed to randomize the combination of selected training samples so that the results are reproducible and comparable across different settings.

4.1. User Dependent Recognition

For the user dependent recognition experiment, we train the classifier with 5 samples randomly drawn from each gesture of a single user, and use the remaining 5 samples for testing. We repeat the experiment for each of the 21 right-handed users. The results are shown in Figure 2 Exp 1. The baseline with only the acceleration feature has the highest mean error rate, 3.92%. The lowest mean error rates for implicit and explicit 6D data are 1.20% and 0.41% respectively. We show that a high level of accuracy is attainable with only 5 training samples per gesture for a specific user.

4.2. User Independent Recognition

We randomly select five right-handed users, and train the classifier with their gesture data. We then perform recognition on the gestures of the remaining 16 right-handed users and 7 left-handed users respectively. This case is equivalent to training the recognizer in advance and having new users simply come in and use the system. Each setting is repeated 200 times to compute the mean error rate. Figure 2 Exp 2.1 and 2.2 show the recognition results of right-handed and left-handed data.

The error rate exceeds 30% in the worst case (acceleration-only). For the right-handed testing data, the best mean error rates for implicit and explicit 6D motions are 14.76% and 6.49%. The confusion matrix shows that the diagonal swiping gestures are noticeably ambiguous. This is due to the fact that some of the subjects rendered the diagonal swiping gestures close to their horizontal swiping motions; e.g., `SwipeUpleft` or `SwipeDnleft` is realized very close to `SwipeLeft`. If we remove the four diagonal gestures, the best mean error rates can be improved to 7.13% and 4.20%. In Figure 2 Exp 2.2, the steep drop of error rates of the explicit motion features may have resulted from fewer left-handed testing samples.

The mean error rate of the user independent case is about 5 to 10 times higher than that of the user dependent case. In general, the feature set of positions and orientations works best. The position-only features are second, followed by the feature set of accelerations and angular speeds. The acceleration-only feature performs worst. Both smoothing and the temporal extension improve the recognition rate, but the latter is more effective. We can further reduce the error rate by integrating them together.

4.3. Verifying with Hoffman's Data Set

It is interesting to compare the performance with different data set. First, we try to reproduce Hoffman's experiments with his own data

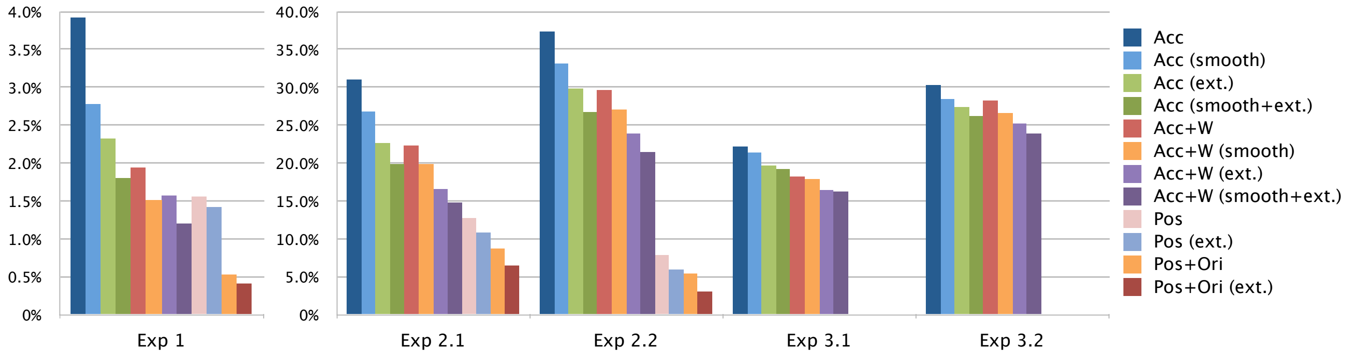


Fig. 2: The mean error rates of (Exp 1) user dependent recognition on 6DMG, (Exp 2.1 and 2.2) user independent recognition on 6DMG, and (Exp 3.1 and 3.2) user dependent recognition on Hoffman's set. The legends Acc, Acc+W, Pos, and Pos+Ori mean acceleration-only, acceleration plus angular speed, position-only, and position plus orientation respectively. The bracket indicates the modification in use, i.e., smoothing and/or temporal extension.

and settings[2], and the results confirm with the reported numbers in [8]. Then, we run experiments on Hoffman's data set with the same setup in Section 4.2. Hoffman's data set only records the motion gestures using accelerometers and gyroscope in the Wii Remote and Motion Plus. His gesture set originally has 25 gestures performed by 12 right-handed and 5 left-handed users. We have to exclude the "mirror" gestures that are opposite between right-handed and left-handed users: `TennisSwing`, `GolfSwing`, `Parry`, `Lasso`, and `Spike`. We randomly select five right-handed users to train the classifier, and then perform recognition on the gestures of the remaining 7 right-handed users and 5 left-handed users respectively. Figure 2 Exp 3.1 and 3.2 show the results of right-handed and left-handed gestures.

We prove that both smoothing and the temporal extension are still effective, although the reduction in the error rate is less than that in our data set. We run the statistical hypothesis test on left-handed and right-handed results of the implicit 6D gestures. It supports that the mean error rates are significantly different ($p < 0.05$ for both Exp 2 and 3). Based on both 6DMG and Hoffman's data sets, we postulate that the right-handed and left-handed gestures are different to a certain degree even with the same gesture definition. However, we need more left-handed motion data to prove this speculation.

5. CONCLUSION

In this work, we first introduce 6DMG, a motion gesture database of both implicit and explicit 6D motion information of 20 distinct gestures. Hoffman's statistical feature set is used as our baseline. We extend it from accelerations and angular speeds to positions and orientations. We also propose two techniques to improve the recognition accuracy: smoothing and the temporal extension. Smoothing works as a low-pass filter to combat with the noisy data from our inertial sensors. The temporal extension compensates the shortcomings of the statistical features that take no account of the time series nature of motions. After extracting the features, the linear classifier is used to recognize the motion gestures. We examine both user dependent and user independent recognition configurations.

Based on our results, the real positions provide much better accuracy than the accelerations. The motion gestures are mainly about spatial trajectories, but the rotational information can provide supplementary cues to further boost the recognition rate. Both smoothing and the temporal information improve the recognition performance, but the latter is more effective. Even with the same gesture definition, the difference in right-handed and left-handed motions matters for the gesture recognition. We show that the temporal information is crucial for motion gesture recognition. A more thorough time series analysis on motion gestures is needed. In the future, we hope to further improve the recognition accuracy and replace the push-to-gesture scheme with automatic gesture spotting.

6. REFERENCES

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

[2] Jani Mäntyjärvi, Juha Kela, Panu Korpipää, and Sanna Kallio, "Enabling fast and effortless customisation in accelerometer based gesture interaction," in *Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia*, 2004, MUM '04, pp. 25–31.

[3] Christoph Amma, Dirk Gehrig, and Tanja Schultz, "Airwriting recognition using wearable motion sensors," in *Proc. of the 1st Augmented Human Intl. Conf.*, 2010, AH '10, pp. 10:1–10:8.

[4] Jiayang Liu, Lin Zhong, Jehan Wickramasuriya, and Venu Vasudevan, "uwave: Accelerometer-based personalized gesture recognition and its applications," *Pervasive and Mobile Computing*, vol. 5, no. 6, pp. 657 – 675, 2009, PerCom 2009.

[5] Jacob O. Wobbrock, Andrew D. Wilson, and Yang Li, "Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes," in *Proc. of UIST '07*, 2007, pp. 159–168.

[6] Sven Kratz and Michael Rohs, "Protractor3d: a closed-form solution to rotation-invariant 3d gestures," in *Proceedings of the 16th international conference on Intelligent user interfaces*, 2011, IUI '11, pp. 371–374.

[7] Dean Rubine, "Specifying gestures by example," *SIGGRAPH Comput. Graph.*, vol. 25, pp. 329–337, Jul. 1991.

[8] M. Hoffman, P. Varcholik, and J.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.

[9] T Flash and N Hogan, "The coordination of arm movements: an experimentally confirmed mathematical model," *The Journal of Neuroscience*, vol. 5, no. 7, pp. 1688–1703, 1985.

[2] We would like to thank Michael Hoffman for sharing his motion gesture data set and the loader program.