The impact of motion dimensionality and bit cardinality on the design of 3D gesture recognizers

Radu-Daniel Vatavu*

University Stefan cel Mare of Suceava, str. Universitatii nr. 13, 720229 Suceava, Romania

Received 17 May 2012; received in revised form 10 November 2012; accepted 26 November 2012

Communicated by E. Motta
Available online 12 December 2012

Abstract

The interactive demands of the upcoming ubiquitous computing era have set off researchers and practitioners toward prototyping new gesture-sensing devices and gadgets. At the same time, the practical needs of developing for such miniaturized prototypes with sometimes very low processing power and memory resources make practitioners in high demand of fast gesture recognizers employing little memory. However, the available work on motion gesture classifiers has mainly focused on delivering high recognition performance with less discussion on execution speed or required memory. This work investigates the performance of today’s commonly used 3D motion gesture recognizers under the effect of different gesture dimensionality and bit cardinality representations. Specifically, we show that few sampling points and low bit depths are sufficient for most motion gesture metrics to attain their peak recognition performance in the context of the popular Nearest-Neighbor classification approach. As a practical consequence, 16x faster recognizers working with 32x less memory while delivering the same high levels of recognition performance are being reported. We present recognition results for a large gesture corpus consisting in nearly 20,000 gesture samples. In addition, a toolkit is provided to assist practitioners in optimizing their gesture recognizers in order to increase classification speed and reduce memory consumption for their designs. At a deeper level, our findings suggest that the precision of the human motor control system articulating 3D gestures is needlessly surpassed by the precision of today’s motion sensing technology that unfortunately bares a direct connection with the sensors' cost. We hope this work will encourage practitioners to consider improving the performance of their prototypes by careful analysis of motion gesture representation rather than by throwing more processing power and more memory into the design.

© 2012 Elsevier Ltd. All rights reserved.

Keywords: Gesture recognition; Gesture dimensionality; Sampling rate; 3D gestures; Classifiers; Bit cardinality; Bit depth; Euclidean distance; Angular cosine distance; Dynamic time warping; Hausdorff; Gesture toolkit

1. Introduction

The recent availability of low-cost motion sensing technology embedded in mobile devices (Lane et al., 2010) has led to a wide proliferation of systems and applications employing gesture commands (Li, 2009; Liu et al., 2009; Ni and Baudisch, 2009; Ruiz and Li, 2011; Zhai et al., 2009). Nowadays, user-interface practitioners and designers have at their disposal a wide range of devices able to sense motion: mobile phones (Hinckley et al., 2000; Murao et al., 2011; Rekimoto, 1996), game controllers (Hoffman et al., 2010; Lee, 2008), and even wrist watches (Kim et al., 2007). Practitioners also benefit of a large selection of machine learning algorithms for recognizing gestures. These include Nearest-Neighbor (NN) classifiers that work with various gesture metrics (Anthony and Wobbrock, 2010; Kratz and Rohs, 2010, 2011; Li, 2010; Vatavu et al., 2012a; Wobbrock et al., 2007) but also more elaborate approaches such as Hidden Markov Models (HMMs) (Schlömer et al., 2008), Support Vector Machines (SVMs) (Wu et al., 2009), and Adaptive Boosting (Hoffman et al., 2010).

When considering the practical needs for developing and using such gestural interfaces, the NN classification approach stands out among its peer techniques for reasons such as ease...
of implementation for practitioners and ease of customization for users. The technique is simple to understand, implement, and debug by a practitioner not particularly interested in mastering all the complex details of more elaborate machine learning procedures. For such reasons, a $S$-family of gesture classifiers ($S_1, S_N, S_P$) has been proposed in the human–computer interaction community to assist designers and practitioners implementing gesture recognition on new platforms (Anthony and Wobbrock, 2012; Li, 2010; Vatavu et al., 2012a; Wobbrock et al., 2007). As for the advantages for users, new commands can be easily added to the gesture set without the need to retrain or change the inner structure of the recognizer as would be the case for learning new state transition probabilities for HMMs (Schlömer et al., 2008), support vectors for SVMs (Wu et al., 2009), or weights for neural networks (Bailador et al., 2007).

The Nearest-Neighbor technique has been successfully used to classify gestures with near 99% accuracy while employing the Euclidean distance (Kratz and Rohs, 2010; Wobbrock et al., 2007), angular cosine similarity (Anthony and Wobbrock, 2012; Kratz and Rohs, 2011; Li, 2010), dynamic time warping (Liu et al., 2009; Wobbrock et al., 2007), and minimum-cost point cloud alignments (Vatavu et al., 2012a). However, besides recognition rate, the performance of a classifier is also judged by its execution speed and memory requirements. In the NN approach, both execution time and required memory depend directly on the representation adopted for gestures in terms of number of sampling points (gesture dimensionality) and precision of the measurement process (gesture bit cardinality). These factors become critical as sensing gradually disappears into the ambient through miniaturization (Ni et al., 2010) forcing designers to optimize execution time and minimize memory consumption for devices with sometimes extremely limited resources.

To discuss just one such example, eZ430-Chronos from Texas Instruments\(^1\) is a wrist watch that can capture accelerated motion with its embedded 3-axis accelerometer, store data in its 32 KB of flash memory, and process it with a 20 MHz 16-bit microcontroller. However, in order to store a gesture set such as the one from (Hoffman et al., 2010) with enough training samples to assure robust recognition, a minimum of 94 KB would be needed\(^2\) which is three times the memory of the device! Therefore, even in the age of practically unlimited amounts of memory that get cheaper by the day and 1 GHz processing available on mobile devices,\(^3\) the particular attention to data representation as manifested since the early days of computing (Agerwala, 1976; Das and Nayak, 1990) is still actual.

Data dimensionality and bit cardinality are also closely related to other design decisions that practitioners need to take, especially when implementing functions in dedicated hardware. Implementing functions in hardware (classifiers included) represents sometimes the last remaining option for designers to speed-up their code. For example, Sart et al. (2010) argue that software optimization ideas for dynamic time warping are close to exhaustion and therefore any new enhancement would come from moving computations on dedicated hardware such as GPUs (Graphics Processing Units) and FPGAs (Field-Programmable Gate Arrays). As a result, designers have already started to consider such options for processing human motion such as the FPGA data glove design of Park et al. (2008). Also, besides keeping memory consumption low, practitioners of dedicated hardware are interested in the bit depth of their architectures in order to reduce consumed power (Mallik et al., 2006), minimize circuit area (Lee et al., 2005), and reduce latency in their designs (Zhang et al., 2010).

Despite such important connections between gesture representation and system performance, there is no study investigating the performance of 3D gesture classifiers under various sampling rates (gesture dimensionality) and bit depths (gesture bit cardinality). However, we argue that such a study would be useful in providing assistance to practitioners in optimizing their specific designs. In the lack of such information, prototypers have been experimenting different options for their designs with the result of very different gesture representations being reported which may confuse a newcomer to the field (see Table 1 illustrating a few examples). Although an important topic with practical implications, the fundamental problem of finding the intrinsic representation of motion data has been only marginally addressed by researchers. In this line of work, the Protractor gesture classifier (Li, 2010) used a reduced dimensionality to optimize the original $S_1$ recognizer (Wobbrock et al., 2007). Recognition experiments reported in Vatavu (2011) showed that low data dimensionality can still deliver high recognition accuracy but for 2D motions only. Working on time series, Bagnall et al. (2006), Rakthanmanon et al. (2011), Xi et al. (2006) showed that data mining algorithms can benefit from reduced bit cardinality in representing their data and Hu et al. (2011) employed the Minimal Description Length (MDL) framework to investigate the natural intrinsic representation of time series in terms of approximation model, dimensionality, and alphabet cardinality. Building on such previous works, Rakthanmanon et al. (2012) exploited lower bounding and early abandoning techniques (Keogh et al., 2009) to search in trillions of data points fast and accurately. Vatavu (2012) explored the bit depth of point-based gesture representations and found that 2D motions can be represented using lower bit cardinalities without affecting recognition rate considerably. However, understanding the true dimensionality and bit cardinality of 3D gesture data is still unanswered despite the important

\(^{1}\)http://www.ti.com/tool/ez430-chronosDCMP=ChronosHQSOther-OTChronos

\(^{2}\)25 gestures $\times$ 10 training samples $\times$ 64 points $\times$ 3 channels ($x, y, z$) $\times$ 2 bytes per channel $=96,000$ bytes. The value of 64 sampling points is suggested by Wobbrock et al. (2007) for 2D gestures while Kratz and Rohs (2010) used 150 points for 3D motions.

\(^{3}\)Such as Apple’s A4 and A5 processors for iPhone\(^6\) and iPad\(^7\).
implications for practitioners prototyping the way toward the interactive gadgets of ubiquitous computing.

This work is the first investigation of the impact that gesture dimensionality and bit cardinality have on the performance of 3D gesture recognizers. We provide empirical evidence that few sampling points and low bit depths are enough for most gesture metrics to attain peak recognition performance with the Nearest-Neighbor classification technique. We do so by computing recognition rates on a large gesture corpus (≈ 20,000 samples) for both user-dependent and user-independent training scenarios. Specifically, we found eight sampling points more than sufficient for Euclidean and Cosine metrics to deliver high recognition performance while a linear relationship was detected between gesture dimensionality and the size of the gesture set for dynamic time warping. Also, only 3–5 bits per x, y, z gesture channels were found to provide sufficient representation resolution for the tested metrics to deliver their highest level of recognition accuracy. In turn, the impact on execution time and memory requirements is considerable. We report 16x faster recognizers needing 32x less memory while still delivering high level recognition performance.

In addition to our empirical findings, we propose a mathematical model for explaining the effect of gesture dimensionality and bit cardinality on recognizers’ performance. A toolkit is also provided to assist practitioners in optimizing the sampling rate and bit depths of their 3D gesture designs. We validate our toolkit on gesture sets that are publicly available in the community.

We believe the contributions of this work will impact the community of practitioners of gesture-based interfaces with the following implications:

(1) Inform performance-oriented design: Low memory requirements are sometimes inevitable (e.g., the motion sensing wrist watch example) and practitioners need to reduce memory consumption of their designs. However, besides memory reduction, small dimensionalities and low bit depths can be exploited by several hardware architectures such as FPGAs in order to increase parallelism, reduce circuit area and power consumption (Kinsman and Nicolic, 2010; Lee et al., 2005; Mallik et al., 2006).

(2) Inform cost-oriented design: Practitioners can make an informed decision about the precision of sensing they actually need for their application (e.g., deciding whether to include an 8-bit instead of a 12-bit or 16-bit precision accelerometer into the design can help reducing the total cost). For example, practitioners developing with Phidgets⁴ (Greenberg and Fitchett, 2001) may have to decide whether they need 16 bits of precision for sensing accelerated motion⁵ or just 12 bits at half price.⁶

(3) Foster exploration of new software architectures for gestures: New software architectures have started to emerge for gestures providing practitioners with web services that deliver gesture recognition (Kohlsdorf et al., 2011; Van Seghbroeck et al., 2010; Vatavu et al., 2012b) or gesture suggestions from large databases collected by exploiting the “wisdom of the crowd” (Ouyang and Li, 2012). Such architectures emerging from the practice of Service-Oriented Computing⁷ reduce programming load and encourage code reusability across new platforms. However, in order for such designs to become practical, network bandwidth for transferring gesture data needs to be minimized. Therefore, careful analysis of gesture representation in terms of dimensionality and bit depth is needed.

(4) Promote conscientious design: An important conclusion of this work is that throwing more resolution at a problem won’t necessarily improve accuracy. Instead, employing more resolution than needed may even have negative effects such as more memory and power consumption and even lower recognition performance in some cases (Sima and Dougherty, 2008). We hope this work will encourage practitioners to consider improving the performance of their prototypes by

---

Table 1

<table>
<thead>
<tr>
<th>Recognizer</th>
<th>Dimensionality</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$ (Wobbrock et al., 2007)</td>
<td>64 points</td>
<td>2D Euclidean</td>
</tr>
<tr>
<td>Protractor (Li, 2010)</td>
<td>16 points</td>
<td>2D Cosine</td>
</tr>
<tr>
<td>SHARK’s proportional shape distance (Cao and Zhai, 2007; Kristensson and Zhai, 2004)</td>
<td>25 points</td>
<td>2D Euclidean</td>
</tr>
<tr>
<td>SN (Anthony and Wobbrock, 2010)</td>
<td>96 points</td>
<td>2D Euclidean</td>
</tr>
<tr>
<td>SN-Protractor (Anthony and Wobbrock, 2012)</td>
<td>96 points</td>
<td>2D Cosine</td>
</tr>
<tr>
<td>$S_3$ (Kratz and Rohs, 2010)</td>
<td>150 points</td>
<td>3D Euclidean</td>
</tr>
<tr>
<td>Protractor 3D (Kratz and Rohs, 2011)</td>
<td>32 points</td>
<td>3D Cosine</td>
</tr>
<tr>
<td>uWave (Liu et al., 2009)</td>
<td>Variable</td>
<td>3D DTW</td>
</tr>
</tbody>
</table>

---

careful analysis of gesture representation rather than attempting to do so by throwing more processing power and more memory into the design.

2. Gesture preliminaries

This section introduces the main concepts used throughout the article such as gesture dimensionality and bit cardinality and briefly discusses the metrics employed for the recognition experiments. We understand by gesture a set of 3D points ordered by their acquisition time

\[ p_i = (x_i, y_i, z_i) \in \mathbb{R}^3 | i = 1, n \]

where \( n \) is the size of the set and \( x_i, y_i, z_i \) are coordinates in 3D which can be position or acceleration values. For this study, we employ gesture metrics that work directly on gestures represented as sets of points. Although recognition techniques that use features instead of raw points do exist (Corey and Hammond, 2008; Hoffman et al., 2010; Rubine, 1991), the straightforward nature of point-based representation brings the advantage of ease of implementation for prototypers. Also, previous works showed comparable recognition rates between point-based and feature-based representation approaches (Wobbrock et al., 2007). As a result, point sets have been used by the largely adopted $1 (Wobbrock et al., 2007). Protractor (Li, 2010), $N and $N-Protractor recognizers (Anthony and Wobbrock, 2010, 2012).

2.1. Gesture dimensionality

The number of points that describe the motion gesture is referred in this work as gesture dimensionality. We make distinction between the set of acquisition points (raw data) as delivered by the capture device and the set of processed points fed into the classifier and use the dimensionality concept to refer to the latter. Data points acquired from the capture device usually undergo preprocessing such as filtering for removing unnecessary points or acquisition errors and interpolation for inserting new points into the motion trajectory. For most gesture metrics, it is current practice to resample gestures uniformly into a fixed number of points before submitting them for classification (Kratz and Rohs, 2010, 2011; Li, 2010; Wobbrock et al., 2007). The goal of such a procedure is to normalize gestures in terms of the same number of points in order to allow direct point-to-point comparisons needed by some metrics (e.g., the Euclidean distance). Resampling in the arc-length domain is preferred to resampling in time in order to make recognizers invariant to articulation speed. We skip here the implementation details of such sampling procedures and refer the interested reader to actual pseudocode available in Anthony and Wobbrock (2010), Vatavu et al. (2012a) and Wobbrock et al. (2007). For the rest of the work, we assume that all gestures have been uniformly resampled into \( n \) points such that the distance between any two consecutive points \((p_i, p_{i+1})\) is constant

\[ \|p_{i+1} - p_i\| = \frac{1}{n-1} \sum_{i=1}^{n-1} \|p_{i+1} - p_i\| \] (2)

We must note here the difference between points that contain position coordinates \((x, y, z)\) and acceleration data points \((a_x, a_y, a_z)\). While the two represent different concepts, they all come down in the end to real-valued data points in \(\mathbb{R}^3\) when being stripped of their original definition and meaning. Therefore, they both represent real values describing some phenomenon that is being measured, which does not affect the way in which the Euclidean distance is computed in the \(\mathbb{R}^3\) space or in which the Nearest-Neighbor classifier is searching for the closest neighbor. For these reasons, we operate directly on acceleration points in this work and use the word point to denote acceleration data measurements in 3D.

We are interested in this work in the minimum dimensionality needed to represent gestures without affecting recognition performance, a value which we express in points per gesture (ppg). Fig. 1(a–c) illustrates the effect of dimensionality (with \(n=128, 32,\) and 8 points) on the representation of a tennis swing accelerated gesture acquired with the Nintendo Wii Remote™ game controller.

2.2. Gesture bit cardinality

The bit cardinality of a set represents the number of bits needed to represent each element from the set. For example, 1-bit cardinality sets can only represent 0 and 1; a 2-bit cardinality can represent values 0, 1, 2, and 3; while 8-bit cardinality can store the set \(\{0, 1, \ldots, 255\}\). The intrinsic cardinality of a set represents the minimum bit cardinality needed to represent the set. For example, although \(\{1, 2, 3\}\) can be represented as an array of bytes (with each value using 8 bits), its intrinsic cardinality is of only 2 bits.

In order to introduce the notion of gesture bit cardinality, we discuss an analogy with a common concept in computer graphics, the number of bits needed to represent the color of a pixel for bitmap images. The notion is referred to as color depth or bit depth and expressed in bits per pixel (bpp). Similarly, we work with the number of bits needed to represent a gesture point. Following the bitmap colors analogy, a gesture channel is a dimension on which gesture data is being recorded such as \(x, y,\) and \(z\). Therefore, we use the term gesture bit cardinality or bit depth to denote the number of bits needed to represent each gesture channel. With this definition, the number of bits per gesture (bpg) will equal the gesture dimensionality \(n\) multiplied by the number of channels \(3\) and the number of bits needed to represent each channel.

As we discuss later in the paper, today’s gesture representations are an artifact of hardware and file formats (such as 32 or 64-bit) which do not reflect the true cardinality of motion data. As a result, this work employs...
bit depth reduction strategies. If $B$ is the number of bits used to represent a gesture channel $x$, we can adjust the bit depth of the channel to $T < B$ bits by performing a simple transform

$$x = \frac{x}{2^B - 1} \cdot (2^T - 1)$$  (3)

Although more elaborate bit reduction strategies exist (Agerwala, 1976; Das and Nayak, 1990), previous works that analyzed data mining techniques for classifying time series have not found any major effect of bit reduction strategy on classification performance (Rakthanmanon et al., 2011). Obviously, one cannot obtain more resolution than is already present intrinsically in the acquired gesture for using $T > B$. However, resolution is definitely lost for $T < B$. We are interested in this work in the minimum bit depth needed to represent gestures without affecting recognition performance, a value which we express in bits per gesture (bpg). Fig. 1(d,e) illustrates the effect of reducing gesture bit cardinality for the tennis swing motion.

### 2.3. Gesture metrics

We briefly describe in this section the metrics employed in the recognition experiments reported in this work. The metrics were selected from today’s popular Nearest-Neighbor (NN) gesture recognizers that employ point-based gesture representations (Anthony and Wobbrock, 2010, 2012; Kristensson and Zhai, 2004; Kratz and Rohs, 2010, 2011; Li, 2010; Liu et al., 2009; Wobbrock et al., 2007). We decide to focus our analysis on NN recognizers only as opposed to more advanced techniques such as HMMs (Schlömmer et al., 2008) or SVMs (Wu et al., 2009) due to the multiple advantages they bring to practitioners and users of such systems (Li, 2010; Wobbrock et al.,...
To mention a few, NN recognizers expose intuitive innerworkings for practitioners, are easy to train and debug. Also, users can simply add new training samples or new gestures in order to increase the recognizer accuracy or extend the gesture set of the application, without the need to retrain or reconfigure the inner structure of the recognizer (e.g., computing new state transition probabilities for HMMs).

The Euclidean metric has been frequently used together with the NN classification approach for recognizing gestures (Anthony and Wobbrock, 2010; Kratz and Rohs, 2010; Kristensson and Zhai, 2004; Wobbrock et al., 2007). The metric computes the dissimilarity between two gesture sets of points by reporting the maximum value of all the minimum Euclidean distances computed from each point of the first set to all points in the second set (Rucklidge, 1996):

\[
\max \left\{ \sup_{p_i \in \mathbb{P}} \inf_{q_j \in \mathbb{Q}} \| p_i - q_j \|, \sup_{q_j \in \mathbb{Q}} \inf_{p_i \in \mathbb{P}} \| p_i - q_j \| \right\}
\]

The Modified Hausdorff distance reports the average instead of the maximum value (Dubuisson and Jain, 1994):

\[
\max \left\{ \frac{1}{n} \sum_{i=1}^{n} \inf_{q_j \in \mathbb{Q}} \| p_i - q_j \|, \frac{1}{n} \sum_{j=1}^{n} \inf_{p_i \in \mathbb{P}} \| p_i - q_j \| \right\}
\]

We include the Hausdorff and Modified Hausdorff metrics for our recognition tests due to their popularity for matching templates in computer vision (Rucklidge, 1996), their previous exploitation in recognizing hand-drawn sketches (Kara and Stahovich, 2004), and relatively good performance for recognizing unconstrained multi-stroke gestures (Vatavu et al., 2012a). However, many works have warned against Hausdorff being sensitive to outliers (Rucklidge, 1996; Stahovich, 2011) which may also impact the performance of these metrics when recognizing gestures under different dimensionality and cardinality representations.

2.4. Dimensionality, bit cardinality, and metric performance

Nearest-Neighbor classifiers employ a metric and a training set. The classification result is obtained by comparing the candidate gesture against all the samples in the training set using the given metric. Therefore, classification time depends on how fast the metric is to compute and how many samples are available in the training set \( T \). All the metrics that have been discussed in the previous section depend on gesture dimensionality \( n \) either linearly (Euclidean and Cosine) or quadratically (DTW, Hausdorff, and Modified Hausdorff). Therefore, the time complexity of a Nearest-Neighbor classifier employing them will be \( O(n \times T) \) or \( O(n^2 \times T) \) depending on the metric. Also, the training set requires \( O(n \times T \times B) \) memory where \( B \) is the number of bits per gesture channel. As a result, gesture dimensionality has a linear or quadratic impact on classification time while both dimensionality and bit depth determine the amount of memory required for delivering the classification result.

3. The impact of gesture dimensionality on recognition rate

We start our investigation with an experiment designed to understand how recognition accuracy is affected by the dimensionality of input gesture motion. The effect of dimensionality on recognition rate is analyzed under two scenarios: user-dependent and user-independent training. In the first scenario, gesture recognizers are trained and
tested on data acquired from the same user. For the user-independent scenario, recognizers are tested with gestures captured from different users than those providing samples for the training procedure. User-independent training is generally the preferred option for interactive systems as new users do not have to go through sometimes elaborate and time-consuming training procedures and can start using the system immediately. However, variations in gesture execution among different users make gesture classifiers deliver lower recognition rates in the user-independent case than they would for the user-dependent training scenario. Therefore, in order to provide a comprehensive understanding of how dimensionality affects recognition performance, we present and discuss results for both training scenarios.

We expect recognition rates to be higher for larger dimensionalities due to the increased resolution of motion representation that would presumably help recognizers to discriminate gestures better. We therefore state the following hypotheses guiding this study:

\( H1 \): Gesture dimensionality affects the recognition accuracy of classifiers with higher sampling rates delivering better overall recognition performance due to finer representation.

\( H2 \): Gesture dimensionality affects the performance of classifiers trained under user-dependent and user-independent scenarios alike.

\( H3 \): Some gesture motions are likely to be more affected by dimensionality than others in terms of individual recognition rates.

The gesture set of Hoffman et al. (2010) was employed for this study.\(^8\) The set contains 25 different acceleration gestures performed by 17 participants for 20 times, giving a total number of \( 25 \times 17 \times 20 = 8500 \) gesture samples. Details on the acquisition procedure, apparatus, and participants can be found in Hoffman et al. (2010). Fig. 2 shows a visual illustration of the gestures included in this set. In all following experiments, gestures have been rescaled with shape preservation and translated to origin in order to assure translation and scale invariance properties for the tested recognizers. Such uniformization represents a common preprocessing step for gesture recognition (Anthony and Wobbrock, 2010; Li, 2010; Vatavu, 2011; Vatavu et al., 2012a; Wobbrock et al., 2007). Specifically, the following procedure was carried out for each participant. \( T \) training samples were randomly selected for each gesture type to act as the training set. One extra sample (different from the first \( T \)) was additionally selected for each gesture type in order to build the testing set. Recognition rates were computed for all metrics on this testing set. The selection/test procedure was repeated for 100 times for each pair of the following factors that were controlled during the experiment:

1. **Metric**: Euclidean, Angular cosine, DTW, Hausdorff, and Modified Hausdorff;
2. **Dimensionality**: Six sampling rates were used starting from as low as 4 points up to 8, 16, 32, 64, and 128 points;
3. **Number of training samples**: \( T \) for each gesture type: \( T = 1, 2, 4, 8, \) and 16.

In order to thoroughly analyze the effect of dimensionality on recognition performance, tests started with a gesture dimensionality of \( n = 4 \) sampling points only, a value which doubled iteratively until it reached 128 sampling points per gesture. We chose to vary the number of training samples per gesture (\( T \)) as Nearest-Neighbor classifiers perform better when more samples are available in the training set. This is explained by the fact that each class is better represented in the feature space when more samples are available (Webb, 2002). Also, a previous work has shown considerable improvements in recognition rates from \( T = 5 \) to \( T = 15 \) samples per gesture type for this particular set of gestures (Hoffman et al., 2010). Overall, we report recognition results from

\[
17 \text{ participants} \times \\
25 \text{ gestures} \times \\
5 \text{ metrics} \times \\
6 \text{ sampling rates} \times \\
5 \text{ values for training samples} \times \\
100 \text{ repetitions for each } T \\
\approx 6.4 \times 10^6 \text{ recognition tests.}
\]

Fig. 3 illustrates the effect of dimensionality over recognition performance for all metrics. Accuracies start around 85% for Euclidean, Cosine, and DTW with just four sampling points and stay above 90% for 8+ points. Lower recognition rates were found for the Hausdorff metric which reached its peak performance of only 85% for the maximum tested dimensionality of 128 points which confirmed accuracy concerns from the literature review (Kara and Stahovich, 2004; Rucklidge, 1996). Modified Hausdorff performed better despite a modest start (only 77% for four sampling points) with recognition rates staying over 90% accuracy for 16+ points, which is in agreement with the good results obtained recently with...
Friedman tests (Friedman, 1937) showed a significant effect of sampling rate on recognition performance overall ($\chi^2(5, N = 10625) = 13247.621, p < 0.001$) as well as for each metric individually (at $p < 0.001$). However, post-hoc Wilcoxon signed-rank tests (Wilcoxon, 1945) revealed several nonsignificant or small-effect differences. For example, tests showed nonsignificant differences for the recognition rates delivered by DTW starting with 32 points ($\chi^2(2, N = 2125) = 0.696, n.s.$) while only small Cohen effects ($r < 0.1$) were detected between ($n=8, n=16$) and ($n=16, n=32$) for Euclidean, Cosine, and DTW. It is interesting to note that the recognition accuracy attained by Euclidean and Cosine using just eight sampling points (91.3%) is very close to the highest rate obtained for these metrics (91.7%). For DTW, rates were also close together (at less than 0.3% difference) starting with 16 sampling points with the highest rate being 96% for $n=32$, 64, and 128. The same was found for Modified Hausdorff for which rates were at less than 0.2% difference starting with 16 points. The Hausdorff metric showed the worse performance struggling to achieve 86% for the maximum sampling rate of 128 points.

Fig. 3. Recognition rates vs. gesture dimensionality for user-dependent training. Rates are averaged across all $T$ values. Error bars represent 95% CI (values too small to be noticeable at the scale of the vertical axis). Note how differences in accuracy become very small for $n \geq 8$ points for Euclidean, Cosine, and DTW and $n \geq 16$ for Hausdorff and Modified Hausdorff.

Note that Fig. 3 shows the average performance of the recognizers working with different sizes of training sets ranging from $T=1–16$ samples per gesture type. We therefore continue the analysis by illustrating the effect of the number of training samples on recognition rate (Fig. 4). As expected, the number of training samples affected significantly the recognition rate overall ($\chi^2(4, N = 12750) = 35710.585, p < 0.001$) as well as for each metric individually (at $p < 0.001$). Recognition rates for Euclidean and Cosine started with 81% with one sample and rose up to 95.0% with eight samples and 96.5% with 16 training samples per gesture type. DTW delivered better performance with 88.5% with

---

Cohen suggests the following guidelines for the effect size $r$ of a statistical test: 0.1 for small, 0.3 for medium, and 0.5 for large effect sizes (Cohen, 1992).
just one training sample, reaching 95.2% with four samples, and attaining its peak of 97.7% accuracy with 16 loaded samples per gesture type. The accuracy of the original Hausdorff metric was poor and outperformed by its Modified version which reached 95.0% with 16 samples. Despite their simplicity, all metrics (except Hausdorff) managed to attain very high recognition rates for this difficult set of 25 gestures. Their average performance stayed over 92% with only $T=4$ samples per gesture type and rose up to 97% accuracy with $T=16$ samples. It is interesting to note that the results obtained with these simple metrics (and low dimensionalities) are comparable with those delivered by more complex machine learning techniques. For example, Hoffman et al. (2010) (p. 64) report between 90 and 95% recognition rates for this set using AdaBoost and $T$ from 5 to 15 samples. Also, a linear discriminator using a large set of 41 features outperformed these metrics with just 2% delivering between 94 and 99% accuracy with up to 15 training samples per gesture type (Hoffman et al., 2010) (p. 64).

3.2. User-independent recognition results

We continue our analysis with the user-independent training scenario for which lower recognition rates are expected. This expectation is explained by the extra amount of variability in gesture input (called between-user variation) that adds to the within-user execution variation and makes the recognizer’s task more difficult.

The following testing procedure was used for computing recognition rates for the user-independent scenario. $P$ participants were randomly selected for delivering training data. For each of these participants, $T$ samples were randomly selected for each gesture type in order to build the training set (with $P \times T$ samples available for each gesture class). One additional participant (different from the first $P$) was randomly selected for testing from which one sample for each gesture type was selected for the testing set. This selection/testing procedure was repeated for 100 times for each selection of $P$ participants and 100 times for each $T$, resulting in a number of $100 \times 100 = 10,000$ repetitions. The following factors were controlled during the experiment:

1. **Metric**: Euclidean, Cosine, DTW, Hausdorff, and Modified Hausdorff;
2. **Dimensionality**: Six sampling rates starting from as low as four points up to 8, 16, 32, 64, and 128 points;
3. **Number of training participants**: $P=1, 2, 3, 4, 5, 10, 15$;
4. **Number of training samples** per gesture type: $T=1, 2, 4, 8, 16$.

We varied the number of participants used for training ($P$) in order to investigate how recognizers perform when more input variation is available to them in the training set. The values for the rest of the factors (sampling rate $n$ and number of training samples $T$ per gesture type) were the same as in the user-dependent experiment. We report recognition results from

- 7 values for training participants $P \times 100$ repetitions for each $P$
- 25 gestures $\times$
- 5 metrics $\times$
- 6 sampling rates $n \times$
- 5 values for training samples $T \times$
- 100 repetitions for each $T$

$\approx 2.6 \times 10^6$ recognition tests.

Fig. 5 illustrates the effect of dimensionality over recognition performance for all metrics. As expected, rates are lower than in the user-dependent scenario as recognizers had to model more variation in gesture execution. This time, Euclidean, Cosine, and DTW started with a low 46% accuracy with $n=4$ sampling points and reached 60% and 65% (DTW) for $n=128$. Hausdorff metrics exhibited the same low performance with the highest rates of just 36% and 39% obtained for $n=128$ sampling points. Although recognition rates are not good enough for practical applications, we must note that they represent averaged recognition performance for all $T$ and $P$ values and, as noted before, this particular gesture set is hard to recognize. We will later detail how each $P$ and $T$ influence performance and will report much higher recognition rates. For the moment, we are solely interested in how variations in sampling rate affect recognition accuracy which is the main purpose of this experiment.

Friedman tests showed a significant effect of dimensionality over recognition performance overall ($\chi^2_{(5, N = 4575)} = 5043.480, p < 0.001$) as well as for each metric individually (at $p < 0.001$). However, nonsignificant differences were found for Euclidean, DTW, and Modified Hausdorff for $n \geq 32$ while post-hoc Wilcoxon signed-rank tests showed
nonsignificant or small Cohen effects between \( n = 16 \) and \( n = 32 \) for all metrics. Also, small Cohen effects \( (r < 0.2) \) were found between \( n = 8 \) and \( n = 16 \) for Cosine, Euclidean, and Modified Hausdorff. It is interesting to note that the differences in recognition rate were less than 1.5% for Euclidean and Cosine for \( n = 8 \) and stayed under 0.8% for \( n = 16 \). For DTW, recognition rates varied by only 0.8% for \( n = 16 \). Similar small differences in recognition rates \( (< 1\%) \) were found for Hausdorff and Modified Hausdorff for \( n = 32 \).

Recognition performance reported in Fig. 5 represents averaged values for all \( T \) (number of training samples) and \( P \) (number of training participants). However, we would expect higher recognition rates for large \( T \) and \( P \) values. Fig. 6 reports the effect of \( T \) on recognition performance. As \( T \) increased from 1 to 16 training samples per gesture type, recognition improved by at least 7% for all metrics. Euclidean and Cosine reached 62% while DTW delivered 65% for \( T = 16 \). An improvement in performance was also seen for the Hausdorff metrics, although they remained under 40% accuracy. Friedman tests confirmed a significant effect of \( T \) over recognition rate for all metrics (at \( p < 0.001 \)).

We are also interested in the effect that the number of training participants \( P \) has over recognition rate. In the measure that more between-user variation is available to recognizers in the training set, their discrimination performance should improve. Fig. 7 illustrates recognition rates vs. number of training participants (values are averaged for \( n \geq 16 \) and \( T \geq 8 \)). The highest rates were 72% for Euclidean and Cosine and 76% for DTW which were obtained for \( P = 15 \) participants and at least \( T = 8 \) training samples. Hausdorff metrics still could not manage to model the between-user variation properly even with data from 15 participants available in the training set (recognition rates were under 51%). Unfortunately, these results cannot be compared this time with those obtained by the linear classifier and AdaBoost techniques employed by Hoffman et al. (2010) that only report results for smaller subsets of 9, 10, and 13 gestures out of all 25 (therefore using 50% of the available data). However, we can note DTW being the most accurate metric delivering 76% accuracy on this difficult set of gestures. As we are not mainly interested in recognition rate but rather in the effect of sampling over recognition performance, we stop this analysis here (although we could further ask what is the subset of gestures out of the 25 for which practical rates of 98–99% could be delivered as in Hoffman et al. (2010) but it is not our main purpose).

### 3.3. Dimensionality and individual gesture classification performance

Next to average recognition performance reported for the entire gesture set, we are also interested in the effect of dimensionality on individual rates for each gesture type in order to understand whether some motions are more exposed to down-sampling than others.

Fig. 8 shows individual recognition rates for each gesture and each dimensionality \( n \) for both user-dependent and user-independent training scenarios. These values represent breakdowns of the averaged results reported in Figs. 3 and 5. In order not to occupy too much of the paper estate, we show averaged recognition rates for all metrics, all \( P \) and all \( T \) values. Fig. 8 shows a clear cut-off point between \( n = 4 \) and \( n = 8 \) and all the other sampling rates (which delivered similar recognition accuracies). This was confirmed by Friedman tests for both user-dependent and user-independent training. For user-dependent, tests showed significant differences for all \( n \)
(\chi^2_{3, N = 25} = 91.737, p < 0.001) and nonsignificant differences between \( n = 16,32,64, \) and \( 128 \) (\( \chi^2_{3, N = 25} = 5.709, n.s. \)). The same was found for user-independent (\( \chi^2_{3, N = 25} = 69.233, p < 0.001 \)) and a marginally significant difference was found between \( n = 16,32,64, \) and \( 128 \) (\( \chi^2_{3, N = 25} = 11.000, p = 0.012 \)).

In order to better understand the effect of dimensionality over each gesture individually, we computed the maximum difference between recognition rates obtained for \( n \in \{16, 32, 64, 128\} \). The average difference was 2.6% (\( \text{sd} = 1.3\% \)) for user-dependent and 2.0% (\( \text{sd} = 1.9\% \)) for user-independent. The most affected gestures were Lasso and Twister in both training scenarios (with 5–6% difference). The top-3 least-affected gestures were Stop, Slash, and Forward for user-dependent (under 1.2% difference) and Spike, LineDown, and Stab for user-independent (< 0.5 difference). In total, 50% of all gestures were affected by more than 2.5% difference in recognition rate for user-dependent when varying dimensionality (and only 25% of gestures for user-independent). These results show that some gestures are more exposed to down sampling than others and therefore confirm hypothesis H3.

### 3.4. Gesture dimensionality and N-best lists

We continue our analysis by looking at the effect of dimensionality on recognizers’ N-best lists. An N-best list (Wobbrock et al., 2007) represents an ordered list of all the scores produced by a recognizer for each gesture type when classifying a candidate gesture. In accordance with the Nearest-Neighbor approach, the closest neighbor (the one located at the minimum distance from the candidate gesture) wins the classification. When using the formalism of the ordered N-best list, the nearest neighbor is found at the top of the list followed by the next closest neighbor, and so on. Fig. 9 shows the N-best lists for all metrics. In order to perform direct comparisons and analyze the performance of our recognizers, lists were normalized so that the nearest neighbor always corresponds to a distance of 1.0 (denoting best match) and the last entry in the list to 0.0 (meaning the recognizer’s poorest suggestion). A clear cut-off point between the first and the second score in the N-best list generally shows a strong confidence of the Nearest-Neighbor recognizer in its first suggestion. Also, a sufficiently large distance between the first and second score would allow a practitioner to easily choose a rejection threshold for rejection-based classifiers.

The results complement our previous findings for the performance of each recognizer. For the user-dependent scenario, the slowest downfall along the N-best list belongs to Hausdorff and Modified Hausdorff with a normalized score for the second class of 0.90. The Euclidean distance came next with a downfall to 0.84 for the second suggestion. Cosine and DTW delivered the best performance with confidence scores of 0.76 and 0.75, respectively. A Friedman test confirmed a significant effect of metric on the scores in the output N-best list (\( \chi^2_{4, N = 25} = 90.063, p < 0.001 \)). Nonsignificant differences
were shown by a post-hoc Wilcoxon signed-rank test between Hausdorff and Modified Hausdorff ($Z = -1.897$, n.s.). Similar results were found for the user-independent scenario: Hausdorff and Modified Hausdorff have the slowest downfall (0.95), followed by Euclidean (0.89), Cosine (0.87), and DTW (0.86).

The effect of dimensionality on the N-best lists is shown in Fig. 10. In order to save paper state, we only show averaged values for all metrics similar to our previous display of the individual recognition rates (Fig. 8). The smallest decrease in metric score from the first to second class was found for $n=4$ (from 1.00 to 0.87 in the user-dependent and 1.00–0.92 in the user-independent case). Starting with $n=8$ points, this difference seems to remain constant (from 1.00 to 0.82 and 1.00 to 0.90, respectively) and the N-best lists overlap. This observation was confirmed by a Friedman test that showed a significant effect of sampling rate $n$ over the scores in the user-dependent ($\chi^2_{(5,N = 25)} = 96.575, p < 0.001$) and user-independent N-best lists ($\chi^2_{(5,N = 25)} = 109.953, p < 0.001$). A nonsignificant difference was detected for $n=16$+ points for user-dependent training ($\chi^2_{(13,N = 25)} = 4.385, n.s.$).

3.5. Summary

Before proceeding further with the analysis, we provide a short summary of the main findings on the effect of dimensionality on recognition performance:

1) Dimensionality had a significant effect over recognition rate overall but small-effect and/or nonsignificant differences were also detected for particular cases. These findings partially confirm hypothesis H1 which suggested that much higher recognition accuracies would be expected for large sampling rates. Although significant, differences were very small for practical purposes. In general, recognition rates were close together starting with $n=8$ or $n=16$ sampling points depending on the metric type. We can therefore affirm that the extra information contained in large resolution representations does not contribute significantly to the classification process. Although not specifically observed for the tested sampling rates, our findings could extrapolate and suggest that large resolutions could also pick up potential noise in the data which might affect recognition accuracy negatively. Such a consequence is very likely as it connects to the “peaking phenomenon” encountered in pattern recognition (Sima and Dougherty, 2008) that describes the situation in which adding more features up from one point does not improve but actually increases the classification error.

2) An analysis of individual recognition rates for each gesture type showed a clear cut-off point in recognition
performance between \( n \in \{4, 8\} \) and all the other rates (for which the average difference in recognition rate was 2\%). Some gestures were found to be more affected by downsampling which confirmed hypothesis H3. This also suggests that the optimum sampling rate is specific for the gesture set employed by the interface. In this regard, practitioners would benefit of an assisting tool informing on the right resolution to use in their designs in order to optimize recognition rate and execution time (later in the paper we discuss such a tool).

3. The downfall of the N-best lists was slower for smaller dimensionalities but a clear cut-off point could be observed between \( n=4 \) points and all the other values. The recognizer confidence in its decision (i.e., the first neighbor) was little influenced by dimensionality for \( n \geq 8 \) points.

4. All the above observations applied for both user-dependent and user-independent scenarios which confirmed hypothesis H2.

4. The combined effect of gesture dimensionality and the size of the gesture set

The results obtained in the first experiment on a rather difficult set of gestures (Hoffman et al., 2010) are intriguing as they show that even low sampling rates are enough in order to obtain high recognition accuracies. However, in their practice, designers are facing various application requirements for which various number of gestures will be employed. Except for extreme cases (Kristensson and Zhai, 2004; Zhai et al., 2009), a small number of gesture commands are likely to be proposed for an application in order to facilitate interface learnability, memorability, and ease-of-use (Nielsen et al., 2003). In this light of usability considerations, the set of 25 gestures employed in the previous experiment should provide a sufficiently high upper margin for the number of gestures used by most gesture-based interfaces today.

We continue our investigation in order to understand the relationship between the size of the gesture set and the minimum dimensionality that can be used in order to attain peak recognition performance. We understand by the size of a gesture set the number of different gesture types that make up the set (e.g., 25 as in Fig. 2). Following intuition and as informed by the results of the first experiment, we expect small gesture sets to be discriminated just as well with few sampling points (low dimensionality) while recognizers will probably need more representation resolution (higher dimensionalities) in order to discriminate gestures in larger sets. Therefore, we propose the following hypothesis:

\[ H4: \text{A relationship exists between the minimum gesture dimensionality required to deliver peak recognition performance for a given gesture set and the size of the set.} \]

A new experiment was conducted in order to verify this hypothesis. The number of gestures in the set (denoted in the following by \( r \)) was varied from 5 to 10, 15, 20, and 25.

For each set size, \( r \) gestures were randomly selected from the available 25 of Fig. 2 (Hoffman et al., 2010). The experiment started with a low dimensionality of just \( n=4 \) sampling points. This value was incremented iteratively until no significant differences were detected between the recognition rates of tested \( n \) and those reported by a reference sampling rate.\(^\text{11}\) The Wilcoxon signed-rank test was used to validate significance at \( p < 0.05 \). This procedure was repeated for 100 times for each size of the set \( r \), each time selecting a new set of \( r \) gestures from all the possible combinations of the 25 gestures:

\[
\binom{25}{r} = \frac{25!}{(25-r)!\cdot r!}
\]

As in the first experiment, the number of training samples per gesture was varied from \( T=1 \) to 2, 4, 8, and 16 and the number of training participants \( P \) from 1 to 2, 3, 4, 5, 10, and 15.

Fig. 11 shows the minimum sampling rates that were found to deliver the same (nonsignificantly different) recognition accuracies as \( n=32 \) for the user-dependent case. Results were similar for user-independent and therefore omitted to save paper estate. Linear regression models were derived for each metric \((R^2 > 0.92)\) and are shown in Table 2. The slopes were as small as 0.09 for Euclidean and Cosine which shows a weak dependence between dimensionality and the size of the gesture set for these metrics. The average value of \( n \) was just 6.4 in both cases (sd=0.7). Therefore, eight sampling points per gesture would seem a sufficient upper limit for the Euclidean and Cosine metrics to recognize gestures from this set. A much larger variation in dimensionality was found for DTW with a mean of 10.8 sampling points (sd=2.3). This result can be explained by the elastic nature of the dynamic time warping algorithm that strives to find the optimum alignment even when it does not exist, also noted by other works (Kristensson and Zhai, 2004; Wobbrock et al., 2007). Hausdorff and

\(^{11}\)Thirty-two sampling points were selected as the reference dimensionality. Our previous testing from the first experiment showed good recognition rates for \( n=32 \), not significantly different than those delivered by higher rates. Also, other works on 3D gesture recognition have used this sampling resolution before (Kratz and Rohs, 2011).
Table 2
Linear regression models for the dimensionality \( n \) that achieves the same recognition accuracy as a reference dimensionality of 32 points on a set of \( r \) distinct gestures.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Linear regression model</th>
<th>Fit ( R^2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Euclidean</td>
<td>( n=0.09 \times r + 5.10 )</td>
<td>0.92</td>
</tr>
<tr>
<td>Cosine</td>
<td>( n=0.09 \times r + 5.00 )</td>
<td>0.92</td>
</tr>
<tr>
<td>DTW</td>
<td>( n=0.29 \times r + 6.40 )</td>
<td>0.93</td>
</tr>
<tr>
<td>Hausdorff</td>
<td>( n=0.44 \times r + 8.00 )</td>
<td>0.95</td>
</tr>
<tr>
<td>Modified-Hausdorff</td>
<td>( n=0.39 \times r + 8.30 )</td>
<td>0.94</td>
</tr>
</tbody>
</table>

Modified-Hausdorff distances had larger slopes (0.4) and larger average dimensionalities (14.2 points).

Hypothesis H4 is therefore partially confirmed. Although linear regressions were successfully derived for each metric, Euclidean and Cosine showed a very weak dependence between dimensionality and gesture set size. Few sampling points (8) were needed for these metrics to attain high recognition performance irrespective of how many gestures were in the set. The hypothesis was confirmed for DTW, Hausdorff, and Modified-Hausdorff showing that small gesture sets can be discriminated with low gesture dimensionalities while larger sets need finer representation resolutions.

5. The impact of gesture bit cardinality on recognition rate

We continue our investigation by analyzing the effect of bit cardinality on recognition performance of the proposed gesture recognizers. The experiment design was similar to the first study including both user-dependent and user-independent training. We expect recognition rates to be higher for larger bit depths as gesture motions will be represented at finer resolutions. Similar to hypotheses H1–3 for gesture dimensionality, we propose verifying the following hypotheses on the impact of bit depth on recognition rate:

\( H5 \): Gesture bit cardinality affects the recognition accuracy of classifiers with larger bit depths delivering better overall recognition performance due to finer representation.

\( H6 \): Gesture bit depth affects the performance of classifiers trained under user-dependent and user-independent scenarios alike.

\( H7 \): Some gesture motions are likely to be more affected by bit depth rather than others in terms of individual recognition rates.

5.1. User-dependent recognition results

Recognition rates were computed using the same procedure employed by the user-dependent experiment for gesture dimensionality with the following factors being controlled this time:

1. **Metric:** Euclidean, Angular cosine, DTW, Hausdorff, and Modified Hausdorff;

2. **Bit cardinality:** Seven different bit depths \( B \) starting from as low as 2 bits for each \( x, y, z \) channel and continuing with 3, 4, 5, 6, 7, and 8 bits per channel;

3. **Number of training samples:** \( T \) for each gesture type: \( T=1, 2, 4, 8, \) and 16.

The values for bit depth \( B \) started with as low as 2 bits and reached a maximum of 8 bits per channel. As the gesture set used for the experiment (Hoffman et al., 2010) is composed of gestures captured with the Wii Remote controller that delivers 8 bits of precision through its embedded 3-axis accelerometer,\(^\text{12}\) there was no point testing beyond 8 bits per channel. Recognition rates obtained for \( B=8 \) bits were used as reference when assessing the performance delivered by smaller bit depths. A fixed sampling rate of \( n=32 \) points was used for the experiment as informed by the previous results on gesture dimensionality (for which \( n=32 \) achieved recognition rates comparable with those delivered by much higher dimensionalities yet better when compared to smaller sampling rates, see Figs. 3 and 5).

Fig. 12 illustrates the effect of bit cardinality on recognition performance for all metrics. We are interested in how recognition rates obtained for low bit depths compare to the performance delivered by 8 bits. Interestingly, Euclidean and Cosine delivered 89.2% accuracy for 2 bits only which was just 2% smaller than the maximum accuracy attained for 8 bits (91.5%). Also, starting with \( B=3 \) bits, both Euclidean and Cosine delivered recognition rates over 91.3%. Wilcoxon signed-rank tests confirmed a nonsignificant difference between \( B=3 \) and \( B=8 \) for both Euclidean (\( Z=-1.130, n.s. \)) and Cosine (\( Z=-1.567, n.s. \)), DTW started with 92.0% accuracy with 2 bits and attained maximum recognition of 95.5% for \( B=8 \). A Wilcoxon test showed a nonsignificant difference between \( B=4 \) and \( B=8 \).

for DTW ($Z = -0.976, n.s.$). The Hausdorff metrics had a worse start but they also managed to deliver recognition accuracies of 84.4% and 90.8% for $B=5$ bits per channel that were not significantly different from those obtained with 8 bits ($Z = -1.170$ for Hausdorff and $Z = -1.587$ for Modified Hausdorff, n.s.).

5.2. User-independent recognition results

The following factors were controlled for the user-independent experiment:

(1) Metric: Euclidean, Angular cosine, DTW, Hausdorff, and Modified Hausdorff;
(2) Bit cardinality: Seven different bit depths $B$ starting from as low as 2 bits for each $x$, $y$, $z$ channel and continuing with 3, 4, 5, 6, 7, and 8 bits per channel;
(3) Number of training participants $P$: 1, 2, 3, 4, 5, 10, and 15;
(4) Number of training samples $T$ for each gesture type: $T=1$, 2, 4, 8, and 16.

Fig. 13 shows the recognition rates obtained for all metrics while varying bit cardinality. Again, we are interested in how lower bit depths manage to compare with the maximum of $B=8$ bits per channel. Results were similar with those obtained for the user-dependent training scenario. Nonsignificant differences were found between the recognition rates of $B=3$ and $B=8$ for Euclidean ($Z = -2.531, n.s.$) and Cosine ($Z = -2.459, n.s.$); between $B=4$ and $B=8$ for DTW ($Z = -1.313, n.s.$); and between $B=5$ and $B=8$ for Hausdorff ($Z = -0.799, n.s.$) and Modified Hausdorff ($Z = -0.141, n.s.$).

5.3. Bit cardinality and individual gesture classification performance

We continue the analysis by looking at individual gesture recognition rates in order to understand whether some motions are more affected by changes in bit cardinality than others. For this, we computed individual recognition rates for each gesture type and each bit depth, and calculated the average difference in recognition rate between $B=8$ and $B=3$ bits per channel for all recognizers (as $B=3$ was found to be the smallest bit depth to deliver nonsignificantly different accuracies when compared to 8 bits). The average difference was 3.3% ($sd=2.5\%$) for user-dependent and 2.1% ($sd=1.8\%$) for user-independent training. For user-dependent, the top-3 gestures that were least influenced by bit depth were $Forward$ (0.03%), $Left$ (0.26%), and $Circle$ (0.28%) while the top-3 most influenced motions were $LineUp$ (6.60%), $Stop$ (7.90%), and $Zorro$ (9.10%). For user-independent, $Left$ (0.09%), $OpenDoor$ (0.33%), and $LineDown$ (0.35%) were least influenced while $Square$ (4.30%), $Stop$ (6.40%), and $Zorro$ (6.60%) were the most affected by variations in bit cardinality.

5.4. Summary

As for the gesture dimensionality experiment, we summarize our findings on the effects of bit cardinality and verify the validity of the formulated hypotheses:

(1) Bit cardinality had a significant effect on recognition rate overall but many important nonsignificant differences were also found. Specifically, 3 bits per channel were found to deliver the same (nonsignificantly different) recognition rate as a maximum of 8 bits for Euclidean and Cosine. The same was found for 4 bits for DTW and 5 bits for Hausdorff and Modified Hausdorff. While hypothesis H5 was partially confirmed, we found that low bit depths can deliver the same high levels of recognition performance similar to those provided by the maximum available precision in representing data.

(2) The same effects of bit cardinality were found for both user-dependent and user-independent training scenarios which confirms hypothesis H6. Even more, the same bit depth values that delivered the same level of recognition performance as $B=8$ were found in both scenarios (3, 4, and 5 bits per channel depending on metric type).

(3) Some gestures were more affected by bit cardinality than others with differences in recognition rates between $B=3$ and $B=8$ bits per channel varying from 0.03% to 9.10% which confirmed hypothesis H7.

6. The impact on execution time and system memory

Besides recognition rate, two more factors define the performance of a gesture recognizer: response time and the amount of memory required to deliver the classification result. We discuss in the following the impact of gesture dimensionality and bit cardinality on these two important factors.
6.1. The effect of dimensionality on execution time

Gesture dimensionality and the size of the training set are the main factors that influence execution time. Standard implementations of Nearest-Neighbor recognizers compare the candidate gesture with every sample in the training set in a sequential manner, which makes the size of the set linearly related to execution time. For user-dependent training, the size of the set is given by the number of distinct gestures (G) multiplied by the number of training samples per gesture type (T). For user-independent training, data from multiple participants needs to be used so the size of the training set multiplies accordingly by P.

We must note that better search techniques can be devised to lower the execution time of the sequential search procedure by employing specialized data structures that partition the feature space, with K-d trees being a popular choice (Friedman et al., 1977) (see also Zezula et al., 2006 for a review). However, such approaches, although promising in terms of delivering low theoretical complexity and low execution times, may prove difficult to implement by a practitioner prototyping a gesture recognizer. Even for experienced practitioners and developers, such data structures may present challenges in implementation which detract from rapid prototyping and testing.

To the best of the authors’ knowledge, such advanced techniques have not found their way in the HCI community simply because of the above reasons, despite the fact that such speed improvements are needed in many cases. It is for similar reasons that the S-family of gesture recognizers (S1, SN, SP) has been introduced to the community (Anthony and Wobbrock, 2010; Vatavu et al., 2012; Wobbrock et al., 2007). Therefore, we decide to consider in the following the effect of dimensionality on the standard, accessible, and popular linear search implementation of the Nearest-Neighbor approach, while we note the existence of faster yet more complex techniques (Zezula et al., 2006).

The dependence on gesture dimensionality is linear for the Euclidean and Cosine metrics and quadratic for DTW, Hausdorff, and Modified-Hausdorff as discussed under the introductory sections of the article. We can therefore summarize the time complexities of classifying a candidate gesture with the Euclidean and Cosine as $O(n \times G \times T)$ for user-dependent and $O(n \times G \times T \times P)$ for user-independent while DTW, Hausdorff, and Modified-Hausdorff execute in $O(n^2 \times G \times T)$ and $O(n^2 \times G \times T \times P)$ time, respectively. We need again to note that various DTW optimizations exist that improve execution time by means of exploiting geometric constraints for the wrapping path (Sakoe and Chiba, 1978; Itakura, 1975) or lower bounds (Rakthanmanon et al., 2012). Keogh (2002) provides a review of such DTW optimization techniques. However, we prefer to use the original, unmodified version of DTW in order not to consider the effect of extra parameters for the analysis (such as the width of the Sakoe-Chiba band or the Itakura parallelogram that are connected to error rate Rakthanmanon and Keogh (2004)) or the complexity of extra implementation such as reversing query/data role, reordering early abandoning, or employing cascades of lower bounds (Rakthanmanon et al., 2012) that may prove too complex to implement for devices with limited programming environments (see the introductory section for such examples). In doing so, we consider DTW as any other motion gesture distance that executes in quadratic time. With these considerations, we are able in the following to assess and discuss the effect of dimensionality on metrics that execute in linear and quadratic time at a more general level, without being influenced by specific optimization techniques that are particular for one metric or the other.

We measured execution times on a mobile phone with a 520 MHz processor (ARM920T PXA27x), 116 MB onboard memory, and running Windows Mobile® 6 Professional. The choice of processor was a compromise between fast versions available on the market (e.g., the 1 GHz of iPhone/iPad) and the extreme low processing power delivered by miniaturized devices (e.g, the 20 MHz CPU Speed of the eZ430-Chronos wrist watch, as discussed in the introductory section). Execution times were measured for a set composed of $G=10$ gestures, a value selected to approximate the requirements of an average gesture-based interface: a number high enough to support many functionalities yet limited to help memorability and recall. For the user-dependent tests, a number of $T=4$ samples per gesture type was used as informed by the recognition rates displayed in Fig. 3. For user-independent testing, a number of $P=5$ training participants was used as informed by results obtained in the user-independent experiment and reported in Fig. 7. A compromise was sought between the size of the training set as influenced by $T$ and $P$ and recognition accuracy for both scenarios. Therefore, the size of the training set was 10 (gestures) $\times$ 4 (samples)=40 for user-dependent and 10 (gestures) $\times$ 4 (samples) $\times$ 5 (participants)=200 samples for user-independent testing. Gesture dimensionality varied from $n=8$ up to 128 sampling points ($n=8$ was the smallest dimensionality that delivered good recognition rates when compared to higher $n$ values, see Figs. 3 and 5).

Table 3 shows the execution times (in milliseconds) needed for each recognizer to classify a candidate gesture. Both the Euclidean and Cosine metrics delivered the result in reasonable times even for large gesture dimensionalities. Response time were 9 ms for the Euclidean metric and 20 ms for Cosine when classifying a gesture in the user-dependent case with $n=16$ points. Maximum response times were 360 ms and 760 ms, respectively measured in the user-independent scenario and using the maximum dimensionality of $n=128$ sampling points. The execution times of DTW, Hausdorff, and Modified Hausdorff were considerably affected by gesture dimensionality. Starting with $n=32$ sampling points, response times became unacceptable for real-time interfaces: 2–3 s for $n=64$ in the
user-dependent case which went up to 60 s for DTW and $n=128$ in the user-independent scenario.

These measurements confirm the importance of gesture dimensionality analysis for devices with low processing resources. When correlating these values with the results of the recognition performance analysis (Figs. 3 and 5), we see that by simply representing gestures using $n=16$ instead of $n=64$ points, classification times improve by 4x for Euclidean and Cosine and by 16x for DTW and Hausdorff while maintaining the same high recognition accuracies. Also, when compared to the $S3$ recognizer of Kratz and Rohs (2010) that samples gestures in $n=150$ points, the improvement in classification speed becomes 9x for the Euclidean distance and 87x for DTW.

### 6.2. The effect of dimensionality and bit cardinality on system memory

In order to deliver the classification result, Nearest-Neighbor recognizers need to compare the candidate gesture with all the samples stored in the training set. Therefore, the memory requirements for such recognizers are given by the size of the training set. The size of the set depends on gesture dimensionality $n$ (how many data points are used to represent each motion), bit depth $B$ (how many bits per gesture channel), number of training samples per gesture type $T$, and number of training participants $P$. Therefore, the amount of required memory is $3 \times n \times B \times T \times P$ bits for user-dependent and $3 \times n \times B \times T \times P$ bits for user-independent training.

Table 4 illustrates the memory needed to store the training set for various bit depths and gesture dimensionalities. Bit depths of 16 and 32 bits are also included in the table as they represent today’s standards for microcontroller and CPU architectures. This means that a practitioner would naturally allocate a full 32-bit word size for storing a given value simply because the hardware architecture supports it natively. However, we showed before that such a high bit depth is not representative of the intrinsic bit cardinality of gesture data for which 3, 4, and 5 bits per channel are sufficient for our metrics. Therefore, actual memory saving in practice can be from 150 KB (needed for storing $n=64$ points on 32 bits) to 4.7 KB ($n=16$, 4 bits) in the user-independent case, which means 32x less memory needed for the design.

### 7. Modeling the impact of gesture dimensionality and bit cardinality on recognition accuracy

The results of the gesture dimensionality and bit cardinality experiments are extremely intriguing as they show...
that high recognition performance can be obtained even for low sampling rates and small bit depths. In turn, such small data dimensionality means faster execution times and lower memory requirements for recognizers. However, while experiments showed that only few sampling points represented on few bits are sufficient to deliver high recognition performance, a deeper look into what causes this phenomenon is mandatory. We therefore complement these empirical results with a probabilistic model explaining the impact of these factors on recognition rate.

The major risk that can occur when either dimensionality or bit cardinality get reduced is for motion gestures belonging to different classes to be perceived as being similar by the recognizer just because of losing representation resolution. The risk can be expressed in terms of the probability that a recognizer would not discriminate two gestures when represented with low data dimensionality and bit cardinality. We start our argumentation by introducing the concept of equivalence for points and sets of points on which we base our calculation of such a probability.

7.1. Probabilities of equivalence for points and gestures

For all further discussion, we assume that gestures have been normalized in the \( D = [0,1]^3 \subset \mathbb{R}^3 \) unit cube. Such uniformization is a common preprocessing step in order to assure scale-invariant gesture recognition and has been used for all recognition experiments reported in this article. The introductory sections refer to such preprocessing steps and point to actual pseudocode for implementing them.

**Definition.** Points \( p, q \in D \) are equivalent with respect to \( \epsilon \in \mathbb{R}^+ \) (and we denote \( p \equiv q \)) if the Euclidean distance between \( p \) and \( q \) is less than \( \epsilon \)

\[
p \equiv q \iff \|p - q\| \leq \epsilon
\] (12)

Let \( \alpha \) and \( \beta \) denote two motion gestures and let \( \alpha[n]\) and \( \beta[n] \) be their resampled versions into \( n \) points (which we refer to as \( n \)-samplings). The concept of equivalence can be extended to motion gestures by applying the \( \epsilon \) definition for each pair of corresponding points, as follows:

**Definition.** The \( n \)-samplings \( \alpha[n]\) and \( \beta[n] \) of motion gestures \( \alpha \) and \( \beta \) are said to be equivalent (\( \alpha[n] \equiv \beta[n] \)) if all their corresponding points are equivalent:

\[
\alpha[n] \equiv \beta[n] \iff \alpha_i \equiv \beta_i \quad \forall i \in \{0,n-1\}
\]

With these definitions, we can calculate the probability that two gestures \( \alpha \) and \( \beta \) that are not equivalent with respect to their \( n \)-samplings (\( \alpha[n] \neq \beta[n] \)) become equivalent when sampled at a lower \( m < n \) rate (\( \alpha[m] \equiv \beta[m] \)). For this, we denote by \( e(n) \) the number of pairs of equivalent points from \( \alpha[n] \) and \( \beta[n] \). Under the above assumption (i.e., \( \alpha[m] \equiv \beta[m] \)), we must necessarily have \( m \leq e(n) \). We also denote by \( \lambda(n) \) the proportion of equivalent pairs, \( \lambda(n) = e(n)/n. \) Obviously, \( \lambda(n) \) will fall into \([0,1]\). The probability that \( \alpha[m] \equiv \beta[m] \) for \( m < n \) is given by

\[
e(n) - e(n) - 1 \cdot e(n) - (m-1) n \cdot n - (m-1) \leq (\lambda(n))^m
\] (13)
in which \( e(n)/n \) is the probability that the first pair of points are equivalent, \( (e(n) - 1)/(n-1) \) the probability of the second pair, and so on for each of the \( m \) pairs. In order for the probability computed in Eq. (13) to hold, the \( m \) points must be selected out of the \( n \) points, which makes the \( m \)-sampling a subset of the \( n \)-sampling. Therefore, we must restrict \( m \) to divisors of \( n \), i.e., \( m|n \). This constraint is in accordance with our experiment methodology in which we used sampling rates in a geometric progression with a ratio of 2 (i.e., 4, 8, 16, 32, 64, and 128 points).

The probability of equivalence defined for two gestures can be extended for a set of \( r > 2 \) gestures by calculating the probability that at least two gestures out of \( r \) are equivalent when being sampled into \( m \) points. For this, we employ similar definitions and notations. Let \( \lambda_{ij}(n) \) denotes the proportion of equivalent pairs for gestures \( i \) and \( j \) with \( i,j = 1,r \) and \( i \neq j \). An upper bound for the probability that the \( m \)-samplings of gestures \( i \) and \( j \) are equivalent was calculated before as \( \leq (\lambda_{ij}(n))^m \). Therefore, the probability for at least one pair of gestures out of \( r \) to be equivalent will have the following upper bound:

\[
\leq \sum_{r=1}^n \sum_{i=1}^r (\lambda_{ij}(n))^m
\] (14)

7.2. Upper bounds

We are interested in the following in estimating the values of \( \lambda(n) \) (and \( \lambda_{ij}(n) \), respectively) in order to calculate upper bounds for Eqs. (13) and (14). The upper bounds will inform on the maximum values of the probability that two gestures become equivalent due to loss of representation while being downsampled.

**Theorem.** The probability that two points \( p, q \in D \) are equivalent can be approximated as follows:

\[
P(p \equiv q) \approx \frac{4\pi}{3} \epsilon^3
\] (15)

Intuitively, this result can be explained by the ratio of a sphere of radium \( \epsilon \) centered on one of the points and the volume of the unit cube. Nevertheless, we provide a full proof below.

**Proof.** The probability function of the Euclidean distance between two points picked at random in the unit cube is

\[
P(t) = \frac{-t^2 \cdot ((t-8) \cdot t^2 + \pi \cdot (6t-4))}{6}
\] (16)

where \( t \leq 1 \) is the distance between points (Weisstein, 2012). Therefore, the probability that the distance between \( p \) and \( q \) is at most \( \epsilon \) will be:

\[
P(t \leq \epsilon) = \int_0^\epsilon P(t) \, dt \approx \frac{4\pi}{3} \epsilon^3 - \frac{3\pi}{2} \epsilon^4 + \frac{8}{5} \epsilon^5 - \frac{1}{6} \epsilon^6
\]
which can be approximated by \((4\pi/3)^3\) for small values of \(\epsilon \ll 1\).

If the \(n\) pairs of points are chosen independently from \(x\) and \(\beta\) then the random variable \(e(n)\) follows the binomial distribution with the probability of success \(p = (4\pi/3)^3\). In this case, the expected value for \(e(n)\) will be \(E[e(n)] = np\) (Ghahramani, 2000 p. 181) which makes the expected value for \(\lambda(n)\) to be \(p\). However, we are interested in the expected value of \((\lambda(n))^m\) that appears in Eqs. (13) and (14):

\[
E[(\lambda(n))^m] = \frac{1}{m!} \sum_{k=0}^{n} \binom{n}{k} p^k (1-p)^{n-k}
\]

(17)

**Theorem.** The sum \(S(n, m)\) of Eq. (17) is a polynomial in \(n\) of order \(m\) with the coefficient of \(n^m\) being \(p^m\).

**Proof.** By using \((\lambda)^m = \frac{\lambda^m}{m!}\), putting \(k-1 = i\), and expanding \((1+i)^{m-1}\) in (17), it can be showed that:

\[
S(n, m) = np \sum_{i=0}^{m-1} \binom{m-1}{i} S(n-1, i)
\]

with \(S(n, 0) = S(n-1, 0) = 1\).

Next, the principle of induction is used. For \(m=1\), \(S(n, 1) = np\) and for \(m=2\), \(S(n, 2) = n^2p^2 + np - p^2\). We assume that \(S(n, m) = n^{m-1}p^m + O(n^{m-1})\) where by \(O(n^{m-1})\) we denote a polynomial in \(n\) of order \(m-1\). Then:

\[
S(n, m+1) = np \sum_{i=0}^{m} \binom{m}{i} \cdot (p^r(n-1)^1 + O(n^{-1}))
\]

in which \(n^m\) appears once for \(i=m\) with the coefficient \((m^m p^m)\). Therefore, \(S(n, m+1) = n^m + 1p^m + O(n^{m-1})\). □

Using these findings, we can write:

\[
E \left[ \left( \frac{e(n)}{n} \right)^m \right] = \frac{S(n, m)}{n^m} = \frac{p^m n^m + O(n^{m-1})}{n^m}
\]

If we consider \(n\) to be large enough\(^{13}\) then we can approximate the expected value of \((\lambda(n))^m\) to \(p^m\). We can further use Markov’s inequality (Ghahramani, 2000, p. 437) in order to obtain an upper bound value \(\lambda\) for \((\lambda(n))^m\) for which the probability \(P((\lambda(n))^m \geq \lambda)\) can be neglected (e.g., the probability is less than 0.01):

\[
P((\lambda(n))^m \geq \lambda) \leq \frac{E[(\lambda(n))^m]}{\lambda} \approx \frac{p^m}{\lambda} = 0.01
\]

Therefore, by choosing \(\lambda = 100 \cdot p^m\), we are 99% confident that \((\lambda(n))^m \leq \lambda\). By using this in Eq. (14), we obtain an upper bound for the probability of equivalence of the \(m\)-samplings for sets of \(r\) gestures:

\[
P(m, r) \leq 50 \cdot r(r-1) \cdot \left(\frac{4\pi}{3} \epsilon^3\right)^m
\]

(18)

The only factor left unknown so far is the constant \(\epsilon\) which defines the equivalence of two points (see Eq. (12)). However, \(\epsilon\) is directly related to bit depth. Small \(\epsilon\) values represent fine bit depths while large \(\epsilon\) values illustrate the effect of low bits per gesture channel. As \(\epsilon\) falls in \([0,1]\) and represents the radius of the sphere around a sampling point on the gesture path, we can derive the following upper bound:

\[
\epsilon \leq \frac{1}{2B}
\]

(19)

where \(B\) is the bit depth per channel used to represent the data. The constant 2 comes from the fact that \(\epsilon\) is defined as radium and therefore \(2\epsilon\) represents the diameter of the equivalence sphere. This observation allows updating Eq. (18) to

\[
P(m, r) \leq 50 \cdot R(R-1) \cdot \left(\frac{\pi}{6B}\right)^m
\]

(20)

In practice, template-based recognizers work with several samples \(T\) stored for each gesture class. Therefore, we need to consider \(T\) for each of the \(r\) gesture classes which makes the upper bound become:

\[
P(m, r) \leq 50 \cdot R(R-1) \cdot \left(\frac{\pi}{6B}\right)^m
\]

(21)

where \(R\) is the number of gestures used in the training set: \(R=r \times T\) for user-dependent training and \(R=r \times T\times P\) for user-independent.

In order to understand the practical implications of these upper bounds, we calculated them for \(r=25\) distinct gestures (Hoffman et al., 2010), \(T=16\) training samples per gesture type, and \(P=15\) training participants (values which correspond to a maximum size of the training set, \(R=25 \times 16 \times 15 = 6000\) samples). The calculated probability for \(B=2\) bits per channel and \(m=8\) sampling points is 0.61 (61% chance that downsampling will make the recognizer confound gestures). However, when \(B \geq 3\), the probability stays below \(3.6 \times 10^{-5}\) for \(m=8\) and below \(2.9 \times 10^{-46}\) for \(m=32\) points. Such theoretical calculations support the empirical findings of the experiments.

8. Gesture analysis tool

Experimental results showed that few sampling points and bit depths can attain the same high level of recognition accuracy as much finer resolutions. For some metrics, empirical results also showed that dimensionality is related to the number of distinct gestures included in the gesture set. Specifically, we found an upper bound of eight points to be more than enough for the Euclidean and Cosine metrics, while a linear dependence relating sampling rate and gesture set size was derived for DTW and Hausdorff distances (suggesting 15 and 20 points respectively for sets up to 25 gestures). However, variations from these values are likely to occur in practice when practitioners have to meet design requirements for specific applications and specific gesture sets. Also, analyses on individual classification rates showed that some gestures are more exposed to

\(^{13}\)We can safely make this assumption as \(n\) can be viewed as a very fine sampling of a continuous motion gesture.
downsampling than others. Therefore, we estimate that practitioners would benefit from a gesture analysis tool to assist and inform their hardware and software designs.

With these considerations in mind and following the tradition of other researchers providing gesture design tools (Ashbrook and Starner, 2010; Bragdon et al., 2009; Kohlsdorf et al., 2011; Long et al., 1999; Lyons et al., 2007), we present the Gesture Dimensionality Analysis Toolkit (GDATK). GDATK is meant to help practitioners choose the right values for dimensionality and bit depth when tuning gesture recognizers in their designs (see Fig. 14 for a snapshot). GDATK allows practitioners to test the performance of their gesture metric (Euclidean, Cosine, DTW, Hausdorff, or Modified Hausdorff) by reporting recognition rates and execution times for a particular gesture set under different sampling rates and bit depths. Our tool is similar to the Minimum Description Length technique of Hu et al. (2011) that is used to conduct an exhaustive search over all possible dimensionality and cardinality values to find the best model, dimensionality, and cardinality for representing a time series using a minimum number of bits. Hu et al. (2011) also consider approximation models for time series in their technique (e.g., Discrete Fourier Transform, Adaptive Piecewise Linear Approximation, and Piecewise Linear Approximation) while GDATK relies on the original data alone in order to reduce the effort required for such additional implementations for practitioners.

The tool is freely available for download from the corresponding author’s web page. In the following, we show the usefulness of this toolkit by repeating the recognition experiments for other gesture sets available in the literature: the uWave (Liu et al., 2009) and the 6DMG set (Chen et al., 2012).

8.1. Recognition performance on the uWave set

The set contains eight accelerated motion gestures (see Fig. 15) performed by eight participants in seven different days with 10 repetitions per day. Therefore, the number of available samples is 4480. The set can be downloaded from. The mean number of points for the gestures in the set is 99 (sd = 47.7) and was used as benchmark for the recognition rate tests. As Euclidean, Cosine, and DTW showed the highest recognition performance in our first experiment, we only report results for these recognizers and the user-dependent training scenario.

The recognition rate was 97.5% for the Euclidean distance with \( n = 99 \) sampling points. When using \( n = 6 \) points only (as informed by the results of Fig. 11), an accuracy of 96.8% was obtained (less than 1% difference). A Wilcoxon signed-rank test showed a significant \( (Z = -8.576, p < 0.001) \) yet small effect size \( (r = 0.14) \) between the two. However, when using the toolkit to test further, a number of \( n = 10 \) points were found to deliver 97.5% recognition accuracy, nonsignificantly different from that delivered by 99 points, as confirmed by Wilcoxon tests. Results were similar for the Cosine distance. Recognition rates were 97.3% for \( n = 99 \) and 96.6% for \( n = 9 \) points \( (Z = -7.109, p < 0.001, r = 0.12) \). However, recognition rates did not differ significantly for \( n = 8 \) points (97.2% accuracy). Rates were 99.2% for DTW when using \( n = 99 \), 98.8% with just \( n = 9 \) points \( (Z = -8.377, p < 0.001, r = 0.14) \) and 99.1%

Fig. 14. A snapshot of the GDATK tool for testing recognition performance under various sampling rates and bit depths.
for $n=14$ (n.s.). For this set, the toolkit found that $n=10$ points for Euclidean, $n=6$ for Cosine, and $n=14$ for DTW delivered high recognition rates nonsignificantly different than those obtained with much finer resolutions. This led to classification times up to 15x smaller for Euclidean and Cosine, and up to 50x smaller for DTW.

### 8.2. Recognition performance on the 6DMG set

The set contains 20 acceleration gestures (see Fig. 16) performed by 28 participants with 10 repetitions, giving a total number of 5600 samples. The set can be downloaded from. \(^{15}\) For this set, the mean number of points is 69 (sd=32.1) and was used as reference for recognition tests.

The Euclidean metric delivered 94.6% performance for $n=69$, 93.8% for $n=6$ ($Z=-7.931, p<0.001, r=0.12$), and 94.5% for $n=8$ (n.s.). Cosine delivered 94.5% for $n=69$, 93.8% for $n=6$ ($Z=-6.757, p<0.001, r=0.10$), and 94.5% for $n=8$ (n.s.). DTW rates were 98.0% for $n=69$, 97.6% for $n=13$ ($Z=-9.580, p<0.001, r=0.14$), and 97.9% for $n=20$ (n.s.). For this set, the toolkit found $n=8$ points sufficient for Euclidean, $n=8$ for Cosine, and $n=20$ for DTW to deliver high recognition rates, nonsignificantly different than those obtained with finer resolutions. In turn, classification times were 8x smaller for Euclidean and Cosine, and 11x smaller for DTW.

### 9. Conclusion

This work investigated the impact of gesture dimensionality and bit cardinality on the performance of 3D gesture recognizers. In the recent context of empowering practitioners with easy-to-implement gesture recognizers (such as $S_1$, $S_N$, and $S_P$) to encourage experimentation of gesture-based interfaces for new platforms and environments, we showed that careful analysis of gesture representation is extremely important when designing under constrained resources. We hope the results of this study will complement today’s knowledge and practices of implementing gesture recognizers by informing on gesture representation aspects. Particularly, such aspects become very important when practitioners need to develop for miniaturized gadgets and devices in order to meet the interactive needs of future environments.

While our results have a clear and direct impact on prototyping gesture recognition systems, they can also serve as basis to inform more theoretical investigations into fundamental gesture input. For example, our findings that capturing human gestures does not require the full amount of precision delivered by today’s sensors follow the findings of Bérard et al. (2011) which showed that the human motor control system is much more limited than the fine DPI resolutions delivered by today’s mouse devices. We leave such explorations regarding the capacity of human gesture execution as future work.

This work must be seen as the first attempt to explore representation aspects of 3D gestures in order to provide practitioners with guidelines on how to optimize classification times and reduce memory requirements for their designs. Toward this goal, we provided a detailed exploration of the impact of gesture dimensionality and bit depth in the form of empirical findings, theoretical model, and assisting toolkit. Probably the most important implication of our work will be in encouraging practitioners to adopt an analysis-driven design approach for their prototypes while we showed that the simplistic and uninformed yet common way of handling a problem by throwing more resolution at it (in terms of memory and processing) won’t necessarily improve overall performance. We hope our findings will benefit the new wave of practitioners prototyping gesture-sensing devices and gadgets for the interactive needs of the upcoming ubiquitous computing era.

### References


Stahovich, T., 2011. Sketch-based Interfaces and Modeling. Springer, Ch. Pen-based Interfaces for Engineering and Education.


