Available online at www.sciencedirect.com

SciVerse ScienceDirect

Int. J. Human-Computer Studies 71 (2013) 387–409

International Journal of Human-Computer Studies

www.elsevier.com/locate/ijhcs

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.

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

*Fax: +40 230 524801.
E-mail addresses: raduvro@yahoo.com, vatavu@eed.usv.ro.
URL: http://www.eed.usv.ro/~vatavu

1071-5819/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.ijhcs.2012.11.005

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 $-family of gesture classifiers ($1, $N, $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 and Baudisch, 2009) 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 $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 × 10 training samples × 64 points × 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® and iPad®.