Advance Access publication on 1 July 2015
doi:10.1093/iwc/iwv020

Employing Number-Based Graphical Representations to Enhance the Effects of Visual Check on Entry Error Detection

Huawei Tu[1,2,*], Patrick Oladimeji[2], Sarah Wiseman[3], Harold Thimbleby[2], Paul Cairns[4] and Gerrit Niezen[2]

[1] *College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China*
[2] *Future Interaction Technology Lab, Department of Computer Science, College of Science, Swansea University, 5th Floor, Faraday Tower, Singleton Park, Swansea SA2 8PP, UK*
[3] *UCL Interaction Centre, University College London, London, UK*
[4] *Department of Computer Science, University of York, York, UK*
**Corresponding author: tuhuawei09@gmail.com*

Number entry is a mundane and error-prone task. To find errors, users often rely on visual checks to compare the differences between their instructions and the numbers they have actually input, a task that is difficult for users to do accurately. We therefore propose the use of number-based graphical representations (GRs) as a complement to conventional numeric representations (NR) to enhance visual checks, so users can examine both GRs and NRs to detect errors. We conducted two experiments to explore the issues raised. Experiment 1 examined the effects of GRs and NRs on *representation difference detection* (i.e. checking if two GRs or NRs are identical). The two representations had a comparative performance by time and error rate. In Experiment 2, we investigated the performance of GRs and NRs with number entry tasks. While extending the task time (increased by 38%), number entry with GRs resulted in significantly fewer errors than without GRs (decreased by 60%). Participants also had a high preference for number entry with GRs. Therefore, the proposed technique is promising for number entry error reduction, and that in safety critical applications improved safety can be achieved.

RESEARCH HIGHLIGHTS

- A novel technique, which is based on number-based graphical representations, is proposed to aid users in entry error detection.
- The technique has three advantages. First, it improves entry error detection. Second, there is no extra cost in performing entry process. Last, it is independent of the graphical entry interface it is used with.
- Results of two user studies demonstrate the effectiveness of the proposed technique.
- The technique combines the benefits of using graphs with the familiarity of conventional numeric notations, hence giving much broader insights into numeric interaction.

Keywords: graphical user interfaces; novel interaction paradigms; user interface design; empirical studies in interaction design

Received 18 December 2014; Revised 27 May 2015; Accepted 8 June 2015

1. INTRODUCTION

Number entry is a crucial aspect of using many interactive systems. In safety-critical environments such as air traffic control or in hospital wards, number entry errors can lead to harmful consequences. For example, in 2008, a patient was given drugs at a rate of 68 mL/h rather than at the ordered rate of 6.8 mL/h; this error led to the patient's death (FDA Newsletter, 2014). Indeed, a large proportion of adverse incidents in hospitals have occurred as a result of number entry errors in programming infusion pumps (Vicente *et al.*, 2003). Reduction of entry errors is important to improve the dependability of number entry systems (Thimbleby and Cairns, 2010).

Number transcription is usually composed of several steps: viewing instruction numbers, remembering them in short-term memory, interacting with a user interface, then double-checking the results. The last step is important to reduce entry errors, because unlike text entry, there is no model (such as a word dictionary) that can suggest corrections for numbers as any combination of digits constitutes a legal number.

Visual checking is a widely adopted method due to its advantages of high user preference and short execution time (Barchard and Pace, 2011; Barchard and Verenikina, 2011; Wiseman *et al.*, 2011). Checking simply requires to compare entries with instructions to determine if they match. Nevertheless, despite the popularity and advantages, visual checking tends to have worse effects on error detection than other check methods, like double entry (i.e. entering data twice to validate the results) (Barchard and Verenikina, 2011). Given the widespread use of visual checking for number entry in safety critical applications, there is a clear need to improve its performance.

Entry error detection involves checking for differences between the entered number and the instructions. Ease of search and recognition is affected by what information is explicit in a representation (Larkin and Simon, 1987). For instance, grouping digits can enable users to better recognize, memorize and recall the chunk information (Miller, 1956; Nordby *et al.*, 2002). To ensure error detection efficiency, it is important to seek representations which can support a comparable, if not faster and more accurate, search and recognition performance than just using numeric digits, or numeric representations (NRs).

In this study, we propose a method of using number-based graphical representations (GRs) to help users identify entry errors. The use of GRs plays an important role in many areas such as data analysis (Larkin and Simon, 1987) and clinical setting (Gould *et al.*, 2013; Thimbleby and Williams, 2013). To display numbers graphically, we designed a mechanism that constructs a GR corresponding to a number; each number thus

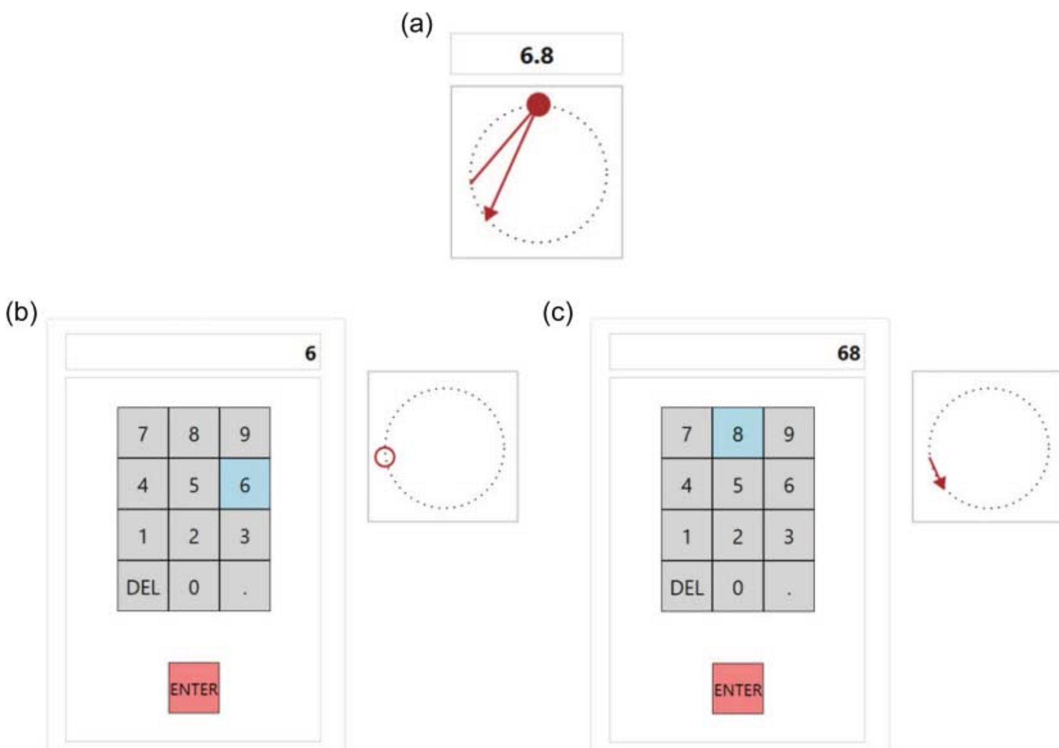

Figure 1. Number entry with graphical representations (GR). (**a**) Instruction number '6.8' and its GR. (**b**) The user enters digit '6', along which the entry-based GR is updated. (**c**) The user enters digit '8' and the GR is updated again. The GRs in (a) and (c) are served as a complement to the two numbers in (a) and (c) for entry error detection.

having a unique GR. During the entry process, an instruction number is presented in conjunction with the related GR, as illustrated in Fig. 1a. An entry-based GR is generated along with the number users input (see Fig. 1b and c). To identify errors, users can compare the instruction GR with the entered GR, in addition to comparing the input number with the instruction number. This provides a novel alternative to detecting entry errors. In addition, the design combines the benefits of using graphs with the familiarity of conventional numeric notations, hence may give much broader insights into interaction than with just numbers. To our knowledge, we are the first to employ GRs to enhance the effects of visual check on entry error detection.

The rest of this paper is organized as follows. We first describe the motivations and rationales of using GRs for checking entry accuracy, followed by the design and implementation of GRs. After a review of related work, we report two experiments to examine the effects of GRs with an analysis of experiment data. Last, we generally discuss the results and future work.

2. MOTIVATING PRINCIPLES

2.1. Enhanced effects of entry error detection

As shown in Fig. 1a and c, our design provides a pair of GRs (instruction GR and entry GR) as an accompaniment to a pair of NRs (instruction and input numbers) to increase users' awareness of the differences between the entry and instruction. Intuitively, the dual-mode representations could improve error detection rates. Furthermore, graphical and numeric information are processed differently and along distinct channels in the human mind, creating separate representations for information processed in each channel (Larkin and Simon, 1987). The mental codes corresponding to these representations are used to organize incoming information that can be acted upon, stored and retrieved for subsequent use. When viewing GRs, early visual processes detect and encode visual primitives such as shape, position and length. Some of these primitives can be processed in parallel; some require serial scanning and identification. On the other hand, viewing digits of a number requires serial scanning the digits. The independent mechanisms of information search and recognition for NRs and GRs indicate that a combined use of the two representations should lead to improvements in error detection.

2.2. No extra cost for entry process

The use of GRs for entry error detection should not require the user to learn new entry skills or perform extra entry steps. This is important to make our technique applicable in practice, because long-term learning and extra entry burden may hinder the wide use of the proposed technique.

2.3. Graphical entry interface independence

The proposed technique should be independent of number entry interfaces. In other words, no matter what graphical interface types are used (e.g. numeric keypad or up-down keypad (Oladimeji *et al.*, 2013)), correct input numbers should have a same GR as the instruction number. This design feature has a significant role in practice. Taking clinical practice as an example, preparing prescription forms and programming infusion pumps are generally performed by different medical practitioners in different departments. When preparing instruction GRs, users do not need to be concerned with the type of number entry interfaces being used.

3. CONSTRUCTING GRAPHICAL NUMBER REPRESENTATIONS

To generate a GR for a number, we need to address how to graphically represent the digits (and the decimal point) in the number and their sequence. There are two design rules to be considered when constructing number-based GRs. One is to design GRs that are as simple as possible, because it is easier for users to view two simple GRs and identify their differences than complex ones. The other is to ensure that each number has a unique GR. Otherwise, the technique may lose its effectiveness if two different numbers share the same GR. Following the two rules, we designed a mechanism to create number-based GRs. As shown in Fig. 2a, the GR is rendered within a dotted circle. Ten digits (0–9) and the decimal point

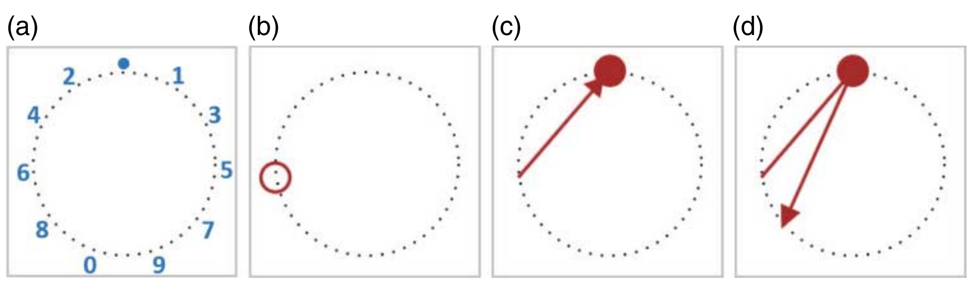

Figure 2. (**a**) Positions on the dotted circle and their corresponding digits. GR representing (**b–d**) typing the successive digits of '6.8'.

(●) are mapped to eleven positions on the dotted circle; these positions are evenly distributed on the circle and the position of the decimal point is at the top of the circle.

An effective GR design should ensure that the entry-based GR differs from the instruction-based GR if an incorrect entry occurs; the greater the difference is, the better the design is. Our pilot study showed that for calculator keypads, which are a common number entry interface, users sometimes pressed a key next to the intended one, resulting in a wrong digit error. Hence, for adjacent keys on the numeric keypad, the positions of the digits they represent should be as far apart as possible on the dotted circle, so as to increase differences between entry and instruction GRs if a wrong digit error occurs. We therefore placed the odd digits on the left side of the circle and the even digits on the right side, as shown in Fig. 2a. Admittedly, it is difficult to propose a way of distributing digits on the dotted circle which works properly for all number entry interfaces. We aim to provide a good design for a widely used interface (calculator keypads) in this study and will investigate different digit distributions for other entry interfaces in future.

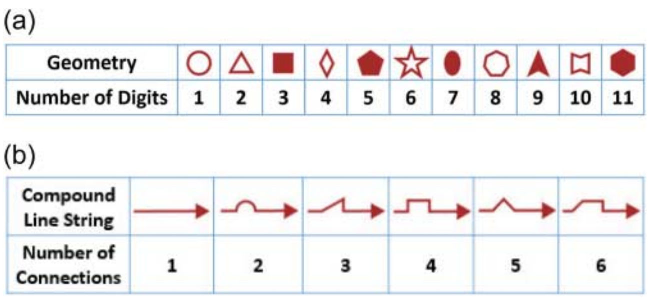

Figure 3. (**a**) Geometric shapes and the repetition times they represent. Apart from the circle, solid and hollow shapes appear alternately to make users easily perceive the number of digits. (**b**) Compound line strings and the number of connections they represent. Note we only present certain numbers of shapes and lines strings due to space restrictions.

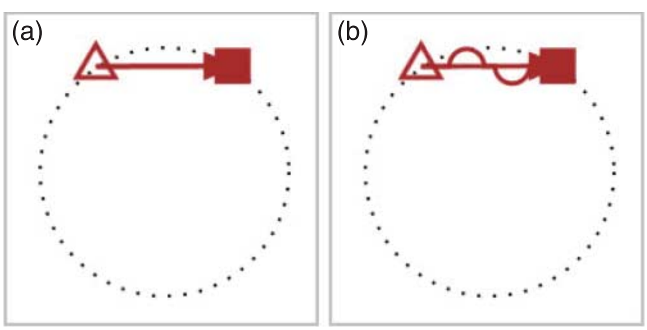

Figure 4. (**a**) Initial GR design for the number '11221' and '12121'. (**b**) Modified GR design for the number '12121'. For '11221', the modified GR design displays the same pattern as found in (a).

The process of generating a GR is as follows. A number can be regarded as a sequence of digits. For the first digit, a small circle is displayed in the designated position on the dotted circle (Fig. 2b). The small circle disappears when the subsequent digit or the decimal point is input. A line connecting the positions of the two consecutively input characters is displayed, with an arrow indicating the order (Fig. 2c). This step iterates until the last digit is input (Fig. 2d). A solid shape placed at the top of the circle is used to represent the decimal point. This makes the decimal point prominent in the GR (see Fig. 2c), as decimal point related errors are serious and common (Oladimeji *et al.*, 2011; Wiseman *et al.*, 2011).

In designing the mechanism, we paid attention to two special number types. One refers to the number having repeating digits like '222215'. There is no need to depict the sequence for the '2222' part as the digits are the same; instead, we need to show the repeat count. We designed a set of solid and hollow geometric shapes to represent the number of repetitions, for example, a triangle means that a digit appears twice. Figure 3a lists a set of geometric shapes and the repetition times they represent. Note the open circle, which is designed to indicate digits appearing once, is only used in 1-digit number graphs. We do not use it for multi-digit number graphs, as the designated positions on the dotted circle for the digits can fulfill the purpose. In addition, graphs are more likely to be simpler without open circles than with them. This follows the first design rule. In Fig. 4b, the hollow triangle on the dotted circle indicates digit '2' has been entered twice, and the solid square means '1' has been entered three times.

The other type is the number that contains repeating digit groups (i.e. one that consists of two or more digits) such as '11221' or '12121'. Without further modification, the two numbers have an identical GR shown in Fig. 4a. The reason is that the line connecting digit '1' and '2' does not encapsulate their sequence. To solve this issue, we designed a set of compound line strings to depict the number of times that a digit sequence occurs (see Fig. 3b). For instance, both '12'

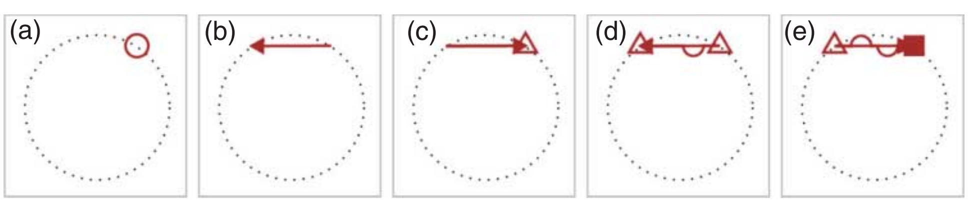

Figure 5. An example of using half-circle line strings to represent the connection between '1' and '2'. (**a–e**) shows the step-by-step generation of the GRs for '12121'.

and '21' occurs twice in the sequence '12121' so we use a half-circle line string to represent the connection from '1' to '2' and from '2' to '1', respectively (see Fig. 4b). Note that when the connection is from a small digit (e.g. '1') to a bigger one (e.g. '2'), the shape denoting the repetition count (e.g. the half-circle) on the line string is within the major segment defined on the dotted circle by the chord formed by the line string; otherwise, the shape is within the minor segment (see Fig. 5).

Overall, for a number, each digit has a unique position on the dotted circle and the digit sequence can be represented by a unique set of line strings, hence each number has a unique GR.

4. RELATED WORK

4.1. Reduction of number entry errors

As humans we are all prone to error (Reason, 1990), even in some seemingly mundane tasks such as number entry. To reduce number entry errors, it is important to understand the errors being made. In a noteworthy study, Wiseman *et al.* (2011) proposed a taxonomy for classifying and organizing 350 collected entry errors to facilitate further research. The taxonomy was based on Norman's Action Cycle (Norman and Draper, 1986): one for errors whilst taking action and the other for errors whilst evaluating action. The former category includes goal slips (e.g. digit(s) missing), intention slips (e.g. leading zero omission), action specification slips (e.g. decimal instead of 0) and execution slips (e.g. 0 instead of decimal). The latter one has perception slips (e.g. digit(s) missing), interpretation slips (e.g. leading zero correction), and action evaluation slips (e.g. one digit wrong). We will evaluate the performance of our technique with reference to the taxonomy.

We summarized previous studies related to reduction of number entry errors into three main categories: interface and technique design, number related factors and task contexts. The role of interface and technique design is recognized for improved entry accuracy, resulting in many studies. Some studies have investigated the effects of visual check, read aloud (one person reads instructions out loud while the other person examines entries) and double entry on data entry (Barchard and Pace, 2011; Barchard and Verenikina, 2011). Generally, double entry resulted in fewer entry errors, but at the cost of a longer entry duration. This is also reflected in the study by Wiseman *et al.* (2011), which examined the performance of double entry and visual check on the application of checksum technique on entry error reduction. Oladimeji *et al.* (2011, 2013) conducted a series of studies to investigate the effects of entry interface types (e.g. numeric keypads vs. up-down keypads) on speed, errors and severity of errors. Up-down keypads led to fewer errors than numeric keypads. While these studies provided useful methods to achieve high entry accuracy, none of them can eliminate error entirely.

Tu *et al.* (2014) summarized number-related factors into numeric attributes and presentation variables. Numeric attributes include number length (short or long), number type (integer or decimal), and magnitude and frequency. Presentation variables comprise presented positions (near or far) and font appearance. Their study examined how number length, type and position affect entry performance. Long numbers had more errors than short numbers. Number type and position did not affect entry errors. Psychology studies have focused on the effects of number length and digit-grouping formats on immediate recall (Miller, 1956; Nordby *et al.*, 2002). The capacity of short term memory is considered as 7 ± 2 elements and grouping digits can improve recall effects. The magnitude and frequency of numbers used in hospitals have also been studied to improve number entry interface design for medical devices (Wiseman *et al.*, 2013). Furthermore, some studies have investigated visual aspects of numbers to improve NR. For example, Thimbleby (2013) discussed some problematic uses of seven segment displays and argued a need for more legible alternatives than standard seven segment displays. Studies have showed that decreasing numeric presentation quality can increase entry accuracy (Soboczenski *et al.*, 2013). Studies (Sandnes, 2013; Sandnes and Huang, 2013) also proposed to map numbers into phrases so that the user remembers phrases instead of value sequences when performing number copying tasks. Results showed this strategy can reduce the chances of human errors.

Besides interface design and number factors, some studies also examined how task contexts like the emotional state of the user affect entry errors (Cairns *et al.*, 2014). Results showed people in negative emotions (e.g. anger) are less accurate than those in positive emotions (e.g. contentment).

4.2. Graphical representations

The advantages conferred by GRs has attracted widespread interest in the areas of cognitive science and human–computer interaction. Here we highlight a few lines of work that bear direct relevance to the design we proposed. Larkin and Simon (1987) proposed that diagrammatic representations exploit perceptual processes by grouping together relevant information and hence make processes such as search and recognition easier than sentential representations (e.g. text). Zhang and Norman (1994) found that multiple representations of the same task can offer different degrees of external representations, which in turn affects problem-solving efficacy. Their study inspired the present work.

Some studies have applied GRs to error reduction. For example, Thimbleby and Williams (2013) have proposed the use of nomograms—1D slide rule-like representations—to support users in calculating drug doses. Results showed that these representations can significantly reduce the number of errors made in medical dosage calculation tasks. Inspired by Thimbleby's work, Gould *et al.* (2013) proposed a variation of GRs to make it easier for users to reason about the setup of infusions. The purpose of using a bar graph with a scale ruler is not the same as using GRs in our design. The former method is aimed at improving off-by-order-of-magnitude errors, while the latter helps detect even very small number entry errors. The review results show that no studies have employed more abstract GRs to help users detect number entry errors.

5. EXPERIMENT 1: EFFECTS OF GR AND NR ON REPRESENTATION DIFFERENCE DETECTION

Entry error detection is a task of finding out representation differences between the instruction and the entry, i.e. seeing if the instruction number and the entered number match. In this experiment, we examined the user's ability to detect representation differences in terms of GRs and NRs, respectively. We presented a set of GR pairs and a set of NR pairs to the participants and asked them to check if the two elements in each pair were different.

5.1. Experiment apparatus

The experiment was conducted on a Dell S2340T multi-touch monitor connected to an Asus S550c laptop. The monitor was 53.32×31.20 cm with a resolution of 1920×1080 pixels. The laptop had an Intel CORE i5 CPU and 8 GB memory. During this experiment, the screen was mounted on a fixed surface perpendicular to the surface of the desk. Participants were asked to use the experimental interface with a DELL USB optical Mouse.

5.2. Representation pairs

In this experiment, participants were required to check if a representation (NR or GR) pair had identical or different elements. Each representation had 80 pairs, which were divided into two identification types: 40 *identical element pairs* and 40 *different element pairs*. The *identical element pair* reflects the case where the instruction and entry are the same while the *different element pair* means the instruction and entry are different. As the GR pair is generated based on the NR pair using the methods presented in section ‘Constructing Graphical Number Representations’, we only detail the way of creating NR pairs.

We produced three groups of numbers: instruction numbers (Group A), correctly entered numbers (Group B) and wrongly entered numbers (Group C). In Group A, 40 numbers were randomly created based on Tu *et al.* (2014), which revealed the effects of number length and type on entry performance. All numbers were different. Twenty numbers were of the *integer type* and the others were of the *decimal type*. Numbers fell into two further categories according to their number lengths: for both integer and decimal numbers, there were ten *short numbers* (five 2-digit numbers and five 3-digit numbers) and ten *long numbers* (five 7-digit numbers and five 8-digit numbers). The length of the short and long numbers were determined based on previous study by Nordby *et al.* (2002), who showed that immediate recall performance degraded rapidly beyond six digits.

The numbers in Group B were identical to those in Group A, as Group B consisted of correctly entered numbers. To create incorrectly entered numbers in Group C, we copied the numbers in Group A to Group C and modified them according to the six main error types summarized in previous studies (Oladimeji *et al.*, 2011, 2013): wrong digit, digits transposed, digit missing, digit added, decimal point missing and decimal point added. For each number length in each type, the modification to each number randomly referred to one of the six error types; different numbers having different error types. Note we did not generate missing decimal point errors for *integer type* and decimal added errors for *decimal type*.

To create a NR pair with identical elements, we selected a number from Group A and the corresponding number from Group B. To construct a NR pair with different elements, we chose a number from Group A and the corresponding number from Group C.

5.3. Participants

Sixteen participants (7 female) from 23 to 48 years of age ($\mu = 33.2$, $\sigma = 7.2$) took part in the experiment. One was left handed. All knew how to use the mouse well. All of them had normal or corrected to normal vision, and no motor impairments. Participants were staff members in the local university and received £5 for their time.

5.4. Experiment design

The experiment was a within-subject repeated measures design. There were three independent variables: representation type (graphical representations (GRs), and numeric representations (NRs)), identification type (pairs with identical elements and with different elements) and number length (short numbers and long numbers). The dependent variables were identification time, identification errors and the subjective preference of participants.

5.5. Task and procedure

The experiment involved two tasks: NR-related task, in which a number pair was displayed (see Fig. 6b) on the screen, and GR-related task, in which a GR pair was rendered on the screen (see Fig. 6d). The font size of the numbers was 6.75 mm. The radius of the dotted circle which held the GR was 2.16 cm. The font size and the circle were large enough to show the details of GRs clearly. Each task had 80 trials. In each trial, a representation pair was randomly selected from the corresponding collection generated in the section ‘Representation Pairs’. Different trials had different representation pairs. For each trail, participants were instructed

Figure 6. Illustrating the experimental conditions before and after pressing the START key. (**a**) NR condition, before pressing START. (**b**) NR condition, after pressing START. (**c**) GR condition, before pressing START. (**d**) GR condition, after pressing START.

to determine if the two representations shown were the same as accurately and fast as possible. Once they had determined this, they were required to confirm their selection by clicking a key with the mouse; the green SAME key indicated the two representations were the same and the red DIFFERENT key meant they were different. After finishing the current trial, participants were required to press the START key for the next trial (see Fig. 6a and c). The purpose of using different colors on the keys was to increase participants' awareness of the key that they intended to click.

The experiment consisted of a practice phase and a test phase. In the practice phase, participants sat in a chair in front of the experiment device and were instructed how to perform the task. During the test, each participant was required to do 80 trials for each task. The test trials were different from the practical trials to avoid learning effects. The order in which the representation pairs were shown to each participant was randomized. The task order was counter-balanced across the participants. In summary, the experiment consisted of 16 participants $\times$ 80 trials $\times$ 2 representations $\times\,(1 + 1)$ blocks (practice + test) = 5120 trials.

At the end of the experiment, a questionnaire was administered to gather subjective opinions. The two representations were rated by the participants on three dimensions: identification time, identification accuracy and ease of identification. Participants were required to rate these representations on a 7-point scale (1 for worst, 7 for best).

5.6. Data analysis

Because the task was to identify representation differences, we do not need to consider 'motor learning' effects. If the participants could understand how to perform the task after the practice, it can be regarded that the data collected in the test phase are able to reflect user performance on representation difference identification. The practice block was removed from the data analysis. Before analyzing variance, we observed histograms of the data and found that it appeared to fit the normal distribution.

5.6.1. Identification time

Identification time was calculated as the duration from clicking the START key to clicking the SAME or DIFFERENT key for confirmation. Short identification time indicates that users can easily determine if the elements of the representation pair were different.

We began by looking at mean identification time by representation types (NRs vs. GRs) averaged across identification types (pairs with identical elements vs. pairs with different elements). This gives equal weight to each identification type and provides an overview performance of the representation types. Repeated measures ANOVA shows representation type had no significant main effect on identification time ($F_{1,15} = 4.19$, $P = 0.06$). The mean value was 2335 ms in the NR condition and 2173 ms in the GR condition. For both

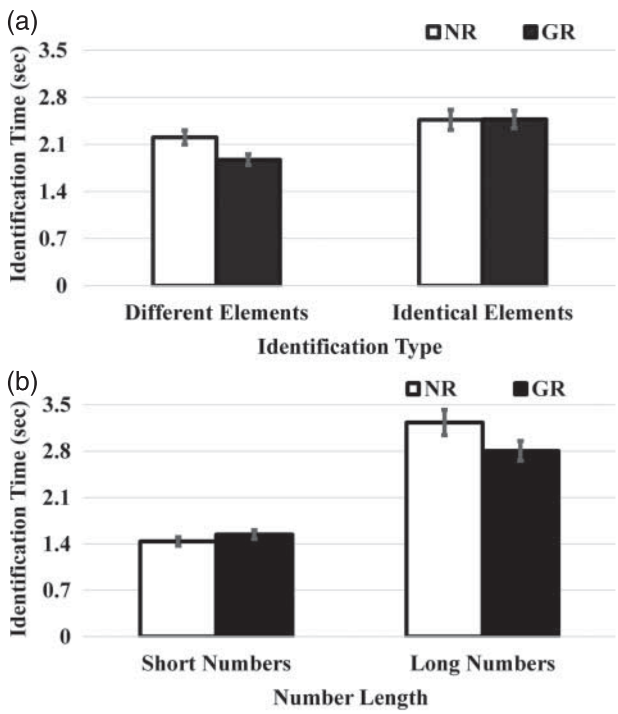

Figure 7. Identification time for each representation type in different (**a**) identification types and (**b**) number lengths. Error bars represent 0.95 confidence interval.

representations, participants spent similar time identifying representation differences.

As illustrated in Fig. 7a, representation type significantly interacted with identification type ($F_{1,15} = 30.79, p < 0.01$). For the identification type with different elements, GRs had a shorter time than NRs. The mean time was 2204 ms for GRs and for 1873 ms NRs. Conversely, for the identification type with identical elements, GRs had a slightly longer time than NRs. The mean time was 2466 ms for GRs and for 2473 ms NRs.

To investigate the effect of representation type for each identification type, we further calculated the simple main effect of representation type at two levels of the identification type, respectively. Representation type had a significant main effect on identification time for the identification type with different elements ($F_{1,15} = 19.35$, $P < 0.01$), but no significant main effect for the identification type with identical elements ($F_{1,15} = 0.01$, $P = 0.94$). With GRs, participants were able to detect representation differences in a shorter time than NRs.

There was also a significant interaction between representation type and number length ($F_{1,15} = 20.91, p < 0.01$). As shown in Fig. 7b, compared with NRs, GRs led to a longer average time in short number condition (1542 ms for GRs vs. 1438 ms for NRs) but a shorter average time in long number condition ($\mu = 2804$ ms for GRs vs. 3232 ms for NRs). We looked into how number length affected the performance of representation types by analyzing the simple main effect of representation types at two levels of the number length, respectively. We found that representation type had a significant main effect for both number lengths ($F_{1,15} = 5.00$, $P < 0.05$ for short numbers, and $F_{1,15} = 10.70$, $P < 0.01$ for long numbers). With GRs, participants tended to spend a shorter time for long numbers than with NRs, but the reverses was true for short numbers.

5.6.2. *Identification errors*

An identification error was committed when the participant clicked the *same* key for the representation pair having different elements or clicked the *different* key for the representation pair having identical elements. We analyzed error rate using repeated measures ANOVA. No significant main effect was found on identification errors for representation type ($F_{1,15} = 0.49$, $P = 0.49$). The average rate was 0.020 for NRs and 0.016 for GRs. In other words, the two representation types did not significantly differ in identification errors. However, a significant main effect was found on identification errors for identification type (whether the elements are the same or not) ($F_{1,15} = 12.24$, $P < 0.01$) and number length ($F_{1,15} = 12.02$, $P < 0.01$). The mean error rate for the identification type with different elements ($\mu = 0.03$) was significantly larger than that for the type with identical elements ($\mu = 0.01$). Long numbers resulted in significantly more errors than short numbers ($\mu = 0.01$ for short numbers and 0.03 for long numbers).

There was no significant interaction effect between representation type and identification type ($F_{1,15} = 0.04$, $P = 0.85$), as well as between representation type and number length ($F_{1,15} = 0.03$, $P = 0.88$).

5.6.3. *Subjective feedback evaluation*

Repeated measures ANOVA revealed no significant main effect for representation type on subjective identification time ($F_{1,15} = 1.52$, $P = 0.24$) and on identification accuracy ($F_{1,15} = 2.19$, $P = 0.16$). The mean rate of speed was 4.38 for NRs and 5.13 for GRs. And the mean rate of accuracy was 4.25 for NRs and 5.13 for GRs. The subjective preference of the two representation types was fairly consistent with the time and error performance. Interestingly, a significant main effect for representation type was found on ease of identification ($F_{1,15} = 5.08$, $P < 0.05$). Participants generally felt that it was easier to detect representation differences with GRs ($\mu = 5.19$) than with NRs ($\mu = 4.06$).

5.7. Discussion

The data analysis of identification time and errors shows that GRs led to an overall comparable performance than NRs on representation difference detection. Therefore, it is reasonable to anticipate that the effect of GRs on entry error detection would not be worse than that of NRs. Furthermore, according to identification time analysis for identification type with different elements, GRs were better at highlighting differences than NRs. The subjective feedback evaluation also shows higher preference for GRs over NRs in terms of ease of detection. Overall, the results show that GRs are promising to aid users in entry error identification.

Note that our basic goal is to use GRs as a complement to NRs. If the use of GRs is completely compatible with existing input interfaces and is preferred by a meaningful number of users, it does not need to outperform NRs in performance to be valuable. The similar performance of GRs and NRs indicates potential advantages of using GRs for entry error detection.

6. EXPERIMENT 2: NUMBER ENTRY WITH OR WITHOUT THE PRESENTATION OF GR

The results in Experiment 1 reflect potential benefits of GRs on number entry error detection. In Experiment 2, we further tested the effects of GRs in the context of a number entry task with the presentation of GRs. While it is reasonable to expect that two detective patterns can result in a better error detection performance than one, it is still important and informative to measure the performance of number entry with GRs in comparison with number entry without GRs in order to gain empirical insights. Another goal of this experiment was to close the design-iteration loop by testing with participants who had no knowledge of, or bias from, the research and design insight. We aim to observe users' actual behaviors on number entry with GRs and evaluate the strengths and weaknesses of GRs.

6.1. Experiment apparatus

The monitor and laptop in Experiment 1 were used in Experiment 2. During the experiment, the touch screen was mounted on a fixed surface tilted at 30° to the surface of the desk to reduce fatigue (Sears *et al.*, 1993). The participants were asked to input numbers with direct touch.

6.2. Instruction numbers

We randomly generated two number sets for the experiment following the method in Experiment 1. One is for the practice phase and the other is for the test phase. Each set had 104 different numbers randomly generated based on Tu *et al.* (2014). For each set, 52 numbers were of the *integer type* and the others were of the *decimal type*. Numbers fell into 2 further categories according to their number lengths: for both integer and decimal numbers, there were 26 *short numbers* (12 2-digit numbers and 14 3-digit numbers) and 26 *long numbers* (12 7-digit numbers and 14 8-digit numbers). For decimal numbers, the location of the decimal point was evenly distributed amongst the digits. The length of the short and long

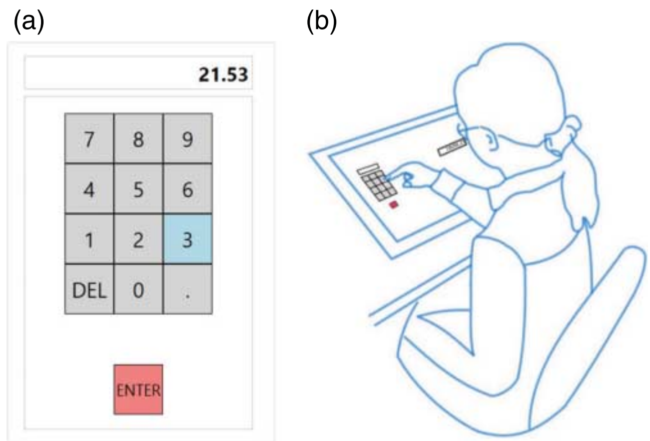

Figure 8. (**a**) Experiment interface without GRs. (**b**) A participant in the experiment environment. The instruction number was shown on the right of the entry interface.

numbers were determined based on previous study by Nordby *et al.* (2002).

6.3. Number entry interfaces

The user interface without GRs (*NoGR interface*) was a calculator keypad (see Fig. 8a). Twelve square keys in gray were arranged without key spacing. The key size was 18.9 mm per side, which was large enough to be clicked with a finger (Sears *et al.*, 1993). A red key (measuring 18.9 mm per side) was placed under the number pad with a constant gap of 18.9 mm, for controlling the start and end of a trial. The text of the control key was set as START initially. Pressing the START key indicated starting a trial. Once a trial started, the key text changed to ENTER. Participants were asked to click this key when they finished a trial. The key text then changed to START again, prompting participants to start the next trial. During the entry process, participants were asked to use the DEL key to remove digits one by one from the most right digit and were not allowed to position the cursor by touching the screen. The touchscreen used the lift-off key activation mechanism with visual feedback. During the experiment, the user interface was to the left side of the screen (see Fig. 8b), and the instruction (font size 6.75 mm: 25 pixels) was shown on the right side of the entry interface. The font size of the input number was 6.75 mm. The font size was large enough to allow participants to read the numbers clearly.

For the interface with GRs (*GR interface*), instruction numbers and input numbers were presented along with the related GRs (see Fig. 1).

6.4. Participants

Thirty-two participants (13 male) from 24 to 47 years of age ($\mu = 33.4$, $\sigma = 6.8$) were recruited for the study. Sixteen participants were asked to perform number entry tasks with the *NoGR interface*, and the others were asked to enter numbers with the *GR interface*. Six were left-handed. All had prior experience using touch screens and were familiar with number entry on numeric keypads. All of them had normal or corrected to normal vision, and no motor impairments. Participants were staff members in the local university and received £5 for their time.

6.5. Experiment design

The experiment was a between-subjects design. The number entry interface style was the independent variable and it had two levels: *NoGR interface* and *GR interface*. The dependent variables were entry time, the number of undetected errors, the number of corrected errors and subjective evaluation.

6.6. Task and procedure

The experiment consisted of a practice phase and a test phase. In the practice phase, participants were first instructed how to perform the task. Then participants were asked to sit in a chair in front of the experiment device (see Fig. 8b) and enter 104 numbers following the instructions. During the test, each participant was required to enter 104 numbers as well. The instruction numbers used in the test phase differed from numbers used in the practice phase. Participants were asked to use the index finger of the dominant hand to enter the presented number as accurately and quickly as possible. Participants were required to correct entry errors using the DEL key if they found any. They were then instructed to press the ENTER key to confirm the current trial and then press the start key for the next trial. The order in which the numbers were presented for each participant was randomized.

For the *NoGR interface* condition, in each trial, an instruction number was shown on the right of the entry interface. For the *GR interface* condition, an instruction number and its related GR was shown on the right of the entry interface. The font size of the instruction number and the input number was 6.75 mm. The GR was displayed within a dotted circle with a 2.16 cm radius. During the entry process, a GR would be automatically generated according to the input number.

In summary, the experiment consisted of 32 participants $\times 104$ numbers $\times$ $(1+1)$blocks (practice + test) = 6656 trials. After the experiment, we asked the participants to evaluate the number entry interface they used.

6.7. Data analysis

Only data gathered in the test phase was analyzed.

6.7.1. Entry errors

Entry errors were classified into uncorrected errors and corrected errors (Oladimeji *et al.*, 2011). Uncorrected errors were trials for which the user transcribed and confirmed a wrong number. Corrected errors were defined as the number of DEL keystrokes used to delete incorrectly entered digits and decimal points. For each error type, error rates, calculated as the ratio of the sum of errors to the number of all trials, were analyzed using the Wilcoxon signed-rank test because the data were not normally distributed.

No significant difference was found between number entry with GRs ($\mu = 0.163$) and without GRs ($\mu = 0.167$, $Z = -1.02$, $P = 0.16$) for corrected errors. However, it was found that number entry without GRs ($\mu = 0.025$) had significantly more uncorrected errors than that with GRs ($\mu = 0.010$, $Z = -1.78$, $P < 0.05$). Recall the purpose of this study was to reduce uncorrected number entry errors. The result is a compelling evidence that GRs are an effective supplement to NRs for entry error detection.

In total, we collected 41 uncorrected errors by 10 participants for *NoGR interface* and 14 uncorrected errors by 7 participants for *GR interface*. Inspired by previous studies (Oladimeji *et al.*, 2011; Wiseman *et al.*, 2011), these errors were divided into six categories: wrong digit(s) errors, anagram errors, digit(s) missing errors, digit(s) added errors, missing decimal point errors and decimal added error (see Table 1). We analyzed these errors in detail as follows (Fig. 9).

(i) Wrong digit(s) errors: wrong digit(s) errors occurred when at least one digit of the input number was incorrect. There were 16 instances of this error for *NoGR interface* and 7 for *GR interface*.
(ii) Transposition errors: Transposition errors happened when two adjacent digits were swapped in the transcribed value, such as inputting 954 instead of 945. Participants committed 4 instances of this error for *NoGR interface* and 0 for *GR interface*.

Table 1. Unnoticed errors for *NoGR interface* and *GR interface*.

	Error type						
Interface	Wrong digit(s)	Transposition	Digit(s) missing	Digit(s) added	Missing decimal	Decimal added	Total
NoGR interface	16	4	12	4	4	1	41
GR interface	7	0	6	0	1	0	14

(iii) Digit(s) missing errors: digit(s) missing errors occurred when at least one digit from the intended value was missing from the transcribed value. There were 12 instances of this error for *NoGR interface* and 6 for *GR interface*. Participants tended to miss digits for numbers that included more than two consecutive identical digits, as 4 of the 12 errors were related to such number type. For example, participants input ‘688663’ instead of ‘6888663’

(iv) Digit(s) added errors: digit(s) added errors were committed if at least one digit was wrongly added in the transcribed value. There were 4 instances of this error for *NoGR interface* and none for *GR interface*. Half were related to the numbers including more than two consecutive identical digits.

(v) Missing decimal point errors: missing decimal point errors occurred when a decimal point was absent from the transcribed number but was present in the instruction. There were 4 instances of this error for *NoGR interface* and 1 for *GR interface*.

(vi) Decimal point(s) added errors: decimal added errors were produced when there were more decimal points in the input number than in the instruction number. Only a single instance of this error was found for *NoGR interface*.

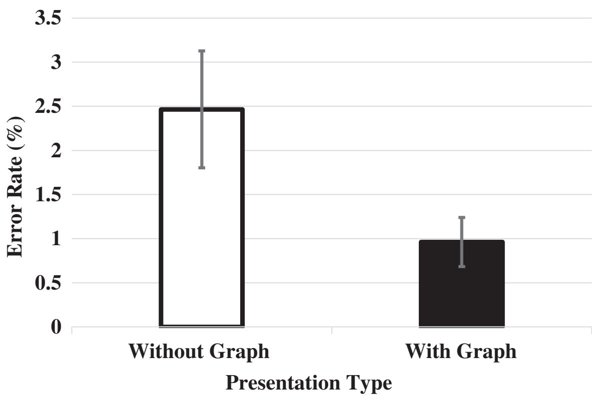

Figure 9. Error rate for number entry with or without GRs in different error types. Error bars represent 0.95 confidence interval.

6.7.2. Time performance

Time performance is a standard quantitative metric used in assessing the usability of user interfaces. Although the purpose of using GRs is to reduce entry errors, it is still important to examine time performance to gain insights of overall effects of using GRs on number entry.

For experimental trials with uncorrected errors (3.7% of all trials), these trials cannot provide valid data for time analysis, hence were removed from the experimental data for time analysis. The independent samples *t*-test was used in the data analysis, as the data had normal distribution.

Total time was computed as the period from the release of the START key to the release of the ENTER key. As illustrated in Fig. 10, *GR interface* had a significantly longer time than *NoGR interface* ($t_{30} = -5.67, p < 0.01$). The mean time was 3656 ms for *NoGR interface* and 5049 ms for *GR interface*. To look into the effects of GRs on entry process, we further measured time performance for the initiation, execution and commit phases, respectively.

Initiation time was defined as the duration from the release of the START key to the contact of the first digit key. Initiation time reflects the duration that users view and mentally process the instruction. The time for *GR interface* ($\mu = 1428$ ms) was significantly longer than *NoGR interface* ($\mu = 1081$ ms) ($t_{30} = -3.25, P < 0.01$).

Execution time was the time elapsed between the contact of the first digit key to the release of the last digit key. *GR interface* led to 12.9% longer time than *NoGR interface*, but this was not significant ($t_{30} = -1.96, P = 0.07$). On average,

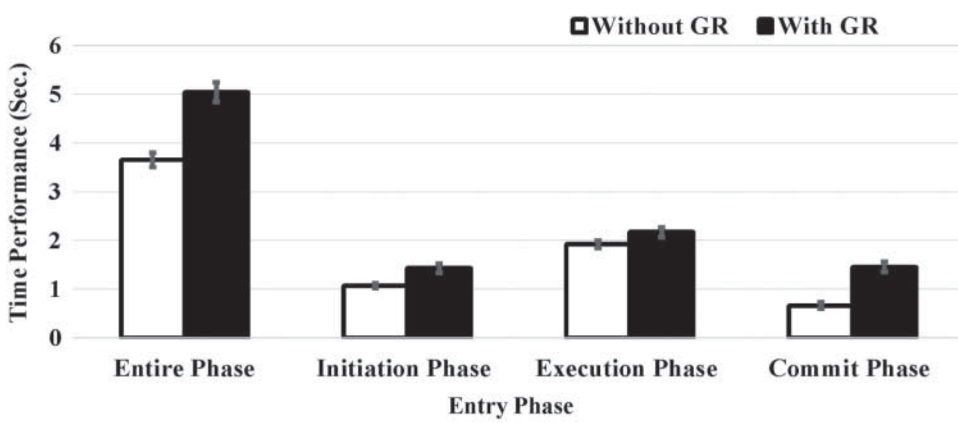

Figure 10. Time performance for number entry with or without GRs in different error types. Error bars represent 0.95 confidence interval.

GR interface led to a time of 1925 ms while *NoGR interface* had a time of 2167 ms.

Commit time was calculated as the duration from the release of the last digit key to the release of the ENTER key. There was a significant difference between *GR interface* and *NoGR interface* ($t_{30} = -6.29, P < 0.01$). The mean value was 667 ms for *NoGR interface* and 1458 ms for *GR interface*, revealing that participants dedicated more time to complete entry tasks with GRs.

6.7.3. *Subjective evaluation*

Fifteen of the sixteen participants who used *GR interface* reported that GRs were helpful for entry error detection. One participant said: 'the GR was really good and aided accuracy.' Another indicated, 'it was much easier to check accuracy using GRs.' Five of the sixteen participants also mentioned that it was slower for number entry with GRs but the time cost was acceptable for gaining higher accuracy. One participant said, 'I felt I was slower with GRs but more conscious of making an error.' Two participants mentioned that it was interesting to use *GR interface* because they had not experienced it before. Overall, the evaluation results were in good agreement with the results of data analysis and showed participants' positive feedback toward the use of GRs.

6.8. Discussion

We examined the performance of using GRs in number entry tasks. Compared with not using GRs, using GRs resulted in a much lower uncorrected error rate (decreased by 60%) at a price of a longer entry time (increased by 38%). For all error types, the use of GRs had a lower error rate. The subjective evaluation also indicates that using GRs was preferable to participants.

According to the time performance analysis, the use of GRs did not affect the execution phase, but had strong effects on the initiation and commit phases. As anticipated, viewing GRs cannot facilitate data entry, so participants did not tend to view GRs during entry process. It is also understandable that GRs affected the commit phase as participants were encouraged to look at GRs and NRs at the post-entry process. However, we did not expect GRs could influence the initiation phase, because during this phase it was not a necessary step to view GRs for number entry. One reason might be because participants were interested in GRs, therefore spending some time viewing them. It is of interest to investigate how GRs attract user attention at the initiation phase in the future.

The experiment results also indicate a need to promote the effects of visual checking on entry error detection. Although the participants using the *NoGR interface* were required to correct any entry errors they committed, 10 of them failed to notice their entry errors. It is understandable that people entering numbers are to be expected to make errors. However, given that entry errors may lead to adverse events in safety critical fields, it is important to design number entry systems which can help users detect errors.

7. GENERAL DISCUSSION AND FUTURE WORK

7.1. The effects of number entry with GR

Number entry is required in many fields including healthcare, finance and many day-to-day activities. Although advances in optical character recognition and other techniques make number entry increasingly rely on computers, manual data entry is still an indispensable task for people, such as transcribing data from paper into databases. This research proceeds on the premise that errors will occur and sometimes go unnoticed. Given human nature, errors are inevitable from time to time, which requires us to design effective techniques to detect and manage error.

While visual checking is a commonly used method for entry error detection, previous studies showed that number-based visual checking is not sufficient and there is scope to improve performance (Barchard and Pace, 2011; Barchard and Verenikina, 2011). In this study, we explored the use of number-based GRs to enhance the effects of visual checking. The most familiar rationale for more than one representation is to benefit from the redundant mental processes supported by different representations. We examined the performance of GRs through two empirical experiments. In Experiment 1, participants were asked to detect representation differences in terms of NRs and GRs, respectively. Results showed the two representations had similar performance on representation difference detection, indicating potential effects of using GRs on error identification.

We further measured how GRs affect entry error detection in Experiment 2, in which participants were required to perform number entry tasks with or without GRs. While resulting in 38% increase over input time, number entry with GRs resulted in 60% error reduction than that without GRs. An important aspect of entry error detection is getting users to dedicate more efforts in error identification. In the NR design, the confirmation step could be optimized by comparing the transcribed number to a number stored in short term memory when the instruction was read. In contrast with the GR design, the process of comparing the GRs must be carried by referring explicitly to both GRs. This accounts for the greater amount of time spent in the GR condition and possibly the higher rate of error detection.

It is worth emphasizing that the purpose of this study is not to entirely eliminate entry errors with our technique. We cannot attain this because 'to err is human' (Reason, 1990). We aim to provide a technique which can achieve a higher entry accuracy without much loss of entry speed.

7.2. Improving GR design

Although the current GR design has already achieved significantly improved effects of visual check on error detection, there are some directions that can be pursued to extend the current work.

Color information is an important dimension of GR. Using colors can enrich GR design and provide more features for users to detect representation differences. We did not employ this information to our GR design due to the consideration that instructions are not always shown in color and avoiding using color information can enable a broader use of our technique. Future work will explore how to involve color information in the design of number-based GRs.

In Experiment 2, we tested the effectiveness of GRs in a process where input-based GRs were rendered during number entry process. This process is similar to that of number entry without GRs, where input numbers were presented along with keystrokes. Hence, the basis of the comparison of the two tasks is reasonable in terms of experimental process. Future work will examine the effective of GRs in alternative experiment design. For example, the input-based GR pops up when participants press the ENTER key, so as to force users to compare GRs with identify errors. This may lead to varied error detection effects.

In current GR design, we only proposed certain numbers of shapes and compound line strings, which can represent numbers with limited length (e.g. numbers having 11 identical digits). For long numbers which cannot be represented with GRs under current design, we suggest using 'chunking' method to group digits (Nordby *et al.*, 2002) and constructing a GR for each digit group; a long number hence having a set of GRs. In addition, we will improve the design of shapes and line strings based on the research of human perception of geometric shapes and lines. Also, alternative GRs (e.g. different distributions of digits on the circle) will be implemented and tested against the proposal in this paper.

7.3. Applicability of the technique

The application of our technique to number entry system design needs to address how to present instruction-based GRs and entry-based GRs. A limitation of our technique is that it cannot be applied to number entry tasks involving hand-written originals, because instruction number-based GRs cannot be created in this case. Our technique is feasible for cases where GRs can be generated along with the input of instruction numbers. Taking programming infusion pumps on hospital wards as an example, instruction GRs can be generated based on instruction numbers and printed out along with prescription forms. The entry GRs can be rendered on the screen of the infusion pump if the device has the functionality of generating GRs. Currently, we are building an open source syringe pump platform (www.openpump.org) and will implement the GR design on this platform. The proposed technique is a cost-effective method and can be applied to a wide range of number entry systems.

8. CONCLUSIONS

Reduction of entry errors is important for many safety critical fields. Our study proposed the concept of number-based GRs to aid in entry error detection and demonstrated its effectiveness by empirical investigations. Our work offers a novel insight into number entry system design: in particular, we have shown that safety of numeric user interfaces can be improved. Perhaps there are other ways that need to be explored that provide better trade-offs, such as being faster or requiring less screen space. Nevertheless the obvious criticism of our approach that user time increases slightly must be seen in perspective—number entry is only part of a larger task, and the time increase for the entire task is marginal. And in almost all applications, an improvement in safety *gains* considerable time in avoided adverse incidents (and the loss in time for their subsequent management), which suggests the time/safety trade-off is more complex than at first appears and itself should be the subject of further investigation, particularly now we know it is possible to improve safety with novel techniques such as we proposed.

ACKNOWLEDGEMENTS

The main work was done during the first author was a postdoc researcher in Swansea University.

FUNDING

Funded as part of CHI + MED: Multidisciplinary Computer–Human Interaction research for design and safe use of interactive medical devices project EPSRC Grant Number EP/G059063/1.

REFERENCES

Barchard, K.A. and Pace, L.A. (2011) Preventing human error: the impact of data entry methods on data accuracy and statistical results. Comput. Hum.Behav., 27, 1834–1839.

Barchard, K. A. and Verenikina, Y. (2013) Improving data accuracy: Selecting the best data checking technique. Comput. Hum. Behav., 29, 1917–1922.

Cairns, P., Pandab, P. and Power, C. (2014) The Influence of Emotion on Number Entry Errors. In Proc. 32nd Annual ACM Conf. Human Factors in Computing Systems, CHI'14, New York, NY, USA, 2014, pp. 2293–2296. ACM.

US Food and Drug Administration (FDA). medsun: Newsletter #33, february 2009. http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/medsun/news/newsletter.cfm?news=33 (accessed June 22, 2015).

Gould, S.J.J., Cox, A.L. and Brumby, D.P. (2013) Using Graphical Representations to Support the Calculation of Infusion Parameters. In Kotze, P., Marsden, G., Lindgaard, G., Wesson, J. and Winckler, M. (eds), Human–Computer Interaction-INTERACT 2013, pp. 721–728. Lecture Notes in Computer Science, Vol. 8120. Springer, Berlin, Heidelberg.

Larkin, J.H. and Simon, H.A. (1987) Why a diagram is (sometimes) worth ten thousand words. Cogn. Sci., 11, 65–100.

Miller, G.A. (1956) The magical number seven, plus or minus two: some limits on our capacity for processing information. Psychol. Rev., 63, 81–97.

Nordby, K., Raanaas, R.K. and Magnussen, S. (2002) The expanding telephone number Part 1: Keying briefly presented multiple-digit numbers. Behav. Inf. Technol., 21, 27–38.

Norman, D.A. and Draper, S.W. (1986) User Centered System Design. New Perspectives on Human–Computer Interaction. L. Erlbaum Associates, Inc., Hillsdale, NJ.

Oladimeji, P., Thimbleby, H. and Cox, A. (2011) Number Entry Interfaces and their Effects on Error Detection. In Human–Computer Interaction-INTERACT 2011, pp. 178–185. Springer.

Oladimeji, P., Thimbleby, H. and Cox, A.L. (2013) A Performance Review of Number Entry Interfaces. In Human-Computer Interaction-INTERACT 2013, pp. 365–382. Springer.

Reason, J. (1990) Human Error. Cambridge University Press.

Sandnes, F.E. (2013) An Error Tolerant Memory Aid for Reduced Cognitive Load in Number Copying Tasks. In Stephanidis, C. and Antona, M. (eds), Proc. HCII2013, pp. 614–623. Lecture Notes on Computer Science, Vol. 8010.

Sandnes, F.E. and Huang, Y.P. (2013) A Computer Supported Memory Aid for Copying Prescription Parameters into Medical Equipment based on Linguistic Phrases. In Proc. IEEE SMC2013, pp. 3447–3451. IEEE Press.

Sears, A., Revis, D., Swatski, J., Crittenden, R. and Shneiderman, B. (1993) Investigating touchscreen typing: the effect of keyboard size on typing speed. Behav. Inf. Technol., 12, 17–22.

Soboczenski, F., Cairns, P. and Cox, A.L. (2013) Increasing Accuracy by Decreasing Presentation Quality in Transcription Tasks. In Human–Computer Interaction-INTERACT 2013, pp. 380–394. Springer.

Thimbleby, H. (2013) Reasons to Question Seven Segment Displays. In Proc. SIGCHI Conf. Human Factors in Computing Systems, CHI'13, New York, NY, USA, 2013, pp. 1431–1440, ACM.

Thimbleby, H. and Cairns, P. (2010) Reducing number entry errors: solving a widespread, serious problem. J. R. Soc. Interface, 7, 1429–1439.

Thimbleby, H. and Williams, D. (2013) Using Nomograms to Reduce Harm from Clinical Calculations. In IEEE International Conference on Healthcare Informatics (ICHI), pp. 461–470, IEEE.

Tu, H., Oladimeji, P., Li, Y., Thimbleby, H. and Vincent, C. (2014) The Effects of Number-Related Factors on Entry Performance. In Proc. 28th BCS Conf. Human–Computer Interaction, BCS-HCI'14, pp. 246–251.

Vicente, K. J., Kada-Bekhaled, K., Hillel, G., Cassano, A. and Orser, B.A. (2003) Programming errors contribute to death from patient-controlled analgesia: case report and estimate of probability. Can. J. Anesth., 50, 328–332.

Wiseman, S., Cairns, P. and Cox, A. (2011) A Taxonomy of Number Entry Error. In Proc. 25th BCS Conf. Human–Computer Interaction, BCS HCI'11, Swindon, UK, 2011, pp. 187–196. British Computer Society.

Wiseman, S., Cox, A.L. and Brumby, D.P. (2013) Designing devices with the task in mind which numbers are really used in hospitals? Hum. Factors, 55, 61–74.

Wiseman, S., Cox, A.L., Brumby, D.P., Gould, S.J. and O'Carroll, S. (2013) Using Checksums to Detect Number Entry Error. In Proc. SIGCHI Conf. Human Factors in Computing Systems, CHI'13, New York, NY, USA, 2013, pp. 2403–2406, ACM.

Zhang, J. and Norman, D.A. (1994) Representations in distributed cognitive tasks. Cogn. Sci., 18, 87–122.