# Vulture: A Mid-Air Word-Gesture Keyboard

**Anders Markussen** amark@diku.dk
**Mikkel Rønne Jakobsen** mikkelrj@diku.dk
**Kasper Hornbæk** kash@diku.dk
Department of Computer Science, University of Copenhagen
Njalsgade 128, Building 24, 5th floor, 2300 Copenhagen, Denmark

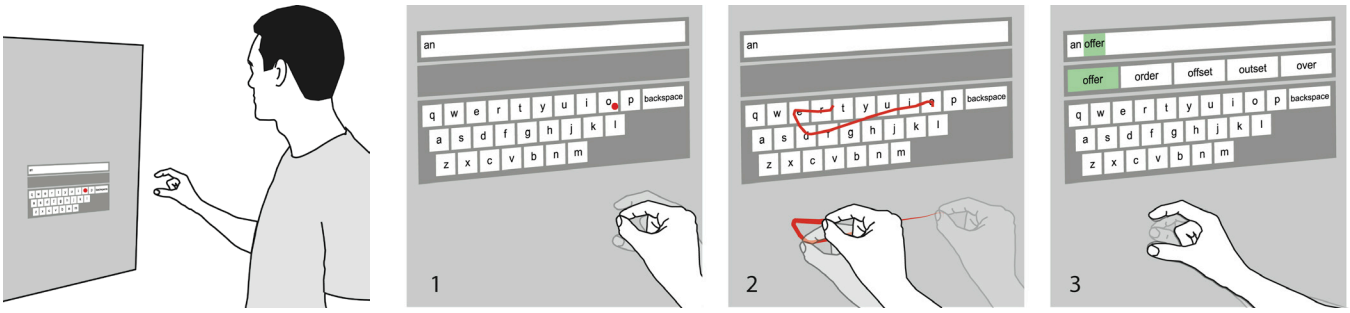


**Figure 1. Text entry using word-gestures in mid-air: By moving the hand, the user places the cursor over the first letter of the word and (1) makes a pinch gesture with thumb and index finger, (2) then traces the word in the air—the trace is shown on the screen. (3) Upon releasing the pinch, the five words that best match the gesture are proposed; the top match is pre-selected.**

## ABSTRACT
Word-gesture keyboards enable fast text entry by letting users draw the shape of a word on the input surface. Such keyboards have been used extensively for touch devices, but not in mid-air, even though their fluent gestural input seems well suited for this modality. We present Vulture, a word-gesture keyboard for mid-air operation. Vulture adapts touch based word-gesture algorithms to work in mid-air, projects users' movement onto the display, and uses pinch as a word delimiter. A first 10-session study suggests text-entry rates of 20.6 Words Per Minute (WPM) and finds hand-movement speed to be the primary predictor of WPM. A second study shows that with training on a few phrases, participants do 28.1 WPM, 59% of the text-entry rate of direct touch input. Participants' recall of trained gestures in mid-air was low, suggesting that visual feedback is important but also limits performance. Based on data from the studies, we discuss improvements to Vulture and some alternative designs for mid-air text entry.

## Author Keywords
Word-gesture keyboard; shape writing; text entry; mid-air interaction; in-air interaction; freehand interaction.


*CHI 2014,* April 26 - May 01 2014, Toronto, ON, Canada

http://dx.doi.org/10.1145/2556288.2556964

## ACM Classification Keywords
I.3.6 Methodology and Techniques: Interaction techniques

## INTRODUCTION
Mid-air interaction is an emerging input modality for large displays [20], mobile phones [9], augmented reality [23], and desktop computers [29]. Facilitated by improved tracking equipment, mid-air techniques cover many types of interaction. For instance, mid-air pointing enables selection and manipulation of objects (e.g., [2, 6, 10, 28, 30]). Writing text in mid-air, however, has received less attention. Text entry is an important activity and supporting it in mid-air would be beneficial for a number of scenarios such as work in sterile conditions (e.g., operating theatres), in augmented reality (e.g., with Google Glass), and when writing on public displays. It has been shown that people can write in mid-air with devices such as game controllers and dedicated gloves [21], but also using their hands [12, 18]. However, text-entry rates for mid-air interaction are low, around 13 [18] to 18.9 WPM [27]; the latter rate was obtained with tactile feedback on errors and no character production on errors. Furthermore, most techniques support only single-character text entry (e.g., [18, 21, 27]). So mid-air text entry is still relatively slow and better techniques should be developed to make mid-air text entry practical. Hence, we study if speed can be improved by moving from selection-based text entry to gestural input.

Word-gesture keyboards (WGK) (e.g., SlideIT, Swype and ShapeWriter) have gained popularity (e.g., a WGK is now shipped as standard on Android devices) and perform well on touch screens [11, 36, 37]. The key idea of WGKs is that the user enters a word by drawing the pattern formed by its letters on the input surface rather than by typing the letters. When implemented with the QWERTY layout,

WGKs allow users to benefit from previous experience. Furthermore, WGKs provide a fluent way of writing words as gestures while also supporting simple tapping input.

The present paper suggests that WGKs may be beneficial to mid-air text entry. However, transferring WGK to mid-air is hard: what are the delimiters of words, what is the equivalent of tapping and releasing in mid-air, and will the prediction algorithms for WGKs, developed for direct surface input [13], work in mid-air?

In the rest of the paper we describe a system, Vulture, for doing mid-air text entry that answers some of these questions. We also describe two formative studies of Vulture: one study estimates the text-entry rate of Vulture and another study compares the performance and recall of gestures with Vulture to a touch-based WGK.

## RELATED WORK

To our knowledge, the potential of WGKs in mid-air have not before been evaluated, but the literature offers relevant research on in-air gestures [31], freehand gestures [30], and mid-air interaction [20, 27]. This walkthrough of related work focuses on mid-air pointing, mid-air text entry, and WGKs, as these directly relate to the focus of this paper.

### Mid-air Pointing

Much research on mid-air interaction concerns pointing [2, 6, 10, 28, 30]. Mid-air pointing is generally done through variations of ray casting that differ in speed, error rate, and fatigue. Generally, ray-casting techniques are fast for coarse movements, but provide limited precision due to for instance hand tremor. Furthermore, ray-casting techniques are generally distance-dependent [27], meaning that precision degrades as the user moves away from the display. One way to minimize distance-dependence is to project movement orthogonally onto the display as done by Markussen et al. [18]. Orthogonal projection limits the user's reach, but maintain a constant control-display ratio across distances.

### Mid-air text entry

Earlier work on text entry in mid-air takes two main forms. In *selection-based techniques*, users make series of movements and selections to produce individual characters. Shoemaker et al. [27] used a Nintendo Wiimote for mid-air interaction, and proposed three selection-based text-entry methods using a 3D cube layout of letters, a circular layout, and a regular QWERTY layout. The QWERTY-based keyboard was preferred and fastest with 18.9 WPM. Contrary to more recent text-entry studies, Shoemaker et al. used a restricted text-entry setup that provided tactile feedback on errors and no character production on errors. Markussen et al. [18] adapted three text-entry methods to selection-based mid-air text entry. A mean text-entry rate of 13.2 WPM was observed for a QWERTY-based text-entry method that used orthogonal projection of the hand position onto the display.

In *gesture-based techniques*, users produce gestures that are interpreted as letters. Ni et al. [21] created AirStroke, a Graffiti based text-entry method that was evaluated with word completion (11.0 WPM) and without (6.5 WPM). Kristensson et al. [12] showed that continuous recognition of gestures within a defined input zone in front of the user is possible using the Graffiti alphabet [5]. The system was implemented using a Kinect. The focus of the paper was on gesture recognition rather than on text-entry rate, and the paper reported no performance measure relating to speed.

Two alternatives to these approaches have been researched. Mid-air handwriting recognition has been explored using various sensors and camera-based recognition systems (e.g. [1, 25, 26]). Experiments have primarily focused on recognition quality, with recognition rates of up to 97%, rather than text-entry rate. However, regular hand-writing is limited to approximately 15 WPM [7]. We expect mid-air handwriting to be subject to similar limitations. Another alternative for mid-air text entry is sign language for which conversational speeds of 175 – 225 WPM have been reported [19]. However, sign language recognition is challenging and learning sign language requires more training than most users are willing to invest.

### Word-Gesture Keyboards (WGKs)

WGKs, originally referred to as shape writing keyboards, were introduced by Zhai and Kristensson [11, 13, 35, 36, 37]. Instead of typing a word, the user draws its shape (the line connecting the letters in the word) on top of a visual representation of the keyboard. The method was designed for stylus input and has later proven useful for touch-based text entry. On mobile devices, WGKs have gained enough success to become a standard part of many soft keyboards. Studies have shown that novice users of WGKs write 25 WPM after 35 minutes of practice and 46.5 WPM for single well-practiced phrases [11].

An important part of implementing a WGK is to develop efficient and effective shape recognition algorithms. Previous studies of WGKs give few details on the algorithms behind the recognition of word gestures, possibly due to the commercialization of WGKs for mobile phones. To our knowledge, SHARK$^2$ [11, 13] provides the most detailed descriptions of a WGK implementation.

SHARK$^2$ bases its recognition on two recognition channels: A shape channel and a location channel. The channels estimate the probability of a given shape being a word from the vocabulary. The shape channel normalizes the drawn shape to a specific location and size, and estimates how well it matches the shape of each word in the vocabulary. The normalization has the disadvantage that a relatively large number of ambiguous shapes occur in a normal English vocabulary. The location channel helps minimize these ambiguities by distinguishing similar shapes by their different locations. Ultimately, the probabilities from the two channels are integrated into one probability measure.

We are unaware of work trying to adapt WGKs to mid-air; the next section outlines the main challenges in doing so.

## VULTURE: A MID-AIR WORD-GESTURE KEYBOARD

Below we describe how interaction with Vulture works, the design choices made, and how the word-gesture recognition in Vulture is implemented.

### Writing with Vulture

The operation of Vulture is illustrated in Figure 1. The system works by tracking the user's hand and fingers. To enable input using both elbow and wrist movement, the hand's position, orientation, and an initial calibration of the user's pinch is used to estimate the position where pinches are expected to occur. This position controls a cursor, represented as a dot. The interaction was originally based on orthogonal projection, but a control-display (CD) ratio was applied to support interactions from a distance while maintaining a readable user interface. The user writes a word by placing the cursor in the first letter of the word, making a pinch gesture (with the index finger and the thumb), then tracing the letters of the word, and finally releasing the pinch. While the user is not writing, the cursor is red. Upon pinching the cursor turns green and the cursor's movement is traced over the keyboard. After completing a word-gesture, the five words that best match the gesture are shown in the list of suggestions; more than five was expected to provide little improvement [14]. The user can continue writing and implicitly confirm that the highlighted word was a *match* or the user can *select* a word from the list of suggestions. Also, the user can *undo* the suggested word in the text input field by selecting backspace or *delete* previously typed words by multiple selections of backspace. The four basic interactions—*match, select, undo,* and *delete*—will be used extensively in later parts of the paper.

### Designing a Mid-Air WGK

Previous WGKs have been designed for use on touch- or stylus-enabled surfaces. Bringing WGKs away from the surface and into mid-air implies three major considerations.

#### *Separating words*

Surface-based WGKs benefit from an implicit delimiter, in that a gesture begins when the finger or stylus is put on the surface, and ends when it is lifted away from the surface. This implicit delimiter is not available in mid-air. We consider four options: (1) Implement a WGK that requires no delimiter, requiring auto segmentation of input in order to identify words similar to what has been done for unistroke gestures [12]; (2) incorporate delimiters into the shape that is written (e.g., pig-tails [33] or crossing a certain part of the keyboard to end a word); (3) use some of the extra degrees of freedom available in mid-air (e.g., depth) to indicate word separation; or (4) use recognizable hand gestures that can serve as word delimiters (e.g., pinching while gesturing).

All four options could potentially provide good results, but the first two require recognition algorithms significantly different from existing WGK implementations; they could be topics for future research if WGKs shows to be useful in mid-air. We tested (3) and (4) with users and found the last option to be liked the most. We therefore settled on pinching as a word delimiter in Vulture.

#### *Separation of motor space and display space*

One advantage of surface-based WGKs is the direct coupling of input and output space, which gives users direct feedback on interaction as it occurs. A mid-air WGK offers no such direct coupling. Designing the mapping of input and output space for mid-air interaction opens up a complex design space. We prototyped different mappings of the input and output space (e.g., providing a slanted or horizontal input plane to minimize user fatigue). However, users found it hard to adapt to these mappings. To minimize effects of separation, we therefore chose a planar input space parallel to the display.

#### *Size of motor space*

The size of the motor space that users are interacting within can potentially impact performance. However, a priori prediction of the optimal size is hard. Even with advanced biomechanical simulations, the interplay between recognition algorithms and biomechanics is highly complex. To add further complexity, the optimal size is expected to vary among users and to evolve over time. We informally tested different sizes of motor space to identify a size that allowed for fast movements while maintaining the user's ability to draw shapes of sufficient quality. We ended up mapping the boundaries of the keyboard to a rectangle of approximately 20 × 5.5 cm.

### Word-Gesture Recognition

Vulture is based on the same integration of shape and location channels as SHARK$^2$. Below, we describe the specific implementation of Vulture.

Initially, each word in the dictionary is processed into a template for comparison. The optimal shape, calculated as the line connecting the key center of each of the letters in the word, is resampled to a fixed number of equidistant points. We found 50 samples to perform well. Furthermore, a normalized shape is calculated based on the resampled shape and the shape channel calculation described below.

Before processing a drawn shape, the shape is resampled to the same number of equidistant points as the dictionary templates. To minimize the number of ambiguities and to reduce processing time, the vocabulary is pruned by considering only words with an optimal starting position less than 1 key-width and ending position less than 2 key-widths away from the evaluated shape's start and end point.

We iteratively developed and refined the implementation of the shape and location channels based on user feedback

from informal evaluations. Below we describe the implementation found to perform the best.

*Shape channel*

The resampled shape is translated to the origin and normalized to make it location and size invariant. Normalization is done by scaling the shape to a square in the same way as in the $1 recognizer [34].

The distance function that produces the output of the shape channel is identical to the one used in SHARK$^2$: The output of the shape channel is the average spatial distance between corresponding equidistant points in the compared shapes.

*Location channel*

The location channel score $x_L$ is calculated as follows:

$$x_L = \sum_{i=1}^{N} \alpha(i)\delta(i)$$
$$\delta = \max(\|u_i - t_i\| - r, 0)$$

where $\alpha(i), i \in (1,N), \sum_{i=1}^{N} \alpha(i) = 1$ are the sample weights, $u$ is the unknown shape that are being compared to the template word $t$, and $r$ is a radius that is used to reward individual samples.

The location channel has two primary features. First, each sample index has an associated alpha weight. This is used to modify the weight of the individual samples so that some samples can receive higher weight. Second, the pairwise distances between corresponding samples are calculated. If the sample distance is less than a half key-width ( $r$ ), the sample distance is set to zero. This has the effect that slightly more words are considered good candidates by the location channel, and increases the shape channel's discriminatory power in cases of location similarities. The sample distances are weighed using the alpha weights and summed to provide the location channel score.

Participants in a pilot study felt more precise in the starting location of the word than in the rest of the word. We collected data showing that the first sample of the trace fell within the desired key 95 % of the time, whereas the last sample of the trace was only within the desired key 80 % of the time. Hence, we decided to assign extra weight to the first sample using the following weighing function:

$$\alpha(i) = \begin{cases} 0.1, i = 1 \\ \frac{0.9}{N-1}, otherwise \end{cases}$$

*Single key words*

Due to the scaling function of the shape channel, single key input becomes a challenge. Very short shapes drawn in an attempt to produce single characters are rescaled and the shape channel promotes longer words with similar shapes.

To avoid this behavior, shapes of lengths less than 0.4 key width are truncated to contain only the starting point before resampling, effectively cancelling the unwanted effect of the shape channel scaling.

## STUDY 1: THE POTENTIAL OF A WGK IN MID-AIR

This study was designed to provide us with: (1) estimates of the potential text-entry performance of mid-air WGKs, both users' initial performance and improvement over time, (2) text-entry data for tuning the parameters of Vulture, and (3) data for analyzing the errors that users make.

### Study Design

Participants completed 10 sessions of text entry using Vulture. During each session, participants transcribed 4 blocks of 10 phrases sampled randomly (no phrases occurred twice in a session) from the MacKenzie and Soukoreff corpus [16]. This resulted in a total of 6 participants × 10 sessions × 4 blocks × 10 phrases = 2400 transcribed phrases.

### Study Setup

The study was conducted on a 2.8 × 1.2 m large high-resolution display, one use case of mid-air text entry, with 7680 × 3240 pixels (see Figure 2). We used an OptiTrack motion-capture system, providing 100 frames per second of tracking information. Although expensive, the system ensures reliable results and minimizes the effects of tracking noise. Furthermore, as low-cost tracking systems are currently improving rapidly, using the OptiTrack system could also improve comparability with future research. To minimize the effects of participants' hand tremor, we smoothed the input using the 1€ filter [4].

Participants stood 1.5 m from the display while using Vulture. To make the keyboard easy to read, the window containing the keyboard was 40 × 11 cm. The difference in keyboard size in motor space and display space results in a CD ratio of 1:2.

For the study, Vulture matched shapes against a dictionary of 10,246 words. This number was based on the assumption that an average English speaker has a vocabulary of approximately 10,000 words [22]; it is also similar to that of the initial design criteria for SHARK$^2$. The dictionary was constructed from the British National Corpus (ftp://ftp.itri.bton.ac.uk/) by taking words with a frequency greater than or equal to 600 and that contain only alphabetic characters. To support word-gesture based transcription of

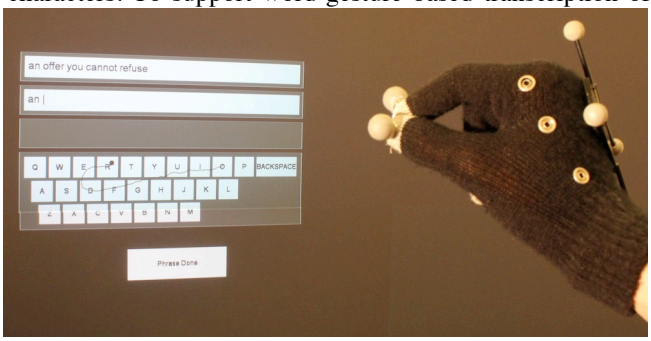


**Figure 2: The study used a large high-resolution display and users wore a glove with reflective markers in order to track the position of their hand and fingers.**

the MacKenzie–Soukoreff corpus [16], words missing from the corpus were added to the dictionary.

To enable analysis, all tracking data were logged along with details on text entry including the four basic interactions described earlier, the word gestures produced, the presented and transcribed phrases, and timing data.

### Participants

We recruited 6 paid volunteers (1 male), 5 right-handed, Participant ages ranged from 21 to 53 ($M$ = 29.3). Participants' previous experiences with WGKs were between 1 and 3 ($M$ = 1.3) on a seven-point scale from "1 - No experience" to "7 - Expert". Their written English skills were between 3 and 6 ($M$ = 4.5) on a seven-point scale from "1 - No English skills" to "7 - Native".

### Procedure

Before text entry began, the system was calibrated. During calibration, participants defined the hand position that mapped to the center of the keyboard by placing their elbows to their sides and the writing arm bent to 90 degrees. Hand movement relative to this posture was mapped to cursor movement on the keyboard. This posture was preferred by pilot participants.

In each of the 10 sessions, participants were allowed some time to (re)-familiarize themselves with the operation of Vulture before starting to write. On average, participants wrote 1.8 phrases at the beginning of each session.

Participants were instructed to write the phrases "as quickly and accurately as possible - as if typing e-mail to a colleague". Between each block of 10 phrases, users were allowed a break of up to three minutes. To complete a phrase, participants clicked a "Phrase done" button shown below the keyboard. Before continuing to the next phrase, participants were shown a window with their mean text-entry rate and the rate of the previous phrase.

After each session, participants were interviewed about their impression of Vulture, the recognition quality, and any technical issues they had experienced. On average, each session lasted approximately 30 minutes.

### Dependent Measures

On the phrase level, we analyze text-entry rate and error rates; on the word-gesture level we analyze correctness and distance measures.

*Text-entry rate* is measured in Words Per Minute (WPM), as in [3], with this formula

$$WPM = \frac{|T|}{S} \times 60 \times \frac{1}{5}$$

where $|T|$ is the length of the transcribed string, S is time in seconds. We use $|T|$ instead of $|T-1|$ since the time to produce the first character is included when timing word gestures [17].

*Error rate* is based on Minimum Word Distance (MWD). MWD is calculated in same way as the Minimum String Distance (MSD) [15], but on a per-word level rather than on a per-character level. Hence, MWD describes the minimum number of word-substitutions, -insertions, and -deletions needed to make strings identical. MWD error rate is calculated in the same way as MSD error rate, but on a per-word level:

$$MWD\ error\ rate = \frac{MWD(P,T)}{\overline{S_P}} \times 100\%$$

where $P$ and $T$ are the sets of words in the presented and transcribed strings, and $\overline{S_P}$ is the mean size of the optimal alignments calculated on a per-word level rather than a per-character level. MWD error rates will generally be higher than MSD error rates, because whole words are classified as wrong if they contain one or more erroneous characters.

*Correctness* of a single word gesture was determined by comparing it to the users' intended word. As long as the current words in the transcribed string match the beginning of the presented string, we consider the next word in the presented string to be intended. When the strings do not match (e.g., due to errors earlier in the string), we classify the users' intention manually based on a visualization of the drawn shape on top of a keyboard, the currently transcribed words, and the presented string: We did this manually both because deletion makes this classification non-trivial and so as to learn from the errors users made.

*Distance measures* are reported as centimeters of movement in the input plane.

### Results

#### *Text-Entry Rate*

Figure 3 shows the mean text-entry rate per participant over sessions. Participants reached a mean text-entry rate of 20.6 WPM ($SD$ = 7.3) in the last session, which is 75% faster than the first session ($M$ = 11.8 WPM). The development of text-entry rates in Figure 3 indicates that participants could improve further with more practice. As expected, a repeated measures analysis of variance (RM-ANOVA) on WPM with session as factor revealed a main effect, $F(9, 45) = 17.593$, $p < .001$. We also found participants' movement speed to consistently increase over sessions (70% from first session to last session) as users were familiarized with Vulture. An RM-ANOVA on movement speed with session as factor showed a main effect, $F(9, 45) = 12.916$, $p < .001$, providing a potential explanation of the increased performance over time.

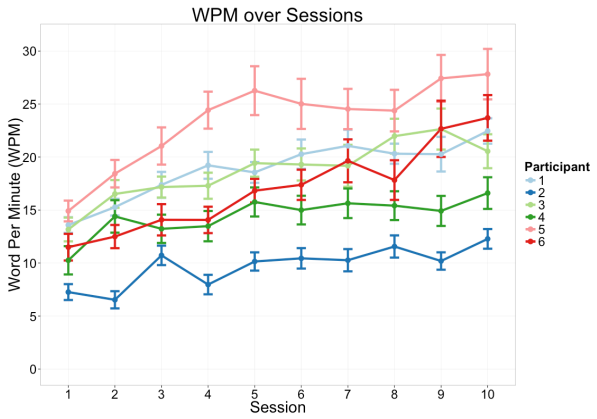


**Figure 3: Text-entry rates in Words Per Minute (WPM) over sessions with error bars showing 95% confidence intervals.**

Figure 3 also suggests large performance differences among participants. This is related to differences in movement speed: the mean movement speed of the fastest participant was 155% higher than that of the slowest.

### *Errors*

The mean MWD error rate was 4% (*SD* = 9.3). An RM-ANOVA with session as factor did not reveal a main effect, $F(9, 45) = .699$, $p = .707$. Of the 2400 transcribed phrases, 1941 phrases (80.9%) contained no transcription errors and for 1330 (55%) of those no corrections were made. Surprisingly, the highest MWD error rate (*M* = 8%) was found for the participant with the lowest text-entry rate, indicating that acceptance of errors in the transcribed string may not be the primary factor for higher text-entry rates.

### *Interaction with Vulture*

Next, we look in more detail on how users interacted with Vulture. Table 1 shows an overview of users' actions. The table reports median durations and distances so as to reduce the influence of outliers. The most frequent action was *match*, writing a word gesture and implicitly confirming the default suggestion (73%). In 9.7% of the word-affecting actions the user had to *select* another word than the default. Performing a *select* is generally done either because of word ambiguities (words with the same optimal shape) or due to the shape being close, but not close enough to make the intended word a *match*. In 8.1% of the actions, users had to *undo* the gesture they had produced, presumably because the intended word was not among the alternatives. In 9% of the actions, users had to *delete* a previously confirmed word to fix errors earlier in the transcribed string.

Table 1 also shows the correctness of word production compared to the users' intended word. It is seen that 3,070 of the 16,356 (19%) produced word-gestures are either *undone* or *deleted*, suggesting that a relatively large amount of time is spent correcting errors.

For the *match* class, 18% of the words were incorrect, whereas for the *select* class only 3% were incorrect. For the matches, participants seem to have produced some unintentional word-gestures, resulting in unintentionally confirmed words. This typically occurred when participants did only small finger movements when pinching. We identified 966 of the 2,022 incorrect gestures with a duration of less than 100ms (48%) that we deemed as unintentional.

For correct words, the gesture time is larger for *matched* than for *selected* words. This can in part be explained by increased precision in participants' gestures as expressed by the mean sample distance and gesture start/end distance in Table 1.

The low duration and length of word gestures that are *undone* have two potential explanations: (1) some of the very short word-gestures caused by the pinch implementation will have to be undone; (2) participants were often observed to realize an error and abandon their word-gesture early, possibly resulting in a lower median gesture duration and length than other classes of action.

The median accuracy measures for correct words are generally relatively precise taking into account the actual motor space key-width of 1.8 cm. We see this as an indication that participants continue to be visually bound rather than starting to rely on recalled shapes as suggested by Kristensson and Zhai [13]. The study design does not allow us to conclude if participants are visually bound, but the indication is supported by our own experiences from the development process, where we continually felt bound by visual feedback rather than typing based on gesture recall.

| Class (% of total) | Correct | N | Gesture Duration (s) | Selection- / Backspace-Click Duration (s) | Mean Sample Distance (cm) | Gesture Start Distance (cm) | Gesture End Distance (cm) | Gesture Length (cm) | Gesture Speed (cm/s) |
|---|---|---|---|---|---|---|---|---|---|
| Match (73.2%) | Yes | 11130 | 2.28 | | 0.54 | 0.36 | 0.36 | 18.93 | 8.13 |
| | No | 2022 | 0.135 | | 3.32 | 0.78 | 3.40 | 0.00 | 0.00 |
| Select (9.7%) | Yes | 1689 | 1.9 | 1.82 | 0.77 | 0.37 | 0.41 | 16.94 | 8.99 |
| | No | 57 | 1.63 | 1.9 | 1.44 | 0.43 | 0.77 | 19.20 | 9.20 |
| Undo (8.1%) | | 1458 | 1.28 | 1.86 | 2.84 | 0.71 | 1.89 | 8.91 | 6.60 |
| Delete (9.0%) | | 1612 | | 0.18 | | | | | |

**Table 1: The word-affecting actions done with Vulture (*N* = 17,968). Mean Sample Distance is the mean distance between the individual samples of the resampled input shape and the resampled optimal shape of the intended word. Gesture Start/End Distance is the distance from the start/end point of the input gesture to the start/end point of the shape of the intended word.**

### Improving Vulture

We did two improvements to Vulture based on the results. First, we modified the pinch implementation so as to reduce the number of unintended word-gestures.

Second, the interaction data provided by the study allowed us to adjust the parameters of the recognition algorithm. Optimization of a WGK based on SHARK$^2$ is a non-trivial task due to the interactions between the different channels [11]. Hence, the optimizations did not aim to find a globally optimal set of parameters. Instead they were done by running the recorded word-gestures through permutations of parameters, aiming to bring as many word-gestures as possible into the *match* and *select* categories. We ended up using a set of parameters that resulted in approximately 5% more of the total gestures being *matched*.

## STUDY 2: TOUCH AND GESTURE RECALL

The results from Study 1 show that WGKs can improve text entry in mid-air. However, there is a discrepancy compared to the text-entry rates of 25 WPMs that have been reported for touch-based WGKs after just 40 minutes of practice [36]. In order to directly compare the performance of WGKs in mid-air and touch, we conducted a second study. In particular, we were interested in understanding how the separation of motor space and display space impacts performance. Also, while we saw an increase in performance over time in Study 1, we were unable to tell whether users become faster because they learn the shapes of familiar words and shift from visually-guided tracing of words [36]. Thus, we examine how well users recall word gestures.

### Study Setup

For the mid-air interaction, the same apparatus was used as in Study 1. Touch on the large display was detected with diffused surface illumination. Input from six cameras was analyzed using Community Core Vision and multiplexed to form input for tracking touch points.

The touch WGK used the same algorithm and parameters as the mid-air WGK. However, the touch keyboard was scaled and relocated to the size of the motor space of the mid-air keyboard (20 x 5.5 cm), making it comparable to the size of the keyboard on a landscape oriented 10.1-inch tablet.

### Study Design

The study used a between-subjects design with input modality as the independent variable: One group of participants used Vulture, the other group used the same WGK implementation with touch interaction. A between-subjects design was used to avoid knowledge transfer and other interference between touch and mid-air conditions.

Each session consisted of two parts. First, participants repeatedly transcribed three phrases from the MacKenzie and Soukoreff corpus [16]. To select the phrases for transcription, the mean WPM for each phrase was calculated for the last session of study 1. The three phrases selected were those with the WPM closest to the mean WPM of that session ("destruction of the rain forest", "an offer you cannot refuse", "dolphins leap high out of the water").

The second part tested the participants' ability to recall word-gestures. Participants had to produce each of the 15 unique word gestures that comprised the transcribed phrases. They had no visible keyboard on the display, and no cursor. A blank rectangle marked the previous location of the keyboard.

### Participants

We recruited 12 paid volunteers (5 male), all right-handed. None of them had participated in Study 1. The participants were randomly assigned each of the two conditions (six participants each). Participant ages ranged from 18 to 31 years ($M$ = 24.4). Participants' previous experiences with WGKs were rated between 1 and 4 ($M$ = 1.8) and their written English skills were between 5 and 7 ($M$ = 5.6); scales as in Study 1.

### Procedure

Before starting the actual text entry, participants were allowed to familiarize themselves with the operation of the keyboard. We encouraged participants to experiment with doing fast movements during training.

After training, participants transcribed 9 blocks of 6 phrases (2 repetitions of each phrase per block in randomized order), resulting in a total of 648 transcribed phrases. Participants were allowed to rest between blocks. Text entry is completed as unrestricted text entry, allowing users to delete and correct previously entered text. Participants were instructed to perform the text entry tasks as in Study 1.

After transcribing all the phrases, participants were asked to produce each of the 15 unique words in the phrases. For this part, only the word that the users were to produce and the keyboard frame (that is, without any keys) was shown. Participants received no feedback on touch or mid-air tracking. Participants had to produce gestures based only on their memory of the word gestures as well as the location and size of the input space. Participants were asked to produce each word twice. The words were shown in randomized order.

### Results

#### *Text-Entry Rate*

Figure 4 shows text-entry rates for the two methods over the blocks of the study. An RM-ANOVA with method as between-subjects factor found touch to be significantly faster than mid-air, $F(1, 10) = 10.681$, $p < .05$. On the last block, touch text entry ($M$ = 47.5 WPM) was 69% faster than mid-air text entry ($M$ = 28.1 WPM). Compared to the first study, the initial text-entry rates are much higher. This suggests either that the optimizations of the recognition

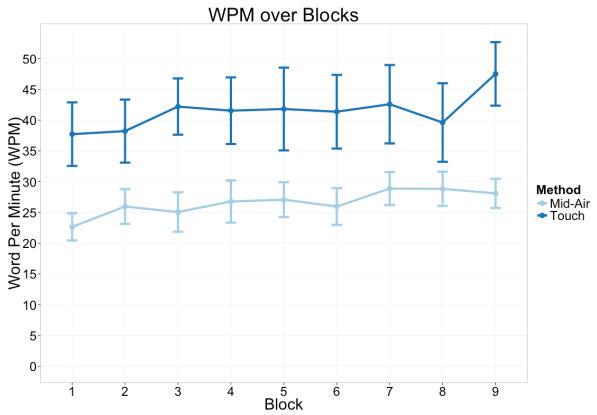


**Figure 4: Text entry rates in words per minute (WPM) over blocks with error bars showing 95% confidence intervals.**

algorithms have had an effect or that the instruction to try fast movements during training boosted the initial performance. The reason is most likely a combination of both. Touch is 26% faster on the last block than on the first. Mid-air is 24% faster.

### *Errors*

An RM-ANOVA with method as between-subjects factor found that the MWD error rate for touch ($M$ = 3.7%, $SD$ = 8.1) and mid-air ($M$ = 1.7%, $SD$ = 5.7) was not significantly different, $F(1, 10) = 2.648$, $p = .135$.

### *Interaction With Vulture*

Table 2 categorizes participants' behavior into the same 4 classes as in Study 1 and shows that touch and mid-air have similar behavior compared to Study 1, but with a slightly higher percentage of *matched* words for mid-air. It is worth noting that the improved *match* percentage is similar to the ~5% effect of the WGK optimization.

Study 1 found gesture speed to be the most important factor affecting text-entry rate. Touch gesture speed ($M$ = 21.44, $SD$ = 13.83) in study 2 is 74% faster than mid-air ($M$ = 12.36, $SD$ = 4.89). Comparing this to the 69% difference in text-entry rate supports that gesture speed is a good indicator for text-entry rate.

### *Recall*

We analyzed participants' recall with regards to the location, size, shape of gestures, and gesture speed. As expected, participants were more accurate in recalling the location and size of gestures with touch input than in mid-air because they could point directly to the keyboard frame on the display. This can be seen from Table 3, which shows data for word gestures produced in the text entry part of the experiment (the first two rows) compared to participants' recall of the word gestures (the last two rows). We note two important findings: (a) Gesture start distance averaged about 1 key-width for touch compared to more than 5 key-widths for mid-air; and (b) participants made twice as long recall gestures as they did in the text-entry part for mid-air input, whereas recall gestures and text-entry gestures were comparable in length for touch input.

| Method | *Match* | *Select* | *Undo* | *Delete* |
|---|---|---|---|---|
| Mid-air | 78.4% | 8.6% | 7.9% | 5.1% |
| Touch | 72.7% | 9.4% | 10.2% | 7.7% |

**Table 2: Distribution of Study 2 interaction classes across mid-air and touch interaction.**

In order to assess participants' ability to recall the shape of word-gestures, we ran the recalled gestures and the text-entry gestures through Vulture's shape channel (note that we used only the shape-channel in order to normalize for location offsets). Table 3 also shows how the intended word would be ranked if only the shape channel was used (the very high rankings indicate that the shape channel alone is very poor). The ranks for the recalled gestures are more than double that of the text-entry gestures, indicating that the participants' ability to recall the shapes of gestures was low. Moreover, participants seem to recall gestures worse for mid-air (rank 808.7) compared to touch (rank 696.3). As for participants' ability to redo shapes from memory, several participants stated that they recalled gestures by imagining the keyboard layout in front of them, thus making them bound by an imagined keyboard rather than relying on remembered movement patterns.

Touch and mid-air show very similar movement speed in the recall conditions, whereas participants' movements are much slower for mid-air during text entry. This suggests that visibility of the keyboard slows participants only for mid-air input.

## DISCUSSION

We have suggested that WGKs may be beneficial to mid-air text entry. The issues in designing Vulture, a WGK that works in mid-air, have been described and two studies aiming to characterize its performance have been run. Next we try to answer some remaining questions about WGKs.

### Do WGKs in mid-air work?

Study 1 showed text entry rates of 20 WPM; Study 2 showed 28 WPM using a small phrase set. Earlier studies of mid-air text entry have found 13.2 [18] to 18.9 WPM [27]. Thus, the studies indicate that Vulture provide text-entry rates that surpass earlier work.

Study 2 showed that Vulture worked comparably to touch in some aspects: an equal number of errors were made, the types of actions made were similar, and the gestures produced were comparable. However, participants were much slower at making gestures in mid-air than for touch; text-entry rates, therefore, were also about 60% lower. Due to the increased complexity of performing mid-air text entry compared to touch-based text entry, we did not expect mid-air text entry to compete with touch input. This hypothesis was confirmed by our studies.

Zhai and Kristensson theorize that the use of WGKs "automatically shifts from the ease end (visual tracing) to

| Method | N | Shape channel rank (lower is better) | Gesture Start Distance (cm) | Gesture Length (cm) | Gesture Speed (cm/s) |
|---|---|---|---|---|---|
| Mid-air (feedback) | 2119 | 315.9 (*SD*=1385.6) | 0.80 (*SD*=1.75) | 20.20 (*SD*=12.70) | 12.36 (*SD*=4.89) |
| Touch (feedback) | 2265 | 333.0 (*SD*=1385.0) | 0.83 (*SD*=1.61) | 18.68 (*SD*=11.81) | 21.44 (*SD*=13.83) |
| Mid-air (recall) | 180 | 808.7 (*SD*=1735.1) | 9.79 (*SD*=5.36) | 41.56 (*SD*=21.06) | 23.70 (*SD*=12.31) |
| Touch (recall) | 180 | 696.3 (*SD*=1582.7) | 1.75 (*SD*=1.44) | 20.43 (*SD*=9.22) | 21.21 (*SD*=9.29) |

**Table 3: Word-gesture statistic for text entry and recall. Shape channel rank is the location of the intended word in a list of dictionary words sorted ascending by the shape channel score of the dictionary words.**

the efficient end (recall gesturing)". Participants in the present studies did not seem to shift from visual tracing. In contrast, they seemed to depend on visual feedback even after substantial training (Study 1) and after practicing the same words repeatedly (Study 2).

### Why are mid-air WGKs slower?

Figuring out why Vulture was slower than touch seems to be key to improving it. We see three potential explanations. First, the input plane and the display plane are decoupled in mid-air text entry. Users must therefore mentally couple their gestures in motor space to the keyboard and feedback on the display; the principle of stimulus-response compatibility [24] suggests that difficult.

Second, users of Vulture seem to rely heavily on visual feedback; this has been shown problematic in other studies of text entry [32]. Data suggest that mid-air users have slower movement speed than touch users, but that they start and end equally precisely; across both studies, they seem to have emphasized accuracy over speed. This observation is supported by the increased movement speed for the mid-air recall condition.

Third, users of Vulture must explicitly delimit words by pinching, while touch provides implicit delimiting of words. Several participants said that pinching was natural in that it resembled gripping a pen. However, pinching adds to the complexity of text entry in mid-air compared to touch input.

### How can mid-air WGKs be improved?

A first potential improvement of Vulture would be to pace users, with the aim of increasing their gesture speed. Ideas for doing so include diminishing visual feedback based on speed or diminishing visual feedback over the course of a gesture (strong initially, then gradually removing feedback).

A parameter of the design of Vulture that could be varied is the size of the motor space. In particular, reducing the size of the users' gestures might increase their speed and decrease accuracy.

The results on gesture recall from Study 2 indicate that text entry with no visual feedback does not seem promising. The idea proposed in Imaginary Interfaces [8] of using the non-dominant hand to provide an explicit reference point might make recall of location better. This explicit point of reference could potentially provide the location context needed to support text entry with limited visual feedback.

Word gestures seem to work well in mid-air, because they remove some of the need to repeatedly select characters, which makes other mid-air text-entry techniques tedious to use (e.g., [18]). Other techniques for predictive text entry may be beneficial to mid-air. One such technique is to add a language model to the gesture recognition engine: this would allow users to make less precise and thus faster gestures for producing the intended word.

## CONCLUSION

Word-gesture keyboards (WGKs) allow efficient text entry by tracing of words instead of typing individual letters. WGKs are widely used on touch-based devices. This paper demonstrates how WGKs can be adapted for use also in mid-air. Empirical results from two studies show clear usability benefits compared to existing mid-air text-entry methods. Several issues in designing for mid-air interaction have been discussed. A key issue is that input space and output space are separated, which seems to make interaction more mentally demanding: Participants' gesture movements in mid-air text entry were slower, but with the same accuracy as in touch-based text entry. Based on the empirical results we discussed ideas for improving mid-air WGKs and raised key questions for future research.

## ACKNOWLEDGEMENTS

This work has been supported in part by the Danish Council for Strategic Research, grant 10-092316.

## REFERENCES

1. Amma, C., Georgi, M., and Schultz, T., Airwriting: Hands-Free Mobile Text Input by Spotting and Continuous Recognition of 3d-Space Handwriting with Inertial Sensors. In *Proc. ISWC*, (2012), 52-59.
2. Banerjee, A., Burstyn, J., Girouard, A., and Vertegaal, R., MultiPoint: Comparing laser and manual pointing as remote input in large display interactions. *Int. J. Hum.-Comput. Stud. 70*, 10 (2012), 690-702.
3. Bi, X., Chelba, C., Ouyang, T., Partridge, K., and Zhai, S., Bimanual gesture keyboard. In *Proc. UIST*, ACM (2012), 137-146.
4. Casiez, G., Roussel, N., and Vogel, D., 1 € filter: a simple speed-based low-pass filter for noisy input in interactive systems. In *Proc. CHI*, ACM (2012), 2527-2530.
5. Castellucci, S.J. and MacKenzie, I.S., Graffiti vs. unistrokes: an empirical comparison. In *Proc. CHI*, ACM (2008), 305-308.

6. Cockburn, A., Quinn, P., Gutwin, C., Ramos, G., and Looser, J., Air pointing: Design and evaluation of spatial target acquisition with and without visual feedback. *Int. J. Hum.-Comput. Stud. 69*, 6 (2011), 401-414.
7. Devoe, D.B., Alternatives to Handprinting in the Manual Entry of Data. *IEEE Transactions on Human Factors in Electronics HFE-8*, 1 (1967), 21-32.
8. Gustafson, S., Bierwirth, D., and Baudisch, P., Imaginary interfaces: spatial interaction with empty hands and without visual feedback. In *Proc. UIST*, ACM (2010), 3-12.
9. Jones, B., Sodhi, R., Forsyth, D., Bailey, B., and Maciocci, G., Around device interaction for multiscale navigation. In *Proc. MobileHCI*, ACM (2012), 83-92.
10. Jota, R., Nacenta, M.A., Jorge, J.A., Carpendale, S., and Greenberg, S., A comparison of ray pointing techniques for very large displays. In *Proc. GI*, Canadian Information Processing Society (2010), 269-276.
11. Kristensson, P.O., 2007. *Discrete and Continuous Shape Writing for Text Entry and Control*, Doctoral dissertation, Linköping University, Sweden.
12. Kristensson, P.O., Nicholson, T., and Quigley, A., Continuous recognition of one-handed and two-handed gestures using 3D full-body motion tracking sensors. In *Proc. IUI*, ACM (2012), 89-92.
13. Kristensson, P.O. and Zhai, S., SHARK2: a large vocabulary shorthand writing system for pen-based computers. In *Proc. UIST*, ACM (2004), 43-52.
14. MacKenzie, I.S., Chen, J., and Oniszczak, A., Unipad: single stroke text entry with language-based acceleration. In *Proc. NordiCHI*, ACM (2006), 78-85.
15. MacKenzie, I.S. and Soukoreff, R.W., A character-level error analysis technique for evaluating text entry methods. In *Proc. NordiCHI*, ACM (2002), 243-246.
16. MacKenzie, I.S. and Soukoreff, R.W., Phrase sets for evaluating text entry techniques. In *Proc. CHI*, ACM (2003), 754-755.
17. MacKenzie, I.S. and Tanaka-Ishii, K., *Text entry systems : mobility, accessibility, universality*. Boston : Morgan Kaufmann, Amsterdam, 2007.
18. Markussen, A., Jakobsen, M., and Hornbæk, K., Selection-Based Mid-Air Text Entry on Large Displays. In *Proc. INTERACT*, Springer Berlin Heidelberg (2013), 401-418.
19. McGuire, R.M., Hernandez-Rebollar, J., Starner, T., Henderson, V., Brashear, H., and Ross, D.S., Towards a one-way American sign language translator. In *Proc. AFGR*, (2004), 620-625.
20. Nancel, M., Wagner, J., Pietriga, E., Chapuis, O., and Mackay, W., Mid-air pan-and-zoom on wall-sized displays. In *Proc. CHI*, ACM (2011), 177-186.
21. Ni, T., Bowman, D.A., and North, C., AirStroke: bringing unistroke text entry to freehand gesture interfaces. In *Proc. CHI*, ACM (2011), 2473-2476.
22. *Oddities in words, pictures and figures*. Reader's Digest, London, 1975.
23. Piumsomboon, T., Clark, A., Billinghurst, M., and Cockburn, A., 2013. User-Defined Gestures for Augmented Reality. In *INTERACT*, P. Kotzé, G. Marsden, G. Lindgaard, J. Wesson and M. Winckler Eds. Springer Berlin Heidelberg, 282-299.
24. Proctor, R.W. and Vu, K.P.L., *Stimulus-Response Compatibility Principles: Data, Theory, and Application*. Taylor & Francis, 2006.
25. Schick, A., Morlock, D., Amma, C., Schultz, T., and Stiefelhagen, R., Vision-based handwriting recognition for unrestricted text input in mid-air. In *Proc. ICMI*, ACM (2012), 217-220.
26. Shengli, Z., Zhuxin, D., Li, W.J., and Chung Ping, K., Hand-written character recognition using MEMS motion sensing technology. In *Proc. AIM*, (2008), 1418-1423.
27. Shoemaker, G., Findlater, L., Dawson, J.Q., and Kellogg, B.S., Mid-air text input techniques for very large wall displays. In *Proc. GI*, Canadian Information Processing Society (2009), 231-238.
28. Shoemaker, G., Tang, A., and Booth, K.S., Shadow reaching: a new perspective on interaction for large displays. In *Proc. UIST*, ACM (2007), 53-56.
29. Sutton, J., Air painting with Corel Painter Freestyle and the leap motion controller: a revolutionary new way to paint! In *Proc. ACM SIGGRAPH Studio Talks*, ACM (2013), 1-1.
30. Vogel, D. and Balakrishnan, R., Distant freehand pointing and clicking on very large, high resolution displays. In *Proc. UIST*, ACM (2005), 33-42.
31. Wigdor, D. and Wixon, D., *Brave NUI World: Designing Natural User Interfaces for Touch and Gesture*. Morgan Kaufmann Publishers Inc., 2011.
32. Witt, H., Lawo, M., and Drugge, M., Visual feedback and different frames of reference: the impact on gesture interaction techniques for wearable computing. In *Proc. MobileHCI*, ACM (2008), 293-300.
33. Wobbrock, J.O., Myers, B.A., and Chau, D.H., In-stroke word completion. In *Proc. UIST*, ACM (2006), 333-336.
34. Wobbrock, J.O., Wilson, A.D., and Li, Y., Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. In *Proc. UIST*, ACM (2007), 159-168.
35. Zhai, S. and Kristensson, P.O., Shorthand writing on stylus keyboard. In *Proc. CHI*, ACM (2003), 97-104.
36. Zhai, S. and Kristensson, P.O., The word-gesture keyboard: reimagining keyboard interaction. *Commun. ACM 55*, 9 (2012), 91-101.
37. Zhai, S., Kristensson, P.O., Gong, P., Greiner, M., Peng, S.A., Liu, L.M., and Dunnigan, A., Shapewriter on the iphone: from the laboratory to the real world. In *Proc. CHI*, ACM (2009), 2667-2670.