PianoText: Redesigning the Piano Keyboard for Text Entry

Anna Maria Feit[1]
afeit@mpi-inf.mpg.de

Antti Oulasvirta[1,2]
oantti@mpi-inf.mpg.de

[1]Max Planck Institute for Informatics & Saarland University
[2]Aalto University, School of Electrical Engineering

ABSTRACT

Inspired by the high keying rates of skilled pianists, we study the design of piano keyboards for rapid text entry. We review the qualities of the piano as an input device, observing four design opportunities: 1) chords, 2) redundancy (more keys than letters in English), 3) the transfer of musical skill and 4) optional sound feedback. Although some have been utilized in previous text entry methods, our goal is to exploit all four in a single design. We present *PianoText*, a computationally designed mapping that assigns letter sequences of English to frequent note transitions of music. It allows fast text entry on any MIDI-enabled keyboard and was evaluated in two transcription typing studies. Both show an achievable rate of over 80 words per minute. This parallels the rates of expert Qwerty typists and doubles that of a previous piano-based design from the 19th century. We also design *PianoText-Mini*, which allows for comparable performance in a portable form factor. Informed by the studies, we estimate the upper bound of typing performance, draw implications to other text entry methods, and critically discuss outstanding design challenges.

Author Keywords

Text entry; the piano keyboard;

ACM Classification Keywords

H.5.2. User-Interfaces: Input devices and strategies

INTRODUCTION

This paper presents *PianoText*, a novel text entry method for MIDI-capable piano keyboards (Figure 1). In its design we exploit the motor skill of pianists and combine features of music and text entry research. *PianoText* allows pianists to enter text at the typing rate of expert Qwerty typists while requiring only a fraction of their training time.

Underlying our research is amazement at the performance of skilled pianists. When playing the Flight of the Bumblebee at regular speed, a pianist clocks 17 notes per second, which would translate to an astonishing 204 words per minute (wpm). Keying rates of elite pianists can even reach 30 notes per second. [24] Such feats demand an amazing coordinated performance of the whole musculoskeletal system of the upper extremities to control for position, force and rhythm of key presses [10]. This raises the question whether we can utilize such capacity to enter *text* on the piano at similar speed. However, one must note that music performance is the result of intense practice of the same piece. A text method limited to a few preselected phrases would be practically useless. But if concentrated practice is not allowed, can piano-based typing still reach high keying rates?

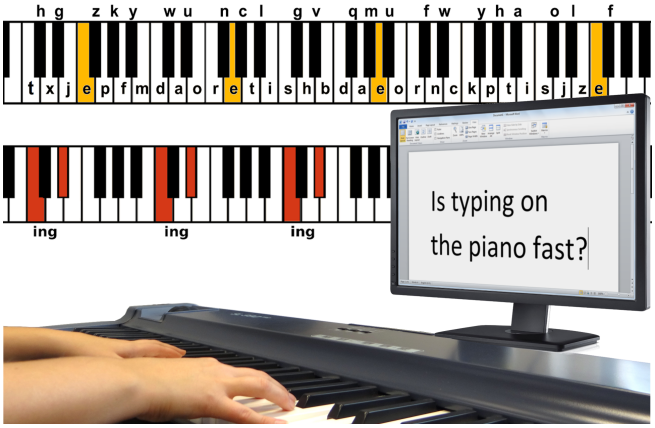

Figure 1: *PianoText* allows text entry on any MIDI capable piano. A computationally designed mapping offers multiple keys for one letter (yellow) and translates chords (red) to letter sequences in English

Interestingly, among the first typewriter designs, predating the familiar Qwerty keyboard, was the piano-based Hughes-Phelps telegraph that had an alphabetical mapping of letters to piano-like keys [1] (Figure 2). It allowed a maximum rate of 40 wpm in professional use [4], but was forgotten after the success of button-based typewriters. Later, features of the piano gave inspiration to chording devices [6, 9] and more recently to chording gestures on multitouch displays [16], but the *full* keyboard was never revisited.

Our goal is to explore the potential of the piano for fast text entry, considering features that were missed in earlier attempts. PianoText is essentially a *mapping* that translates key presses on the piano to letters and letter sequences in English. The design of the mapping exploits four concepts. Although known and used in other devices, their combination in one text entry method is unique:

1. **Redundancy:** The high number of keys allows us to map one letter to multiple piano keys, creating "islands" of statistically co-occurring letters on different parts of the piano. This decreases the hand-travel distance and allows for more hand alternation.
2. **Chords:** In addition to single key strokes, a combination of keys can be pressed to enter frequent and long n-grams

DIS 2014, June 21–25, 2014, Vancouver, BC, Canada.

http://dx.doi.org/10.1145/2598510.2598547

with only one movement. This scales up performance with practice, while PianoText can also be used without chords.

3. **Skill transfer:** Our mapping respects frequently played note transitions in piano music, assuming that those are familiar to pianists and thus quick to type. Although the mapping is learnable by anyone, this approach allows pianists to adopt the method quicker.
4. **Sound:** PianoText can be played with or without sound. The musical, although atonal feedback identifies the key presses, allowing the typist to monitor the input.

The design of the mapping follows a computational approach. Optimization methods have been previously used in virtual keyboard design to address the NP-complete problem of mapping letters to keyslots [30]. However, to make use of the pianist's expertise, our objective function must consider *two* frequency distributions: music (source) and language (target), whereas normally only language is considered. To further constrain the algorithmic search we visit studies of piano playing and text entry to derive a heuristic approach that considers the frequency, distance and trajectories of key presses.

To critically assess performance achievable with PianoText, we conducted two empirical evaluations with two different typing tasks. The first involved a concert level pianist typing sentences by "sight reading" from text translated into sheet music. It analyzed sentences translated with different mappings for their performance and playability. The second study was conducted with a hobby pianist training for over 140 hours to memorize the mapping and generate text. Both studies show a top range of over 80 wpm, which compares favorably to other techniques after similar amount of practice.

To understand if piano-based typing allows a more practical form factor than a full-sized piano, we design *PianoText-Mini* which is of comparable width to the regular keyboard and also implements error correction. Empirical evaluation shows that its fewer and narrower keys enables similar text entry rates as the full-sized version and more efficient error correction than regular text entry methods.

Based on the obtained results we critically discuss the value of the features exploited in the design of PianoText. We estimate an upper bound for piano-based typing performance and identify remaining challenges to the design. We conclude by discussing opportunities in how other text entry methods might make use of redundancy and chords.

To sum up, this paper advances the design of text entry methods by investigating the piano keyboard as an input device. It makes three contributions:

1. It presents PianoText, a high performance text entry method that doubles former rates in piano-based typing and reaches a level comparable to the best existing methods.
2. It presents a novel computational approach to letter-to-key mappings grounded on existing literature and leverages statistics of music *and* language for more efficient input.
3. It reports first empirical investigations and analysis of two core design concepts in piano-based typing: redundancy and chords.

Figure 2: The Hughes-Phelps telegraph is a piano-based text entry device used in the 19th century by telegraph operators. Its design used a quasi-alphabetical ordering of letters to piano-like keys.

CHARACTERISTICS OF THE PIANO FOR TEXT INPUT

We analyze the piano keyboard first as an input device and then from the musical perspective. This allows us to infer novel opportunities to enhance typing performance.

The Piano as an Input Device

The piano keyboard is a two-handed input device operated by discrete key presses. The design of individual keys and their physical layout differs from the standard Qwerty keyboard. One key is about 2.5 cm wide. With 88 keys this covers a total width of about 120 cm. Upon pressing, button displacement is about 1 cm, which is much larger than the 0.1–0.5 cm of the standard keyboard. Unlike Qwerty, any combination of fingers can be used in simultaneous key presses. The elongated shape of the keys better supports chord presses than the rectangular keys of Qwerty. The piano also naturally offers audio feedback, which identifies the pressed key as well as stroke characteristics such as force and dynamics.

The very different layout of piano keys, on "1.5" versus 3 rows on the Qwerty keyboard, has deeper implications for input. First, it implies a much greater role for lateral movements of the arm. However, the wider and larger keys require less precise aiming. Secondly, because of the large number of keys there is no fixed finger-to-key-mapping. The pianist decides the so-called "fingering" dynamically to find a smooth transition between the notes. However, there is a home position similar to the home row of the Qwerty keyboard.

The Piano as a Musical Instrument

The 88 piano keys are organized in 7 octaves where each octave contains 12 keys, one for each of the pitch classes. As shown in the helix model in Figure 3, the classes repeat circularly while the height of the pitch increases [7]. On the piano, every pitch class (here D) is represented by 7 different keys whose tones are perceived to have the same acoustic color.

A piece of music can be said to consist of two parts: melody and harmony. The melody is formed by sequential presses, while the harmony makes use of chords, simultaneously sounding tones. This is realized by pressing multiple keys at once. A chord can be defined by its root and the intervals of its tones. Similar to the circular model of pitch, the same chord can be played in different octaves. The sound color of the chord is perceived as being the same; only the height differs. This is a principle we will later exploit.

Similar to other highly practiced tasks, motor memory plays an important role in music performance. When designing for

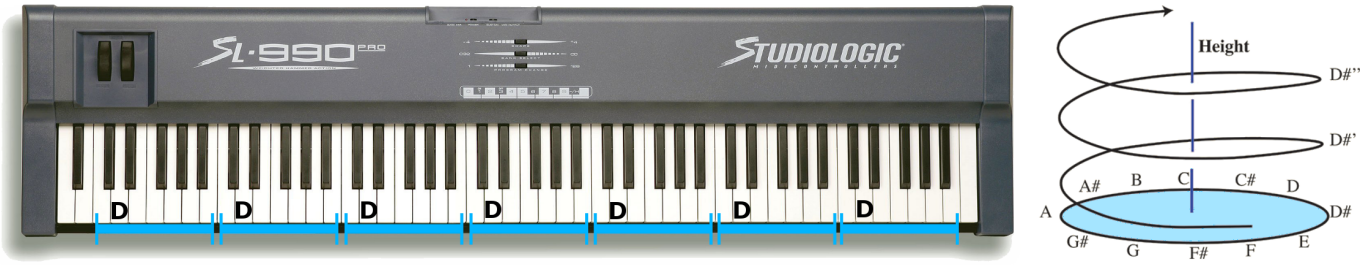

Figure 3: The piano keyboard affords the use of chords and mapping letters multiply. The 88 keys are organized into seven octaves in which the 12 pitch classes repeat while the height of the tones increases. The linear and circular dimension of pitch is demonstrated by the helix model on the right [7].

text entry, we look at the basic structures a pianist has to practice regularly in order to maintain their level of skill. Finger exercises on the major and minor scales for example are part of their daily practice and determine the movements a pianist can perform quickly and accurately.

While playing music, the pianist relies on the sound feedback of the instrument. It guides the location of the fingers by identifying the pressed key and is essential for keeping rhythm and calculating the necessary stroke velocity to convey the right expressiveness [13]. Moreover, the sound has structure: Strict composition rules allow an experienced pianist to predict upcoming motor responses.

IMPLICATIONS FOR DESIGN AND RELATED WORK

Table 1 provides a point-by-point comparison of the piano against other button- and chord-based text entry devices operated by two hands. Although none of the features are unique, the combination is. To provide an overview of typing performance, the table also reports rates achievable with 50-200 hours of practice. We note that the vast differences among the studies in users groups, training and measurements prevent reliable comparison, so this assessment is tentative. However, rates above 50 wpm can be considered promising.

To guide our attempt at making the best out of these features, we summarize the above considerations into four design opportunities for improving text entry performance:

1. Redundancy: The concept of *redundancy* describes the fact that different actions lead to the same input. It is inspired by the periodicity of the pitch classes over the octaves of the piano. Most text entry methods that implement redundancy are limited to the use of multiple spaces, such as the two-thumb keyboard *KALQ* [23]. Our hypothesis is that a larger number of replicated keys can shorten the hand-travel distance and improve hand alternation. However, it may also increase the cognitive demand on the user.

2. Chords: Musical chords inspired the design of text entry devices [6, 9] and more recently chording gestures on multitouch displays [16]. Studies showed that a high number of chords can be memorized and that chording devices can surpass Qwerty in early stages of training [15]. However, there are different ways to implement chords:

1. Due to lack of space, pressing a combination of keys may be used to enter a single letter, requiring fewer buttons than letters in the alphabet. Engelbart's keyset [9] is one of the most famous of such devices.
2. Pressing and holding a series of keys is used for example in the implementation of hotkeys, where the *Ctrl* key changes the mode of operation. Seibel [28] implemented user-defined shortcuts to enter n-grams on the Qwerty. Performance improved by 12-25% after 70 hours of training.
3. Pressing a combination of keys *at once* to enter a sequence of letters is used by shorthand machines like the stenograph. Projects like Plover [17] try to bring stenography to the Qwerty user. While reports suggest great benefits, they require extensive training of up to three years [3].

Other devices combine different types of chords. The one-handed chording keyboard Twiddler [18] implements the first variant but also investigates the use of so-called *multicharacter chords*. They implemented 12 chords to enter n-grams of at least 3 characters, which improved the typing rate by 8%.

3. Expertise transfer: In our design we want to leverage the motor skill of pianists. Our hypothesis is that finger trajectories common in music will allow pianists higher input speed as well as shorten the learning process. To our knowledge, transferring expertise to another domain is an approach that has not been followed in the design of text entry methods.

4. Sound: While the pianist relies on the sound feedback, Qwerty typists have much simpler auditory feedback of button presses. In contrast to the piano, the sound only serves as a confirmation that a key was pressed, which is already enough to improve performance in mobile text entry [5]. The complex musical feedback of the piano may further benefit training time and accuracy.

Device	wpm (training hours)	Reference	Width, height (cm)*	No. keys**	Redundancy	Chords	Sound
Qwerty	50 (200)	[27]	20, 6	26	○	○	○
Soft Qwerty	58 (n/a)	[12]	25, 6	26	○	○	◒
2-thumb	37 (19)	[23]	2× 5, 5	28	◒	○	◒
Chord kbd	60 (60)	[15]	20, 8	10	○	●	○
Stenograph	34 (111)	[3]	16, 10	22	○	●	○
H-P Telegr.	40 (n/k)	[4]	28, 15	26	○	○	○
This paper	81 (140)		121, 14	88	●	●	●

● covered ◒ partially covered ○ not covered

Table 1: Comparison of the piano against some other button- and chord-based text entry methods.
***: Estimated area with alphabetical keys.**
****: English, spacebar omitted.**

DESIGN APPROACH

We consider the design of a letter-to-key mapping from English language to the piano keyboard. The goal for our mapping is to minimize the time between two successive keystrokes, that is the inter-key interval (IKI). This is addressed by the following objectives:

- **Minimize inter-key distances**: Following previous work on keyboard optimization, we aim to *locally* minimize the average distance between consecutive key presses [19, 30], that is key presses performed with the same hand. Particularly detrimental are long "jumps" that require large lateral arm movements and visual attention.
- **Maximize hand alternation**: Studies of motor performance suggest that movement preparation [26] is a major factor in expert typists' and pianists' performance. When hand alternation is increased, the proportion of responses benefiting from this effect increases.
- **Exploit existing skill**: Frequently executed motor programs of a pianist can be performed with shorter IKIs and should therefore be favored.

To address these objectives, we relax the assumption of a one-to-one mapping, as used in the Hughes-Phelps telegraph, and aim to create a one-to-many mapping with two components: (1) A **letter mapping** where letters can be multiply instantiated, and (2) a **chord mapping** which assigns musical chords to frequent and long n-grams (the third alternative in the previous section). Redundancy and chords allow us to approach all three of the design objectives. Mapping one letter to different keys on the piano creates "islands" of key clusters where frequent letter sequences are close together and local inter-key distance is very small. Hand alternation is used to quickly switch between these clusters and is enabled by mapping frequent letters at least twice, once to each side of the piano. Redundancy also provides a way to leverage the *motor expertise* of a pianist, offering more keys to map frequent letter pairs to musical intervals. The pianist's skill is also exploited by the chord mapping that assigns only chords common in music. Exploiting the periodicity of the piano, a chord can be entered in any octave. This enables quick hand alternation and reduces the hand-travel distance for pressing a chord.

TECHNICAL IMPLEMENTATION

The combinatorial problem of one-to-one mapping of letters to key slots is NP-complete and the possibility of one-to-many mapping, as in our case, increases the search space tremendously. For instance, if we require all letters to be used at least once, the 88 keys of a piano keyboard can be mapped to the 26 letters of English in $26^{88-26} + 26! \approx 5 \times 10^{87}$ ways. With chords mapped to n-grams the search space is even larger.

We create an algorithm that formulates the problem of finding a one-to-many mapping in terms of placing letters on a 1-D grid, following an approach that has been found beneficial in the analogous (but simpler) problem of virtual keyboard design [19, 30, 8]. Unfortunately, the previously deployed optimization methods are not applicable here, because there are no quantitative predictive models for aimed movements

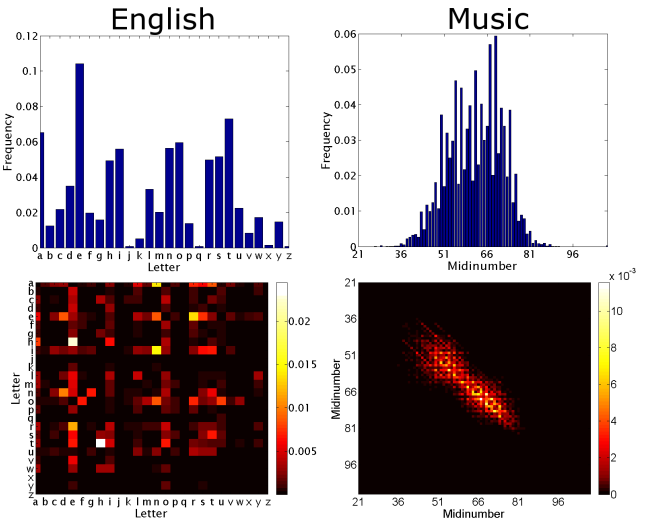

Figure 4: The frequency distributions of 1-grams (top) and 2-grams (bottom) of the English language versus piano music.

in piano playing. Instead we build on the hypothesis that the most frequently encountered notes and note transitions will also be the fastest to respond. We implement a greedy algorithm with additional constraints to construct a mapping that assigns *n-grams* of music and language, allowing a pianist to use the same finger trajectories in entering text as for playing music. Thus the algorithm needs to consider *two* distributions: music (source) *and* language (target).

N-gram Distributions for Music and English

For mapping, we acquired the distributions of 1-grams (single notes) and 2-grams (note transitions) from ten sight-reading practice books. Those contain basic musical structures common to many different pieces and thus well practiced by pianists. The distributions of letters and letter pairs in the English language was acquired from a corpus of classical literature[1]. For the chord mapping, we consider all major and minor chords in music, their inversions and the corresponding harmonic intervals, as well as the 100 most common words in the English language[2].

Figure 4 shows the distributions of 1- and 2-grams of English and the analyzed music corpus. In the top right graph we can see that the keys located in the middle of the piano are more frequent. The oscillatory structure shows that the white keys are used more often than the black keys. The diagonal structure in the bottom right graph shows that intervals in music, that is key transitions, are only composed up to a limited size, which should be respected by the mapping.

While we could match the statistics of 1-grams with a simple one-to-one mapping, the categorical organization of letters does not allow this for the 2-gram distributions. However, this is a crucial point in finding a good mapping, as the motor expertise of a pianist is based on frequent musical intervals. Creating a one-to-many mapping by introducing redundancy will enable us to match the 2-gram statistics as well.

[1] http://www.data-compression.com/english.html
[2] http://oxforddictionaries.com/

Mapping Algorithm

We use a greedy algorithm with additional constraints to construct a letter-to-key mapping. Following three steps, it favors frequently played notes while avoiding infrequent intervals and minimizing hand-travel distances:

1. **Frequency:** Go through the letters in order of their frequency. Map each letter to the next most frequent note. Start by mapping 'e', the most frequent letter in English, to A4 the most frequent note in the music corpus.
2. **Interval:** After each mapped letter go through all frequent bigrams that contain this letter. Check if the corresponding note transition occurs frequently in music. If this is not the case, remap the letter to the next most frequent note and check again.
3. **Distance:** After all letters are mapped once, go through all frequent letter pairs and check the inter-key distance of the corresponding notes. If the distance is too large, reduce it by mapping the more frequent letter to an additional key.

A more detailed description of the algorithm and a pseudocode implementation can be found in Appendix A. We used it to create two different letter-to-key assignments: (1) The output of the algorithm when applied to the whole piano is called the *unity*-mapping. (2) The *split*-mapping is constructed by applying the algorithm twice. First only to the right side of the piano and then to the left side as well. This ensures that most of the letters are mapped at least twice and accessible by each hand. The split-mapping has the potential to further improve hand alternation and is implemented in PianoText.

The letter mapping is extended by a *chord*-mapping that assigns n-grams to chords of the major and minor scales. We map those n-grams that maximize the gain of a chord:

$$G(chord, n\text{-}gram) = (T_{n\text{-}gram} - T_{chord}) \cdot P(n\text{-}gram) \quad (1)$$

where $T_{n\text{-}gram}$ denotes the time to enter a n-gram by single key presses, T_{chord} the time to enter a chord. $P(n\text{-}gram)$ denotes the frequency of the n-gram in a given corpus.

OUTCOME: PIANOTEXT

We analyze the outcome, PianoText, with respect to the frequency distributions of the English language as given by the Enron email dataset [29]. The letter-to-key assignment is shown in Figure 1. 55 keys are used to map most of the letters at least twice. The chord mapping assigns 57 chords to frequent n-grams. They cover 34% of English and enter on average 3.6 characters, ranging from 2 to 7. The full mapping is shown in the Appendix A. To improve hand-travel distance we map the space to the *pedal* of the piano, which is controlled by the foot. In the following we look at the successful implementation of the three design objectives:

- *Hand alternation* is possible at nearly all times through the implementation of redundancy and chords. On average, each letter is mapped 2.1 times, ranging from 1 key for infrequent letters like 'q' or 'z' up to 3 or 4 keys for very frequent letters, like 't' or 'e'. Chords can be entered in every octave of the piano, that is at 7 different positions.

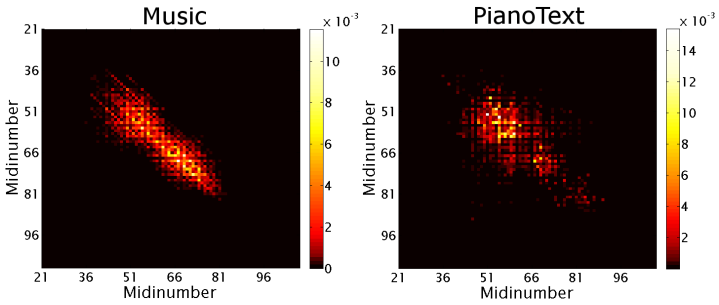

Figure 5: Key transitions of music versus PianoText. The lighter the color the more frequent the key pair. The similarity of the two graphs shows that the pianist can use the same key-trajectories for entering text as for playing music on the piano.

- The average *inter-key distance* is 6 keys, an improvement of 33% with respect to the Hughes-Phelps telegraph (9 keys). Chords can be entered in any octave, which further improves inter-key distance.
- *Skill transfer:* PianoText maps the most frequent letter pairs to the most common intervals in music, allowing pianists to use their existing motor skill. This can be seen in Figure 5 showing the frequencies of 2-grams in playing music and entering text on the piano.

STUDY 1: COMPARISON OF MAPPINGS

In the first study, our purpose is to assess the input performance and playability of sentences translated by different mappings, that is the difficulty for a piano player to enter the notes corresponding to a sentence. The *unity*- and the *split*-mapping (see section *Mapping Algorithm*) are compared in several conditions that highlight different features of piano-based typing. We created a *music-transcription task* in which we first translate sentences into sheets of music according to each mapping. They are then played by a professional *sight-reader*, a pianist skilled in playing music directly from the page. This task emulates the skilled level of performance in a transcription task and lets us estimate upper performance rates of piano-based typing. Because of the rarity of such pianists, our sample was limited to a single expert subject, a lecturer at the local university of music. Single-subject studies are common in studies of expert performance and justifiable when the purpose is to estimate maximum rates [11].

Method

Participant: The participant was 39 years old and volunteered without a fee. He has played the piano since he was 5 and has won more than 40 national and international competitions in piano-playing. Currently he teaches classes for advanced students at the university of music in Saarbrücken.

Task materials: The unity- and split-mapping were tested in two conditions: with and without the chord mapping described in the previous section. To test the influence of different chord mappings, the *unity*-mapping was tested in two more conditions: (1) the *N-Grams* condition in which chords were mapped to the 30 most frequent letter pairs and 8 most frequent prefixes and suffixes; (2) the *Words* condition where chords were mapped to the 100 most common English words. For all mappings, several practice sheets and 2 - 4 test sheets

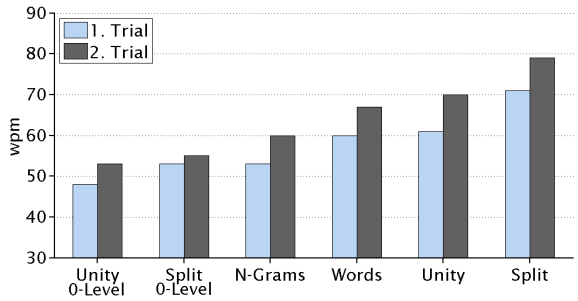

Figure 6: Typing rates in a music transcription task for the mapping variants tested in Study 1.

were created with phrases adopted from a recently published corpus: the Enron email dataset [29]. Each sheet had on average eight phrases (196 characters).

Apparatus: Our piano apparatus is a Yamaha grand piano (Mark IV Series Disklavier Full-Function Grand Piano DC2M4) with a MIDI recording option. Audio output was on and the performance was recorded on video.

Procedure: Two sessions were organized on separate days. The pianist was explained that the goal is to play as quickly as possible and disregard tempo and rhythm. Before the actual trials, he practiced four sheets of increasing complexity to adapt to the atonality and the goal of playing fast. Then, two previously unseen sheets of each mapping were played, each two times (trial 1, trial 2). A sheet was presented to the pianist, who had to start playing after 10 seconds. Performance was measured from the first note to the last notes.

Results

Typing performance was measured in words per minute (wpm), assuming 5 characters/word. Error rate was calculated as the proportion of letter-level omission and commission errors computed from the MIDI recordings.

Figure 6 shows the results. The best performance was achieved by the split-mapping, which reached an average rate of 71 wpm in the first and 79 wpm in the second trial. The top performance seen was 84 wpm. This mapping is implemented in PianoText. Three observations are made. 1) Mappings involving chords are superior to letter mappings, at the cost of committed errors (6.8% vs. 2.97%) 2) Chords' benefit is highest when mapped to longer sequences (words). In many cases, the total entry time of the letter pair does not exceed the cost of a chord press. 3) The split-mapping surpassed the unity-mapping by 16%, though they use the same number of keys and chords. This shows the benefit of mapping one letter to each side of the piano, allowing hand alternation at all times.

The translated music sheets favor note structures that are common in music and thus cognitively as well as motorically fast to respond to. The pianist stated that the music was similar to atonal pieces he previously played and was thus easy to adapt to. However, he needed some training time to implement our instruction to play every note as fast as possible without keeping an overall rhythm, very uncommon in music but necessary to reach high input speed.

STUDY 2: SKILLED PERFORMANCE

In study 1 we looked at transcription from sheet music. The goal in study 2 is to assess the learnability of the mapping given the large number of keys and chords introduced by redundancy. While in the music transcription task the to-be-pressed keys were determined by the given sheet, here the typist must decide which of the many alternatives to use. The availability of a skilled piano typist also enabled us to further investigate the effects of chords and redundancy as well as the role of sound unique to the piano.

In total 140 hours of training were carried out to memorize the mapping, which compares roughly to the intermediate range of typing expertise with standard keyboards [27].

Method

Participant: Our participant is German and studied English for 7 years in school. She is a 20-year-old hobby pianist with 11 years of training through weekly instruction. She practices and plays about 3 hours per week. A participation fee of 10 €/h was paid. Her performance in a controlled test of typing speed with a Qwerty keyboard is 72 wpm (5% error rate) for German and 44 wpm (6% error rate) for English.

Material and Apparatus: For exercises, regular tests and the final experiments, stimulus sentences were taken from the Enron email dataset [29]. The piano was a Studiologic SL 990Pro. We implemented a training program in Java to display and control the exercises and tests. It presents stimulus phrases and supports different options for showing or hiding feedback on wpm, error rate and correctness of user's input. We implemented a driver for PianoText to be used in any application, which translates the piano keystrokes to key events of the operating system.

Training Program: The training spanned 25 weeks with an average of 6 hours per week. The first three weeks were dedicated to memorizing the letter mapping. Chord practice started within the seventh week, 4–10 per week with more at the beginning, over 7 weeks. Care was taken that basic exercises, like practicing frequent letter pairs and chords, were repeated with an expanding schedule. After week 14 we added special practice on metronome-paced typing, overloading of chords, and accuracy. In addition the participant should practice her skills in online typing games. Example stimuli and more details can be found in Appendix B.

Regular Performance Measurement: Typing performance was measured in words per minute, of 5 characters/word. Error rate was measured with the Damerau-Levenshtein distance. To test the performance, we created 21 phrase sets of 35–40 sentences from the Enron email dataset. The sentences were annotated using their memorability metadata as the cut-off [29]. To avoid a confounding effect of repetition, the test sentences were novel and never encountered in practice or previous tests. These assessments were administered once every week. The difficulty of the sets increased throughout the training, because the number of sentences with similar memorability and length was limited. However, for comparability, the last test consisted of sentences with intermediate length and memorability rate.

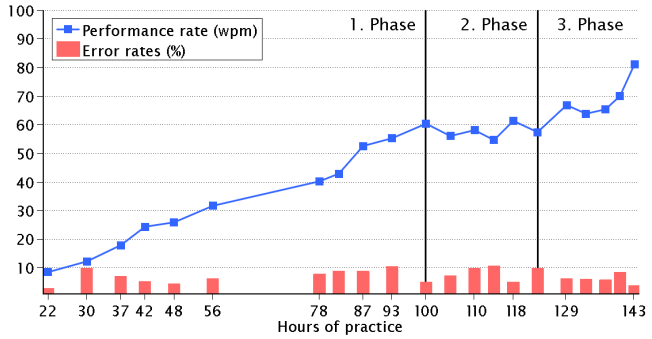

Figure 7: Typing performance in Study 2 over the training time of 140 hours. Measured weekly in controlled tests.

Controlled Experiments: After the end of training, three 4-hour sessions were organized to investigate the effect of redundancy, chords, hand alternation and sound. We describe the controlled experiments within the *Results* section.

Results: Learnability

The development of performance over time in Figure 7 shows that it is possible to learn the mapping despite its complexity. A hobby pianist who has memorized the mapping can surpass the rate of a concert-level pianist entering text by sight reading from translated sheets (Study 1).

The learning curve in Figure 7 consists of three phases. The first describes a constant performance increase over 100 hours of training, from the initial performance of 8.5 wpm, measured after the participant memorized the letter mapping, up to an intermediate performance of 60 wpm. The exercises and factors that were most influential in this period were 1) practicing frequent letter pairs and words, repeatedly with increasing intervals; 2) the successive introduction of chords, training the memorization as well recognition in text; 3) touch typing, that is without looking at the keys, where the existing orientation on the piano led to a performance increase of more than 10 wpm within one week.

The second phase of the learning curve shows a plateau lasting 23 hours of training. In this time, special exercises were performed that aimed to correct mistakes and better exploit musicality. The participant was meant to imagine sounds of words and play at the tempo of a metronome, but neither exercise showed immediate effects.

The third phase shows a further increase of performance over 20 hours of practice. In this phase special exercises were conducted that focused on the fast and accurate use of chords together with single key presses: the inter-key interval before and after chord presses improved and hand alternation after chord usage increased. Practicing chords also reduces the error rate, because one erroneous chord inflates the error rate by several characters.

After 140 hours of training, a typing rate of 81 wpm was reached. Through the whole study, error rates ranged between 2.0% and 10.7% decreasing to 4.0% in the end. The participant could master the use of 57 chords which covered 28% of her entered text, with an error rate of 11%. The video recording of the last test is provided in the supplementary material.

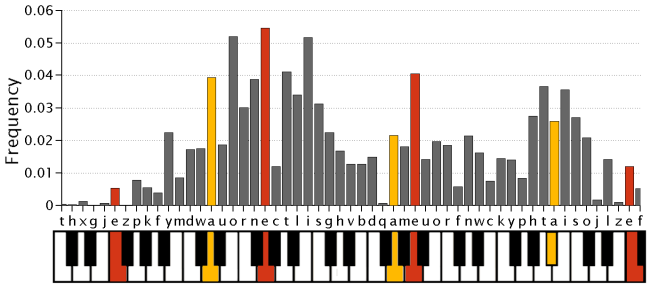

Figure 8: Frequency of strokes on the piano keyboard in Study 2. Usages of 'a' and 'e' are highlighted to show the exploitation of redundant keys.

Results: Controlled Experiments

We conducted a series of controlled experiments that analyzed the gains through chords and redundancy, showed the importance of hand alternation and investigated the effect of sound feedback.

Redundancy: To estimate the gain of redundancy, the participant was given a set of 16 frequent words of 7–9 characters each. She was asked to practice them in two conditions: freely on the whole piano and restricted to use only the keys numbered 21–40 of the mapping. The latter case restricted the choice of response to one key per letter. Mean IKI was 142 ms in the restricted condition and 116 ms in the non-restricted, a statistically significant difference, $t(14)=-3.57, p=0.003$. Mean inter-key distance was found to be 6 keys in the restricted and 4.4 keys in the non-restricted condition, again a statistically significant difference, $t(14)=2.32, p=0.037$. Thus, mapping letters at least twice such that each hand can reach one of the keys improves performance by 18% and reduces the inter-key distance by 26%.

The distribution of keystrokes on the piano in Figure 8 shows that the pianist makes use of the redundant keys. As an example, the letters 'a' and 'e' are highlighted. Analysis of exercises and tests showed that the typist exhibits a fixed response to frequent letter sequences. Depending on the context, the letter 'a' is for example entered with key number 32 for the word "read", or key number 47 in the word "thanks".

Chords: To estimate the gain of chords, the participant was presented 14 sentences containing 1–3 chords, each in a context of single letters. The sentences were practiced in two conditions: with and without chords. Average top speed was 99 wpm when utilizing chords and 70 wpm without chords. A t-test showed a statistically significant difference, $t(13)=4.86, p<0.001$. The letter sequences covered by chords were analyzed for their time to be entered by a chord or by single key presses. The benefit of each chord could then be computed as given by Equation 1.

The 57 chords implemented in PianoText cover 34% of text. On average, pressing a chord was found to be 90% faster than entering the letter sequence by single key presses. This increases the text entry rate by 30.2%.

Hand Alternation: We investigated the combination of single key presses and chords. IKIs (the interval between chord and key presses) were measured for letter-chord pairs in different conditions: played by alternating hands (2H) or using

one hand (1H). The pairs were distinguished by the keys involved: only white keys or black and white keys that require fore/aft movements. The mean IKI of white-key pairs was 265 ms in the 1H condition and 119 ms in the 2H condition. Pairs containing black and white keys had a mean IKI of 279 ms in the 1H condition and 103 ms in the 2H condition. This shows that hand alternation is more than twice as fast as playing with one hand. Surprisingly, this was even found in the case where no fore/aft movement was required, comparable to the home row usage on the standard keyboard. In contrast, for single key presses it is known that hand alternation brings no benefit when typing in the home row or pressing only white keys on the piano [26].

Sound: Surprisingly, the audio feedback of PianoText was not found to influence typing performance. On two separate days, the participant was asked to type two sets of 35 sentences each in two conditions: on one day with, on the other without sound. Visual feedback was provided throughout. In the first set she reached an average performance of 84 wpm (6.1% error) with sound vs. 85 wpm (9.0% error) without sound. A t-test showed no statistically significant difference, $t(22)$=$-0.09, p$=0.93 (for error rate: $t(22)$=$0.65, p$=0.52). The same was found for the second set. Average performance was 81 wpm (7.1% error) with sound vs. 75 wpm (5.4%) without sound. No statistically significant difference was found, neither in performance ($t(16)$=$-1.72, p$=0.104) nor in error rate ($t(16)$=$-0.87, p = 0.4$).

PIANOTEXT–MINI: REDESIGN FOR A SMALLER FORM FACTOR

The results reported above are promising for piano-based typing but were obtained with a 120 cm-wide keyboard. To understand how this scales down to the form factor required by present-day computing devices, we implemented a reduced version similar in size to a regular physical keyboard in workstation PCs. PianoText-Mini is implemented on a commodity keyboard and also has error correction (Figure 9). It is inexpensive, lightweight and almost as portable as a regular physical Qwerty keyboard.

The theory of generalized motor programs [25] suggests that the skill trained on the full PianoText should be transferable to the smaller version. The transition is similar to switching between regular keyboards of different sizes or on different devices, as explored for example in [12]. Moreover, although we unavoidably lose much redundancy, the smaller size could also bear performance benefits. Hand-travel distances will be smaller and it could better support blind typing because long lateral arm movements are not needed.

Keyboard: We use a *Korg microKEY 37* USB MIDI piano with 37 keys, dimensions of 56.5 × 14 × 5 cm and a weight of 1kg. The individual keys are 2 cm wide and button displacement upon pressing is 0.8 cm.

Mapping: We cut the mapping of PianoText at the 37th key, utilizing the letters from lower 't' to upper 'r' (Figure 9. This contains the full alphabet and covers 90.15% of key presses performed in Study 2. PianoText-Mini maps 10 letters multiply. The 37 keys offer 3 octaves for entering chords.

Figure 9: We redesign PianoText to be used on a 37 key piano, similar in size to a physical Qwerty keyboard. It shows the applicability of piano-based typing in a regular working environment.

Error correction: A letter sequence can be deleted by pressing the 2-key chord consisting of the notes C and C# (the interval of a small second). Like the chords, it can be entered in any octave. However, the sound feedback is dissonant and thus unique among the chords. Periodic availability yields a low inter-key distance of up to five keys and allows hand alternation. We implemented the chord to delete letters in the same unit they were entered. Accordingly, a n-gram entered by an erroneous chord can be deleted by only one key press.

Evaluation

We conducted two short studies to test the performance of PianoText-Mini and the practicality of error correction.

Method: Participant, software and materials were the same as in Study 2. A training time of 5 hours and standard exercises practicing bigrams, chords, frequent words and full sentences were given to familiarize the new device. Special exercises favored letter sequences previously entered on the cropped part of the piano. At the end of the training a performance measurement was conducted as described in Study 2. For error correction, the typist was introduced to the new method and given a set of sentences to freely practice the detection as well as the correction of errors. After 30 minutes of training she was asked to transcribe two sets of 30 sentences each and told to correct errors if detected immediately.

Results: After 5 hours of training the typist reached a performance of 79 wpm with an error rate of 7%. In the error correction study, keystrokes per character (KSPC [20]) were measured as a metric for error correction. KSPC increased from 0.78 (no error correction) in Study 2 to 0.97. The error rate dropped to 1.4%. These results show that PianoText-Mini provides a portable alternative to the full PianoText without sacrificing performance and offers efficient error correction.

DISCUSSION

Many researchers have already pointed out that we need to challenge the Qwerty keyboard as the standard text entry method [21] and even rethink its form factor [22] in order to find substantial improvements in text entry.Thus, this paper seeked to find inspiration from another high performance bimanual task: piano playing. We analyzed the potential of the piano to produce *text* instead of music and argued that redundancy and chords in particular could offer distinct benefits. Following this idea, we modified an existing computational

approach to keyboard design to exploit the frequency distributions of music and English language. PianoText is the first text entry device that combines the concepts of redundancy and chords, offers sound feedback and allows pianists to benefit from their existing motor skill.

The two empirical studies demonstrate the learnability of the mapping despite its apparent complexity. After 140 hours of training, the 55 different keys and 57 chords were memorized and utilized. The achieved rate of over 80 wpm compares favorably to *expert* Qwerty typists, who need a training time of 200 hours to reach a typing rate of 50 wpm, as cited in [27], and hundreds to thousands more to reach the level of 60 – 113 wpm [14]. Such levels of performance are rarely to be found in recent text entry research (Table 1). In the design of PianoText-Mini we explored the possibility of a small and affordable version. The smaller form factor and implementation of error correction improves the applicability of piano-based typing.

We conclude with a discussion of the potential expert performance of piano-based typing and suggest design ideas for other categories of text entry.

Upper Bound for Piano-based Typing?

We see several opportunities to improve the design of PianoText and the accompanying training program:

Number of chords: We can increase the number of chords to improve performance as done in [28] . We estimate the concrete benefit of this by extrapolating our findings on the gain of chords. The 57 chords of PianoText cover 34% of language. They were found to be 90% faster than entering the n-gram by single key presses and thus improve typing speed by 30.2%. Thus, 100 chords, covering nearly 50% of English, would increase performance by 45%.

Mapping Optimization: The mapping could be improved with respect to hand-travel distance. Our algorithm optimizes the layout for the transfer of musical skill at the cost of expert performance. It favors musical intervals of 3 and 4 semitones (minor and major third) over those of 1 or 2. Relaxing this constraint could optimize the average inter-key distance from 6 to 4 keys, an improvement of 33%.

Hand alternation: We could increase the practice on hand alternation. We showed that hand alternation after a chord press can halve IKI. Alas, we neglected to deliberately train this and hand alternation was only performed for ca. 40% of chords. Assuming they cover 50% of text, training to alternate hands before and after each chord, would *halve* IKI of 44% of the keystrokes. This would increase performance by up to 10%.

Training: After 140 hours of training, the performance in Study II was still strongly increasing. By excluding the worst 5% of keystrokes, we can estimate expert performance to reach a rate of over 100 wpm.

Given those improvements, we can make an informed guess on the upper performance achievable with piano-based typing. If we assume that those factors can be independently designed in a single mapping, we can simply add up the gains and estimate performance to reach a level of 130–160 wpm.

Implications for Other Text Entry Methods

Some of our findings may be used to inform the design of traditional text entry methods:

1. *Stylus-based text entry:* Adding redundant keys to create letter clusters as on PianoText might similarly decrease hand-travel distance. The question remains how the findings of the piano transfer to higher dimensional keyboards.
2. *Two-thumb typing:* A full alphabet for each of the two thumbs allows for constant hand alternation. In this case, the model used in [23], (Eq. 5-6) predicts a performance increase of 20%. However, 52 keys will be difficult to implement on a smaller touch display. Future work could investigate the trade-off between number of keys and performance gain and look at the use of chords with two thumbs.
3. *Physical keyboards:* Chords can be implemented on top of single letter input. This combination was shown to improve the performance of PianoText by over 30%. The possibility of pressing a chord in different octaves contributed significantly to this high rate, allowing for hand alternation at all times. By splitting the keyboard as for the *split*-mapping, this could be leveraged on the Qwerty keyboard as well.

CONCLUSION AND OUTLOOK

This paper presents a proof-of-concept for piano-based typing. The results are promising and invite more research to better understand if it may eventually provide a serious alternative to regular text entry methods. Many properties are favorable: the technique is portable and allows a high level of performance that may scale up better with practice than standard methods. We see several opportunities for future work.

Form factor: MIDI-enabled pianos are widely available in different dimensions, but the keys of smaller versions do not offer the same haptic feedback as the graded hammer action keys of a grand piano. Future work should investigate the effect of number of keys on performance and its limitations. However, chords were shown to be accurate and efficient on touch displays [16]. Piano implementations on touchscreen devices might benefit from the expansion of chord usage.

Non-pianists and musical structures: The benefits of redundancy and chords generalize to all users, but a longer training might be needed to practice the complex movements on the piano.On the other hand, the mapping could take even better advantage of the high-level structures of piano music that allow skilled pianists to perform without visual feedback. A mapping with a higher level of redundancy or a dynamic one may be needed in order to recreate those structures. It is unclear if such a complex mapping would be learnable.

Stroke velocity: Other features of the piano could be used for input. For example the stroke velocity, exactly controlled by pianists, could differ between capital and lower case letters.

Audio feedback: The impact of sound feedback is still unclear. Crucial in music performance, its value in user interfaces was found ambivalent. Being beneficial in mobile text entry [5] it showed no effect on performance in piano-based typing or gesture interfaces [2]. More elaborate studies should further analyze the effect of sound feedback.

Redundancy: The design principle is prevalent in many user interfaces. Links are replicated in hypertext and a document can be saved via a toolbar, hot-key or regular menu. However, it is rarely exploited in the design of text entry methods. Future work should explore its benefits, as well as possible drawbacks due to cognitive or motor overhead.

Those improvements might turn the traditional musical instrument into an exciting contender to the regular Qwerty keyboard and inspire the design of radically new text entry methods.

ACKNOWLEDGMENT

We thank M. Scholtes, F. Antonicelli and K. Vertanen. This research was funded by Max Planck Institute for Informatics and the Cluster of Excellence on Multimodal Computing and Interaction. The PianoText software, usable with any MIDI-capable piano, is available at `www.annafeit.de/pianotext`.

REFERENCES

1. Adler, M. *Antique Typewriters, from Creed to QWERTY*. Schiffer Pub., 1997.
2. Andersen, T. H., and Zhai, S. writing with music: Exploring the use of auditory feedback in gesture interfaces. *TAP 7*, 3 (2010).
3. Beddoes, M. P., and Hu, Z. A chord stenograph keyboard: a possible solution to the learning problem in stenography. *IEEE Transactions on Systems, Man and Cybernetics 24*, 7 (1994), 953–960.
4. Bowers, B. Electrical engineering 100 years ago. *British Journal of Audiology 13*, S2 (1979), 1–4.
5. Brewster, S. Overcoming the lack of screen space on mobile computers. *Personal and Ubiquitous Computing*, 3 (2002).
6. Buxton, B. Chord keyboards. `http://www.billbuxton.com/input06.ChordKeyboards.pdf`, 2012.
7. Deutsch, D. Pitch circularity. `deutsch.ucsd.edu`.
8. Dunlop, M., and Levine, J. Multidimensional pareto optimization of touchscreen keyboards for speed, familiarity and improved spell checking.
9. Engelbart, D. C. Design considerations for knowledge workshop terminals. In *Proc.1973, national computer conference and exposition*, ACM (1973), 221–227.
10. Ericsson, A. K. *The Cambridge handbook of expertise and expert performance, Chapter 26: Music*. Cambridge University Press, 2006.
11. Ericsson, K. A., and Williams, A. M. Capturing naturally occurring superior performance in the laboratory: translational research on expert performance. *Journal of Experimental Psychology: Applied 13*, 3 (2007), 115.
12. Findlater, L., Wobbrock, J. O., and Wigdor, D. Typing on flat glass: examining ten-finger expert typing patterns on touch surfaces. In *Proc. CHI'11*, ACM (2011), 2453–2462.
13. Furuya, S., and Soechting, J. F. Role of auditory feedback in the control of successive keystrokes during piano playing. *Experimental Brain Research*, 2 (2010), 223–237.
14. Gentner, D. Expertise in typewriting. Tech. rep., DTIC Document, 1984.
15. Gopher, D., and Raij, D. Typing with a two-hand chord keyboard: will the qwerty become obsolete? *IEEE Transactions on Systems, Man and Cybernetics*, 4 (1988), 601–609.
16. Lepinski, G. J., Grossman, T., and Fitzmaurice, G. The design and evaluation of multitouch marking menus. In *Proc. CHI'10*, ACM (2011), 2233–2242.
17. Lifton, J., and Knight, M. Plover, the open source steno program. `http://plover.stenoknight.com/`, 2012.
18. Lyons, K. e. a. Experimental evaluations of the twiddler one-handed chording mobile keyboard. *Human-Computer Interaction 21*, 4 (2006).
19. MacKenzie, I., and Zhang, S. The design and evaluation of a high-performance soft keyboard. In *Proc. CHI'99*, ACM (2011), 25–31.
20. MacKenzie, S. I., and Soukoreff, W. R. Text entry for mobile computing: Models and methods, theory and practice. *Human–Computer Interaction 17*, 2-3 (2002), 147–198.
21. MacKenzie, S. I., T.-I. K. *Text entry systems: Mobility, accessibility, universality*. Morgan Kaufmann, 2010.
22. Norman, D. A., and Fisher, D. Why alphabetic keyboards are not easy to use: Keyboard layout doesn't much matter. *Human Factors*, 5 (1982).
23. Oulasvirta, e. a. Improving two-thumb text entry on touchscreen devices. In *Proc. CHI'13*, 2765–2774.
24. Rumelhart, D. E., and Norman, D. A. Simulating a skilled typist: A study of skilled cognitive-motor performance. *Cognitive Science*, 1 (1982), 1–36.
25. Schmidt, R. A., and Lee, T. *Motor Control and Learning, 5E*. Human kinetics, 1988.
26. Schmuckler, M. A., and Bosman, E. L. Interkey timing in piano performance and typing. *Canadian Journal of Experimental Psychology*, 2 (1997), 99.
27. Schneider, W. Training high-performance skills: Fallacies and guidelines. *Human Factors*, 3 (1985), 285–300.
28. Seibel, R. Data entry through chord, parallel entry devices. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 2 (1964), 189–192.
29. Vertanen, K., and Kristensson, P. A versatile dataset for text entry evaluations based on genuine mobile emails. In *Proc. MobileHCI'11*, ACM (2011), 295–298.
30. Zhai, S., Hunter, M., and Smith, B. Performance optimization of virtual keyboards. *Human–Computer Interaction*, 2-3 (2002), 229–269.

Appendix A:

Our goal is to assign the letters of the English language to the keys of the piano keyboard such that: (1) the pianist can use the same note transitions for entering text as when playing music, (2) dissonant and infrequent musical intervals are avoided and (3) frequent letter pairs (bigrams) are close together and can be played within one hand.

We formulate this problem in terms of placing items on a 1-D grid and use a greedy algorithm with additional constraints to create the letter-to-key mapping. To match the requirements we make use of the fact that there are more keys on the piano than letters in the alphabet and allow for a *one-to-many* mapping. This means that one letter can be mapped to multiple piano keys (the *redundancy* concept). The algorithm *constructs* the mapping from scratch, by assigning letters to keys one after another, following three steps:

1. **Map by frequency:** Map letters to keys in the order given by the frequency distributions of language and music.
2. **Check interval:** After each mapped letter, check if frequent bigrams containing this letter are mapped to note transitions common in music. If this is not the case, remap the letter to the next most frequent note and check again.
3. **Check distance:** After mapping all letters once, check the inter-key distance of frequent letter pairs. If the distance is too large, reduce it by mapping the more frequent letter to an additional key.

The following pseudocode describes the implementation of these principles.

Algorithm 1 FIND MAPPING

INPUT: 1- and 2-gram distributions of music and language
OUTPUT: a mapping of each letter to a set of notes

```
1: procedure FindMapping(orderedLetters, orderedNotes)
2:     for all  l in orderedLetters do
3:         mapping(l) := next(orderedNotes);                ▷ Step (1)
4:         while !(checkInterval) do                        ▷ Step (2)
5:             replace assignment with next most frequent note
6:         end while
7:     end for
8:     for all frequent bigrams do
9:         if !(checkDistance) then                         ▷ Step (3)
10:            assign additional note
11:        end if
12:    end for
13: end procedure
```

Let us go through the algorithm step by step with help of examples. There are two main loops. The first one goes through all letters in the order of their frequency and assigns each to one piano key. Let us assume we are in the 7. iteration. The mapping so far would look like this:

The next to-be-assigned letter, according to its frequency, is the letter ‘s’. In line 3 it is assigned to the next most frequent note of the corpus, which is C4:

The while loop in line 4 goes through all frequent letter pairs consisting of ‘s’ and the letters mapped so far: ‘es’, ‘as’, ‘st’, ‘is’, ‘se’, ‘so’, ‘si’, ‘ts’, ‘ss’, ‘ns’, ‘os’, ‘us’. For each of the pairs, the interval of the corresponding notes is considered and checked for its frequency in the music corpus. For example ‘es’ corresponds to the note transition from A4 to C4. They form the interval of an augmented sixth, frequently used in music. If one of the letter pairs corresponds to an interval infrequent in music, for example a dissonant interval, the algorithm will try to reassign the letter to the next most frequent note. This would be done in line 5. However, for the letter ‘s’ all intervals are harmonic and frequent. Thus C4 is a valid assignment for the letter ‘s’ and the algorithm can go on to the next iteration. In this way, at the end of the first for loop (line 7), all letters are assigned to *one* note.

The second loop (line 8) considers *all* frequent bigrams in the order of their frequency. The corresponding keys are checked for their distance. If the interval is larger than 8 semitones (a perfect fifth), the more frequent letter is mapped to an additional piano key, such that a smaller interval is formed (Step 3). The additional key is chosen such that it minimizes the interval to the other letter while maximizing the distance to the redundant key. Let us consider the following mapping, where every letter is mapped only once:

To clarify the function of step 3, we will go through two iterations of the second for loop. The most frequent bigram is ‘th’. It is considered first. The corresponding notes, G4 and D4, have a distance of 7 semitones. D is the dominant in the G major scale and the fifth, formed by those notes, is a very frequent interval in music. Thus *checkDistance* in line 9 returns true and we can continue with the next letter pair. In the eighth iteration, the algorithm considers the bigram ‘ed’. The corresponding notes, A4 and D3, have a distance of 20 semitones, exceeding the distance limit of 7 semitones. Thus we go on to line 10 and assign an additional key to the more frequent letter, the ‘e’. To maximize the benefit, it is mapped to C#3, which is far away from the other key assigned to ‘e’, while minimizing the distance to the ‘d’:

In this way a mapping from single notes to single letters is constructed.

However, to generate the *split*-mapping that is implemented in PianoText, we needed to apply the algorithm twice. The idea was to ensure that most of the

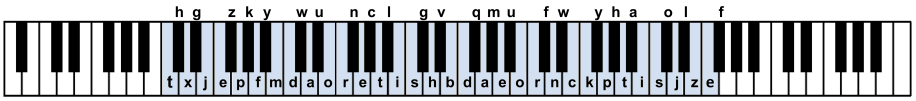

Figure 1: The mapping from letters to notes, as optimized by the algorithm.

keys are mapped at least twice, once on the right and once on the left side of the piano, such that it can be easily entered by each hand. Therefore the algorithm was first applied to the right part of the piano, starting from A3. Additionally only the most frequent bigrams were considered, occurring with a frequency greater 0.008 in the corpus. This ensures that the stronger right hand can reach all letters and most frequent bigrams. In the second run, the whole piano was considered as well as less frequent bigrams (frequency of 0.0008). In this way, nearly all letters were assigned an additional key on the left part of the piano. Only the letters in the range of A3 to E4, reachable by both hands, were not necessarily mapped twice. The outcome of this procedure can be seen in Figure 2. Note that the mapping does not utilize all the 88 keys of the piano. The ones on the far left and far right are used very infrequently in our music corpus and are thus excluded from the mapping. Very frequent letters, such as the 'e', are mapped three or four times, at least once on each side of the piano. More infrequent letters, such as 'b' or 'v' are mapped only once, but in the middle of the piano and thus reachable by each hand.

Figure **??** shows the mapping from n-grams to chords as used in Study 2. The n-grams where chosen in such a way that they optimize the performance benefit of the chord. The more frequent and longer an n-gram, the higher the gain for entering it with only one stroke. In addition we favored those n-grams that were particularly slow to enter with single key presses, such as 'take' or 'yes'. We estimated the gain of a chord mapped to a certain n-gram with the following equation:

$$G(\textit{chord}, \textit{n-gram}) = (T_{\textit{n-gram}} - T_{\textit{chord}}) \cdot P(\textit{n-gram}) \quad (1)$$

where $T_{\textit{n-gram}}$ denotes the time to enter a n-gram by single key presses, $T_{\textit{chord}}$ the time to enter a chord. $P(\textit{n-gram})$ denotes the frequency of the n-gram in a given corpus. The time it takes to press a chord was assumed to be constant. To estimate the time for entering a n-gram we looked at the average inter-key intervals from study 1. The frequency of n-grams was taken from `http://www.sketchengine.co.uk/`.

the	G	B	use	#F	D	ate	G	#D
that	E	G	over	D	B	because	E	C
th	G	#A	inter	G	C	yes	E	#G
and	#F	A	but	E	A	make	G	C E
to	#D	G	not	F	D	ly	D	#A
in	B	D	ent	B	G	con	E	G B
ment	F	A	would	#D	C	could	E	#C
with	#A	D	what	C	C	of	#D	B
ing	C	#D	by	D	D	take	C	E G
this	C	E	will	F	F	ght	B	E G
which	A	C	other	A	A	how	C	E A
there	C	G	ter	B	B	about	#F	B
up	D	F	nes	G	E	when	E	A C
where	F	C	one	C	#G	ted	B	E
pl	#C	E	from	B	E #G	like	F	#A
him	D	G	ver	#A	D G	her	#A	F
have	E	B	between	C	A	just	B	#F
able	D	A	these	G	B D	good	E	G C
for	G	D	tion	F	#A D	many	A	D

Figure 2: The mapping from n-grams to chords as used in study 2. The letters after each n-gram describe the notes that form one chord. They can be played in any octave.

Appendix B:

We trained a hobby-pianist over 140 hours to memorize the mapping, generate text and transcribe given sentences. The training spanned 25 weeks with on average of 6 hours per week. A large number of exercises was created focusing on different aspects of typing, piano playing, chords and redundancy. The participant trained on her own during the week, following the instructions given by the training software, as shown in Figure 1. Once a week we met with the participant to conduct a controlled performance test. This meeting was also used to give feedback on finger movements and typing manners and to discuss new exercises.

The first three weeks were dedicated to memorize the letter mapping. Chords were introduced in blocks of 4–10 per week, starting within week 7. They accompanied other exercises for gaining speed when typing with single letters. Based on the performance in previous weeks, we created special exercises that focused on statistically slow letter sequences, and increased the complexity of phrases. We introduced practices for large lateral movements (jumps within one hand), the combination of both hands on different parts of the piano and the spontaneous reaction on infrequent letter pairs. It was taken care that basic exercises, like frequent bigrams, chord and single letter mappings, were repeated with an expanding schedule. In addition to the training software, the participant was told to practice her skills in online typing games. Table 1 shows the exercise schedule and example stimuli. Sentences were taken from the Enron email data set [3] and websites for training english as a second language [1, 2].

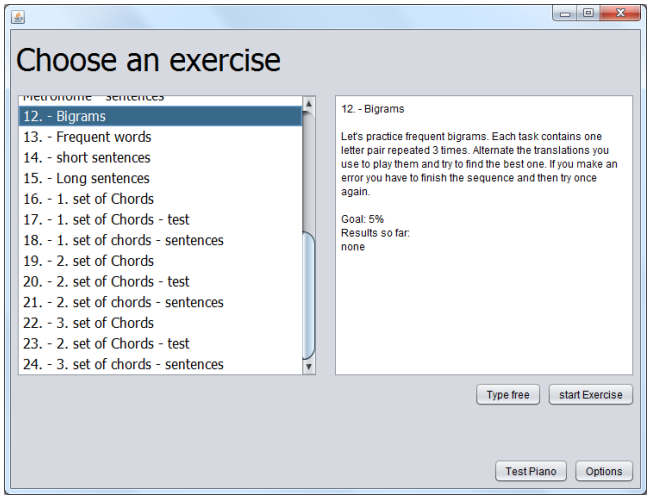

Figure 1: The main screen of the training software. The list on the left displays the available exercises. The right window shows the description of the chosen exercise and previous performances

Weeks	Exercise	Stimulus Example	Goal
1– 3	Letter-to-key mapping	see Figure 2	Memorize letter positions
from 3	Frequent bigrams and words	*th in es st an* *they can give only want*	Resolve multiplicity, memorize best translation
	short sentences	*if he wants it* *i am on my way* *i am on a plane [3]*	speed up
7–14	chord-mapping	see Figure 3	memorize chords that enter a sequence of letters
from 7	sentences containing n-grams mapped to chords	*i am not aware of any* *joel will handle this* *what a nice note [3]*	detect chords in letter sequences, learn to use them in combination with single key presses
from 8	for all exercises: try to type without looking at the hands		increase performance by avoiding constant shift of visual attention
10	public performance	sentences requested by audience	free text generation and speed up
from 11	longer sentences		speed up
from 12	focus on slow letter sequences	*incredible complicated reservations somewhere*	speed up
from 13	play random letter sequences on a particular part of the piano	*edxpheth geedet heweth thad etamte*	practice hand combination on different parts of piano
	short texts	see Figure 4	speed up
from 14	practice words with high error rate		decrease error rate
	metronome: type sentences to a beat, successively increase beat	see Figure 5	speed up and decrease error rate
from 15	inner hearing: imagine the sound of a word before typing it		improve performance by making use of the audio feedback
	jump: practice words that require larger lateral arm movements		speed up arm movements and reduce the need for visual attention
from 17	sentences with many chords	*what would you do* *but what about her* *see Figure 6*	improve inter-chord interval
	German sentences	sentences from [3] translated into german such as: *ich werde dich morgen nacht anrufen wenn ich zurueck komme*	quickly resolve multiplicity for unpracticed letter sequences, spontaneously decide for best translation
from 22	practice hand-alternation before and after chord usage	*there was a lot going on between now and then* *i would like to take care of you and your sister*	improve speed for chords

Table 1: The exercise program for learning fast text entry with PianoText.

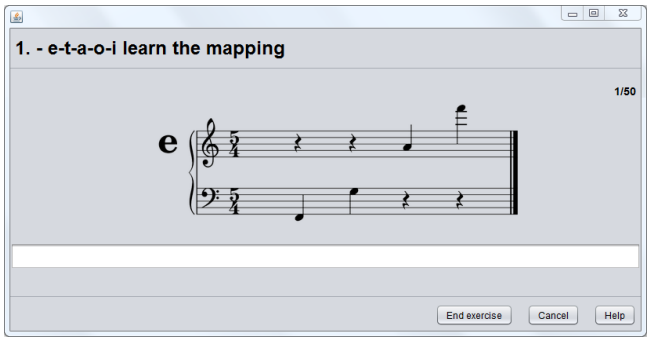

Figure 2: Very first exercise that trains the user to memorize the mapping. The notes assigned to each letter are presented and the user has to press each one of them.

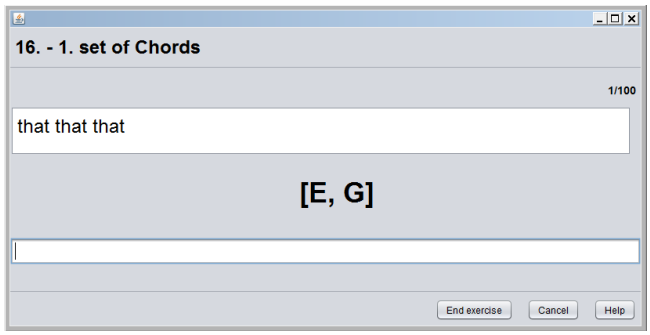

Figure 3: Exercise that introduces the chord mapping. The note names are shown in the bottom. The user has to enter the given n-gram three times by pressing the chord in different octaves.

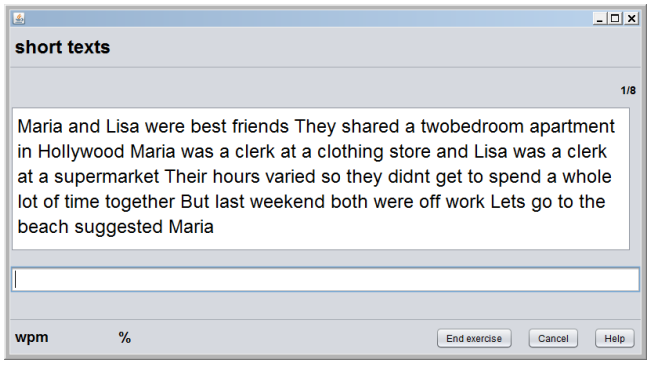

Figure 4: In this exercise the user has to write a small text. This improves performance by practicing the spontaneous input of text. The texts are taken from [2]

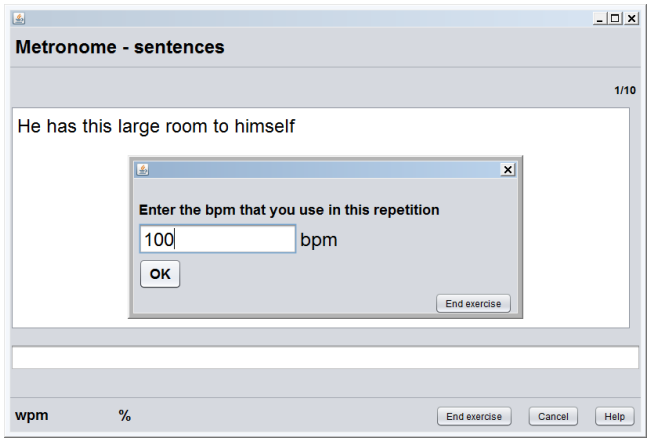

Figure 5: A metronome exercise. The user has to enter the beats per minutes. Regular tones then indicate the speed at which the keys have to be pressed.

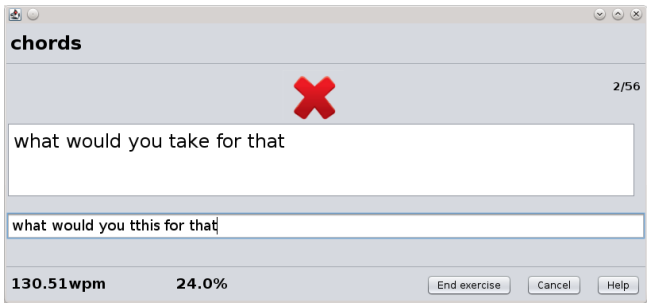

Figure 6: The sentence in this exercise contains 5 n-grams that can be entered by a chord: 'what', 'would', 'take' and 'that'. The goal is to decrease the time between the key presses. The red cross is shown after entering the last word and indicates that the user made an error.

Exercise sentences from

[1] Kelly, C. A website for studying english as a second language. `www.manythings.org`.

[2] Lee, R. C. English as a second language. `www.rong-chang.com`.

[3] Vertanen, K., and Kristensson, P. A versatile dataset for text entry evaluations based on genuine mobile emails. In *Proc. MobileHCI'11*, ACM (2011), 295–298.