



Using Rhythmic Patterns as an Input Method

Emilien Ghomi Guillaume Faure Stéphane Huot Olivier Chapuis Michel Beaudouin-Lafon
ghomi@lri.fr gfaure@lri.fr huot@lri.fr chapuis@lri.fr mbl@lri.fr

Univ Paris-Sud (LRI)
F-91405 Orsay, France

CNRS (LRI)
F-91405 Orsay, France

INRIA
F-91405 Orsay, France

ABSTRACT

While interaction techniques that use the temporal dimension have been used for a long time, such as multiple clicks or spring-loaded widgets, more advanced uses of *rhythmic patterns* have received little attention in HCI. Using such temporal structures to convey information can be particularly useful in situations where the visual channel is overloaded or even not available. In this paper we introduce *Rhythmic Interaction* as the use of rhythms for input. We report the results of two experiments that show that (i) rhythmic patterns can be efficiently reproduced by novice users and recognized by computer algorithms, and (ii) rhythmic patterns can be memorized as efficiently as traditional shortcuts when associating them with visual commands. Overall, these results demonstrate the potential of Rhythmic Interaction and open the way to a richer repertoire of interaction techniques.

ACM Classification Keywords

H.5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

Author Keywords

Rhythm; tapping; hotkeys; learning; patterns; Morse code

INTRODUCTION

Rhythm plays an important role in our everyday life. Temporal patterns are of course critical in experiencing music, but they also underlie periodic actions such as walking, breathing or chewing, and they are even necessary for understanding speech prosody. Rhythm is so deeply embedded in our experience of living that it can be used to cure some diseases such as stress or sleep disorders [29].

While perceiving and reproducing rhythm is recognized as a fundamental human ability by physiologists and neuropsychologists, it is still underused as an interactive dimension in HCI. In common desktop environments, interaction relies heavily on manipulating graphical widgets, simple mouse clicks and keyboard shortcuts. This basic vocabulary, however, is often extended by using *spatial* or *temporal* features, as with mouse gestures or multiple clicks.

Although the spatial dimension has been the focus of much HCI research on interaction techniques based on hand postures or gestures, e.g. [1, 3, 16], the temporal dimension has received little attention so far. We propose to use rhythm as an input method and introduce *Rhythmic Interaction* as a complementary way to control interactive systems. Rhythmic Interaction can be used in any event-driven environment for a variety of input modalities: clicking the mouse, hitting keyboard keys or a touch-sensitive surface, moving a motion-sensing device, etc. However, it has competitive advantages for tactile screens, since it requires less screen space than gestural interaction and no visual attention [33]. This article presents a first exploration of the design space of Rhythmic Interaction in order to address the following questions:

- *Feasibility.* Even if perceiving and performing rhythm is quite natural, are users able to reproduce, learn and memorize patterns? Can they use them to trigger commands?
- *Interaction design.* The number of possible rhythmic patterns is virtually infinite and they can be presented in several ways. Which patterns make sense for interaction and how to design a vocabulary? What feedback helps executing and learning patterns?
- *Technical issues & Integration.* Like most continuous high-level input methods, e.g. voice, marks, gestures, Rhythmic Interaction relies on a recognizer to segment and interpret user input. How to design effective recognizers that do not require training?

In the rest of this paper, we survey related work and then define a framework for Rhythmic Interaction, narrowing the scope of our study to vocabularies of rhythmic patterns that are relevant in the context of HCI. Then, we report on two experiments where the patterns are rhythmic sequences of taps performed on a tactile trackpad to trigger commands. The first one tests the ability of novice users to reproduce individual patterns, while the second one compares the ability of users to memorize the association of commands to rhythmic patterns vs. keyboard shortcuts. We also describe the recognizers that we created for these two experiments, and draw some conclusions regarding the design of pattern vocabularies and appropriate feedback.

BACKGROUND & RELATED WORK

The literature in cognitive science has studied the perception, reproduction and use of rhythm from several perspectives: physiology, e.g., perception and action [13, 19], knowledge and learning, e.g., language [24], artistic applications, e.g., music [23], etc. Two major studies on the psychology

E. Ghomi, G. Faure, S. Huot, O. Chapuis, and Michel Beaudouin-Lafon. Using Rhythmic Patterns as an Input Method. In CHI '12: Proceedings of the SIGCHI Conference on Human Factors and Computing Systems, 10 pages, to appear, ACM, 2012.

© ACM, 2012. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version will be published in *CHI'12*, May 5–10, 2012, Austin, Texas, USA.

of rhythm in music are reported by Fraisse [12] and by Clarke [7]. In this section, we give an overview of the literature in cognitive science that is most relevant to Rhythmic Interaction and then focus on the few studies of using time and rhythm to control interactive systems.

Rhythm in Music and Games

For musicians, rhythm is one of the most important features in music, together with melody and harmony. Many traditional forms of music combine simple rhythmic structures played in parallel to build up higher level aggregated rhythms, called “polyrhythms” [2]. From a cognitive point of view, these practices suggest that the cognitive load required to deal with an elementary rhythmic pattern is light enough to allow their combination in more complex structures.

Highly trained musicians are able to create and play an incredible amount of rhythmic variations. Simple patterns, however, can be reproduced by everyone, as illustrated by the success of popular musical games such as Guitar Hero, TapTap or Donkey Kong, where rhythmic structures are recognized and reproduced by non-musicians of all ages.

Rhythm in Cognitive Sciences

In physiology and neuropsychology, numerous studies report that humans have a natural perception of rhythm, thanks to rhythmic mechanisms that are involved in the internal functioning of our organism (heart beat, sleep cycles, etc.) [13] and their relation with periodic external phenomena (day/night, seasons, etc.). When listening to music, we constantly try to infer the beat, i.e., to perceive regular and salient events, or to group events into rhythms [18]. In fact, we systematically try to perceive rhythm even when none is present [25] or when being told to avoid it [12].

Rhythm perception is deeply related to the motor system [5]. Since childhood, humans are used to tap their feet, clap their hands, snap their fingers and move in synchrony with music. These activities are common and seem simple, even though they involve complex rhythmic structures. Outside music, periodic activities of different frequencies are pervasive in our everyday life. For example, chewing or walking are known to have universally preferred rates [13, 19].

While these studies attempt to explain how and why we perceive and produce periodicities, they rarely deal with the reproduction and memorization of rhythmic patterns associated to tasks that we address in this article.

Rhythm as an Input Method in HCI

Rhythm is built on the temporal dimension, which is commonly used in interactive software. For example, long clicks are often distinguished from short clicks to trigger different commands based on temporal criteria. The concept of “dwelling”—freezing the interaction for a short amount of time—is also used to segment gestural interaction [15] or to explicitly switch mode [10]. Rhythmic Menus [22] successively highlight items at a given rate while the mouse button is pressed. When the user releases the button, the current item is selected.

Some techniques are based on the temporal grouping of events. Double click is the simplest and most common case, but some studies also explored rhythmic motion: Motion Pointing [11] assigns different periodic motions to graphical objects in a scene or items in a pie menu; The user selects the object or menu item of interest by performing the corresponding motion. In Cyclostar [20], the user controls continuous parameters, such as the speed of zooming, by performing elliptical oscillatory gestures. The rate of the circling motion controls a parameter of the resulting command.

In the above cases, rhythmic aspects are reduced to periodicity. To the best of our knowledge, only a few techniques involve the reproduction of rhythmic patterns. Five-key [31] is a text entry system based on rhythmic sequences, where letters can be entered with only five keys. However, efficiency and learning were not studied systematically. In [9], tempo reproduction is used to select a particular song in a music library by tapping on a mobile device or shaking it. But relying only on tempo raises some scalability issues that were not assessed. Finally, Tapsongs [33] is an alternative to textual passwords where users tap a rhythmic pattern that they have registered with the system for authentication.

MOTIVATION

Our goals are more general than Five-key and Tapsongs: we want to design vocabularies of rhythmic patterns that users can learn easily and perform reliably in order to trigger commands. This approach is somewhat similar to the use of Morse code for encoding characters. However, the design of Morse code was driven by information theoretic issues rather than usability, and while early computers were able to decode human-produced Morse code [4], it has rarely been used in HCI [6]. Our objective is to propose a comprehensive framework to design rhythmic patterns for interaction, with efficient recognizers that do not need training.

Advantages of Using Rhythms for Input

The design of new techniques based on Rhythmic Interaction is not the main focus of the present article. However, we have identified a number of potential advantages of using rhythm to interact with computer systems. First, as evidenced by research in Cognitive Science, there is a direct correspondence between performing a rhythm (action) and listening to a rhythm (potential audio stimulus and feedback). Second, rhythms can be performed in a variety of situations: while performing a rhythm requires as little as a single degree of freedom of one finger, many movements can be performed rhythmically and captured using different sensors, e.g., tapping fingers, tapping feet, or nodding the head.

Gestural interaction typically requires space to perform the gestures, and often interferes with the display space on a small touchscreen. By contrast, Rhythmic Interaction only uses temporal features. Rhythms can be performed on a small area of a tactile device, even in an eye-free context.

Finally, rhythmic structures can be designed in a hierarchical way. By using common prefixes among different patterns, a natural hierarchy emerges that can be internalized by users, facilitating memorization and recall.

Applications of Rhythmic Patterns

Rhythmic patterns are not meant to replace more conventional command input methods. Instead, it is an alternative that may be more adapted to specific situations, such as eye-free operation. It is also a way to enhance existing methods with a richer vocabulary. For example rhythmic patterns could give access to a restricted set of commands such as speed-dialing a phone number, navigating an e-book or switching mode in an application.

In some situations, rhythmic patterns can simplify interaction. For example, bookmarks, menu items or contacts are often organized hierarchically. Rhythmic patterns could match this hierarchy or provide an alternate hierarchy such as organizing contacts by their first name. Also, since rhythmic patterns can be performed without visual attention, they can be used with a tactile device in the pocket or while driving, or in the dark, e.g. to shut down an alarm clock, or even with devices that do not have a display.

Rhythmic Interaction also offers novel solutions to well-known problems. Tapping on the back of a hand-held device can be captured without extra sensors, thanks to built-in accelerometers or microphones [28]. For example, a rhythmic pattern performed while receiving a phone call could add the caller to the contact list, or display extra information such as battery life or signal level. Patterns could also be performed with the non-dominant hand or another part of the body such as the feet [30], to switch mode or ignore an incoming call.

RHYTHMIC PATTERNS FOR INTERACTION

Our definition of a rhythmic pattern comes from music: The elementary structure in music is called a *motif*, which is defined as a “melodic, rhythmic, or harmonic cell” [21]. A *rhythmic motif* represents the temporal structure of a set of notes and consists of the relative durations of notes and silences. Notes and silences can have eight different durations in standard musical pieces, and motifs can contain many notes, leading to a huge number of possible rhythmic motifs.

Considering the number of commands and actions often used when interacting with computers, such an expressive power is not required. Therefore we propose a restricted definition of *rhythmic pattern* (or simply *pattern*) more adapted to HCI. A rhythmic pattern is a sequence of taps¹ and breaks whose durations are counted in *beats*. The beat is the basic unit of time and its duration is defined below). We define the complete set of possible patterns with the following rules:

- Taps can be of three types: *impulse* (a hit on a touch device or a click), *short* tap (one beat) or *long* tap (two beats). A tap starts at the beginning of a beat, and there cannot be more than one tap per beat.
- Breaks can be of two types: *short* (one beat) or *long* (two beats). A pattern cannot begin or end with a break, and there cannot be two successive breaks.

This definition of taps and breaks is based on our empirical observation that computer users are familiar with the distinction between instantaneous and long clicks or taps. By adding

¹The word “tap” reflects our focus on using a tactile device for input.

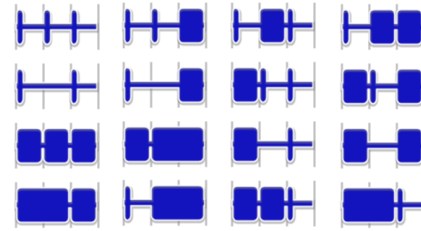


Figure 1. The 16 three-beat patterns defined by our rules. Each rectangle represents a tap. The thin gray lines show the beats.

a third duration and by taking breaks into account, we offer designers more possibilities for selecting a set of patterns among the possible combinations. In comparison to Morse code, we do not need the “intra-character”, “inter-character” and “inter-word” breaks that are specific to the coding of language, and we do not allow more than one tap per beat.

The *length* of a pattern is the sum of the durations of its taps and breaks. To simplify reproduction and memorization, we focus on patterns between two and six beats long. The rules above define 5 two-beat patterns, 16 three-beat patterns (Figure 1), 53 four-beat patterns, 171 six-beat patterns and 554 six-beat patterns. By comparison, the total number of patterns with n taps is 3^{2n-1} , i.e. 199,290 patterns with up to six taps.

In this entire study, beats occur at the tempo of 120 BPM (2Hz). Thus, the onsets of two consecutive taps are separated by at least 500 ms, i.e. a beat is half a second. This corresponds to a common tempo of human motor actions, e.g. walking [19], and of contemporary music [23].

As a first step, we only consider rhythmic patterns performed by tapping on a touch-sensitive surface. While keyboards, accelerometers [17, 9] or eye blinks [32] can probably be used for Rhythmic Interaction, it is out of the scope of this article. We also do not address the segmentation of patterns from other input. Simple solutions that should be tested include segmenting in time by preceding each pattern with a specific short sequence of taps, or segmenting in space by performing patterns on a specific location of a device.

A key aspect of this research is to design a recognizer that can reliably identify the patterns produced by users. In a first experiment, we used a *structural* recognizer to assess users’ ability to produce patterns accurately. Based on the results, we designed a *pattern classifier* that accounts for user inaccuracies while still discriminating the patterns in the vocabulary. This classifier was used in a second experiment where we assessed users’ ability to memorize associations between patterns and commands in an applicative context.

EXPERIMENT 1: RHYTHMIC PATTERN REPRODUCTION

In order to assess the potential of Rhythmic Interaction, we conducted a first experiment where novice users were asked to replicate patterns presented to them in visual and/or audio form by tapping on a touch surface. The goal was to assess the accuracy of the reproduction and to compare the effects of several feedback mechanisms while performing patterns.

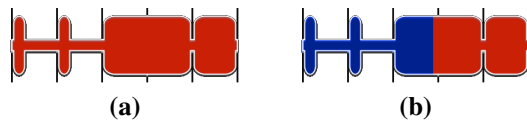


Figure 2. The stimulus in Experiment 1. A static representation is displayed (a) and fills up in synchrony with the audio playback (b).

Recognizer

The recognizer that we designed for this experiment is based on the above rules for defining the patterns. It first extracts the rhythmic structure as a list of taps and breaks and infers their respective types (impulse, short and long) using autonomous heuristics. The reconstructed pattern is then checked against the vocabulary used for the study.

In order to identify the type of every tap and break in the sequence, the recognition algorithm uses k-means clustering iterated 500 times on duration values. The algorithm builds clusters corresponding to the possible types of taps and breaks: impulse, short and long². A minimum distance of 200ms between the duration clusters is enforced, corresponding to a maximum tempo for the pattern to be recognized. If two clusters are closer than that distance, they are merged and will be recognized as a single tap type. Thus, if the pattern is performed too fast, events of different types may be confused by the recognizer. For cluster identification, the reference durations for short and long taps or breaks are set to 500ms and 1000ms respectively, and the maximum duration of an impulse or release is set to 180ms.

After clustering, breaks that correspond to the rest of a beat after an impulse are removed from the reconstructed pattern. The resulting pattern is then looked up in the vocabulary to check if it matches the stimulus provided to the participant.

Note that this recognizer is intentionally very strict, in order to assess the participants' ability to precisely reproduce the patterns. In particular, if the reconstructed pattern is not in the vocabulary, the recognizer will systematically return an error. With minimal knowledge about our definition of rhythmic patterns, the algorithm is able to identify the type of every tap and break in a sequence even in tricky situations, such as when there is just one type of events. Thanks to clustering, the recognizer adapts to the tempo (the overall tempo can be inferred by comparing the centroids of the clusters.)

Apparatus and Participants

The experiment was implemented in Java and conducted on a 13" Apple MacBook (Intel processor). Participants tapped the rhythmic patterns on the embedded multitouch trackpad.

Twelve unpaid volunteers (six female) participated in this experiment, with age ranging from 23 to 53 (mean 29, median 27). Five of them had never practiced music.

Stimulus

The pattern to reproduce is presented to the participant with a stimulus combining a static graphical representation of the pattern, visual animation and audio (Figure 2). The visual

²A break with a duration of an impulse is called a *release*. It occurs between two adjacent taps.



Figure 3. Visual feedback while tapping a pattern.

stimulus is a stationary shape depicting the whole rhythmic pattern, where each rectangle represents an event (Figure 2a). This shape is then progressively filled (Figure 2b) in synchrony with audio playback. Beats are marked with thin gray lines to visualize the durations of events.

For audio playback and animation, impulses last 125ms and the tempo is set to 120 BPM or 2Hz (500ms period). This value is above the "synchronization threshold" [27] for both visual and auditive stimulus, ensuring that participants can perceive and perform it accurately. The audio stimulus is a 440Hz A, played by the General MIDI Instrument "English Horn" and held at a constant sound level. We chose this sound as it is soft enough for the subjects to endure during the experiment, but has clear onset and release.

Feedback

Participants are presented with 4 input *FEEDBACK* conditions while reproducing rhythmic patterns. The *Audio* feedback plays the same sound as the stimulus as long as the participant is touching the surface of the trackpad. The *Visual* feedback is based on the graphical representation of the stimulus. The rectangles representing the events appear dynamically while the subject is tapping on the trackpad (Figure 3). The *AudioVisual* feedback combines the two previous methods, and there is no feedback at all in the *None* condition.

The *Audio*, *Visual* and *AudioVisual* feedback methods are expected to help learning, e.g. in novice mode. Conversely, the *None* condition corresponds to the situation where an expert user is performing patterns in an eyes-free manner without audio feedback.

Vocabulary

For this experiment, we selected 30 rhythmic patterns among the 799 patterns with two to six beats generated by the rules described earlier. This vocabulary (Figure 4) contains 4 two-beats patterns, 8 three-beats patterns, 8 four-beats patterns, 6 five-beats patterns and 4 six-beats patterns. We explicitly featured fewer patterns for the extreme situations (two, five and six beats). Among the patterns with the same duration, we tried to balance the number of events. For example, for the 8 four-beat patterns, 2 contain two events, 3 contain three events and 3 contain four events.

Task

A trial consists in reproducing a rhythmic pattern according to the *FEEDBACK* condition, right after it is presented twice in a row. The participant performs the pattern by tapping on the trackpad with the index finger of her dominant hand. The recognizer then computes the temporal structure of the input and matches it with that of the stimulus. At the end of the trial, the participant is notified about the success or the failure of the match before advancing to the next trial.

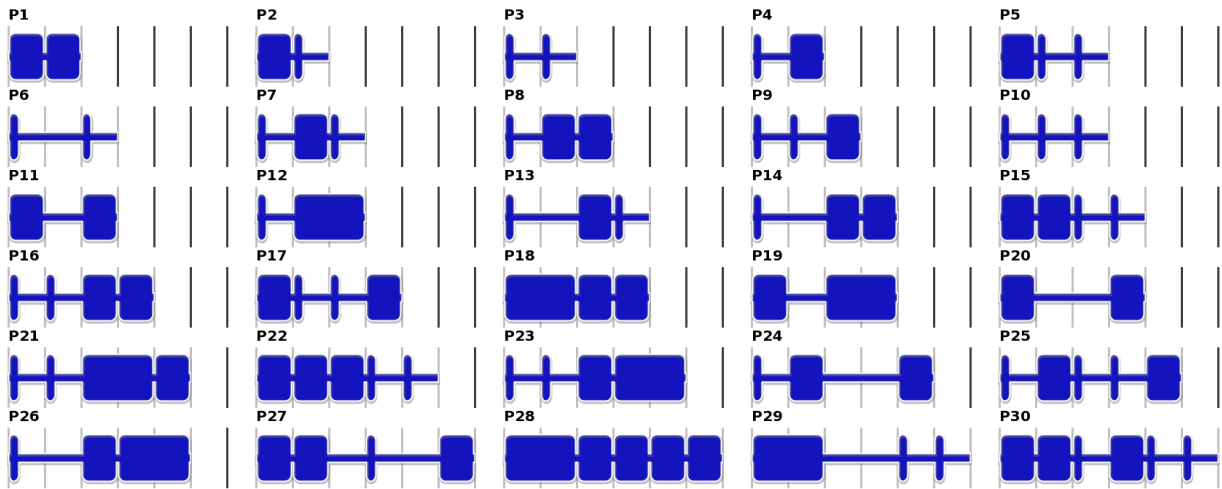


Figure 4. Vocabulary used in the first experiment.

Design and Procedure

The experiment is a 2×30 within-subject design with factors: (i) **FEEDBACK**: *Audio*, *Visual*, *AudioVisual* and *None*; and (ii) **PATTERN**: P1 – P30 (Figure 4).

At the beginning of the session, each **FEEDBACK** condition is introduced to the participant with a short block of 15 random trials. Then, the participant is asked to perform two warm-up blocks of 15 trials in the *AudioVisual* feedback condition, which we hypothesize provides the best feedback to become familiar with the task. The 3 first trials of the first warm-up block are performed by the experimenter to demonstrate the feedback condition to the participant. The second warm-up block is interrupted if the participant reports to be confident enough to start the experiment.

During the main session, measured trials are grouped into blocks according to the **FEEDBACK** factor. The presentation order for **FEEDBACK** is counterbalanced across participants with a Latin square. Within each block, the 30 patterns are repeated twice in randomized order. A practice block of 15 randomly selected patterns is performed prior each measured block and participants are allowed to have breaks between and in the middle of each block. Thus, we collected $12 \text{ participants} \times 4 \text{ FEEDBACK} \times 30 \text{ PATTERN} \times 2 \text{ repetitions} = 2880$ measured trials. Participants were instructed to be as accurate as possible by paying attention to the discrimination of different types of taps and breaks. Each participant took about one hour to complete the sessions, after which they were asked to rank the feedback methods according to the difficulty of the task on a 5-point Likert's scale.

Quantitative Results

The overall success rate is 64.3%. This may seem low, but recall that our recognizer is deliberately very strict regarding the temporal structure of patterns, and that it can recognize all 799 patterns with two to six beats, not just the 30 patterns in the study. The precise reproduction of the rhythmic patterns in the study is similar to playing a percussion instrument, a task that musicians can take years to master.

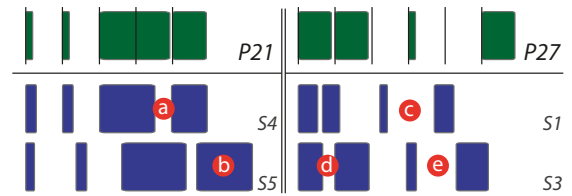


Figure 5. Two stimuli (P21 and P27) with reproductions errors by subjects of Experiment 1. S4: the last break is too long (a); S5: the last tap is too long (b); S1: the last break is too short (c); S3: the first break is too long (d), the last break is too short (e).

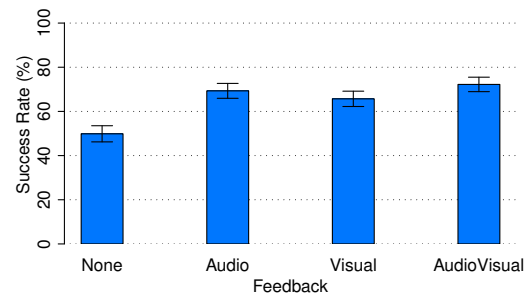


Figure 6. Success rate for each **FEEDBACK** condition.

Figure 5 shows typical reproduction errors by study participants, such as release breaks that are too long and recognized as short breaks, or breaks or taps that are too similar to be separated during clustering. Interestingly, errors seem more frequent with breaks than with taps, which is consistent with the finding that users tend to be more precise when performing notes than pauses [26].

A one-way ANOVA for **FEEDBACK** (with participant as a random variable) reveals a significant effect on success rate ($F_{3,33} = 15.4, p < 0.0001$). This effect can be observed in Figure 6³. Post-hoc t-tests with Bonferroni correction show that the *None* condition is significantly worse than all other feedback conditions. It is not surprising that the absence of feedback while performing the pattern significantly degrades the accuracy of rhythm reproduction.

³In all figures, error bars show the 95% confidence interval.

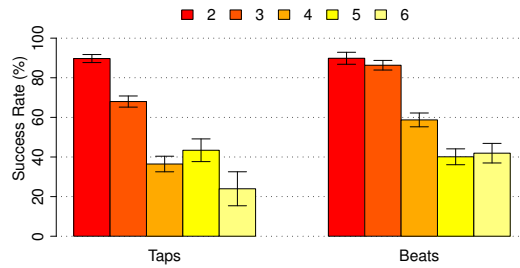


Figure 7. Success rate by number of taps and by length in beats.

Regarding *PATTERN*, a one-way ANOVA reveals a significant effect on success rate ($F_{29,319} = 25.1, p < 0.0001$). We observe a large deviation of the success rate for some patterns: from 16% with P27 to 98% for P10. Fifteen patterns have a success rate of at least 70%: P1–P7, P9–P13, P19, P20, and P29. All have at least 3 taps and all but P29 are less than four-beats long. However, some three-tap patterns have a low success rate (below 50%): P14 and P18 (both with 4 beats).

We could not identify similarities among the patterns that were difficult to reproduce. However, the number of taps and beats are the most obvious characteristics that can influence the ease of reproduction. In fact, we found a significant effect on success rate for taps ($F_{4,44} = 54.1, p < 0.0001$) and beats ($F_{4,44} = 85.5, p < 0.0001$), without significant interaction with *FEEDBACK*. Post-hoc t-tests support this hypothesis since, in most cases, the highest recognition rates were achieved for patterns with a small number of taps or beats (Figure 7).

Qualitative Results

Six participants out of 12 preferred the *Audio* feedback, 3 the *Visual* feedback, 2 the *AudioVisual* feedbacks and 1 no feedback. Moreover, 6 participants ranked *AudioVisual* second and 8 ranked the *None* condition last. Note that many participants pointed out that *AudioVisual* was confusing, providing too much information. They explained that in most cases, they chose one feedback (visual or auditive) and tried to ignore the other. Half of them preferred the *Audio* feedback because it was more related to rhythm than graphics.

We assessed the subjective difficulty of the task with the statement “I found it difficult to reproduce rhythmic patterns”. Seven participants disagreed or strongly disagreed, 4 neither disagreed nor agreed, and only one agreed, but at the same time disagreeing for the *None* and *Visual* feedbacks.

Overall, both quantitative and qualitative results are encouraging and support our hypothesis that rhythmic patterns, as defined by our framework, is a viable input technique for interactive tasks. While quantitative results support the need to provide feedback while performing input, qualitative results inform on the type of appropriate feedback. Finally, an analysis of recognition errors gives insights on how to create a recognizer that would be more suitable for real applications.

A PATTERN CLASSIFIER

The goal of the structural recognizer in Experiment 1 was to assess how accurately participants could reproduce a stimulus pattern. This recognizer is deliberately strict, accounting only for variations in the overall tempo of the pattern, and

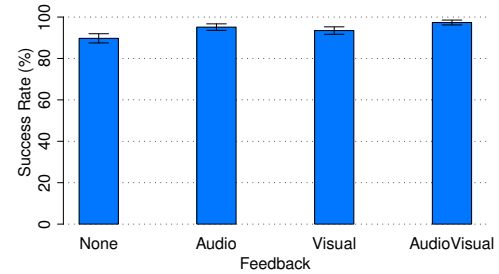


Figure 8. Revised success rate by FEEDBACK for the pattern classifier.

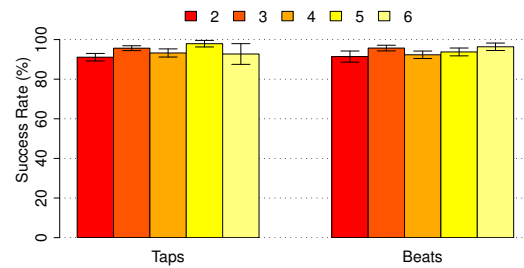


Figure 9. Revised success rate by number taps and length in beats for the pattern classifier.

it does not take advantage of the fact that the input patterns are assumed to be part of a known vocabulary. We designed a second recognizer for use in actual applications, that classifies an input pattern against a vocabulary.

In order to recognize a sequence of taps, this *pattern classifier* first counts the number of taps in the sequence and considers the subset of the vocabulary with that number of taps. Then, it calculates a score for each candidate pattern. First, it infers the duration of a beat by considering the duration of the sequence of taps and the number of taps of the candidate. Using this value, it scales the pattern to match the duration of the input sequence and sums the temporal differences of events onsets and durations. A duration of a quarter beat is used for impulses and releases between consecutive events (when lifting the finger from the device). Finally, the score is weighted by the ratio between the inferred beat duration and the 120 BPM reference (500ms).

This classifier is less strict than the structural recognizer because it will always match an input pattern to a pattern in the vocabulary if it is the only one with the same number of taps, unless a threshold is set on the lowest acceptable score. Moreover, normalization makes the recognizer match patterns that are homothetic of each other. This is the reason for weighing the score by the relative beat durations.

We tested this classifier with the data and vocabulary of Experiment 1. The overall success rate rose to 93.9%, more in line with the expectations of an applicative context. As with the previous recognizer, a one-way ANOVA for *FEEDBACK* reveals a significant effect on success rate ($F_{3,33} = 7.2, p = 0.0007$) (Figure 8).

Figure 9 shows that unlike the structural recognizer, success rate does not decrease with pattern “complexity”: there is no significant effect of the number of taps or the length on





























CMD1	CMD2	CMD3	CMD4	CMD5	CMD6	CMD7	CMD8	CMD9	CMD10	CMD11	CMD12	CMD13	CMD14
													
R1 = P20	R2 = P11	R3 = P10	R4 = P9	R5 = P19	R6 = P4	R7 = P3	R8 = P2	R9 = P1	R10 = P29	R11 = P18	R12 = P6	R13 = P28	R14 = P12
													
Ctrl+Y	Shift+H	Ctrl+X	Shift+E	Ctrl+R	Shift+F	Ctrl+N	Shift+B	Ctrl+D	Shift+T	Ctrl+H	Shift+G	Ctrl+A	Shift+W

Figure 10. Commands used in Experiment 2. Pxx refers to the patterns of Experiment 1 (see Figure 4).

success rate⁴. Instead, we observe that success rates are affected by the similarity between patterns: a complex pattern can be recognized quite reliably provided that it is sufficiently different from other patterns with the same number of taps. For example, P30 is the only pattern made of 6 taps in our set, making recognition failure occur only when the subject tapped a wrong number of taps. By contrast, P17 seems to be more “complex” than the “simple” pattern P20 but the former has a 100% success rate and the latter 82%. In fact, the recognizer sometimes confuses P20 with P11. However, a post-hoc t-test with Holm correction reveals no significant difference between patterns for success rates.

In summary, we found that this classifier was well adapted to actual applications. In particular, a designer can create a vocabulary that minimizes the risk of patterns being confused.

EXPERIMENT 2: RHYTHMIC PATTERNS MEMORIZATION

In order to further validate Rhythmic Interaction, we conducted a second experiment to test whether patterns can be memorized and recalled in order to be used as an alternative to standard techniques for triggering commands. We compared rhythmic patterns with standard hotkeys in a “learn and recall” experiment similar to Appert and Zhai’s comparison of gesture shortcuts with hotkeys [1], itself inspired by Grossman et al’s study of hotkeys [14].

Variables

We compare two techniques for triggering commands (T_{ECH} factor): *Hotkey* and *Rhythm*. A third condition, *Free*, lets participants choose the technique they prefer.

Each command C_i is a triplet associating an image I_i , used as a stimulus for this command, and two triggering techniques: a rhythmic pattern R_i and a hotkey K_i . The command set (CMD factor) has 14 commands: C_1, \dots, C_{14} . We chose the images symbolizing the commands in a set of common objects and fruits (Figure 10).

For the rhythmic patterns, we selected 14 patterns of varying complexity from Experiment 1 and randomly assigned each pattern to a command. For the hotkeys, we created combinations of a modifier (*Shift* or *Ctrl*) and a letter. The letters were chosen so that they did not match the first letter of the name of the object representing the command, as in [1]. The goal is to avoid giving an unfair advantage to hotkeys, since there is no similar mnemonic association between rhythmic patterns and command names. Furthermore, the mapping between commands and hotkeys often varies by application and language. Figure 10 shows the resulting assignment.

⁴This could be due to the fact that in the vocabulary, there were few patterns with five taps or beats.

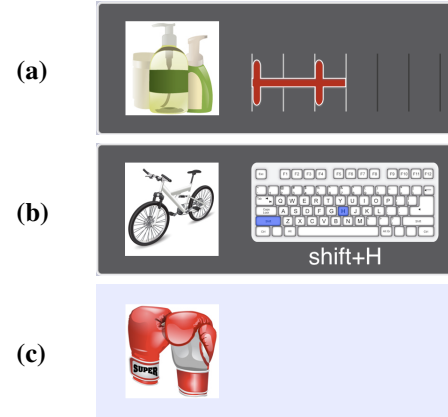


Figure 11. Stimulus in the learning phase for the *Rhythm* (a) and *Hotkey* (b) conditions, and in the testing phase for both conditions (c).

Task

The primary task of the experiment is to activate a command (C_i), presented by its stimulus image (I_i), with the triggering technique corresponding to the current T_{ECH} condition (R_i or K_i). The experiment has two phases: *learning* and *testing*.

During the learning phase, both the image I_i and the corresponding triggering technique (R_i or K_i) are shown to the participant. For rhythmic patterns, the static graphical representation is displayed next to the image (Figure 11a) and the audio stimulus is played twice. Hotkeys are presented with a short animation of the corresponding key-press sequence, also repeated twice, and text (Figure 11b).

In the testing phase, participants are presented with the image I_i only (Figure 11b). According to the current T_{ECH} condition, they must perform the corresponding hotkey K_i or rhythmic pattern R_i . If they forgot which trigger to perform, they are strongly encouraged to invoke a help screen by pressing the *SPACE* key. The task then switches to the learning mode, presenting the shortcut to perform as described above.

In both phases, the participant must perform the rhythmic pattern or the hotkey. For rhythmic patterns, we use the Audio-only feedback since Experiment 1 showed that it was effective and participants preferred it. Also, this avoids interference with the visual interface. For hotkeys, participants receive the usual kinesthetic feedback while pressing keys.

After entering each hotkey or pattern, the participants are asked to indicate which trigger they were trying to perform (Figure 12). Then, participants are notified of the correctness of their answer. If the answer is correct, they are given the result of the recognition. If not, the correct trigger is presented before moving to the next trial. The reason

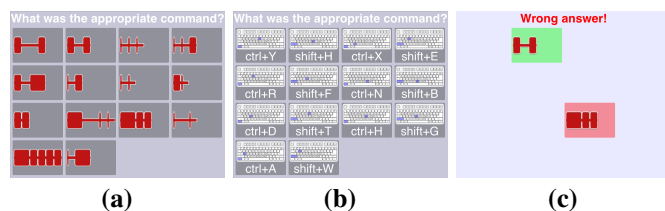


Figure 12. Confirmation in the *Rhythm* (a) and *Hotkey* (b) conditions. Feedback for a wrong answer in the *Rhythm* condition (c).

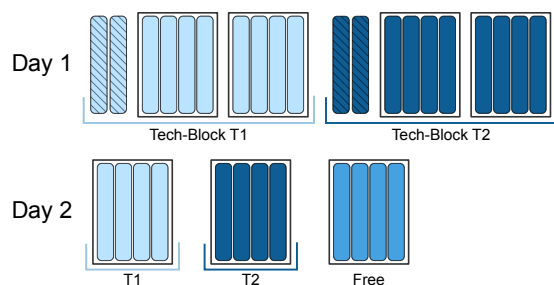


Figure 13. A sample session. Hatched sub-blocks are learning trials, boxed sub-blocks are sub-sessions.

for this procedure is that we are primarily interested in the memorization of the associations, not the participants' ability to perform the triggers. For rhythmic patterns, it also allows us to test the recognition rate of the classifier.

Apparatus & Participants

We used the same apparatus as in Experiment 1. We recruited 14 participants (5 female), aged between 22 and 33 (mean 26, median 26). Five of them had participated in Experiment 1.

Design & Procedure

The experiment is a within-subject design with technique (TECH) and command (CMD) as primary factors. The experiment is split into two sessions held on two consecutive days. The first day, all participants are presented with rhythmic patterns in a 5 minutes practice session based on Experiment 1. We use TECH as a blocking factor, counterbalanced across participants. The second day, a *Free* block is added at the end of the testing phase. In this block, participants can choose to use *Rhythm* or *Hotkey* for each trial, but cannot get help.

Each TECH-block is divided into several sub-blocks of 15 trials: (i) 2 learning sub-blocks with 4 testing sub-blocks each on the first day; (ii) 4 testing sub-blocks on the second day. Thus, the testing phase of the experiment is split into SUBSESSIONS of 60 trials each: two on the first day to evaluate immediate memorization of triggering commands and one on the second day to test mid-term recall (Figure 13).

In order to simulate a more realistic setup, where some commands are more frequently used than others, we assign an apparition frequency to each of the 14 commands following a Zipf distribution [14, 1]. For the learning phase we use the frequencies (6, 6, 3, 3, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1) and for the testing phase (12, 12, 6, 6, 4, 4, 3, 3, 2, 2, 2, 2, 1, 1). The 14 commands are combined with these frequencies using 7 different permutations, and each frequency assignment is

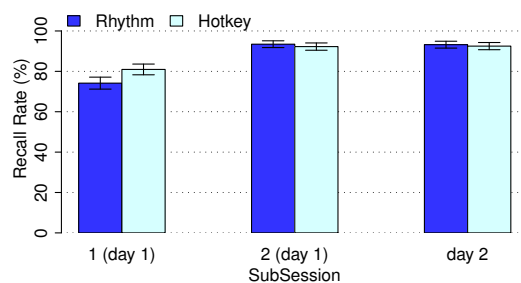


Figure 14. Recall rate for both techniques by sub-session.

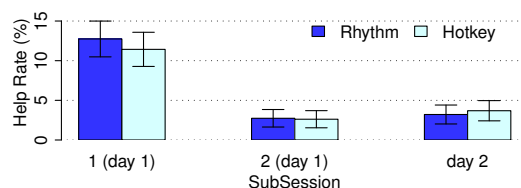


Figure 15. Help usage rate for both techniques for each sub-session.

counterbalanced across participants, resulting in the same number of trials for each command overall. The presentation of the trials is randomized across consecutive pairs of sub-blocks.

The experiment takes about one hour on day 1 and 30 minutes on day 2, after which participants are given a questionnaire to collect subjective observations and preferences.

Quantitative Results

Our main measures are (i) *recall rate*, the percentage of correct answers in the testing phase without help; and (ii) *help rate*, the percentage of trials where the participants used help in the testing phase. We analyze the results according to TECH and the three sub-sessions of the experiment by considering these measures in the model $TECH \times SUBSESSION \times Rand(PARTICIPANT)$.

We find a significant effect of SUBSESSION on the recall rate ($F_{2,26} = 103, p < 0.0001$). A post-hoc t-test with Bonferroni correction shows that the recall rate is significantly lower only between the first sub-session and the two following ones (Figure 14). There is no significant effect of TECH on recall rate ($F_{1,13} = 0.61, p = 0.4474$), but the ANOVA reveals a significant interaction effect $TECH \times SUBSESSION$ ($F_{2,26} = 5.36, p = 0.0113$). Post-hoc t-tests with Bonferroni correction show a significant difference between *Rhythm* and *Hotkey* for the first sub-session (74% and 81% respectively). For the remaining sub-sessions, the results are extremely close between the two techniques with a recall rate of about 93% (Figure 14).

For the use of help, an ANOVA reveals a significant effect of SUBSESSION ($F_{2,26} = 17.3, p < 0.0001$), no effect of TECH ($F_{1,13} = 0.04, p = 0.8532$), and no $TECH \times SUBSESSION$ interaction effect ($F_{2,26} = 0.62, p = 0.545$). We find only one significant difference among sub-sessions: help was used more often in the first sub-session than in the two subsequent ones (see Figure 15).

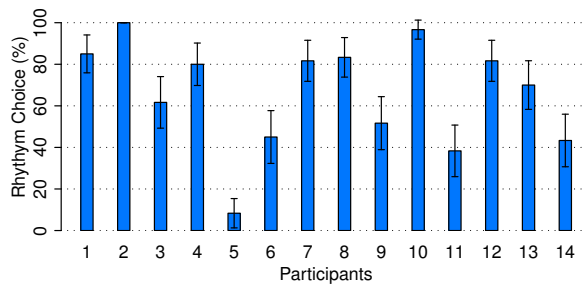


Figure 16. Percentage use of *Rhythm* by participant (*Free* condition).

Results for rhythmic patterns and hotkeys are quite similar, suggesting that rhythmic patterns can be memorized as successfully as hotkeys without mnemonics. This is a remarkable result considering how widespread hotkeys are.

Recall rates are consistent across commands. Considering only the *Rhythm* condition, we build the model $CMD \times SUBSESSION \times Rand(PARTICIPANT)$ for recall rate and help and see a significant effect of CMD on recall rate ($F_{13,169} = 1.17$, $p = 0.025$). A post-hoc t-test with Holm corrections shows significant differences only between R_3 and R_{13} (recall rate about 97%) and R_{10} , R_{11} and R_{14} (~80%).

To test our classifier, we compare the pattern recognized by the classifier with the answer selected by the participant using the model $TECH \times SUBSESSION \times Rand(PARTICIPANT)$. We find a significant effect of $TECH$ ($F_{1,13} = 5.34$, $p = 0.038$), with *Rhythm* having a significant lower success rate than *Hotkey*: 85.2% vs. 91.8%. The success rate for *Hotkey* is surprisingly low, as we expect few if any errors when entering hotkeys. This may be due to participants changing their mind as to which was the right hotkey when they see the answer sheet. For *Rhythm*, the rate is also lower than expected, but the same phenomenon may have occurred. Indeed, the success rate of *Rhythm* relatively to *Hotkey* is 92.8%, close to the rate obtained on the data for Experiment 1 (94%).

Qualitative Results

Figure 16 shows the percentage of trials where participants used rhythmic patterns in the *Free* condition, on the second day of the experiment. Ten participants (out of 14) used rhythmic patterns more often than hotkeys. Seven participants used rhythmic patterns more than 80% of the time, while only one participant used rhythmic patterns less than 20% of the time.

The answers to the questionnaire were generally positive, confirming the previous results. Out of the 14 participants, 9 preferred using the rhythmic patterns, 3 the hotkeys, 2 had no preference. Those who preferred using rhythmic patterns did so mostly because of the “fun factor” of tapping rhythms, but also because it could be performed “in place” on the trackpad, even for a novice user, without having to visually search the keys on the keyboard. On the other hand, several participants noticed that hotkeys are faster to perform and preferred to use hotkeys when the corresponding pattern is too long.

Regarding memorization, some participants reported using mnemonics related to the rhythm itself in order to help memorization. For instance, a subject linked the “boxing gloves”

command and the corresponding pattern P9 (Figure 4) to a “pif paf boom” onomatopoeia that, for him, echoed the “short short medium” structure of the pattern. Another participant also reported linking the pattern structure with the pronunciation of the object’s name, e.g., “toma-to-to-to-to” for command 13 and pattern P28. Subjects also used the graphical representation of patterns to memorize them, which supports our design for this representation. For example, one participant stated that “the rhythmic pattern’s visual representation for the cherry looks like a cherry”.

These comments suggest that users elaborate efficient strategies for the memorization of rhythmic patterns, based on the rhythm itself or its visualization. Since commands were assigned to rhythmic patterns randomly, we did not expect such associations, but this finding opens the way to studying ways to reinforce these associations. This is commonly done for gestures, e.g., a question mark for help, and hotkeys, e.g. Ctrl-S for Save. In particular, various strategies could be explored to create visual “cheatsheets” for rhythmic patterns or display them next to menu commands, like hotkeys.

In addition, the complexity of performing rhythmic patterns can be turned into an advantage for memorization. Since deeper and greater numbers of levels of encoding and processing help memory [8], combining motor and auditive perception of rhythmic patterns may help users memorize, i.e., encode, their associations with commands.

SUMMARY AND PERSPECTIVES

In this paper we studied the use of rhythmic patterns in HCI. We explored Rhythmic Interaction as an opportunity to generalize the primitive use of rhythm in existing techniques, e.g., long click and double click, as well as to promote a new input modality. Since Rhythmic Interaction relies on the time dimension instead of the spatial and visual dimensions used by most input methods, it is well suited when space is limited or when visual attention is not available.

We presented a grammar for creating rhythmic patterns as well as two recognizers that do not require training. A first experiment evaluated the ability of casual users to reproduce rhythmic patterns very precisely with different feedback conditions. We found that some complex patterns can be difficult to reproduce in such a precise way, but that audio and/or visual feedback improve accuracy. After analyzing recognition errors, we designed a different recognizer that reached 94% recognition rate for the 30-pattern vocabulary of Experiment 1. We ran a second experiment to investigate the memorization of associations between rhythmic patterns and commands, i.e., *rhythmic shortcuts*. The results suggest that rhythmic patterns are recalled as efficiently as traditional hotkeys and that users create effective mnemonic strategies to associate rhythms with commands.

This work demonstrates the potential of rhythmic patterns as an input method, and contributes a 14-pattern vocabulary that has proven usable by novice users. Beyond triggering commands and switching modes in standard desktop environments, rhythmic patterns could be used in many contexts: eye-free control of a mobile device, such as a cellular phone

or a mobile player; remote control of interactive environments such as wall-size displays by tapping on wearable sensors without the need for visual attention; selection of an object on a tabletop when it is not easily reachable, etc.

Our future work will address issues such as the segmentation of patterns, the scalability of the vocabularies and the speed of execution, which are important for the design of Rhythmic Interactions. Another area for future work is the use of multiple fingers or both hands to tap patterns and to combine rhythmic interaction with other interaction techniques. More complex actions than tapping should also be explored to enter rhythmic structures, such as performing sequences of gestures or keyboard taps, as well as the use of the temporal dimension to convey additional information. Furthermore, rhythmic *output*, such as vibration patterns on mobile devices, seems worth studying since perception and performance of rhythmic patterns are tightly linked. Finally, the power of rhythmic interaction could be expanded by exploiting syntactic features used in music such as performing sequential or parallel combinations of patterns.

ACKNOWLEDGEMENTS

We thank Caroline Appert for feedback on earlier versions of this paper, our participants and the anonymous reviewers.

REFERENCES

- Appert, C., and Zhai, S. Using strokes as command shortcuts: cognitive benefits and toolkit support. In *Proc. CHI '09*, ACM (2009), 2289–2298.
- Arom, S., Thom, M., Tuckett, B., and Boyd, R. *African polyphony and polyrhythm: musical structure and methodology*. Cambridge university press, 1991.
- Baudel, T., and Beaudouin-Lafon, M. Charade: Remote control of objects using free-hand gestures. *Comm. ACM* 36 (1993), 28–35.
- Blair, C. R. On computer transcription of manual morse. *JACM* 6, 3 (1959), 429–442.
- Brochard, R., Touzalin, P., Despres, O., and Dufour, A. Evidence of beat perception via purely tactile stimulation. *Brain Res.* 1223 (2008), 59 – 64.
- Chen, S.-C., Chien, C.-Y., Chang, W.-M., and Lin, S.-W. A new assistive communication system for the serious disabled. In *Proc. iCREATE '08*, START Centre (2008), 59–64.
- Clarke, E. Rhythm and timing in music. *The Psychology of Music* 2 (1999), 473–500.
- Craik, F. I., and Lockhart, R. S. Levels of processing: A framework for memory research. *J. Verbal Learning and Verbal Behavior* 11, 6 (1972), 671–684.
- Crossan, A., and Murray-Smith, R. Rhythmic interaction for song filtering on a mobile device. In *Proc. HAID '06*, Springer (2006), 45–55.
- Faure, G., Chapuis, O., and Roussel, N. Power tools for copying and moving: useful stuff for your desktop. In *Proc. CHI '09*, ACM (2009), 1675–1678.
- Fekete, J.-D., Elmqvist, N., and Guiard, Y. Motion-pointing: target selection using elliptical motions. In *Proc. CHI '09*, ACM (2009), 289–298.
- Fraisse, P. Rhythm and tempo. In *The Psychology of Music*, Academic Press (1982), 149–180.
- Glass, L. Synchronization and rhythmic processes in physiology. *Nature* 410, 6825 (2001), 277–284.
- Grossman, T., Dragicevic, P., and Balakrishnan, R. Strategies for accelerating on-line learning of hotkeys. In *Proc. CHI '07*, ACM (2007), 1591–1600.
- Hinckley, K., Baudisch, P., Ramos, G., and Guimbretiere, F. Design and analysis of delimiters for selection-action pen gesture phrases in scriboli. In *Proc. CHI '05*, ACM (2005), 451–460.
- Kurtenbach, G., and Buxton, W. User learning and performance with marking menus. In *Proc. CHI '94*, ACM (1994), 258–264.
- Lantz, V., and Murray-Smith, R. Rhythmic interaction with a mobile device. In *Proc. NordiCHI '04*, ACM (2004), 97–100.
- Large, E. W. Periodicity, pattern formation, and metric structure. *J. New Music Research* 30, 2 (2001), 173 – 185.
- MacDougall, H. G., and Moore, S. T. Marching to the beat of the same drummer: the spontaneous tempo of human locomotion. *J. Applied Physiology* 99, 3 (2005), 1164–1173.
- Malacria, S., Lecolinet, E., and Guiard, Y. Clutch-free panning and integrated pan-zoom control on touch-sensitive surfaces: the cyclostar approach. In *Proc. CHI '10*, ACM (2010), 2615–2624.
- Manuel, R. *Histoire de la Musique*. Encyclopédie de la Pléiade, Gallimard, Paris, France, 1960.
- Maury, S., Athènes, S., and Chatty, S. Rhythmic menus: toward interaction based on rhythm. In *Proc. CHI '99 EA*, ACM (1999), 254–255.
- Moelants, D. Preferred tempo reconsidered. In *Proc. ICMPC '02*, AMPS (2002), 580–583.
- Petitto, L. A., Holowka, S., Sergio, L. E., Levy, B., and Ostry, D. J. Baby hands that move to the rhythm of language: hearing babies acquiring sign languages babble silently on the hands. *Cognition* 93, 1 (2004), 43 – 73.
- Potter, D. D., Fenwick, M., Abecasis, D., and Brochard, R. Perceiving rhythm where none exists: Event-related potential (erp) correlates of subjective accenting. *Cortex* 45, 1 (2009), 103–109.
- Rammsayer, T., and Lima, S. Duration discrimination of filled and empty auditory intervals: Cognitive and perceptual factors. *Perception and Psychophysics* 50 (1991), 565–574.
- Repp, B. H. Rate limits in sensorimotor synchronization. *Advances in Cognitive Psychology* 2, 2 (2006), 163–181.
- Robinson, S., Rajput, N., Jones, M., Jain, A., Sahay, S., and Nanavati, A. Tapback: towards richer mobile interfaces in impoverished contexts. In *Proc. CHI '11*, ACM (2011), 2733–2736.
- Sacks, O. *Musicophilia: Tales of Music and the Brain*, 2 ed., vol. 1. Vintage Books, New York, USA, 2008.
- Scott, J., Dearman, D., Yatani, K., and Truong, K. N. Sensing foot gestures from the pocket. In *Proc. UIST '10*, ACM (2010), 199–208.
- Szentgyorgyi, C., and Lank, E. Five-key text input using rhythmic mappings. In *Proc. ICMI '07*, ACM (2007), 118–121.
- Westeyn, T., and Starner, T. Recognizing song-based blink patterns: Applications for restricted and universal access. In *Proc. FGR '04*, IEEE (2004), 717–722.
- Wobbrock, J. O. Tapsongs: tapping rhythm-based passwords on a single binary sensor. In *Proc. UIST '09*, ACM (2009), 93–96.