Henkjan Honing
MMM Group, NICI, University of Nijmegen
P.O. Box 9104, 6500 HE Nijmegen, The Netherlands
Music Department, University of Amsterdam
Spuistraat 134, 1012 VB Amsterdam, The Netherlands
honing@nici.kun.nl

From Time to Time: The Representation of Timing and Tempo

Timing plays an important role in the performance and appreciation of almost all types of music. It has been studied extensively in music perception and music performance research (see Palmer 1997 for a review). The most important outcome of this research is that a large part of the timing patterns found in music performance—commonly referred to as *expressive timing*—can be explained in terms of musical structure, such as the recurrent patterns associated with the metrical structure that are used in jazz swing, or the typical slowing down at the end of phrases in classical music from the Romantic period. These timing patterns help in communicating temporal structure (such as rhythm, meter, or phrase structure) to the listener. Furthermore, timing is adapted with regard to the global tempo: at different tempi, other structural levels of the music are emphasized, and the expressive timing is adapted accordingly. In short, in music performance there is a close relationship between expressive timing, global tempo, and temporal structure. One cannot be modeled without the other (see Figure 1).

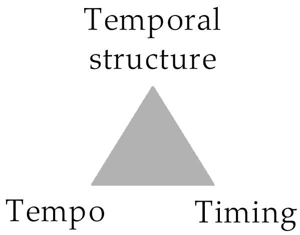
Existing computational models of expressive timing (e.g., Clarke 1999; Gabrielsson 1999) are primarily concerned with explaining tempo variations, using *tempo curves* (specifying tempo or the reciprocal of duration as a function of the position in the score) as the underlying representation. Although a useful means of measuring tempo patterns in a performance, tempo curves have been shown to fall short as an underlying representation of timing from a musical perspective (Desain and Honing 1991, 1993) and a psychological perspective (Desain and Honing 1994). For instance, some types of timing, like *chord spread* (the asynchrony in performing a chord), ornaments (like grace notes), or the timing between parallel voices simply cannot be measured or represented as tempo deviations. Furthermore, in some musical situations or styles of music, where the global tempo is mostly constant, *event-shift* (Bilmes 1993)—measured as the deviation with respect to a fixed beat or pulse in a constant tempo—offers a more natural way of representing timing.

First, this article will review existing representations of timing and tempo common in computational models of music cognition and in programming languages for music. Their differences are discussed, and some refinements will be proposed (referred to as *time maps*, or TMs). The second part presents an alternative representation and model for time transformation: so-called *timing functions* (TIFs, an acronym chosen to distinguish them from TFs, *or time functions*, described by Desain and Honing 1992). This knowledge representation differs in two important aspects from earlier proposals. First, expressive timing is seen as a combination of a tempo component (expressing the change of rate over a fragment of music), and a timing (or time-shift) component that describes how events are timed (e.g., early or late) with respect to this tempo description. Second, expressive timing can be specified in relation to the temporal structure (e.g., position in the phrase or measure), as well as in terms of performance-time, score-time, and global tempo. In addition, TIFs support compositionality (how simple descriptions can be combined into more complex ones) and maintain consistency over musical transformations (how these descriptions of timing and tempo should adapt when other parts of the representation change), both important design criteria of the formalism.

Another design criterion is that timing transformations (e.g., the application of an expressive timing model to a score representation) are part of the representation, instead of only acting on a score—the difference between a knowledge representation and a data representation. To realize this, it is crucial to have access to the timing transformations themselves, not only to the result of their applica-

Computer Music Journal, 25:3, pp. 50–61, Fall 2001

Figure 1. Three aspects of music performance that are closely related: expressive timing, global tempo (or rate), and temporal structure (such as rhythm, meter, or phrase structure).



tion (as is the case in most music representation systems, e.g., Dannenberg 1993). In other words, to be able to perform multiple transformations (as required by music sequencers and expression editors), or compose a number of transformations into a complex transformation (essential in programming languages for music or in combining partial computational models of expressive timing), the transformations themselves should be an object within the representation, not just functions applied to it. A practical example will clarify this.

Imagine one applies a jazz-swing transformation to a score representation in a sequencer, resulting in a performance with some expressive timing added. Next, the user applies a global tempo transformation to this result, simply speeding up the performance. The result of this second transformation, however, will sound strange. This is because the swing pattern is closely related to the beat at the original tempo, which is now changed by the tempo transformation. In order to obtain the desired result, the swing pattern transformation must adapt itself, in retrospect, to the new tempo. It must, in fact, be a function of tempo not known at the time of application. In general, this means that previously applied transformations (or nested transformations, for that matter) must sometimes adapt themselves when new transformations are applied. (Note: an alternative could be to describe, with every new transformation, how all the other aspects of the representation should be kept consistent. However, this is a virtually impossible task.) Defining this so-called "behavior under transformation" is essential in situations where a representation of timing is actually used (such as in music editors or computational modeling), instead of being a static description of a performance—the difference between a knowledge representation and a data representation (cf. "The Vibrato Problem" described in Honing 1995; Dannenberg, Desain, and Honing 1997).

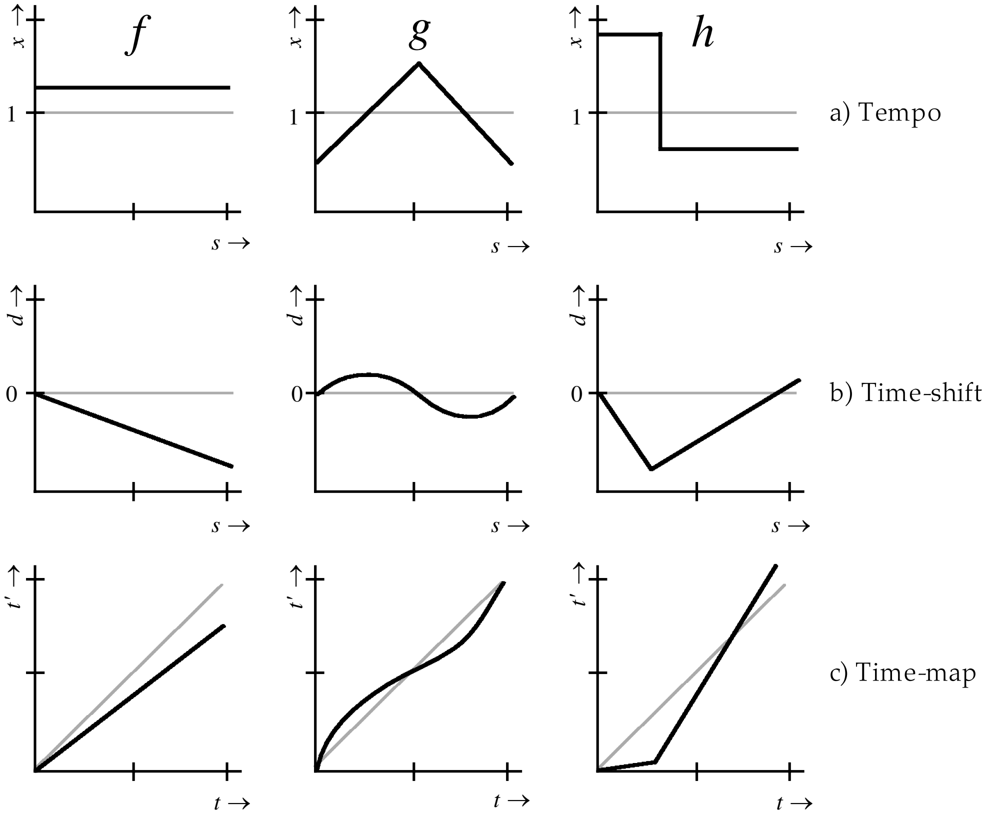
The Representation of Tempo and Timing

There are a number of ways of representing expressive timing. The three most frequently encountered ones will be described here. *Tempo functions* are primarily used in music psychology research. They form the output of several generative models of expressive timing (e.g., Clynes 1995; Todd 1992; Sundberg 1988). Most of this research is concerned with keyboard music from the Baroque and Romantic periods, where indeed *tempo rubato* (expressive tempo fluctuations) serves an important expressive function. In some more recent studies, *time-shift* (or event-shift) patterns (i.e., timing measured as deviations from a regular pulse) are analyzed, for example, in studies of timing in jazz ballads (Ashley 1996) or in Cuban percussion music (Bilmes 1993). In computer music, *time-maps*—also referred to as time-deformations (Anderson and Kuivila 1990) or time-warps (Dannenberg 1997)—are primarily used. They express performance-time directly as a function of score-time, and can, in principle, describe both time-shift and tempo-change.

Mathematically, tempo changes can be expressed as time-shifts and vice versa: they are equivalent under some constraints that will be mentioned later (see Figure 2). However, they are musically very different notions. Tempo change is associated with, for example, *rubato*, *accelerando* (speeding up), and *ritardando* (slowing down), while time-shift involves, for instance, accentuating notes by delaying them a bit or playing notes "behind the beat," both apparently independently of the tempo. Also, from a perceptual point of view, it seems that listeners do perceive tempo relatively independently from timing, a point that will be discussed later in this article.

Before proposing a formalism that takes these observations into account the three existing representations mentioned above (i.e., tempo curves, time-shift

Figure 2. (a) Three arbitrary tempo functions, f, g, *and* h, *and their equivalent representations as a time-shift function (b) and a time-map (c).* f *depicts a constant tempo,* g *a linear tempo-change (a "give and take" rubato; see Figure 3 for an example in musical notation), and* h *a sudden tempo-change. For a restricted class of functions (to which* f, g, *and* h *belong), one can freely convert from one representation to the other (see text for details).*



functions, and time-maps) will be presented. After their formal definition (in a functional style, without any typing for simplicity), their composition (in the mathematical sense) will be shown, compositionality being an essential strength of all three alternatives.

A *tempo function* (or tempo curve) can be expressed as a function of score-time s (a rational number denoting symbolic score-time) returning a tempo factor x (a real number):

$$f(s) \rightarrow x. \tag{1}$$

To obtain the performance-time at score-time s, one must integrate the tempo function up to that score-time. Tempo functions can be composed by multiplying their individual results:

$$(f \otimes g)(s) = f(s) \times g(s). \tag{2}$$

Equation 2 states that the composition ($\otimes$) of two tempo functions (f and g) applied to a score-time s is equivalent to applying each individual tempo function to that score-time and multiplying ($\times$) the results.

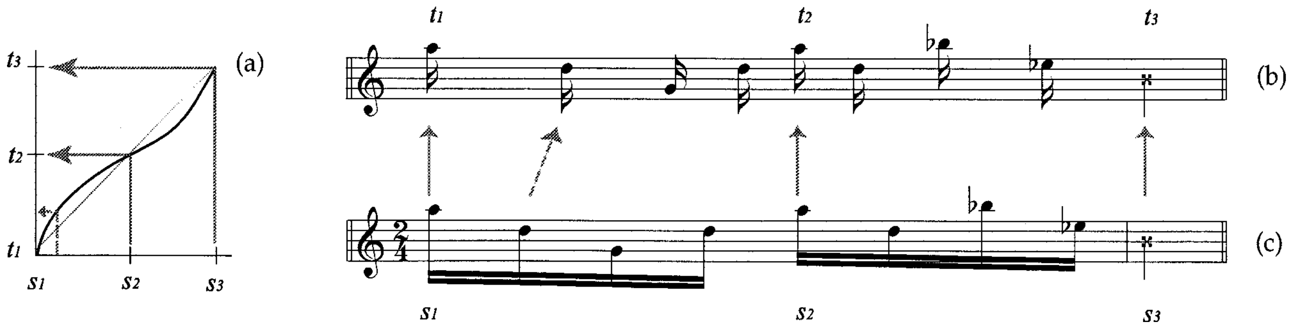
A *time-shift function* (or event shift) can be expressed as a function of score-time s (a rational number denoting symbolic score-time) returning a deviation interval d (a real number), that is, the amount of time an event is shifted with respect to its score-time:

$$f(s) \rightarrow d. \tag{3}$$

To obtain the performance-time at score-time s, one can simply add the deviation d to it. In principle, time-shift functions can change the order of events with respect to the score. (Note that when this occurs, a time-shift function can no longer be converted into a tempo function.) Time-shift functions can be composed by adding the results of the components:

$$(f \otimes g)(s) = f(s) + g(s). \tag{4}$$

Figure 3. A time-map (a), a score (c), and a score (b) with spacing indicating its timing. The time-map represents a "give and take" rubato with an acceleration (from t_1 *to* t_2*) and a deceleration (from* t_2 *to* t_3*), but with regularly timed beats (at* t_1, t_2, *and* t_3*; see tempo function* g *in Figure 2). The result of applying the time-map (a) to the score (c) is shown in (b). Solid gray lines indicate the regularly timed beats, and the dashed gray line illustrates an example of a delayed note.*



Finally, a *time-map* is defined as a function from pre-perturbed time t (a real number) to perturbed time or performance-time t' (a real number):

$$f(t) \rightarrow t'. \qquad (5)$$

They can be composed using function composition:

$$(f \otimes g)(t) = f(g(t)). \qquad (6)$$

Time-maps can take score-time s .as argument, but this is then just a special case of pre-perturbed time. This has implications for composing time-maps, as will be shown in the next section.

Figure 2 illustrates some examples of prototypical shapes of tempo functions and their representations as time-shift functions and time-maps: a constant tempo (f), a gradual tempo change (g), and an instantaneous tempo change (h). As an illustration in common music notation, the application of function g (a simplistic "give and take" rubato) is shown in Figure 3.

Time-maps are defined as continuous, monotonically increasing functions (Jaffe 1985). This means that time is not allowed to reverse or jump ahead to allow conversions from time-maps to tempo-functions and time-shift functions and vice versa. (The functions shown in Figure 2 belong to this class.)

The composition of two time-maps can be visualized as shown in Figure 4. Here, a time-map f is rotated 90 degrees to the left to connect its x-axis (input) to the y-axis (output) of time-map g, depicting the composite time-map $f(g(t))$ (or $f \circ g$). Note that f and g are used throughout this article to represent both basic and complex composed time-maps.

It is the simplicity of composition and the direct availability of performance-time (by simply looking it up in the time-map, instead of integrating tempo functions) that makes the time-map the representation of choice in most computer music systems. However, time-maps have some limitations that must be resolved before they can be used as a flexible basis for a representation of timing.

Limitations of the Time-Map Representation

Some of the limitations of time-maps will now be discussed, followed by an extension that resolves some of these (referred to as extended TMs).

Score-time is Lost in Composition

One of the problems with TMs (compared to time-shift and tempo functions) is that score-time is lost in composition (see Equation 6). For example, in Figure 4, f is accessed with s' instead of s (the transformed instead of the original score-time). This is problematic when timing needs to be expressed in terms of score position, such as swing, or other types of timing related to metrical time (i.e., patterns linked to the metrical or phrase structure). However, this can be solved simply by making a TM a function of both types of musical time: performance-time and score-time (this will explained below). Note again that f or g can be complex, composed time-maps, so simply reversing the order of application does not solve the problem.

Figure 4. Composition of two time-maps (f ∘ g): Time-map f is rotated 90 degrees to the left to connect its x-axis (input) to the y-axis (output) of time-map g. The gray arrow marks how score-time s is mapped to a perturbed-time s' by g, and, successively, how this s' is mapped to s'' by f.

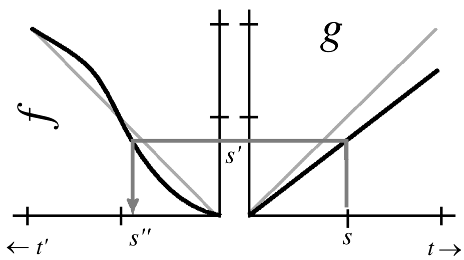
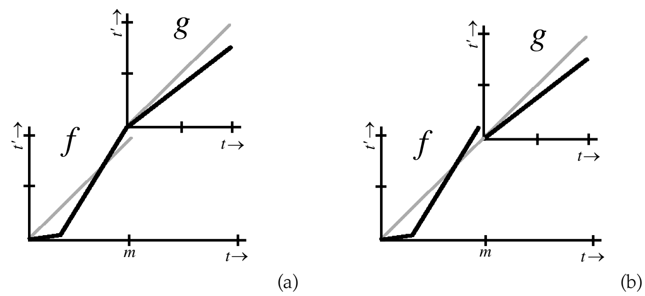


Figure 5. Two ways of concatenating f and g in score-time point m: joining them in performance-time (a) or score-time (b).



Support of Concatenation

There are numerous ways of concatenating two arbitrary time-maps. However, from a musical perspective, two alternatives come to mind: joining them in performance-time (see Figure 5a) or in score-time (see Figure 5b). The first can be interpreted as a continuous change of tempo, the latter a shift of time, changing the timing of events without altering the baseline tempo. The first type of concatenation is supported by Anderson and Kuivila (1990) and Dannenberg (1997). The second type of concatenation is illegal in all time-map implementations, because the resulting time-map might not be monotonously increasing: time-shift functions can change the score order of notes. However, from a musical perspective, this is perfectly plausible. For example, whereas for a sequence of notes the tempo is constant or gradually changing, some notes can be accented by performing them somewhat early or late.

Even with the realization of the need for different kinds of concatenation, one must choose one or the other type, since we cannot tell from the time-map itself whether it is the result of a tempo-change definition or an interpretation as time-shift. This suggests that one should keep both types of timing (tempo-change and time-shift) separate and concatenate each in its own typical way (which will be discussed later).

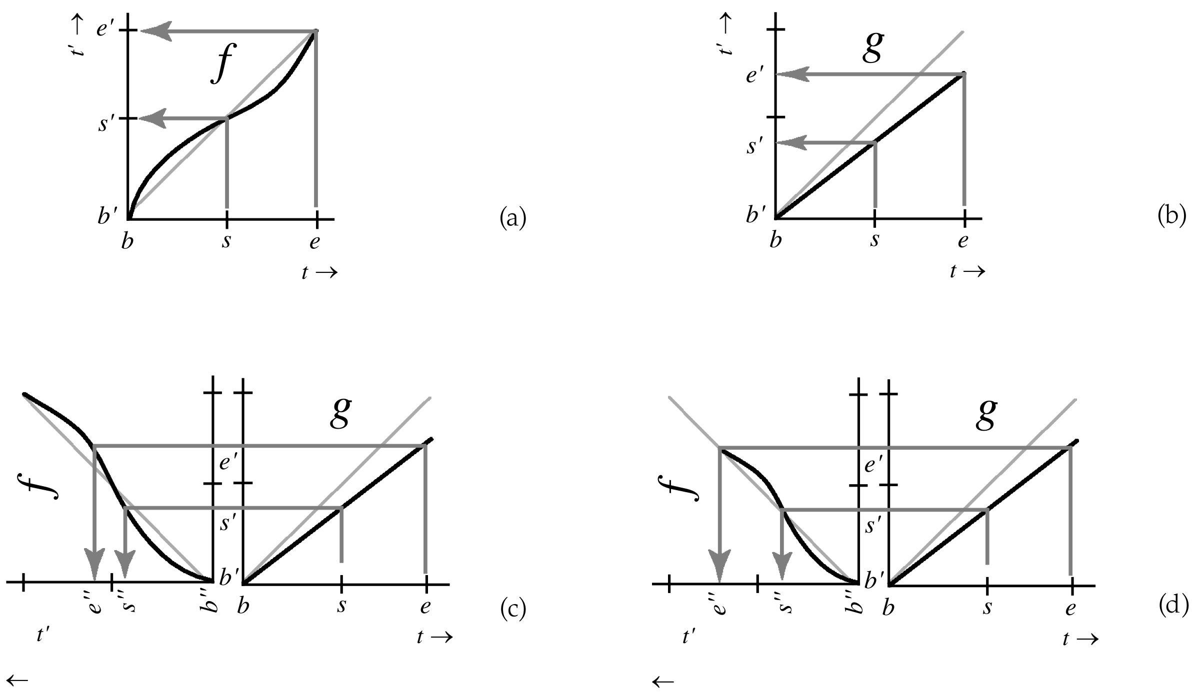
Access to Score and Performance Duration in Composition

Besides the need for score-time in the composition of time-maps (as discussed above), even more temporal information is necessary to make the composition of time-maps as simple as possible. This can be illustrated by looking in detail at the composition of the two time-maps shown in Figure 6. Let us interpret f (Figure 6a) as a simplistic "give and take" type of rubato (see Figure 3) synchronous at every beat (score-times b, s, and e) and speeding up and slowing down between them, and g (Figure 6b) as a faster constant tempo. In combining these two time-maps, one expects to get a composite time-map with a faster tempo and a timing pattern that slows down and speeds up again, but synchronizes on every beat (i.e., the two beats have the same length: $s - b = e - s$). However, as can be seen in Figure 6c, the score-time s is mapped to performance-time s' by g and accesses f in the wrong position (i.e., not at s, but some time before it), resulting in beats of unequal length (i.e., $s'' - b'' \neq e'' - s''$).

What is needed is a way of linking a time-map to a temporal interval (e.g., the length of a bar). As such, a time-map can adapt its definition according to the actual length (in performance time) of the interval, i.e., the length as a result of all previously applied time-maps (in this example g). As an example, in Figure 6d, f is adapted to fit the current length of the bar in performance-time (by inspecting the result of applying g), resulting in a correct lookup in f. (Here, again, changing the application order is not a solution, since f and g can be complex, composed time-maps.)

The problem of losing score-time in composition and explicit support of two types of concatenation will be resolved in an extension of TMs. The third problem of relating a time-map to a temporal interval will be resolved by distinguishing between two types of time-maps in TIFs and relating them

*Figure 6. Problem in the composition of time-maps: a time-map describing a simple "give and take" type of rubato that is synchronous at the every beat (score-times b, s, and e) (a), a time-map of a faster constant tempo (b), an erroneous composition of two time-maps (*f *is accessed in the wrong position) (c), and a correct composition, because* f *is adapted to fit the current length of the bar in performance-time (d). Both beats (from* b' *to* s' *and from* s' *to* e'*) stay of equal length, through at a faster global tempo. (Note that* f *and* g *can be complex, composed time-maps, so simply changing the order of composition is not a solution.)*



to a temporal interval in *generalized timing functions* (GTIFs).

Improving Time-Maps(TMs)

To resolve the restrictions on TMs discussed above, two types of TMs (indicated in plain font) will be defined, one representing time-shift (f^{+}) and one representing tempo-change ($f^{\times}$). Both are functions of performance-time and score-time, and both return a perturbed performance-time t':

$$f^{+}(s, t) \rightarrow t' \qquad (7)$$

$$f^{\times}(s, t) \rightarrow t'. \qquad (8)$$

Both can be composed in the same way. (Note that f and g without superscript are used when the type is irrelevant for an operation):

$$(f \otimes g)(s,t) = f(s, g(s,t)). \qquad (9)$$

Depending of the type of time-map at hand, one can employ either of two types of concatenation. The first, as required for time-shift TMs (cf. Figure 5b) is given by

$$(f^{+} \oplus_m g^{+})(s,t) = \begin{cases} s \le m & f^{+}(s,t) \\ s > m & g^{+}(s,t) \end{cases} \qquad (10)$$

where $\oplus_m$ denotes the concatenation function and m indicates the score-time at which the two functions will be joined. Equation 10 states that the concatenation ($\oplus_m$) of the time-maps f and g applied to a score-time (s) is the application of f before score-time m and the application of g after that point. (The issue of which temporal intervals these functions are defined will be addressed later.)

The second type of concatenation is required for tempo-change TMs (cf. Figure 5a):

$$(f^{\times} \oplus_m g^{\times})(s,t) = \begin{cases} s \le m & f^{\times}(s,t) \\ s > m & (\Delta_m(f^{\times}, g^{\times}))(s,t) \end{cases} \qquad (11)$$

Here, the function Δ_m calculates a new tim.e-map function shifted in time such that g connects it where f ended:

$$(\Delta_m(f^{\times}, g^{\times}))(s,t) = g^{\times}(s,t) + f^{\times}(m,m) - m. \qquad (12)$$

This simply calculates the difference between score-time m and its corresponding performance-time m' (the result of applying the function $f^{\times}$ to score-time m and untransformed performance-time, also m), added to the successive tempo-change function g. Note that the difference between score-time m and its performance-time m' corresponds to the difference in height between the tempo baselines (gray lines) shown in Figure 5a.

Having introduced two types of time-maps and their respective definition for composition and concatenation, I will continue with the description of timing functions that integrate the two types of time-maps.

Timing Functions (TIFs)

The two aspects of timing described above can be combined into one TIF, a tuple consisting of a time-shift function (f^{+}), and a tempo-change function ($f^{\times}$). The symbols **f** and **g** (boldface) are used to refer to such timing functions:

$$\mathbf{f} \equiv \left\langle f^{+}, f^{\times} \right\rangle. \tag{13}$$

Or, in computational terms, a TIF is a data structure containing two TMs, one describing time-shift and the other tempo-change. These will remain independent through composition and concatenation, since this is different for each type of timing. Only at the final stage, when actually applying a TIF, are the components combined, by first applying the tempo-change component ($f^{\times}$) and then the time-shift component (f^{+}) to the current performance-time and score-time.

The evaluation function **E** describes how the result (a new performance-time) is obtained, given a TIF, a score-time, and a performance-time:

$$\mathbf{E}(\mathbf{f}, s, t) = f^{+}(s, f^{\times}(s, t)). \tag{14}$$

This evaluation order ensures that first the tempo baseline (from which the time-shift descriptions deviate) is obtained. The order in which individual components are combined is not relevant anymore. (One could, however, in cases where one wants to have explicit control over this application order, add a third function to the tuple that determines this order: a function of, respectively, the two TMs, s and t.)

Composition

The composition of TIFs is straightforward. It is simply the composition of the individual components:

$$\mathbf{f} \otimes \mathbf{g} = \left\langle f^{+} \otimes g^{+}, f^{\times} \otimes g^{\times} \right\rangle. \tag{15}$$

This states that the composition of the timing functions **f** and **g** is given by the composition of their time-shift components (f^{+} and g^{+}) and their tempo change components ($f^{\times}$and $g^{\times}$). Both are composed as defined in Equation 9.

Concatenation

Concatenation of time-maps can now be described correctly by concatenating the time-shift component in a different way than the tempo-change component. Concatenation of two TIFs is defined as

$$\mathbf{f} \oplus_{m} \mathbf{g} = \left\langle f^{+} \oplus_{m} g^{+}, f^{\times} \oplus_{m} g^{\times} \right\rangle \tag{16}$$

where m indicates the score-time at which the two functions will be joined. Here, the time-shift components (f^{+} and g^{+}) are concatenated according to Equation 10, and the tempo-change components ($f^{\times}$and $g^{\times}$) are concatenated according to Equation 11.

Generalized Timing Functions (GTIFs)

Finally, I will discuss a generalization of timing functions that allows them to be related to the temporal structure (instead of only the current score-time and performance-time). Specifically, a generalized timing function (GTIF) can access the duration of the interval to which it is applied (e.g., a measure or a phrase), its current position (in both score-time and performance-time), and the current tempo. The key idea in realizing this is to provide the definition of a timing function with access to the begin-time and end-time of a temporal interval (in score-time) over which the function is defined, together with the complete "underlying" TM (i.e., the composite function of all previously applied timing transformations). A GTIF is, in fact, a *timing function constructor* (indicated in italic boldface):

$$\boldsymbol{f}(b,e,\mathbf{u}) \rightarrow \mathbf{f}. \tag{17}$$

This states that the constructor $\boldsymbol{f}$ is a function of b (begin-time, a rational number denoting symbolic score-time), e (end-time, a rational number denoting symbolic score-time), and $\mathbf{u}$ (a TIF of all previous applied timing transformations). It returns a new TIF $\mathbf{f}$ that can access all relevant temporal information (both score-times and performance-times) of that time interval.

Composition of two GTIFs is defined as

$$(\boldsymbol{f} \otimes \boldsymbol{g})(b,e,\mathbf{u}) = \boldsymbol{f}(b,e,\boldsymbol{g}(b,e,\mathbf{u})). \tag{18}$$

Concatenation of two GTIFs at time point m (a rational number denoting the point where the two functions are joint in metrical time) is defined as the concatenation of two TIFs attached to intervals $(b, m]$ and $(m, e]$, respectively:

$$(\boldsymbol{f} \oplus_m \boldsymbol{g})(b,e,\mathbf{u}) = \boldsymbol{f}(b,m,\mathbf{u}) \oplus_m \boldsymbol{g}(m,e,\mathbf{u}). \tag{19}$$

Implementation Example

To give an idea of how one could implement GTIFs in a general programming language, some aspects of a realization in Common Lisp (Steele 1990) are presented below.

An implementation of timing functions consists of a number of constructs to define (e.g., `make-tif`), compose (e.g., `compose-tif`), concatenate (e.g., `concatenate-tif`) and evaluate (e.g., `tif-funcall`) the different types of timing functions (i.e. TMs, TIFs, and GTIFs). A complete implementation cannot be presented here, but I will give an example of a GTIF definition to illustrate the actual communication of timing information. (A micro-version implementation of timing functions is available online as an extension to GTF [Desain and Honing 1992; Honing 1995] at www.nici.kun.nl/mmm).

Figure 7 shows an example in Common Lisp of a timing function definition. It shows the constructor functions `make-tif` (Equation 13), `anonymous-gtif` (Equation 17), and `anonymous-tm` (Equation 7). The latter two are equivalent to lambda abstraction) as well as the evaluation function `tif-funcall` (Equation 14). In the section labeled `<body>`, the actual definition (e.g., a model describing how timing is dependent on global tempo and its metrical position) can be placed. These functions (TIFs) can be directly expressed in terms of score-time (begin, end, and duration) and performance-time (begin, end, and duration) of the temporal interval over which the they are defined.

Having this information available, for example, a TM describing how a jazzy groove pattern is related to the metrical structure and the current tempo can be expressed by the time-shift component of a GTIF. It will have access to all previously applied tempo transformations (i.e., the tempo component of $\mathbf{u}$) and can adapt accordingly. As another example, models of expressive tempo change that are stated in terms of metrical position (e.g., Clynes 1995) or position in the phrase structure (e.g., Todd 1992) can be expressed in the tempo-change component of a GTIF. While most these expressive timing models do not state how they should change with, for example, global tempo, the formalism in principle allows for this and can support the way these partial models can be combined.

Related Work

Below I will summarize related work on the representation of timing and tempo in both the computer music and music cognition research communities.

Computer Music Research

The representation of musical time has been a topic of numerous proposals in music representation research (see Roads 1996; Dannenberg 1993; Honing 1993), and a number of formalisms have been proposed. Rogers, Rockstroh, and Batstone (1980) introduces a way to express tempo changes as a function of beat position (score-time) to real-time (i.e., perfor-

Figure 7. Code example (in Common Lisp) of a timing function definition (see text for details).

```
(anonymous-gtif (b e u)
  "Return a timing function, a tuple of a time-shift and tempo-change function."
  (let ((pb (tif-funcall u b b))
        (pe (tif-funcall u e e)))
        ;; bind pb and pe lexically, the current performance times given
        ;; the previously applied timing transformations (u).
    (make-tif
        ;; Construct a TIF consisting of two time-maps.
        :time-shift   (anonymous-tm (s p)
                        ;; Return a time-shift time-map,
                        ;; a function of score-time (s) and performance-time (p),
                        ;; with access to score-begin (b), score-end (e),
                        ;; performance-begin (pb) and performance-end (pe), and
                        ;; previous applied timing transformations (u).
                        <body>) ; definition of a time-map.
        :tempo-change (anonymous-tm (s p)
                        ;; Return a tempo-change time-map,
                        ;; with access to score and performances times (as above)
                        <body>)) ; definition of a time-map.
   ))
```

mance-time). Jaffe (1985) presents similar ideas and promotes the use of a *time-map* (as an alternative to tempo functions). Anderson and Kuivila (1990) introduce tempo functions called *time-deformations* that can be concatenated. However, their tempo functions and time-shift functions (called *pause*) are not integrated into one representation, score-time is lost in composition (like time-maps), and behavior under transformation is not supported.

Dannenberg (1997) proposes a generalization of his earlier work on the Nyquist system for sound synthesis and composition. This research stresses the importance of behavior under transformation (referred to as "behavioral abstraction") and proposes an integrated representation for discrete events and continuous signals called *time-warping*. It supports time-shift and tempo transformations (called *shift* and *stretch*). However, the relationship between time-warps and temporal structure (e.g., allowing time-warps to access duration) is not recognized because of intrinsic design decisions in Nyquist (duration is not available because of its causal, real-time design; see Honing 1995).

Next to the formalization of timing and tempo, various computer music systems have been proposed that attempt to model these phenomena (e.g., Scheirer 1998; Cemgil, Kappen, Desain, and Honing in press). Most of these systems use either tempo-curves, time-shift functions, or time-maps as their underlying representation.

Music Cognition

The representation of time is also an important issue in music perception and cognition research. In the cognitive sciences, several proposals have been made, especially in the domain of temporal logic (van Benthem 1991), that allow reasoning about events occurring in time, resulting in proposals that promote the representation of time as intervals (Allen and Ferguson 1994) or as points (McDermott 1982; see Marsden 2000 for an overview of applying temporal logic to music). These proposals are all essentially discrete.

In music representation research, however, symbolic and numerical descriptions in both the discrete and continuous domain are needed (like discrete notes and their rhythmic structure, contrasted with continuous descriptions of timing, for example), which must be integrated into one representation (see Dannenberg, Desain, and Honing 1997).

Also in music perception a distinction is made between the discrete rhythmic durations, as symbolized by the note values in a score, and the continuous timing variations that characterize an expressive performance (Clarke 1999). A listener is able to separate the temporal information of, for example, an expressively performed rhythm into note durations, expressive timing, and tempo information.

The knowledge representation proposed in this article makes these three aspects explicit by introducing a way in which timing can be expressed in terms of the temporal structure and global tempo (see Figure 1). Secondly, this representation differentiates two components of musical time: tempo, the perception of change of rate related to a process called beat-induction (see Desain and Honing 1999), and timing, the perception of the minute time deviations related to categorical rhythm perception (see Clarke 1999).

Whereas in the music performance literature there exists some discussion of the specific shape of tempo functions in particular and their relationship to human motion (Todd 1999; Friberg and Sundberg 1999; Desain and Honing 1996), the proposed formalism does not make any restrictions as to their specific shape: these models can all be represented in the tempo component of a timing function. Yet the representation stresses the importance of types of timing that are relatively independent of tempo change, and it allows for a description of how these types of timing interact (for example, how "laid-back" timing in a jazz fragment, expressed in the time-shift component of a timing function, should adapt itself to the current global tempo, expressed in the tempo-change component of a timing function).

However, it should be noted that it is still unclear whether the perception of tempo and timing are two separate perceptual processes or one and the same, and what the precise cognitive constraints on the perception of timing are (e.g., Repp 1992). There is a continuing effort to understand what precisely constitutes tempo, how timing is dependent on global tempo or absolute rate, and how it is perceived and performed (Palmer 1997; Gabrielsson 1999), as well as a discussion on how to computationally model these phenomena (see Desain, Honing, van Thienen, and Windsor 1998).

Summary and Conclusion

The first half of this article reviewed existing representations of timing and tempo common in computational models of music cognition and in programming languages for music. A formal analysis revealed their differences, and some refinements were proposed. The second half of the article introduced a knowledge representation of musical time that differs in two important aspects from earlier proposals. First, timing is seen as a combination of a tempo component (expressing the change of rate over a fragment of music, such as *tempo rubato*), and a timing (or time-shift) component that describes how events are timed (e.g., early or late) with respect to this tempo description. Second, timing can be specified in relation to the temporal structure (e.g., position in the phrase or bar), as well as performance-time, score-time, and global tempo.

However, the proposed representation covers only a part of the timing phenomena observed in music performance, concentrating on a continuous description of onset-timing. For instance, asynchrony, like chord spread, is not explicitly supported (functions that map one score-time to different performance times), neither is articulation (offset timing and its relationship to musical streams and structure). These are relatively complex aspects of timing that are still little understood. These will be topic of further research and future extensions.

Acknowledgments

Special thanks to Peter Desain for, as always, inspiring discussions and substantial help in the design of timing functions. I thank Roger Dannenberg for many stimulating discussions on this and related work. Renee Timmers commented on an earlier version of this article, improving its presentation. A version of this article was presented in 1995 at the IBM T. J. Watson Research Center by kind invitation of the Mathematical Science department. This research has been made possible by the Netherlands

Organization for Scientific Research (NWO) as part of the "Music, Mind, Machine" project.

References

Allen, J. F., and G. Ferguson. 1994. "Actions and Events in Interval Temporal Logic." *Journal of Logic and Computation* 4(5):531–579.

Anderson, D. P., and R. Kuivila. 1990. "A System for Computer Music Performance." *ACM Transactions on Computer Systems* 8(1):56–82.

Ashley, R. 1996. "Aspects of Expressive Timing in Jazz Ballad Performance." In B. Pennycook and E. Costa-Giomi, eds. *Proceedings of the 4th International Conference on Music Perception and Cognition.* Montreal, Canada: Faculty of Music, McGill University, pp. 485–490.

Bilmes, J. 1993. "A Model for Musical Rhythm." *Proceedings of the 1993 International Computer Music Conference*. San Francisco: International Computer Music Association, pp. 207–210.

Cemgil A. T., B. Kappen, P. Desain, and H. Honing. In Press. "On Tempo Tracking: Tempogram Representation and Kalman Filtering." *Journal of New Music Research.*

Clarke, E. F. 1999. "Rhythm and Timing in Music." In D. Deutsch, ed. *Psychology of Music*. New York: Academic Press.

Clynes, M. 1995. "Microstructural Musical Linguistics: Composer's Pulses Are Liked Best by the Best Musicians." Cognition 55:269–310.

Dannenberg, R. B. 1993. "Music Representation: Issues, Techniques, and Systems." *Computer Music Journal* 17(3):20–30.

Dannenberg, R. B. 1997. "Abstract Time Warping of Compound Events and Signals." *Computer Music Journal* 21(3):61–70.

Dannenberg, R. B., P. Desain, and H. Honing. 1997. "Programming Language Design for Music." In G. De Poli, A. Picialli, S. T. Pope, and C. Roads, eds. *Musical Signal Processing*. Lisse: Swets and Zeitlinger, pp. 271–315.

Desain, P., and H. Honing. 1991. "Tempo Curves Considered Harmful: A Critical Review of the Representation of Timing in Computer Music." *Proceedings of the 1991 International Computer Music Conference*. San Francisco: International Computer Music Association, 143–149.

Desain, P., and H. Honing, H. 1992. "Time Functions Function Best as Functions of Multiple Times." *Computer Music Journal* 16(2):17–34.

Desain, P., and H. Honing. 1993. "Tempo Curves Considered Harmful." *Contemporary Music Review* 7(2):123–138.

Desain, P., and H. Honing. 1994. "Does Expressive Timing in Music Performance Scale Proportionally with Tempo?" *Psychological Research* 56:285–292.

Desain, P., and H. Honing. 1996. "Physical Motion as a Metaphor for Timing in Music: The Final Ritard." *Proceedings of the 1996 International Computer Music Conference*. San Francisco: International Computer Music Association, 458–460.

Desain, P., and H. Honing. 1999. "Computational Models of Beat Induction: The Rule-Based Approach." *Journal of New Music Research* 28(1):29–42.

Desain, P., H. Honing, H. van Thienen, and W. L. Windsor. 1998. "Computational Modeling of Music Cognition: Problem or Solution?" *Music Perception* 16(1):151–166.

Friberg, A., and J. Sundberg. 1999. "Does Music Performance Allude to Locomotion? A Model of Final Ritardandi Derived from Measurements of Stopping Runners." *Journal of the Acoustical Society of America* 105(3):1469–1484.

Gabrielsson, A. 1999. "Music Performance." In D. Deutsch, ed. *Psychology of Music*. New York: Academic Press.

Honing, H. 1993. "Issues in the Representation of Time and Structure in Music." *Contemporary Music Review* 9:221–239.

Honing, H. 1995. "The Vibrato Problem: Comparing Two Solutions." *Computer Music Journal* 19(3):32–49.

Jaffe, D. 1985. "Ensemble Timing in Computer Music." *Computer Music Journal* 9(4):38–48.

Marsden, A. 2000. *Representing Music Time: A Temporal-Logic Approach*. Lisse: Swets and Zeitlinger.

McDermott, D.V. 1982. "A Temporal Logic for Reasoning about Processes and Plans." *Cognitive Science* 6:101–155.

Palmer, C. 1997. "Music Performance." *Annual Review of Psychology* 48:115–138.

Repp, B. H. 1992. "Probing the Cognitive Representation of Musical Time: Structural Constraints on the Perception of Timing Perturbations." *Cognition* 44:241–281.

Roads, C. 1996. *The Computer Music Tutorial*. Cambridge, Massachusetts: MIT Press.

Rogers, J., J. Rockstroh, and P. Batstone. 1980. "Music-Time and Clock-Time Similarities under Tempo Changes." *Proceedings of the International Computer Music Conference*. San Francisco: International Computer Music Association, pp. 404–442.

Scheirer, E. D. 1998. "Tempo and Beat Analysis of Acoustic Musical Signals." *Journal of the Acoustical Society of America* 103(1):588–601.

Steele, G. L. 1990. *Common Lisp, the Language.* Bedford, Massachusetts: Digital Press.

Sundberg, J. 1988. "Computer Synthesis of Music Performance." In J. Sloboda, ed. *Generative Processes in Music: The Psychology of Performance*. Oxford: Clarendon, 52–69.

Todd, N. P. M. 1992. "The Dynamics of Dynamics: A Model of Musical Expression." *Journal of the Acoustical Society of America* 91(6):3540–3550.

Todd, N. P. M. 1999. "Motion in Music: A Neurobiological Perspective." *Music Perception* 17(1):115–126.

van Benthem, J. 1991. *The Logic of Time.* Dordrecht: Reidel.