Segments and Mapping for Scores and Signal Representations

Dominique Fober[1], Frederic Bevilacqua[2], Roland Assous[3]

[1]*GRAME - Lyon,* [2]*IRCAM - Paris,* [3]*Université Lyon 1*

Abstract

We present a general theoretical framework to describe segments and the different possible mapping that can be established between them. Each segment can be related to different music representations, graphical scores, music signals or gesture signals. This theoretical formalism is general and is compatible with large number of problems found in sound and gesture computing. We describe some examples we developed in interactive score representation, superposed with signal representation, and the description of synchronization between gesture and sound signals.

1   Introduction

The explicit or implicit use of segments is necessary for the description of music. For example, elements such as notes, bars, sections can be considered as time segments. Beyond these obvious examples, symbolic and signal representations of music, including sound and gestures elements, do imply a large variety of segmental representations. Generally, the different applications, from musical notation to interactive system programing, make use of their own *ad-hoc* formalisms. We argue here that a general formalism would facilitate the integration of different segmental approaches for both symbolic and signal music representations. This should foster merging different computer music applications (or the communication between them), especially considering interactive sound systems dealing with graphical representations.

Music representation is fundamentally not linear with respect to the time representation. The possible relationships between the "dynamic time" of the performance and the time of the score can be complex, being potentially non-continuous and non-bijective. In music notation, repeat bars, jumps to *segno* or *coda* are some examples taken from the traditional western music notation, illustrating such a complexity. The dynamic structure of jazz music poses similar issues, making automatic accompaniment a non-trivial task [1, 2].

Music representation, beyond the sole symbolic representation, should also include signal representation. In this regard, the problem of relationship we just mentioned should be generalized to the relationships between various classes of elements, symbol and signal segments.

For example, the classic case of Attack-Decay-Sustain-Release note representation in sound synthesis makes use of concatenated segments for control, that are related to sound processes. This could be generalize to more general gesture modeling in music performance, which can be segmented in preparation, stroke and release phases [3]. Formalizing relationships between such gesture and sound segments represents a current challenge in the design of interactive systems [4, 5].

This paper proposes a simple but general formalism to describe time segments in various spaces. Specifically, we propose an explicit description of the relationships between segments. We will show in particular how this formalism can be used for a wide range of applications, and in particular in dynamic graphical scores and gesture-based interactive systems.

The paper is structured as follows. First, after a presentation of related works, we describe the general problematics of time representations we tackle in this paper. Second, we present our general framework for segments description and segmentation. Third, we describe a formal description of *mappings* which is the term we'll use to refer to relations between segments, and fourth, how these mappings can be extended to *continuous mappings* using simple and intuitive operations. Fifth, we describe more advanced concepts of segments relationships, called *refinement*. All the formal descriptions are accompanied by concrete examples of applications.

2   Related Works

Let consider first the case of music notation. Music scores could be viewed as time representation: for most of the music representations, there is a more or less complex relationship between the graphical space of the notation and the performance time. Such relationships between the "music/performance time" and the music notation are made explicit in some applications.

For example, in [6], the authors discussed the relationship, in graphical space, between time representation in notation and in music performance. Intended to enhance music learning using a mirror metaphor, this work did not attempt to provide for a formal description of the relationship between time and graphic spaces, being rather focused on specific required features for pedagogical applications.

In [7], the authors described a specific time representation, used in music composition tools such Open-

Music, to overcome issues arisen when combining different notation aspects with advanced time structures. An object called "sheet" is introduced in OpenMusic, providing a consistent graphical representation of different musical objects regarding their time dimension. This approach makes use of landmarks to compute ratios between time and space, that are gathered into a global discrete function $x = f(t)$. Nevertheless, this system is designed only for specific objects considering their time systems (e.g. proportional-time, continuous-time objects). It cannot be generalized easily being based on a restricted formalism of time segments.

Representing time relations was partially approached already in the early 90' by the HyTime proposal [8], with the viewpoint of media synchronization but the HyTime initiative never succeeded [9]. In some way, the SMIL specification [10] could be viewed as the successor of HyTime: it is a recommended W3C XML markup language for describing multimedia presentations. Using SMIL, it is possible to describe the temporal behavior of a multimedia presentation, to associate hyperlinks with media objects and to describe the layout of the presentation on a screen.

Another approach has been proposed with the IEEE standard IEEE 1599 [11], which targets applications in which all aspects of music, such as audio and sound, graphical representation, historical data, performance indications, are fully integrated and synchronized. This standard specifies an XML description of multimedia resources, which are all related to an abstract time layer. It contains the necessary information to establish time relationships between various media using time as the *spine* of the description. It is strongly anchored in the music domain and even provides users with a symbolic notational layer.

Hence there are numerous approaches and formats to describe time relationships of which none succeeded in being widely supported. But above all, the existing approaches are mainly descriptive and lack of a formal description of the time relationships that enables a computational approach.

The recent extension of the music score by [12] led to a general formalism, covering the needs of media synchronization beyond the current state of the art. The synchronization issues are approached using *segmentation* and relations between *segments*. Similarly to previous approaches, time constitutes the spine of the system but the specification is independent of the media types and can be easily extended. The formalism we present in the next sections builds on this earlier work, describing a complete formal description of segment and their relationships.
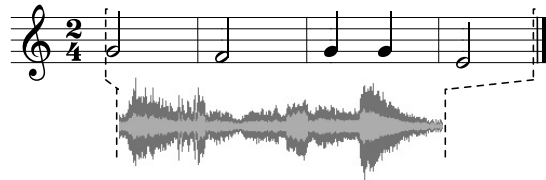

Figure 2: A music score with annotation consisting of a performance represented as a graphic signal. A simple stretch of the annotation would not suffice to represent accurately the relationship between the score and the annotation.

3 Problematics of time representations and their relationships

3.1 Music notation, annotation and *time* space.

Apart the purely graphic notations experiments conducted in the second half of the XXth century [13, 14], a music score generally establishes correspondences between the graphical space and the time space.

For the sake of simplicity, most music applications using a score representation associate a single time position to a graphic position. Nevertheless, even simple music score might imply a graphical space of the music notation that is not organized proportionally to time (Fig. 1). For example, associating a staff to a single time interval generally leads to rough approximations of the actual relationship between time and the graphical space. To ensure accuracy to such a relationship, a finer segmentation of the graphical space is necessary, taking into account every symbol that can be associated to a time dimension. Another common issue is due to the complex relationship existing between the notation and the dynamic time of the performance: this relationship is non-continuous and non-bijective, due to repeat bars, jumps (to *coda*, to *segno*...). Figure 1 illustrates the two issues we mentioned: non-proportionality between time and the graphical space and non-bijective relationship between notated and performed time.

Music score annotation (e.g. a score annotated with textual data, or with a representation of its performance) is another case that requires an accurate relationship between time and space. In the case of static scores (typically printed), the annotation can be added directly to the graphic space without considering its precise time location, the choice of the graphical location being often constrained by readability issue. In the case of dynamic scores (e.g. interactive score), a more formal procedure must be established to guarantee consistency to changes, by taking into account a precise time for the annotation and for the graphical placement of annotation.

Even more complex cases are found when the annotations have also a time dimension (e.g. a signal graphic representation). Figure 2 shows an example of a music score and its performance represented as a graphic signal: a simple stretch of the performance image over the score cannot visually convey accurately the detailed time relationships between the score and the performance.

3.2 Describing time relationships in graphical space

Beyond standard music scores, there is a need for a general system that can describe time relationships

Figure 1: A music score illustrating two issues : non-proportionality between time and the graphical space and non-bijective relationship between notated and performed time (due to repeated bars).

for arbitrary graphical objects. For example, it is useful to combine symbolic scores and various signal representations of sound and gesture elements. This becomes especially important for interactive systems, where these relationships are dynamic, as their graphical representations. We present below two different examples taken from actual interactive applications.

First, Figure 3 illustrates an example where temporal elements such as sound and gesture signal representations are linked to a symbolic score. The gesture signal here corresponds to the 3D acceleration of the hand of a conductor, displayed above the audio waveform. This case is taken from an existing pedagogical system [15] where a teacher or a student control the pace of an audio recording using "conducting gestures", captured with motion sensors. In such an application, it is vital to be able to visualize the symbolic scores synchronously with the signal representations, and to be allowed to draw direct links between them as indicated in Figure 3.

Second, we present here an example taken from a series of experiments performed on an augmented quartet, where bow acceleration was measured synchronously with the sound [16]. Figure 4 displays the score, sound acceleration signals of a particular musical phrase. Interestingly, in this case the score includes relatively complex ornamentation, with no explicit time duration, which also exemplifies the complexity of the time relationships between symbolic and signals representations.

Figure 4 shows that gesture and sound signals can be segmented differently. The different forms of the signals naturally call for different segmentation strategies. First, the gesture signals includes gesture preparation signals that are segments not present in sound signals. Second, segments are typically of different

durations for a given element of the symbolic score. Moreover, this might lead to different elements grouping, resulting finally in structurally different segmentations.

The next sections present the formalism that has been defined to handled the problematic described above. First, segments and segmentations are formally defined. Then the notion of *mapping* (i.e. relationship between segmentations) is introduced. Next, mappings are extended to *continuous mapping*, which represents a way to draw relationships inside a segment. Finally, a *refinement* operation is defined, that is used to build relationships between arbitrary segmentations.

4 Segments and segmentations

In this section we define the notions of *segment* and *segmentation*. We will first introduce the notions of temporal and graphic segments. Next we will generalize these definitions for any abstract space. Finally, we introduce the notion of *resource segmentation*, that is a necessary step to coherently define segment relationships within our formalism.

4.1 Temporal Segment

A *temporal segment* is defined as an interval $i = [t_0, t_1[$ such as $t_0 \leqslant t_1$. The interval is empty when $t_0 = t_1$; we will use the notation $\oslash$ for empty intervals.

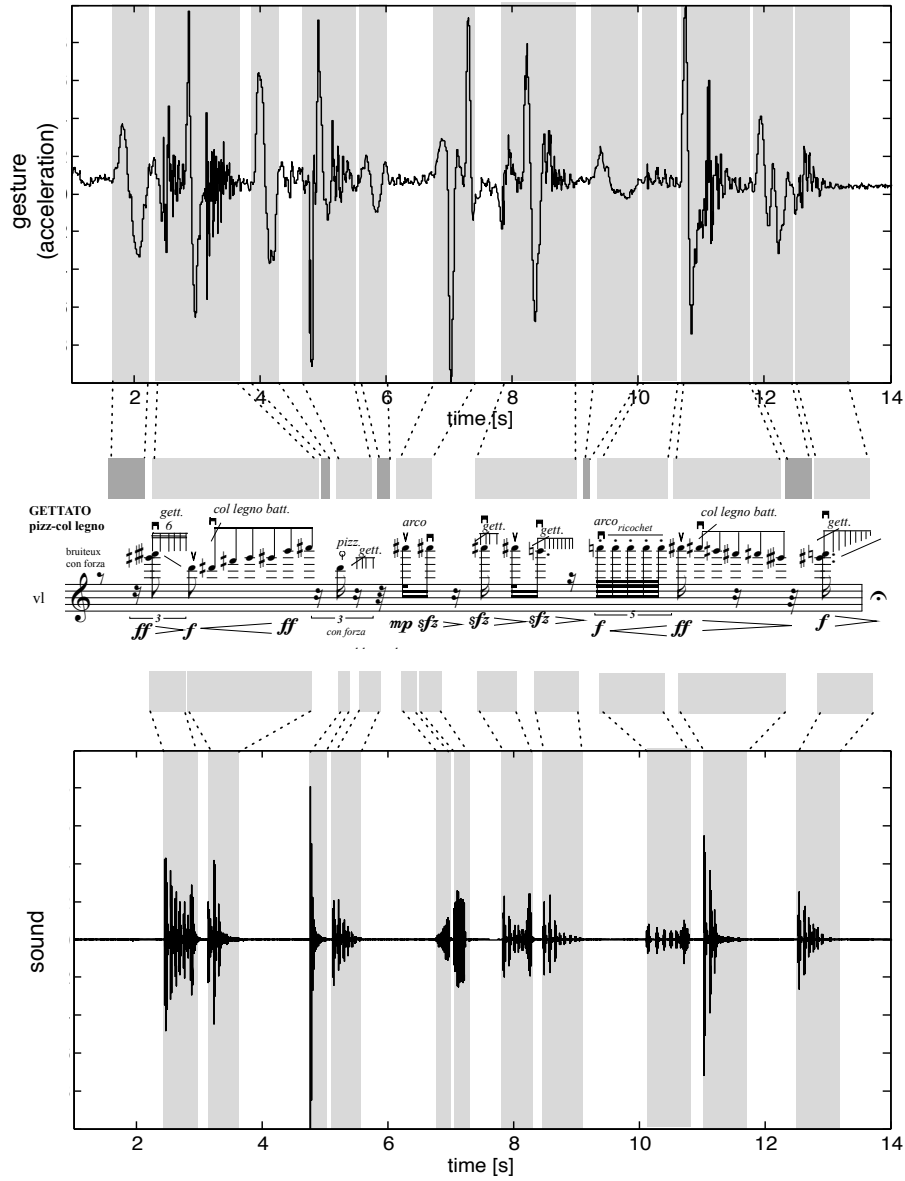The intersection of two temporal segments (figure 5)

Figure 4: Relationship between different segmentation related to gesture or sound signals. Case of an augmented violin, where bow acceleration was recorded synchronously with the audio. Excerpt of the composition *Streicherkreis* by Florence Baschet
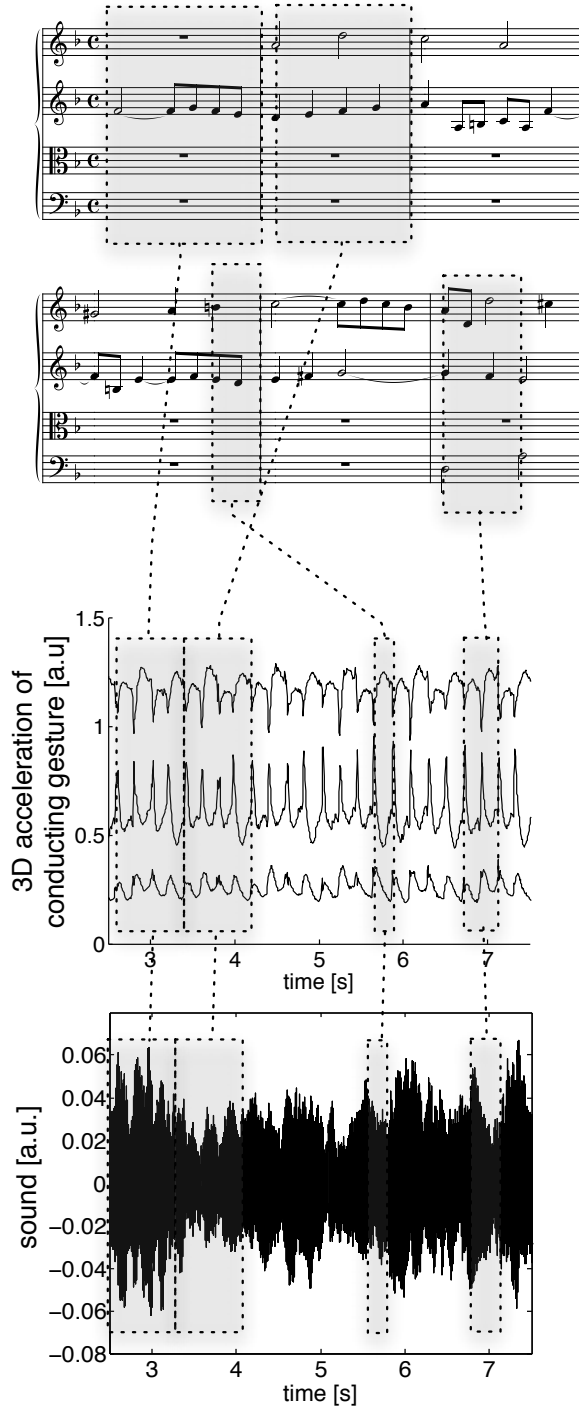
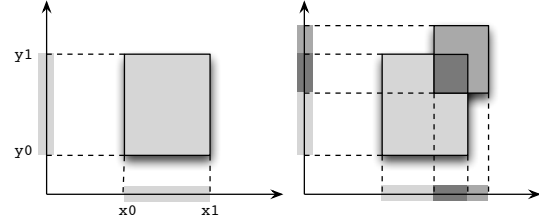Figure 5: from left to right: a temporal segment and the intersection of two temporal segments.

Figure 6: from left to right: a graphic segment and the intersection of two graphic segments.

i_m and i_n is defined by:

$$i_m \cap i_n := \{j \mid j \in i_m \;\wedge\; j \in i_n\}$$

4.2 Graphic Segment

A *graphic segment* (2D case) is defined as the product of two intervals $[x_0, x_1[$ et $[y_0, y_1[$:

$$[x_0, x_1[\times [y_0, y_1[:= \{(u, v) | u \in [x_0, x_1[\wedge v \in [y_0, y_1[\}$$

where $[x_0, x_1[$ is an interval on the abcissa and $[y_0, y_1[$, on the ordinate (figure 6).

A graphic segment $g = x \times y$ is empty when $x = \oslash$ or $y = \oslash$

The intersection operation $\cap$ between two graphic segments $g_m = x_m \times y_m$ and $g_n = x_n \times y_n$ is defined as :

$$g_m \cap g_n := (x_m \cap x_n) \times (y_m \cap y_n)$$

Figure 3: Segments in score, sound and gesture signal representation. Case of conducting gestures, captured with 3D accelerometer, used to continuously control the pace of a recording.

4.3 Segment Generalization

More generally, the notions of temporal and graphic segment can be extended to any dimension n. A n-dimensional segment s is defined as the cartesian product of n intervals $s = i_1 \times ... \times i_n$. When one of these intervals is empty, then s is empty.

The intersection of n-dimensional segments is defined as:

$$s_1 \cap s_2 := (i_1 \cap j_1) \times ... \times (i_n \cap j_n) \qquad (1)$$

where $s_1 = i_1 \times ... \times i_n$ et $s_2 = j_1 \times ... \times j_n$

4.4 Resource Segmentation

By definition, a *resource* is a non empty n-dimensional segment. The segmentation of a resource R, defined on a segment S, is a set of disjoined segments $Seg(R) = \{s_1, ..., s_p\}$ all included in the resource:

$$\begin{cases} \forall i \quad 1 \le i \le p \quad s \subseteq S \\ \forall i, j \quad 1 \le i \le p, \quad 1 \le j \le p \\ \qquad i \ne j \implies s_i \cap s_j = \oslash \end{cases} \qquad (2)$$

The segmentation *domain* is defined as the union of its segments :

$$\mathcal{D}om(Seg(R)) = \bigcup_{i=1}^{p} s_i$$

Note that a segmentation is generally partial, it is not necessary a tessellation of the resource.

5 Mapping

A *mapping* is a binary relationship between *segmentations*, defined by a subset $M \subseteq Seg(R_1) \times Seg(R_2)$. We define the function $M^+ : Seg(R_1) \rightarrow 2^{Seg(R_2)}$ that gives the set of segments from R_2 associated to a segment s from R_1 :

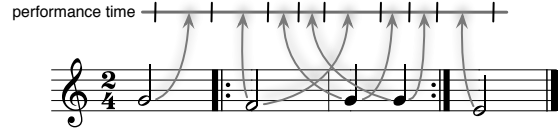$$M^+(s) := \{s' \in Seg(R_2) \mid (s, s') \in M\} \qquad (3)$$

Figure 7: A music score including repeat bars: there is a non-bijective relationship between the notated and performance time.

Similarly we define the function $M^- : Seg(R_2) \rightarrow 2^{Seg(R_1)}$ that gives the set of segments from R_1 associated to a segment s' from R_2.:

$$M^-(s') := \{s \in Seg(R_1) \mid (s, s') \in M\} \qquad (4)$$

Mappings can catch any musical structure as shown by the example below.

5.1 Time to time mapping

A *time to time* mapping catch the overall music structure of a score and its relationship to the performance time. Consider the score as illustrated in figure 7, which includes repeat bars. The relationship between the score time and the performance time can be described by the following relations set (assuming that the repeat section is played twice) :

```
( [0/1, 1/2[ ) → ( [0/1, 1/2[ )
( [1/2, 3/2[ ) → ( [1/2, 3/2[ )
( [1/2, 3/2[ ) → ( [3/2, 5/2[ )
( [3/2, 4/2[ ) → ( [5/2, 6/2[ )
```

where each line associates a score time segment to a performance time segment. Time intervals are expressed as rational values denoting music time, where 1 is the whole note.

A time to time mapping is flexible enough to describe any musical structure, being not constrained to standard musical segmentation (e.g. measures).
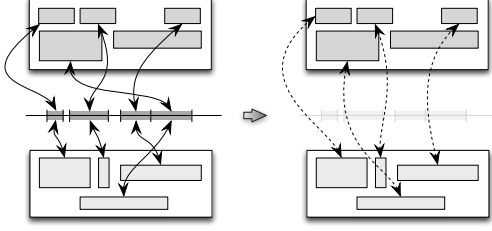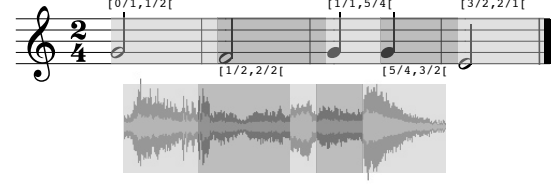
Figure 8: Mappings composition.

Figure 9: Graphic segments of a score and its performance displayed using grey levels and annotated with the corresponding time segments.
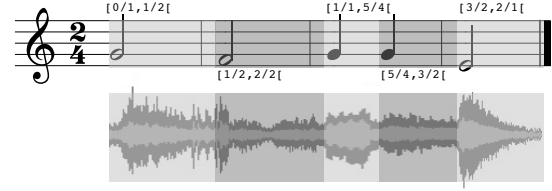

Figure 10: A score and its performance synchronized according to their time relationship using mappings composition.

5.2 Mappings Composition

Composition of mappings is quite straightforward (figure 8): using the mappings

$$M_1 \subseteq Seg(R_1) \times Seg(R_2)$$
$$\text{and } M_2 \subseteq Seg(R_2) \times Seg(R_3),$$

$M_1 * M_2$ is the mapping defined on $Seg(R_1) \times Seg(R_3)$ by:

$$(s, u) \in M_1 * M_2 \Leftrightarrow$$
$$\exists t \in Seg(R_2) \mid (s, t) \in M_1 \ \wedge \ (t, u) \in M_2 \qquad (5)$$

Mappings composition allows to draw relationships between arbitrary segmentations, provided they have a relationship to a common space (e.g. the time space). This is illustrated in figure 8, where relationships between two graphic spaces are obtained from their relationships to the time space.

5.3 Time synchronization in the graphical space

Mapping compositions can express time relationships in the graphical space. Let us consider the problem of a score and the graphical representation of its performance, as previously described. Figure 9 shows the different graphic segments of a score, annotated with the corresponding time segments, along with a graphic signal as a representation of a performance, also segmented in the graphic and time spaces.

Let $M_s \subseteq Seg(Score_g) \times Seg(Score_t)$ be the mapping between the score graphic and time segmentations.

Let $M_p \subseteq Seg(Perf_t) \times Seg(Perf_g)$ be the mapping between the performance time and graphic segmentationss.

The composition $M_s * M_p$ is a set of relationships from the score graphic space to the signal graphic space, directly expressing the time relationships of the two objects as illustrated in figure 10.

6 Continuous Mapping

We introduce the notion of *continuous mapping*, that expresses relationships that are not explicitly described but that can be computed from a given mapping. For a relationship $(s, g) \in M$ a *continuous mapping* describes the relationships of any segment $s' \subset s$.

We first introduce the notions of intervals and segments *varieties*. Next, we will define *congruent varieties*, which provide for the ground for continuous mapping.

6.1 Interval Variety

For $\theta : [0,1] \to [0,1]$ and $I = [a,b[$, we name variety of I, the set of points $\mathcal{V}(I,\theta)$ from I defined by :

$$\mathcal{V}(I,\theta) = \{ (1-\theta(t)).a + \theta(t).b \mid t \in [0,1[\} \quad (6)$$

Intuitively, the variety of an interval expresses the relationship between this interval and its variety using a function θ defined on $[0,1[$.

6.2 Segment Variety

The variety of a segment generalizes the variety of an interval to a list of intervals. It is defined as the list of the varieties of each interval.

Considering a segment $S = (i_1, ..., i_n)$ and $\Theta : n \to (\theta_1, ..., \theta_n)$, the variety $\mathcal{V}(S,\Theta)$ of S is defined as:

$$\mathcal{V}(S,\Theta) = (\mathcal{V}(i_1,\theta_1), ..., \mathcal{V}(i_n,\theta_n)) \quad (7)$$

More generally, we define the variety $\mathcal{V}(S,\Theta^m)$ for $m \neq n$.

$$
\begin{cases}
\mathcal{V}(S,\Theta(m)) = (\mathcal{V}(i_1,\theta_1), ..., \mathcal{V}(i_m,\theta_m), \\
\qquad \mathcal{V}(i_{m+1},id), ..., \mathcal{V}(i_n,id)), \quad m < n \\
\mathcal{V}(S,\Theta(m)) = (\mathcal{V}(i_1,\theta_1), ..., \mathcal{V}(i_n,\theta_n)), \quad m > n
\end{cases}
$$
$$(8)$$

It consists respectively in the extension of the list of functions θ_i from the dimension m to n using the identity function and to the reduction of the list of functions θ_i to the dimension n.

Generally, we will note $\mathcal{V}(S,\Theta)$ to refer to a variety of S, whatever the dimensions of S and Θ.

6.3 Congruent varieties

Two varieties $\mathcal{V}(S,\Theta)$ and $\mathcal{V}(T,\Theta')$ are said congruent when $\Theta = \Theta'$.
We will use the notation $\mathcal{V}(S,\Theta) \equiv \mathcal{V}(T,\Theta')$ to express congruence.

Intuitively, congruent varieties express the fact that the segments proportions and position relationships are equal. It also provides information about the relations between points enclosed in congruent varieties.

Let's consider 2 time intervals $I = [t_1, t_2[$ and $I' = [t'_1, t'_2[$ and a relationship between I and I'. Congruent varieties $\mathcal{V}(I,\Theta)$ and $\mathcal{V}(I',\Theta)$ express how we go through I' when we go through I since

$$(1-\theta(n)).t_1 + \theta(n).t_2 = (1-\theta(n)).t'_1 + \theta(n).t'_2 \mid n \in [0,1[$$

In particular, the relationship between points from I and I' is deduced by linear interpolation when $\theta = id$.

The introduction of a function $\theta \neq id$ to go through a segment corresponds to the typical case of an accelerando, where the acceleration is a not a linear function of time.

6.4 Continuous Mapping Definition

We can now define the continuous mapping. For any mapping $M \subseteq Seg(R_1) \times Seg(R_2)$, we associate a mapping between varieties called *continuous mapping* associated to M as follows:

$$M_{\mathcal{V}}^+(\mathcal{V}(s,\theta)) = \{\mathcal{V}(s',\theta) \mid (s,s') \in M\} \quad (9)$$

$$M_{\mathcal{V}}^-(\mathcal{V}(s',\theta)) = \{\mathcal{V}(s,\theta) \mid (s,s') \in M\} \quad (10)$$

In other words: when two segments are in relationship, then their congruent varieties are also in relationship (figure 11).

7 Segmentations refinement

The mappings composition makes explicit relationships when the resources share the same intermediate segmentation (for example, identical segmentation of the time space). In practice, we often have different segmentations of identical spaces, which cannot
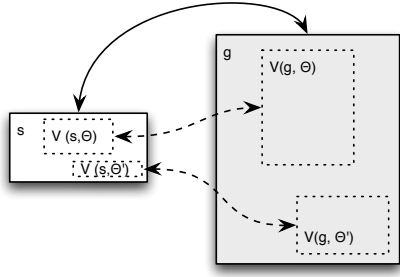
Figure 11: According to the relation between the segments s and g, the continuous mapping expresses relations between included segments using congruent varieties, denoted by the dotted lines.
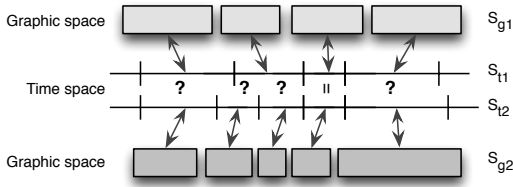

Figure 12: Different segmentations of the same temporal space preventing the composition of the mappings $M \subseteq S_{g1} \times S_{t1}$ and $M' \subseteq S_{t2} \times S_{g2}$.

be *composable* (figure 12), although we have the intuition that such relationship could be defined. To solve this problem, we build a common segmentation by *refinement*.

Let's consider a resource R and two segmentations S and S' defined on R, we define the relationship denoted $\preccurlyeq$ to express that S' is *finer* than S (every segment of S' is included in a segment of S) :

$$S' \preccurlyeq S \quad \Leftrightarrow \quad \forall s' \in S' \quad \exists\, s \in S \quad s' \subset s$$

This relationship could be viewed as a variety of s:

$$S' \preccurlyeq S \quad \Leftrightarrow \quad \forall s' \in S'$$
$$\exists\, (s,\theta) \in S \times [0,1[^{[0,1[} \quad s' = \mathcal{V}(s,\theta) \qquad (11)$$

Note that two segmentations S_1 and S_2 defined on the same resource R have always a common partial refinement S on their intersection when this intersec-
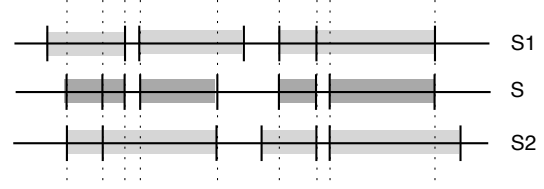

Figure 13: The refinement S of the two segmentations $S1$ and $S2$.

tion is not empty (figure 13) satisfying:

$$\begin{cases} \mathcal{D}om(S) = \mathcal{D}om(S_1) \cap \mathcal{D}om(S_2) \\ S \preccurlyeq S_1 \ \wedge \ S \preccurlyeq S_2 \end{cases}$$

The refinement common to two segmentations S_1 et S_2 may also be expressed as varieties:

$$S \preccurlyeq S_1 \ \wedge \ S \preccurlyeq S_2 \quad \Leftrightarrow \quad \forall s \in S$$
$$\exists\, (s_1,\theta) \in S_1 \times [0,1[^{[0,1[}$$
$$\exists\, (s_2,\delta) \in S_2 \times [0,1[^{[0,1[}$$
$$s = \mathcal{V}(s_1,\theta) = \mathcal{V}(s_2,\delta) \qquad (12)$$

7.1 Example 1: A sliding window with a time dimension

A basic use case of continuous mapping and refinement consists in a sliding window with a time dimension which could serve analysis or performance purposes. The window is aligned to a score to indicate the current time position and time extend as shown in figure 14, where the score time and graphic segmentations are as illustrated in figure 9. The window duration is $1/2$ (a half note) and its current time position ($3/4$) lies in the middle of the time segment $S_t = [1/2, 1/1[$.

The score time segmentation in the window neighborhood is $S = \{s_a = [1/2, 1/1[\quad s_b = [1/1, 5/4[\}$ and the window time segmentation is $C = \{[3/4, 5/4[\}$. We assume that we have the following time to gaphic mapping:
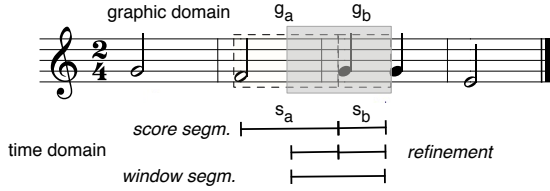
Figure 14: A window aligned to a score at date 3/4. Its duration is 1/2 (a half note). The corresponding time segments are shown with their refinement.

$$s_a \leftrightarrow g_a$$
$$s_b \leftrightarrow g_b$$

where g_a and g_b are gaphic segments.

The segmentation $S_R = \{[3/4, 1/1[\quad [1/1, 5/4[\}$ is a refinement of both S and C.

Using a function $\theta = id$ (i.e. a linear interpolation), the continuous mapping tells us that the first half of the window $[3/4, 1/1[$ is mapped to the second half of the score g_a graphic segment and the second half of the window $[1/1, 5/4[$ is mapped to the g_b graphic segment.

Using mappings and continuous mappings turns out to be a very convenient way to synchronize any object to a score without taking care of the score layout (e.g. without the need to detect whether there is a line break in the middle of my objet). Any object can be expressed in the temporal domain, the mapping and continuous mapping taking care of potential sub-segmentation with every corresponding graphic segments.

7.2 Example 2: Multiples Segmentations in Sound and Gesture Signal Representations

At the beginning of the paper 3.2, we presented an example where gesture and sounds signals were separately segmented and linked to the score (Fig. 4). We can see now how our general formalism can handle such a complex case. First, a comparison be-

tween Figure 4 and Figure 12 reveals clearly that the segmentation refinement is appropriate to formalize the different segmentations such as the gestural and sound segmentation, that operates in the on the same performance time.

8 Implementation

The mapping formalism has been implemented as a C++ library. It is part of the INScore open source project [1] where it provides time synchronization in the graphic space. The system automatically aligns and stretches synchronized objects to the corresponding score location using this library, to easily produce the kind of results illustrated in figure 10. Examples are given in [12][17].

9 Conclusion

We have proposed a simple formalism to describe relationships between segments. This formalism is general since it does not rely in any way on the segments content and semantic. As such, it has musical applications in the graphic domain of the score but also in the gestural or audio domains and allows to create links that would be otherwise complex to express. It also includes a formal description of operations that gives a high flexibility to the way relationships can be described, including how the inner space of a segment relates to another segment space.

10 Acknowledgments

---

[1] http://inscore.sf.net

References

[1] Nicolas E. Gold and Roger B. Dannenberg. A reference architecture and score representation for popular music human-computer music performance systems. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 36–39, Oslo, June 2011.

[2] Dawen Liang, Guangyu Xia, and Roger B. Dannenberg. A framework for coordination and synchronization of media. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 167–172, June 2011.

[3] Jules Françoise, Baptiste Caramiaux, and Frédéric Bevilacqua. A hierarchical approach for the design of gesture-to-sound mappings. In *Sound and Music Computing*, Copenhague, Danemark, Juillet 2012.

[4] Frédéric Bevilacqua, Norbert Schnell, Nicolas Rasamimanana, Bruno Zamborlin, and Fabrice Guédy. Online gesture analysis and control of audio processing. In Jorge Solis and Kia Ng, editors, *Musical Robots and Interactive Multimodal Systems*, volume 74 of *Springer Tracts in Advanced Robotics*, pages 127–142. Springer Berlin Heidelberg, 2011.

[5] N. H. Rasamimanana, F. Kaiser, and F. Bevilacqua. Perspectives on gesture-sound relationships informed from acoustic instrument studies. *Organised Sound (in press)*, 14(2):208 – 216, 2009.

[6] D. Fober, S. Letz, and Y. Orlarey. Vemus - feedback and groupware technologies for music instrument learning. In *Proceedings of the 4th Sound and Music Computing Conference SMC'07 - Lefkada, Greece*, pages 117–123, 2007.

[7] Jean Bresson and Carlos Agon. Scores, programs, and time representation: The sheet object in openmusic. *Computer Music Journal*, 32(4):31–47, 2008.

[8] Steven R. Newcomb, Neill A. Kipp, and Victoria T. Newcomb. The "HyTime ": hypermedia/time-based document structuring language. *Communications of the ACM*, 34(11):67–83, 1991.

[9] Donald Sloan. Learning our lessons from SMDL. In *Proceedings of MAX2002 - First Musical application using XML International conference*, pages 69–73, 2002.

[10] P. Hoschka. An introduction to the synchronized multimedia integration language. *Multimedia, IEEE*, 5(4):84 –88, oct-dec 1998.

[11] Denis Baggi and Goffredo Haus. IEEE 1599: Music encoding and interaction. *COMPUTER*, 42(3):84–87, March 2009.

[12] D. Fober, C. Daudin, Y. Orlarey, and S. Letz. Interlude - a framework for augmented music scores. In *Proceedings of the Sound and Music Computing conference - SMC'10*, pages 233–140, 2010.

[13] P.L. Quist. *Indeterminate form in the work of Earle Brown.* Johns Hopkins University, 1984.

[14] D. Schnebel. *Mo-no: music to read.* DuMont Schauberg, 1969.

[15] Frederic Bevilacqua, Fabrice Guédy, Norbert Schnell, Emmanuel Fléty, and Nicolas Leroy. Wireless sensor interface and gesture-follower for music pedagogy. In *Proceedings of the International Conference on New interfaces for Musical Expression*, NIME '07, pages 124–129, New York, NY, USA, 2007.

[16] Frédéric Bevilacqua, Florence Baschet, and Serge Lemouton. The augmented string quartet: Experiments and gesture following. *Journal of New Music Research*, 41(1):103–119, 2012.

[17] Dominique Fober, Yann Orlarey, and Stephane Letz. Inscore – an environment for the design of live music scores. In *Proceedings of the Linux Audio Conference – LAC 2012*, pages 47–54, 2012.