Automatic Justification and Line Breaking of Music Sheets

P. Bellini, P. Nesi
Dept. of Systems and Informatics, University of Florence
V. S. Marta, 3 50139 Florence, Italy
Tel: +39-055-4796523, Fax: +39-055-4796523
Reference contact person: nesi@dsi.unifi.it

ABSTRACT

Automated music formatting helps composers and copyists to speed up the process of music score editing by facilitating complex evaluations needed to produce music sheets in terms of symbol positioning, justification, etc. Music justification is a complex task to be automatically performed. It involves the evaluation of a large number of parameters and requires context evaluation. In this paper, the approach adopted in a justification engine of a European Research project is presented. The approach solves many of the problems of music justification: alignment of simultaneous symbols in polyphonic music, spacing dependent from the duration of the figures, compactness and readability of the resulting measure, justification of both main scores and parts. In the paper, several justification algorithms are described and compared. Stretching and shrinking of measures is also possible, while keeping the justification through a tuning parameter. The resulting algorithm can also handle automatically many music notation exceptions: for example time inconsistency of the justified measure and presence of non-durational figures, grace notes, change of clef/key signature, etc. The solution proposed presents a module for music line-breaking. This is included in the justification engine as an option for visualizing and printing right margined music sheets. Several examples are reported to highlight both the problems and the solutions adopted.

Index Terms: user interface, automatic formatting, music sheet, music formatting, space management, justification, beat-line, simultaneities, line breaking.

1 INTRODUCTION

Music notation programs are software tools used by different kind of users. The aim of these applications is producing music scores for computer monitor fruition and at the same time obtaining the professional printouts of the score. Music notation programs are the data entry to produce such music scores in the form of digital symbolic files. They permit interaction with the score, like for instance adding or deleting music notation symbols with very low effort and they allow to print high quality score sheets.

Regardless of the user identity, either composers or music publisher's copyists, the passage from the manual composition to the computer assisted editing brings about specific requirements to accelerate the music sheet production. One of the major expectations of the user concerns the assistance provided by the program to the formatting process of the music score symbols in an automatic manner. A music notation program should help the user in producing the symbolic score sheet, through a backend processing that optimizes and arranges the formatting and the visualization of every music symbol on the page. Music formatting is an art refined during centuries by engravers and music copyists, who have established several non-written rules in the field of music sheet layout. Such rules arose from the progressive refining of a set of guidelines for the correct music notation writing. These rules are not reported in literature in a prescriptive form, and yet they are inherited from an expert to the next. Copyists themselves take years to learn correct music notation and do not formalize the rules [15], [12], [16].

Western music is typically divided in measures, each of which has a formal duration specified by its time signature, real execution time is indicated with the metronomic indication. Each measure can be composed of one or more voices. A voice is composed of a sequence of active figures having an inherent duration, like notes and rests, and symbols that do not have an inherent duration and yet they are essential for the right interpretation of the active figures, such as clef changes, key signature changes, grace notes etc. Moreover, the sum of figures duration normally is equal to the time signature of the measure. An active figure starts in a specific moment of the measure, no sooner than all the preceding active figures belonging to that voice have elapsed: this moment is called *beat*. A beat in the measure is univocally identified by summing the duration of the figures preceding the figure falling on that beat. The current beat and the duration of the active figure falling on that beat identify the beat where the next active figure will fall. In the visual representation of music, the music sheet, the beats are identified by a vertical slice. When a single voice is present in the score, the music is called monophonic, whereas when more voices are present the music is called polyphony or polyphonic. In effect, the above described is a too simplified model since several exceptions have to be managed as depicted hereafter.

Some manuals do deal with the subject of writing correct music notation (see for example [12], [15], [8], [18]) and present formatting guidelines for the western music notation quite diffusely accepted in the musician community and consolidated by centuries of use. These guidelines are simply presented through a set of examples. Automation in music formatting helps composers and copyists to speed up the process of music score editing by facilitating complex evaluations needed to write correct music sheets in terms of symbol positioning, justification, etc. Empowering the computer to perform such task allows the composer to increase the efficiency and to concentrate on more creative aspects rather than to arrange visualization aspects. The problems that automatic systems for music formatting have to cope with during the insertion and positioning of symbols in a music score have been outlined in [1], [4], [5], [10], [18]. These problems are much more relevant in Internet on-line applications when the music score received by the end-user has to be reformatted according to the user needs and screen and manipulated. In such instances, the automatic formatting is an important and appreciated feature. In [16], context evaluation conditions and rules for symbols formatting have been formalized as MILLA (Music Intelligence Formatting Language). MILLA is used to describe conditions that are interpreted in real time by an inferential engine so as to apply suitable rules to format the music score in WEDEL Music Editor (see [14], [21]).

Apart from the needs of setting parameters for the position and the insertion of the music notation symbols, space management among the music notation symbols has to be considered as well. The space management process can be divided into two sub-processes:

- music justification, automating the calculation and assignment of the space between symbols in the measure (in input some parameters, like the scale rate of stretching or shrinking, can be taken),
- line-breaking, automating the calculation and assignment of the space between symbols in a set of measures to fit exactly the width of the client area in the monitor or the width of the printed page for prints (in this case, no parameters are needed).

Due to a lack of standardization there are not commonly accepted rules for music spacing in music justification process. In [8], examples of music spacing have been reported to introduce some guidelines in music spacing; they are more than concrete context application cases, for example : "*Good spacing of notes within a music line enables the performer to perceive at a glance the different duration of those notes*" (so-called durational spacing guideline), and "*... notation giving the impression of proportion is all which is required... No slavish adherence to exact* *mathematical proportions is required, whether each measure has the same physical length*". The priority of compactness over direct proportionality between duration and space is then sealed with: "*...effect of relative duration is achieved... values of longer notes are given sufficient space to grant the impression of their duration without wasting manuscript paper*". The objective of no space waste is strictly related to the basic requirement of readability of the music score.

To associate each symbol with a space proportional to its duration does not solve the justification problem. This trivial solution does not take into due account the presence of symbols not having an explicit duration, such as grace notes, clefs and key signature-changes, when they are in the middle of the measure, etc. Besides, it does not consider any compactness of measures, therefore it would not produce readable music scores. A part from *durational* spacing (space estimated on the basis of the temporal duration of the figures), another basic guideline for music spacing deals with polyphony, meaning the event when several voices are coded to play in parallel. The guideline for writing correct music notation states the following "*symbols which are simultaneous in time must be vertically aligned in the score*". We will refer to a set of symbols that have to be played simultaneously as a *simultaneity*, like in [9]. During the execution of several parallel voices, a musician can recognize the time syncopation of the piece from the types of figure in the score (fourths, eights, etc.). In single musician's parts, that event occurs only for polyphonic instruments, such as piano, organ, harp, etc. For the main score of the orchestra conductor the rule which usually applies is as follows. In the main score, the parts of different instruments are typically organized as different voices on several staves. In any case, the vertical alignment of simultaneous symbols, or *simultaneities* as in the following, is a basic requirement for the score readability. For the above reasons, music justification is more difficult than text justification. Note placement must emphasize temporal relationships within the music piece and must vertically align figures falling on the same beat. The justification has to give evidence of a temporal evolution through the vertical alignment of symbols. Factors to be considered are: (i) the complexity of music spacing, (ii) the lack of rules in literature for space computation, (iii) the high number of variables involved and (iv) the wide spectrum of possible exceptions. When considering such previous points, a rule-based approach with context evaluation conditions (as MILLA, see [16]) is not the best architecture to deal with space computation for the justification of a music score. Probably, a procedural approach is better, as outlined also in [4], [5], [6], [17], [19].

In [17], an analysis on both the functionalities required and the music justification problems has been presented. The main problems of justification algorithms are: (i) the vertical alignment of simultaneous symbols, in polyphony and among the different parts of the main score; (ii) the spacing of symbols as a function of their time duration; (iii) the management of collisions to avoid any overlapping of music notation symbols; (iv) the management of different spaces for different views of the same score (main score, parts); (v) the space management of symbols without time duration.

The topic of this paper is the automation of the formatting process of computer represented musical notation. In details it covers two sub-processes in the wide spectrum of the problems arisen with music score formatting: automated justification of music scores and the margin alignment of the music sheet, known as line-breaking. Details related to the MILLA formatting engine developed for the same project can be recovered in [16]. This article presents the solutions adopted for the justification engine of WEDEL Music editor [14]. The results presented in this paper mainly consist in the formalization of a unique algorithm for solving a set of problems of music justification.

WEDEL Music editor is a music notation program developed in a research project partially founded by the European Commission IST work program, WEDELMUSIC, partners of the project: University of Florence (location of the authors), IRCAM, CASA Ricordi (BMG), Suvini Zerboni (Sugar), FHGIGD, SMF, CESVIT, SVB/FNB, ILSP. The WEDEL Music Editor is mainly oriented in producing and using the music scores which are distributed on Internet and thus their automatic formatting on the computer windows and paper according to the end-user's needs – a possible application can be the automatic formatting of music in cooperative music editors such as MOODS [20], [22]. Presently the WEDEMUSIC editor is distributed in Open Source.

The paper is organized as follows. In Section 2, there is an overview of problems that a justification engine must be able to cope with. Next to problems, the solutions obtained are presented. In Section 3, the architecture of the WEDEL Music Editor justification engine is presented with some implementation details. In Sections 4 and 5, the pre-processing and the justification phases of the algorithm are described by using a mathematical formalization. In Section 6, the line-breaking algorithm for line arrangement of music measures is presented. In Section 7, conclusions are drawn. Please notice that all musical examples presented in this paper have been produced with the WEDEL

Music editor and come out of the computation of the justification engine.

2 PROBLEM OVERVIEW

The main reason of music symbols justification in the sheet is to increase music readability. In a correctly justified music score musicians can recover at the first glance the rhythm of the music piece. Spacing dramatically impacts on the readability of a music piece. Music justification deals with the task of spacing a music score's symbols abiding by several requirements determined by common use during the last centuries Such requirements were analyzed in order to enable a correct automation; then, these requirements were divided and transformed into simple steps so as to permit the engine to handle all possible cases and exceptions. The general guidelines of a justification engine are quite intuitive.

1) *Durational spacing*. The spacing algorithm must set the space between the figures. In this way, it becomes intuitive for the reader to recognize duration of notes and rests, and grasp the sense of rhythm of that piece at the very first glance. The space after a figure must be a function of its duration. This function may be complex. The resulting music score is assessed by musicians in term of appearance, nice layout, readability and compactness.

2) *Vertical alignment*. Second requirement for justification engine involves the simultaneities among figures. Figures belonging to different voices/parts and falling on the same beat, must be *vertically aligned* because of visual intuitive recognition of temporal coincidences. For the main score, where all the instruments are present, this means aligning the simultaneous symbols of all the single parts. There may be some exceptions to avoid collisions.

3) *Line-breaking*. The line-breaking consists in stretching/compressing the justification of a music line to obtain the alignment with page left-right margins. For such aspects, in [7] a work focussing on the single part line-breaking has been proposed. The problem can be considered in terms of process controllability to justify a measure so as to reach a certain *targeted width*. This is one of the most required features by publishers and musicians, which also means to avoid producing score lines with too many blank spaces at the end of each line, thus determining lower readability.

In Figure 1, a main score with two parts, each one made of two measures is reported. The two measures contain the same figures, their difference being only the spacing. The second measure is correctly justified taking into account vertical alignments of simultaneous symbols and *durational* spacing; therefore it is much more readable than the first one.

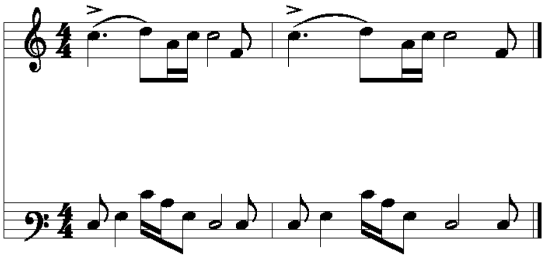


Figure 1 – Justification results in more readability

Several justification problems have been discussed in [17], [12], [9], [6], [1]. In the following, the list of the most complex aspects and guidelines is reported just to give an overview of the problem's complexity. The reported list is not exhaustive, it is a simple collection of the most relevant and common problems. Since they are strongly related one another, in some instances it is almost impossible to provide a specific independent example. On such grounds, after this part, a set of explicative examples is reported, making specific reference through the referring letter to the aspects following below. In order to understand better the impact of both constraints and problems to be faced when justifying music scores, they have been classified in three areas: layout, compactness and flexibility.

Layout

(a) Spacing of music notation symbols must be based on their time duration: a note length affects spacing. More space has to be left after a long note than after a shorter one [17].

(b) Vertical alignment of simultaneities. The simultaneities are univocally identified by the beat they fall on and they correspond in a justified score to a vertical slice. This means that note spacing on one staff depends on the position of the notes in the other staves. The justification engine must, not only associate a space with a figure according to its duration, which would be a trivial process; it must also arrange visualization so as to both align vertical simultaneities and obtain a suitable visual representation. Vertical alignment must be kept among simultaneous figures, and it may occur that some extra space should be left between notes on one staff, because of the intervening notes on another staff [17]. Sometimes, some figures of a voice can be missing, for example when a voice begins in the middle of a measure.

(c) Managing symbols without duration (e.g., grace notes, changes of key signatures or clef) which can occur wherever in the measure. Vertical alignment of simultaneities and spacing of figures must consider also the presence of figures without duration. Furthermore, grace notes must be placed just before the note they are referred to. If more non-*durational* figures are present in parallel voices before the same beatline, the space must be optimized.

(d) Managing non time-consistent measures (with respect to their time signature). A previous check on the time consistency of the measures can help in preventing malfunctions, but an algorithm working only on time consistent measures can offer only very limited performance. Some music editors force the measure into consistency by automatically adding rests. This is a strong restraint to avoid.

(e) Symbols must not overlap one another. Collisions typically occur among *local symbols* when musical markings, accidentals, and duration dots are present in short duration notes or chords or whenever there are horizontal symbols such as slurs, crescendo, decrescendo, etc. Collisions among local symbols have to be managed with smart positioning mechanisms to place symbols around figures and using the justification and line-breaking engines.

Compactness

(f) The user should be granted with some control over the stretching or shrinking of measure justification, as well as over the duration function used for spacing. Even if the user wants a very tight measure, the algorithm must avoid collisions between symbols. Collisions must be automatically managed and solved.

Flexibility

(g) To justify consecutive measures with different justification parameters and modalities has to be possible. This feature is typically required to cope with strong changes of figure density within consecutive measures.

(h) The justification engine must be flexible enough to be able to arrange spacing in different manners. This goal can be reached by providing:

- different types of justification functions (log, linear, difference, etc.) for the estimation of spaces associated with figures (note, rests, chords, etc.).
- tuning parameters to change the ratio of measure stretching/shrinking the spacing among figures.

(i) Different space arrangement for different views of the score (main score, parts) should be provided. Same with different parameters and functions, when justifying the main score and the same measures in each single part.

(j) The algorithm and model for music justification should be able of justifying polyphonic/multi-voice instruments (piano, organ, harp, etc.). These instruments often have more voices starting at different instants along the measure, which means: if only notes are considered, some voices/layers may be time inconsistent. This event occurs quite often, since inserting rests may cause confusion to the player.

In the following sub-sections, there are some examples dealing with the above mentioned problems. Explicit reference is made by using letters -- e.g., (a). A detailed description of the algorithm used for music justification is reported in Section 3.

2.1 Some simple examples of justification

In the justification process, more space has to be left after a long note than after a shorter one: this is called *durational* spacing (a). Two different functions for the estimation of space according to note duration have been implemented in our algorithm: linear and logarithmic (i). In Figure 2, the effect of *durational* spacing is shown: spaces are computed among figures with the linear function in the first measure and with the logarithmic function in the second measure.

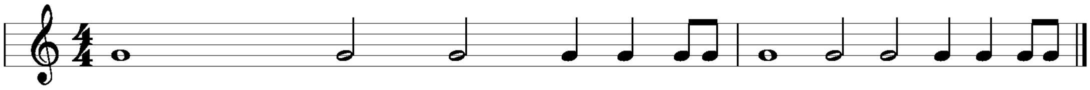


Figure 2 – Durational spacing: linear (first measure) and logarithmic (second).

The linear duration function leads up to an absolute spacing: the simple additive property can be applied to find the absolute position of any symbol along the measure according to the duration of the notes involved -- i.e., a note with double duration gets double space. If shorter and longer notes alternate in the same measure, the linear proportion between space and duration leads to very large spaces among notes. This large spacing may confuse the reader, thus decreasing music readability. For readability and economy of space, the space left after a note has to be a function of the score's music density. The logarithmic spacing function conforms to a logarithmic encoding of duration, and is based on the observation that musicians respond better to differences in spacing of successive notes than to absolute note spacing. For example, it could be possible to encode only variations in figure duration. Logarithmic justification is in any case better since it results in a more compact arrangement of the measure. Music readability is quite subjective, the choice of the justification type (linear or logarithmic) belongs typically to both the copyeditor and publisher's style. It is very important for the copyeditor to control the measure stretching or shrinking through a parameter tuning -- e.g., the bigger the parameter is, the wider the justified measure turns out to be (see Figure 4), see above point (h). The role of the tuning parameter in the justification function will be discussed in Section 5.

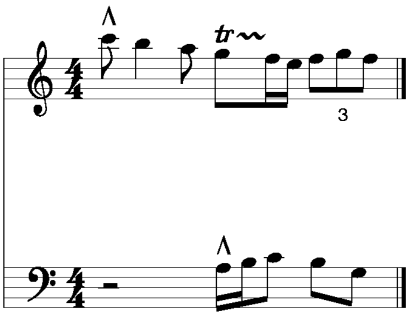


Figure 3 – Logarithmic justification

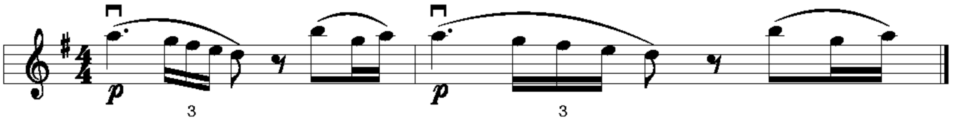


Figure 4 – Justification of the same measure with different values of the tuning parameter

When the tuning parameter has to be small, collisions must be automatically detected and any symbol overlapping strictly avoided, see (e) and (f). Figure 5 shows the "same" measure justified by using different values of the stretching parameters (g): in the first measure the effect of justification with a very small tuning parameter is shown; the result is a very tight measure. All spaces are kept at the minimum with respect to the physical width of the symbols and in many cases the main figures appear separate for the presence of sharp and natural symbols to avoid collisions. The second measure is justified with a greater tuning parameter, additional spaces were needed as well: between the first note of the beam and the following chord to arrange the naturals and flat, between the first note of the measure and the following chord due to the presence of sharps. The third measure is justified with the same tuning parameter of the second measure. In this case, no additional space was required, for there are no alterations.

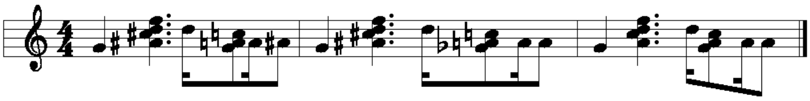


Figure 5 – Stretch and shrink through the tuning parameter and collision handling

Spacing information and arrangement could have to be different in different views of the same music. Different views could mean those associated with main score and parts. In Figures 6 and 7, same measures are shown as visualized in the main score and in the single part (please refer to the third part of Violin). The same measures were justified in different manners: passing from one view to the other the appropriate space arrangement has to be regenerated. In addition, it is very important to store for each measure and for each view a separate set of justification parameters, as it will be discussed in Section 5. Figures 6 and 7 depict the algorithm behavior in presence of symbols without duration (a grace note in the second measure of the third part) (c).

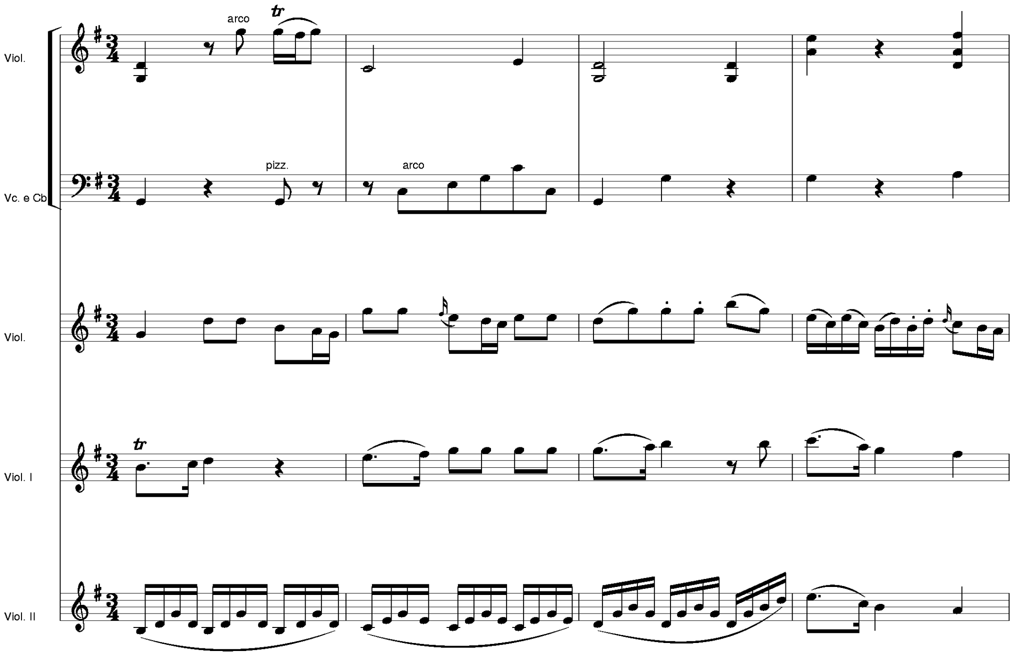


Figure 6 – The main score

Vertical alignment of simultaneities and space arrangement must consider the presence of figures without duration (for example: grace notes, clefs in the middle of the measure, etc.) (b). As shown in Figure 8, the presence of non-durational symbols impacts on the space arrangement of other parallel voices (see beam of grace notes in second measure, or the first grace note in the first measure of clarinet part). The correct music notation positions grace notes just before the note they are referred to (see the position given to the beam of grace notes in first part). If there are more non-durational figures in parallel voices close to the same beat-line, the space must be optimized/reduced. For example, in Figure 8, the alignment between the clef change in the 2nd part and the grace note in the 1st part.

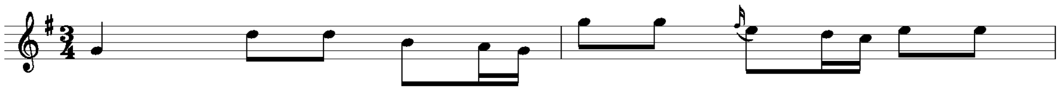


Figure 7 – The third part (violin) with a different justification

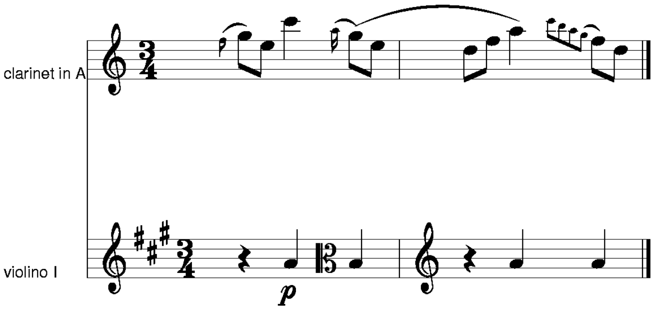


Figure 8 – Justification of non-durational symbols

To check the measure time consistency with the time signature can help to prevent from delivering any wrong measures (d). In addition, to justify time consistent measures is easier than to justify non consistent measures. Unfortunately, this hinders the implementation of a flexible justification module. The algorithm proposed can produce correctly justified measures whenever the latter are either time consistent or inconsistent (see Figure 9). This is very important since during the music notation editing the measure under construction passes through several status where it is not consistent. For example, measures containing only non-durational figures and parts non-time-consistent.

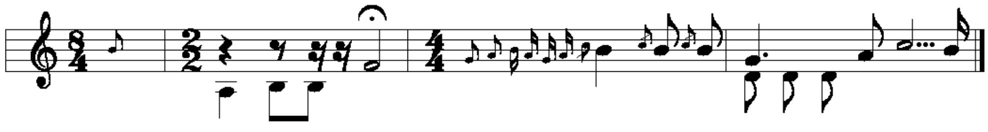


Figure 9 - Justification engine at work on time inconsistent measures: time faults, time exceeding

Small notes are justified as normal size notes in the score: for example when notes of another instrument are transcribed in a musician's part to help him/her in recognizing the execution time (cue notes). At times they are only a means of expression. In Figure 10, the second measure contains many small notes: they are justified as normal sized notes. This has been possible simply by associating a status of durational figures to the notes. The justification algorithm has to cope with graced and cue note in a different manner (non durational or durational), thus producing a correct spacing.

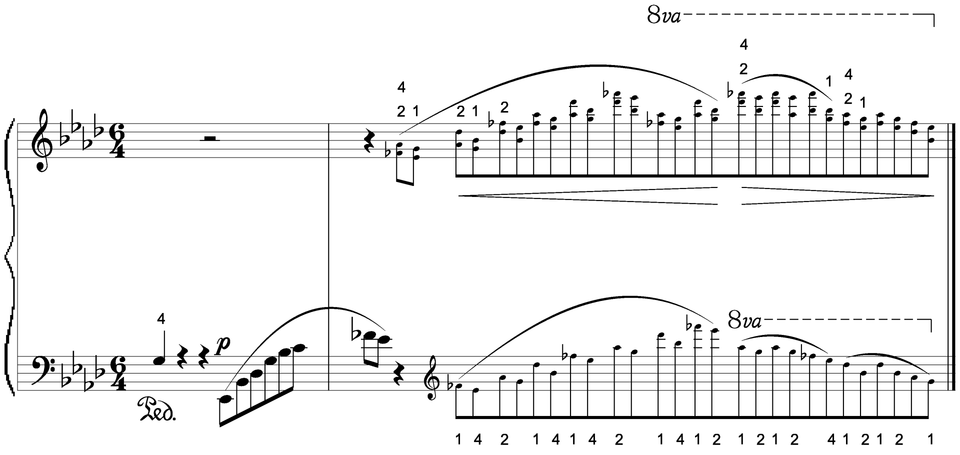


Figure 10 – Small notes, too, have to be justified as *durational* figures if needed.

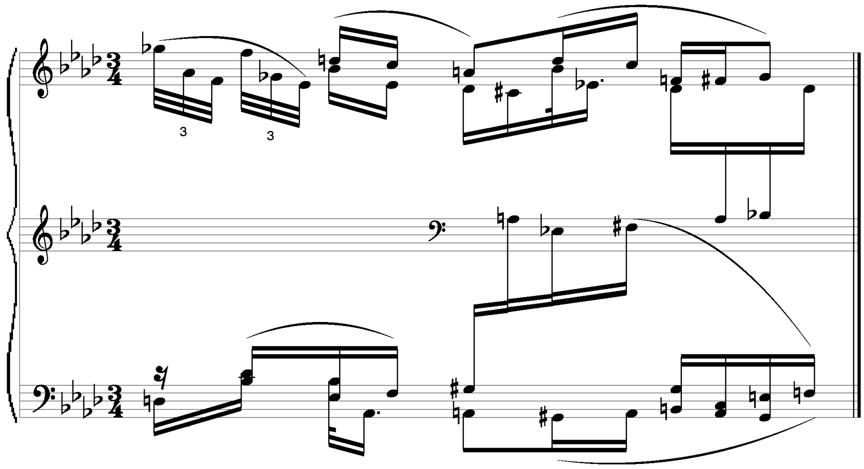


Figure 11 – Whether needed, small notes can be justified, too. See the integration with non time consistent voices.

Some instruments like piano, organ, harp, have single parts on several staves: the justification engine should justify such parts considering all the different voices (j). These voices can sometimes be time inconsistent, like the upward-stem voice in the first staff of the Figure 11. The intention of the composer was to assume the tuplets to be time-consuming for both voices of the first staff. In the model all voices are processed as separated lists; therefore to align correctly the upward-stem voice, some time-consuming rest, though invisible, have to be introduced in this voice before the first note (this is the solution adopted in our model). A different solution may consist in forcing notes to have a precise graphic location, notwithstanding the work of the justification algorithm.

3 GENERAL STRUCTURE OF THE ALGORITHM

In this section, the general structure of the justification algorithm is presented. The objective of the justification is to define the correct spacing between music notation symbols. The inputs of the justification algorithm are the sequences of music notation symbols called *voices*: lists of music notation symbols forming a melodic line in the measure.

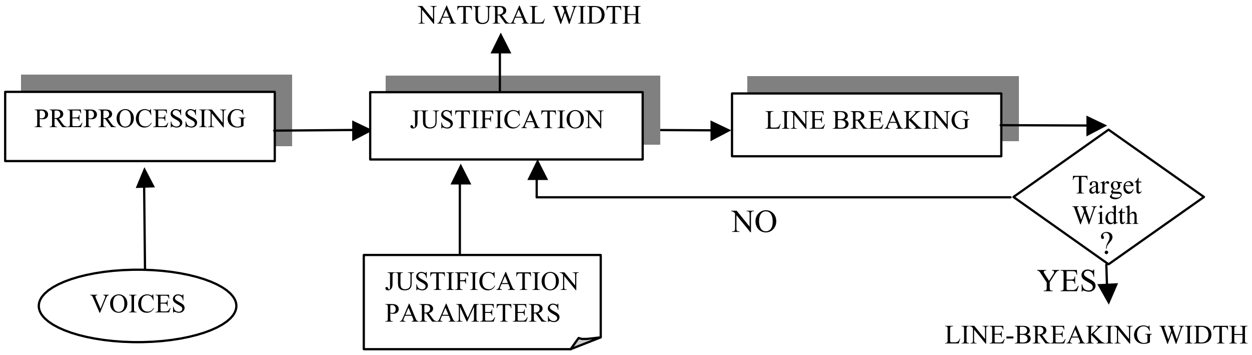


Figure 12 - General flow of justification and line-breaking engines

The voices are preprocessed to collate them in a unique data structure for further elaboration. The aim of this phase is essentially to search time coincidences for figures of different voices inside the measure. The justification task calculates the spaces among symbols. The music notation spacing obtained by direct application of the justification parameters is called the measure's *natural width*. When a set of measures are arranged in a page according to their natural width, it is very likely for the right margin of the page not to be perfectly reached by the sum of the measure natural widths of each music line. To solve this problem, a technique of line-breaking similar to the one used for word spacing is needed. On the other hand, unlike line-breaking for text, the goal of a line-breaking module for music is to produce a set of justified measures in order to fit perfectly the page wideness, instead of inserting space among words or letters. This difference is due to the fact that single measures may have different parameters and justification function.

For these reasons, music line-breaking is obtained by stretching the natural width of the measures involved in the page line. The increment in size each measure has to reach is obtained proportionally to its natural width in order to reach the *Target Width*. The measure stretching for the line-breaking is based on the justification algorithm. This allows to keep the natural spacing of the measure: logarithmic or linear, while maintaining the vertical alignment of simultaneities. In addition, the line-breaking algorithm has to work with time consistent and inconsistent measures. Please note that line-breaking is a dynamic process of space rearrangement. The spacing has to be recomputed every time that: the page (computer window) is resized, the music is changed (adding/deleting symbols), the music score is justified with a different justification function or by using different parameters affecting its initial natural width. Besides, as shown in Section 6 the line-breaking algorithm is iterative, see Figure 12.

The sum of the duration of the figures of a voice should be consistent with the time signature of the measure. The note *duration* is computed in musical fractions of the whole note: for example 1/8, 1/4, 1/16. The same unit of measure is applied to compute the time *interval* among the figures. The *beats* are also computed in terms of fractions and they result from summing figures duration which are present in the preceding beats. Please note that a voice is only a list of musical notation symbols. Each figure knows only its duration, and does not know if there are simultaneous figures in different voices. Information about time progression must be derived from the processing of voices. What kind of information is needed and how to compute it are aspects which be outlined in the following section, for example the beat where each figure falls is easy to compute, simply by summing the duration of the preceding figures in the same voice.

The following sections are focussed on a detailed description of the different steps for the justification algorithm, which concerns a single measure of the score (with one or more voices/layers/parts). There is no loss of generality in describing the justification for one measure of the score. In fact, the presented algorithm performs justification on each measure separately, and the natural width of each measure is not influenced by the presence of other measures and by their contents. Also the line-breaking algorithm, as outlined in Section 6, has its base unit in the single measure.

4 PREPROCESSING PHASE

This Section describes the preprocessing phase of the algorithm, while Sections 5 and 6 are devoted to the justification and line-breaking phases. In single parts of monophonic music, only one voice has to be justified, whereas in single parts of polyphonic music more voices have to be justified. For the main score all voices belonging to the single parts have to be justified. The voices that have to be justified must be preprocessed. The processing phase has to be always performed regardless of voices being referred to a single polyphonic instrument for a single part or to different monophonic and polyphonic instruments for the main score. Therefore, without lack of generality, we can consider the case of more voices (see Figure 13) and they can be modeled as ordered sets:

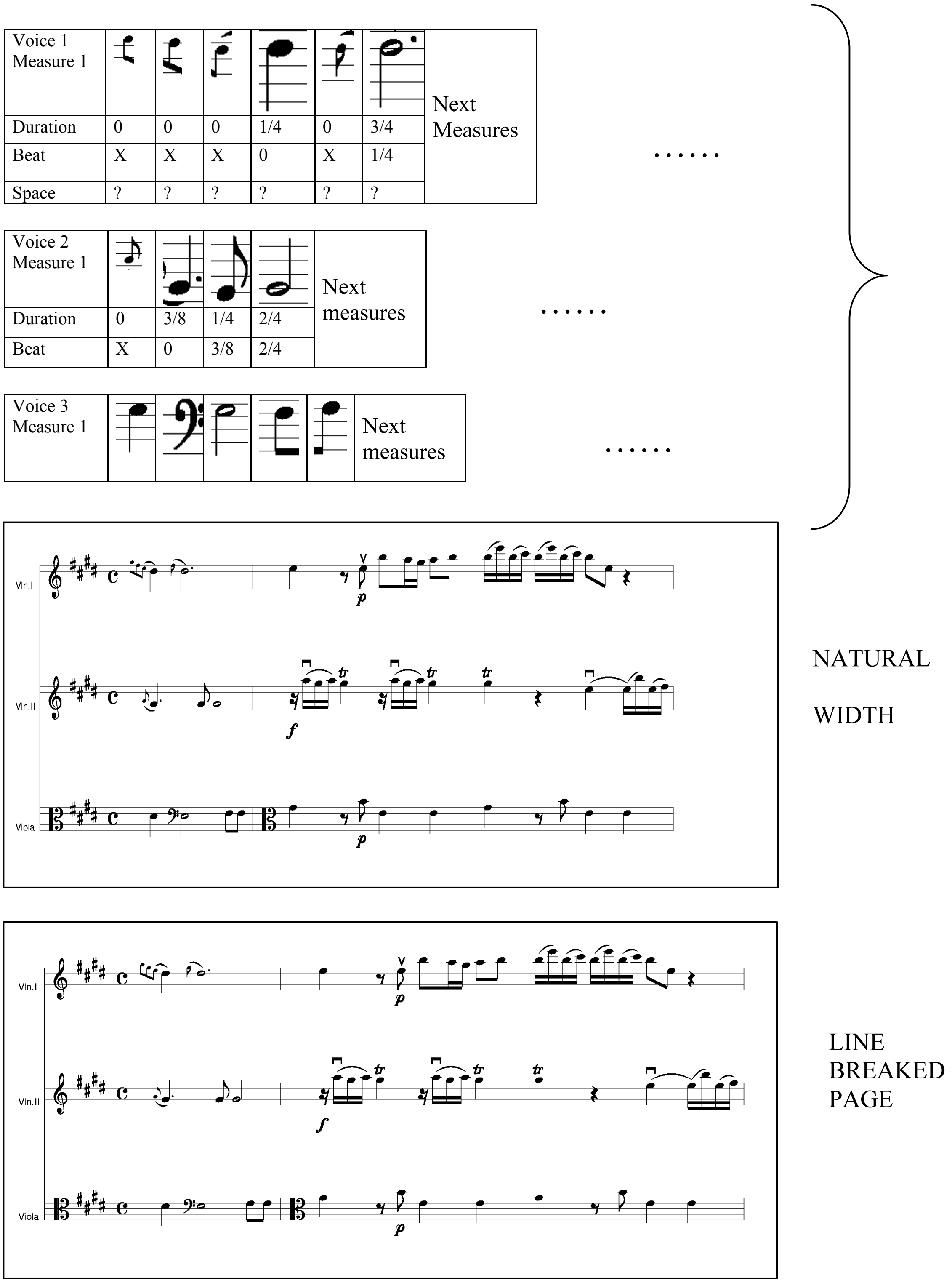
$$\begin{cases} \boldsymbol{v}_1 = \{\boldsymbol{f}_{11}, \boldsymbol{f}_{12}, \dots, \boldsymbol{f}_{1L_1}\} \\ \vdots \\ \boldsymbol{v}_i = \{\boldsymbol{f}_{i1}, \boldsymbol{f}_{i2}, \dots, \boldsymbol{f}_{iL_i}\} \\ \vdots \\ \boldsymbol{v}_{\mathbf{max}} = \{\boldsymbol{f}_{\mathbf{max}1}, \boldsymbol{f}_{\mathbf{max}2}, \dots, \boldsymbol{f}_{\mathbf{max}L_{\mathbf{max}}}\} \end{cases},$$

where: f_{ij} is the j-th figure of i-th voice, $v_1 \dots v_{\max}$ are the voices of the score, L_i is the number of symbols contained in the i-th voice. In Figure 13, voices $v_1 \dots v_3$ are listed in the upper part of the figure with the visually indicated figures.

The following values/attributes are present for each symbol f_{ij} (for example a rest or a note) of the voice:

- $\boldsymbol{f_{ij}.d}$ is the duration of figure f_{ij}, which is its temporal duration: 1/4, 1/8, etc.; some figures may have no effective duration, for example the symbol of change of clef and the grace notes;
- $\boldsymbol{f_{ij}.b}$ is the *beat* of the measure where the figure falls along the time axis. It has to be estimated by the algorithm;
- $\boldsymbol{f_{ij}.w}$ is the space between figure f_{ij} and the next figure of the same voice. It has to be estimated by the algorithm.

The justification algorithm's goal is provide an estimation on the correct space needed among music notation symbols. The space is estimated in terms of a unit of measure regardless of the scale factor used for producing/drawing the music staff. This unit of measure is named the "*space*" and is defined as "the width of the *notehead*". In the algorithm presented hereafter, the estimation of the spaces follows a first preprocessing phase where simultaneous symbols of different voices are grouped. To this end, voices are processed in order to find the beats where each figure falls, then voices are re-elaborated to find simultaneous symbols in each measure and finally they are collated together in *simultaneities*, vectors of simultaneous symbols.



Voice 1 Measure 1							Next Measures
Duration	0	0	0	1/4	0	3/4	
Beat	X	X	X	0	X	1/4	
Space	?	?	?	?	?	?	

Voice 2 Measure 1					Next measures
Duration	0	3/8	1/4	2/4	
Beat	X	0	3/8	2/4	

Voice 3 Measure 1						Next measures

Figure 13 - Justification phases to estimate correct spaces among symbols

A simultaneity is univocally identified by its *beat*. In time-consistent measures and for every voice, the value of beats for the figures ranges from 0 to a value obtained subtracting the duration of the last symbol of the voice from the time signature of the measure.

A *beatline* is the horizontal coordinate where falls a beat in the measure, and where should fall the symbol if it has no physical width. The estimated space from a symbol to the next is applied from its *beatline* to the *beatline* of the next. Since the symbols of music notation have a physical width, only a specific point of the symbol falls on the beatline, usually the left edge of the symbol. For example, the notes have to be placed on the staff to have left edges of the noteheads on the beatline regardless of accidentals or other symbols being associated with the figure. The beatlines of simultaneous symbols must be exactly aligned on a vertical column.

The sequence of the *beats* where figures fall stands for the temporal evolution of the events. The beat where each figure falls can be estimated from the duration of the preceding figures in the same voice, measured in fractions (1/4, 2/4, 3/4, 7/8 etc.). For instance, by considering voice *i* we have:

$$\boldsymbol{f_{ij}.b} = \sum_{r=1}^{j-1} \boldsymbol{f_{ir}.d}$$

The above estimation is preparatory to the identification of simultaneities as shown in Figure 14.

The preprocessing *P()* elaborates the voices $\vec{v}$ in order to estimate the simultaneities:

$$\vec{\boldsymbol{v}}' = \boldsymbol{P}(\vec{\boldsymbol{v}}) = \{\vec{\boldsymbol{v}}_1', \ldots, \vec{\boldsymbol{v}}_i', \ldots, \vec{\boldsymbol{v}}_{\max}'\}.$$

A preprocessed voice v'_i has the same order of elements of v_i , while in some cases new NULL elements are added to mark the absence of a symbol starting at that simultaneity in the voice. The NULL element is found in the structure of Figure 14 when a voice (horizontal line of structure) has no new figure starting by a specific beat, which is univocally identified by the simultaneity (vertical column of structure); at least one figure located in another voice starts at that beat on the basis of data structure construction. All the preprocessed voices have the same number of elements, corresponding to the number of simultaneities of the measure. Figure 14 depicts the processed voices, v'_i, corresponding to the voices of Figure 13:

$$\boldsymbol{v}_i' = \{\boldsymbol{f}_{i1}', \boldsymbol{f}_{i2}', \ldots, \boldsymbol{f}_{iR}'\},$$

where: *R* is the maximum number of simultaneities (or beats) which can be possible in the measure, $f'_{ik} = f_{ij}, \quad (j \le k)$ or NULL. The estimation of f'_{ik} is reported in the following.

The set $\vec{\boldsymbol{v}}'$ is constructed in order to align the simultaneous figures of the different voices. In $\boldsymbol{f}'_{ik}$, ***k*** identifies the figure position in the voice. The beat of the figures vertically aligned (the same column, the same ***k***) is identical: $\boldsymbol{f}'_{1j}.\boldsymbol{b} = \ldots = \boldsymbol{f}'_{ij}.\boldsymbol{b} = \ldots = \boldsymbol{f}'_{\max j}.\boldsymbol{b}$ for ***j* = *1…R***, where $1..max$ is the number of voices and j identifies the position of the figures in the voice.

The preprocessing phase leads to the construction of a structure like the one in Figure 14, where the preprocessed voices *v'* coincide with the lines containing musical symbols. Figures falling on the same beat are simultaneous in time, it is an issue of fact. This relationship is represented in the data structure of Figure 14, aligning vertically the simultaneous figures (see simultaneities 4 and 6 in Figure 14). In such structure, every column corresponds to a beat of the measure where at least a figure falls. A NULL element corresponds to a voice that does not have a figure falling on a beat. In order to find the simultaneities, it is necessary to scan several voices in parallel. The structure can be regarded as the tracking record, on a time base scale, of the events succeeding and of the duration of the symbols encountered. This leads to the construction of the structure having the properties listed above, starting from the voices written as in the upper part of Figure 13. It is based on the insertion of NULL elements and right shifting of the figures.

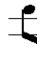
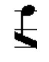
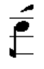
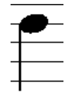
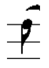
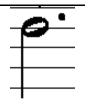
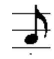
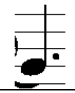
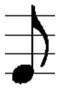
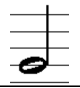
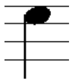
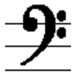
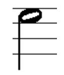
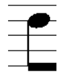
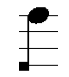
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Vln I							NULL	NULL	NULL	NULL
Vln II	NULL	NULL			NULL	NULL			NULL	NULL
Viola	NULL	NULL	NULL				NULL	NULL		
Beat, $s_l.b$	X	X	X	0	X	1/4	3/8	2/4	3/4	7/8

Figure 14 - Collating simultaneities (columns) from the voices of Figure 13

After building the structure as in Figure 14, which contains the processed voices $\vec{v}'$, to group simultaneous figures together is now possible. This is obtained by considering the columns of the structure (the simultaneities) where the voices preprocessed $\vec{v}'$ are reported, as in Figure 14, and by defining a new vector, called vector of the simultaneities.

$$\vec{\boldsymbol{s}} = \{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_l, \ldots, \boldsymbol{s}_R\},$$

where: $\boldsymbol{s}_l = \{\boldsymbol{f}'_{1l}, \ldots, \boldsymbol{f}'_{il}, \ldots, \boldsymbol{f}'_{\mathbf{max}\,l}\}$ is characterized by having one element (a musical symbol or a NULL element) from each voice and that single element falling on the same beat

$$\boldsymbol{f}'_{1l}.\boldsymbol{b} = \ldots = \boldsymbol{f}'_{il}.\boldsymbol{b} = \ldots = \boldsymbol{f}'_{\mathbf{max}\,l}.\boldsymbol{b}$$

Each simultaneity of vector $\vec{\boldsymbol{s}}$ has the following attributes:

- $\boldsymbol{s}_l.\boldsymbol{b}$ is the *beat* of the simultaneity, which univocally identifies the simultaneity,
- $\boldsymbol{s}_l.\boldsymbol{d}$ is the *duration* in time of the simultaneity, defined as $\boldsymbol{s}_l.\boldsymbol{d} = \boldsymbol{s}_{l+1}.\boldsymbol{b} - \boldsymbol{s}_l.\boldsymbol{b}$
- $\boldsymbol{s}_l.\boldsymbol{w}$ is the space (*width*) between the successive simultaneities s_l and s_{l+1}

The duration of a simultaneity is defined as the time, measured in terms of fractions, between the present and the successive simultaneity. For example, the duration of the sixth simultaneity (column without encountering the next column of symbols) in Figure 14 is 3/8-1/4 = 1/8. In a simultaneity, all the voices present a figure or a NULL element: if the voice had a figure, the next simultaneity, *considering only that voice*, would be located after a time equal to the duration of the figure, which is in that case also the duration of the simultaneity. In Figure 14, the first voice, for example, has a figure of duration 3/4 in the sixth column, which is also the last symbol of the voice and then duration is ideally to the end of the measure: considering only this voice this simultaneity should have a duration of 3/4. In the sixth column of the same figure, the third voice presents a figure with a duration of 2/4: *considering only this voice* this simultaneity should have a duration of 2/4.

On the other hand, if a voice has a NULL element in the current simultaneity, then the next simultaneity, *considering only that voice*, would be located after the remaining elapsing time of the last figure of that voice. The remaining elapsing time of a figure at the current simultaneity is calculated as the difference between the duration of the last figure of that voice before the considered beat and the interval in fractions between the beat where that figure starts and the beat of the current simultaneity. In this case, the duration of the current simultaneity would be equal to that remaining elapsing time. For example, in the sixth column, the second voice presents a NULL element: the figure that continues at the sixth simultaneity is the 3/8 note located in the fourth column on beat 0. Since the beat of the sixth column is 1/4, then (1/4-0)=1/4 is the time elapsed of the 3/8 note at the sixth simultaneity, and (3/8-1/4)=1/8 is the remaining elapsing time of the figure at the sixth simultaneity. This would be also the duration of the simultaneity considering only the second voice.

By determining for each voice each and every simultaneity, the real duration of the simultaneity and the beat where falls the next simultaneity are determined as well. For the sixth column of Figure 14, this value is 1/8, due to the second voice (1/8 is the duration of that simultaneity). The beat of the simultaneity of the sixth column (1/4) plus its duration (1/8) gives 3/8, which is the beat where falls the next simultaneity, containing at least one figure in the second voice, the one giving the minimum duration for the current simultaneity. Therefore, in Figure 14, the simultaneous symbols are drawn in the same columns. For each simultaneity, the beat where it falls is indicated, while the duration can be calculated from the difference of successive beats.

The figures without duration (grace notes, clefs for clef change, flats and sharps for key signature change) are treated in a different manner. Whenever there are figures without duration in the measure they occupy a dummy simultaneity. The dummy simultaneity does not have a duration in terms of time and beats of the measure and the dummy simultaneity does not fall on a beat of the measure (see columns 1, 2, 3, 5). Musically speaking a grace note falls on the beat of the note it is referred to, subtracting from that duration the time needed to play the grace note itself. As for the algorithm presented herein, when assigning a simultaneity of grace notes, the beat of the next note is not relevant. For other figures without duration, such as key signature or clef changes, it is a nonsense to talk about the beat these figures fall on. A figure without duration never falls on a simultaneity with figures having duration.

The dummy simultaneity reserved for figures without duration can contain more than one symbol, for example see columns 3 and 5. These symbols are not simultaneous and the adjective itself has no meaning for figures without duration; but whenever there are more successive simultaneities of figures , they are grouped together in a unique simultaneity if the symbols are on different voices. When there are more consecutive figures without duration in the same voice, then different consecutive dummy simultaneities are reserved in the structure. The figures without duration can be considered as an exception for the justification algorithm, since they do not fall on a beat of the measure and do not have a duration. In fact their spacing is not based on their duration, but only on their physical width. The structure is not affected by the presence of these symbols. In the following section, when discussing about spacing functions, only the simultaneities containing figures with duration are considered. The treatment of the simultaneities without duration (dummy simultaneities), the spacing of the symbols contained in dummy simultaneities and their positioning in the measure are dealt with at the end of the next Section.

The maximum number of simultaneities which are likely to be found in a measure defines a situation where each symbol falls on a different simultaneity -- namely, this is what happens when several voices have figures that do not ever collide in the same simultaneity. It is important to consider also that the first figures with duration, of every voice, are aligned together in the first simultaneity.

5 JUSTIFICATION PHASE

The previous preprocessing phase provides an estimation of the vertical alignment of simultaneous symbols. The aim of the justification phase is to calculate and set the width of each simultaneity, $s_I.w$, according to both a specific duration function (linear or non-linear) and a set of parameters the function depends on. Finally the justification algorithm estimates the space between each figure and the next for each voice (setting for each figure attribute $f_i.w$).

In the following, monophonic and polyphonic music scores are treated in separate subsections.

5.1 Justification functions

When a monophonic measure is justified, there is only a single voice. In this case, each simultaneity contains a symbol of the measure, see Figure 15. No NULL elements are inserted during the preprocessing phase in the voices, so $\vec{v}' = \vec{v}$.

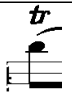
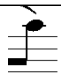
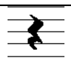
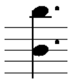
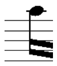
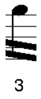
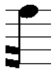
In this case, the time duration of the simultaneity is equal to the time duration of the figure it contains: $s_i.d = f_i.d$. The width of each the simultaneity is the same as the width of the figure and it can be estimated on the basis of the duration of the figure:

$$\boldsymbol{s_i.w = f_i.w = J(f_i.d)},$$

where: *J()* is the function that associates a certain space (width) with a certain time duration. In music scores the usual unit of measure for horizontal distances is the *note head space* that is the width of a black note head, thus a 2.5 spaces distance is equivalent to two note heads and a half. In the following, three of the most used functions for spacing a line of music are listed. Among the possible functions *J(.)* which have been proposed in literature, the simplest is the linear function:

(1) $$J(f_i.d) = a \cdot f_i.d\,,$$

where: constant a represents the scale factor and the change of unit between duration and width.

Vln I							
Beat	0	1/8	1/4	2/4	7/8	22/24	23/24
Duration	1/8	1/8	1/4	3/8	1/12	1/12	1/12

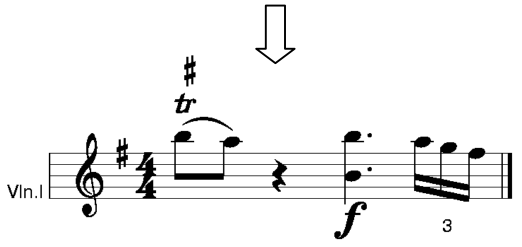


Figure 15 - Simultaneities for monophonic music

The main drawback of the linear function is the lack of compactness of the measures produced when there are short and long notes in the same measure (in monophonic music, measure and voice can be referred to as synonyms). The linear approach associates the shortest figure (which is 1/128 in most music notation editors) with the minimum space possible to avoid overstriking. Longer figures are associated with a space calculated proportionally. This approach usually determines larger spaces provoked by longer notes. Instead of associating the shortest possible space with the figure of 1/128, it could be better to associate such shortest possible space with the shortest figure of the measure by choosing a proper value of a. This allows obtaining more compact measures. The duration of the shortest figure of the measure is defined as:

$$m = \min\{f_1.d, \ldots, f_i.d, \ldots, f_{\max}.d\}$$

where *max* is the number of measure figures.

Choosing an a value dependant on *m*, *a(m)* , the function of justification can be written as

(1a) $$J(f_i.d, m) = a(m) \cdot f_i.d$$

Function (1a) produces slightly more compact measures than those produced by using equation (1). On the other hand, it produces less compact measures when short and long notes alternate in the same measure.

A different function was proposed by Byrd [1],

(2) $$J(f_i.d) = ab^{\log_2 f_i.d}$$

where *a* and *b* are constants used to tune the resulting space. This function leads to an increase in the calculated space by a factor *b* for a doubling of *fi.d* This function offers better results for compactness than those obtained with the linear function. On the other hand, it produces too large spaces for figures with long duration. Dependencies of these constants on *m*, *a(m)* and *b(m),* as done for the previous function, can be introduced to optimize the function.

The following function has been proposed by Gourlay in [6]. This produces a compact measure even when short and longer notes alternate in the same measure:

(3) $$J(f_i.d, m) = \log_2(f_i.d) + T(m, K)$$

where: factor *T(m,K)* is defined as

(3a) $$T(m, K) = K - \log_2(\min\{m, 1/8\})$$

where: ***m*** is the duration of the shortest figure of the measure, and *K* is a variable parameter. If the shortest figure of the measure is 1/8 or shorter, it receives a width of ***K*** *noteheads* of space (in the expression of ***T(m, K) K*** value is usually set to be about 2) and longer figures according to the function calculation. If the shortest figure of the measure is longer than a 1/8 note, then the spaces of the figures are computed as if there were 1/8 figures in the measure so as to avoid associating a too short space with a long note. In the above expression of ***T(m, K)***, ***K*** can be used as input value. This value is used as a tuning parameter for stretching/shrinking the resulting justified measure.

Above considerations do not apply whenever calculating the width of the simultaneities having no duration; in such event only the physical width of the symbols contained in the simultaneity must be taken into account. The space associated with these simultaneities is equal to their *box width*, which is the minimum space to be inserted to avoid overstriking with the musical symbols of the following simultaneity.

5.2 Width of simultaneities in polyphonic music

The functions introduced in the preceding paragraph are used also in polyphonic music to calculate the width of the simultaneities. The complexity of the justification task is greater for polyphonic music than for monophonic music. In polyphonic music, each simultaneity, according to its definition and construction principle, contains *at least* one figure that starts on that simultaneity (see Figure 13 and Figure 14). Generally speaking a simultaneity may contain several figures of different duration and also several *NULL* elements. A *NULL* element, as already explained, corresponds to a figure in a preceding simultaneity in the same voice whose temporal duration has not expired at the beat of the considered simultaneity. The presence of more elements, in comparison with monophonic music, brings forth a wider choice, but also more ambiguity when it comes to decide which is the duration to be used in the preceding formulas. In the polyphonic music justification functions the duration providing best results for compactness is the duration of *the shortest figure among those starting or continuing at the beat of the simultaneity*, by examining all the voices. We named this duration $d^{\,l}$, where l refers to the l-th simultaneity (see Figure 16, *l-th* column of the structure). $d^{\,l}$ takes into account the duration of the figures starting at l-th simultaneity and the duration of the figures whose duration has not expired at l-th simultaneity, for each one of the *max* voices that have to be justified in parallel:

$$\boldsymbol{d^l = \min\{d^l{}_1,\ldots,d^l{}_i,\ldots,d^l{}_{\max}\}}$$

where, indicating with $\vec{s} = \{s_1,\ldots,s_l,\ldots,s_R\}$ the simultaneities of the structure preprocessed as in Section 4 and $s_l = \{f'_{1l},\ldots,f'_{il},\ldots,f'_{\max l}\}$ the elements of the l-th simultaneities

$$d^l{}_i = \begin{cases} f'_{il} & \textit{if } f'_{il}\ {}^1\ \textit{NULL} \\ f'_{ij} & \textit{otherwise, } f'_{ij}\ {}^1\ \textit{NULL with } j£\ l \end{cases}$$

If the element $f'_{il}\ {}^1$ ***NULL*** then for that voice (i-th) the value considered is the duration of the figure. Otherwise, if the element $\boldsymbol{f'_{il}}$ **= NULL**, the value considered is the duration of the figure whose value has not expired at the l-th simultaneity; this figure is indicated in the formula as starting at j-th simultaneity, with $j£ l$.

In the formulas at each simultaneity, of all the values d^l_i determined for all the voices, only the minimum value is used. The above expressions do not take into account the presence of figures without duration. They are treated in a separate manner, as outlined in the previous paragraph. For the simultaneities containing figures without duration the width is not determined by the justification functions introduced above.

Sometimes the shortest figure of the current l-th simultaneity, as defined according to the precedent methodology, does not expire at the beat of the (l+1)-th simultaneity. In Figure 16, for example, the duration of the simultaneity number 6 is only a fraction of the duration of the respective shortest figure, $d^{\,l}$, and this is also true for simultaneity 8. This fraction ranges from 0 to 1 and is defined as the ratio between the duration of the simultaneity and the duration of the shortest figure of the simultaneity:

$$\boldsymbol{e_l = \frac{s_{l+1}.b - s_l.b}{d^l}}$$

where $\boldsymbol{s_l.d= s_{l+1}.b- s_l.b}$ is the duration of l-th simultaneity and e_l is the *fraction of* d^l *elapsing before the next simultaneity*. For example (see Figure 16), at simultaneity 6 the shortest duration figure is 3/8, but only 1/8 has still to expire at the beat 6-th. Therefore, 1/8 is the duration of the simultaneity: e_l = (1/8)/(3/8)=1/3 for simultaneity number 6. The duration of the simultaneity 8 is (3/4-2/4)=1/4 and the shortest duration figure is 2/4, then the value of e_l=(1/4)/(2/4)=1/2.

In monophonic music, all the simultaneities have a value of e_l=1, while in polyphonic music such value may change. The value e_l introduced in the above formulas (1a), (2) and (3) must be adapted to work with polyphonic music.

A better balance of justification spaces is obtained if the width of the simultaneity calculated with the value d^l is weighted by e_l. Please note that, generally speaking, $e_l \neq 1$ when l+1-th simultaneity has a NULL element by the voice where the shortest duration figure of duration d^l for l-th simultaneity is located (see simultaneities 6 and 8 in Figure 16). On the other hand, e_l =1 when l+1-th simultaneity has a note or a rest by the voice where the shortest duration figure of duration d^l for l-th simultaneity is located (see simultaneities 7 and 9 in Figure 16).

According to what stated above, the width of a simultaneity, when it comes to justification of several parallel voices, is calculated with the following expression. It is an extension of the spacing formula (3),

$$s_l.w = J(d^l, T) = e_l \cdot (\log_2 d^l + T) \quad , \tag{4}$$

where: $s_l.w$ is the width of the simultaneity in *noteheads* units, and factor ***T*** is defined as in equation (3a).

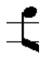
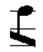
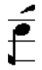
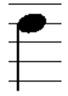
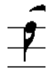
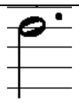
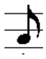
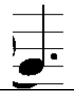
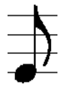
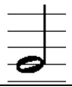
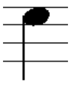
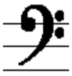
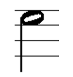
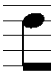
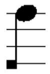
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Vln I							NULL	NULL	NULL	NULL
Vln II	NULL	NULL			NULL	NULL			NULL	NULL
Viola	NULL	NULL	NULL				NULL	NULL		
Beat, $s_l.b$	X	X	X	0	X	1/4	3/8	2/4	3/4	7/8
d_l	X	X	X	1/4	X	3/8	1/8	2/4	1/8	1/8
e_l	X	X	X	1	X	1/3	1	1/2	1	1
Width, $s_l.w$				4		1.528	3	2.5	3	3

Figure 16 - Values of d_l (shortest duration figure of simultaneity), e_l (fraction of shortest duration figure of simultaneity to elapse) and $s_l.w$ (width of the simultaneity)

The extension of a linear function (1) in the event of polyphony is:

$$s_l.w = J(d^l, m) = e_l \cdot d^l \cdot f_c \tag{5}$$

Here e_l , d^l and m are defined as in equation (1) and $f_c = \frac{K}{m}$. ***K*** is an input value used as a tuning parameter for the stretching/shrinking of the measure and is equivalent to the width given to a note of an eight when such note is the shortest one in the measure (this value is typically close to 2).

This function has the same drawbacks of the linear functions, which means it produces too large spaces for big notes, being built on a direct proportionality between duration and space.

Figure 16 reports the values of d^l, e_l and $s_l.w$ of the example in Figure 13. Widths were obtained by using the logarithmic spacing function (4) with K = 3 as the tuning parameter.This figure does not make any reference to the width of the dummy simultaneities containing symbols without duration because dummy simultaneities are discussed in paragraph 5.4.

Simultaneity 7 contains one figure and two NULL elements. The figure is an eight note in the second voice, while the NULL element corresponds to the continuation of a 3/4 note in the first voice (2/4 and an augmentation dot) and a 2/4 note in the third voice. The shortest duration figure is the eight note, so the value d^l=1/8; the note fully expires at the next simultaneity (the beat of the next simultaneity is 2/4, exactly 1/8 after the beat 3/8 where falls the eight note), then e_l = 1. Considering that the shortest figure of the measure has m=1/8, the value of $T(1/8,3) = 3 - \log_2 1/8 = 6$. The spacing function, according to the beatlines of the simultaneity 7 and 8 should be

$$\boldsymbol{s}_1\boldsymbol{.w} = \boldsymbol{e}_l \cdot (\mathbf{log}_2\, \boldsymbol{d}^l + \boldsymbol{T}) = 1 \cdot (\mathbf{log}_2\, 1/8 + 6) = 3 \textit{ space units.}$$

The sixth simultaneity contains two figures and a NULL element. There is a 3/4 note in the first voice (2/4 and an augmentation dot) and a 2/4 note in the third voice. In the second voice, the NULL element represents the continuation of the 3/8 figure of second simultaneity (1/4 note and an augmentation dot): this is then the shortest duration figure and d^l=3/8. The sixth simultaneity falls on the beat 1/4 and the shortest duration figure falls on the beat 0: of the duration 3/8 of the shortest duration figure then an 1/8 has still to expire at the sixth simultaneity, which is a third of its duration, then e_l =1/3. As above,

$$\boldsymbol{s}_1\boldsymbol{.w} = \boldsymbol{e}_l \cdot (\mathbf{log}_2\, \boldsymbol{d}^l + \boldsymbol{T}) = 1/3 \cdot (\mathbf{log}_2\, 3/8 + 6) = 1\mathbf{.}528 \textit{ space units}$$

The second simultaneity contains three figures and no NULL elements, among these figures the shortest has the value of d^l=1/4 (such value being derived both from the first and the third voice equally) and it fully expires at the next simultaneity (the dummy simultaneities are not considered), then e_l =1.

The spacing function, according to the beatlines of simultaneity 1 and 2 is:

$$\boldsymbol{s}_1\boldsymbol{.w} = \boldsymbol{e}_l \cdot (\mathbf{log}_2\, \boldsymbol{d}^l + \boldsymbol{T}) = 1 \cdot (\mathbf{log}_2\, 1/4 + 6) = 4 \textit{ space units}$$

Please note that according to the selected algorithm this would be the space between the second simultaneity and the third one having a duration. Since there is a dummy simultaneity between the two simultaneities with duration (4 and 6) some additional space may be needed for the symbols without duration, namely grace note and clef change. In this particular example, there was no need to insert additional space, since the current space resulting from formula (4) was large enough to account for the grace note and the clef change and there was no overlapping of symbols. This approach assures that no distortion in space (and in score readability) is produced due to such symbols without duration.

The first three simultaneities are dummy simultaneities: simultaneities 1 and 2 contain a grace note each, on the first voice; simultaneity 3 contains two grace notes on the first and second voice. To account for the beamed group of grace notes, some additional space is inserted before the first figure with duration of the voice. This space depends only on the physical width of the grace notes. An identical space is inserted in all the voices to maintain the vertical alignment of the figures falling on beat 0 (simultaneity 4). Paragraph 5.4 will describe the positioning of non durational symbols on the staff, and how to pass from the positioning reported in Figure 16 to the final representation of the measure as depicted in the lower part of Figure 13.

After the computation of the width of the simultaneities, the computation of the space to be associated with each figure, and representing the space to the next figure (which can span several simultaneities) is simply derived. The width of a figure is derived from the calculations of the width of the simultaneities, see Figure 16, by adding the calculated width of the simultaneity where the figure starts and the width of all the following simultaneities containing NULL elements on that voice. For example, the space associated with the 2/4 note in the third voice, simultaneity 6, is given by adding the width of simultaneities 6, 7 and 8.

5.3 Physical dimensions of musical symbols

The value $s_l.w$ being calculated with justification function (4) produces simultaneity widths that take into account only the duration of the musical symbols, not their effective physical dimensions. On the other hand, figures like notes or chords may have alterations (sharps and flats) which increase a lot the physical width of the symbol. The physical width of a figure with all the related symbols which may increase its width (like alterations, augmentation dots) is named as *box width*. Whether considering the point of the symbol where the beatline falls(see Section 4), they can be individuated as a left box width and a right box width, respectively. The left box width of a figure (influenced mainly by alterations) impacts on the width of the preceding simultaneities. In order to avoid symbol overlapping the space after a figure must be bigger than the sum of its right box width and the left box width of the following figure, *in the same voice*. When symbols with a large box width are present, it becomes sometimes necessary to add some additional space into the width of the figure (space from that figure to the next one) calculated with spacing formulas considering both the physical width of the symbol and its related alterations etc.

In case of polyphony, several scenarios are possible whenever the score has large box width symbols. See for example Figure 17: here the chord presents a large box width due to the presence of alterations. Nevertheless, there is no need of additional space with respect to the space produced by the spacing formulas thanks to the lower layer of 16-th notes. In Figure 17, the box width of the chord spans several simultaneities without affecting vertical alignment of the beatlines. the box width of the chord (red box) is highlighted, the ideal vertical coordinate where the beatline (continuation of the stem of the chord) falls has been marked as well, together with , the right box width (B_R) and the left box width (B_L).

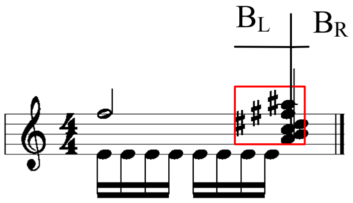


Figure 17 - Box width of a chord spanning various simultaneities with no need of additional space

In Figure 18, there is a configuration of figures with the upper voice needing some additional space for symbols such as augmentation dots and alterations. The additional space is distributed among all the simultaneities between the two figures to preserve the complete readability of the measure. Please note that this measure is not time consistent in the first voice.

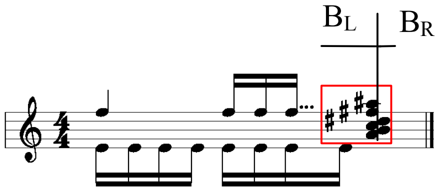


Figure 18 - Additional space spanning various simultaneities

Hereafter the algorithms to check if additional space is needed and to span additional space on various simultaneities are described. The procedure is repeated for each symbol of each voice of the measure. Consider a figure located in the current *l*-th simultaneity and the figure preceding it on the same voice, located for example in the *(l-a)*-th simultaneity. It is possible to examine the chord of Figure 17 for referral. We call *total_box_width* the sum of the right box width of the figure in the *(l-a)*-th simultaneity (for example due to the presence of augmentation dots) and of the left box width of the figure in the *l*-th simultaneity (for example due to the presence of alterations). There will not be overlapping if the sum of the widths of the simultaneities between the two figures, calculated according to the spacing formulas, is bigger than the *total_box_width* between the two figures. We call this difference *z*. It represents the amount of additional space needed to avoid overlapping:

$$\boldsymbol{z = (total\_box\_width - \sum_{l-a}^{l-1} s_x .w)}$$

If *z* is positive, this additional space is distributed among the simultaneities from (*l-a*)-th and (*l*-1)-th. A criterion which is respectful of the duration of the figures is to distribute the additional space among the simultaneities not uniformly, but proportionally to the fraction of the elapsing time e_l of the shortest figure of each simultaneity, as previously calculated. For each *x*-th simultaneity, comprised between *(l-a)* and *(l*-1*)*, the additional space to be summed to the value $s_x.w$ computed with the durational formula is

$$\boldsymbol{\frac{z \cdot e_x}{\sum_{l-a}^{l} e_x}}.$$

After performing the above additions to take into account the physical dimensions of the symbols, it is possible to estimate the final space associated with every figure of the measure. It represents the space with respect to the following figure in the same voice.Finally to determine the right space for each figure (attribute *f_i.w),* it is necessary to sum the widths of the simultaneity where the symbol falls and the width of all the successive simultaneities containing *NULL* in the same voice until a new figure is present, as outlined in the previous paragraph.

5.4 Spacing and positioning symbols without duration

When a measure has symbols without duration, a dummy simultaneity is included. This has a NULL duration and does not fall on a beat. The width associated with the dummy simultaneity depends only on the physical dimensions of the symbol contained. If the dummy simultaneity contains more figures without duration (see columns 3 and 5 in Figure 19) the width of the dummy simultaneity is the width of the largest symbol contained in the dummy simultaneity. For example, in column 5 of Figure 19 the clef is greater than the grace note, so the width of simultaneity 5 is the one of the clef.

The spaces of the dummy simultaneities may influence the width of the *active* figures such as notes and rests falling in real simultaneities. If there is enough space for the non-durational symbol, such space adding is performed without affecting the space determined with the spacing formulas, otherwise some additional space may be added to the preceding figure to avoid overlapping. For example, see simultaneity 4 in Figure 19, the space from the fourth note in first voice (Vln I) and the following note of 3/4 is equal to 4 *spaces,* and no additional space is needed for the grace note. When positioning the symbols on the staff, the 3/4 note will be drawn 4 *spaces* on the right of the fourth note. Enough space is left in the middle of the two notes to draw the grace note.

For the grace notes of the first three simultaneities in the first voice (a beam of grace notes), and in the second voice (a single grace note), some space must be added before the first figures with duration of the measure. In the case of measure depicted in Figure 19, (0.7+0.7+0.7)=2.1 *spaces* have to be left before the first active figure of the measure for all the three voices being justified.

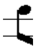
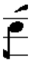
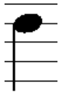
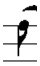
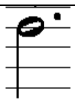
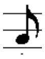
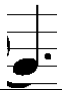
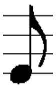
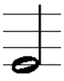
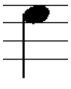
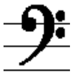
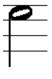
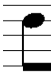
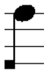
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Vln I							NULL	NULL	NULL	NULL
Vln II	NULL	NULL			NULL	NULL			NULL	NULL
Viola	NULL	NULL	NULL				NULL	NULL		
Width	**0.7**	**0.7**	**0.7**	4	$\boxed{\mathbf{2.3}}$	1.528	3	2.5	3	3

Figure 19 - Widths of dummy simultaneities

Please note that musically and visually it is important that a grace note is drawn in the close left neighborhood of the note it refers to (see Figure 13, for the final draw). The same consideration can be held for a beam of grace notes, which has to be drawn on the left of the note it refers to, and typically linked to it by a slur, and a similar consideration is also valid for a clef change. In that case, small notes are ornaments of the reference note. In other occasions, small notes are used as cue notes to recall the attack of the instrument with respect to some beats of another reference instrument. In that case, they have to be justified and aligned to normal sized notes.

As to the positioning of grace notes, it has to take into account the position of the note grace notes are referred to. The procedure is as follows. The physical dimension of the grace note is subtracted from the horizontal coordinate of the figure with duration it refers to, to obtain the *x*-axis coordinate to position the grace. The previous calculation of symbols physical dimensions grants the absence of any overlapping. The procedure also assures that the grace note is positioned in the close left neighborhood of the note it refers to (see Figure 13, the grace note of simultaneity 5 in the first voice). The positioning of the clef change of simultaneity 5 in the third voice follows a similar procedure. A similar procedure is also needed to place a beam of grace notes before a figure with duration. In this case, in order to calculate the position where to draw every single grace note of the beam ,the number of grace notes of the beam must be considered as well. In Figure 13, by considering the beam of grace notes at the very beginning of the measure, the first grace note of the beam is displayed at the coordinate given by

*coordinate_of_durational_figure - (number_of_grace_notes * width_of_a_ grace_note)*

6 LINE BREAKING

The spaces computed with the formulas introduced in the previous paragraph produce measures that have the so-called *natural widths.* They are the result of a mathematical computation of the justification algorithm by using the constants imposed by the users. As mentioned in Section 3, the process to stretch or compress the measures so as to to fit the page width is called line-breaking.

The line-breaking algorithm adopted in our justification engine is quite flexible, robust and fast with respect to other solutions such as the one proposed in [7], which works only with single parts. The approach proposed can fit both main scores and single parts.

First of all the measures are laid on the page with their natural widths: when adding a measure would result in exceeding the right margin of the page, that measure is removed from the line (see Figure 20). The remaining blank space at the end of the music line must be distributed among the measures of the line proportionally to their natural widths. According to this distribution, a *target width* for each measure is estimated. This procedure results always in a measure stretching. This means that the user has to consider the ***K*** related to the justification algorithm as the minimum value of compression for each selected measure.

A linear or random redistribution of the remaining space at the end of the line to the single figures would result generally in a loss of the alignment and justification properties of the measure/voices. A better solution is to stretch the measures by working on the tuning parameter of the justification algorithm. This allows to keep the durational justification consistent and the vertical alignment for the symbols of the measure unchanged/correct.

Please note that, in the general case of polyphony, with durational and non-durational figures, the justification process for estimating the spacing of all the figures in a measure is not linear and there is not a linear relation between the tuning parameter and the resulting width of the measure. In addition, the formula obtained summing the width of all the simultaneities of the measure to calculate the total width of the measure cannot be inverted in order to estimate the tuning parameter in function of the total width of the measure. This is due to the fact that the tuning factor is spanned in more simultaneities, and results cannot be generalized. The problem of stretching a measure to a certain target width is basically the problem of finding the precise value of the tuning parameter producing the exact target width for a measure. The stretching process of the measure to the target width is obtained by using an iterative process which includes the justification task.

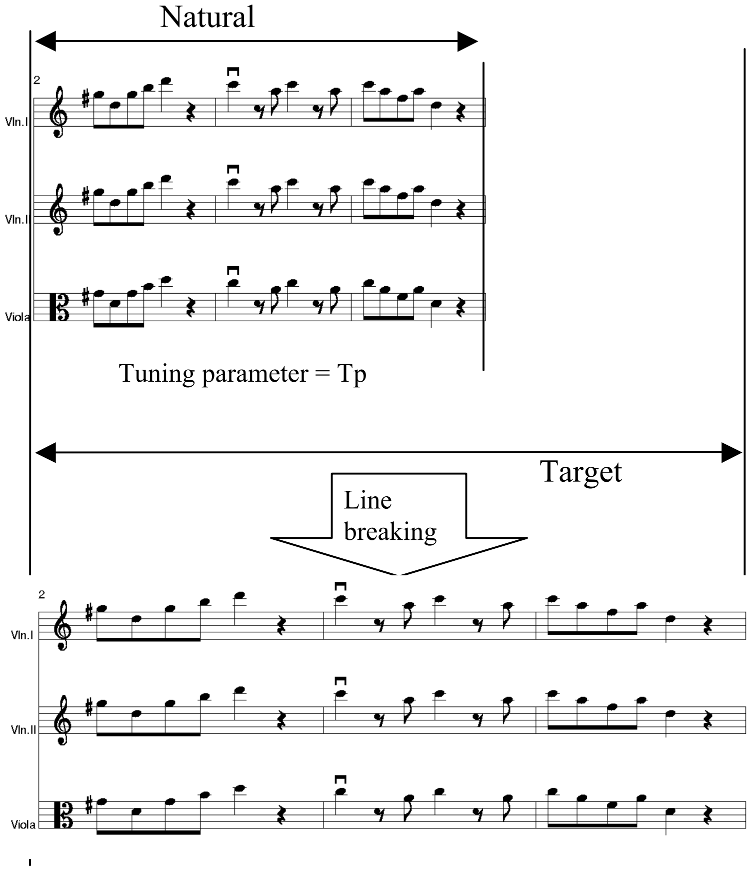


Figure 20 - Natural width and Target width, natural and line-breaked measures. In this case, the decision was taken as to present only three measures in the line.

The justification algorithm is invoked to verify the resulting width of the measure with a certain tuning parameter; the process of finding the desired tuning parameter has to be reliable and very fast so as to minimize the occurrences of such justification algorithm invoking.

An approach based on the bisection algorithm has been used. The convergence of this algorithm to the desired tuning parameter is guaranteed by the relation (between the tuning parameter and the resulting width) being monotone and continuous (see Figure 21). As a first step, two values for the tuning parameter *T1* and *T2* are searched, which could meet the following conditions:

- *T1* that produces a measure width lower than the Target Width
- *T2* that produces a measure width greater than the Target Width

They are assigned by imposing:

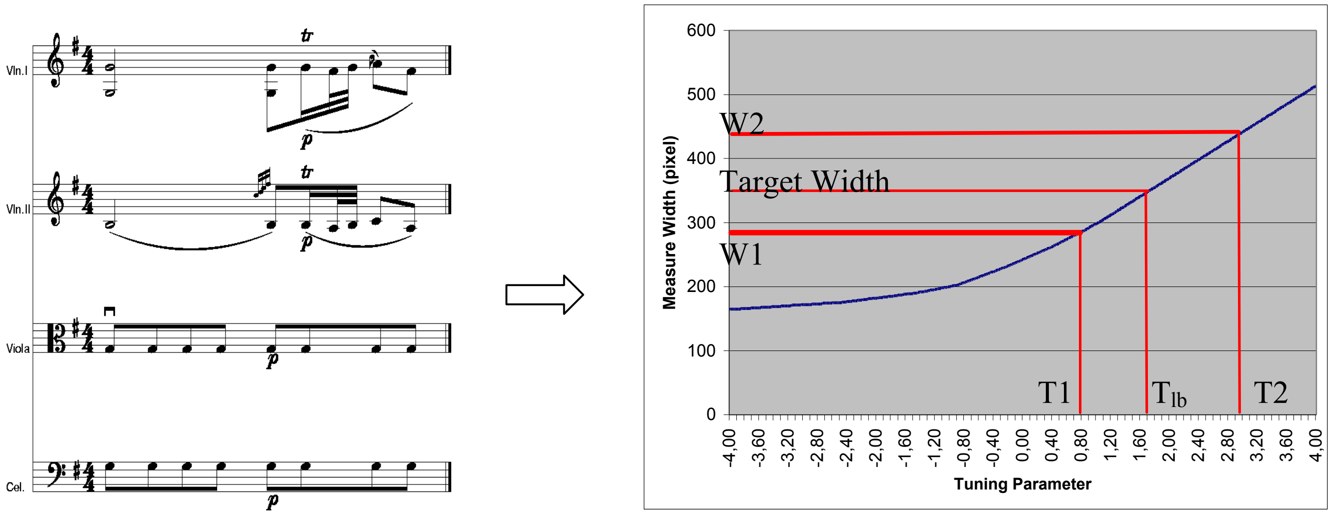
- *T1=T* where *T* is the value imposed by the user for estimating the Natural Width (this is surely the lower bound since the process is always stretching, and never shrinking),

- $T2=n\ T$ where n is the first value that produces a measure width greater than the target width and it is taken from the progression $n=2,4,8,16....$

The next step is to calculate a medium point $MT=(T1+T2)/2$ and its corresponding width, *Wmt*. This width may result with a given error of approximation *E*:

- *Wmt + E > Target Width*: then *T2=MT*, T2 is updated;
- *Wmt - E < Target Width*: then *T1=MT*, T1 us updated;
- | *Wmt - Target Width* | < *E* : then the *MT* is the searched value.

In Figure 21, the typical trend of the width with respect to the tuning parameter is presented. It refers to a logarithmically justified measure with the initial tuning parameter imposed by the user equal to 0.8. With the same logarithmic function and varying the tuning parameter the resulting width of the measure would be the value reported on the y-axis. Similar trends have been noticed for a wide range of music examples and what could be observed is that the resulting relation varies in slope and has slight variation in shapes. Their shape does not affect the convergence of the algorithm.



Initial Tuning parameter = T1 = 0.8

Figure 21 – Line breaking convergence at the Target Width for logarithmic formulas

Two excerpts from "*The Spring*" of A. Vivaldi are presented (Copyrighted by CASA Ricordi). Figure 22 is the main score, wile Figure 23 presents a single part.

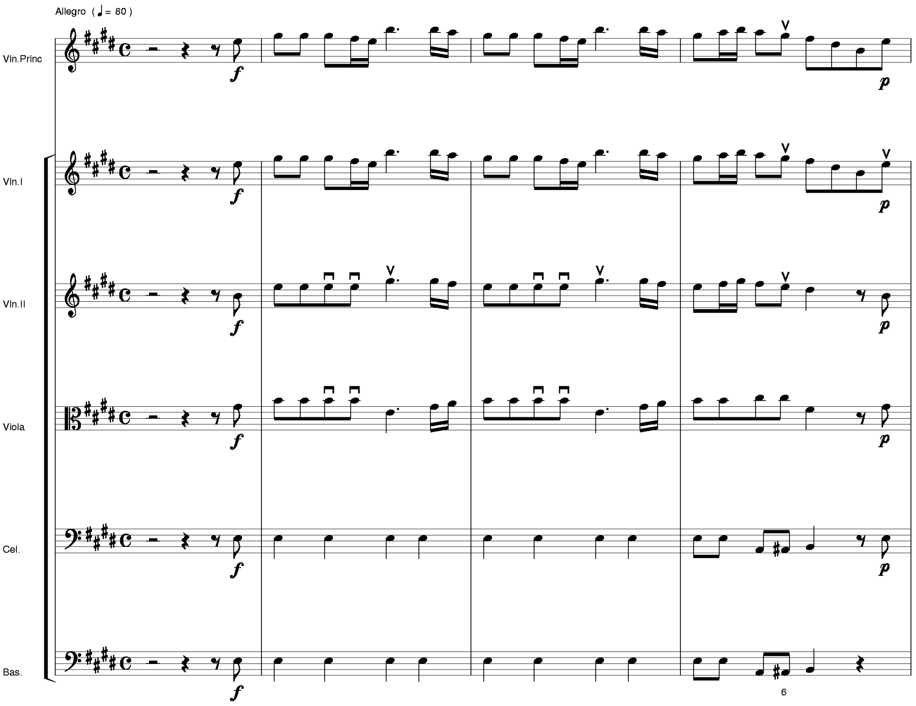


Figure 22 - Right margin alignment of a main score.
The resulting page is formatted ready to be published.

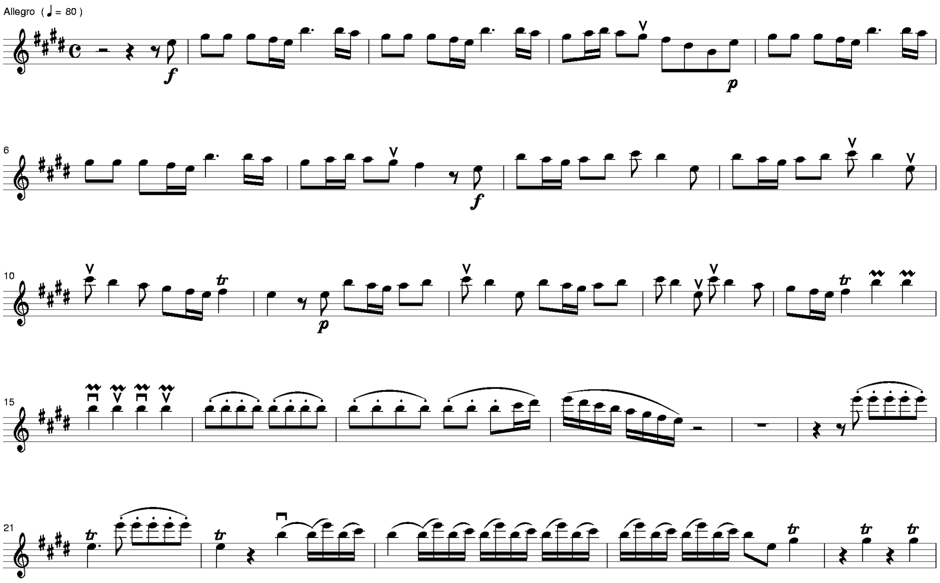


Figure 23 - Right margin alignment of a single part.
The resulting page is formatted ready to be published.

Figure 23 shows the line-breaked music page of the first measures in the part of the First Violin (indicated *Vln Princ.* in Figure 22).

7 CONCLUSIONS

Music justification is a complex task to be performed automatically, involving the evaluation of a large number of parameters and requiring context evaluation. The music justification solution proposed in this paper could solve many of the most common problems: vertical alignment of time simultaneous symbols in polyphonic music, durational spacing, compactness and readability of the resulting measure, management of symbols without duration, justification of main score and parts, etc. Justification has been implemented through a robust algorithm able to cope with many exceptions, for example any time inconsistency of the justified measure and presence of non durational figures. The algorithm has been used by the line-breaking process which has allowed to produce justified and perfectly margined music lines. A certain degree of gain in the readability for execution was achieved together with a certain degree of smartness in the layout. Line-breaking has been implemented through an iterative process. The solution proposed is based on the WEDELMUSIC model, a model presenting a strong distinction between logical description and visualizing aspects of music notation symbols. Future development is expected to account for enhancing the algorithm, so as to cope with several music notation scenarios and scores having parts written in different time signatures (for example Mozart's "Don Giovanni"). In this case automatic justification must work with a different number of measures for the different parts. For example, if one part is written in 4/4 and another in 3/4, the measure barlines of the two parts are not aligned, being necessary to keep unchanged the alignment of simultaneous symbols. In this scenario the base unit of line-breaking are three measures for the part in 4/4 and four measures for the part in 3/4.

8 ACKNOWLEDGMENTS

The authors express their thanks to several experts belonging to the user-group of WEDELMUSIC project which made possible the identification of requirements and problems, and contributed a lot to improve the algorithm by highlighting criticisms and providing examples. A warm thank to the project partners and to the European Commission for funding the project and in particular to Maestro N. Mitolo and Maestro L. Fiorentini of the SMF for their suggestions.

9 REFERENCES

[1] Byrd, D., “Music Notation by Computer”, Indiana University, Dept. Of Computer Science, Ph. D. Thesis, 1984.

[2] Dannenberg, R., “A Structure for Representing, Displaying and Editing Music”, In P. Berg (editor), Proceedings of the International Computer Music Conference 1986, pages 153-160. International Computer Music Association, 1986.

[3] Dannenberg, R., “A Brief Survey of Music Representation Issues, Techniques, and Systems”, published as: Dannenberg, “Music Representation Issues, Techniques, and Systems,” Computer Music Journal, 17(3), pp. 20-30., 1993.

[4] Gourlay J., S., Parrish, A., Roush, D., Sola, F., J. and Tien, Y., “Computer Formatting of Music,” Technical Report, OSU-CISRC-2/87-TR3, Department of Computer and Information Science, The Ohio State University, 1987.

[5] Gourlay, J., S., “A Language for Music Printing”, Communications of the ACM, Vol.29, May, 1986.

[6] Gourlay, J., S., “Spacing a Line of Music”, Technical Report, OSU-CISRC-10/87-TR35, Department of Computer and Information Science, The Ohio State University, 1987.

[7] Hegazy, W., A., and Gourlay, J., S., “Optimal Line Breaking in Music,” Ohio State University, Department Of Computer and Information Science, OSU-CISRC-10/87-TR33, 1987.

[8] Heussenstamm, G., “The Norton Manual of Music Notation”, Ed. W. W. Norton & Company, Inc., 1987.

[9] Gourlay, J., S. and Parrish, A., “Computer Formatting of Musical Simultaneities,” Ohio State University, Dept. Of Computer and Info. Science, OSU-CISRC-10/87-TR28, 1987.

[10] Parrish, A., Hegazy, W. A., Gourlay, J. S., Roush, D. K., and Sola, F., J., “Musicopy: An Automated Music Formatting System,” Ohio State University, Dept. Of Computer and Info. Science, OSU-CISRC-10/87-TR29, 1987.

[11] Rader, G., M., “Creating Printed Music Automatically”, IEEE Computer, June 1996, 1996.

[12] Ross, T., “The Art of Music Engraving and Processing”, Hansen Books, Miami, 1970.

[13] Roush D., “Music Formatting Guidelines” Ohio State University, Dept. Of Computer and Info. Science, OSU-CISRC-3/88-TR10, 1988.

[14] Bellini, P., Bethelemy, J., Bruno, I., Nesi, P., Spinu, M. B. (2003). Multimedia Music Sharing among Mediateques, Archives and Distribution to their attendees. Journal on Applied Artificial Intelligence, Taylor and Francis. http://www.wedelmusic.org

[15] Wood, D., “Hemidemisemiquavers...and other such things. A concise guide to music notation”, The Heritag Music Press, Dayton, Ohio, USA, 1989.

[16] Bellini, P., Della Santa, R., Nesi, P., “Automatic Formatting of Music Sheets”, proceeding of the International Conference on WEB Delivering of Music, WEDELMUSIC2001, IEEE Press, Florence, 23-24 November 2001.

[17] Blostein, D., Haken, L., “Justification of printed music”, Communications of the ACM, Vol.34, No.3, March

1991.

[18] Tatem, J., Sochinski, J., and Roach, J., "Spatial Planning for Musical Output", International Journal of Intelligent Systems, Vol. 4, 1989, pp. 201-218.

[19] Haken, L., Blostein, D., "A New Algorithm for Horizontal Spacing of Printed Music", International Computer Music Conference, Banff, pp. 118-119, Sept. 1995.

[20] Bellini, P., Fioravanti, F., Nesi, P., ``Managing Music in Orchestras", IEEE Computer, IEEE Press, ISSN 0018-9162, pp.26-34, Sept. 1999.

[21] Bellini, P., Nesi, P., "WEDELMUSIC FORMAT: An XML Music Notation Format for Emerging Applications", Proceedings of the 1st International Conference of Web Delivering of Music. 23-24 November, Florence, Italy, pages.79-86, IEEE press.

[22] Bellini, P., Nesi, P., Spinu, M. B., "Cooperative Visual Manipulation of Music Notation", ACM Transactions on Computer-Human Interaction, September, Vol.9, N.3, pp.94-237, 2002.