Ircam documentation

- Research reports
- Musical works
- Software

PatchWork

Kant

Rhythm Quantification Editor

user's manual

Second Edition, April 1996

IRCAM Centre Georges Pompidou

This manual was written by Gérard Assayag, with proofreading by J. Fineberg, and was produced under the editorial responsibility of Marc Battier, Marketing Office, Ircam.

Patchwork was conceived and programmed by
Mikael Laurson, Camilo Rueda, and Jacques Duthen.

The Kant library was conceived and programmed by
C. Agon, G. Assayag, and C. Rueda, with musical expertise by J. Fineberg. Special thanks to V. Verière.

Second edition of the documentation, April 1996
This documentation corresponds to version 1.1 of the library, and to version 2.1 or higher of PatchWork.

Ircam
1 place Igor-Stravinsky
F-75004 Paris
Tel. (33) (1) 44 78 12 33
Fax (33) (1) 42 77 29 47
E-mail : ircam-doc@ircam.fr

IRCAM Users group

The use of this software and its documentation is restricted to members of the Ircam software users group. For any supplementary information, contact:

Marketing Office
Ircam
Place Stravinsky, 1
Paris 75004
France

Tel. (1) 44 78 49 62
Fax (1) 42 77 29 47
E-mail: bousac@ircam.fr

Send comments or suggestions to the editor:

E-mail: ircam-doc@ircam.fr

Mail: Marc Battier, address above

Contents

Résumé

Ce manuel présente la librairie Kant, destinée à la quantification rythmique. L'objet **Kant** effectue une reconnaissance de rythme au moyen d'un algorithme d'approximation qui lui permet de proposer plusieurs résultats.

A ce manuel s'ajoute une documentation plus récente, appelée *Kant 1.5 — Addendum*.

Foreword

Kant is a new model for rhythm quantification where non-periodic segmentation of the input stream is performed prior to tempo recognition. Beat duration is computed by an approximation algorithm equivalent to the one used at Ircam for virtual fundamental extraction. The result is a score where the meter structure is allowed to vary highly — in order to accommodate the particular grouping structure delivered by the segmentation analysis — and is consistent with the tempo. This system is designed to permit the quantification of the complex temporal structures of contemporary music into the changing meters and tempos of contemporary notation.

1 Introduction

The problem of quantification that is generally treated by musical sequencing and notation software is that of eliminating performance fluctuations. Thus an intelligent quantizer would know that, in the style of Chopin, quintuplet accompaniment figures are often played as a slightly accelerating figure of five notes occupying the place of one beat; these shrinking durations had been notated by Chopin as five equal notes.

The direct use by composers of temporal structure, however, poses a very different (although related) set of problems. A contemporary composer who wished to include gently accelerating accompaniment figures, as did Chopin, would be unlikely to consider the acceleration as a detail of execution, assuming that a performer should be able to guess the intention: this would be even more important if the accelerating figures contained varying numbers of notes.

The diversity of styles that coexist in contemporary music has, for better or worse, forced composers to communicate explicitly their intentions rather than imagining all performers to be familiar with their style and its associated performance practices. Therefore an intelligent compositional quantizer would search for a notation that clearly expresses the composers intention of a gentle acceleration. A traditional quantizer could, of course, produce this if it were allowed a large resolution (e.g. to the nearest 32nd note). This would however produce a complexity of notation that would be equally masking of the desired effect as would have been Chopin's solution (in either case you must be told that a gentle acceleration is desired since the written rhythms have not made that clear).

The compositional quantizer must search for a way to express the desired structure that is both accurate and simple enough to be understood. In order to succeed it must make use of the notational tools available to contemporary composer (e.g. changes of tempo and meters). Kant is the first generation of a true compositional quantizer. It posseses its own interface which appears as a specialized PatchWork editor and thus permits the intelligent transcription of temporal musical structures from within PatchWork.

Kant divides the task of quantification into several levels. There are strategies for proposing solutions at each of these levels, but the guidance of a composer is required to choose from, add to and/or select various propositions. It is implicit in Kant that musical notation must ultimately be personal to every composer and thus each composer should be able to use this same system to arrive at his own solution. the system tries be flexible at every level, so that even if a particular sequence is a poor fit with the tools at one level, the next level can compensate for any problems in the previous steps solution.

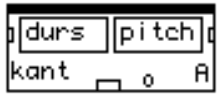
The first step in quantifying a sequence is to divide that sequence into various sections called *archi-measures*. Each archi-measure will be notated in a single tempo.

The second step is to divide each archi-measure into *segments* which will correspond, more or less closely, to measures. This step will provide the metric structure as well as the tempo.

Finally we must resort to a *grid* (similar to that of traditional quantizer, possessing, however, a very different means of choosing the best result).

All of these steps are performed from a single interface which allows interaction and guidance from the composer, permitting him to fine tune the first and second steps as well as the parameters for the last until an acceptable quantification has been found.

2 The Kant Module



Kant Rhythm Quantification Editor

Inputs

durs A list of lists

pitch A list of lists

Output

A list of **c-measure-line** objects.

Kant processes a polyphonic set of duration streams expressed as a list of list of numbers that must be connected to the *durs* parameter. Every single list of number forms a separate voice. Durations are expressed in 1/100th of a second. By convention, rests are expressed as negative numbers. Example: the list ((100 -50 50 100) (80 90 101)) represents 2 voices; the first one contains a 1 second duration, a 1/2 second rest, a 1/2 second duration, then a 1 second duration; the second voice contains three durations, respectively 0.8 seconds, 0.9 seconds and 1.01 seconds.

The *pitch* parameter must be a list of list of midics in the usual PatchWork sense. This list must have the same structure than the duration list (i.e. same number of voices). At the end of the process, pitches will be mapped to durations, one to one, just before the data is sent to the **poly-rtm** editor. If the number of pitches in a voice is smaller than the corresponding number of durations, default pitches (middle C) will be generated at the end of the sequence. By now, pitches must be single pitches, not chords. Example: the list ((6000 6400 6700) (6700 6400 6000)) could be given as a pitch list for the previous duration list. Note that the rests are skipped when mapping pitches to durations; thus we give only three pitches in the first voice, as there are only three plain durations. Please note that for the time being, pitches will be taken into account only in non-polyphonic case (i.e. one voice only).

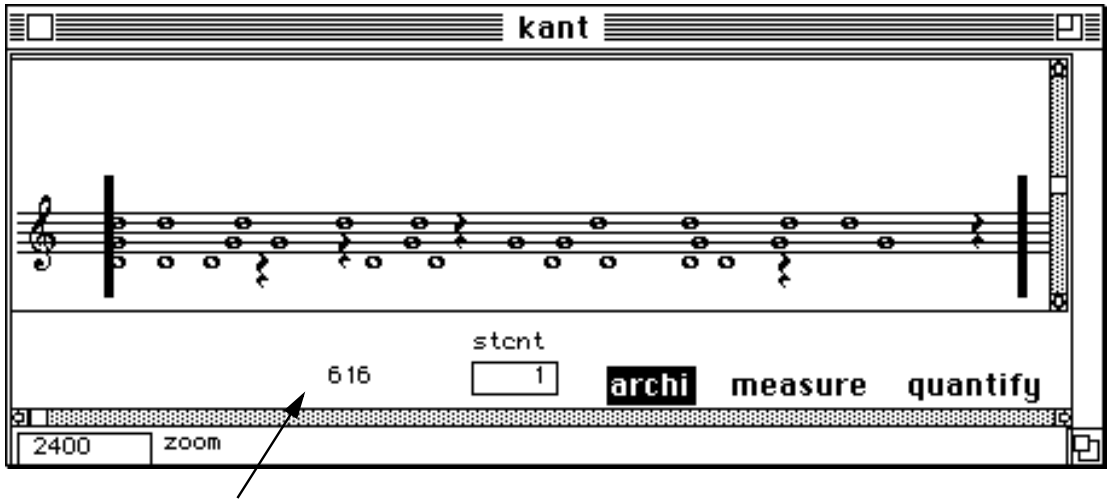
When the user evaluates the output of the **Kant** Box, the result depends on the state of the lock button. If the box is unlocked, the inputs are evaluated, stored into the internal structures of the box, then all the internal state of **Kant** is reset: that is you lose all the work previously done in the editor. The result is then *nil*. If the box is locked, and providing the full quantification process has been performed inside the editor, the result is a list of **c-measure-line** objects, ready to be output to a **poly-rtm** box. Each element in this list is a separate voice that can be send to a **rtm** box.

Saving and restoring a patch with a Kant box

When saving a patch that contains a **Kant** box, remember that the quantification state (i.e. archi-measure positions, measure positions, tempo, etc.) will be lost unless you use the *Save with MN* menu item.

Opening the Kant Editor

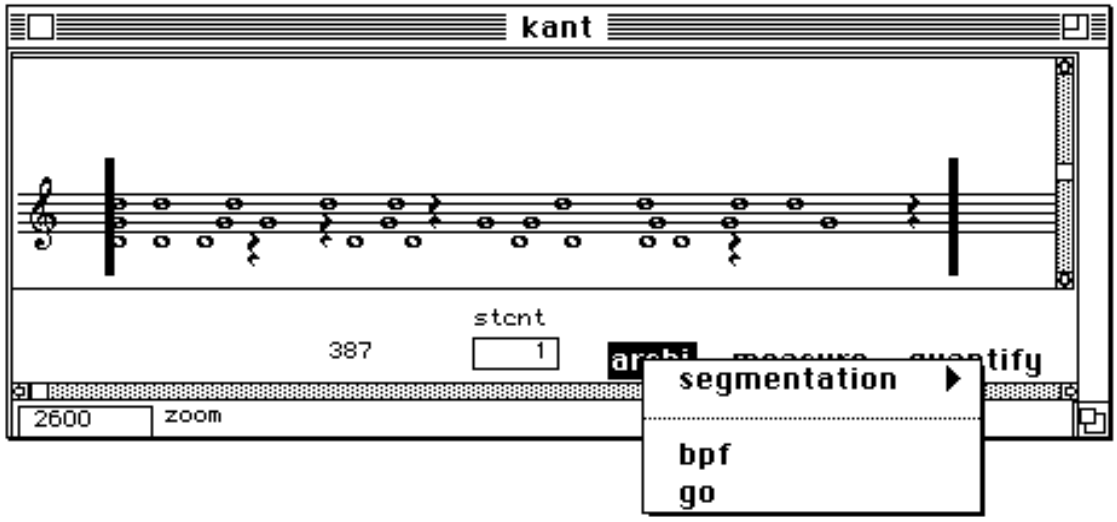
Double-clicking on the **Kant** box after having evaluated it shows a window that looks like this:



Inside the music notation view, only the durations are represented.By convention, each voice is distributed on one line of the staff. Thus we have here a 3-voices sequence. Events onsets are represented by noteheads; rests are represented by the a quarter note rest; events durations are proportional to the space between two consecutive signs.

time coordinates	Indicates the time coordinate at the mouse position in 1/100th of a second.
zoom	Clicking and dragging upwards and downwards inside the box will perform a horizontal zoom-out and zoom-in operation. Option-clicking will increase the zoom step.
stcnt	Staff count. Increasing this value by clicking and dragging upwards and downwards will increase or decrease the number of staff lines in the notation view.
archi	Menu containing the commands that control the archi-measure segmentation phase (step 1).
measure	Menu containing the commands that control the measure segmentation phase and the tempo recognition (step 2).
quantify	Menu containing the commands that control the actual quantification and transcription phase (step 3).

3 Step 1. Segmenting the sequence into archi-measures



The first step inside the **Kant** editor consists in cutting the input sequence into subsequences (**archi measures**) that will be quantified with a single tempo. The object here is to find **articulation points** where the tempo change will happen. By now, you have to do this by hand, as automatic tempo change recognition is still under research. The general principle is that when the durations feel as if they were in a different tempo, they should be notated as such.

Mouse and keyboard operations in the notation view

Insert an articulation point

Option-click anywhere inside the notation view. An articulation point will be inserted above the notehead that is closer to the mouse location. Articulation point appear as a 'Δ' sign above the staff.

Select articulation points

Click on an articulation point symbol 'Δ'. It will become highlighted. If you double-click on any articulation point, the whole set of articulation points will be selected.

Remove articulation points

Select articulation point (s) then press the backspace key on the keyboard.

Move an articulation point

Click on an articulation point ('Δ') to select it. Without releasing the mouse button, move the mouse horizontally. When you're done, release the mouse.

Commands in the menu "archi"

Segmentation

Currently, the only option is manual insertion ("do it yourself"). In the future it will contain commands for automatic archi-measure segmentation.

bpf

Opens a **BPF** (break point function) window containing a graphical representation of the input sequence. Consecutive events are aligned along the x-axis independently from their duration or onset time. For each event, a point in the graph is inserted, whose y-coordinate is proportional to the duration of the event. As short duration events are grouped towards the bottom of the graph and long duration ones are grouped towards the top, it is very easy to localize visually the density distribution and to use this information in order to segment the sequence. See section , "Mouse operations in the bpf view" on page 15 to learn how to insert and remove articulation points inside the bpf window. The proportional notation view and the bpf view are kept mutually informed of occurring editing operation and updated accordingly.

go

Once the articulation points have been inserted, run the command *go* in order to validate the archi-measure segmentation. After *go* has been selected, vertical bars appear where the 'Δ' used to be. The first pair of vertical bars is highlighted, signifying that by default you are going to work further on the first archi-measure. The menu title "archi" is then unhighlighted and the menu title "measure" is highlighted in turn, telling you that you have now the opportunity to get to the second step, i.e. measure segmentation and tempo extraction. You can get back to first step by simply clicking on the menu title **archi**.

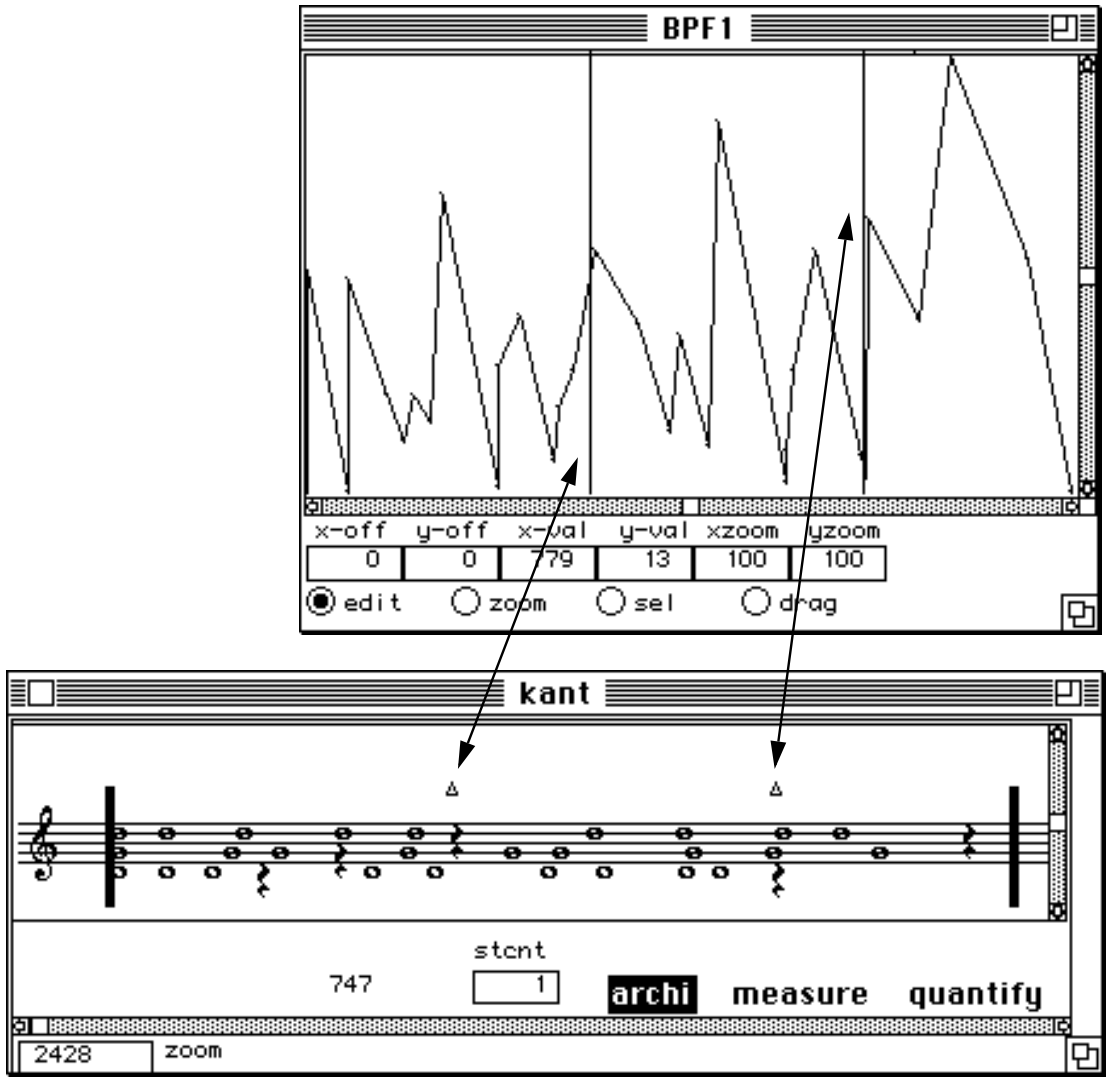
Mouse operations in the bpf view

Insert an articulation point

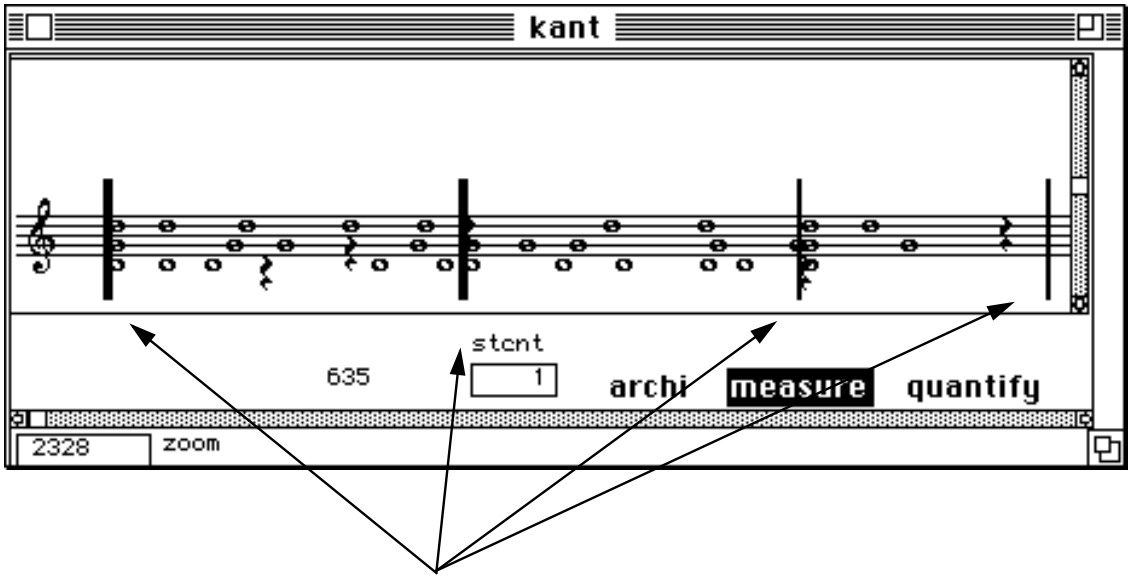
Option-click anywhere inside the **BPF** view. An articulation point will be inserted at the event that is closest to the mouse location. Articulation points appear as vertical lines. Simultaneously, a 'Δ' will be drawn in the music notation view.

Remove an articulation point

Shift-click on a vertical line. The line is erased and the articulation point is removed. The proportional music notation view is updated as well.



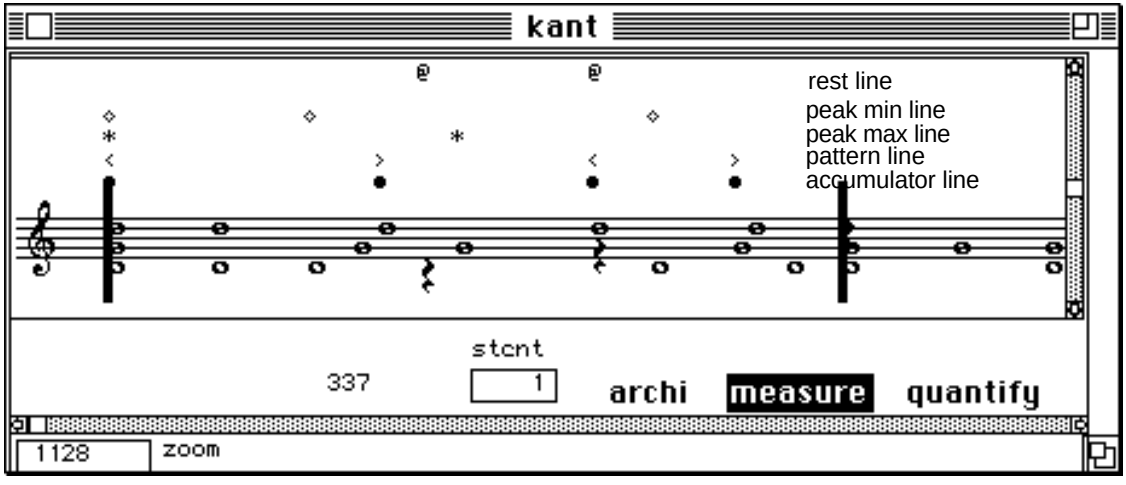
Three archi-measures have been cut by inserting two articulation points.
This is achieved either by editing the notation view (below) or the BPF view (above).



After the "go" command has been run, archi-measures are set and the first one is selected.

4 Step 2. Segmenting one archi-measure into measures

The second step in the quantification process is selecting one archi-measure, then cutting it into **segments** that will later become measures in the final transcription. At the beginning of step 2, the first archi-measure of the sequence is selected by default. Segmentation can be performed by hand or by using one of several automatic segmentation method in the menu **measure->segmentation**. Articulation points between segments appears above the staff line as special signs (• * @ Ø), one for each kind of segmentation. Articulation points for one type of segmentation are displayed along the same, invisible, horizontal line above the staff. The first line above the staff (symbol '•') has a special status: it serves as an **accumulator line** for set-operations between different sets of articulation points. This mean you can perform, for example, the set-union between two sets of articulation points resulting from two different segmentation method. The result of the set-operation is then displayed on the first line above the staff, using the symbol '•'. When performing the *go* command in order to validate the segmentation and pass to step 3, there are two cases: if something has been placed in the accumulator line, only the articulation points from this line will be considered for segmentation. If the accumulator line is empty, all the articulation points appearing on all the other lines will be considered for segmentation. The accumulator line is also the place where articulation points inserted "by hand" will appear.



Mouse and keyboard operations in the notation view

Select an archi-measure

Click on the vertical line to the left of the archi-measure.

Select all archi-measures

Command-click on any vertical line.

Insert an articulation point

Option-click anywhere inside the proportional music notation view. An articulation point will be inserted above the notehead that is closer to the mouse location. Articulation point appear as a '•' sign just above the staff (in the accumulator line).

Select articulation points

Click on an articulation point symbol. It will become highlighted. If you double-click on any articulation point, the whole set of articulation points in the same horizontal line (i.e. for the same segmentation method or accumulator line) will be selected.

Remove articulation points

Select articulation point (s) then press the backspace key on the keyboard.

Move an articulation point

Click on an articulation point to select it. Without releasing the mouse button, move the mouse horizontally. When you're done, release the mouse.

Set-union

Select an articulation point or all articulation points in one line (except in the accumulator set). Press the '+' key. The selected points will be merged into the accumulator line. By repeating this operation you can perform the set-union between any sets of points. The result is always in the accumulator line.

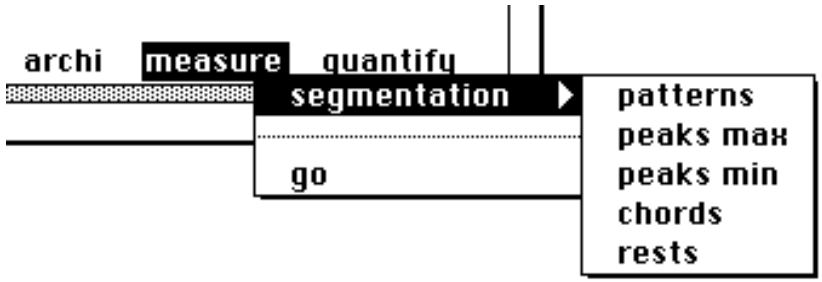
Set-intersection

Select an articulation point or all articulation points in one line (except in the accumulator line). Press the '*' key. Only points in the accumulator that coincide with points in the set will be left in the accumulator. By repeating this operation you can perform the set-intersection between any sets of points. The result is always in the accumulator line.

Chord formation

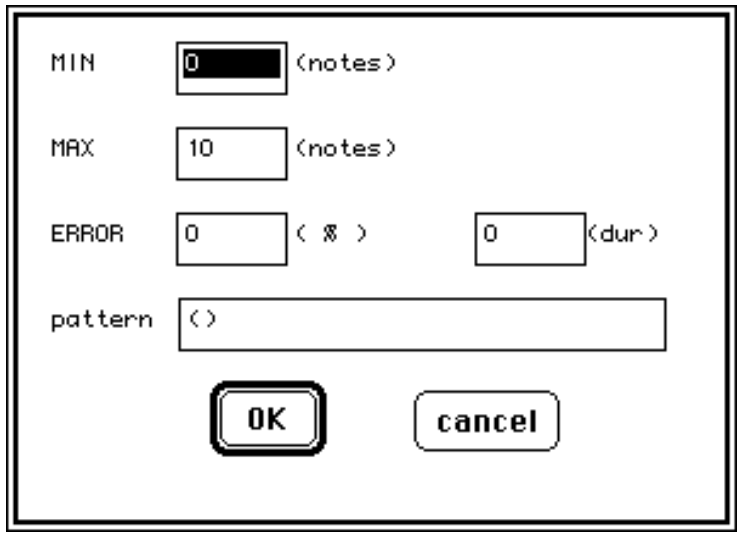
After having built a set of articulation points by using the method **chords** (see below), select one or all the articulation points in the set. Then press the '=' key. Events under a selected articulation point, whose onset times are very close (this is what the chord segmentation method is about), are moved to the group's center of gravity. In the polyphonic case, this forms chords perfectly aligned under the articulation point. If several events on the same voice are moved to the center, all but one are suppressed. This edition mode is destructive, but the original sequence is kept in the system's internal structures, so if you move back from step 2 to step 1 you will retrieve the original.

Commands in the menu "measure"



segmentation->patterns

performs segmentation using the pattern-recognition method. Parameters to be set appear in a dialog box:



MIN	Minimum number of notes inside one pattern.
MAX	Maximum number of notes inside one pattern.
ERROR	Allowed deviation in % or in 1/100th of a second inside one pattern.
pattern	Expression describing the pattern(s) to be recognized.

< means acceleration (decreasing durations), > means deceleration (increasing durations), = means equal durations. * after an expression means that the pattern must be repeated 0 or more times. + after an expression means that the pattern must be repeated 1 or more times. The symbol 'or' expresses an alternative between 2 or more patterns. Two patterns in sequence means they must follow immediately in time. The pattern recognition engine will try to find as many subsequences as it can in the duration sequence, putting articulation points at the boundaries of the subsequences. Recognized subsequences cannot overlap. The recognition engine deals with "fuzzy" recognition, that is it can recognize a subsequence that fits "almost" exactly with the specified pattern. For example inside an accelerando, a few ambiguous notes — whose duration are not smaller than the preceding ones — can be accepted, providing their duration excess stays within a range specified by the error parameter.

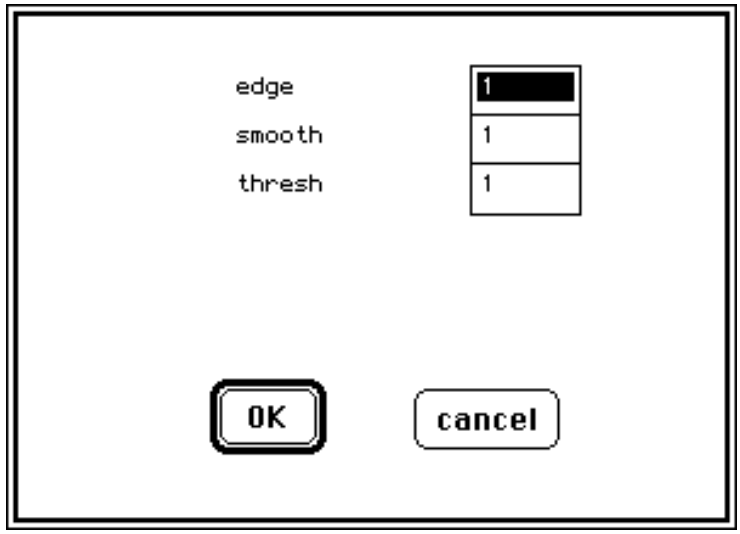
A pattern can be one of the special symbols **<, >, =**, or a pattern can be the combination of other patterns, like **(** *pattern pattern ... pattern* **)**, **(** *pattern* **+) , (** *pattern* ***) , (** *pattern pattern ... pattern* ***or*** **).**

Examples

(< < <)	3 accelerating events
(< < = = > >)	2 accelerating, 2 equal, 2 decelerating events
(< +)	An acceleration of two or more durations
(= +)	two or more equal durations
(> +)	A deceleration of two or more durations
((< +) (= +) (> +))	A subsequence of accelerating events, followed by equal duration events, then decelerating events
((< +) (> +) or)	An accelerando **or** a rallentendo
(((< +) (> +) or) ((= *) (< +) (> +) or))	An accelerando or a rallentendo, immediately followed by an (eventually empty) series of equal duration events, immediately followed by an accelerando or a rallentendo

segmentation->peaks max

Performs segmentation using the duration-maxima method. This method searches for maxima of durations (that appear as peaks in the **BPF** view, see above). Peaks of durations result in agogic accents and thus are often convenients places for segmentation. Parameters to be set appear in a dialog box:



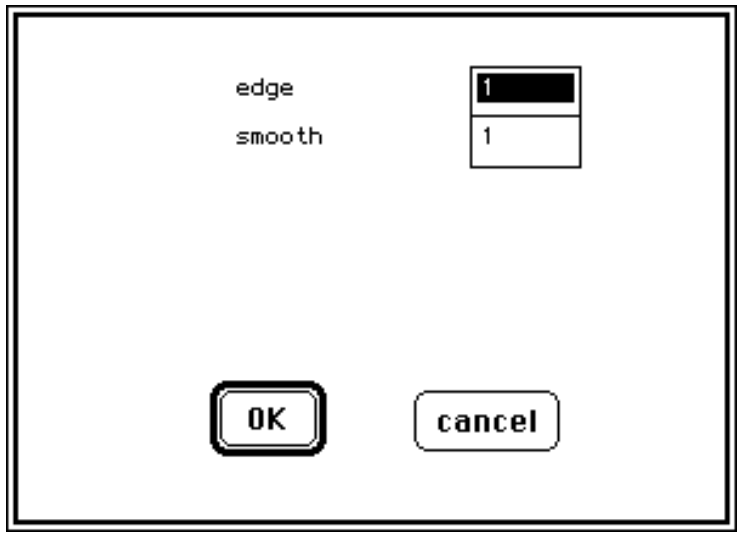
edge — A value of 1 will align the articulation point on the left of the peak duration interval. A value of 2 will align the articulation point on the right of the peak duration interval.

smooth — Smoothing factor (between 1 and 20). Giving a smoothing factor higher than 1 will iterate the peak extraction mechanism, that is, take the peaks of the peaks, etc. The practical effect is that Kant will select higher level peaks (i.e. longer durations), and ignore small local, fluctuations.

thresh — Sets a minimum threshold in 1/100th of a second. Values above this threshold cannot be selected as duration peaks. This combines with the smooth parameter to eliminate lower level peaks.

segmentation->peaks min

Performs segmentation using the duration-minima method. Kant searches for minima of durations (i.e. very short notes). The articulation point is not necessarily put on the minimum, but may occur a few events to its left, where the events density tends to suddenly increase. This method is practical for localizing groups of events of greater density, and to align them as either an upbeat or an attack. Parameters to be set appear in a dialog box:

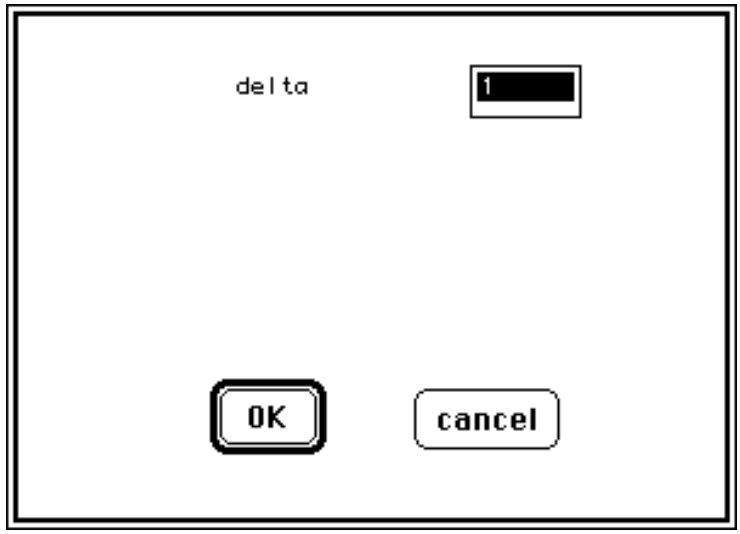


edge A value of 1 will align the articulation point on the left of the selected event. A value of 2 will align the articulation point on the right of of the selected event.

smooth Smoothing factor (between 1 and 20). Giving a smoothing factor higher than 1 will iterate the peak extraction mechanism, that is, take the peaks of the peaks, etc. The practical effect is that Kant will select higher level peaks (i.e. shorter durations, or greater density).

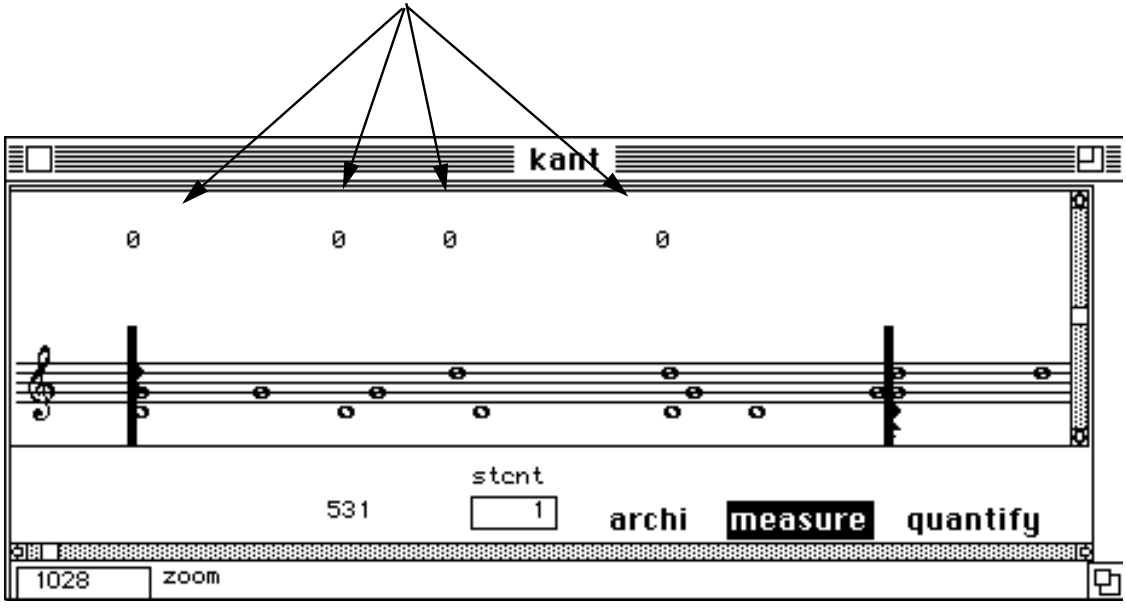
segmentation->chords

Performs segmentation using the quasi-synchronism method. Kant try and localizes events which onsets are very close, thus likely to be quantized on the same onset time (forming chords or grace notes). Parameters to be set appear in a dialog box:

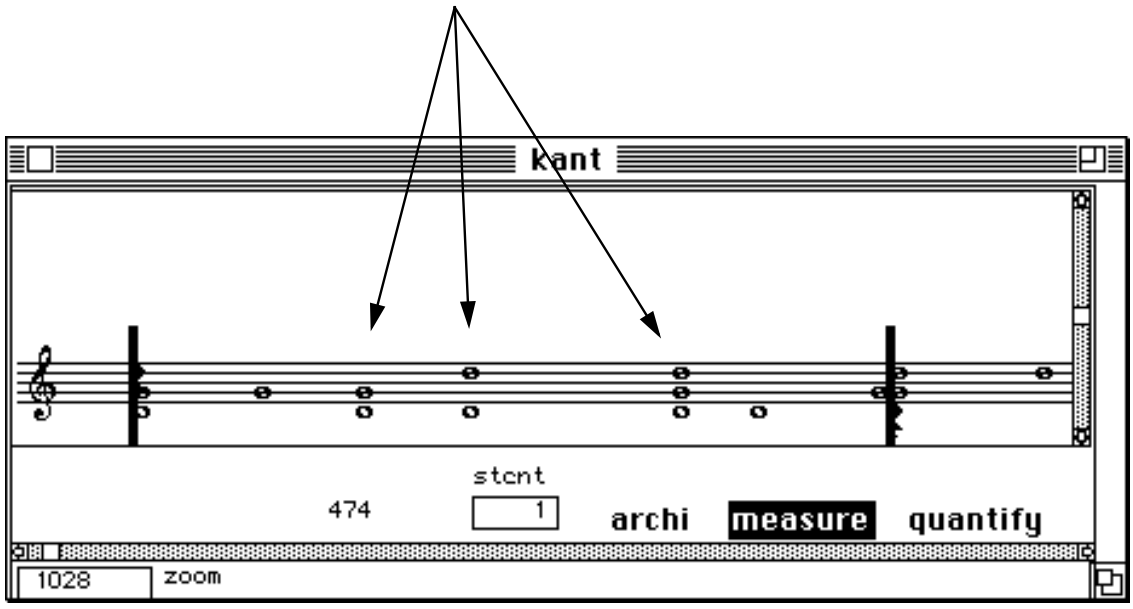


delta Sets the range, in 1/100th of a second, within which the difference between 2 onset times will be considered as equivalent to 0.

articulation points put by the "chords" method, with delta = 10

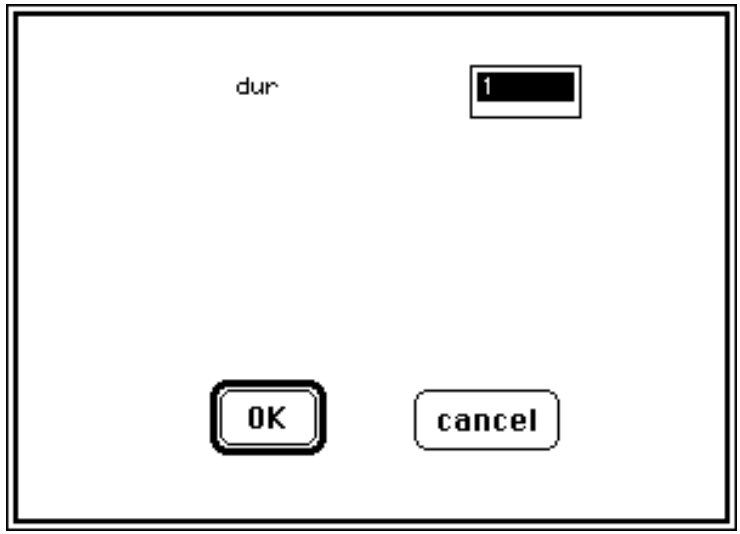


chord alignment performed by the command '='



segmentation->rests

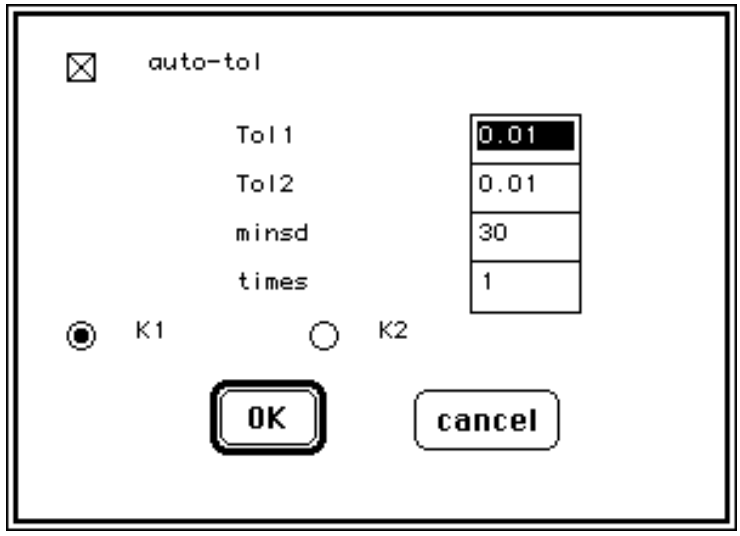
Performs segmentation at the boundaries of rests. Parameters to be set appear in a dialog box:



dur Sets the minimum duration in 1/100th of a second of the rests to be recognized.

go

Once the articulation points have been inserted, run the command go in order to validate the measure segmentation. The following dialog will appear.



auto-tol — You will usually let the auto-tol button checked in order to gain advantage from the automatic tempo search algorithm. Otherwise you have figure out values for the two next parameter.

Tol 1 — Advanced users. See the ICMC article on Kant[1].

Tol 2 — Advanced users. See the ICMC article on Kant.

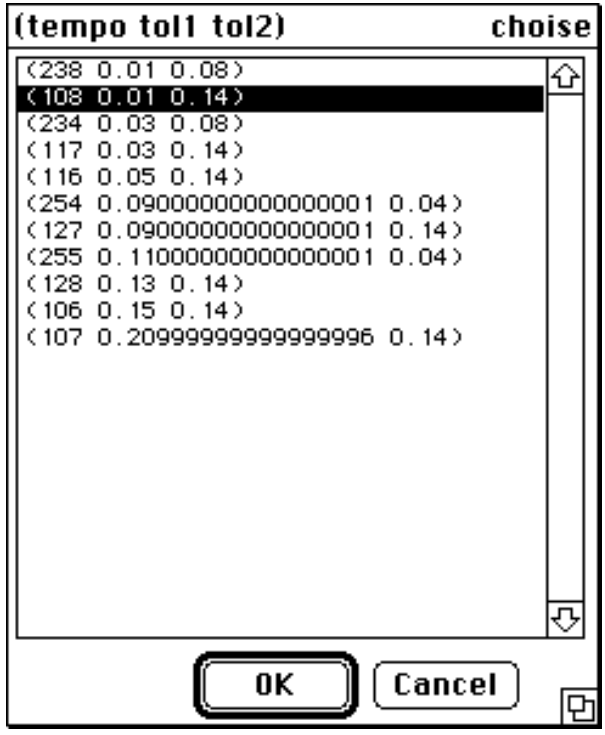
minsd — Minimum duration of a segment in 1/100th of a second. If a segment is shorter than that it will be aggregated to the next one.

times — A factor by which the proposed tempi will be multiplied. If for example Kant proposes tempi ranging from 160 to 200, and you set times to 0.5, then the tempi will range from 80 to 160. The distribution of events inside the segments will not change, only the number of beat falling in every segment. If times is greater than 1, the number of beats will be increased. If times is smaller than 1, the number of beats will be decreased. The second case might be dangerous: a 3 beats bar, with a time factor of 0.5 will be rounded to a 1 beat bar (instead of 1.5) resulting in a significant distorsion of the durations inside that bar with respect to the other bars.

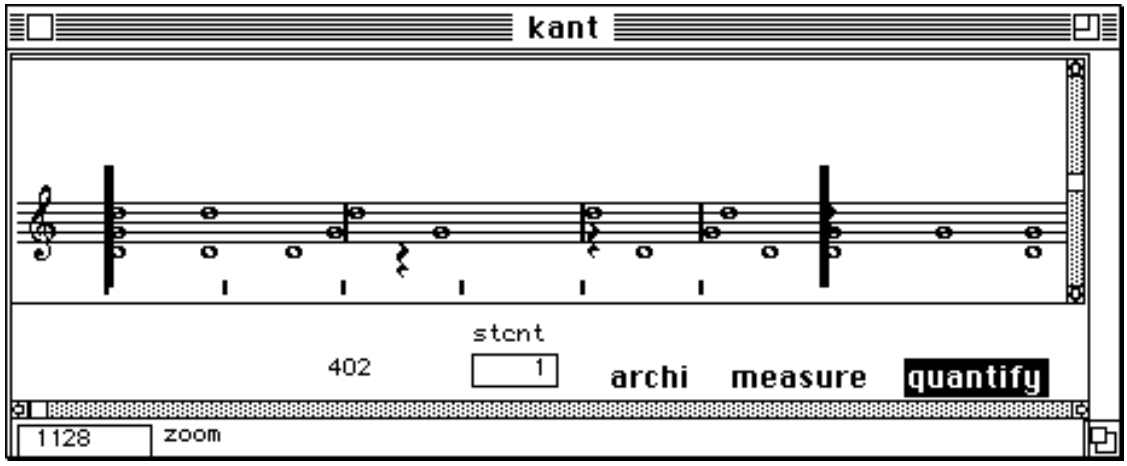
1. Agon C. , G. Assayag, J. Fineberg, C. Rueda. "Kant: a Critique of Pure Quantification", *Proc. of the ICMC*, 1994.

K1, K2 If the K1 button is selected (normal case) Kant will keep the quarter note as the reference for the expression of tempo and allow only measures with the quarter note as the beat (e.g. 3/4). If the K2 button is selected, then Kant will allow the insertion of measures where the beat is the eighth note (e.g. 5/8) wherever it finds that it increases the quality of tempo recognition.

Once you have clicked *OK*, Kant will search for the tempo. Then it will display a tempo list. In each line, the proposed tempo is given, followed by two values of tolerance (*Tol 1* and *Tol 2*) that were used to find it. The smaller these values are, the more precise the recognition is. Choose the tempo that you prefer then click *OK* (or double click on the tempo).



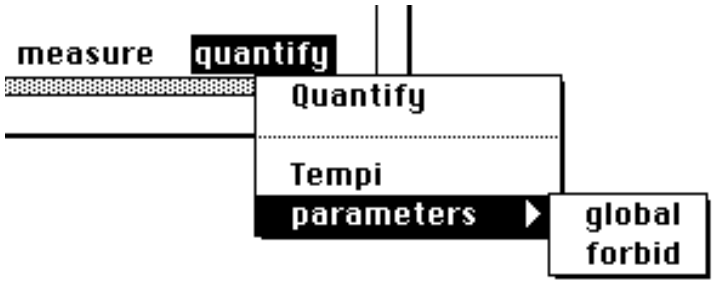
Kant will then display the actual measures by putting bar lines on the staff in the notation view. The beat structure will appear as well underneath the staff line. The menu title **quantify** will become highlighted, telling you it's time now to get to the third step: actual quantification and transcription.



You still have the opportunity to get back to second step by simply clicking on the menu title **measure** or to the first step by simply clicking on the menu title **archi**.

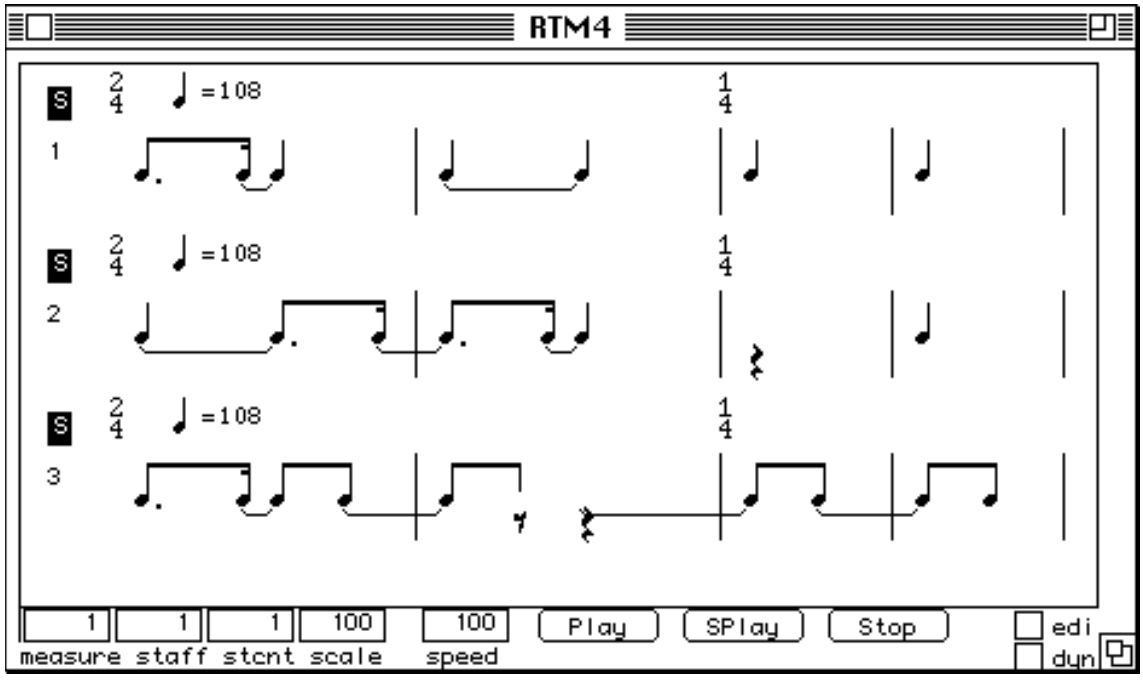
5 Step 3. Performing beat quantification and final transcription

Commands in the menu "quantify"



Quantify

Runs the beat quantification and transcription process. The durations have been processed in order to be aligned with the archi-measure/measure structure; then the durations, along with the pitches and all the meter/tempo information computed in the previous steps are passed to the PatchWork **Quantify** module (see the PatchWork Reference Manual). When the calculation is done Kant displays a **poly-rtm** window with the transcription in musical notation.

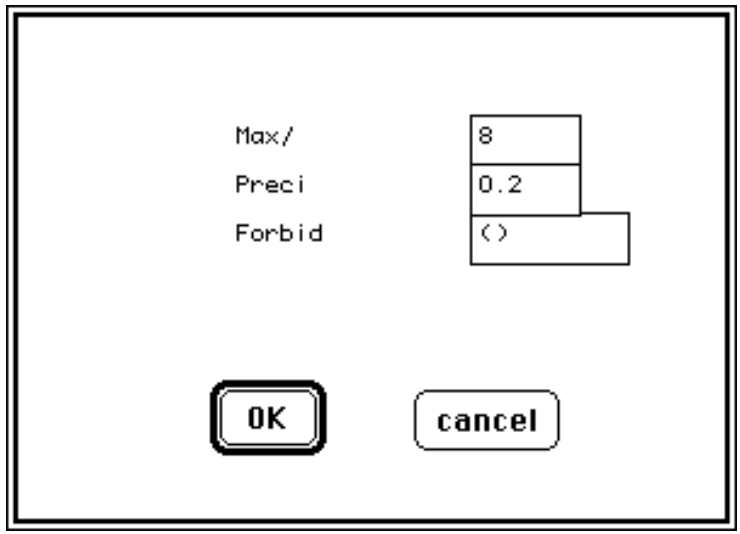


Tempi

This is where you can change (slightly) the tempo (not by more than +5 or -5 units). The durations will be scaled accordingly. This changes only the tempo marked in the final result, leaving the notation untouched. This is useful for aligning the tempos of various archimeasures into rational relationships.

parameters->global

Displays a dialog for tuning the global quantification parameters.



Max/ maximum subdivision of the beat allowed. For a quarter note beat, a value of 9 (sic) means 32nd notes.

preci precision flag, a value between 0.01 and 1. A value near 1 means you want to favour precision in the choice of beat subdivision. A value near 0.01 means you want to favour simplicity (this can sometimes result in the insertion of grace notes as higher subdivisions — e. g. quintuplet, septuplet — will tend to be eliminated, while "simple" subdivisions — e.g. triplet, sixteenth note — will tend to be chosen).

forbid A list of constraints that control the choice of beat subdivision. For example, (5 7 9) will forbid quintuplets, septuplets and 9-uplets. (! 5) will force quintuplet.

parameters->forbid

The **forbid** item in the **parameter** submenu lets you specify the beat subdivision constraints independently for each measure (and even for each single beat). After selecting this item, you will see a small "()" sign above each measure. Click on that sign. Then Kant will display a dialog box where you'll enter the constraint list for that measure. Here are some examples:

(5 7) forbid 5 and 7 for the measure.

((5 7) (! 5)) forbid 5 and 7 for the 1st beat. Force 5 for the rest of the measure.

((5 7) (! 5) ()) forbid 5 and 7 for the 1st beat. Force 5 for the second beat. No constraints for the rest of the measure.

6 Working on several archi-measures at a time

At the beginning of step 2, instead of selecting only one archi measure, it is possible to select all the archi measures at once by command-clicking on any vertical line. If you do so, then for all subsequent operations, Kant will issue as many instances of parameter dialog boxes as there are archi measures in the sequence. You will then have the opportunity to accept (OK) or cancel (CANCEL). Operations will take place only on accepted archi measure. Before the final step (transcription) the accepted archi measures will be concatenated.

This mode is interesting if you know exactly what you're doing because you can then get the whole sequence transcribed at once in a **poly-rtm** window. But it can be very confusing on long sequences where you will probably prefer to work one archi-measure at a time.

References

Agon C. , G. Assayag, J. Fineberg, C. Rueda. "Kant: a Critique of Pure Quantification", *Proceedings of the ICMC*, International Computer Music Association, San Francisco, 1994.

Agon C. *KANT: une critique de la quantification pure*. Mémoire de DEA, Ircam, Université Paris XI, 1994.

Assayag G., C. Rueda. "The Music Representation Project at Ircam", *Proceedings of the ICMC*, International Computer Music Association, San Francisco, 1993.

Brown J.C. "Determination of the meter of musical score by autocorrelation", *J. Acoust. Soc. Am.* 94, 1993, pp. 1953-1957.

Malt, M. *Patchwork — Reference Manual*, Ircam, 1993.

Vérière V. *KANT: quantification rythmique structurée*. Mémoire de DESS, Ircam, Université Paris VI, 1993.

Desain P., H. Honing. "The Quantization Problem: Traditional and Connectionist Approaches", *in* M. Balaban, K. Ebcioglu & O. Laske (Eds) *Understanding Music with AI: Perspectives on Music Cognition*, The AAAI Press, 1992, pp. 448-463.

Index of Symbols and Key Strokes

Index

R

S

T

V

Z