Archives of Control Sciences
Volume 12(XLVIII), 2002
No. 4, *pages 433–455*

Granular computing as an abstraction of data aggregation — a view on optical music recognition

WŁADYSŁAW HOMENDA

In the paper optical music recognition (OMR) is considered as an example of paper-to-computer-memory data flow. This specific area of interest forces specific methods to be applied in data processing, but in principle, gives a perspective on the merit of the subject of data aggregation. The process of paper-to-computer-memory music data flow is presented from the perspective of the process of acquiring information from plain low-level data. The discussion outlines an interpretation of this process as a metaphor of granular computing. The stages of data aggregation and data abstraction are shown as steps leading to the formation of knowledge granules and to recovering dependencies between knowledge granules and between the information included in knowledge granules. An influence of the granular world of music notation on the design of a computer program is presented. The presentation is related to a real computer program of music notation recognition and music knowledge representation and processing. The relationship between the granular structure of music knowledge and user interface of the program is outlined.

Key words: data aggregation, data abstraction, granular computing, information granules, knowledge representation, music notation, music recognition, music representation, user interface design

1. Introduction

The automation of data flow between a paper document and a computer memory is one of the most important tasks in practice of the computerized world of human activity. Though the memory-to-paper direction of this flow is obviously easily solved in most of its merit domains, both aspects - theoretical and practical - of the opposite direction are still a challenge in the society of computer experts. Contemporary scanners and CCD cameras create a high quality electronic image - a plain raster bitmap - of the paper document, but in principle this image obviously does not differ from its origin except in the media of manifestation, i.e. monitor screen rather than paper. Such a basin of scanned primitive data has to be processed in order to gain information, or - perhaps better expressed as - knowledge, brought by this almost endless sequence of numeric bit/byte data.

The Author is with Faculty of Mathematics and Information Science, Warsaw University of Technology, 00-661 Warsaw, Poland.

Received 2.11.2001, revised 13.04.2002.

This primitive data resource and the basic operations on it can be considered as a numeric computation rather than a more structural granular computation, cf. [2,13]. The low-level digitized data has to be converted to more structural pieces of information. Basic methods of image data analysis are independent of image contents. Run lengths, rotations, projections, histograms, moments, filtering are often applied to get properties of an image or its part. Though these basic methods of image processing are universally applied to low-level raster data, utilization of given method must be - of course - carefully analyzed.

Acquired knowledge has to be represented and stored in a format "understandable" by the computer brain, i.e. by a computer program (nb. computer program is a product of human activity) - this is a fundamental observation and it will be exclusively exploited as a subject of discussion in the paper. Of course, a computer program cannot work without low level support - it uses a processor, memory, peripherals, etc., but they are nothing more than only primitive electronic tools and so they are not interesting from our point of view. Processing of such an acquired image of the paper document is a clue to the paper-to-memory data transfer and it is successfully solved for selected tasks, c.f. OCR technology. However, documents that are more complicated structurally than linear (printed) texts raise the problem of data aggregation in order to form information granules as well as the problem of acquisition of implicit information/knowledge that could be concluded from the relationships between the information granules. Documents containing graphics, maps, technical drawings, music notation, mathematical formulas, etc. can illustrate these aspects of difficulties of paper-to-computer-memory data flow, they are research subjects and still raise a challenge for software producers.

In the paper optical music recognition (OMR) is considered as an example of paper-to-computer-memory data flow. This specific area of interest forces specific methods applied in data processing, but in principle, gives a perspective on the merit of the subject of data aggregation. Data flow starts from a raster image of music notation and ends with an electronic format representing the information expressed by a scanned document, i.e. music notation in our case. Several stages of data mining and data aggregation convert the chaotic ocean of raster data into shells of structured information that, in effect, transfer structured data into its abstraction - music knowledge. This process is firmly based on the nature of music notation and music knowledge. The global structure of music notation has to be acquired and the local information fitting this global structure must also be recovered from low level data. The recognition process identifies structural entities like staves, group them into higher level objects like systems, than it links staves of sequential systems creating instrumental parts. Music notation symbols very rarely exist as standalone objects. They almost exclusively belong to structural entities: staves, systems, parts, etc. So that the mined symbols are poured into these prepared containers - structural objects, cf. [3,12,17]. Music notation is a two dimensional language in which the importance of the geometrical and logical relationships between its symbols may be compared to the importance of the symbols alone. This phenomenon requires that the process of music knowledge acquisition must also be aimed at recovering the implicit information represented by the geometrical and logical relationships between the symbols and then at storing the recovered implicit relationships in an appropriate format of knowledge representation.

There are open problems of information gaining and representation like, for instance, performance style, timbre, tone-coloring, feeling, etc. These kinds of information are neither supported by music notation, not could be derived in a reasoning process. Such kinds of information are more subjectively perceived rather than objectively described. The problem of definition, representation and processing of "subjective kinds of information" seems to be very interesting from research and practical perspectives. Similarly, problems like, for instance, human way of reading of music notation may be important from the point of view of music score processing, cf. [10,11]. Nevertheless, processing of such kinds of information does not fit framework of the paper and is not considered.

The process of paper-to-computer-memory music data flow is presented from the perspective of a paradigm of granular computing, cf. [19]. The low-level digitized data is an example of numeric data representation, operations on low-level data are numeric computing oriented. The transforming of a raster bitmap into run lengths of black and while pixels is obviously a kind of numeric computing. It transfers data from its basic form to more compressed data. However, the next levels of the data aggregation hierarchy, i.e. finding the handles of horizontal lines, begins the process of data concentration that become embryonic knowledge units rather than more compressed data entities, cf. [13].

The discussion is focused on the interpretation of this process as a metaphor of granular computing. The stages of data concentration and aggregation and data abstraction are shown in sections 2 and 3. These stages are interpreted as steps leading to the formation of knowledge granules that constitute a space of music knowledge. The space of music knowledge is not a simple collection of granules but rather a sophisticated structure of granules and relationships between them. Recovering the relationships between granules is considered in section 4.

Since an investigation of the subject is aimed at a computer system of music processing that - obviously - has to be useful for its user, so that user interface must be an inherent part. In section 5 the influence of granular world of music notation on user interface and on the communication of the system with the outside world is presented. The relationships between granular structure of music knowledge and user interface of the program are outlined.

2. Data aggregation

The development of an understanding of complex systems is a process of creating mental abstractions that make it possible to ignore irrelevant detail and concentrate on the essential features of systems. There are two main approaches to data analysis - agglomerative and partitive; corresponding to bottom-up and top-down strategies, cf. [2]. In real applications a mixture of both strategies is utilized. This is also the case with

OMR. However, the very first steps in image analysis are of the agglomerative type due to the nature of the initial data, cf. [5,9,13].

The input data in paper-to-computer-memory data flow can be seen as a flat ocean of primitive data - a bitmapped image of a scanned document. Image data can be represented in different formats such as grayscale bitmap, monochrome bitmap, looseless compressed bitmap, etc., but from our perspective this issue is of little importance. Bitmap - monochrome or grayscale - can be seen as data with as many dimensions as the number of pixels. Analysis of such a data resource is, in practice, impossible due to its huge cardinality and dimensionality. So then the aggregation of basic data is necessary to get more reasonably transformed information.

The main assumption is that the investigation is focused on processing basic image data, which is a raster monochrome bitmap. Since data acquired from scanners or cameras are often grayscale or color bitmap, the first step to be done on such original data is a conversion of the grayscale (or color) image to monochrome data. This first operation is mostly done by a kind of thresholding. Gray scale (color) to monochrome conversion is a kind of preprocessing on the input data often done by scanner software and is rarely considered in recognition systems. Gray scale (color) to monochrome data conversion significantly reduces the cardinality of image data by restricting the color of every pixel to one of two monochrome values. However, the number of pixels remains unchanged, so that the dimensionality of the data also remains unchanged. Of course, the direct processing of a basic bitmap that would lead to high level data extraction, e.g. musical symbol recognition, is impossible due to the big complexity of any algorithm working on data of huge dimensionality. Thus, dimensionality and cardinality reduction is still necessary. This reduction is presented here as data concentration and data aggregation stages.

Data concentration is understood as a conversion of the whole image bitmap or - usually - of a local area of raster bitmap into more abstract information units. Data concentration always significantly reduces cardinality and dimensionality of the data it operates on. Such operations as projections, histograms, run lengths, filtering, derivative analysis of projections, gaining vertical and horizontal placement of horizontal line handles and vertical sticks, respectively, are typical data concentration operations.

It is worth mentioning that the other forms of preprocessing - that are sometimes used to remove noise or distortion, to correct skewing, etc. - do not significantly change the size of the data either so that they are not considered as data concentration operations.

Data aggregation is meant here as the integration of concentrated data into information units that support entities of geometrical information. Data concentration is the first step in the process of data aggregation, the data aggregation step is the next step in the process of data aggregation. Having low-level data concentrated into abstract information quantities (as, for example, vertical locations of horizontal lines and horizontal locations of vertical sticks), these abstract information quantities are joined together into more general units of aggregated data such as the position on the page of horizontal lines, of vertical sticks, etc.

2.1. Horizontal projections and horizontal lines' handles

The usual techniques applied in primary data concentration are run lengths, (sections of white/black pixels of a given line of a raster image), projections, histograms, filtering, derivative analysis of projections. These methods allow the catching of the hooks of areas where music notation symbols are placed. Namely, the local area of a raster bitmap is converted into more abstract information units. Information about a horizontal line sections and vertical sticks, about their vertical and horizontal placement, respectively, are examples of such units.

In this section horizontal/vertical projections are discussed as a tool leading to data concentration. Several musical images with associated horizontal or vertical projections display are presented in the paper. For horizontal projection, vertical axis is always related to rows of original image while horizontal axis is related to projection value for respective rows. Left part of a row is a bin containing black pixels of respective row while right part is a bin of while pixels. The meaning of vertical projection is similar. Note: only monochrome images are considered, so then no information about level of brightness or colors is available.

Digital filtering allows for the recovery of basic information from projection results. For instance, the minimal and maximal numbers of pixels in a projection lines and average value of pixels of all projection lines allow for computing a noise parameter. High-pass filtering with noise parameter applied, i.e. considering only these projection lines which give number of black pixels greater than noise parameter, allows for the recovery of basic information of long horizontal/vertical line placement in the respective region. Technical details of digital filtering as - for instance the method of noise parameter computing - are not discussed since they are not important from the main point of the paper.

Further processing of filtered information will allow for the acquisition of information about a line's properties, its precise location, relationship with other symbols, etc.

Staff lines, also called stave, are the most noticeable object of music notation and they are localized at the very first step of the recognition process. Algorithms of separate staff lines location are typically based on horizontal projections. The localization of separate staff lines is usually preceded by finding the horizontal line handles. Theoretically, for non-distorted and non-skewed images, the methods based on horizontal projections should be fully effective: high pass filtering gives a clear image of a stave as five equidistant picks. The height of these picks would be equal to the length of the (non-distorted) stave.

In real images staff lines are too much distorted to give such clear projections, especially when horizontal projection is done for the page width region or even for any wide region. The scanned image of a sheet of music is often skewed, staff line thickness differs for different lines and different parts of the stave, staff lines are not equidistant and are often curved, especially in both ending sections of the stave, staves may have different sizes for an ensemble type of score, etc. These problems mean that projections done in a wide region are often useless for stave location purposes. For these reasons several steps are necessary to recover full information about stave localization.

In Figure 1 a system of scanned piano music notation is presented. The displayed horizontal projection of the middle measure allows for obvious and easy localization of the vertical placement of long horizontal lines. Though it is of quite good quality, data concentration for larger regions becomes affected by the original image defects or paper notation distortions: staff lines that are as long as the image width give picks of different length, not equidistant, etc. These defects are observable for 1/3 width projection of this good quality image. These defects will obviously accumulate when wider region is taken for projection.

In Figure 2 an example of a skewed image is presented. In this case the projection of the width of one measure is almost useless for finding the vertical placement of horizontal lines. If the projection region is narrower, horizontal projection is much more useful in gaining information about the existence of staff lines and about their vertical location. On the other hand, projections performed in a narrow region are distorted by notation symbols such as ledger lines, dynamic ’hairpin’ markings, slurs, note heads, etc. In Figure 2 the horizontal projection is affected by beams and dynamic hairpins.

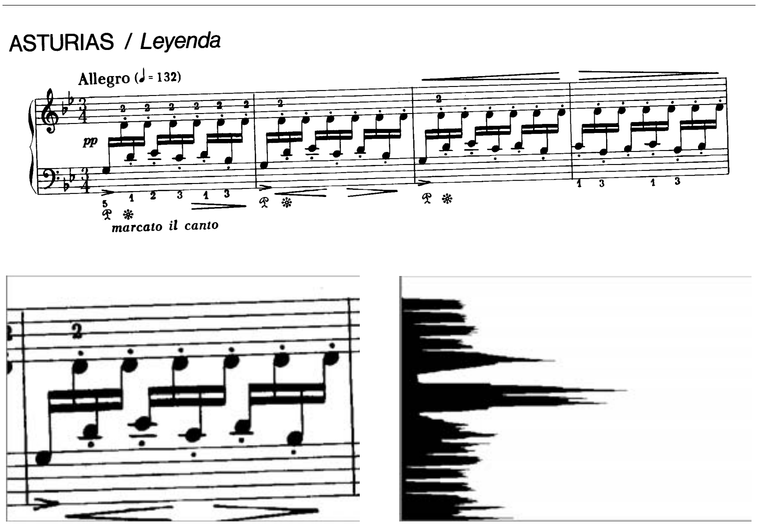
Thus, simple filtering does not give information sufficient for staff line handles location. Therefore further analysis is necessary in the process of data concentration such as, for instance, the analysis of horizontal projections in consecutive narrow regions.

In conclusion - data concentration is a process restricted to the local area of low-level data. The locality of this process depends on the nature of explored data as well as on the features of the explored data.

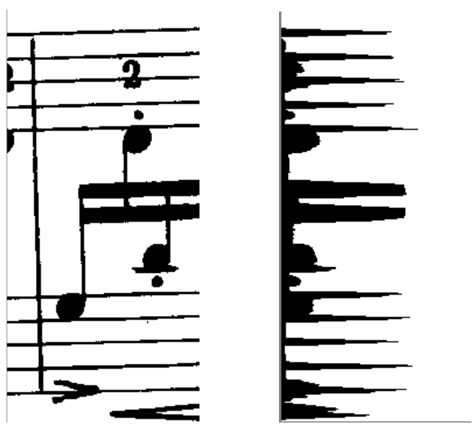
Figure 1. Data concentration - horizontal projection for high quality image

2.2. Vertical projections and vertical sticks

In section 2.1 the method of data concentration based on horizontal projections and high-pass filtering is presented. In this section vertical projections are discussed as a tool leading to data concentration. The projections are explored in restricted regions, such



Horizontal projection of wide region of a skewed image



Horizontal projection of narrow region of a skewed image

Figure 2. Data concentration - horizontal projection for skewed image

as - for instance - a measure, a stave, the space between staves. High-pass filtering as a method of detecting vertical lines due to the fact that vertical lines are of irregular length and are rather short. There are a few exceptions of this rule such as barlines and brackets. In the cases of these exception high-pass filtering is a powerful tool. However, in general high-pass filtering is of less importance for vertical projections than for horizontal projections. Especially, high-pass filtering of vertical projections is not a useful tool in acquiring information about the horizontal placement of vertical sticks. Thus the

Figure 3. Data concentration - vertical projection for stave symbols

next step - after obtaining vertical projections of selected regions - is mainly based on the analysis of a derivative of the vertical projection, c.f. [9]. Analysis of the derivative of vertical projection gives the horizontal placement of vertical sticks. Information acquired by derivative analysis is a step in data concentration, c.f. Figure 3 and 4. This is the first approximation of object location.

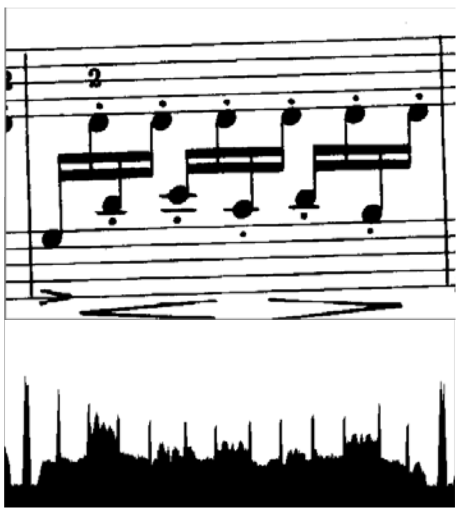


Figure 4. Data concentration - vertical projection for inter-staves symbols

2.3. More methods

The aim of extensive use of horizontal and vertical projections - that are described in previous sections - is to outline the idea of data aggregation stages. Horizontal and vertical projections are very useful and illustrative tools that could be successfully used in discussion on data aggregation process. Projections are also basic and very efficient techniques applied in real OMR systems. However, obviously, data aggregation process is not restricted only to projections. Projections may be of less efficiency or even may be useless in context of symbols that are not horizontal or vertical lines themselves or if they do not have such lines as part of them. For instance, any one of these perpendicular projections is effective when applied to sloped lines like beams or dynamic hearpins, when applied to clef symbols or to noteheads, etc. So then, despite that projections are very efficient methods used in data aggregation process, they have to be supplemented by variety of other methods. It has to be underlined that there is no universal technique that could be successfully applied in data aggregation process. Different methods have to be used in collaborative way in order to gain high level of recognition. Examples of such methods are listed here.

Sloped projection, i.e. projection done in direction different than horizontal or vertical, may be successfully applied in aggregation process to primitive data that represents sloped lines. This technique is a simple variation of perpendicular projections, but it is distinguished here because its application must be associated with a method of projection angle calculation.

Transformation to height/width space (height/weight of the bounding box of given symbol) is - for a wide variety of symbols - a very simple technique that allows for restriction of a spectrum of applicable symbols in classification process in amazingly effective way. In some cases, when utilized with other methods, e.g. if syntactical rules put their restrictions on the set of admitted symbols, height/width space analysis allows for final symbol classification.

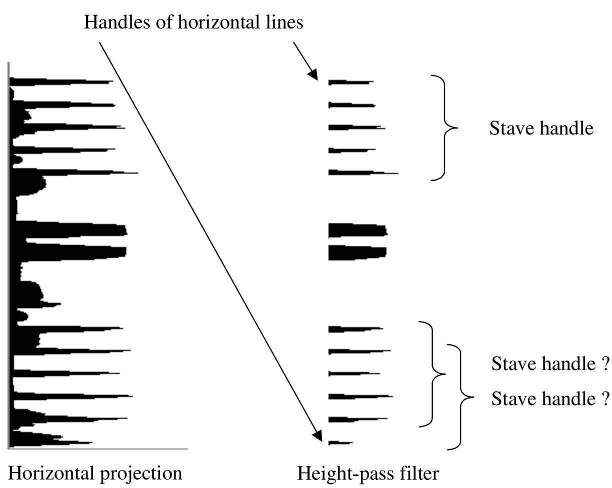


Figure 5. Data concentration - finding line handles and stave handles

Syntactical methods play significant role in OMR in general, cf. [1,4,7,20] and in data aggregation process specifically. Syntactical methods could be applied at any stage of OMR, so then discussion of this section is also valid in context of data abstraction.

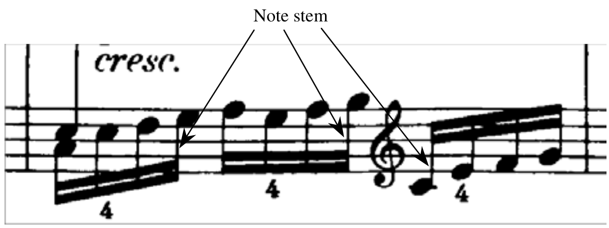


Figure 6. Data aggregation - localization of staves, barlines, vertical sticks

Music and music notation have been changing and developing since it creation and it is reasonable to expect that development process of music and music notation is endless. These observations make clear that formal description of neither music, nor music notation is possible. This is why any attempt to global formalization of music notation, e.g. trial to provide its full syntactical description in a meta-language of formal grammars or any other tool, will lead to failure at current stage of our knowledge of formalization methods. However, syntactical description of local areas of music notation can be efficiently applied. Syntactical techniques are usually based on rules utilized by music notation publishers, cf. [21,23].

Methods based on mathematical theories have their contribution in image analysis, patters recognition and can be successfully applied to music notation recognition. For instance, transformation to moments' space is an efficient method that can be applied to compact symbols of sophisticated shape, e.g. clef symbols and rests. Transformation to moments' space is especially interesting when translation and rotation invariant moments are applied. Statistical estimators and uncertain information reasoning can be efficiently applied in symbol classification. Cf. [8,14,15,16]

Edge detection, contour tracing and shape analysis are efficient methods for long curvature symbols as, for instance, braces/brackets placed vertically in front of a system to define groups of instruments, slurs and ties, arpeggios, etc. Shape analysis is often associated with an approximation of edges of a symbol by mathematical functions. In such a case primitive data are transformed to the space of coefficients of the curves approximating the symbol. Of course, such coefficients are computed with some degree of uncertainty, so then comparison of coefficients of different symbols must include uncertain information processing.

2.4. Data aggregation

Data aggregation is understood here as the integration of concentrated data into information that supports a unit of geometrical information such as, for example, the position on the page of staff lines, of vertical sticks, etc - cf. Figure 6. Data concentration investigated in the two previous sections is the beginning step in the process of data aggregation. Low-level data supported by a raster bitmap was concentrated on information quantities - the vertical locations of horizontal lines and horizontal locations of vertical

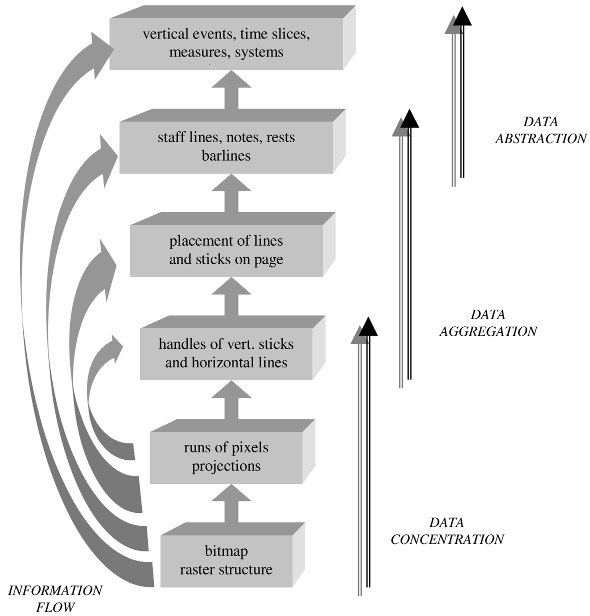


Figure 7. Data concentration, aggregation and abstraction granularity of the process of music notation recognition

sticks. These basic information quantities are concentrated into more general units in the next step of the data aggregation process, i.e. in the step of data aggregation.

Having information about the vertical placement of horizontal lines, further data processing will lead to grouping a five horizontal lines that are equidistant into the handle of stave, cf. Figure 5. It is worth noticing that in the case of a non-distorted image the five staff lines are equidistant. In practice the five staff lines are approximately equidistant and the distance between the staff lines in different staves may differ, i.e. staves of a page of music notation are often of different height. The detailed solution of the problem of linking horizontal line handles to stave handles is of lesser importance from the perspective of this paper and is omitted here.

Once stave handles are recovered from the basic data, which means that the vertical placement of the staves is located, the horizontal extensions of the stave handles are found and, finally, the endings of the staves are detected.

This is a further concentration of primitive low-level data as well as the aggregation of data concentrated in the previous step. Projections in relatively narrow regions used in the process of finding stave extensions and the location of stave endings is a further low-level data concentration. On the other hand, the process of linking horizontal line handles into stave handles is an instance of aggregation of concentrated data, i.e. it is the next step in the data aggregation multi-stage process. Thus, a strict separation of the consecutive steps in the data concentration process is impossible - these steps enable each other.

The observation of the enabling steps of data concentration outlines a universal rule - the exploration of sophisticated space data is not as simple as the application of separated tasks to consecutively concentrated and aggregated entities of data. This rule can be even observed at the very basic level of data concentration. The investigation of vertical projections and finding vertical sticks is a process that can benefit from a low-level raster bitmap as well as from the data concentrated in the process of horizontal projection analysis.

This process can also benefit from the further steps of data aggregation process such as the information about staves location because symbols of music notation are localized on staves, their neighborhood, in spaces between staves, c.f. Figure 3 and 4.

3. Data abstraction

The data concentration and data aggregation considered in section 3 is a process of acquiring geometrical data that is based on low-level raster data. Data concentration and data aggregation create information quantities that are directly supported by a low-level raster bitmap. Horizontal lines and vertical sticks are examples of information quantities that can be extracted from a raster bitmap as a collection of black pixels distinguished from their neighborhood. Such a collection of pixels usually create a simple geometrical shape.

Data abstraction is a process based on information quantities concentrated / aggregated during previous steps. Data abstraction creates information units that describe concepts that either are not directly supported by a low-level raster bitmap or create complex geometrical shapes. Systems, measures and - to some extent - staves as well as notes, vertical events, time slices are examples of abstract entities of information. Data abstraction is a stage in the data aggregation process of a higher level of the information hierarchy than data abstraction and data aggregation levels. However, the distinction between data aggregation and data abstraction does not have clear boundaries. It can be rather compared with a unit of knowledge emerging from plain ocean of data. Figure 7 outlines the idea of the process of data aggregation and abstraction at the example of the score structure identification and music symbols recognition. The steps of this process form granules of music knowledge structures and music notation recognition discussed in the section 4.

3.1. Staves, measures, systems

Staves, systems, measures are basic concepts of music notation, cf. Figure 8. They define the structure of music notation and are considered as information quantities included into the data abstraction level of knowledge hierarchy. The following observations justify such a qualification.

A stave is an arrangement of parallel horizontal lines which together with the neighborhood are the locale for displaying musical symbols. It is a sort of vessel within a system into which musical symbols can be "poured". Displayed on it are music symbols, text and graphics belonging to one or more parts. Staves, though directly supported by low-level data, i.e. by a collection of black pixels, are complex geometrical shapes that represent units of abstract data. A knowledge unit describing a stave includes such geometrical information as the placement (vertical and horizontal) of its left and right ends, staff lines thickness, the distance between staff lines, skew factor, curvature, etc. Obviously, this is a complex quantity of data.

A system is a set of staves that are played in parallel; in printed music all of these staves are connected by a barline drawn through from one stave to next on their left end. Braces and/or brackets may be drawn in front of all or some of them.

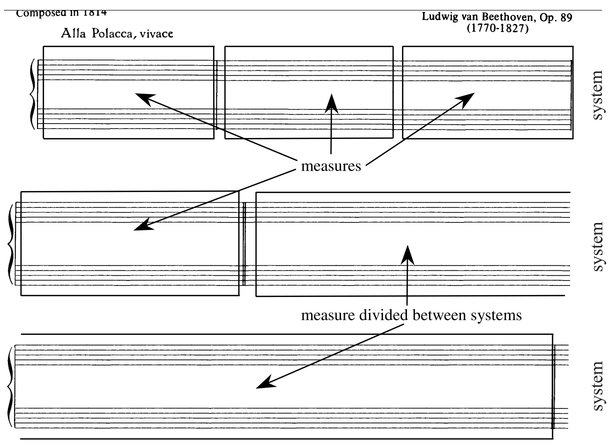


Figure 8. Data abstraction - systems and measures

A measure is usually a part of a system, sometimes a measure covers the whole system or is split between systems, cf. Figure 8. A measure is a unit of music identified by the time signature and rhythmic value of the music symbols of the measure. Thus, like in the above cases, a measure is also a concept of data abstraction level.

3.2. Notes, chords, vertical events, time slices

Such symbols and concepts as notes, chords, vertical events, time slices are basic concepts of music notation, cf. Figure 9. They define the local meaning of music notation and are considered as information quantities included in the data abstraction level of the knowledge hierarchy. Below a description of selected music symbols and collections of symbols are described. Such a collection constitutes a unit of information that has common meaning for musician. These descriptions justify classification of symbols to in the data abstraction level of the knowledge hierarchy.

Note - a symbol of music notation - represents basically the tone of given time, pitch and duration. A note may consist of only a notehead (a whole note) or also have a stem, a notehead and may also have flag(s) or beam(s). The components of a note are

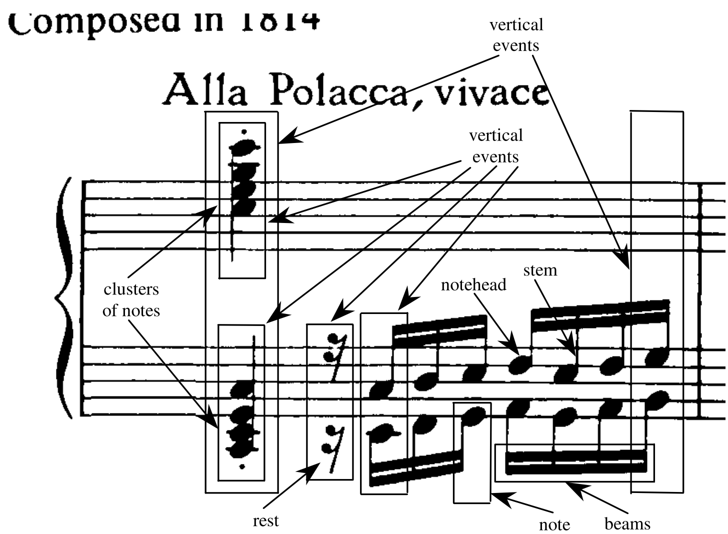


Figure 9. Data abstraction - notes, vertical events, time slices

information quantities created at the data concentration and the data aggregation stages of data aggregation process. This components linked in the concept of a note create an abstract unit of information that is considered as a more complex component of the data abstraction level of the information hierarchy.

A chord is composed of several notes of the same duration with noteheads linked to the same stem (this description does not extend to whole notes due to the absence of a stem for such notes). Thus, a chord is considered as data abstraction.

A vertical event is the notion by which a specific point in time is identified in the system. Musical symbols representing simultaneous events of the same system are logically grouped within the same vertical event. Common vertical events are built of notes and/or rests.

A time slice is the notion by which a specific point in time is identified in the score. A time slice is a concept grouping vertical events of the score specified by a given point in time. Music notation symbols in a music representation file are physically grouped by page and staff, so symbols belonging to a common time slice may be physically separated in the file. In most cases this is time slice is split between separated parts for the scores of part type, i.e. for the scores with parts of each performer separated each of other. Since barline can be seen as a time reference point, time slices can be synchronized based on barline time reference points. This fact allows for localization of recognition timing errors to one measure and might be applied in error checking routine.

4. Granular computing as a metaphor of data aggregation

Staves, systems and measures defined in section 3.1 are understood as pure concepts describing their geometrical properties and relationship only between these concepts. However, a stave can also be seen as a sort of vessel within a system into which musical symbols can be "poured". Displayed on it are music symbols, text and graphics belonging to one or more parts. This meaning of stave is at a higher level of data abstraction than the purely geometrical meaning of a stave - it includes the pure stave as well as all the symbols belonging to that stave. Besides symbols that can be strictly assigned to a stave, music notation includes symbols that cannot be strictly assigned to one stave or are not assigned to any stave - such as barlines, sometimes - beams and notes, etc. Barlines connect all the staves of a system, so they cannot be linked to a specific stave, a note stem can have its upper end in one stave and head in lower stave, beams can connect notes going up and down to both staves of the piano system, cf. Figure 1 and 2.

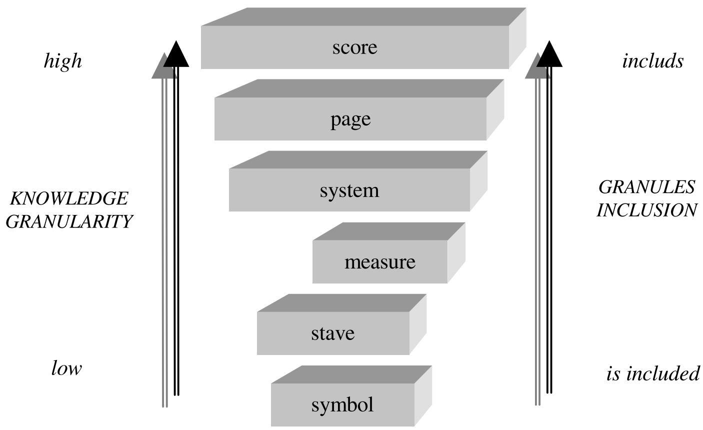


Figure 10. Music notation, the geometrical knowledge granularity

Music notation is a complex system of symbols, structures and concepts tied by the relationships between them. In the section 4 the investigation is focused on the knowledge structure of music notation. Two perspectives are considered: the geometrical hierarchy related to the process of optical music recognition and the logical hierarchy relative to the process of identification of the relationships between the entities of information. Both hierarchies outline the granularity of the process of acquiring music knowledge from music notation.

4.1. Geometrical structure

The geometrical structure of music notation is displayed in Figure 10. This feature of data structure, or better to say: knowledge structure, can be symbolically well expressed as the granular structure of this knowledge and, following the idea expressed in [18], can be outlined as a hierarchical system of granules. Every unit in this hierarchy consists of a collection of units of lower levels. For instance, the score is a collection of pages and - indirectly - includes granules of lower levels; the page is a collection of systems and - indirectly - includes granules of lower levels. This simple inclusion is broken at lower levels: the measure, as a vertical part of the system, is not included in any stave. On the other hand, apart from one-stave systems and measures occupying the whole system, the measure does not include any whole stave. A similar irregularity of granules pyramid is observed in case of music notation symbols.

Most of symbols belong to one stave and are included in one measure, e.g. notes, rests, clefs. However, some of them belong to one stave, but extend beyond the measure horizontally, e.g. slures, ties, dynamic hairpins. Other symbols, which are included in one measure, may extend beyond one stave vertically, e.g. barlines, arpeggios, notes of voices visiting neighbouring stave, cf. Figure 1 and 2.

In this structure of knowledge the page is a unit separated by restrictions of a sheet of paper. In most cases there is no important relation between the page and its systems, so page can be seen as a simple collection of its systems. The exception of the first page, where the composer's name, the title of the piece, etc., are printed, does not break this rule because these items of information are logically connected to the level of the whole score rather than particular page. Despite this, it is convenient to keep this level in the knowledge hierarchy. Although the page can be seen as the simple union of its system, other relations in knowledge structure are not so straightforward and simple. Neither the stave is a simple collection of its symbols nor the system is only a set of staves. Unlike page, the whole score is not only a collection of its systems.

Apart from the parted score, i.e. a score with all parts separated one from another rather than joined into systems, there are relations between systems and between staves, e.g. barlines, repetitions, ties, slurs, dynamic hairpins, etc. Although these relations are examples of a logical structure of a score, they obviously have to be considered as geometrical features.

The importance of relations between symbols of music notation from geometrical point of view comes from their influence on recognition of music symbols. Recognition of music symbols is merely understood as a process that is geometrically based, unlike classification of symbols and context identification of them, which are mostly seen as a process based on logical structure of the score.

It is worth mentioning that geometrical structure of music knowledge granularity is an abstract concept that is not directly supported by data structures included in the program. The relationship between such abstract concepts knowledge and physical data structures representing this knowledge often is very close. However, direct projection of an abstract concept in physical data structures of implemented system is rather rare. Such the direct projection would create serious implementation difficulties. The data structures of the implemented system must reflect the abstract concepts of knowledge as well as technical conditions.

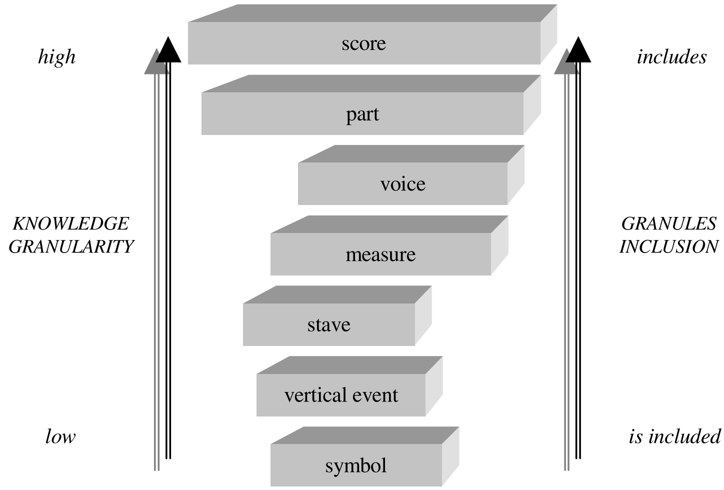


Figure 11. Music notation, the logical knowledge granularity

4.2. Logical structure

The geometrical structure does not include another important entities of knowledge, let us say granules: parts, voices, vertical event, etc. And, what is more important, the interpretation of granules is significantly different than in the geometrical case. The entities of logical structure of music notation are understood as units of data describing symbols and units representing the relations between symbol. In this duality of data relations between symbols are at least as important as symbols themselfs. So, the levels of geometrical and logical pyramids of granules are not the same even if they have the same labels. For instance, the score is a simple collection of pages without attention put to relations between other granules in the meaning of geometrical structures.

And vice versa, the score is more a net of relations between systems, parts and voices etc. when the logical meaning of this term is understood. In this case the page structure has little meaning. The logical pyramid contains granules that have no geometrical meaning. Examples of such granules are: parts, voices, vertical events. This logical pyramid of knowledge granularity is shown in Figure 11.

5. User interface - an abstraction of knowledge granularity

One of the most important factors in music software (and software in general) is its simplicity and the naturalness of its interface. The long learning curve required by a complicated and badly designed interface could be a bane for any computer program. Users have to spend their valuable time learning the software, rather than benefiting from its power. "Designing for the user" is the most important rule in user interface design.

Almost all the problems with modern software user interface design originate from well-intentioned, intelligent and capable people focusing on the wrong things. Technology is the engine that drives user interface design in contrary to the natural direction of human behavior. Instead of technology and implementation tasks, user interface design must focus its gaze on the goals toward which users strive, even if they themselves are sometimes unaware of them - cf. [6,24].

This observation outlines a typical conflict between software developers and software users. Software developers and software users usually have different knowledge, experience, training, expectations, etc. Software developer perception is always twisted by technological factors, technical education and experience, the development tools applied in software implementation, operating system properties, hardware restrictions, etc.

Due to his role in software circulation, the developer is not able to change his point of view to the one represented by the user, who - in most cases - is not an expert in the computer world. Developer thinking often has a tendency to simplify the nature of the implemented subject, to bend the merit of the subject to the available tools, to bend to the restrictions of the available development environment. And, finally, the developer's thinking is also affected by the strengths and weaknesses of the program - he knows them well, even if he is unaware of this fact.

On the other hand, the software user is rarely a computer/software expert. Often he has an entirely different education and experience, the computer is rarely a tool he uses everyday. The user expects program behavior to be consistent with his habits, experience, knowledge. This expectation can be satisfied when user interface is focused on the nature of the implemented issue. He will be unable to learn an interface focused on technology and implementation issues, but it would be easier to learn something that is of the nature of the implemented problem.

A user's thinking is almost always closed within the merits of his problem, his operations almost exclusively consist of creating, editing and updating of the abstract data of the problem he works on. And if the interface provides tools based on the merits of the given problem, on its information units, the user has the opportunity to understand and easily learn the program and its interface. So then user interface should be driven by the data abstraction of the problem considered.

An example of a data driven interface is shown in Figure 12. The Begin Recognition dialog box of Smart Score music software, cf. [22] provides several useful tools applicable in the music notation recognition process. Among them the mastering of all the pages of a score is shown. Two types of music notation can be distinguished: the conductor score, i.e. score with all the parts (part means a notation describing the music of one instrument or performed by one musician) linked together into systems, and a part score, i.e. a score with parts performed by different instruments separated from each other. So, music scanning software should either automatically distinguish between these two types, or its interface should provide an easy tool to define the type of score. Since confidence in the automatic distinguishing of a score type is rather low and - on

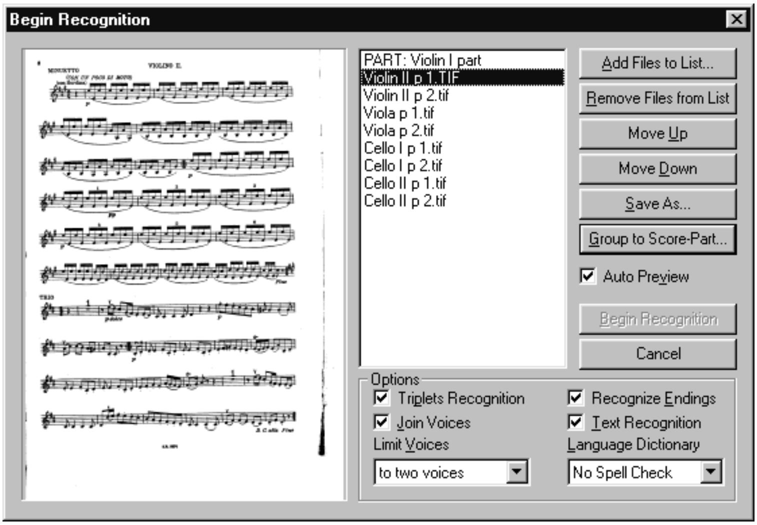


Figure 12. Music notation recognition - creating score parts

the other hand - this feature could easily be defined by the user, the presented software relies on direct user definition of the score type.

For a part type score all the pages (scanned images of paper pages) should be listed part by part, then the pages of every part should be grouped in a single part. In Figure 12 a tool used for the grouping of pages of a piece of music for a string quintet is presented. The part of Violin I has been grouped in one part. Highlighting both pages of Violin II in the list of file names and pressing Group to Score-Part will create the Violin II part of the score, and so on.

Referring to knowledge granularity pyramids outlined in Figures 10 and 11, a collection of the pages of the score (geometrical hierarchy) is linked to a unit that creates an entity of the logical hierarchy - a score part. Of course, linking a list of pages into a part of a given instrument is an operation that is obvious for a musician, a software user. Well-designed data structures of music software should support both knowledge structures - geometrical and logical. A musician mastering this software is unaware of all the technical problems raised by this operation, he simply does a natural operation that is similar to selecting pages of music notation of Violin I, Violin II, Viola, ... parts and providing them to performers of these instruments.

On the other hand, it is desirable to provide the software user with a tool that will permit operating on score parts of an already created score acquired by recognition, imported from a disk file or created from scratch. In the meaning of logical knowledge structure - Figure 11 - the user is able to insert/remove a score part to/from an existing score, cf. Figure 13. The operation of adding a score part affects both knowledge hierarchies: geometrical and logical. Note that the geometrical and logical hierarchies are abstract concepts explaining data structures allocated in computer system in order

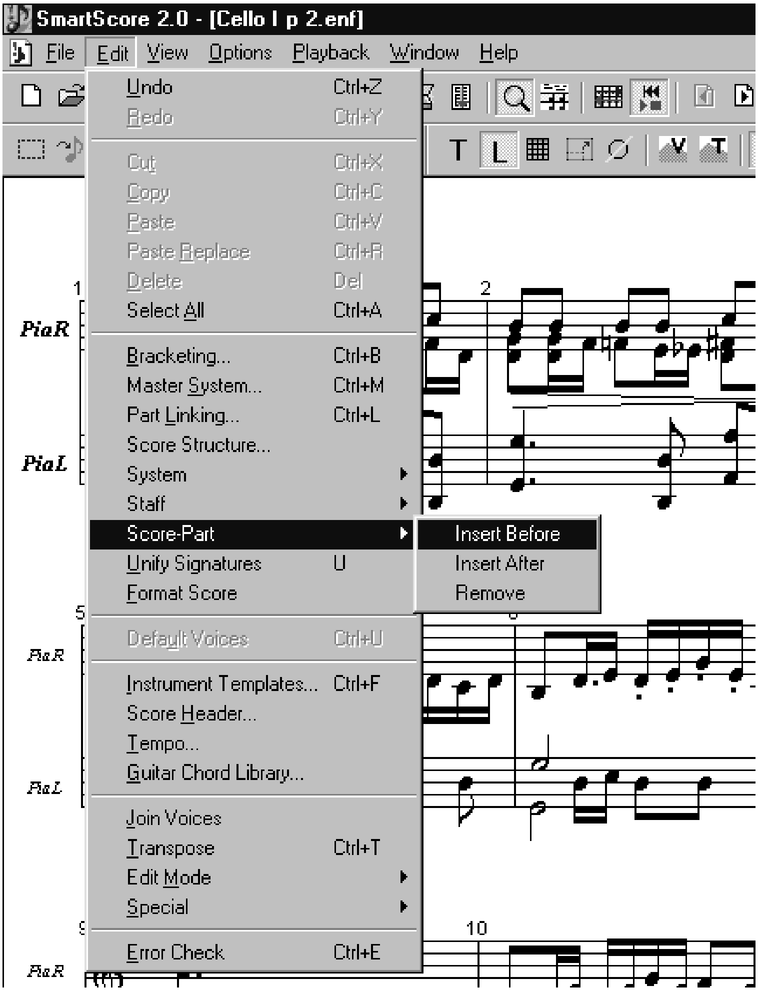


Figure 13. Score structure mastering

to support respective information. Of course, the data structure is created and allocated in computer memory in order to represent new information quantity. However, creating, allocating in computer memory and maintaining a new data structure is a technical problem that should not bother a software user. Moreover, this technical problem should not affect the process of user interface designing. Otherwise, if the process of user interface designing become affected by technical problems, user interface instead of reflecting the natural direction of human behavior will be twisted to technology and implementation perspective.

6. Conclusions

The granular computing paradigm is concerned with the processing of information at levels of abstraction that are more closely related to human information processing than to numerical data computing. The challenges that are raised by granular computing paradigm are focused on the improvement of understanding of the abstractions of knowledge forming process, the efficient derivation of such abstractions, meta-modelling, qualitative reasoning and other operation modelling human behavior. All these essential components of granular computing are intuitively utilized in soft computing, often without the awareness of software developers. In the paper some aspects of the granularity of data aggregation and data abstraction as a process of understanding a quantum of information supported by primitive data of rater bitmap are investigated. The discussion is related to Optical Music Recognition issue interpreted as paper-to-computer-memory data flow. This specific area of interest forces specific methods to be applied in data processing, but in principle, gives a perspective on the merit of the subject of data aggregation. The process of data aggregation and data abstraction investigated in the paper move information processing beyond the framework of phenomenological connections between the detailed models of data. In this sense, granular computing paradigm is well placed to integrate structural and behavioral system abstractions and to investigate the dependencies between system modelling domains. The user interface of the system is outlined as a domain depending on the internal, technical factors of the system as well as on the user point of view of the system interface. The difference between developer perspective on user interface design and user expectations of it is outlined. Designing of user interface focused on the natural direction of human behavior rather than on technological factors is underlined as one of the most important conditions of user friendly software creation.

The idea of granular computing, though does not provide a new technological edge break, is a perfect tool to express the problems of optical music recognition. It gives a view at OMR from a new, fresh and interesting perspective. This new perspective could allow for better understanding of the structure of music knowledge representation and processing, of the structures of music notation from the point of its recognition, for better understanding of recognition methods and algorithms, as well as better understanding the structure of the recognition process itself. Finally, the granular approach to the knowledge representation helps software developer to focus his gaze on the goals toward which users strive rather than on technological factors and developer's technical education and experience, what is clue point in gaining high level of software usability.

References

[1] J.P. ARMAND: Musical score recognition: A hierarchical and recursive approach. *Proc. 2nd Int. Conf. on Document Analysis and Recognition*, (Cat. No.93TH0578-5), (1993), 906-909.

[2] A. BARGIELA and W. PEDRYCZ: Classification and Clustering of Granular Data. *Joint 9th IFSA World Congress and 20th NAFIPS Int. Conf.*, Vancouver, (2001), 1696-1701.

[3] K. BARBAR, M. DESAINTE-CATHERINE and A. MINIUSSI: The semantics of Musical Hierarchies. *Computer Music J.*, **17**(3), (1993), 30-37.

[4] D. BLOSTEIN and H.S.BAIRD: A Critical Survey of Music Image Analysis. In H.S.Baird, H.Bunke, K.Yamamoto (Eds), *Structured Document Analysis*, Springer-Verlag, (1992), 405-434.

[5] N. CARTER and R. BACON: Automatic Recognition of Printed Music. In H.S.Baird, H.Bunke, K.Yamamoto (Eds), *Structured Document Analysis*, Springer-Verlag, (1992).

[6] A. COOPER: About Face: The essentials of User Interface Design. IDG Books Worldwide, Inc., 1995.

[7] H. FAHMY and D. BLOSTEIN: A graph grammar programming style for recognition of music notation. *Machine Vision and Applications*, **6** (1993), 83-99.

[8] M. FERRAND and A. CARDOSO: Scheduling to reduce uncertainty in syntactical music structures. Advances in Artificial Intelligence. *Proc. 14th Brazilian Symp. on Artificial Intelligence*, (1998), 249-258.

[9] I. FUJINAGA: Optical music recognition using projections. Master's thesis, McGill University, Montreal, Canada, 1988.

[10] T.W. GOOLSBY: Eye movement in music reading: Effects of reading ability, notational complexity, and encounters. *Music Perception*, **12**(1), (1994), 77-96.

[11] T.W. GOOLSBY: Profiles of processing: Eye movements during sightreading. *Music Perception*, **12**(1), (1994), 97-123.

[12] W. HOMENDA: Optical pattern recognition for printed music notation. *Proc. SPIE's Int. Symp. on Aerospace/Defense Sensing & Control and Dual-Use Photonics*, Orlando, USA, (1995), 2490: 230-239.

[13] W.HOMENDA: Optical Music Recognition: the Case of Granular Computing. In *Granular Computing: An Emerging Paradigm*, Physica Verlag/Springer Verlag, (2001), 341-360.

[14] T.KINOSHITA, ET.AL.: Note recognition using statistical information of musical note transitions. *J. Acoustical Society of Japan*, **54**(3), (1998), 190-198.

[15] G.E. KOPEC, ET.AL.: Markov source model for printed music decoding. *Proc. SPIE - The International Society for Optical Engineering*, (1995), 2422: 115-125.

[16] B.R. Modayur, et.al.: MUSER: a prototype musical recognition system using mathematical morphology. *Machine Vision and Applications*, **6**(2-3), (1993), 140-150.

[17] NIFF, Notation Interchange File Format, Document ver. 6a.3, October 1998.

[18] W. Pedrycz: Neural Networks in the Framework of Granular Computing. Manuscript, 2001.

[19] W. Pedrycz: Granular Computing: An Introduction. *Joint 9th IFSA World Congress and 20th NAFIPS Int. Conf.*, Vancouver, (2001), 1349-1354.

[20] D.S. Prerau: Do-re-mi: A program that recognizes music notation. *Computers and Humanities*, **9** (1975), 25-29.

[21] T. Ross: The Art of Music Engraving and Processing. Hansen Books, Miami, 1970.

[22] Smart Score v.2.0, www.musitek.com, Ojai, CA, October, 2001.

[23] K. Stone: Music Notation in Twentieth Century: A Practical Guidebook. W.W.Norton & Co.,New York, 1980.

[24] H. Taube: Stella: Persistent Score Representation and Score Editing in Common Music. *Computer Music J.*, **17**(3), (1993), 38-50.

[25] WWW Page, Optical Music Recognition Bibliography, by Ichiro Fujinaga, October 2000: http://gigue.peabody.jhu.edu/ ich/research/omr/omrbib.html