Politechnika Wrocławska

Wydział Informatyki i Zarządzania
kierunek studiów: Informatyka
specjalność: Inżynieria Komputerowa

Praca dyplomowa - magisterska

OMR for sheet music digitization.
(pol. OMR w cyfryzacji zapisu nutowego)

Krzysztof Jastrzębski

słowa kluczowe:
OMR system, Pattern recognition
Music Sheet, MIDI, MusicXML

krótkie streszczenie:
Praca opisuje działanie systemów OMR służących do automatycznego rozpoznania i cyfryzacji zapisu nutowego. Opisuje możliwości dostępnych na rynku programów OMR oraz problemy jakie powodują niedoskonałości odczytu. Przeprowadzone testy obrazują jakie typy zapisu nutowego mogą być odczytane. Dodatkowo zostały opisane najbardziej popularne formaty zapisu cyfrowego notacji muzycznej.

opiekun pracy dyplomowej	..........dr inż. Marek Kopel........	....	....
	Tytuł/stopień naukowy/imię i nazwisko	*ocena*	*podpis*

*Do celów archiwalnych pracę dyplomową zakwalifikowano do:**

a) *kategorii A (akta wieczyste)*
b) *kategorii BE 50 (po 50 latach podlegające ekspertyzie)*

* *niepotrzebne skreślić*

pieczątka Instytutu, w którym student wykonywał pracę

Wrocław 2014

Abstract

Poniższa praca opisuje zasady działania systemów OMR służących do automatycznego rozpoznawania oraz cyfryzacji zapisu nutowego. Poprzez zastosowanie się do 4 najważniejszych procesów tych systemów w rezultacie otrzymujemy cyfrowy zapis nut który może być zapisany w postaci różnych formatów muzycznych specjalnie przystosowanych do przechowywania tego typu informacji. Dane zapisane w formatach takich jak MusicXML, MIDI czy też NIFF pozwalają użytkownikowi na swobodną edycje uzyskanych symboli muzycznych, odtworzenie danego zapisu nutowego w formie dźwiękowej czy też podział jednego systemu nutowego na kilka różnych. Praca ta zawiera również testy 8 dostępnych na rynku systemów OMR na podstawie których można zauważyć problemy oraz wyzwania wynikające z procesów rozpoznania oraz interpretacji zapisu nutowego. Przetestowanych zostało 21 różnych wersji nut począwszy od prostych zapisów, poprzez partytury na zapisach perkusyjnych kończąc. Dzięki tym testom możliwe jest zaobserwowanie jakie typy zapisu nutowego ciągle są niedopracowane lub są pomijane przez twórców takich systemów jak OMR.

The following work describe rules of operation of OMR systems which are used to automatically recognition and digitization of music sheet. Through use of the 4 main processes of those systems in result we receive digital form of music notation which can be later saved in the form of different types file formats that are especially created to storage such type of information. Data stored in such format like MusicXML, MIDI or NIFF allow users to easy edition of obtained music symbols, play given music sheet in the form of sounds and division of one music system into few others. This work include also tests of 8 available on the market OMR systems based on which can be seen different problems and challenges resulting from the processes of recognition and interpretation of music sheet. Has been tested 21 different version of music sheet starting with simple notes, through orchestral score and finished on drum set music sheet. Based on those tests it is possible to see what kind of music sheet are still underdeveloped or are omitted be developers of such OMR systems.

Content:

1. Introduction.

Optical Music Recognition (OMR) is a form of optical character recognition that use different method and algorithms to convert printed music sheet into its digital form. It is often compared to OCR (Optical Character Recognition), in that it allows you to switch from one image to the symbolic and semantic description of its content, methods of treatment and analysis of digital images. After recognition, music notes should be saved in appropriate file format for later play and edition.

First attempt of music recognition was in end of 1960s at MIT. The research had aim to develop method of capturing stave to recognize and translate all character that laying on it. First commercial score recognition software was "Midi scan" developed and published by the company Musitek in 1991 (The program was later renamed into "SmartScore").

After recognition of musical symbols, it is possible to repeat music sheet and change it at will (transcription, pitch transposition, arrangements, etc.). A significant time saving is achieved through the automatic recognition, since the task of manual input of all characters (particularly in long music sheet) is avoided. Later conversion into an audio format like MIDI allows musicians to listen to music written on music sheet.

To better understand the following work in chapter 2 are described basic music theory and rules that codify the structural relationships (for example organization of group of notes), graphics (such as the position of the symbol on the staff lines) and syntactic (metric, key etc.). Those rules provide a main information usable for recognition of music sheet.

Because OMR systems based on the rules and method of pattern recognition, in chapter 3 are described basic methods and processes that comes from this field of artificial intelligence.

There are existing many of music sheet file format. Most of the software that are used to editing musical symbols used their own file format. In chapter 4 are described most popular and standard file formats that are used to interchange information between systems that recognized musical characters and systems that are used for creating and editing music sheet.

Most of OMR systems are based on 4 basic processes: Detection of staff lines, symbol location, music symbol identification and semantic of music notation. After going through those processes system return to user music sheet in form ready for saving as a one of music sheet format or for printing. Detailed information about those processes can be found in chapter 5.

In OMR systems appear problems that are very numerous and occur at different stages of the analysis. Examples include segmentation difficulties due to high degree of interconnection between symbols (the presence of staff lines in particular), all difficulties related to printing defects (split symbols, poorly positioned, etc..) and variability of symbol fonts. While musical scores are highly structured documents, apparently following the defined rules, we note that in practice these rules are very flexible. All these problems connected with recognition of music sheet are described in chapter 6.

In chapter 7 are described tests of 8 different OMR systems that are available on the market. Some of those systems are small and simple in use, other include many of additional features that helps to better analyze recognized music sheet. Because in real live there are available many different types of music sheet (for example single stave voice, piano 2 stave voice, drum set stave voice …) I selected 21 different samples of music sheet and check how accurate and fast are those 8 chosen OMR systems. Samples include music sheet generated by computer using music editor software and scanned music sheet. Whole samples are appended in chapter 9.

2. Basic music theory.

To better understand the following work it is necessary to provide basic definition and music theory that later on will be used in this document.

2.1. Staff, Stave.

Fig. 2.1: Staff lines

In theory, the staff is a set of five horizontal lines to represent different pitch. It is intended to receive the figures like notes and rest, keys, and some other ancillary symbols.

The five lines of the staff are equally spaced by four space. Lines and spacing are numbered from bottom to top. Note (the form used to indicate durations) are placed either on the lines or between the lines. The pitch is indicated by the position of the head of the notes.

2.2. Bar line.

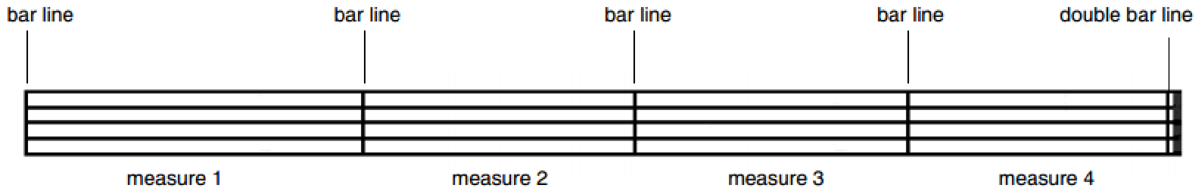


Fig. 2.2: Bar lines and measures

Bar line refers to a grouping of certain set of musical notes on values with the same beats (measures).

2.3. Note.

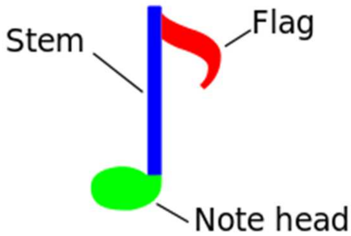


Fig. 2.3: Elements of note

In music, a note is either a symbol to represent the pitch and relative duration of a sound, or the pitch itself of a sound. The note represent a time (horizontal dimension of notation). By its position on the staff, it represents also pitch (vertical dimension).

2.4. Notes duration.

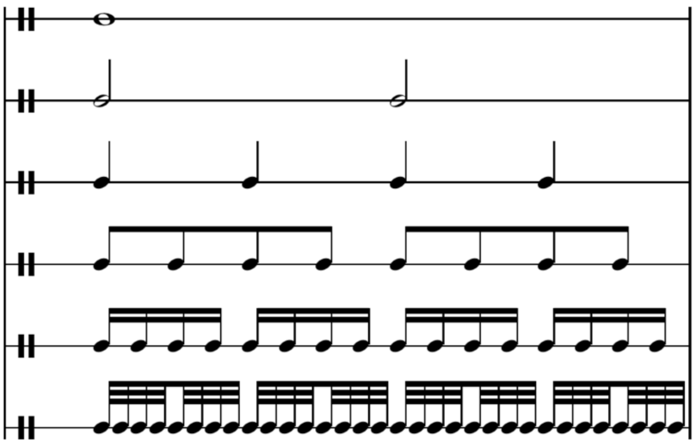


Fig. 2.4: Division of note

1. Whole note - in modern musical notation note with the longest, basic rhythmic values, which are obtained by dividing the remaining values.
   The whole note is divided into two half notes, four quarter, eight eights, etc., and is determined by the unfilled oval head without a steam or flag. Sometimes a whole note is

used for determining the rhythm of the duration of a single cycle, even if the meter shows a different length. It is counted to four.

2. Half note - note in the musical notation of duration equal to half of a whole note, two quarter-, four eights, etc. We count it to two.
   Half note is marked as unfilled oval head with a stem, generally on the right side facing up the notes located below the third line and down the left side of the notes located on the third line or above. Half note is marked without flag.
3. Quarter-note in musical notation lasting one quarter of a whole note or half a half note, dividing into two eights. A quarter note is marked as filled oval head without flag and with a stem generally on the right side facing up for notes located below the third line or down the left side of the notes located on the third line or above.
4. Eighth note-note in musical notation indicating the sound of duration equal to one-eighth of a whole note.
   Eights are written as oval, blackened head with a vertical stem and one flag.
   Flags are always placed to the right of the stem. If several eights adjacent to each other, they can be combined into a single symbol using a single beam.

2.5. Clef.

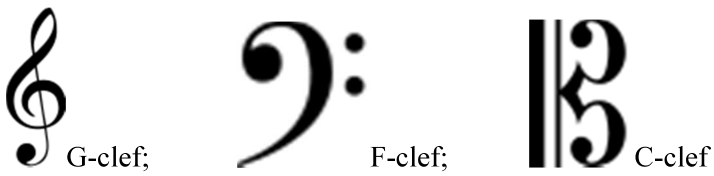


Fig. 2.5: Three types of Clef

In theory, a clef is a graphic sign placed at the beginning of the staff which indicates the pitch associated with each line.

The key is necessarily the beginning of any staff, but can also occur further to indicate a change of key. There are three key figures: G, F and C. Drawing key is derived from the letters, which had become accustomed to put on appropriate line of the original staff.

2.6. Alteration.

Alteration (also called accidental)- In music, the alteration term can mean three things:

1. Alteration refers firstly to the modification of the initial pitch of a note. The reasons for this change: modulation , transposition, ornamentation ...
2. Alteration means the charge symbol to indicate this changes.

3. It is also possible to consider an alteration represents the new note as amended

Three are three different alteration symbols: the sharp, the flat and natural.

Fig. 2.6: Alteration symbols

2.7. Scoring rules

Music theory codifies writing music. The rules are graphical or command syntax. Here is list of the most important rules.

The graphical rules relating to the position of the symbols:

1. Accidental must be placed before the note it alters, and in the same vertical position.

2. A point (doted note) of time must be placed after the note head.

3. Point of accent is placed above the note head.

Syntactic rules for the tone of the measure and the metric:

4. The number of beats per measure should always match a given metric at the beginning of the first staff.

5. The notes are generally grouped in time, multiple time or a fraction of time, to facilitate the reading rhythm. For example, a simple measure, the reference time is the quarter (2/4, 3/4, 4/4). We then find groups of eighth note whose total duration is one or more quarter note, or sometimes even a sixteenth note.

Fig. 2.7.1: 4/4: 1+1+1+1/5+1/5

Fig. 2.7.2: 3/4: 1+1+1

Fig. 2.7.3: 3/4: 2x1+1| 3x1

For a ternary rhythm, e.g. 3/8, 6/8, 12/8, duration is equivalent to a dotted quarter,

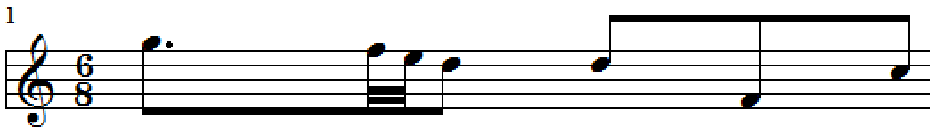


Fig. 2.7.4:6/8: 1 and 1/5 +1 and 1/5

but sometimes equivalent to periods of one or two group of notes.

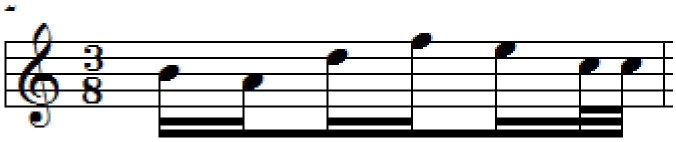


Fig. 2.7.5: 3/8: 1/5+1/5+ 1/5

Note that silence can replace notes in the group, regardless of the metric.

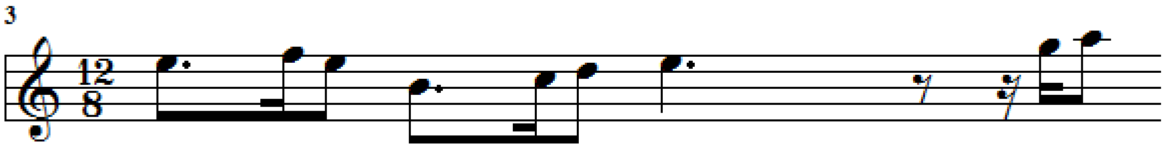


Fig. 2.7.6: 12/8: 1 and 1/5 +1 and 1/5 +1 and 1/5 +1 and 1/5

6. Alterations to key follow a predefined order and indicate the key of the music sheet. They are implicitly applied to all notes of the same name (same height near the octave): for example, all the way partition.

7. Alteration is applied to the next note, but also, by implication, to all notes of the same name present in the rest of the measure.

These rules are still applied with varying degrees of flexibility, or with a few exceptions.

Typically, a strict rule is the rule 4 on the metric. It is always respected, except upbeats (before the first measure), or repeats. In the latter case, this is the sum of the bar before repeats, and the extent to which it is returned, which must satisfy the metric.

Most of the other rules are flexible rules, that is to say, they are generally adhered to, but that it may be released or, they can be applied in different ways.

For example, Rule 7 indicates that it is not theoretically useful to recall an alteration in the same measure. Nevertheless, one can find redundant alterations, which do not provide additional information and therefore should not theoretically be present, but that facilitate reading.

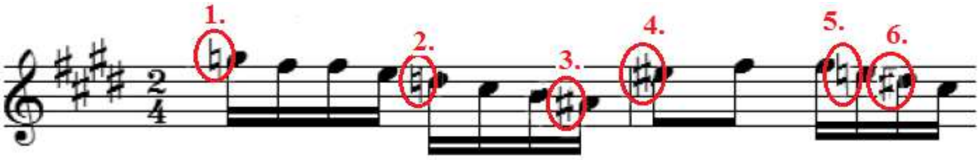


Fig. 2.7.7: Exception from rules no 7

Naturals 1 and 2 (Fig. 2.7.7) cancel alterations to the key, the natural 5 cancels the sharp 4 that precedes in the same measure. In contrast, the sharp 6 is redundant because it is present implicitly (in the key). It is easier to read because the D was natural sign in the previous measurement.

Fig. 2.7.8: redundant natural

The natural sign 2 is redundant: it succeeds B flat, but in a new measure, and it is not altered in the key. Again, it allows easy reading.

Rule 5 concerning the rhythmic arrangements of notes, is also flexible rule, because the notes can be grouped in different ways, each arrangement is consistent with the rule, and that rule can be released for phrasing questions.

Finally, the graphic rules indicate approximately the position of the symbols relative to each other. Nevertheless, one can find offsets varying from one edition to another, or even within the same music sheet, depending on the density of symbols.

Conclude this section by noticing that all these rules that codify musical notation act at several levels. Some are related to the structure of symbols (groups of notes) or their relative positions, others are purely syntactic. Some are local, others involve very distant symbols. Finally, some rules express binary constraints (between two symbols), other constraints of higher order. All jointly involved in the interpretation.

3. Pattern Recognition.

Pattern recognition is a set of techniques and methods to identify computer patterns from raw data to make a decision depending on the category assigned to this pattern. It is considered that this is a branch of artificial intelligence that makes extensive use of machine learning techniques and statistics. The issue of recognition of music sheet image can be complex because it requires in some cases to be robust in translation, rotation and zoom. It may also happen that the documents can be degraded (noise, spots, cuts, etc.).

General methods have been developed for pattern recognition to automatically extract information from sensitive data to characterize the shape classes (learning) and automatically assign data to these classes (recognition). The implementation of these general methods for specific problems leads to introduce the concept of recognition process

3.1. The scheme of the pattern recognition.

It is based on the classic process of pattern recognition scheme to describe the main processes to be performed and objectives.

Main stages of pattern recognition scheme:

- Scanning: get a representation of the data to be processed which can be handled by machine.
- Preprocessing: noise removal, normalization, re-sampling, contrast enhancement, etc..
- Calculation of representations: obtain a representation of data consistent with the learning tools and decision used.
- Learning: from a set of examples, construct a representation of classes.
- Analysis: assign to an unknown form a class.
- After treatment: validate the decisions of the analysis on the basis of knowledge (the domain).

In practice, a system of pattern recognition is often away from this scheme. Upstream treatments are often necessary to isolate the form to recognize its context, which in itself is a recognition problem (segmentation shape / bottom boundary of a shape in a set). Subsequent treatments are also useful to validate decisions and possibly challenge.

The sequential nature of the recognition process, as shown in this scheme is not always the best option. Thus, an error in the segmentation of the form to acknowledge necessarily increase the risk of poor recognition. It is possible to introduce a loop in the process, undermining the segmentation after analysis of the results of recognition. It is also possible to combine segmentation and recognition.

3.2. The methods of the pattern recognition.

Methods of pattern recognition are often grouped into broad classes identified by statistical, syntactic, structural and hybrid (a combination of others).

At these classes are different ways of representing the practices and classes and different methods for learning and recognition. Pattern recognition are interested in the simulation of a human being in a recognition task.

The first approach to the simulation of a human recognition is a fully automated processing where the "master" label to simulate the outcomes that will be used for training and validation of recognition. Statistics and syntactic methods meet this approach. The difficulty to automatically learn the descriptions that characterize the classes led to ask the expert to clarify these descriptions to help build the representation of classes and criteria of decision. This is where the structural methods involved. In fact the recognition cannot be reduced to an analysis of the form, this analysis is often an element in a decision making process that involves other information or other knowledge.

3.3. Pretreatment.

During pretreatment, the raw data gathered from sensors are converted into data that are later use in recognition process. It is necessary to eliminate from raw data all noises and obstacles that not include any important information for recognition. For example, information on the shape of the objects to be recognized is contained in the array of pixels resulting from scanning an image but treatments will be needed, for example, separating the shape of the bottom and follow the contour.

For pretreatment, the designer using the knowledge they have on sensors, data types, the problem and the learning and recognition methods it will use. Pretreatments are useful for removing noise which may be due to the sensor or interference with other signal sources.

Signal and image processing are the main sources for the methods of pre-processing: filtering, contrast enhancement, edge extraction or skeletons, modeling time signal extraction primitive, etc..

Pretreatments and learning have common goals to eliminate spurious information and retain relevant information for recognition.

3.4. Representation.

The pattern recognition system must have representation forms and rating classes.

Representation types are determined by the methods used. The contained representations are determined by the goals of recognition.

Some types of representation are most appropriate to express some content. For example, intra-class variability that occurs as a random distribution of characteristics of the shape around highly probable values will be well represented by statistical models. For cons, the structures (relations between components of a pattern) can more easily be expressed by structural representations.

Information that can be measured on the data to recognize it is called characteristic (or descriptor) information. For example: the average amplitude of a signal over a time window, the energy in a frequency band, the height to width ratio of a handwritten character, the average gray level of an image area etc..

The form of a basic component is called a primitive form (primitives). The primitives are not decomposable. For example, a line segment or a loop.

3.5. Representation of classes.

The generalization of the information carried by the copies of the training set for the representation of classes emerge learning aims. These representations must be able to overcome the variability of copies of a class during the decision making. Validation of learning is done by the recognition rates obtained on a test set consisting of different copies of those in the training set. For approaches that exclude the learning, decision criterion (a measure of similarity) shall pay only the intra-class variability.

3.6. Similarity.

What is problematic to recognize shapes as copies of a class is the one hand the intra-class and second class separability. Copies of a class are not identical, they are similar, while the copies belonging to different classes are not alike. The problem will be to formalize this resemblance.

The distribution of copies in the space of representation rarely show a perfect separation between the different classes, which causes ambiguous decisions . We can associate a value with the decision to indicate how confident it was taken .

If the classes are represented in extension , the similarity can be measured :

- The distance of an unknown copy prototype is the pattern matching. Methods : Dynamic Programming for comparing strings , graphs , elastic masks ...

- The distance of an unknown example copies whose membership class method of k-near - neighbors - are known.

If the classes are represented by languages, logic descriptions, the similarity is given by the position of the unknown exemplary with respect to the spaces defined by the representations of classes. Methods: statistical decision theories, parsing for classes represented by languages.

3.7. Statistical methods.

By the situation where there is a body of work labeled for supervised learning (as opposed to methods that start with a zero knowledge on the labels achievements and seek to grouper on similarity criterion) is considered.

The objective is always to assign an unknown realization of belonging to his class by minimizing the error decision. This problem can be solved in many ways, the choice of method depends in part on prior knowledge that we have on the probability distribution of copies of classes.

Bayesian case corresponds to decision criteria constructed from probability distributions. The parametric case is a priori knowledge about the laws of probability that must estimate the parameters.

3.8. Structural methods.

Arguably, to clear most of the difference between statistical and structural methods, on the one hand is represented by a given feature vectors and the other is represented by a description in the form of components and relationships between these components.

The notion of structure, though subject to many definitions, reveals however, the existence of a decomposition of the whole (the structure) into parts and relationships between these parts. Structural pattern recognition methods adopted representations making explicit the relationships between elements of the form. They are clearly more suitable for processing visual structures where the components are related by spatial relationships.

The decomposition of a shape can show a hierarchy of components. Is called primitive elementary components, information of finer granularity that cannot be broken (the level of information of these primitives is determined by the system designer recognition).

A first phase of the recognition will be to reduce the raw data to a set of components, that is to say fragments in the segment, following one or more levels of decomposition. Descriptive features of a structure are of type fragment but also traits that express configurations such as symmetry or closure, or events that occur on the form as a change of direction at an angular point or even characteristics as the height to width ratio, the moments, the center of gravity, information that characterizes the overall structure.

Relationships between components can be explained by different representation formalisms. The simplest is the chain of components, particularly adapted to the case of one-dimensional signals and contours.

4. Music sheet file formats.

4.1. MIDI

The Musical Instrument Digital Interface or MIDI is a protocol for communication and control for exchanging data between electronic musical instruments , one or more of these "instruments" can be computers. MIDI is managed by the International Midi Association.

The connection is unidirectional (simplex). The great advantage of this protocol is its interactivity: it serves both for control and for recording. Under this term are grouped several standards relating to the logical protocol , physical interface, file format and attribution sounds.

The first public demonstration took place at the NAMM (National Association of Music Merchants) in 1983 , between a Jupiter -6 (Roland) and a Prophet- 600 (Sequential Circuits) by two representatives and their founders, Mr. K Roland and Dave Smith for Sequential Circuits. Dave Smith is also one of the designers of the MIDI standard.

The information is sent digitally, in series, at a speed of 31250 bauds, approximately 3.8 Kb/s. Each connection sends standard musical messages as note-on (start note), note-off (end note), volume, pitch bend (modulation of the pitch of the note) and modulation signals encoded with an identifier channel (there may be up to 16 channels).

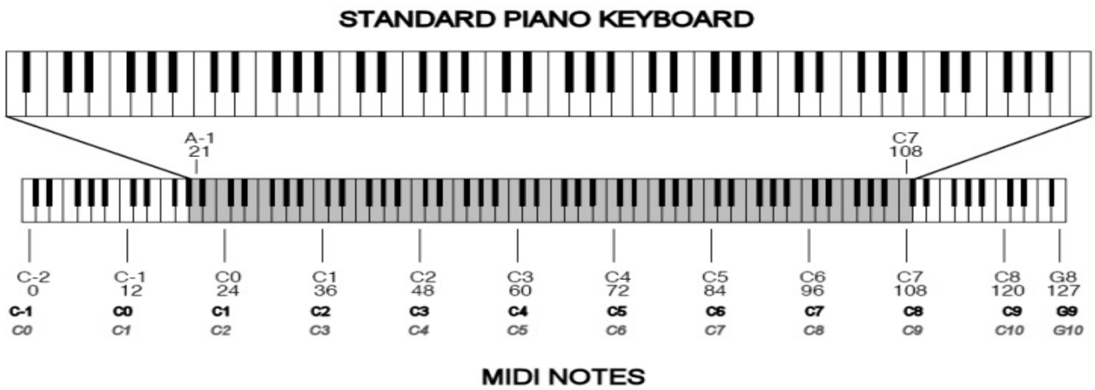


Fig. 4.1: Range of MIDI notes

Note messages range from C-1 (note no. 0, located 5 octaves below the C4 underneath the partition clef, 8,175 Hz) and G9 (note no. 127, 5 octaves above average terrain, 12 557 Hz) with a resolution of one semitone. The pitch can be adjusted ± half ton with an accuracy of 1/4096 of a semitone using pitch bend message. Thus we find MIDI converters for most musical instruments (trumpets, guitars, etc..). This allow, for example, to play the piano with a guitar or synthesizer with a trumpet.

Format MIDI SMF (Standard MIDI File) was defined in 1988 to store the MIDI commands on disk by adding time information.

They can create either by connecting a MIDI instrument to the MIDI input of a computer or using a sequencer software. MIDI format is very compact.

There are three different formats of MIDI files:

0: Only a track containing the messages of the 16 channels;

1: number of tracks played simultaneously;

2: more tracks played sequentially (rarely used).

The files are usually type "1" because it is more convenient to separate the tracks, but some synthesizers support only Format 0. Various software can convert one format to another.

Currently, a standard midi file comes with the extension .mid or .midi.

KARAOKE MIDI files have the extension .kar instead .mid, but they are real MIDI files with a karaoke track with lyrics that some hardware / software's are capable of displaying scrolling in manner and synchronized with music.

4.2. MusicXML

MusicXML is an open file format based on XML for musical notation.

MusicXML was developed by Recordare LLC , deriving concepts from different existing academic formats (like MuseData or Humdrum) . It was designed for the transmission of musical scores.

Version 1.0 was released in January 2004. Version 1.1 followed in May 2005 with an improved format support. Version 2.0 was released in June 2007 and includes a standardized compressed format. All these versions were defined by a series of DTD (document type definition). XML Schema (XSD) for version 2.0 was made public in September 2008 . Version 3.0 was released in August 2011 as a DTD and XML schema .

From September 2008, MusicXML was supported by more or less large part by more than a hundred music notation programs , including:

Final scores , Sibelius, Pizzicato, Guitar Pro, capella, PriMus and most of software for optical music recognition

MusicXML format is based on XML and therefore it is easy to program on nearly any computer platform. Currently, most programs (such as Finale, Sibelius, Symphony Pro) that edit music, support this technology. As a result, the choice of this format seems to be reasonable in any OMR systems. Besides, due to the fact that it is based on XML format should not be problems with processing read out character from the OMR.

"Hello Word" in MusicXML:

Fig.4.2: Normal note that will be convert into MusicXML

For example whole note "C", meter 4/4 in G-clef (Above picture) is describe as follow:

```
<?xml version="1.0" encoding="UTF-8"
standalone="no"?>
<!DOCTYPE score-partwise PUBLIC
    "-//Recordare//DTD MusicXML 3.0 Partwise//EN"
    "http://www.musicxml.org/dtds/partwise.dtd">
<score-partwise version="3.0">
  <part-list>
    <score-part id="P1">
      <part-name>Music</part-name>
    </score-part>
  </part-list>
  <part id="P1">
    <measure number="1">
      <attributes>
        <divisions>1</divisions>
        <key>
          <fifths>0</fifths>
        </key>
        <time>
          <beats>4</beats>
          <beat-type>4</beat-type>
        </time>
  <clef>
          <sign>G</sign>
          <line>2</line>
        </clef>
      </attributes>
 <note>
        <pitch>
          <step>C</step>
          <octave>4</octave>
        </pitch>
        <duration>4</duration>
        <type>whole</type>
      </note>
    </measure>
  </part>
</score-partwise>
```

Fig. 4.3: "Hello Word" in MusicXML[19]

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

This is just ordinary declaration demanded in every XML file.

```
<!DOCTYPE score-partwise PUBLIC
"-//Recordare//DTD MusicXML 3.0 Partwise//EN"
"http://www.musicxml.org/dtds/partwise.dtd">
```

This part of the code says that we want to use MusicXml. It use public declaration which include internet location for DTD

```
<score-partwise version="3.0">
```

This is type of main document. Element <score-partwise> consist of parts which consist of measures .

```
<part-list>
  <score-part id="P1">
    <part-name>Part 1</part-name>
  </score-part>
</part-list>
```

MusicXML file begins from providing header that contain different part of music in the notation. Above example contain only one element (one part). It is necessary to write attribute id and name of the element.

```
<part id="P1">
```

In this moment document begins with first document element.

```
<measure number="1">
```

Begins measure.

```
<attributes>
```

Begins elements attributes. Attribute contain key information needed to note interpretation.

```
<divisions>1</divisions>
```

Each note in MusicXML file has own length. “division” provide unit of measure for element length in division on quarter note. Because in this example exist only one (whole) note, document didn’t need to divide anything that’s why value is equal to 1.

```
<key>
    <fifths>0</fifths>
</key>
```

KEY is used to describe music key of given element. In above example there is whole note in key C-major (Without any flats or sharps) that's why <fifths> is equal to "0". For comparison, if key would be in D-major (with two sharps) <fifths> would by equal "2" and for key in F-minor (with one flat), fifths> would by equal "-1"

```
<time>
  <beats>4</beats>
  <beat-type>4</beat-type>
</time>
```

Element <time> represent time signature (tempo).

```
<clef>
  <sign>G</sign>
  <line>2</line>
</clef>
```

Element <clef> is responsible for introducing a music clef. In this example there is treble clef that lies on second (from the bottom) line (line G)

```
</attributes>
<note>
```

In this moment document stop describing attributes of music sheet and start with description of notes.

```
<pitch>
  <step>C</step>
  <octave>4</octave>
</pitch>
```

Note step this is „C" in the 4th octave .

```
<duration>4</duration>
```

Value of division was set as 1 for each quarter note, that's why duration is equal 4.

```
<type>whole</type>
```

Element <type> says , that given not is whole. It can by deduce from length of duration however it is easier to work with booth description.

```
</note>
</measure>
</part>
</score-partwise>
```

And finally document close all parts of the code.

4.3. NIFF

NIFF is a file format for exchanging data between different music notation programs and editing applications. Its architecture is a combination of the work of many software developers , publishers , and even users.

NIFF was a response to the needs of publishing professionals , who suffered from the lack of a standard file-sharing music notation format. It represent music graphically and data are in the binary form.

Rules of creating NIFF file format are based on RIFF (Resource Interchange File Format) designed be Microsoft. Music characters with similar behavior are grouped together and then related groups create lists. Size of list and group is define inside their structure.

4.4. SMDL

SMDL an architecture for musical information, representation alone, or combined with text and graphics.

The Standard Music Description Language is a standard for developing the exchange of scores and recordings between composers, musicians and music publishers. A definition and an explanation of its four areas and its basic structure helps to understand its ramifications. SMDL is the first proposal of an ANSI (American National Standards Institute) in 1984, but since then, it has undergone many changes and has contributed to the emergence of new standards, like NIFF.

5. General scheme of music sheet recognition

In contrast to the optical character recognition of text (OCR) in which characters can be recorded in sequence (usually horizontal sets of character) , music notation introduce another dimension (Time is recorded in horizontal form and pitch in vertical). The detection of the spatial arrangement of notes, dynamics, fingerings , the accents and other additional notation elements is an additional problem in the optical detection of notes .

General process of OMR software can be divided into two phase: Working and learning.

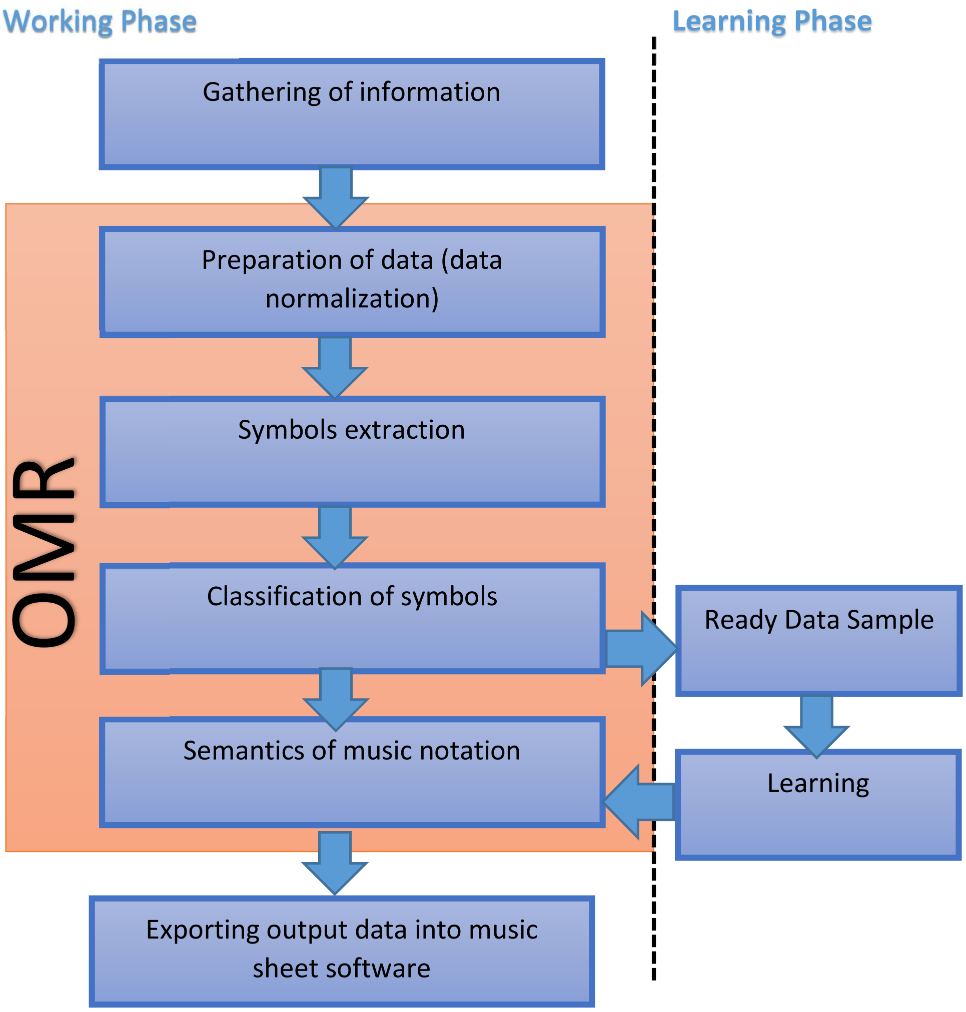


Fig. 5.1: General process of OMR software's

Information is gathered via different sours. It is possible to read data directly from a hard disc connected to PC in the form of image file like bitmap image file or Tagged Image File Format (TIFF). Second option is to use the scanner. Some programs are also able to read PDF files with notes.

There are 4 steps in each OMR system:

1. Staff (stave) line detection- The rate of recognition is limited, depending on the quality of the scanned document that's why first step in the process in OMR system are preparation of data sample. By preparation some computer software are able to eliminate potential source of errors like dust on sheet, removing irrelevant text or can rotate whole documents to obtain best position of notes. Such modification are done

automatically by more advanced software, however many small programs required from user to prepare music sheet by use of different image processing method.
The most important part of first step of OMR system is identification of staff lines position. Usually stave line are removed by using of special algorithms. This what is important for human being can cause noise for computer system. Remove of 5 lines is in most case quite ease because those lines are regular and straight and that make them easy to locate. System must check continuity of black pixels and similarity to others line using horizontal projection.
Problems occur when line are not straight. It happened very often that during scanning process, document was shifted. Good OMR system however can determine beginning and end position of five lines and using image processing correct wrongly scanned sheet. Another problem is that stave lines are in the contact with other symbols. By deleting whole lines in result symbols would be defected what could make problem in further recognition. Solution to this can be algorithm that remove stave line only if there are no other pixel below and above line.

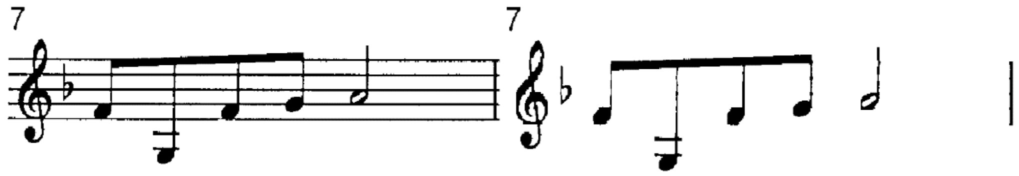


Fig. 5.2: Example of staff removal (images before and after this process)

After removing of line using OCR method, system remove texts, lyrics, dynamic symbols etc.

2. Symbol location- Each symbol that occur on music sheet must be located and mark. To locate every related pixels that remain after removing of stave it can be use flood-fill algorithm.
   The flood-fill algorithm is by diffusion in a conventional computer graphics algorithm which changes the color of a connected set of pixels of the same color defined by contours. It is frequently used by programs handling raster images. The flood-fill algorithm by diffusion takes 3 parameters for a given image: the position of the starting pixel (also called germ), the target color and the color replacement. The algorithm identifies all pixels of the image that are connected by a path to the germ of

the target color and then latter replaces by replacement color. There are several ways to structure this algorithm, but they all use a queue for storage or a stack for storage method to perform the calculation.

If the image is formed conventionally square or rectangular pixels, the 4-connected variant considers the four neighbors of a pixel having an edge in common with the latter.

After detection of symbol, pixels can remain colored or like in the image bellow, around located symbol are draw box.

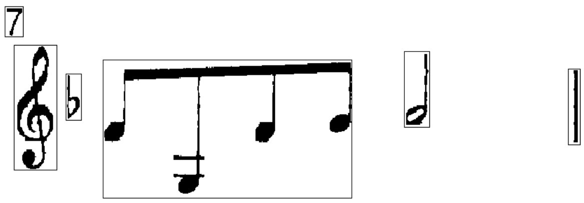


Fig. 5.3: Example of symbols location

3. Music Symbol identification- To all symbol located in previous step must be assigned their type. Such determination is done by comparing located symbol with template from database. There are however problems with similarity of different symbols. To reduce mistakes some of the software use Hough transform[1]. The principle underlying the Hough transform is that there is an infinite number of lines passing through a point, the only difference is the direction (angle).

   The purpose of the transform is to determine which of these lines pass closer to the expected pattern.

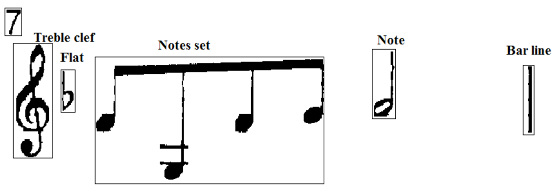


Fig. 5.4: Example symbols identification

4. Semantics of music notation- In this step all symbols and relations must be defined and recorded in the form that allow music sheet software to read them. Each OMR software have own way of presenting notes and other music symbols. For user of such

[1] The Hough transform is a technique of pattern recognition invented in 1962 by Paul Hough, used in digital image processing.

programs the most important is to later convert such symbols into format possible to read by music editor software.

```
Staff System 1:
Staff 1:
{4F 3G 4F 4G} (0.5), {4A} (2.0) |
```

Fig. 5.5: Example semantic of music notation

6. Problems connected with music sheet digitization

OCR music sheet is an area very specific, very different example of character recognition can be found in, as well as a large number of additional symbols.

At first glance, the problem seems relatively simple. Indeed, musical notation involves a relatively small number of symbols, and it is fairly well codified by scoring rules. These can be used at several levels, just to check the consistency of the recognition result, to lead the process of recognition, to render the semantic content from primitive recognized to extract the correct solution among a set of possibilities for recognition.

The difficulties are in fact very large. The first one origin from the segmentation step. This preliminary step is used to locate and isolate the musical symbols before applying the algorithm that will recognize them. In the case of music, the symbols are widely interconnected by staff lines, bars and group of notes. Staff lines interfere in three ways: they connect symbols that should be separated, they camouflage the outline symbols, they fill the hollow symbols. Thus, it is unclear whether, without scope, some pixels are black or white.

At this structural difficulty is compounded by the difficulties related to the quality of the original, often poor input image. It is noted in particular a large number of sheets of the cut segments, and two splitting certain symbols, or conversely, noise connections, but also print defects often result of the high density of the symbols.

Thus, the segmentation of the image into coherent musical entities is a very delicate stage. Generally it begins with deletion of staff lines which deteriorates pretreatment symbols identification. The imperfection of the segmentation result to generate ambiguity, that is to say, the individual analysis of an object can lead to several possible interpretations.

A second difficulty is due to the variability of forms. Can be found in different editions of the same symbol forms varied (different fonts). Symbols can vary significantly even within the same score, mainly because of the imperfection of printing. The result again is a risk of ambiguity, if class models used in recognition does not quite match the symbols of the processed partition.

The concept of musical symbol is important and must be specified. In the previous section I have defined symbols as entities that the musician perceives and interprets (whole note, half note, accents etc.). However, there are various definitions in the literature concerning the optical recognition of music. The simplest is to define a symbol as a set of connected pixels after erasing stave lines, however a group of notes is a single symbol. Many authors, for example J. V. Mahoney, distinguish two categories of symbols: isolated symbols and compound symbols, called iconic or symbolic signs and signs built respectively. Symbolic signs, such as accents or dynamics symbols, are such characters, almost invariant shape and size. Iconic signs are in fact made of a different spatial arrangement of primitives. Thus, a group of four semiquavers consists of four heads of black notes, 4 stems, 2 bars (flags) group. Note heads are relatively invariant, however, the stems and bars are set in group size and orientation. That the compound symbol is well recognized, must all notes be the primitives. It should be noted that, if the set of all primitives is very small, the number of possible arrangements, on the contrary is practically unlimited.

Recognition systems described in the literature do not all have the same defined set of primitives. It is related to objectives and the proposed method. However, most grow the level of decomposition too far. For example, perform a recursive segmentation of objects to the most elementary components (flag, stems, note heads). The reconstruction is performed by introducing the syntactic and structural knowledge. Difficulties are encountered in the sub-segmentation and recognition of the ambiguity on the class of primitives segmented.

There are also a number of rules of musical notation that express the interaction between musical symbols. I have identified the most important in the previous section. Many of musical rules express soft constraints between two or more symbols graphically close or very distant. Thus, tampering must be consistent with the changes in the key, with other changes in the extent and optionally in the preceding steps. In case of ambiguity in the class of accidentals (sharp, flat or natural), check their mutual consistency, knowing that generally no single possible combination, because of the flexibility of musical rules. On the other hand, several rules are usually put together in playing to recognize and interpret a musical symbol.

For example, to find the full interpretation of a dotted[2] eighth, part of a group of notes, it must find its height and length. In particular, to validate the presence of the point of time, it is not only that this is correctly positioned near the head note, without being located above head or to the next note because it is rather a staccato[3] dot but also notes that the group to which it belongs corresponds to a common group, and the number of beats in the measure is consistent with the metric. Not only test of graphic rules between neighboring primitives, but also take into consideration symbol groups to test syntactic rules. We see, therefore, through this example, the scoring rules are at different levels of interpretation: the graphic rules are applied directly on the primitive, while the syntactic rules are at a higher level of interpretation. However, these rules are complementary and applying one of them is not sufficient to confirm with certainty the presence of the point of time. Therefore, the fusion of all the information from the application of different rules is crucial, but difficult, because the information is very different in nature and are at different levels of interpretation.

The issue is on the one hand to detect, segment and recognize the primitives on most reliably base, which should be well defined, on the other hand, the best model a priori knowledge, graphic and syntactic, to disambiguate the extracted primitives and restore the high-level interpretation. The complexity of this task is on several levels:

- Ambiguity is important, because the printing defects, difficulty to segment coherent entities in the partition, and the variability of the primitives.
- This ambiguity is difficult to remove, because if the number of primitives is small, the number of arrangements of primitives, it is infinite.
- Rules notation for expressing are in most flexible and non-strict constraints or apply with varying degrees of accuracy.
- The scoring rules may involve a large number of symbols graphically distant from each other.
- Scoring rules are very different in nature, they are found at all levels of interpretation, however, are interdependent.

[2] Dotted note- In theory, the extension point is a sign placed after the figure of a note or rest, to extend the term of this figure

[3] Staccato-is a type of phrasing in which notes patterns and musical phrases must be performed with suspensions them.

7. Test and comparison of OMR systems

This section describe tests of 8 OMR systems. At the beginning are described all systems and then each of those system will be tested to recognized 21 different type of music sheet.

7.1. Description of systems

7.1.1. ”SharpEye 2” Music Scanning

Fig. 7.1.1.1: Logo of SharpEye 2

SharpEye 2 is a recognition program of musical characters. It converts scanned images of printed files to MIDI, MusicXML NIFF or just to play processed music. This is difficult task to achieve for a computer program, and there will be some errors if the music is complex or tight, or if the given image is of lower quality. SharpEye includes a music notation editor that was especially designed to control and correct the output before saving as a music file.

SharpEye 2 allows to scan via the TWAIN[4] interface. It is also possible to use another program to scan. SharpEye input must be saved in the TIFF or BMP format, and then load it into system. Producer inform user that scan must be made in black and white, and in resolution of 300 dpi for most music. Higher resolution may be preferable if the music is printed in small type. User must try scan as straight as possible.

After running SharpEye 2 program display 2 separate windows: Main window with all options and one with input image. After loading or scanning new image in the input image window user must select area of recognition by drawing working area around selected notation. After that user need only to click blue arrow in the main window to start recognition process.

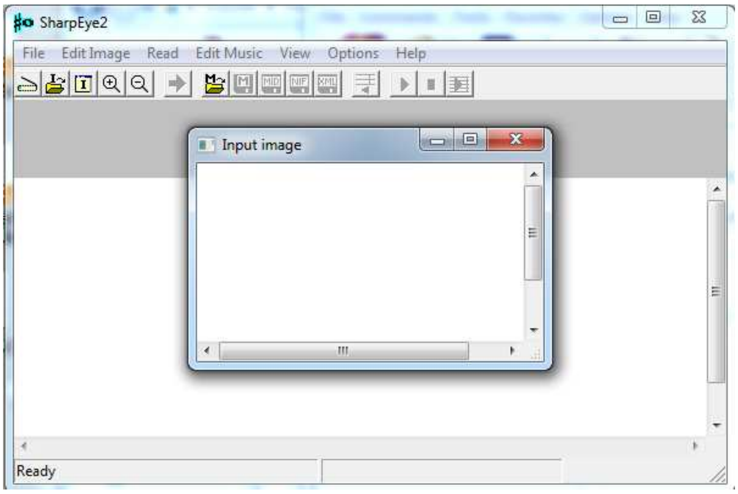


Fig. 7.1.1.2: Interface of SharpEye 2

[4] TWAIN is a standard protocol for computer primarily to link an image scanner to a computer, allowing any processing software to control TWAIN-compatible scanner to monitor and launch the image acquisition, irrespective of the model and manufacturer of the scanner.

SharpEye 2 allow user after recognition to edit and/or to correct result. Handling with errors are very easy and quick. If some note are recognized improper, user can modify its position by simple drag and drop on the correct stave line. If duration of given note are also wrong then program display possible changes so user need only to select proper value of selected note.

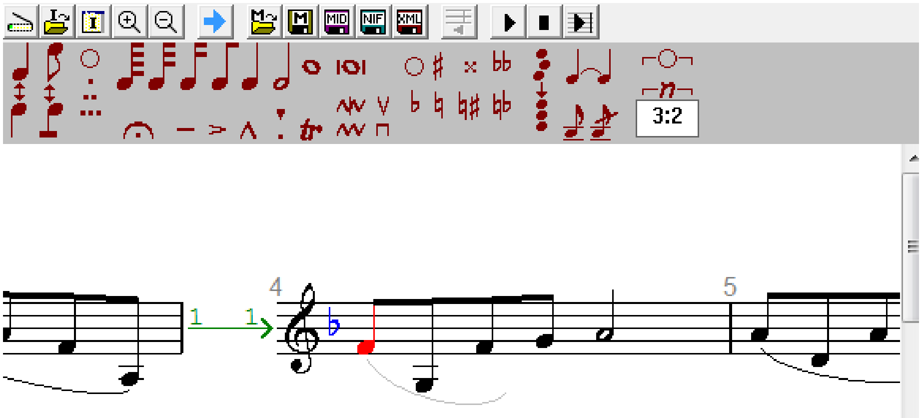


Fig. 7.1.1.3: Dialog window of notes edition; SharpEye 2

One of the very useful option of SharpEye is playing of recognized music notation as a MIDI file. This option is implemented with intention only for easier error tracking so if user need to change instrument or to process MIDI file, necessary is to use additional MIDI sequencer.

SharpEye 2 allow user to save recognized notation in one of the four file format: MIDI file for further edition in MIDI sequencers and for playing content of music sheet, NIFF and MusicXML that later can be used in score editor software's and SharpEye own file format (.mro) which allow to continue process after open unfinished file.

7.1.2. Capela- Scan 7

Fig. 7.1.2.1: logo of Capella- Scan 7

Capella is a publishing music notation software for Microsoft Windows, developed by Capella Software AG Söhrewald Germany. The software is available in German (original version), in English, Dutch, Finnish, and - with less current versions - Polish and Czech. The first version of the software was released in 1992. The main programmers are Hartmut Ring, retired math professor (since 2011) at the University of Siegen, and since 2008 Bernd Jungmann.

Inputting notes can be insert by the computer keyboard, the mouse, or by a musical keyboard with a MIDI interface. Capella offers export and import MIDI files and generic format MusicXML.

User can switch between the views of the full score and an extract of one or more voice, editing, presentation, printing and playing. Capella comes with six fonts musical notes and symbols for various styles and tablature for guitar. Hand tools allow to exceed the automatic placement and shape of the program notes for greater precision of reading.

Capella comes with a library of classes to extend the software with new features with programs written in Python. Documentation and the file format and programming interface is available on the website Capella software AG. Various functions are performed by Capella in such scripts. Using these expansion capabilities, the programmer Bernd Jungmann has created a capella interface for blind using a computer braille, these scripts are freely available on the Web. User can print sheets in Braille using a service of Deutsche Zentralbücherei für Blinde zu Leipzig (de), the central library for the blind in Germany in Leipzig.

In compare to SharpEye Capella on first sight looks much more complicated. The amount of provided options are greater than in SharpEye and most of the icons and pictogram are not so intuitive.

Fig. 7.1.2.2: Interface of Capella –Scan 7

However after getting more familiar with Capella user can notice that process of recognition are very similar like in SharpEye. At the beginning User must scan music sheet using scanner or simply load it from hard drive. Input file must be saved in tiff file format, however creators of this software though also about others formats by including additional file converter which automatically change selected input file into tiff.

After file is loaded user can start recognition by clicking F5 or selecting icon . Instead of creating new MIDI file, Capella draw on the original input file its own graphically represented notation. User by selecting given note can change it property using various of different options. It is also possible to delete selected musical symbols.

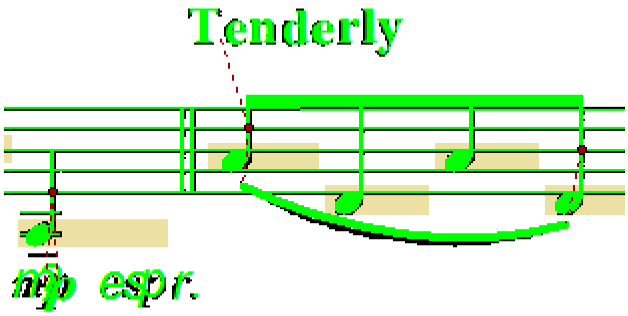


Fig. 7.1.2.3: Way of coloring recognized notes; Capella- Scan 7

Saving recognized and modifyied music sheet is as simple as load new file. By simple cliking on save icon user can save procesed file. Capella allow to store music sheet in three diferrent file format: MIDI, MusicXML and in capela score (.cap and .capx).

Because of big complexity of Capella functions at the beginning user can have problems with correct process of music sheet. Very useful feature of Capella is buldin Tutorial. This tutorrial include 6 lessons: “First Steps”, “Correct Recognition Error”, “Edit Text”, “Analyse Rhythm, Polyphony”, “Assign Voices, Tamplate System” and “Improve Rate of Recognition”. To each lessons are include sample music sheet and instuctions step by step how to finish given tasks.

7.1.3. OMeR 2.1

OMeR (Optical Music easy Reader) is the printed music score optical recognition add-on for Melody Assistant and Harmony Assistant. This add-on allow for scanning or loading music sheet. After conversion process file can by edit or listen in Melody or Harmony Assistant!

Fig. 7.1.3.1: Logo of OMeR

OMeR is a Shareware software in the miscellaneous category developed by Myriad Software. From an image of a partition, OMeR will locate music symbols and generate a .lfs document.

To run OMeR necessary is to first install SToccata Font (or Myriad Music Font for MAC operating system).

User interface of OMeR looks very poor because this add-on is created just for music recognition. There is only minimum option that are necessary to process music sheet into digital form.

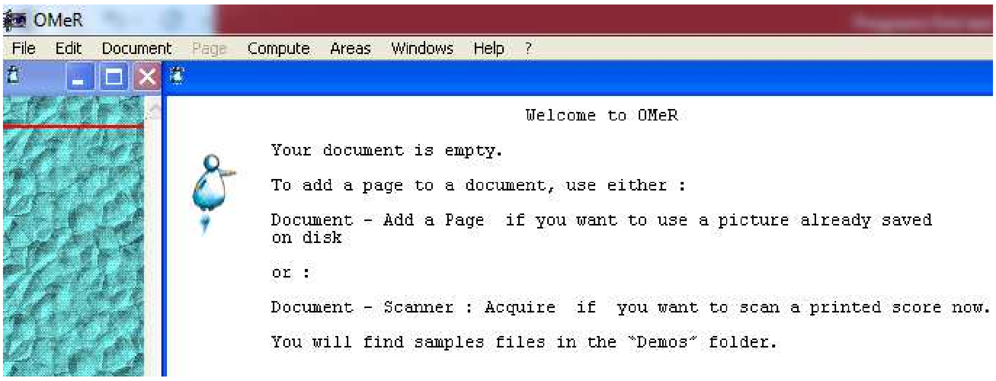


Fig. 7.1.3.2: Interface of OMeR System

OMeR didn't allow user to store result into common music file format. There is only possibility to store file in OMeR own file format. In compare to previous programs (SharpEye or Capella) OMeR have lack of such option like playing recognized notes to find errors or making correction using special editors. After computation of new file OMeR only draw music notes on the original input file in four different color. Green mean correctly computed symbol, orange- Adjusted symbol, red- Ignored symbol and purple- areas.

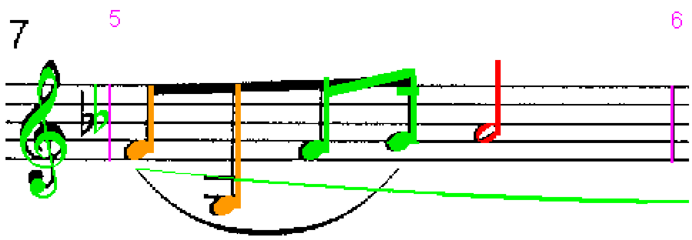


Fig. 7.1.3.3: Way of coloring recognized notes; OMeR

After first test of this program I notice that accuracy of OMeR recognition is quite poor. This program ignored many notes in my sample music sheet and because of missing edition function it is not possible to make recognized file better.

7.1.4. SmartScore X Pro

SmartScore X pro is a professional score writing software developed by Musitek Corporation. First version occur in 1991 as MIDISCAN for windows.

Fig. 7.1.4.1: Logo of SmartScore X Pro

SmarScore is one of most effective and powerful OMR system nowadays on the market. In addition to standard OMR options, this software include various of additional feature for music edition. Producers also claim that SmartScore accuracy is equal to +-98 percent.

After install and quick review of this program user shouldn't have any problems with operating this system despite the fact of big number of different icons and options.

Fig. 7.1.4.2: Interface of SmartScore X Pro

In compare to Capella software, SmartScore used in interface icons and pictogram with better metaphors of real life which helps user to better understand the software and more intuitive work.

The input file are generally all image file format. At the beginning of recognition process SmartScore present to user various of option. Besides selecting of input file user can also chose such option like:

- language of text included in music sheet,
- select option of recognition lyrics outside of system in case when music sheet contains choral lyrics
- select Join Offset Voice- in case when different voices appear horizontally offset,
- ask system to recognize tablature and percussion staff lines,
- If selected music sheet contain pedal marks SmartScore also can recognize them and add to result
- User also can select bowing marks if his music sheet contain them
- Ossia[5] recognition

SmarScore also include file converter which in easy way allow user to select pdf as a input file.

After music sheet is recognize SmartScore display to user “System Report” that include number of founded systems, how many pages include recognized sample, largest and smallest system with number of parts recognized. User at this moment can also select different instruments to recognized systems. If number of parts are not the same like in the original music sheet, SmartScore provide additional image editor which helps to improve image sample.

[5] A ossia is an alternative musical passage that can be played instead of the original. Ossia word comes from the Italian "o sia" meaning alternately. The Ossias are common in operas and solo piano pieces. In general, Ossias are easier than the original passage.

Image editor include the most important options needed to correct imperfections on music sheet scan. Some of the useful feature of image editor are:

- Delete unwanted part of score
- Manually draw brackets or bar lines
- Add or delete detail
- Correcting Skew
- Invert colors
- Draw staff line in case if original image miss line or some lines are low visibility
- Thresholding option. In case if note or line are very small or thin, darkening can improve recognition.

Thanks to build in image editor user don't need to use additional software to improve accuracy of reading music sheet.

After recognition process is finished User can play created midi file and correct notes if this is necessary. SmartScore provide also many additional options like adding to recognized file different rhythms by simple selecting from defined list. User may also manipulate with tempo and dynamics. Using staff properties it is also possible to convert normal 5 line staff into guitar TAB, bas TAB and also to different percussion line. In addition SmartScore provide lot of option connected with text and lyrics thanks to which user is able to create karaoke.

SmartScore allow for storing processed file into one of the tree standard file formats: MIDI, NIFF, MusicXML and SmartScore own file format with extension ".enf".

7.1.5. PhotoScore Ultimate 7

Fig. 7.1.5.1: Logo of PhotoScore Ultimate 7

PhotoScore Ultimate is the scanning tool scores for transposition, arrangement and edition. In a few seconds , PhotoScore Ultimate is able to recognized in addition to printed music sheet also handwritten music characters. It read from PDF , notes, links, shades, lyrics, guitar tablature, diagrams, legal agreements and more. PhotoScore Ultimate provides a range of feature such as a "browser rhythmic mistakes" to identify and correct errors in implementation rhythm or a search function and replacement handles error correction. PhotoScore Ultimate 7 instantly recognizes handwritten music, without resorting to a description of the script style. Version 7 features additional enhancements such as integrated care and improved PDF files, more accurate recognition of printed texts, faster, simple and multipurpose editing functions and a component optimized Pages. Some of the most interesting feature are as follow:

- Rhythmic mistakes browser
- Rewrite feature of the partition
- Search and replace functionality of automatic scanning and reading after treatment Printing
- Recognition of handwritten music
- Reading articulation marks
- Play triplets and other tuplets
- Reading notes and ornamental replica
- Reading notes and ligatures overlapping scopes
- Playing guitar chord diagrams
- Read guitar tabs with four strings
- Playing percussion staves 1, 2 and 3 lines
- Playing bars double recovery and triple
- Reading times end, the coda and references
- Reading marks pedal and ornaments
- Alteration types : 7
- Types of keys: 8
- Direct transposition in PhotoScore
- Recording audio files WAV / AIFF

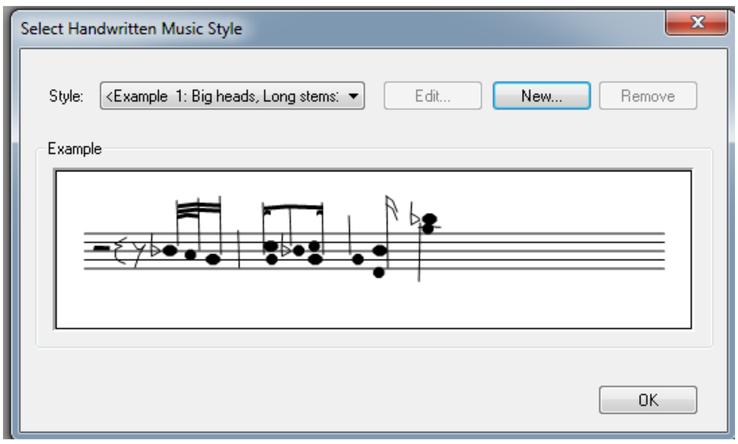


Fig. 7.1.5.2: Dialog window of Handwritten music style; PhotoScore Ultimate 7

In compare to other OMR systems PhotoScore is able to recognize handwritten music. To perform this user must first select handwriting style. There are 17 different possibilities:

1) basic,
2) big heads, Long stems
3) big heads, short stems
4) casual, heavy

5) casual, heavy, compact
6) casual, mixes heads
7) compact
8) Detached, loose
9) Jazz, casual
10) Jazz, smart
11) Oblique heads, casual
12) Oblique heads, smart
13) Small heads, long stems
14) Slanted steams, casual
15) Slanted steam, smart
16) Smart, slanted heads
17) Very smart

To each style there is assigned example so user can compare them with his own music sheet sample. It is also possible to create new style which is more suitable for the user. In option "Edit Handwritten music style" user can set different parameters for note head type, quarter rest type, pen type, size, spacing, slant and smartness. Such function distinguishes PhotoScore from other software.

Fig. 7.1.5.3: Interface of PhotoScore Ultimate 7

On the first contact with PhotoScore user receive very little amount of function. All icons and pictogram are clear and because of that controlling the system is very intuitive. Process of reading file is very easy. User can chose one of tree option: scan pages, open PDF's file or open image file. After selecting proper function PhotoScore do everything automatically and present to user recognized file which can be played or edited using mouse and additional toolbar. Transposition of whole score are also very easy and can be done by few click of mouse.

Saving recognized file are possible in MIDI, NIFF, MusicXML, PhotoScore own file format with extension ".opt" and also in wave audio file ".wav"

7.1.6. Audiveris

Fig. 7.1.6.1: Logo of Audiveris

Audiveris is currently the only OMR system in the "open source", which is constantly evolving. This has its pros and cons. The biggest advantage of Audiveris is its portability: it runs equivalent on Windows, Mac OS X and Linux. In addition, the source code is available to people outside of the development team and can be viewed and improved.

However, it is not easy with such OMR programs to achieve really good results like the commercial software's (i.e. SmartScore or PhotoScore). Very often it is quicker if you create a score right from the start and manually enters all notes into score editors systems.

Audiveris based process of recognition on system called Tesseract. Tesseract is a free software for optical character recognition (OCR). Originally developed as proprietary software from Hewlett -Packard between 1985 and 1995. After that it was never updated in the next decade. It was then released as open source in 2005 by Hewlett Packard and the University of Nevada, Las Vegas, and released with the Apache license , version 2.0. The development of Tesseract is currently sponsored by Google. Like all OCR programs , Tesseract serves to read the text from image, usually obtained by means of a scanner. The results are very good with regard to the recognition of characters , but lacks the ability to preserve the layout of the pages, for example, tables or columns. Initially restricted to ASCII characters, however in October 2011 Tesseract supports UTF-8 characters and recognizes 33 languages .

Because Tesseract is fully trainable it is possible to use it in OMR systems. Process of learning new characters is not easy however thanks to such feature user can teach software how to recognize notes, sharps, flats etc.

Developers of Audiveris instead of crating different installation file for different operating systems, decided to create lunching program in JNLP[6] file. To run Audiveris system necessary is also to install Ghostscript. GPL Ghostscript is a software package for processing the PostScript and PDF files. Ghostscript is responsible for Conversion between PDF and

[6] Java Network Launching Protocol (JNLP) is a file format associated with the Java Web Start technology. It is able to easily deploy Java applications from a simple web browser.
JNLP file describing the application and its dependencies and post a link to that file so that the system deploys the application directly. This requires, however, that Java is installed on the client site.

PostScript formats, Convert PDF to Images (png, jpeg, tiff, etc.) with configurable quality. Print files to PostScript format standard supported by most printers. Display Postscript / PDF files by using filters.

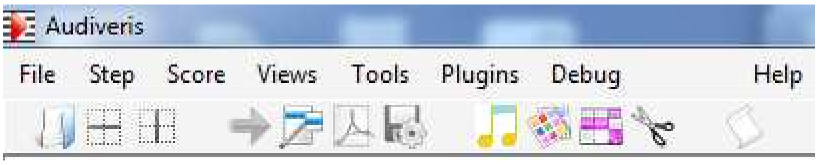


Fig. 7.1.6.2: Interface of Audiveris

After start, system display to user simple and small interface with only few options. Of course main purpose is to convert printed music sheet into digital form. In compare to previous OMR systems Audiveris have lack of scanning option. User can only load file from memory or using drag and drop method, move file into Audiveris application. Input file format are: PDF, TIFF, JPG, PNG and BMP. Transcribe (conversion into digital form) of notes are very simple and can be done in several way. Firs way is to select from menu "Score / Transcribe", second is to select blue arrow from the toolbar bellow menu, third option is to select from menu "Step/ SCORE".

After sheet are transcribe Audiveris display to user 4 main windows (panels).

1. Biggest window that contain our music sheet (original image + data extracted) with all notes and additional symbols colored in 6 different colors. Each color characterize different recognized symbol. Clef are in pink, key symbols dark blue, measure symbols in red, note head are orange and flag are in light blue. Stems line and staff lines are drawn as a thin lines. Audiveris also allow to select sheet display modes. There are three different mode:
    a. Physical mode- glyphs[7] are colorized based on recognized shapes and based on physical coordinates of those shapes.
    b. Logical mode- display logical score entities
    c. Combined mode- is a combination of both: physical and logical mode.
2. Boards panel describe vertical set of boards from the input image and from recognition results.
3. Events window display all log of the events that appeared from the moment of starting the Audiveris.
4. Error panel which show all detected by system error during process of recognition.

[7] A glyph is a graphical representation (among countless possible) a typographical sign, i.e. one character (character glyph) or an accent (accent glyph)

Transcribed data can be farther printed, played using additional plugins or export into MusicXML file.

The most interesting function of Audiveris is possibility to train OMR system with new symbols. Training are based on neural networks evaluation of provided samples. There are few methods of training.

1. Persistence of manual assignments- After turning this option on, user can manually select symbol from transcribed data set. If such symbol are recognized correctly it can be saved as a training sample.
2. Saving all score samples- If transcribed data didn't have any errors and user is sure that all symbols and notes are recognized correctly using this option is possible to save whole sheet as a training sample.
3. Sample verifier- This option is created to provide user interface for reviewing different glyphs that are used to training. Selection of glyphs are done in tree following steps:
    a. Selecting folder with training materials,
    b. Selecting shape that appeared in selected folder
    c. Selecting glyphs that are correspond to selected shape

   User can analyze selected glyphs and correct it if it's necessary. This will improve further recognition.
4. Trainer- Trainer option is created with purpose to training of the Neural Network evaluator. Training process have 4 stage:
    a. Selection- Selecting of glyph XML files.
    b. Neural Training- This panel is responsible for monitor and lunch of the training
    c. Validation- during and after training user can test evaluator
    d. Regression Training- This option compute regression of used parameters of linear evaluator.

7.1.7. VivaldiScan

Fig. 7.1.7.1: Logo of VivaldiScan

VivaldiScan is one of the products developed by goVivaldi Inc. and Allegroassai spa. Their produce software connected with music sheet editing. VivaldiScan derived from SharpEye system. According to developers by implementing new algorithms accuracy of VivaldScan is equal to 99 percent.

Recognition procedure and interface are very similar to SharpEye. There are some problem with input file format. I tried to scan and load this same image on all tested OMR software. Even that VivaldiScan is based on SharpEye it has problem with reading this same file. Error message inform me that TIFF image cannot by compressed using LZW[8] which is most famous compression type in TIFF file.

7.1.8. PDFtoMusic Pro

Fig. 7.1.8.1: Logo of PDFtoMusic PRO

PDFtoMusic Pro is software that was developed to direct conversion of PDF file into format that are accepted by music editing systems (MIDI, WAV, AIFF, MusicXML and KAR). In terms of music sheet saved using PDF file we can distinguish to types: file generated from the level of music editor software, sequencers etc. and pdf with scanned music sheet. Using PDFtoMusic only this first type can be correctly recognize. As producer describe PDFtoMusic this is not software for scanned music sheet.

After starting PDFtoMusic ask user to select one PDF file. System automatically starts data analysis. It analyze structure of PDF file, used fonts, line, number of staff and systems, text etc. All error that occur are listed in Error window. After finishing of whole process system display our document, tool bar above, on the right side miniature of pages. All error are depict by warning triangle.

Fig. 7.1.8.2: Interface of PDFtoMusic PRO

Software interface is very easy and intuitive, however more advanced option required more time to work out. PDFtoMusic allow user to play recognized file. To do this there are special buttons in the document window. User have also possibility to change instrument, set volume, stereo panorama and octave transposition of each voice. Useful function is “Staves and Systems” where user can split or merge staves

[8] LZW (Lempel-Ziv-to Welch) is an algorithm for lossless data compression. This is an improvement of the invented by Abraham Lempel and Jacob Ziv LZ78 algorithm in 1978. LZW was established in 1984 by Terry Welch

7.2. Tested software general information:

Tab. 7.2.1:Comparison of general information about tested OMR systems

Name	Publisher	Input File	Output File	Platform	Price	Size
SharpEye 2	Visiv	bmp, tif, scanner	MIDI, MusicXML, NIFF	Windows	169 US$	2,84 MB
Cappela- Scan 7	Capella Software	bmp, gif, pdf, png, PS, tif, scanner	Capella, MIDI, MusicXML	Windows	249.95 US$	104 MB
OMeR (Optical Music easy Reader) 2.1	Myriad	gif, jpg, pdf, png, tif, scanner	Melody/Harmony Assistant	Windows, Mac OS	25 US$	8,25 MB
SmartScore X2 Pro	Musitek	tif, scanner	Finale, MIDI, NIFF, MusicXML	Windows, Mac OS	399 US$	76,3 MB
PhotoScore Ultimate 7	Neuratron	bmp, pdf, scanner	MIDI, MusicXML, NIFF, WAV	Windows, Mac OS	€299	25,9 MB
Audiveris	open source	PDF, TIFF, JPG, PNG, BMP	MusicXML	Java	free	134 MB
VivaldiScan	goVivaldi	bmp, tif, scanner	Vivaldi, MusicXML, MIDI	Windows, Mac OS	€79	22,8 MB
PDFtoMusic Pro	Myriad	PDF	MIDI, WAV, AIFF, MusicXML and KAR	Windows, Mac OS	199 US$	22,5 MB

7.3. Test Samples

In this chapter I will test presented in previous section OMR systems. To check which of this software is better I selected 21 different samples of music sheet. In real word there are many way to present music notation (for example single stave voice, piano 2 stave voice, drum set stave voice …), that's why it is necessary to test if given OMR software is developed to recognize notation from different music sheet pattern or just to recognize simple one stave line voice. The test samples[9] are as follow:

0. Simple voice of violin music sheet where exist only one stave line voice. This sample is generated using music sheet writer software and it is save directly into pdf file. Because of that there is no additional noise in the sense of dust, curve staff line and any other noise which occur during scanning process. This sample also have limited additional symbols like lyrics or chords annotation. The density of characters are average (there is average number of notes in one measure).

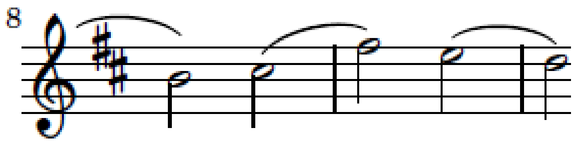


Fig. 7.3.1: Part of sample no 0

1. Sample with two different size of Staff. It happens very often that main voice is piano stave lines and in addition, above this stave line occur leading voice of other instrument with melody (in this example violin voice). Such music sheet helps to better accompany by pianist other musicians, however to distinguish piano voice from additional voice, piano stave lines are printed with bigger font size. Given sample have average density of character in one measure. The level of noise is small. In addition this sample contain changes of clef on single stave line which is additional challenge for OMR systems.

Fig. 7.3.2: Part of sample no 1

2. Simple piano sheet. One of the most common music sheet. Double (piano) stave line generated by music sheet writer software with average density of character in one measure. Lack of noise and additional character like lyrics.

---

[9] Whole sample can be found in attachment.

Fig. 7.3.3: Part of sample no 2

3. 4 hand piano music sheet. First two line in treble clef and 3rd and 4rd in bass clef. Given sample have small density of character in one measure and did not contain any noise.

Fig. 7.3.4: Part of sample no 3

4. Sheet with additional characters. In addition to music notation there existing also lyrics. Sample is generated so don't contain additional noise . density of character in one measure is low.

Fig. 7.3.5: Part of sample no 4

5. Low density of character music sheet with 4 different voices plus lyrics. There is lack of additional noise because sample was generated by music sheet writer software.
6. Sample of music sheet with notes on very height position (5 additional line above stave). This kind of music sheet is characteristic for example for flute voice. Sample didn't contain additional noise or characters. Density of notes in one measure is small.

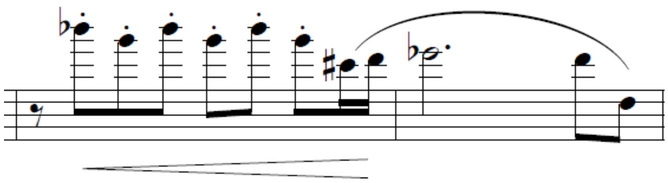


Fig. 7.3.6: Part of sample no 6

7. Score for big band: 15 different voices with different clef. 7 voices with bas clef, 7 with treble and one piano voice. This sample is generated, with average desity of character in one measure. It contain also additional character in the form of chord annotation.

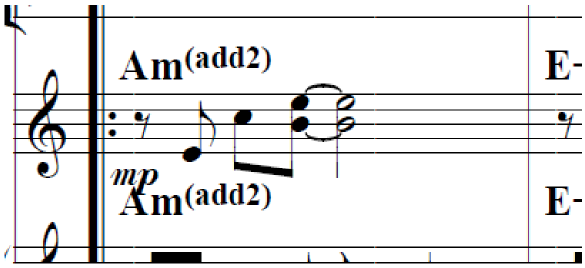


Fig. 7.3.7: Part of sample no 7

8. Orchestral score with small staff line and 17 voices. Typical orchestral score used by conductor of orchestra. Sample contain 17 different stave lines, density of notes in one measure is small, and there is no occurrence of noise.
9. Simple drum set music sheet. Typical voice of drum set with small density of character in one measure. Lack of noise and additional characters.

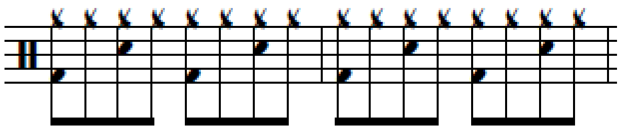


Fig. 7.3.8: Part of sample no 9

10. More complex drum set music sheet. More realistic because it contain lot of single measure repetition which is quite common in drum set scores. Sample have average density of character in one measure and don't have any noise.

Fig. 7.3.9: Part of sample no 10

11. One line percussion voice (snare drum). It happens very often that instead of whole stave lines composer use only one line. This is because some of instrument are used only for providing rhythms but no melody (percussion instruments). Density of

character in one measure in this sample is big. There occur additional character like tremolo.

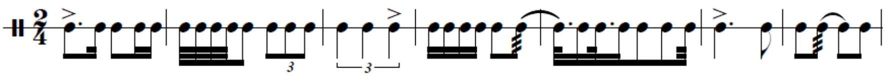


Fig. 7.3.10: Part of sample no 11

12. Bass voice plus bass TAB. To check I Tab lines didn’t interfere in recognition process I decided to test such sample. There is no additional noise and density of character are small. In addition this sample contain chords annotation.

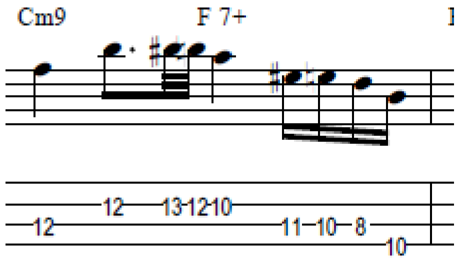


Fig. 7.3.11: Part of sample no 12

13. Scanned score with small amount of noises. This time score are not generated by computer but scanned. This sample contain some small noises in the form of dust. There is average density of character in on measure.

Fig. 7.3.12: Part of sample no 13

14. More complex drum set voice with full notes (without measure repetition). Sample is generated with additional characters characteristic for drum set music sheet (open hi-hat)

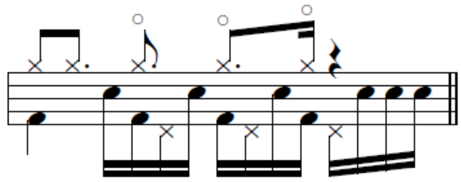


Fig. 7.3.13: Part of sample no 14

15. Scanned score with height amount of noises. The density of characters are average.

Fig. 7.3.14: Part of sample no 15

16. Scanned score with low amount of noises but shifted horizontally during scanning process.

Fig. 7.3.15: Part of sample no 16

17. Scanned score with low amount of noises but printed on yellow sheet. According to instruction of most OMR systems, music sheet should be printed in black and white to better recognition. I will check what will happen if this rule is omitted.

Fig. 7.3.16: Part of sample no 17

18. Hand wrote music sheet version 1. To check how OMR system deals with hand wrote music sheet I reaper two sheet written by to different persons (two different style of writing)

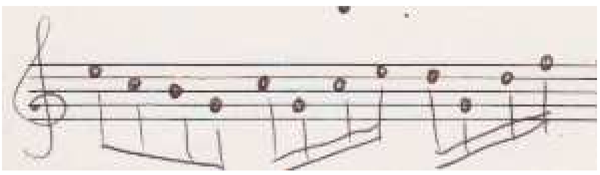


Fig. 7.3.17: Part of sample no 18

19. Hand wrote music sheet version 2.

Fig. 7.3.18: Part of sample no 19

20. Music sheet with two different size of notes but one size of stave lines.

Fig. 7.3.19: Part of sample no 20

7.4. Tests description

Tests are based on calculation how many basic symbols (notes and rests) will be recognized in each of those samples. Before all test was perform I calculated number of all notes and rests in each samples. Those two types of music symbols are most crucial for musicians to play music. Other symbols like dynamics, chords annotation, key change, meter, legato, etc. will be notice in comments or in tests summarize. Alteration symbols (sharps, flat, naturals) will be treated as a part of note. If given note will be recognized correctly but assigned to it alteration symbol will be omitted then such note will be count as a mistake.

Time of recognition was capture using two additional software:

- Free Screen Video Recorder
- Cool Timer 5.1.7.0

Some of the presented OMR systems display time of recognition process, however to obtain precise time from the moment of starting recognition until presenting results, whole process is recorded together with running timer. Based on analysis of recording it is possible to capture time of whole process with accuracy of millisecond.

Each sample was converted into tiff file format using “Convert PDF to Image” software. Properties of the output file was set into 100% of quality and 600 DPI resolution. Thanks to such settings OMR systems are able to precise recognize given sample.

After process of recognition was finished, number of correctly recognized characters was counted manually measure after measure. To decrees mistake of calculation each result was counted twice.

Result of all tests are presented in tables with 5 different columns:

- SN- Sample number
- ToR- Time of recognition (in seconds)
- TnoC- Total number of characters in the samples (rests and notes)
- CR- Correctly recognized characters
- Comments- Additional comment concerning recognition of given sample

7.4.1. Test of SharpEye 2

First tested system is SharpEye 2. Small software with very simple graphic user interface.

Tab. 7.4.1:Tables with results of testing SharpEye 2

SN	ToR	TnoC	CR	Comments
0.	12.11	351	351	In addition all repetition and legato characters was recognized correctly
1.	14.32	480	466	Program have problems with acciaccatura[10]. Mostly omit it or instead of two acciaccatura use only one normal note.
2.	9.90	256	254	Again problem with acciaccatura
3.	13.25	163	162	Again problem with recognition of acciaccatura
4.	31.01	375	339	Problem occur with additional voice written with smaller font. Another problem was to not recognized key change.
5.	12.57	179	155	System have problem with recognition of sixteenth notes flag.
6.	13.29	199	199	
7.	18.07	247	247	
8.	20.01	690	516	Not recognized key and some of the clefs. Problems with whole notes; not recognized drum set voice
9.	19.86	787	0	System didn't recognized drum set music sheet
10.	11.83	125	0	System didn't recognized drum set music sheet
11.	-----	92	-----	WARNING: System can't find any stave.
12.	6.63	98	83	Didn't recognized last stave with only one measure. Small problems with some single notes.
13.	27.12	261	100	Bass clef not recognized. Many single notes not recognized
14.	12.71	419	0	System didn't recognized drum set music sheet
15.	1:00.98	237	124	Though in the sample are a lot of noise, System recognized many symbols correctly
16.	12.44	138	40	System omits whole measures
17.	-----	439	-----	WARNING: System can't find any stave.
18.	25.85	191	0	System recognized only empty staves
19.	34.03	161	0	System recognized only empty staves
20.	14.21	204	88	System recognized only bigger font and omit everything written in smaller one.

The biggest problem of this system was to recognized stave with only one line. Such staves are used to represent some of percussion instrument which generate only one sound (for example tambourine, claves or snare drum). On this stave composer by using music symbols describe rhythm of such percussion instruments. Because of that system was created

[10] Small note appearing before a main note which not influence onto measure.

to recognized full five line stave, after testing given sample on the screen appeared warning about problem with finding stave.

Next problem was to recognized drum set music sheet. Such notation in addition to normal notes use also special type of characters similar to triangle, x-sign, rhombus, slash and many other. SharpEye 2 in return present only empty staves.

Some of the other problems was with wrong recognition of key.

Fig. 7.4.1.1: Problem with correct recognition of key; SharpEye2

In example above is depicted music sheet containing piano voice plus one additional violin voice. According to music rules this is not possible that one from three voice are written in different key.

OMR system also hade problem with recognition of smaller font in one music sheet.

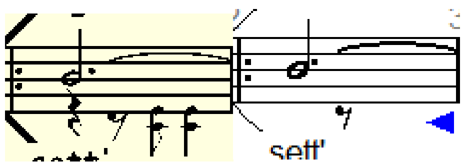


Fig. 7.4.1.2: Problem with correct recognition smaller notes; SharpEye2 (on left original image; on right result music sheet)

Picture above present this same measure, first one this is original voice and second one is result of recognition. Notes that are written using smaller font are omitted.

Despite SharpEye 2 had also other problems which are described in comment column, it also had some positive aspects. In most of cases legato, piano, lyrics and other character was recognized correctly. In sample number 15 where is lot of noises, this system recognized many of characters correctly.

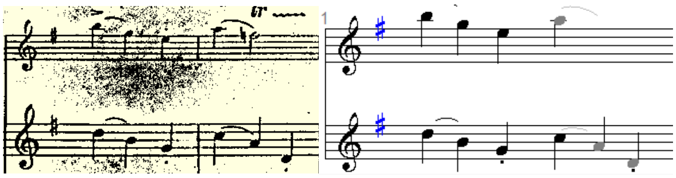


Fig. 7.4.1.3: Recognition of sample with lot of noise; SharpEye2 (on left original image; on right result music sheet)

System recognized correctly over 50 percent of measured character correctly.

7.4.2. Test of Capella-Scan 7

Next tested OMR system was Capella-Scan 7.

Tab. 7.4.2:Tables with results of testing Capella-Scan 7

SN	ToR	TnoC	CR	Comments
0.	8.71	351	348	Omits 2 notes; Shifted by a semitone higher on note
1.	8.07	480	30	On first stave omit one sharp in the key. Shifted whole staves in times which produces mistakes
2.	9.32	256	60	Because of acciaccatura was recognized as a normal notes whole music sheet was shifted by value of each acciaccatura.
3.	10.1	163	111	Shifts in time and missing some notes
4.	14.8	375	79	Shifts in time and missing some notes
5.	9.68	179	179	
6.	9.86	199	160	Shifts in time of last one and half stave
7.	11.16	247	137	Mistake with reading on of the flag shifted other notes in measure by one sixteenth note
8.	17.29	690	52	Shifts of almost all staves;
9.	10.53	787	0	System didn't recognized drum set music sheet
10.	10.27	125	0	System didn't recognized drum set music sheet
11.	-----	92	-----	WARNING: System can't find any stave or music symbols
12.	8.13	98	11	Shifts in time and missing some notes
13.	20.05	261	7	Shifts in time
14.	8.04	419	0	System didn't recognized drum set music sheet
15.	-----	237	-----	WARNING: System can't find any stave or music symbols
16.	-----	138	-----	WARNING: System can't find any stave or music symbols
17.	16.73	439	0	Shifts in time
18.	-----	191	-----	WARNING: System can't find any stave or music symbols
19.	-----	161	-----	WARNING: System can't find any stave or music symbols
20.	-----	204	-----	WARNING: System can't find any stave or music symbols

In compare to SharpEye 2, Cappella-Scan had bigger problems with recognition of stave. System didn't recognized stave on 6 samples. In addition to one line percussion stave, program didn't recognized stave or music symbols on hand written music sheet, sample with two different size of font and music sheet with turned stave horizontally.

Second biggest problem was shifting whole measures in time which cause dramatic results.

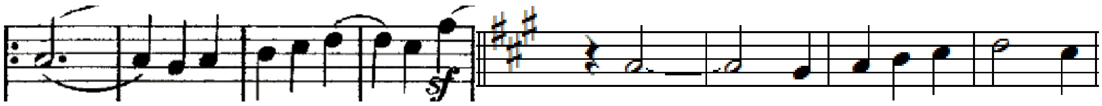


Fig. 7.4.2.1: Problem with shift whole measure in time; Capella-Scan 7 (on left original image; on right result music sheet)

In this figure all notes are shifted in time by one quarter note. Most of the notes are recognized correctly however all of them are on wrong position in the measure.

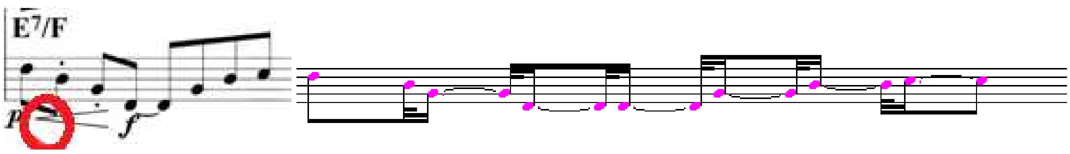


Fig. 7.4.2.2: Problem 2 with shift whole measure in time; Capella-Scan 7 (on left original image; on right result music sheet)

In above example is depicted wrong recognition of one flag. In original sample crescendo symbol are located to close to one eight note which case recognition of given note as thirty-second note. In result other notes are shifted in time. Such shifts case notation that are hart to read by human.

7.4.3. Tests of SmartScore X pro

Next system is most expensive from all tested. SmartScore X pro possess intuitive and simple interface which help to operate whole system.

Tab. 7.4.3:Tables with results of testing SmartScore X pro

SN	ToR	TnoC	CR	Comments
0.	9.75	351	351	
1.	6.65	480	446	Problems with smaller font
2.	5.97	256	153	Instead of acciaccatura system insert one whole note
3.	6.27	163	158	Wrong recognition of measure with notes very close to each other
4.	16.87	375	272	Problem with smaller font and whole notes
5.	4.86	179	179	
6.	8.68	199	197	System omit 1 note and 1 natural
7.	8.70	247	146	Instead of eight note system recognized quarter note
8.	10.71	690	618	System add unnecessary whole notes; Not recognized drums set special characters
9.	8.03	787	0	System wrongly recognized drum set notes
10.	5.31	125	21	System wrongly recognized drum set notes, however recognized measure repetition
11.	-----	92	-----	WARNING: Recognition failed
12.	5.82	98	52	Missing notes
13.	14.59	261	147	Missing notes or wrong interpretation
14.	7.97	419	0	System wrongly recognized drum set notes
15.	12:41:38	237	58	Too much noise disturb in recognition
16.	-----	138	-----	WARNING: Recognition failed
17.	21.51	439	293	Problem with smaller font and wrong interpretation of some notes
18.	16.21	191	0	System recognized only empty stave
19.	34.27	161	0	System recognized mostly empty stave and some notes without any meaning
20.	11.47	204	80	System omit all smaller notes

SmartScore X pro also didn't recognized single line staves. Another not recognized sample was sample with shifted stave. In samples with handwritten notes system recognized only empty staves.

In samples with two different size of fonts program correctly recognized only bigger font and smaller one treated us a noise.

Another problem was wrong interpretation of some flags.

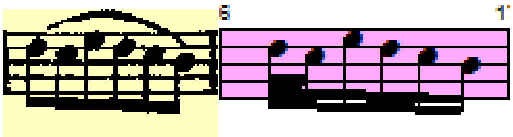


Fig. 7.4.2.1: Problem with interpretation of flag; SmartScore X pro (on left original image; on right result music sheet)

In above example we can notice that original sample contain only 6 eight note in one measure, however result of recognition present 2 thirty-second notes and 4 sixteenth note. This problem was caused by interpreting lower line from stave as a part of group of symbols. Such situation shouldn't take place because in the first step of OMR process stave lines usually are removed.

7.4.4. Tests of PhotoScore Ultimat 6

Another tested system is PhotoScore Ultimate 6. This is only system capable to recognition of handwriting notes.

Tab. 7.4.4:Tables with results of testing PhotoScore Ultimate 6

SN	ToR	TnoC	CR	Comments
0.	19.51	351	351	
1.	20.55	480	164	System recognized most of the notes in two different font
2.	17.43	256	256	System correctly recognized all notes together with acciaccatura
3.	19.35	163	163	System correctly recognized all notes together with acciaccatura and all additional symbols
4.	34.11	375	337	In one stave system omit change in key. Some of the smaller notes was omitted
5.	17.50	179	176	Wrong interpretation of flags
6.	18.85	199	198	
7.	24.44	247	246	One missing sharp
8.	39.58	690	571	Problem with drum set scores and incorrect interpretation of key
9.	26.70	787	0	System didn't recognized drum set music sheet
10.	15.07	125	0	System didn't recognized drum set music sheet
11.	3.95	92	58	System recognized single line staff.
12.	14.34	98	83	System didn't read last line of music sheet; System read TAB as a separate voice
13.	32.95	261	155	Noise and height density of symbols in one measure cause many mistakes
14.	18.00	419	0	System didn't recognized drum set music sheet
15.	1:06.12	237	115	Noise and height density of symbols in one measure cause many mistakes
16.	20.47	138	138	Automatic image preprocessing analyze angle of stave and read it correctly
17.	-----	439	-----	System didn't recognized staves
18.	1:55.63	191	38	Only few symbols was recognized correctly
19.	3:56.86	161	0	System didn't recognized any of notes correctly
20.	22.93	204	169	Mostly problem with smaller font

After selecting most appropriate handwriting style system recognized only few symbols correctly form 2 samples containing such music sheet. Recognition of those samples also took much longer time compare to recognition of other samples.

As advantage this is first from tested system that recognized one line stave.

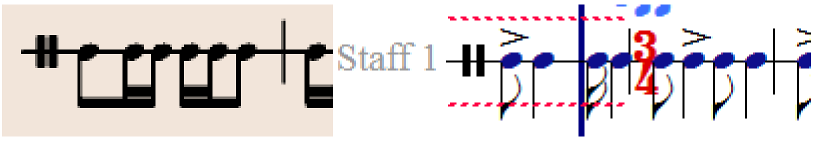


Fig. 7.4.4.1: Recognition of one line stave; PhotoScore Ultimate 6 (on left original image; on right result music sheet)

In sample number 12 which contain bass tablature (TAB) and stave, PhotoScore Ultimate 6 recognized not only stave but also separate voice based on TAB.

This system contain big database of additional symbols thanks to reading of recognized music sheet are easier.

7.4.5 Tests of OMeR

OMeR is the next tested system.

Tab. 7.4.5:Tables with results of testing OMeR

SN	ToR	TnoC	CR	Comments
0.	3.61	351	323	Wrong interpretation of some flag and missing some notes
1.	13.06	480	87	Wrong meter (instead 3/4 system recognized 4/4); Big number of mistakes
2.	3.70	256	245	Good recognition of acciaccatura. Some problems with flag and alterations symbols
3.	4.41	163	134	System omit single notes
4.	9.76	375	322	Problem occur with additional voice written with smaller font, some of alteration symbols and whole notes
5.	3.80	179	173	Wrong key in 2nd measure; problems with flags
6.	3.83	199	151	Wrong interpretation of some flag and alteration symbols
7.	5.31	247	36	System omit some measures and shift many measures in time
8.	7.16	690	0	System shift in time many measures. Instead of original 9 measure in line system create 46 measures.
9.	3.62	787	0	System didn't recognized drum set music sheet
10.	-----	125	-----	System stop recognition because didn't found any music notation
11.	-----	92	-----	System stop recognition because didn't found any music notation
12.	4.17	98	81	Didn't recognized last stave with only one measure. Small problems with some single notes.
13.	-----	261	-----	System stop recognition because didn't found any music notation
14.	3.66	419	0	System didn't recognized drum set music sheet
15.	30.80	237	13	System omit one from 3 stave and recognized only some single notes.
16.	10.37	138	6	Incorrectly recognized notes from many different measures overlap each other without any logic
17.	-----	439	-----	System stop recognition because didn't found any music notation
18.	-----	191	-----	System stop recognition because didn't found any music notation
19.	-----	161	-----	System stop recognition because didn't found any music notation
20.	5.48	204	21	Measures are shifted in time. Many notes were omitted. Only single notes was recognized correctly.

OMeR don't recognize 6 samples. On the screen only appear alert that "System stop recognition because didn't found any music notation"

In sample that this system recognized was many shifted in time. Such shifts unable recognized file to be read by human. In addition many of recognized symbols was wrongly interpreted.

Process of recognition took very short time, however result was not satisfying.

7.4.6. Tests of AudiVeris

Audiveris is the only open source software. User don't have to pay for it, however system didn't give good results.

Tab. 7.4.6:Tables with results of testing AudiVeris

SN	ToR	TnoC	CR	Comments
0.	18.36	351	318	System change key where is no any change. Didn't recognized some of half note. Wrong interpreted some of notes.
1.	-----	480	-----	System return alert about error. System not recognized any music symbols in given sample.
2.	13.13	256	149	Input some unnecessary rest. Not recognized acciaccatura
3.	18.23	163	91	System didn't change key, Not recognized first four line and had problem with interpretation of many notes
4.	31.77	375	250	Only part of all notes was recognized correctly
5.	15.81	179	179	
6.	17.57	199	191	Not recognized some of alteration and notes
7.	-----	247	-----	System return alert about error. System not recognized any music symbols in given sample.
8.	1:06.85	690	492	System didn't recognized drum set music notations. Wrong recognized key and meter
9.	30.03	787	0	System didn't recognized drum set music notations
10.	22.64	125	0	System didn't recognized drum set music notations
11.	-----	92	-----	WARNING: System can't find any stave.
12.	16.03	98	0	To many notes are overlap each other which case problem with reading.
13.	-----	261	-----	System return alert about error. System not recognized any music symbols in given sample.
14.	17.70	419	0	System didn't recognized drum set music notations
15.	-----	237	-----	System return alert about error. System not recognized any music symbols in given sample.
16.	28.68	138	0	System recognized only empty staves
17.	-----	439	-----	System return alert about error. System not recognized any music symbols in given sample.
18.	-----	191	-----	System return alert about error. System not recognized any music symbols in given sample.
19.	-----	161	-----	System return alert about error. System not recognized any music symbols in given sample.
20.	-----	204	-----	System return alert about error. System not recognized any music symbols in given sample.

The time of recognition of each sample was long. System didn't recognized 9 from 21 sample. In 5 sample system didn't recognized correctly any of notes.

It look that system can correctly recognized only simple notes and without any noise.

In many samples Audiveris made mistakes.

Fig. 7.4.6.1: Problem with interpretation of notes; Audiveris (on left original image; on right result music sheet)

Some of those mistakes concern wrong interpretation of flag or like in example above unreasonable change of meter.

7.4.7. Tests of VivaldiScan

VivaldiScan derived from SharpEye system. Graphic interface is similar but contain some additional features. Vivaldi scan is cheaper from SharpEye 2 however accuracy of recognition are similar.

Tab. 7.4.7:Tables with results of testing VivaldScan

SN	ToR	TnoC	CR	Comments
0.	7.42	351	351	
1.	9.08	480	458	Program have problems with acciaccatura
2.	5.85	256	150	Not recognized acciaccatura
3.	7.13	163	158	Omit some notes and acciaccatura
4.	15.68	375	290	Not recognized key change, additional voice written with smaller font and some of the whole note
5.	6.62	179	155	System have problem with recognition of sixteenth notes flag.
6.	6.48	199	198	Omit one flat
7.	9.34	247	245	Omit one sharp and one note was shifted by semitone
8.	13.24	690	496	Key not recognized, missing quarter rests, half and whole notes
9.	10.16	787	0	System didn't recognized drum set music notations
10.	6.06	125	0	System didn't recognized drum set music notations
11.	-----	92	-----	WARNING: System can't find any stave.
12.	3.53	98	82	Didn't recognized last stave with only one measure plus problems with some of notes
13.	15.59	261	86	Many single notes not recognized
14.	7.25	419	0	System didn't recognized drum set music notations
15.	40.73	237	69	System omit many of measures in each stave. Many of notes was wrong recognized
16.	7.67	138	0	Measures was mixed together and notes was shifted in time.
17.	17.29	439	265	System didn't recognized smaller font and wrongly interpreted some of the notes
18.	18.72	191	0	System recognized only empty staves
19.	21.09	161	0	System recognized only empty staves
20.	8.94	204	74	System didn't recognized smaller notes and also made many mistakes with bigger notes

VivaldiScan is faster than SharpEye but made more mistakes. VivaldiScan omit many of symbols.

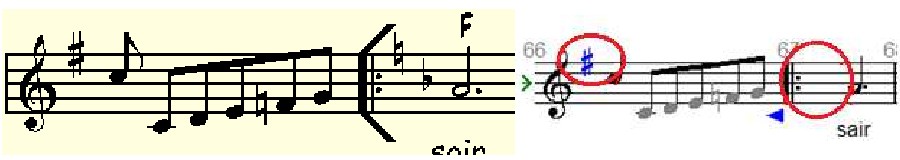


Fig. 7.4.7.1: Problem with omitting some notes; VivaldiScan (on left original image; on right result music sheet)

In above example is depicted how VivaldiScan omit key change. Another problem was with recognition of sixteenth note flags:

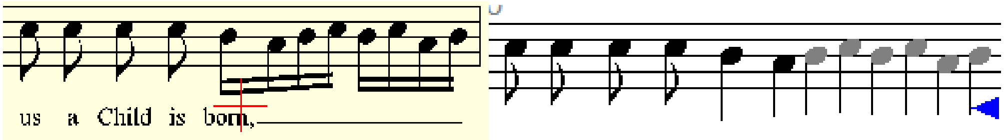


Fig. 7.4.7.2: Problem with interpretation of flag; VivaldiScan (on left original image; on right result music sheet)

As advantage VivaldiScan suggest to user where can be mistake by adding blue arrow near errors. By using of special feature in few step most of mistakes can be repair.

7.4.8 Tests of PDFtoMusic PRO

PDFtoMusic PRO this is software developed with purpose to recognized file generated from the level of music editor software but not scanned document. From all tested OMR systems PDFtoMusic PRO is the only software that use original PDF file instead their TIFF converted version.

Tab. 7.4.8:Tables with results of testing PDFtoMusic PRO

SN	ToR	TnoC	CR	Comments
0.	6.32	351	351	
1.	-----	480	-----	System display: “Sample can’t be processed because it doesn’t include any exploitable music data”
2.	7.41	256	170	System omit many notes
3.	6.86	163	158	Problem with some alteration symbols and acciaccatura
4.	7.55	375	38	System incorrectly recognized most of the flag.
5.	7.66	179	179	Build in player use voice simulation to sing lyrics according to music notation
6.	6.05	199	199	
7.	16.98	247	124	System omit one measure (in the middle of all staves) and incorrectly recognized some of alteration symbols.
8.	18.43	690	673	System don’t recognized correctly few key and some alteration symbols
9.	5.10	787	757	System recognized most of the drum set notation symbols
10.	5.37	125	67	System recognized most of the drum set notation symbols and measure repetition
11.	5.18	92	92	
12.	4.88	98	88	Didn’t recognized last stave with only one measure and put 2 rests in wrong place.
13.	-----	261	-----	System display: “Sample can’t be processed because it doesn’t include any exploitable music data”
14.	5.19	419	214	System omit many measures
15.	-----	237	-----	System display: “Sample can’t be processed because it doesn’t include any exploitable music data”
16.	-----	138	-----	System display: “Sample can’t be processed because it doesn’t include any exploitable music data”
17.	-----	439	-----	System display: “Sample can’t be processed because it doesn’t include any exploitable music data”
18.	-----	191	-----	System display: “Sample can’t be processed because it doesn’t include any exploitable music data”
19.	-----	161	-----	System display: “Sample can’t be processed because it doesn’t include any exploitable music data”
20.	-----	204	-----	System display: “Sample can’t be processed because it doesn’t include any exploitable music data”

Because this system is not designed to recognize scanned document, during tests it didn't recognized 8 samples. As a response system display information that "Sample can't be processed because it doesn't include any exploitable music data".

Samples generated by music software was however process with good accuracy of recognition. PDFtoMusic PRO made some simple mistake like omitting single notes and alteration symbols or mistake position of rest and notes in measure:

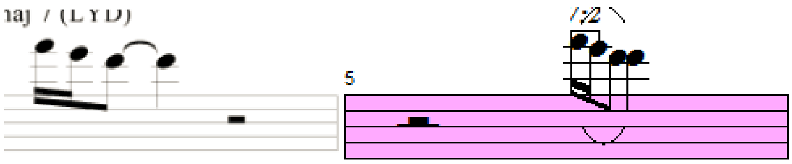


Fig. 7.4.8.1: Problem with changed position of notes and rests; PDFtoMusic PRO (on left original image; on right result music sheet)

System also recognized most of the drum set notation. In addition in sample with choir voices (sample 5), PDFtoMusic PRO use voice imitation to read lyrics and sing according to music sheet.

7.5. Tests results

After all tests are done it is possible to set together all results and see where OMR systems had problems, how fast they are and what kind of music sheet was well recognized.

7.5.1 Sample recognition accuracy

At the beginning let see which of 21 samples was problematic and which one was detected without any bigger problems. The table below include 4 columns:

1. Column "SN" this is number of tested sample
2. "TnoC" means total number of character in sample
3. "AVGnoC" represent average number of correctly recognized character by OMR systems
4. "PerOR" this is column that show in percent how many character was recognized in each sample

Tab. 7.5.1: Overall samples recognition

SN	TnoC	AVGnoC	PerOR
0.	351	343	97,72%
1.	480	206,375	42,99%
2.	256	179,625	70,17%
3.	163	141,875	87,04%
4.	375	240,875	64,23%
5.	179	171,875	96,02%
6.	199	186,625	93,78%
7.	247	147,625	59,77%
8.	690	427,25	61,92%
9.	787	94,625	12,02%
10.	125	11	8,80%
11.	92	18,75	20,38%
12.	98	60	61,22%
13.	261	61,875	23,71%
14.	419	26,75	6,38%
15.	237	47,375	19,99%
16.	138	23	16,67%
17.	439	69,75	15,89%
18.	191	4,75	2,49%
19.	161	0	0,00%
20.	204	54	26,47%

From this table it is visible that over half of the samples are recognized in less than 25 percent. The biggest problem occur in two samples with handwriting music sheet. First one was recognized only in 2,49 % and second one wasn't recognized at all.

Other problem appear during recognition of drum set music sheet. Most of the tested system were not created with purpose to read such notes. In compare to standard music sheet, drum set notation contain many of additional characters and density of character in one measure is much larger. From the other hand Drum set notation are indifferent for any alteration sign, change of key and clefs. Usually most of measures are similar that's why OMR systems shouldn't have problems with recognition of such music sheet.

Another problem was recognition of one line stave that are used to describe many of percussion instruments. Because that in most of OMR system first part of recognition process is stave line detection where system try to find all stave based by horizontal projection and regular distance between each line, such systems omit single lines recognized them us a not relevant in this process set of characters. Developers of such systems didn't consider different staves from normal 5 line.

During recognition of scanned sample with different amount of noises, rotated horizontally during scanning process or printed on yellow sheet appear another problems. Not all of tested software was able to detect correctly angle of stave and big number of noises cause wrong interpretation of many characters.

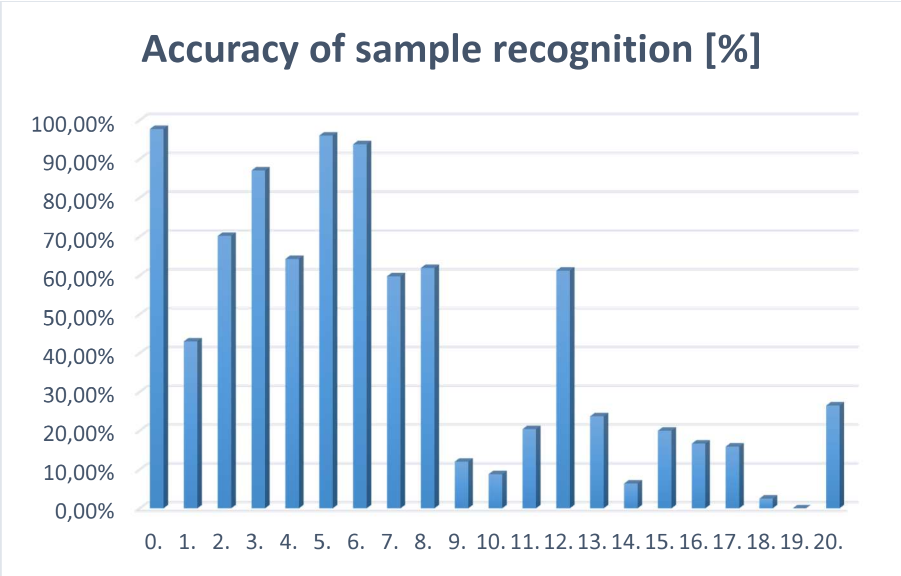


Diagram 7.5.1: Accuracy of sample recognition [%]

According to above statistics only 5 from 21 samples was recognized in over 70 percent. That was sample generated by computer (not scanned) with simple characters. From those five sample the worst result was obtained from sample 2. That's because this sample have two different size of stave: piano voice written with bigger size and main violin voice. Such types of music sheet are used very often to help pianist better accompany to other musicians. In this example most of the notes written with bigger font was recognized correctly, however on smaller font many of character was omitted or interpreted incorrectly. With other four samples most of OMR systems didn't had any problems. Usually problem that appear was just omitting singles notes.

7.5.2. Time of recognition

In the table below is described how many times (in seconds) was necessary to recognized given sample. Time of all OMR systems was set together and from each row was selected shortest and longest time of recognition. The cell shaded on red means the longest time spend on recognition process and cell shaded on green shortest time.

Explanations of column header:

1. SN- Sample number
2. #Eye- SharpEye 2
3. Ca-Sc- Capella- Scan 7
4. SmScX- SmartScore X Pro
5. PhScU- PhotoScore Ultimate 7
6. OMeR- OMeR (Optical Music easy Reader) 2.1
7. AuVer- Audiveris
8. VivScn- VivaldiScan
9. PDF2M- PDFtoMusic Pro
10. TOTAL- Sum of time
11. AVG/Samle- Average time spend on one sample

Tab. 7.5.2:Time of recognition

SN	#Eye	Ca-Sc	SmScX	PhScU	OMeR	AuVer	VivScn	PDF2M
	T[s]	**T[s]**	**T[s]**	**T[s]**	**T[s]**	**T[s]**	**T[s]**	**T[s]**
0.	12.11	8.71	9.75	19.51	3.61	18.36	7.42	6.32
1.	14.32	8.07	6.65	20.55	13.06	-----	9.08	-----
2.	9.90	9.32	5.97	17.43	3.70	13.13	5.85	7.41
3.	13.25	10.1	6.27	19.35	4.41	18.23	7.13	6.86
4.	31.01	14.8	16.87	34.11	9.76	31.77	15.68	7.55
5.	12.57	9.68	4.86	17.50	3.80	15.81	6.62	7.66
6.	13.29	9.86	8.68	18.85	3.83	17.57	6.48	6.05
7.	18.07	11.16	8.70	24.44	5.31	-----	9.34	16.98
8.	20.01	17.29	10.71	39.58	7.16	1:06.85	13.24	18.43
9.	19.86	10.53	8.03	26.70	3.62	30.03	10.16	5.10
10.	11.83	10.27	5.31	15.07	-----	22.64	6.06	5.37
11.	-----	-----	-----	3.95	-----	-----	-----	5.18
12.	6.63	8.13	5.82	14.34	4.17	16.03	3.53	4.88
13.	27.12	20.05	14.59	32.95	-----	-----	15.59	-----
14.	12.71	8.04	7.97	18.00	3.66	17.70	7.25	5.19
15.	1:00.98	-----	12:41:38	1:06.12	30.80	-----	40.73	-----
16.	12.44	-----	-----	20.47	10.37	28.68	7.67	-----
17.	-----	16.73	21.51	-----	-----	-----	17.29	-----
18.	25.85	-----	16.21	1:55.63	-----	-----	18.72	-----
19.	34.03	-----	34.27	3:56.86	-----	-----	21.09	-----
20.	14.21	-----	11.47	22.93	5.48	-----	8.94	-----
TOTAL	6:10,19	2:52,74	16:05,02	13:10,34	1:52,77	5:02,8	3:57,87	1,42,98
AVG/sample	19,48	11,52	50,79	39,52	7,52	25,23	11,89	7,92

In result system which spend longest time on recognition process of all sample was SmartScore X Pro with total time of over 16 minutes and system with smallest time was PDFtoMusic Pro with total time loess then 2 minutes. However if we look closer to each row it is possible to see that in 13 samples the worst time obtain PhotoScore Ultimate 7. Similar situation is with shortest time. On 10 sample best time was obtained by OMeR system.

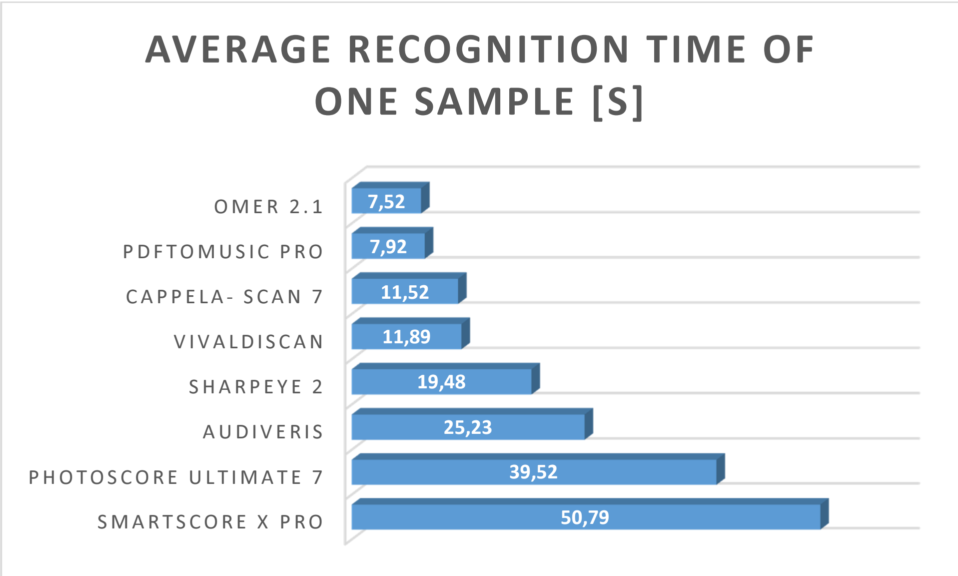


Diagram 7.5.2: Average recognition time of one sample

Diagram above depict the average time needed to recognized one sample by tested OMR systems. Fastest OMR systems proved to be OMeR 2.1 and PDFtoMusic PRO with time less than 8 second per sample. Systems that needed most time to recognized each sample was PhotoScore Ultimate 7 and SmartScore X PRO with time 39.52 and 50.79 seconds per sample. In the next step of tests analysis, those results should be compared with information's about accuracy of each tested OMR systems.

7.5.3 Accuracy of OMR Systems

In the diagram below are described percent of recognized character from all 21 samples.

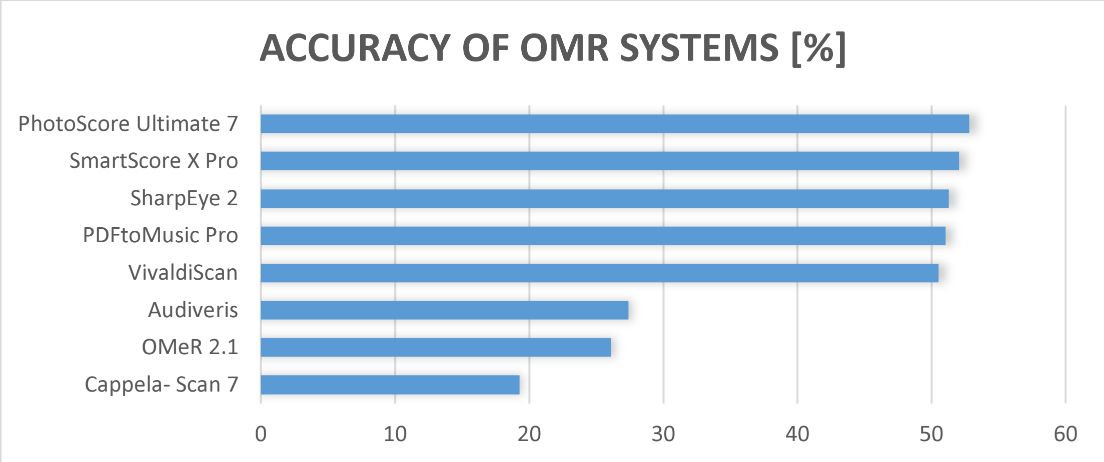


Diagram 7.5.3: Accuracy of OMR systems

If we compare those result with time of recognition is seen that short time of recognition often didn't mean good recognition of music sheet. OMeR 2.1 appear to be quickest system, however its accuracy are on the level of 26.13% and it recognized only 15 samples. PhotosScore Ultimate 7 with level 52.82% of recognized samples seems to be the best system from all tested. Table 7.5.3 contain information about accuracy of recognition of each OMR system (how many characters was identified correctly in each sample). The cell shaded on red means the lowest recognition rate and cell shaded on green highest rate.

Tab. 7.5.3:Recognition accuracy of OMR systems

SN	TnoC	#Eye		Ca-Sc		SmScX		PhScU		OMeR		AuVer		VivScn		PDF2M	
		CR	%	CR	%	CR	%	CR	%	CR	%	CR	%	CR	%	CR	%
0.	351	351	100,00%	348	99,15%	351	100,00%	351	100,00%	323	92,02%	318	90,60%	351	100,00%	351	100,00%
1.	480	466	97,08%	30	6,25%	446	92,92%	164	34,17%	87	18,13%	-----	-----	458	95,42%	-----	-----
2.	256	254	99,22%	60	23,44%	153	59,77%	256	100,00%	245	95,70%	149	58,20%	150	58,59%	170	66,41%
3.	163	162	99,39%	111	68,10%	158	96,93%	163	100,00%	134	82,21%	91	55,83%	158	96,93%	158	96,93%
4.	375	339	90,40%	79	21,07%	272	72,53%	337	89,87%	322	85,87%	250	66,67%	290	77,33%	38	10,13%
5.	179	155	86,59%	179	100,00%	179	100,00%	176	98,32%	173	96,65%	179	100,00%	155	86,59%	179	100,00%
6.	199	199	100,00%	160	80,40%	197	98,99%	198	99,50%	151	75,88%	191	95,98%	198	99,50%	199	100,00%
7.	247	247	100,00%	137	55,47%	146	59,11%	246	99,60%	36	14,57%	-----	-----	245	99,19%	124	50,20%
8.	690	516	74,78%	52	7,54%	618	89,57%	571	82,75%	0	0,00%	492	71,30%	496	71,88%	673	97,54%
9.	787	0	0,00%	0	0,00%	0	0,00%	0	0,00%	0	0,00%	0	0,00%	0	0,00%	757	96,19%
10.	125	0	0,00%	0	0,00%	21	16,80%	0	0,00%	-----	-----	0	0,00%	0	0,00%	67	53,60%
11.	92	-----	-----	-----	----	-----	-----	58	63,04%	-----	-----	-----	-----	-----	-----	92	100,00%
12.	98	83	84,69%	11	11,22%	52	53,06%	83	84,69%	81	82,65%	0	0,00%	82	83,67%	88	89,80%
13.	261	100	38,31%	7	2,68%	147	56,32%	155	59,39%	-----	-----	-----	-----	86	32,95%	-----	-----
14.	419	0	0,00%	0	0,00%	0	0,00%	0	0,00%	0	0,00%	0	0,00%	0	0,00%	214	51,07%
15.	237	124	52,32%	-----	-----	58	24,47%	115	48,52%	13	5,49%	-----	-----	69	29,11%	-----	-----
16.	138	40	28,99%	-----	-----	-----	-----	138	100,00%	6	4,35%	0	0,00%	0	0,00%	-----	-----
17.	439	-----	-----	0	0,00%	293	66,74%	-----	-----	-----	-----	-----	-----	265	60,36%	-----	-----
18.	191	0	0,00%	-----	-----	0	0,00%	38	19,90%	-----	-----	-----	-----	0	0,00%	-----	-----
19.	161	0	0,00%	-----	-----	0	0,00%	0	0,00%	-----	-----	-----	-----	0	0,00%	-----	-----
20.	204	88	43,14%	-----	-----	80	39,22%	169	82,84%	21	10,29%	-----	-----	74	36,27%	-----	-----
TOTAL	6092	3124	51,28%	1174	19,27%	3171	52,05%	3218	52,82%	1592	26,13%	1670	27,41%	3077	50,51%	3110	51,05%

Explanations of column header of above table:

1. SN- Sample number
2. #Eye- SharpEye 2
3. Ca-Sc- Capella- Scan 7
4. SmScX- SmartScore X Pro
5. PhScU- PhotoScore Ultimate 7
6. OMeR- OMeR (Optical Music easy Reader) 2.1
7. AuVer- Audiveris
8. VivScn- VivaldiScan
9. PDF2M- PDFtoMusic Pro
10. TOTAL- Total number of recognized characters
11. CR- Correctly recognized characters
12. %- Correctly recognized characters [in percent]
13. TnoC- Total number of characters in sample.

8. Conclusion

Writing this thesis has enabled me to step into the world of OMR, optical music recognition. When I chose this topic I thought it to be very similar approach to the OCR (Object character recognition). In fact I quickly realized that the music score, even when it was generated by music software and written in a more or less standard way, was causing many challenges. First of all in compare to OCR systems, OMR have additional dimension. Where in OCR text occur in sequence of symbols written one after another in music sheet recognition appear another dimension. To horizontal dimension of symbols by positioning those symbols on the staff lines, music sheet represent vertical dimension (pitch). Before music sheet is recognized by use of OMR systems first whole sheet must go through image preprocess to eliminate noises caused by additional unimportant text, chords annotation or even by stave lines. Staves seems to be very important aspect in music recognition because without it human being are not able to play given notation correctly. In fact this what is necessary for human it's only obstacle for computer. However removing staff lines case another problems. Characters that are written on the stave are in contact with those 5 lines and after removing those line system also can remove some part of necessary character like notes, alteration or rests. Because of that farther part of recognition process can incorrectly interpret those characters. Another big challenges are various of different music rules which are very flexible and have many exceptions. Even small number of music characters cause many problems cause they can be join together to obtain another character.

Problems which caused those (and many more) challenges are seen in my test results. Most of tested OMR systems are commercial and very expensive software's. User need to pay lot of money to use them but in result he can correctly recognized only music sheet without any additional noises, generated by other music software. There are different type of music sheet. Choir use different then pianist, drummers use different music sheet then bass players and so on. That's why in my tests I selected 21 different music samples which will help given group of musicians to choose OMR system that will best fulfill their needs.

Based on obtained results 3 OMR systems are very interesting and worth of use. First one this is PDFtoMusic PRO which appear to be second fastest system in recognition and one from the four systems that recognized over 50% of characters correctly. However there is one disadvantage of this system. Because this system was created with purpose of recognition generated music sheet it has problem with scanned document. Second software this is PhotoScore Ultimate 6. This system obtain best result of recognition and recognized 20 samples. It include many interesting feature and its training set contain many characters. Problem of this system was only time of recognition which appear to be the longest, however if for user more important is recognition accuracy then time this software can be very useful. Third option this is SharpEye 2. Small and very simple in use software that recognized over 50 % of character from 19 samples.

Generally most of tested OMR systems hadn't to much problem with recognition of generated by computer music sheet. Such samples was recognized in over 90%. Biggest problem seems to be handwriting notes and drum set notation. Single stave percussion line was recognized only by 2 systems and scanned document with many additional noises was recognized only in parts. Most of tested OMR systems allow user to correct wrongly recognized or interpreted characters. Some of those system even suggest where can be mistake, however correction of notes which was recognized in les then 30 % can induce of spending big amount of time on such correction. In such situation fastest way to obtain digital music sheet can by writing them from the beginning using music sheet writing software's.

9. Attachments

Fig. 9.1:Sample no "0"; "Sechs Sonata fur Violine" J.S. Bach

Partita III BWV 1006

"Sechs Sonaten für Violine"

Johann Sebastian Bach (1685-1750)

3. Gavotte en Rondeau

BWV 1006

Fig. 9.2:Sample no "1"; "Sonata No. 6 in A Major, Op. 30, No. 1" L. Beethoven

Fig. 9.3:Sample no "2"; "Rondo Alla Turca" W.A.Mozart

1

Rondo Alla Turca

The 3rd. movement from Sonata K.331

W.A.Mozart

Fig. 9.4:Sample no "3"; "Fantasia" W.A. Mozart

Fig. 9.5:Sample no “4”; “Une valse musette” Philippe Berdif

Fig. 9.6:Sample no "5"; "For unto Us a Child is Born" G.F. Handel

For unto Us a Child is Born

From Messiah

Choir and Orchestra

George Frideric Handel

1685-1759

Fig. 9.7:Sample no “6”; “Vent de Folie” Didlier Favre

Flûte

VENT DE FOLIE

Didier FAVRE

Fig. 9.8:Sample no "7"; "Ballade pour Anne" Guy Bergeron

Guy Bergeron (418) 842-6234 Ballade pour Anne p.2

Fig. 9.9:Sample no "8"; "Gerganla" Vilio Volante

Fig. 9.10:Sample no “9”; “Gerganla” Vilio Volante

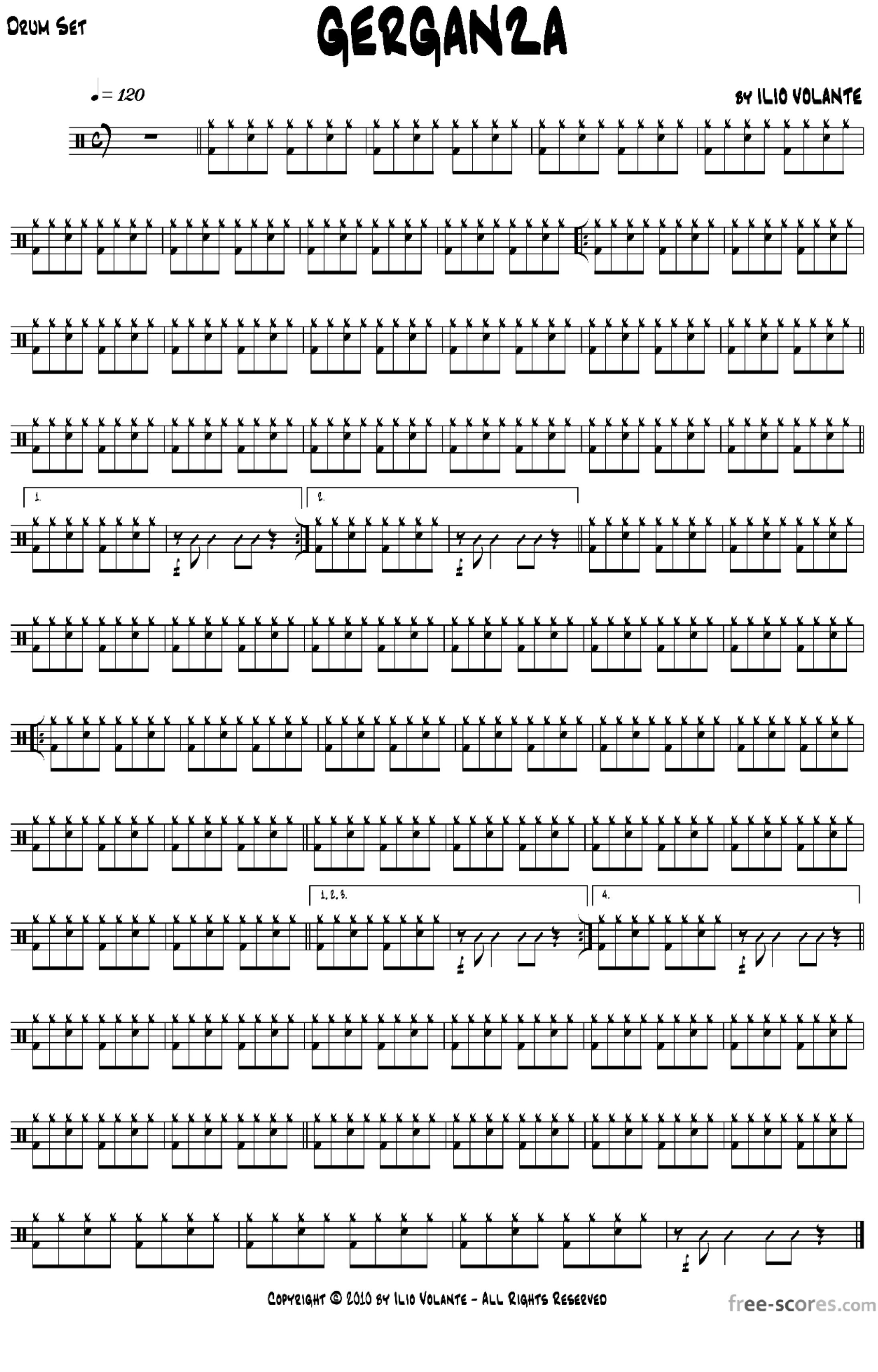


Fig. 9.11:Sample no "10"; "Tribale Poursuite" F. Rossoni

Tribale Poursuite

Drums

Frédéric Rossoni
1991

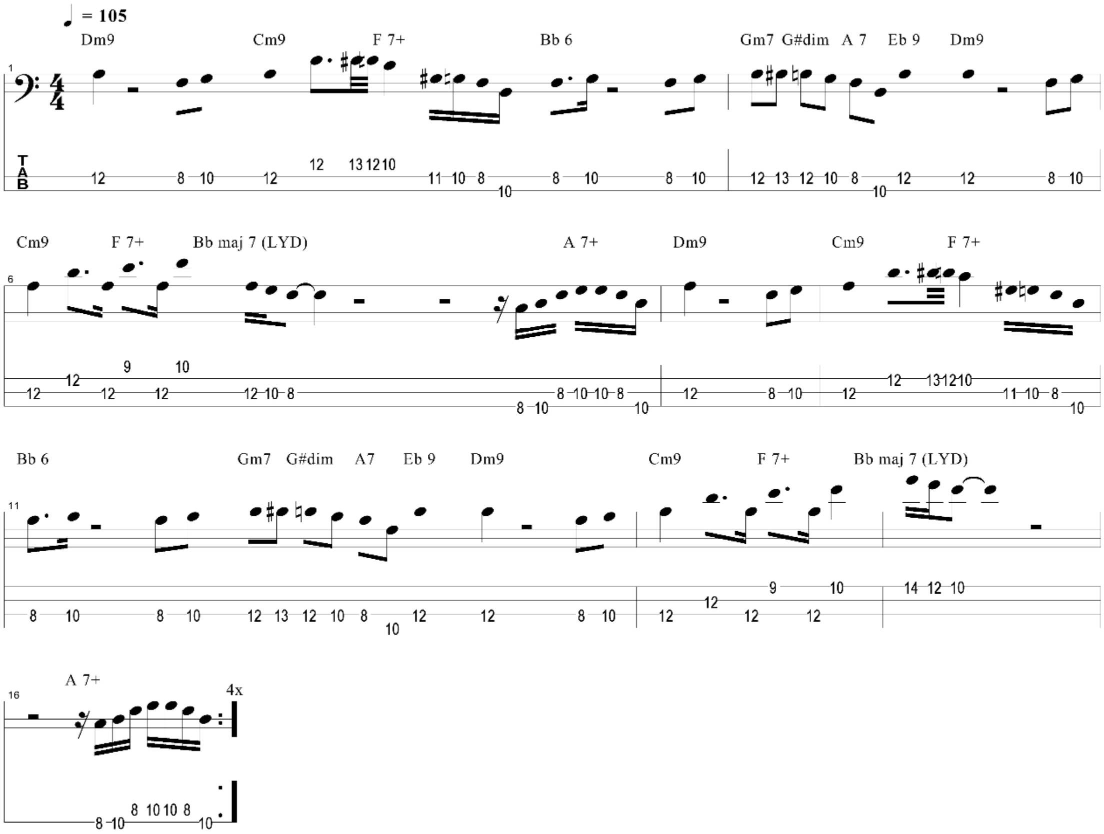
Fig. 9.12:Sample no "10"; "An Original" Andreas Losiewicz

Fig. 9.13:Sample no "12"; "Double Bass" Stephane Fernandez

Double Bass

Stéphane FERNANDEZ

Music by Stéphane FERNANDEZ



Page 1/1

Fig. 9.14:Sample no "13"; "Adeste Fideles"

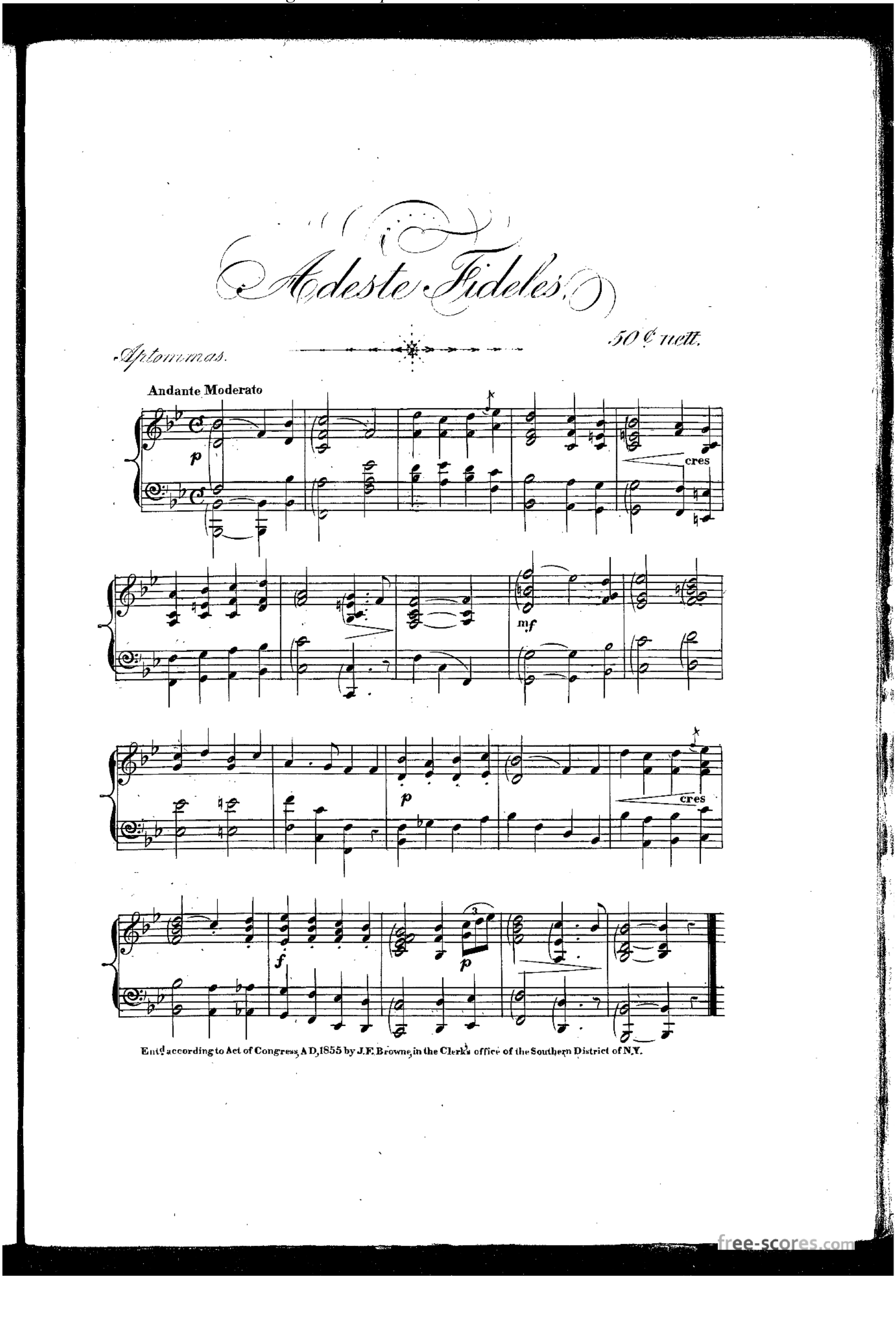


Fig. 9.15:Sample no "14"; "Back In Black" AC/DC

Back In Black

Fig. 9.16:Sample no "15"; "Bolling Suite for Flute and Jazz Piano"

Fig. 9.17:Sample no "16"; "Schindler's list THEME" J. Williams

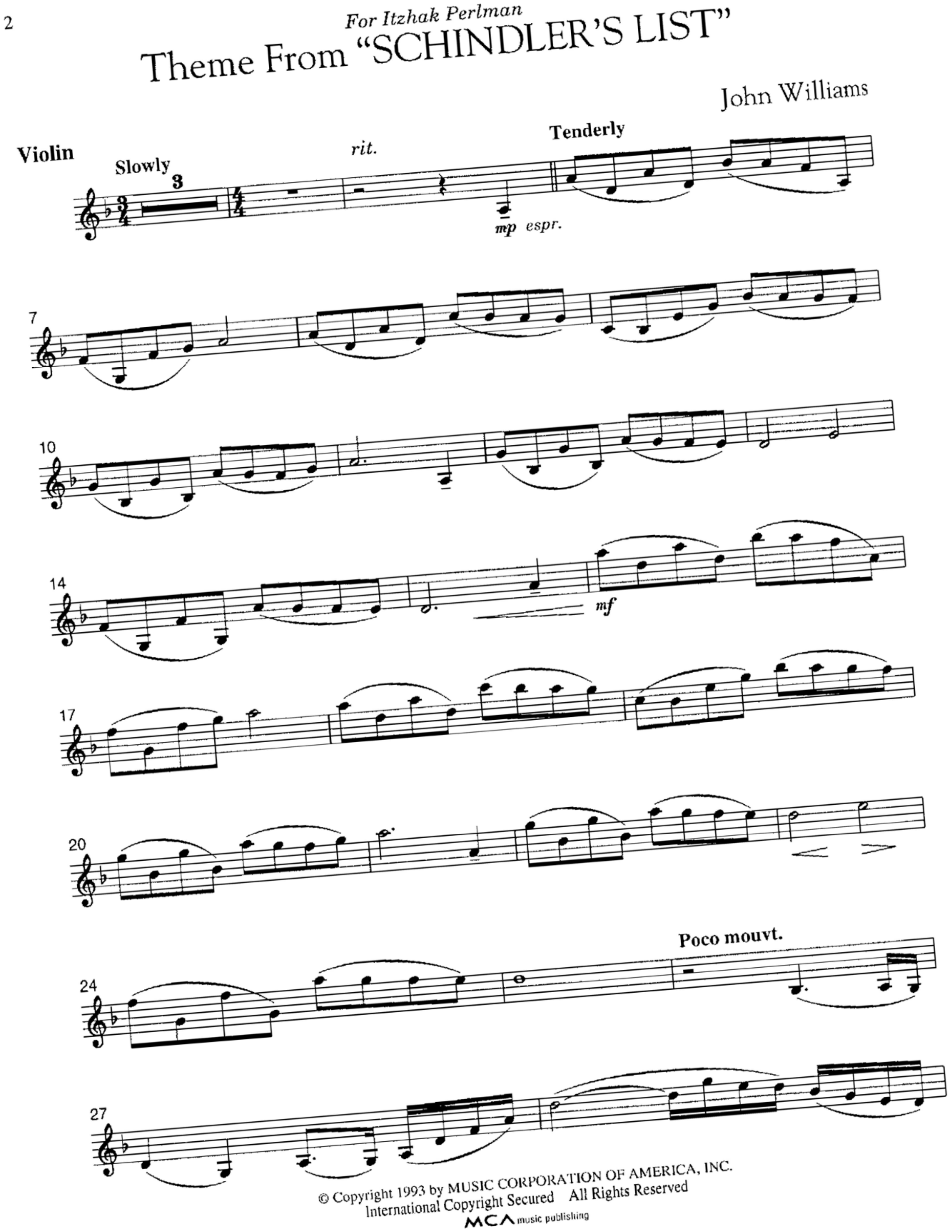


Fig. 9.18:Sample no "17"; "Rondo" Carlo Munier

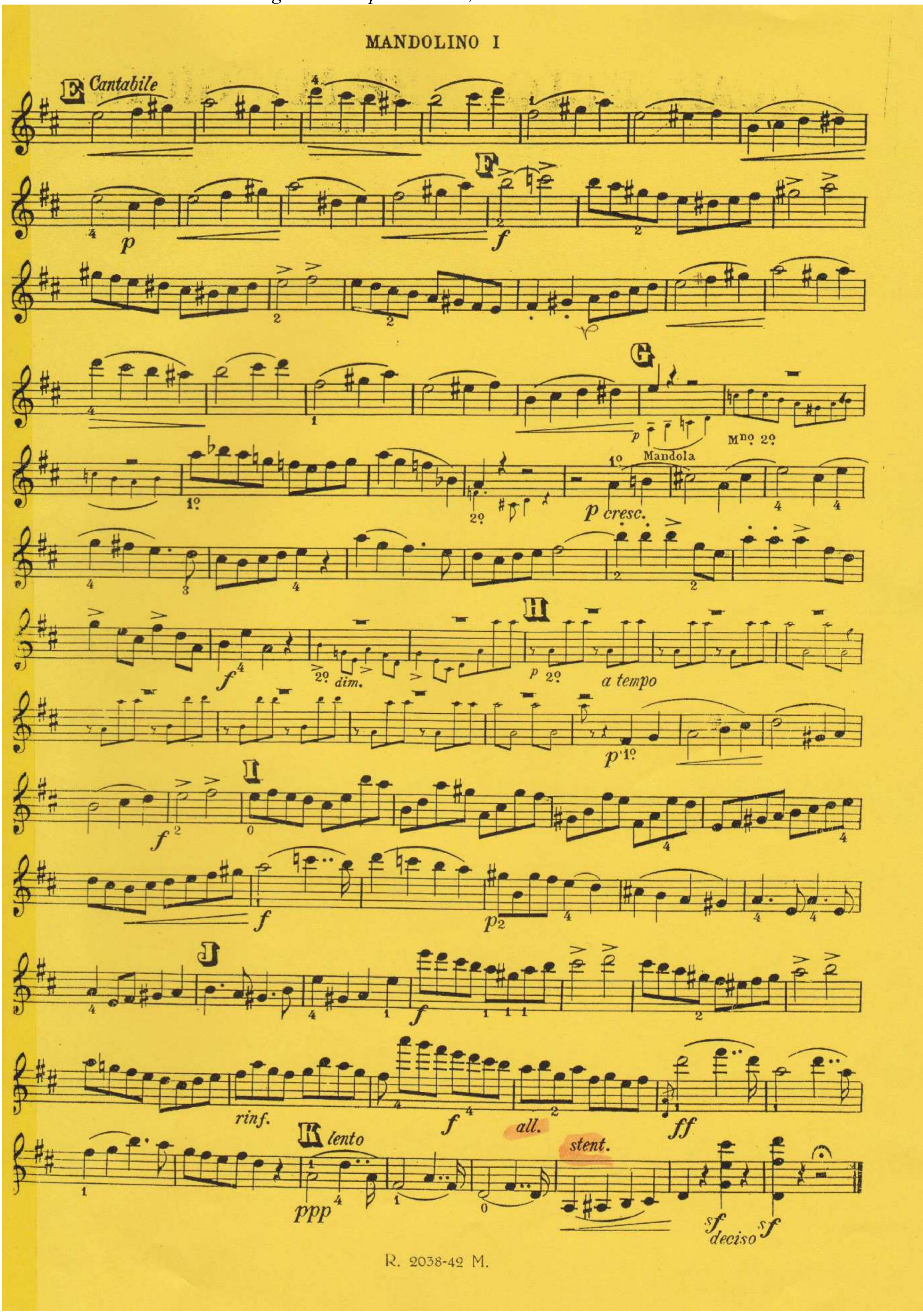


Fig. 9.19:Sample no "18"; "Invention" J.S. Bach

Fig. 9.20:Sample no “19”; “Czas nas uczy pogody” K. Dębski

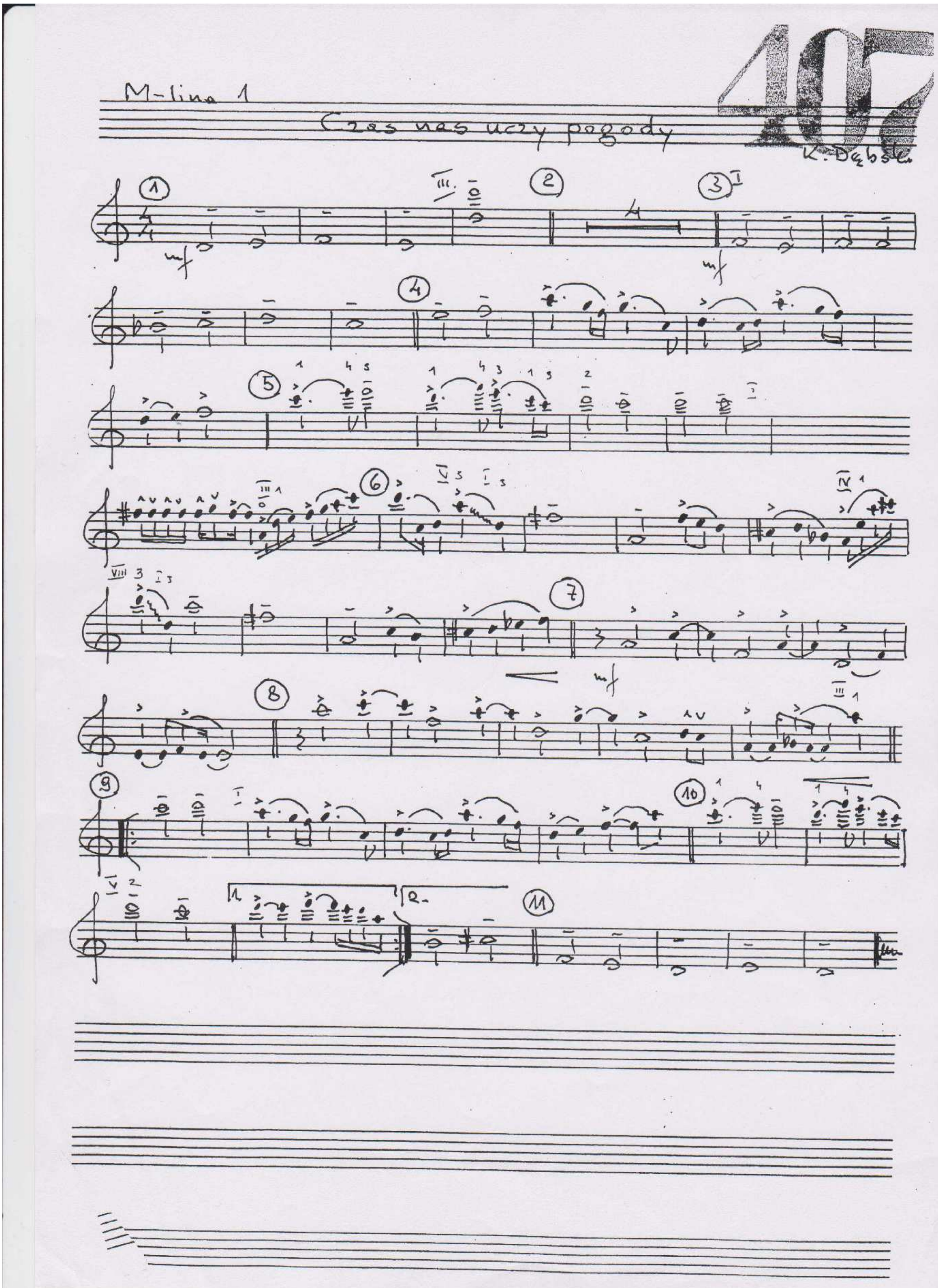


Fig. 9.21:Sample no "20"; "Naprawdę nie dzieje się nic" G. Turnau

NAPRAWDĘ NIE DZIEJE SIĘ NIC

Z repertuaru Grzegorza Turnay'a

KOLEKCJA NUT – Jerzy Kuberski, 75-323 KOSZALIN (muzykant-48@o2.pl)

10. Bibliography

[1]. Susan Ella George: *Visual Perception of Music Notation*: On-Line and Off Line Recognition, IRM Press, 2004, United Kingdom

[2]. David Bainbridge, Tim Bell: *An extensible Optical Music Recognition system*, Nineteenth Australasian Computer Science Conference, 2001, Australia

[3]. David Bainbridge, Tim Bell: *Identifying music documents in a collection of images*, Seventh International Conference on Music Information Retrieval (ISMIR), 2006, Kanada.

[4]. P. Bellini, I. Bruno, and P. Nesi: *Assessing optical music recognition tools*, Computer Music Journal 31 (1): 68-93, 2007

[5]. Bullen: *Bringing sheet music to life: My experiences with OMR*, code4lib Journal 3. 2008

[6]. A Rebelo,, G. Capela, J. Cardoso; *Optical recognition of music symbols: A comparative study*, International Journal on Document Analysis and Recognition 13 (1): 19–31, 2010

[7]. I. Fujinaga, F. Paszkiewicz, A. Rebelo, C. Guedes, A. Marcal, and J. Cardoso; *Optical music recognition: State of the art and open issues for handwritten music scores*, International Journal of Multimedia Information Retrieval (March 2012): 1–18. 2012

[8]. Jonathan Harnum: *Basic Music Theory, 4th ed.: How to Read, Write, and Understand Written Music*, CreateSpace Independent Publishing Platform, 2013

[9]. J. P. Marques de Sa: *Pattern Recognition: Concepts, Methods, and Applications*, Springer, 2001

[10]. Steven R. Newcomb: *Standard Music Description Language, Journal Computer - Special* issue: Computer-generated music Volume 24 Issue 7, July 1991 Page 76-79

[11]. Visiv Ltd 2013, *visiv SharpEye Music Scanning*, Visiv Ltd, accessed 3 March 2014, <http://www.visiv.co.uk/>

[12]. Musitek Corporation 2013*, Musitek SmartScore Music Scanning Software*, Musitek Corporation, accessed 3 March 2014 <http://www.musitek.com/>

[13]. goVivaldi Inc., Allegroassai spa. 2014, *Vivaldi Studio Music Scanning Software*, Vivaldi Studio, accessed 3 March 2014 <http://www.vivaldistudio.com/>

[14]. Neuratron Ltd 2013, *PhotoScore Music Scanning Software*, Neuratron Ltd, accessed 3 March 2014 <http://www.neuratron.com/>

[15]. Myriad 2013, *OMeR Music Scanning Software*, Myriad, accessed 4 March 2014 <http://www.myriad-online.com/en/products/omer.htm/>

[16]. Myriad 2013, *PDFtoMusic Music Scanning Software*, Myriad, accessed 4 March 2014 <http://www.myriad-online.com/en/products/pdftomusic.htm/>

[17]. Capella-software AG 2014, *Capella- Scan 7 Music Scanning Software*, Capella-software AG, accessed 4 March 2014 <http://www.capella.de/us/>

[18]. Hervé Bitteur 2014, *Audiveris Music Scanning Software*, Hervé Bitteur and others, accessed 4 March 2014 <https://audiveris.kenai.com/>

[19]. MakeMusic Inc 2014*, MusicXML*, MakeMusic Inc, accessed 10 February 2014 <http://www.musicxml.com/>

[20]. MIDI Manufacturers Association Incorporated 2014, *MIDI Manufacturers Association*, MIDI Manufacturers Association Incorporatedc, accessed 10 February 2014 <http://www.midi.org/>