Project Acronym: Europeana Sounds
Grant Agreement no: 620591
Project Title: Europeana Sounds

D2.3 Linking Music to Scores Delivery Report (software, documentation)

Revision: Final

Date: 03/04/2015

Authors: Alexander Schindler (AIT)

Sergiu Gordea (AIT)

Abstract: This document describes the work on the *Linking Music to Scores pilot* in Task 2.4.1. The intention behind the task was to evaluate the applicability of state-of-the-art score-following technologies to content provided by Europeana partner organizations and its integration within the Europeana technical infrastructure.

Dissemination level	
Public	X
Confidential, only for the members of the Consortium and Commission Services	

Coordinated by the British Library, the Europeana Sounds project is co-funded by the European Union, through the ICT Policy Support Programme as part of the Competitiveness and Innovation Framework Programme (CIP) http://ec.europa.eu/information_society/activities/ict_psp/

Revision history

Version	Status	Name, organisation	Date	Changes
0.1	ToC	Alexander Schindler, AIT	09/03/2015	
0.2	1st draft	Alexander Schindler, AIT	12/03/2015	
0.3	2nd draft	Sergiu Gordea, AIT	17/03/2015	Introduction
0.4	3rd draft	Alexander Schindler, AIT	18/03/2015	Review version
0.5	Final draft	Alexander Schindler, AIT	27/03/2015	Input from reviewers
1.0	Final	Richard Ranft, BL		Layout, minor changes

Review and approval

Action	Name, organisation	Date
Reviewed by	Marnix van Bechum, Utrecht University	18/03/2015
Approved by	Coordinator and PMB	31/03/2015

Distribution

No.	Date	Comment	Partner / WP
1	02/04/2015	Submitted to the European Commission	BL/WP7
2	02/04/2015	Posted on Europeana Pro website	BL/WP7
3	02/04/2015	Distributed to project consortium	BL/WP7

Application area

This document is a formal output for the European Commission, applicable to all members of the Europeana Sounds project and beneficiaries. This document reflects only the author's views and the European Union is not liable for any use that might be made of information contained therein.

Statement of originality

This document contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Project summary

Europeana Sounds is Europeana's 'missing' fifth domain aggregator, joining APEX (Archives), EUscreen (television), the Europeana film Gateway (film) and TEL (libraries). It will increase the opportunities for access to and creative re-use of Europeana's audio and audio-related content and will build a sustainable best practice network of stakeholders in the content value chain to aggregate, enrich and share a critical mass of audio that meets the needs of public audiences, the creative industries (notably publishers) and researchers. The consortium of 24 partners will:

- Double the number of audio items accessible through Europeana to over 1 million and improve geographical and thematic coverage by aggregating items with widespread popular appeal such as contemporary and classical music, traditional and folk music, the natural world, oral memory and languages and dialects.
- Add meaningful contextual knowledge and medium-specific metadata to 2 million items in Europeana's audio and audio-related collections, developing techniques for cross-media and cross-collection linking.
- Develop and validate audience specific sound channels and a distributed crowd-sourcing infrastructure for end-users that will improve Europeana's search facility, navigation and user experience. These can then be used for other communities and other media.
- Engage music publishers and rights holders in efforts to make more material accessible online through Europeana by resolving domain constraints and lack of access to commercially unviable (i.e. out-of-commerce) content.

These outcomes will be achieved through a network of leading sound archives working with specialists in audiovisual technology, rights issues, and software development. The network will expand to include other data-providers and mainstream distribution platforms (Historypin, Spotify, SoundCloud) to ensure the widest possible availability of their content.

For more information, visit http://pro.europeana.eu/web/europeana-sounds and http://www.europeanasounds.eu

Copyright notice

Contents

Executive summary: D2.3 Linking Music to Scores Delivery Report 6

1 Introduction 6

2 Introduction to score-following 7
2.1 What is score-following? 8
2.1.1 Symbolic to MIDI alignment 8
2.1.2 Symbolic to Audio alignment 8
2.2 Score-following system components 9

3 Optical Music Recognition 11
3.1 Score-following system components 11
3.2 Optical Musical Recognition software 13
3.2.1 Proprietary software 13
3.2.2 Open source software 14
3.2.3 Accuracy of OMR software 14
3.3 OMR software used for Europeana Sounds 15
3.3.1 Evaluated software 15
3.3.2 OMR software used for prototyping: Audiveris 15

4 Symbolic music processing 17
4.1 Extracting layout information 17
4.2 Concatenating multiple score pages 20
4.3 Processing repetitions 21

5 Feature extraction 21
5.1 Audio feature extraction 21
5.1.1 General audio feature extraction 22
5.2 Symbolic music feature extraction 23
5.3 Audio-to-score alignment relevant music features 23

6 Audio to score alignment 26
6.1 Dynamic time warping 26
6.2 Assign audio-timecodes to measures from the music score 28
6.2.1 Mapping Chroma features to audio 28
6.2.2 Mapping Chroma features to measures 28

7 Graphical User Interface 29
7.1 HTML5 framework 29

8 Requirements and pre-requisites for development 30
8.1 Development environment 30
8.1.1 Python libraries required for processing 30

8.1.2 Python libraries required for visualisation 32
8.2 Content requirements 33
8.2.1 Technical dependencies 33

9 Results 39
Prototype score following player 40

10 Considerations towards integration 40
10.1 Integration scenarios 40
10.1.1 Scenario 1 40
10.1.2 Scenario 2 41
10.1.3 Scenario 3 42
10.1.4 Scenario 4 43
10.1.5 Scenarios beyond state-of-the-art 45

11 Summary 45

12 References 46

Appendix A: Terminology 46

Executive summary: D2.3 Linking Music to Scores Delivery Report

The intention behind Task 2.4.1 *Linking Music to Scores pilot* was to evaluate the applicability of state-of-the-art score-following technologies to content provided by Europeana partner organizations and its integration within the Europeana technical infrastructure.

The task was approached through a prototypical implementation of a score-following system. First, necessary components were identified. Second, available open-source components were investigated. Based on this a Python based prototypical implementation of an audio-to-score alignment system was created, which output was used to create HTML5 based interfaces to visualize the calculated alignment. This score-following Web interface was used to demonstrate and evaluate the objectives and obstacles of the chosen approach.

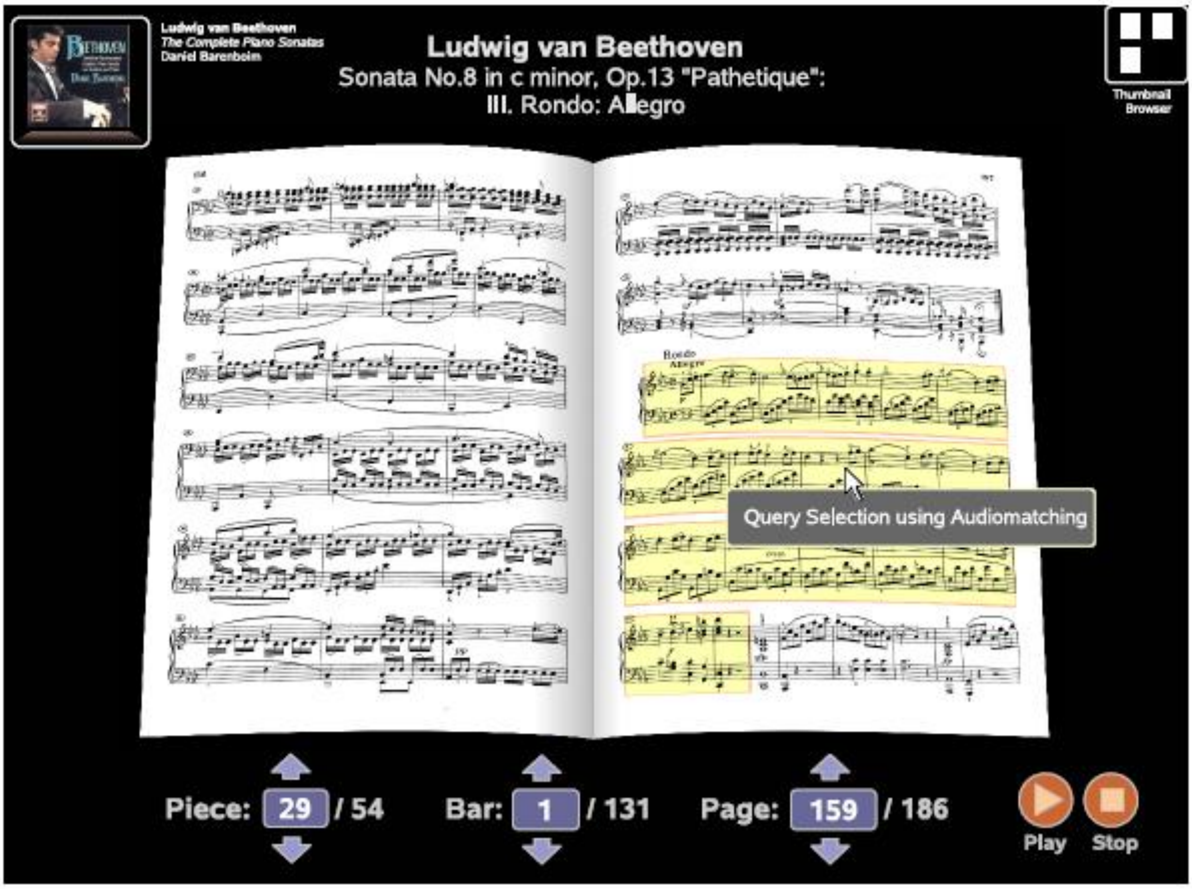
1 Introduction

Music is an important component of the human society even in ancient times. It was and it is produced and consumed by people from all social categories in their daily life. The development of ICT technologies nowadays facilitates the creation, distribution and consumption of music content. This development has a high impact on the current European and global society as recognized by the European and International Music Councils within the Manifesto for Youth and Music in Europe[1]. This document emphasises the idea that music making and music performances imply lifelong learning activities for professional or hobby musicians as well as for the consumers. To enhance these learning activities it is very important to bridge the gap between the formal, non-formal and informal education. Online instrument playing tutorials with sheet music are great tools support self-learning activities, however they typically operate with digitally-born content or manually created, non-interactive music alignments[2].

Within the scope of the Europeana Sounds project, the consortium aggregates and publishes large amounts of music and music related content, which has a great potential to experience diversity, inspire creativity and support music education. Unfortunately, the majority of this content is not digitally born content and consequently, it cannot be used within the existing music education tools.

The Task T2.4.1 *Linking Music to Scores Pilot* of Europeana Sounds project has the goal to search for technical solutions for interacting with sheet music and corresponding audio registrations. It investigates automated algorithms for processing digitized sheet music, processing audio content and discovering the relationships between different representations of the same artwork. The envisioned score-following system can be compared to the approach developed for the Probado project [REF 1]. An interface similar the Score Viewer Interface described in [REF 2] is contemplated.

[1] http://www.emc-imc.org/youth/manifesto-for-youth-and-music-in-europe/
[2] https://www.youtube.com/watch?v=u2lSZyxK7eA



Piece: 29 / 54 Bar: 1 / 131 Page: 159 / 186

Figure 1: The Score Viewer Interface of the Framework for managing multimodal digitized music collections developed for the Probado digital library initiative [REF 1] and installed at the Bavarian State Library in Munich (image taken from [REF 2]).

2 Introduction to score-following

Music is a multifaceted subject that is generally recognised to be of acoustic nature. This usually does not apply for the various forms of music descriptions. Such descriptions, also referred to as music notation, represent music in symbolic form. Various notation standards have been developed over the last centuries which have been used for composition, distribution, preservation, etc. The main difference between symbolic music and recorded or perceived music is, that scores only describe how a piece of music is intended to be performed. In other words: It is a guideline that is used by a performer to play a certain piece of music. This process is thus open for personal interpretation. The performer might know the track and might on the one hand try to align his performance to this previous interpretation. On the other hand, the track might be completely unknown, or the performer may decide to interpret the score in their own way. Either of these cases creates different sounding performances, based on the exact same score.

2.1 What is score-following?

Score-following relates to the process of aligning the notes of a score to its interpretation during a specific performance [REF 3]. This might be during a live performance, or to an audio recording. Research on automatic score-following through computers distinguishes between different scenarios:

- Symbolic to MIDI
- Symbolic to Audio

Their main differences are based on the digital representation of music. Symbolic music is usually represented in a machine processable form, such as MIDI[3] or MusicXML[4]. Recorded music is represented as a time-series of sampled audio. More generally speaking, sampled audio is an exhaustive sequence of numbers representing the measured audio energy at a certain time. Information provided in this form is not suitable for automatic processing at first hand and has to be reduced and transformed into an appropriate representation. This process is the focus of the research domain music information retrieval (MIR). While, in the symbolic representation of music, it is clear which note is being played at a certain time, in digital audio this information has to be deduced from the spectral properties of the sampled audio through digital signal processing.

2.1.1 Symbolic to MIDI alignment

This scenario is based on the prerequisite that the performed music is available in the same symbolic format of the score. This is the case for set ups where a musical instrument is connected to the computer through a Musical Instrument Digital Interface (MIDI). Music events, such as hitting a key on an electronic piano equipped with a MIDI interface, are immediately communicated to the connected computer and are available in an interpretable from. Having both sources - scores and performance - in a comparable format makes it more convenient for further processing (e.g. score-following, automatically assisted training, etc.).

2.1.2 Symbolic to Audio alignment

While symbolic music unambiguously describes which note is played at which time of the track, this does not apply to recorded music. The main challenge with sampled audio is that it is a mix of frequencies, originating usually from a multitude of individual instruments and voices, which sources currently cannot fully be separated again, after they have been fixed in an audio mix. This topic is subject to slowly progressing research. Musical notes refer to audio frequencies (e.g. concert pitch = 440Hz). Thus, it seems obvious, that sampled audio can be transcribed into symbolic music by assigning note values to audio frequencies. In a simplified approach this works for monophonic tunes played by a single instrument. Having multiple instruments playing polyphonic tunes (like chords and harmonies) creates overlapping frequencies, partial frequencies caused by the instrument’s timbre and other influences in the overall audio mix, which cause complex distributions of the sound energy over the frequency spectrum of the recording. Thus, it is not computationally distinguishable anymore which

[3] https://en.wikipedia.org/wiki/MIDI
[4] http://www.musicxml.com/

notes have been played by the distinct instruments. To align symbolic music to sampled audio, both types have to be transformed into a representation that can be compared directly. This is the prevalent case for the score-following prototype developed during the Europeana-Sounds project and will be described in the remainder of this document.

2.2 Score-following system components

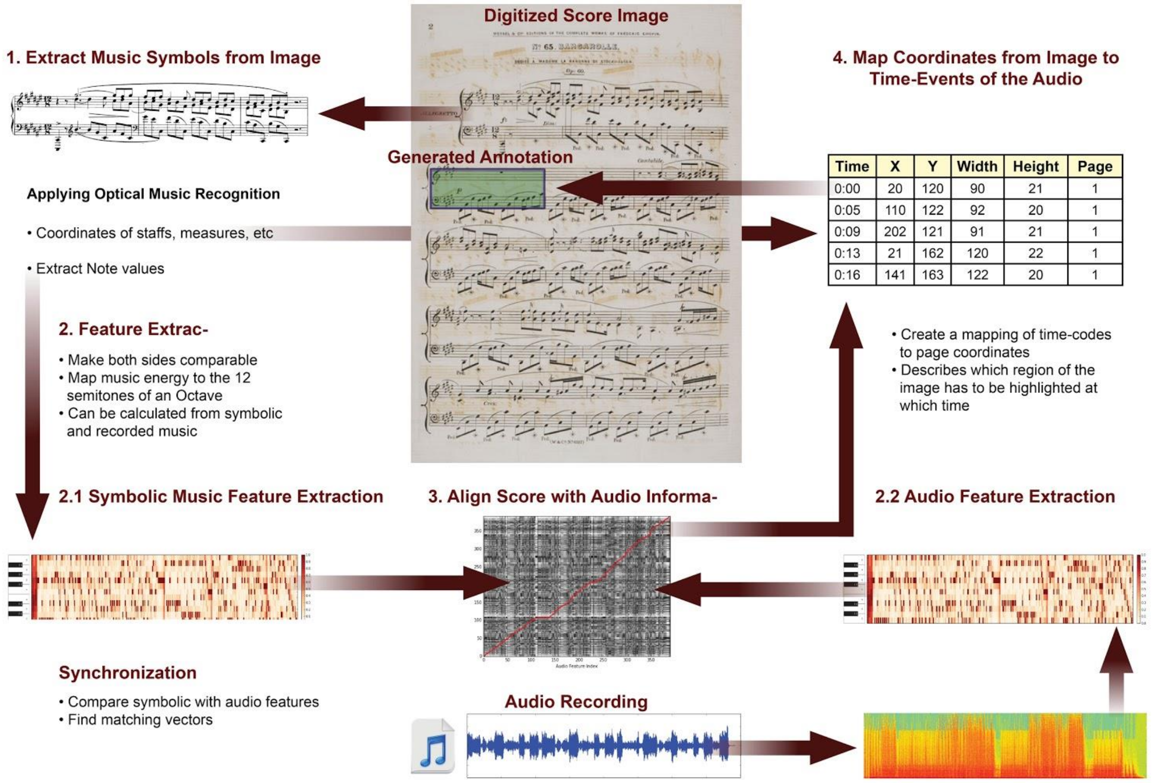
A Score Following System (SFS) generally consists of the following components [2-4]:

- Optical Music Recognition
- Feature Extraction
- Audio to Score Alignment
- Graphical user Interface

Optical Music Recognition (OMR) is the process of extracting symbolic music information from digitized score pages. This task, which is tightly related to Optical Character Recognition (OCR)[5] is explained in more detail in *Section 3: Optical Music Recognition*. *Feature extraction* refers to algorithmic data transformation and aggregation routines. Discriminative and semantic descriptive values are calculated from the audio spectrum or the symbolic music description. These processes are further detailed in *Section 5: Feature Extraction*. *Audio to Score Alignment* is based on comparable features extracted from recorded audio and symbolic descriptions. This process of finding an optimal alignment between the music description and its interpretation is described in *Section 6: Audio To Score Alignment*. Information resulting from these three major steps provides a mapping between the elements depicted on music scores and sequences in the audio recording. *Section 7: Graphical User Interface* provides an overview of how to use state-of-the-art Web technology to interactively visualise this mapping by providing a score following prototype.

The workflow of audio-to-score alignment is depicted by the following chart:

[5] https://en.wikipedia.org/wiki/Optical_character_recognition


Digitized Score Image
1. Extract Music Symbols from Image
4. Map Coordinates from Image to Time-Events of the Audio
Generated Annotation
Applying Optical Music Recognition
• Coordinates of staffs, measures, etc
• Extract Note values
2. Feature Extrac-
• Make both sides comparable
• Map music energy to the 12 semitones of an Octave
• Can be calculated from symbolic and recorded music
• Create a mapping of time-codes to page coordinates
• Describes which region of the image has to be highlighted at which time
2.1 Symbolic Music Feature Extraction
3. Align Score with Audio Informa-
2.2 Audio Feature Extraction
Synchronization
• Compare symbolic with audio features
• Find matching vectors
Audio Recording

Time	X	Y	Width	Height	Page
0:00	20	120	90	21	1
0:05	110	122	92	20	1
0:09	202	121	91	21	1
0:13	21	162	120	22	1
0:16	141	163	122	20	1

Figure 2: An overview of the process behind score following.

1. **Symbolic music information extraction:** In this step the symbolic description of the music is extracted from the music sheet images. This includes the semantic description (e.g. notes, measures, clefs, etc.) and the page layout (e.g. coordinates of the staffs, measures, systems, etc.).

2. **Feature Extraction:** In terms of audio processing symbolic music differs greatly from recorded audio. The two music representations have to be converted into a comparable from, which is accomplished through mapping the music energy to the twelve semitones of the musical octave. This procedure is applied to both forms in small subsequent time intervals, providing two lists of audio features.

3. **Audio to Score Alignment:** The previously calculated lists of features can be used to align the score information to the recorded audio. The feature vectors are compared and most similar consecutive sequences are identified.

4. **Audio to Image Mapping:** The indexes of the aligned features are cross-referenced to their page numbers and coordinates in the score image as well as to the corresponding time-events of the audio file.

5. **Visual Representation of the Results:** To visualise the computational results an HTML5 Webpage has been created. The required functionality to show the music sheet images, interact

on audio events, highlight the currently played measure and handle page turns appropriately, has been implemented in Javascript.

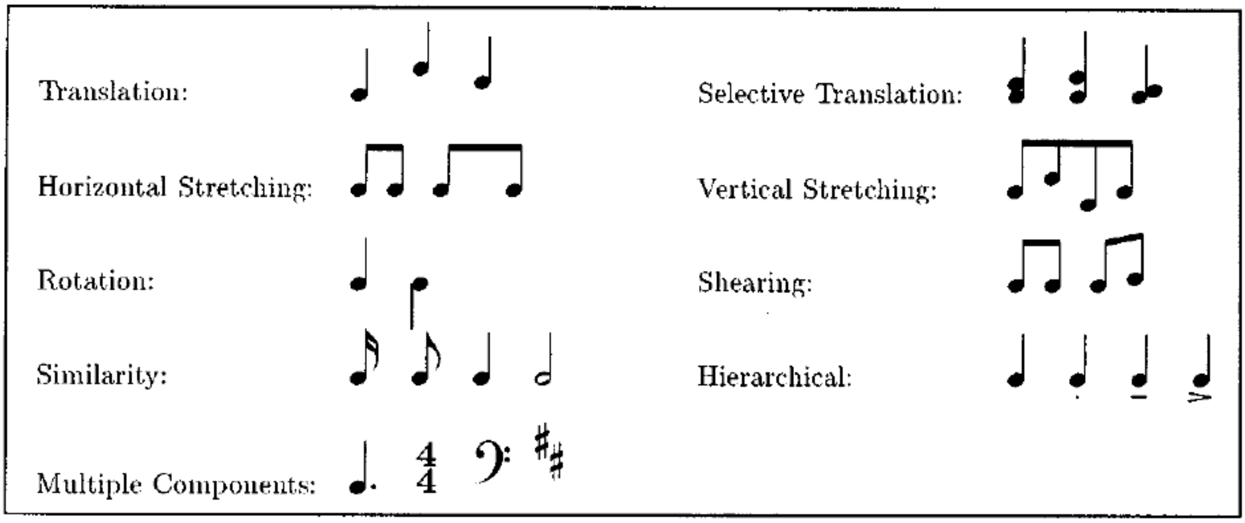
3 Optical Music Recognition

Optical Music Recognition (OMR) [REF 7] is similar to Optical Character Recognition (OCR) and intends to convert music scores into a machine interpretable form. Actually, OMR is a descendant of OCR. Most systems are based on top of OCR engines and extend their functionality by adding layers that recognise and interpret music notations. Contrary to the advanced solutions provided for optical character recognition, OMR is still a challenging research topic and far from being solved. In other words: The results of OMR technology are currently less reliable than those of the OCR technology it builds upon.

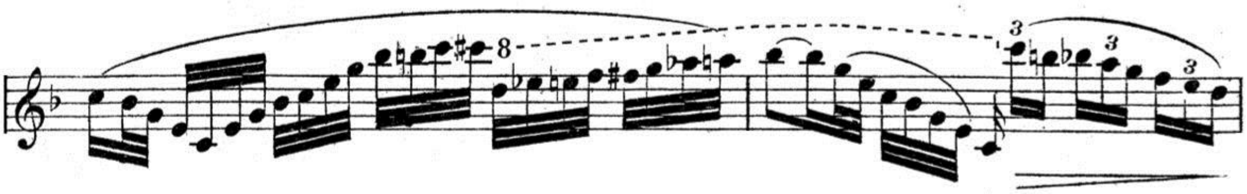
3.1 Score-following system components

The aim of optical music recognition is to get a score from the paper into a machine interpretable format, to have it processed or played. This requires converting it from its analogue into a digital form, which is usually accomplished through scanning a page. The result is a digital image of the score which can be displayed on a screen or printed. Yet, it provides no more information to the computer than its analogue form (it is a digital image file). The visual information of the image has to be extracted and converted into a processable format such as MIDI or MusicXML. Similar to OCR, document image analysis is applied to detect the page layout and to recognize where the semantically relevant information is located.

The advantage of traditional text recognition over OMR is that once the locations of columns and lines are known, characters can be recognised one by one. There is no further dimension to keep in mind that influences the recognition of the consecutive character. Music scores on the other hand rely on conditions set at the start of a line or even further behind. Different clefs result in different note assignments. This information is given at the left side of a staff and has to be remembered for the entire line. Yet, this is just one of many challenges OMR has to face. There are many ambiguities that are comprehended by musicians, but are really difficult to translate into a general model or template for music notation. The following table, taken from [REF 8] lists some of these problems:



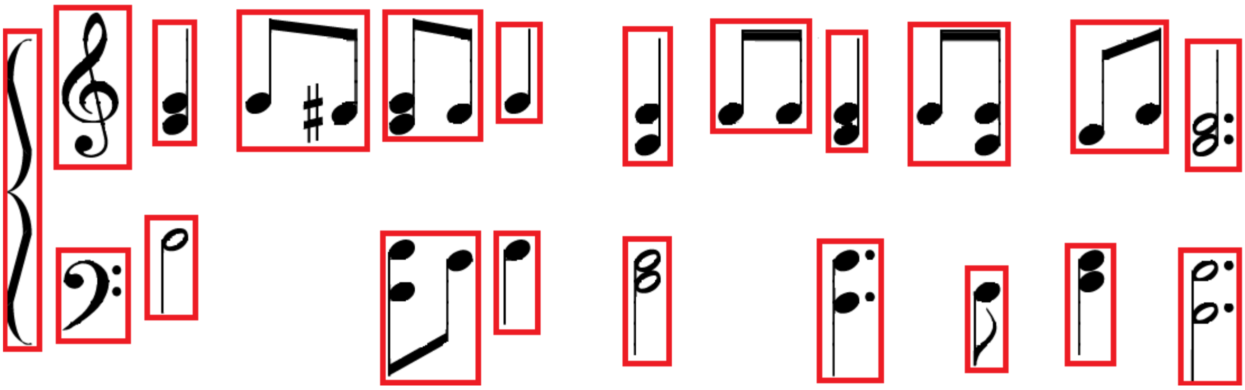
Most of these variations are based on layout considerations. Usually readability is traded for strict syntactical rules. Sometimes, strong deviations are still comprehended, just because nothing else would make sense. From the point of view of a rule-based recognition engine, this results in vast definitions of exceptions. Apart from notes, music scores also contain a wide range of different content types such as text, punctuations, lines, etc.



Generally all optical character recognition systems – and music scores are just a special form of notations – imply document analysis, pre-processing steps such as layout detection, rotation, contrast enhancement, binarisation, etc. All these steps are required to improve the precision of the further steps.

Staff line and object detection: The identification of the staff lines is fundamental and a common first step of all implementations of OMR systems. Based on the spatial locations of the staffs, important parameters such as staff spacing, note sizes and heights, etc. can be estimated. Once the spatial information is known, the staff lines are not required anymore and are usually removed. The remaining musical objects are located and segmented into isolated shapes.

Classification of musical shapes: Having the staffs removed makes the identification of the musical objects seem much easier. Still the various permutations of possible ways to transcribe music are problematic. Thus, a common solution is to approach the classification on a sub-symbol level. The musical shapes are divided into smaller items (e.g. heads, stems, etc.) and identified on their primitive level.



When all the shapes have been successfully labelled, the primitive shapes of the image are replaced by perfectly formed geometrical shapes. At this stage it is already possible to load the extracted information into an external editor to correct errors of the recognition engine.

Semantic interpretation of the musical objects: The clean version of the musical score is still lacking a proper semantic interpretation. Approaches to this final step of OMR are usually based on representing the identified items as a graph that is updated sequentially over several passes. Within these passes, spatial relationships are identified and links between the shapes are created. This step often relies on extensive musical knowledge that is modelled as a set of rules or templates.

Finally, the resulting graph is stored in a format that is interpretable by external music software (e.g. MusicXML, MIDI, etc.).

3.2 Optical Musical Recognition software

A small set of OMR software is available up to now and is mentioned in this report for completeness. Yet, it has to be stated that OMR is still subject to research and that none of the provided systems can be deemed fully reliable and generally applicable.

3.2.1 Proprietary software

The following list provides an overview of proprietary OMR systems.

- **SmartScore[6]:** a fully integrated OMR software. All tasks of a recognition process can be accessed and performed from a single Graphical User Interface (GUI). The OMR process is to some extend customisable. The software includes a MIDI editor to post-process the extracted score information.
  - Output: MIDI, MusicXML
- **SharpEye[7]:** provides a simple graphical interface with drag and drop option to initiate the recognition process. A simple editor is provided to correct recognition errors. Batch processing for processing multiple images sequentially is provided.
  - Output: MIDI, NIFF, MusicXML

[6] http://www.musitek.com/
[7] http://www.visiv.co.uk/

- **PhotoScore[8]:** a fully integrated OMR software which claims to recognise even handwritten music to some extent. A full MIDI editor is provided for post-processing.
  - Output: MIDI, NIFF, MusicXML
- **Capella-Scan[9]:** integrated OMR software including MIDI editor to post-process extracted score information.
  - Output: Capella, MIDI, MusicXML

3.2.2 Open source software

Using open source software is a requirement for Europeana Sounds. The list of open source OMR software is short. Further obstacles are that some projects seem to be discontinued and others are announced to become closed source solutions in a near future.

- **OpenOMR[10]:** OMR system written in the Java programming language.
  - License: GPLv2
- **Audiveris[11]:** Integrated OMR solution written in Java but based on the Tesseract[12] recognition engine. Software contains a MIDI editor for post-processing and provides a command line interface including a batch processing mode.
  - License: GPLv2
  - Output: MusicXML
- **Gamera[13]:** not a packaged recognition system but a toolset that can be used to create new systems. Often used in academia to create or prototype new OMR algorithms or approaches.
  - License: GPLv2
- **Aruspix[14]:** OMR software for early music prints.
  - License: GPLv3

3.2.3 Accuracy of OMR software

Proprietary software commercially available often promises accuracies of up to 99% for printed music scores. Currently there are no standardised benchmark datasets to evaluate the performance of OMR

[8] http://www.neuratron.com/photoscore.htm
[9] http://www.capella.de/us/index.cfm/products/capella-scan/info-capella-scan/
[10] http://sourceforge.net/projects/openomr/
[11] https://audiveris.kenai.com/
[12] https://code.google.com/p/tesseract-ocr/
[13] http://gamera.informatik.hsnr.de/addons/musicstaves/
[14] http://www.aruspix.net

systems. Only a few evaluations are known to literature, most of them focussing on freely available solutions, including open source software and research prototypes.

Different solutions have been provided to amend the problem of inaccurate OMR output including large-scale.

3.3 OMR software used for Europeana Sounds

One major premise for the development of an audio-to-score alignment system for Europeana Sounds was the use of open-source software. The principle of the project using and creating open source software is set out in the Europeana Sounds Description of Work[15].

3.3.1 Evaluated software

An extensive search in academic literature revealed that currently two open source software packages are suitable for the application in Europeana Sounds: *Audiveris* and *Gamera*. Both systems were tested and evaluated to reasonable extent. Due to the design properties of Gamera - especially being a toolset rather than a dedicated package - this solution provided problems during installation, initiation and evaluation. Finally, it was decided to use Audiveris due to its usability, integration of features and acceptable performance.

3.3.2 OMR software used for prototyping: Audiveris

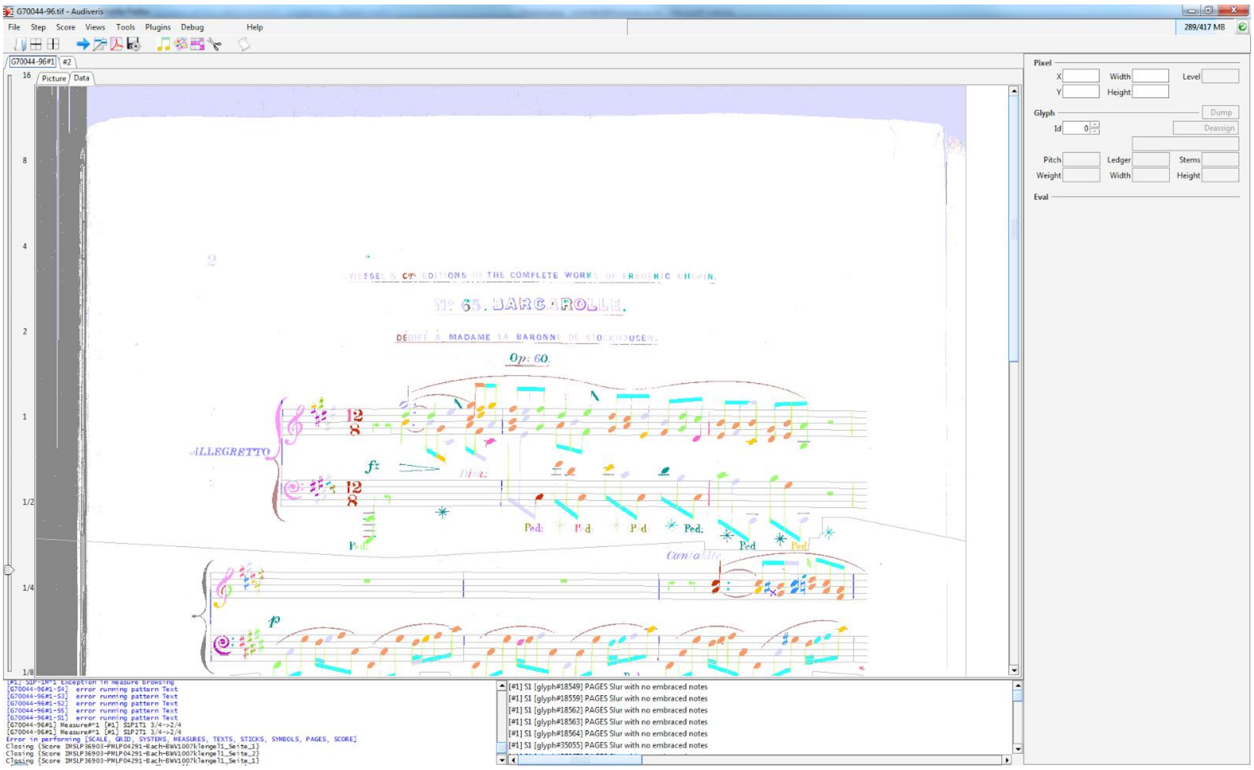
Audiveris is a well-designed open source OMR software that is already available in version 4.3. The next major version increment has been announced for April 2015.

The software is intended to be used directly from the Web via Java Webstart. By simply clicking on the “Launch” button on the Web page, the required binaries are loaded and the Java Web start interface starts the application. This is a convenient solution for private usage but to integrate the tool into audio-to-score alignment workflow, the binaries need to be available locally. Audiveris is still an open source project and its sources can easily be compiled.

Graphical User Interface

Audiveris provides a graphical user interface to load images and start the recognition process. Extracted score information is superimposed on the original image. Different visualisations are provided to reproduce the results.

[15] EC-GA including Annexe 1 (“Description of Work”) p105



```
[#1] S1P-1M*1 Exception in measure browsing
[G70044-96#1-S4]  error running pattern Text
[G70044-96#1-S3]  error running pattern Text
[G70044-96#1-S2]  error running pattern Text
[G70044-96#1-S5]  error running pattern Text
[G70044-96#1-S1]  error running pattern Text
[G70044-96#1] Measure#*1 [#1] S1P1T1 3/4->2/4
[G70044-96#1] Measure#*1 [#1] S1P2T1 3/4->2/4
Error in performing [SCALE, GRID, SYSTEMS, MEASURES, TEXTS, STICKS, SYMBOLS, PAGES, SCORE]
Closing {Score IMSLP36903-PMLP04291-Bach-BWV1007klengel1_Seite_1}
Closing {Score IMSLP36903-PMLP04291-Bach-BWV1007klengel1_Seite_2}
Closing {Score IMSLP36903-PMLP04291-Bach-BWV1007klengel1_Seite_1}
```

```
[#1] S1 [glyph#18549] PAGES Slur with no embraced notes
[#1] S1 [glyph#18559] PAGES Slur with no embraced notes
[#1] S1 [glyph#18562] PAGES Slur with no embraced notes
[#1] S1 [glyph#18563] PAGES Slur with no embraced notes
[#1] S1 [glyph#18564] PAGES Slur with no embraced notes
[#1] S1 [glyph#35055] PAGES Slur with no embraced notes
```

Figure 3: Graphical User Interface of the Audiveris OMR software.

Command Line Interface

Audiveris intended application is through its GUI but it also provides an extensive command line interface. All relevant tasks can be triggered and executed from this interface. The advantage of this approach is that it facilitates the integration of Audiveris into complex workflows requiring OMR. A further advantage over the graphical interface is the additional batch option which provides the possibility to process instantiate the software without starting a user interface. This is an essential option to run Audiveris as a separated process on hosts that provide no windowing managers.

```
Version:
  4.3.3406
Arguments syntax:
  [-help]                           Prints help about application arguments and stops
  [-batch]                          Specifies to run with no graphic user interface
  [-step (STEPNAME|@STEPLIST)+]     Defines a series of target steps
  [-option (KEY=VALUE|@OPTIONLIST)+]  Defines a series of key=value constant pairs
  [-script (SCRIPTNAME|@SCRIPTLIST)+] Defines a series of script files to run
  [-input (FILENAME|@FILELIST)+]    Defines a series of input image files to process
  [-pages (PAGE|@PAGELIST)+]        Defines a set of specific pages to process
  [-bench (DIRNAME|FILENAME)]       Defines an output path to bench data file (or
directory)
  [-print (DIRNAME|FILENAME)]       Defines an output path to PDF file (or directory)
  [-export (DIRNAME|FILENAME)]      Defines an output path to MusicXML file (or
directory)

Known step names are in order (non case-sensitive):
  LOAD         : Reload the sheet picture
  SCALE        : Compute general scale
  GRID         : Retrieve the grid of all systems
  SYSTEMS      : Split all data per system
```

```
MEASURES     : Retrieve measures from bar sticks
TEXTS        : Retrieve texts items in each system area
STICKS       : Extract vertical & horizontal sticks
SYMBOLS      : Apply specific glyph patterns
PAGES        : Translate glyphs to score items
SCORE        : Build the final score
PRINT        : Write the output PDF file
EXPORT       : Export the score to MusicXML file
```

Pros and Cons of Audiveris

- Pros
    - open source
    - MusicXML output
    - Command line Interface
    - batch mode
- Cons
    - No handwritten scores
    - Plans to close source in version 5
    - Availability through Web APIs

4 Symbolic music processing

This section describes relevant topics related to processing the information extracted in the previous step using optical music recognition. Symbolic music has various digital representations such as MIDI, MusicXML, ABC[16] and GUIDO[17]. The most powerful among them is MusicXML which was released first in 2004 and describes not only the composition but also the layout of the score, lyrics and textual annotations and further features.

4.1 Extracting layout information

Layout description in MusicXML is an addition to the symbolic music description. It is a by-product of optical music recognition. To efficiently detect the music information, the layout of the page, systems, staff lines, measures, etc. have to be detected. This information is finally attached to the resulting MusicXML file. The following MusicXML excerpt shows information stored in the header section, describing general layout properties of the processed page:

[16] http://en.wikipedia.org/wiki/ABC_notation
[17] http://en.wikipedia.org/wiki/GUIDO_music_notation

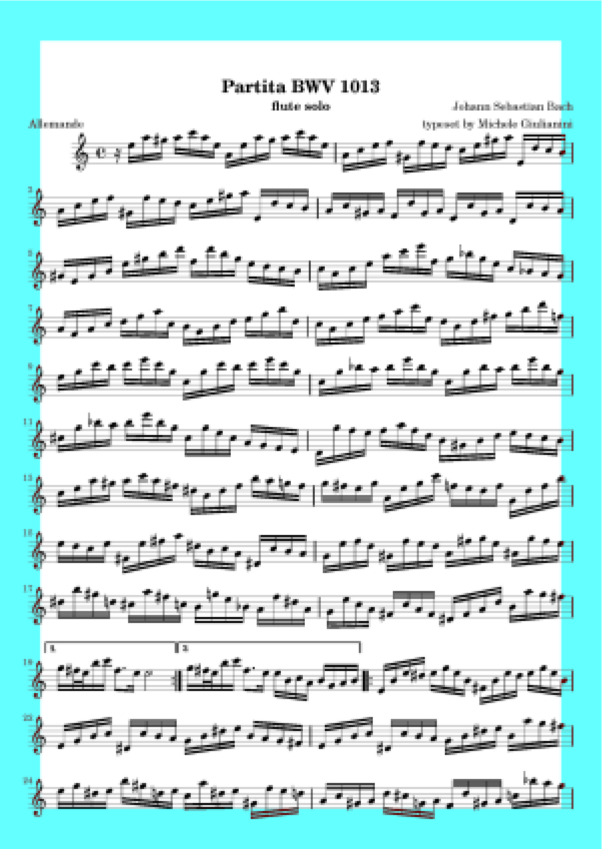
```
<score-partwise version="3.0">
      <identification>
            <encoding>
                  <software>audiveris 4.3.3406</software>
                  <software>ProxyMusic 3.0.110</software>
                  <encoding-date>2014-05-28</encoding-date>
            </encoding>
            <source>./carmen-1.png</source>
      </identification>
      <defaults>
            <scaling>
                  <millimeters>7.1120</millimeters>
                  <tenths>40</tenths>
            </scaling>
            <page-layout>
                  <page-height>1667</page-height>
                  <page-width>1200</page-width>
                  <page-margins type="both">
                        <left-margin>80</left-margin>
                        <right-margin>80</right-margin>
                        <top-margin>80</top-margin>
                        <bottom-margin>80</bottom-margin>
                  </page-margins>
            </page-layout>
            <lyric-font font-family="Serif" font-size="10"/>
      </defaults>
```

Layout descriptions in MusicXML are provided in relative form. To abstract from the digital representation, distances are not measured in pixels but in partials of the measured staff height. This information is provided in the scaling section. Everything is measured in tenths of staff space.

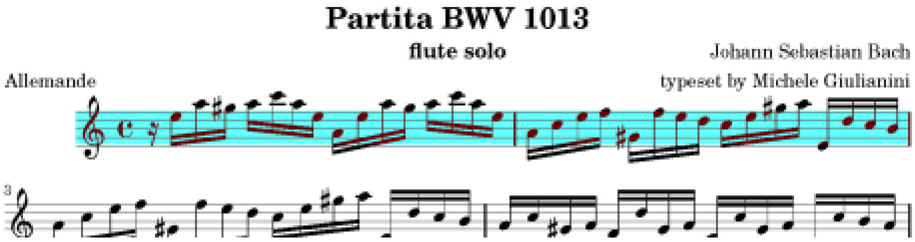
Tenths are then scaled to millimetres within the scaling element. Margins, page sizes, and distances are all measured in tenths to keep MusicXML data in a consistent coordinate system as much as possible. The translation to absolute units is done in the scaling element, which specifies how many millimetres are equal to how many tenths. For a staff height of 7 mm, millimetres would be set to 7 while tenths is set to 40. The scale factor for digital images can be calculated as follows:

scale = (millimetres * dpi) / (tenths *millimetres_per_inch)

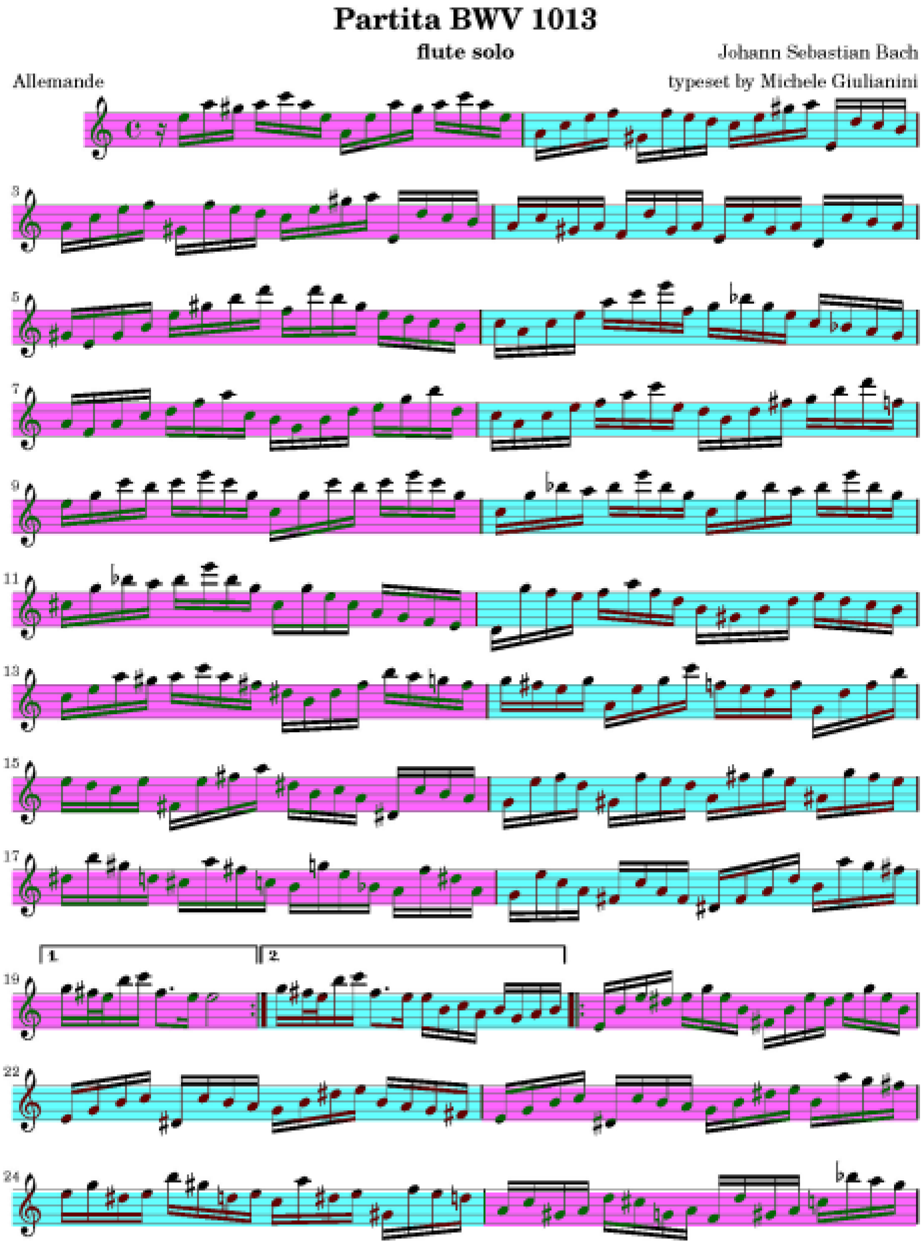
where *dpi* corresponds to the dots per inch (DPI) value of the image and *millimetres per inch* to the constant value of 25.4. Applying this scale factor to the layout description of the provided excerpt can be used to highlight the initial layout elements within the scanned page:



Further score elements have to be handled system and staff wise. For example the following image highlights the first staff of the first system of the page:



Having all information extracted from the layout sections of the MusicXML file it is possible to create a complete mapping of the measures provided on the page of the score:



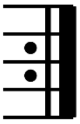
This information will be essential in the final visualisation step. After the two music representations have been aligned, the spatial coordinates of the measures are used to highlight the currently played measure.

4.2 Concatenating multiple score pages

Depending on the use-case and applied scenario, it might be necessary to combine separately processed pages into a complete digital representation of the original music score. Most OMR software are capable of processing a set of input files, multi-page documents or single pages. In some scenarios digitised images of scores have to be processed separately. This results in distinct MusicXML files, which have to be merged into a combined score. During the concatenation of the pages, care needs to be taken to retain their layout information. Furthermore the indexes of the measures indicating a page turn event have to be determined.

4.3 Processing repetitions

Repetitions are an important musical concept where sequences are repeated. In music scores repetitions are indicated by repeat signs:



These signs are detected by OMR software and indicated in the MusicXML file. Proper detection in OMR and extraction from MusicXML is crucial for the proper functioning of a score-following system. Missing a repetition in the score, leads to different length of the two music representations. The score-following system expects to map a missed repetition only once while in the audio recording it is played twice. This will lead to misaligned scores and a messed up timing during the visualisation of the scores.

5 Feature extraction

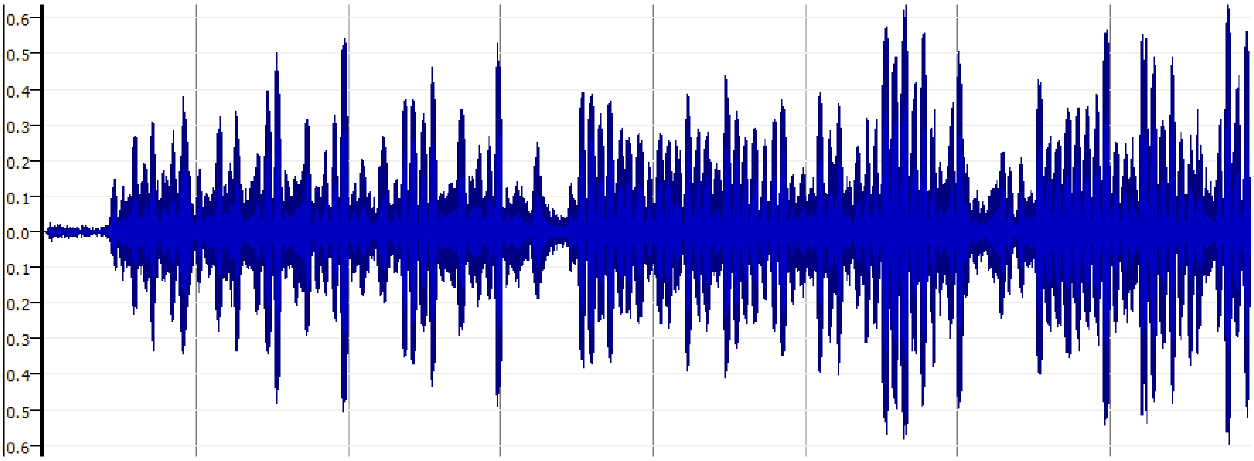
Feature extraction is the process of calculating distinctive values from either the recorded audio or from the symbolic music. Features aggregate information and provide a description of the corresponding track. Depending on their complexity, music features can be used to capture information about music timbre, rhythm, onsets, tempo, etc.

5.1 Audio feature extraction

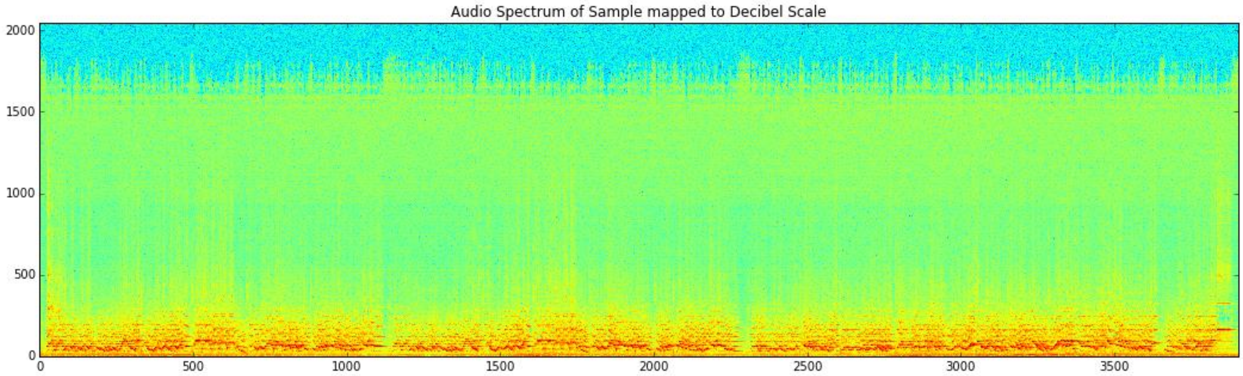
A typical CD quality mainstream radio track has an average length of three minutes. This means, that the song is digitally described in Pulse-code Modulation (PCM) by almost 16 million numbers (3 [minutes] x 60 [seconds] x 2 [stereo channels] x 44100 [sampling rate]). This information requires 30MB of memory and a considerable amount of time to process. Processing the small number of 100 tracks, which relates to about 10 audio CDs, would require about 3GB of memory, which is currently about the average size of memory provided in personal computers. Processing 100000 songs would require 3TB of memory, which requires vast resources (e.g. acquisition, hosting, energy consumption, etc.) and is only possible in an academic or industrial setting. Consequently, there is a strong desire to reduce the information provided in an audio track and to distil it into a smaller set of representative numbers that capture higher level information about the underlying track.

5.1.1 General audio feature extraction

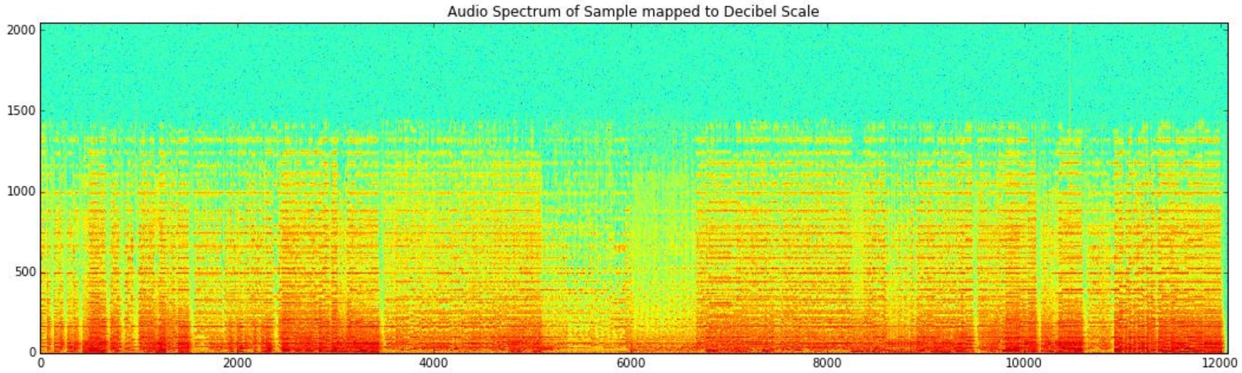
The following image depicts the waveform of the recorded audio. The lines represent the sampled sound energy at discrete points in time. This representation provides an overview of the distribution of energy over time. Events with high loudness values are observable which might be interpreted as onsets of a tune.



Waveforms represent sampled audio in the time domain. To apply more complex analysis to the audio it has to be transformed to the frequency domain. This is usually accomplished through applying a Fourier Transform to the sampled audio. This essential part of any audio feature extraction algorithm is based on the Fourier's postulation that every complex continuous signal can be explained by decomposing a wave into its component frequencies and phases. This is a lossless transformation which can be reversed through applying an Inverse Fourier Transform Transform. The result of the Fourier transform can be visualised through a spectrogram:



This spectrogram shows the distribution of the spectral energy of Johann Sebastian Bach's *Partita for Flute allemande* (BWV 1013). This is a monotonic track played by a single flute. Thus, it is possible to observe the tune in the spectrogram. It is depicted by the red regions in the range from 0 to 2500Hz. This is not possible in polyphonic Music. This following image shows the spectrogram for Johann Sebastian Bach's *Toccata and Fugue in D Minor* (BWV 565).



Although this track was also performed by playing a single instrument, it is a polyphonic track and the energy is distributed across multiple bands of hearing. A primitive segmentation of the track is still observable, but no more conclusions about melodies or differentiation between accompaniment and leading voice can be drawn.

This representation still provides too much information for automatic processing. Common scenarios in music information research performing statistical prediction to classify, index or recommend music tracks. These methods are usually computationally demanding. The processing time is generally related to the dimensionality of the data used to calculate certain music related properties. Data reduction is based on transforming the spectrogram and sequentially calculating descriptive measures, which are aggregated by calculating statistical moments to capture their deviations over time.

5.2 Symbolic music feature extraction

Audio feature extraction has to assess information about played notes in a pre-processing step through signal processing. The result is often unsatisfactory. Thus, many audio features do not attempt to capture the played notes and just provide statistical descriptions of the audio spectrum. Symbolic music feature extraction has the advantage that played notes are precisely defined. The disadvantage is that timbre information is missing, which is yet not relevant for audio-to-score alignment.

5.3 Audio-to-score alignment relevant music features

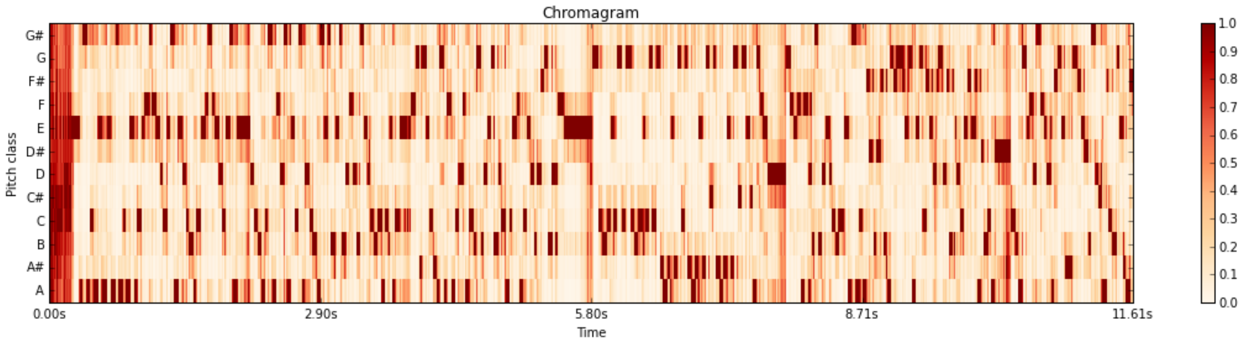
Audio-to-score alignment has the challenge that two completely different digital representations of music have to be compared and aligned. Thus, both forms have to be transformed into the same representations. This is usually accomplished through Chroma features.

Chroma Features:

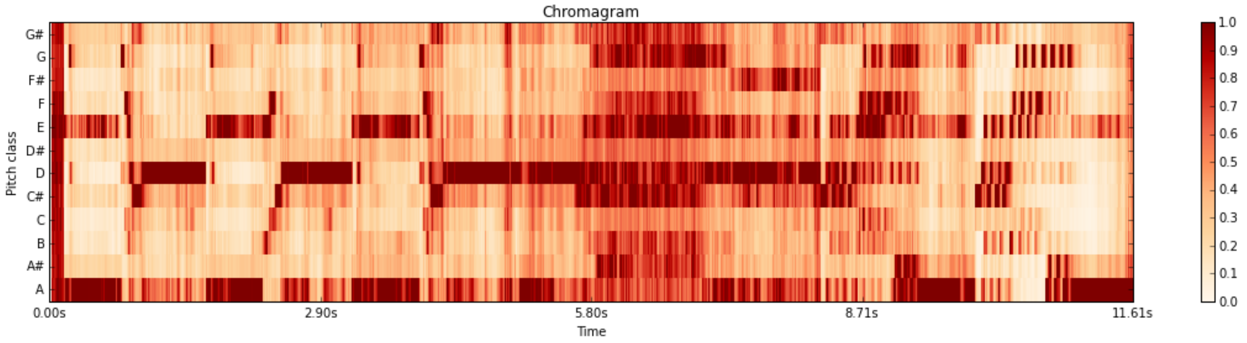
Chroma Features represent the 12 distinct semitones (or chroma) of the musical octave. This results in one or a sequence of twelve dimensional vectors where - for example - the bin that corresponds to the pitch class A captures the spectral energy of A0 and all its corresponding sub-band pitches A1, A2, etc.

Chroma from Sampled Audio:

The calculation is accomplished by sequentially projecting the entire spectrum onto these 12 bins through decomposing the audio signal into 88 pitch sub-bands and calculating the sum of all sub-bands belonging to the same pitch class. The following image visualizes the Chromagram of Johann Sebastian Bach's *Partita for Flute allemande* (BWV 1013):

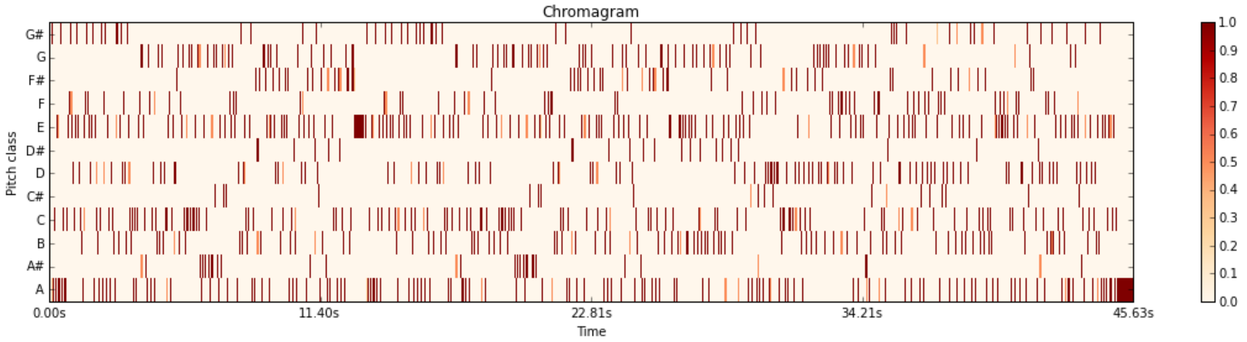


This track is monotone and played by a single instrument. In this special case, it is possible to see the played tones in the chromagram. Usually music is polyphonic, which is already the case when playing two tones simultaneously. The following Chromagram depicts Johann Sebastian Bach's *Toccata and Fugue in D Minor* (BWV 565):

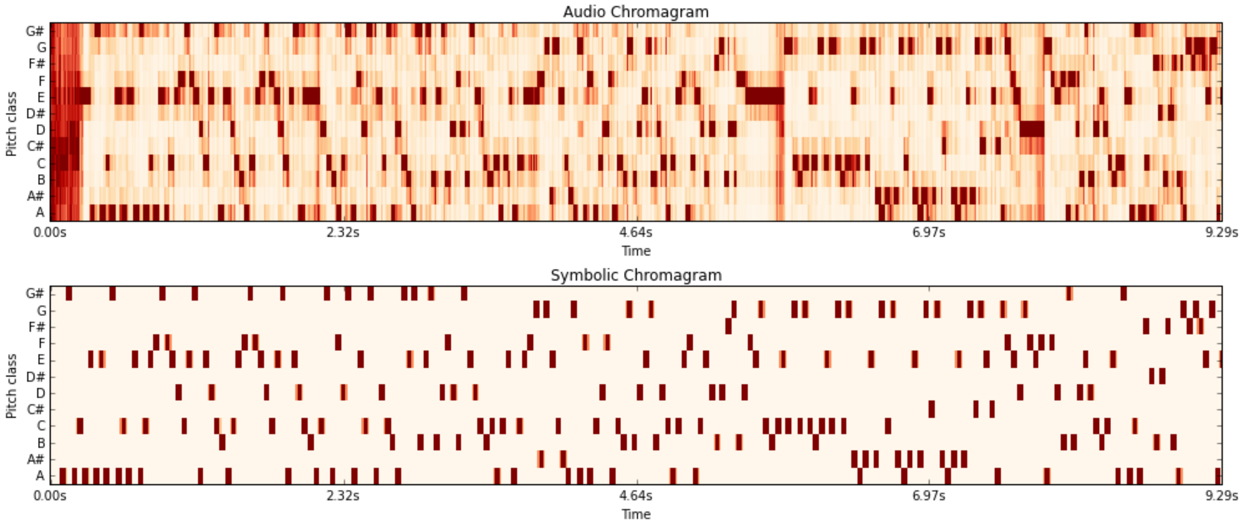


Chroma from Symbolic Music:

Chroma features can be easily extracted from MusicXML or MIDI. The following image shows the Chromagram of the MIDI representation of Sebastian Bach's *Partita for Flute allemande* (BWV 1013).



It can be observed that the Chromagram of the MIDI representation is clean and distinct compared to Chroma features extracted from sampled audio. This is because the corresponding note of a feature is described by the music symbols instead of being mapped from a frequency distribution. The following image compares the Chromagrams of chroma features extracted from sampled and symbolic music. Similar structures can be observed:



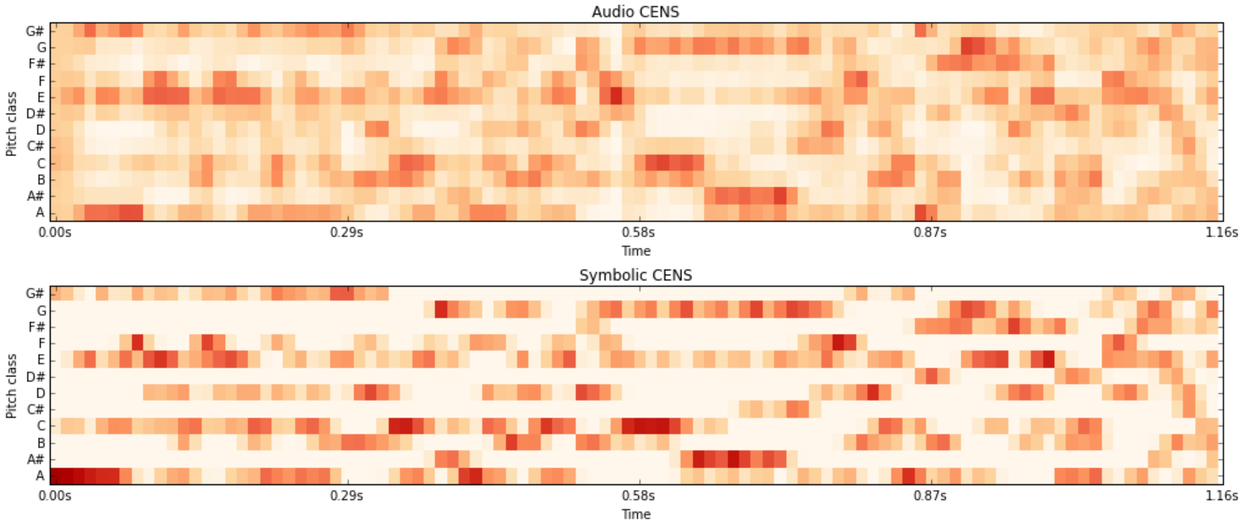
CENS Features:

Symbolic music and recorded audio may diverge significantly. Besides individual interpretations and variations of the composition, variations in tempo, onsets, instrumentation, tuning, etc. are common. To introduce further degrees of robustness against such variations, the Chroma Energy Normalized Statistics (CENS) features have been described.

CENS features are calculated from Chroma features. Thus, calculation for Chroma features extracted from symbolic music is identical to their calculation from audio based Chroma features. First the features are normalised against differences in sound intensity and dynamics. Using short-time statistics the Chroma energy distribution is calculated over a larger analysis window, to compensate for variations in tempo and articulation. These transformations provide the following advantages for audio-to-score alignment systems:

- Correlation to the harmonic progression of the underlying music
- Invariant to variations in dynamics
- Robustness to variations in timbre and instrumentation
- Compensations to variations in time and different realizations of note groups (e.g. trills, arpeggios, etc.)

The following image shows the CENS features calculated for sampled and symbolic music side by side:



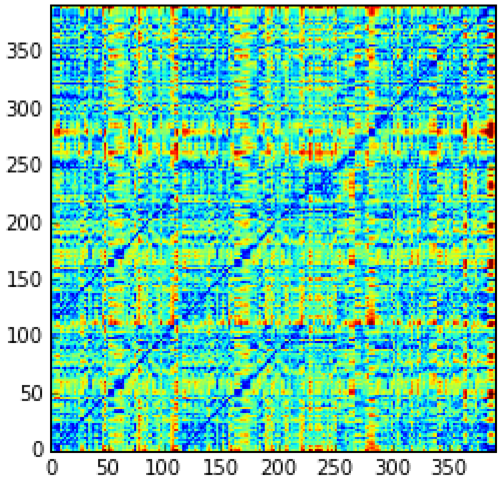
6 Audio to score alignment

The previous steps provided relevant information through extracting music information from either the digitised images as well as the sampled audio. This chapter covers the actual processing of this data to align the two different representations.

Up to now the two sources have been converted into the same representation - the Chroma features, respectively their normalised form - the CENS features. Yet, different lengths of the source files or different resolutions of the analysis windows result in different number of features extracted. Thus, the initial position is that two lists of Chroma features have to be matched. It has to be assumed that these lists are of different length and that the opponent items do not match. The aim is to find matching items in both lists. A well-known technique to accomplish this task is called Dynamic Time Warping.

6.1 Dynamic time warping

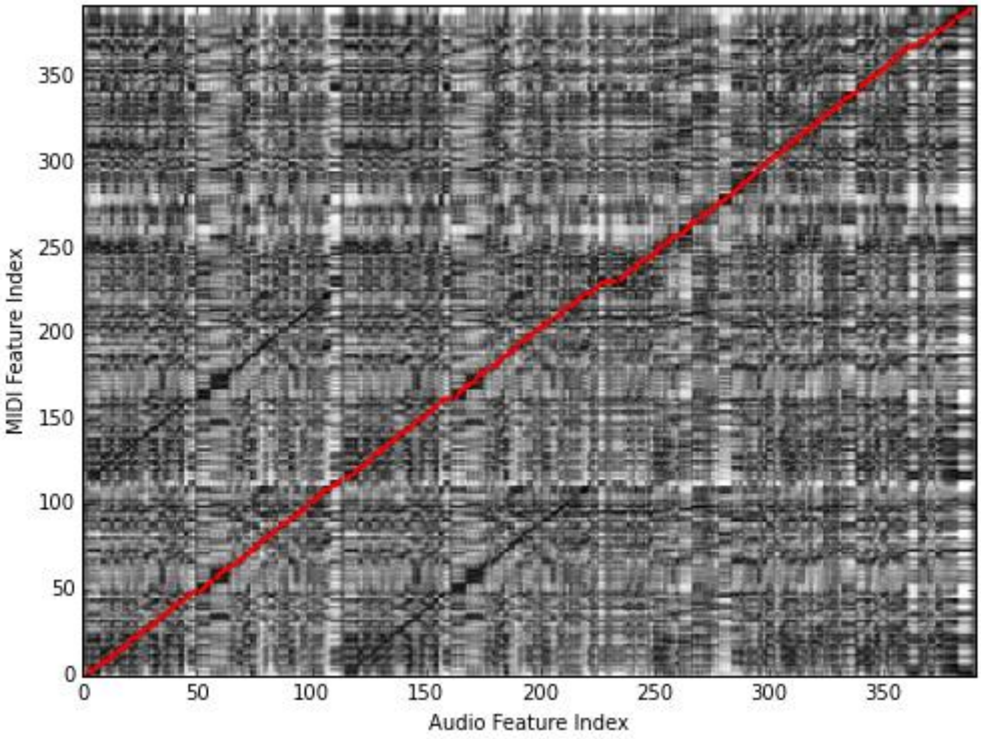
Dynamic Time Warping (DTW) is an algorithm for time series analysis originating from automatic speech recognition. It is intended to find an optimal alignment between two given sequences. Based on a local distance measure, the two sequences of Chroma features are compared. First a distance matrix for all elements is calculated.



By recalling that the elements of the lists are numerical vectors where each element of this such a vector represents the energy distribution of one of the 12 pitch classes, the vector distance of two vectors can be interpreted as a measure of musical dissimilarity. The higher the distance between two Chroma vectors, the less similar their musical source is in respect to the measured pitch classes.

By interpreting the values of the distance matrix as costs, where most similar elements have low costs and identical elements have zero costs, the goal of DTW is to find an alignment between the Chroma features extracted from sampled audio and those extracted from symbolic music, that has minimal overall cost. Because the mapping is time-dependant the following three conditions have to be met:

- **Boundary Condition:** the warping path through two lists of length m and n has to start at the position (1,1) and to stop at position (m,n). In case of audio-to-score alignment this guarantees that both representations - the score and the recording - start at the very beginning and stop at the end of the tracks.
- **Monotonicity Condition:** the indexes of both sequences should increase monotonically. Infringement of this condition would result in mappings that run backwards in time.
- **Step Size Condition:** This condition demands that the path progresses continuously and ensures that each element of one list is mapped to an element of the other and vice versa.



6.2 Assign audio-timecodes to measures from the music score

Having achieved an alignment between the Chroma feature sequences extracted from symbolic music and sampled audio, this alignment is used to create a concrete mapping between image regions and time events in the audio file. The image regions correspond to the identified measures of the music sheets (see Section 5) and audio time events to discrete sections of the recording. The mapping is accomplished through back-mapping the indexes of the aligned Chroma vectors to the previously extracted information.

6.2.1 Mapping Chroma features to audio

For the audio part this is a trivial approach based on solving a linear equation. To calculate the Chroma features the sampled audio is transformed into the frequency domain. Typically a short-time variant of the Fourier transform is used with a fixed window size. Based on this size and the sample rate of the recording, the length of the audio segment that is described by a single chroma vector can be estimated.

6.2.2 Mapping Chroma features to measures

The mapping of the Chroma vectors is more complicated and contains spatio-temporal relations that have to be considered.

First, the Chroma indexes have to be mapped to the digital representation of the music score, which can be MIDI, MusicXML or any other format. The features have been calculated from this representation. There are different ways to approach this problem. If the measure boundaries are stored in the digital representation of the symbolic music, the mapping of the indexes to the corresponding measures is trivial. If this information is not available, a good approximation is to divide the number of indexes by the number of measures and assign each measure its corresponding number of Chroma vectors. This approach is legitimate due to the virtual fixed tempo of a symbolic music track. Thus, the symbolic music can be assumed to be the invariant, deterministic part in relation to recorded audio.

The second part consists of mapping the music data to regions on different images. This strongly relies on the mapping that has been created in previous steps. It requires the following properties:

- distinctive measure IDs that are linked to the symbolic music
- continuous numbering of the measures identified in the images
- correct assignment of measure IDs to page numbers

Based on this the symbolic indexes can be linked with the measure indexes of the images. Special attention has to be set on repetitions, which have to be decomposed into a sequential list of measures.

7 Graphical User Interface

The last major component in a score-following system is the audio-visual representation of the calculated results. This requires a framework that can interact and manipulate audio events to either trigger visualisation routines or jump to distinct sections of the audio file.

7.1 HTML5 framework

The current major version of the HTML stack implements all necessary requirements to realise a score-following interface. It provides functions to manipulate images and to interact with audio. These components are glued together through Javascript code to facilitate the interactive synchronisation during the replaying of the audio.

HTML5 Audio

The audio-tag is a relatively new but powerful addition to the HTML specification. It provides an interface to capture audio events including access to the audio stream itself. This means that the audio processing and feature extraction algorithms could theoretically be executed within the webpage without any requirement for backend processing except for storing the results.

The HTML5 audio-tag is used to play the audio file. The API is used to listen to audio events which contain information about the audio segment that triggered this event. Based on the timestamp of this event the corresponding page and measure number can be retrieved from the previously generated look-up table. This information is passed on to visualisation routines that highlight the currently played measure.

Visualisation of Currently Played Measures

Interactive visualisations on Web pages are easy realised through Cascaded Style Sheets (CSS). Based on the extracted information about the spatial location of the measures on the distinct images, it is possible to superimpose regions on the images to highlight the underlying content. The Javascript library

Annotorious has been developed for such tasks. Annotations highlighting image regions can easily be defined based on relative coordinates within the image. This open source library was further adopted to meet the requirements of a score-following system. A click-handler was added to capture click events on annotations. These events trigger external Javascript functions to look-up the timestamp in the audio file via the index number of the clicked measure and to jump to the corresponding sequence within the audio stream. Further functionality has to be added to handle page-turn events. Again, special attention has to be paid to repetitions.

8 Requirements and pre-requisites for development

This section describes the technical requirements and dependencies to develop and implement an audio-to-score alignment or score-following system.

8.1 Development environment

The prototype was developed in the programming language Python[18]. This decision was based on the intrinsic properties of the Python language, which facilitates rapid prototyping and experimenting. This especially is the case for scientific computing areas such as image processing and music information retrieval. The Numerical Python (Numpy)[19] module is aligned to Matlab[20] and provides a similar interface to numerical computations. Language, interpreters and the majority of the modules are open source.

The prototype was developed within an IPython Notebook[21] environment, which is an interactive Python shell within a Web browser. This environment is a highly donated project that currently benefits from appreciation in many scientific communities due to its capabilities of capsuling the possibilities of experimentation, development, visualisation and documentation within a single environment.

8.1.1 Python libraries required for processing

The following mentioned Python modules are required to read and process data. These include music data input modules to read symbolic and sampled music as well as modules for data analysis and signal processing.

Numpy - Numerical Python

NumPy is the fundamental package for scientific computing with Python. It contains among other things:

- a powerful N-dimensional array object
- sophisticated (broadcasting) functions

[18] https://www.python.org/
[19] http://www.numpy.org/
[20] http://www.mathworks.com/products/matlab/index.html?s_tid=gn_loc_drop
[21] http://ipython.org/notebook.html

- tools for integrating C/C++ and Fortran code
- useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

License: BSD License

Usage in Audio-to-Score alignment:

- feature calculation
- synchronisation
- similarity estimations
- layout calculations

Scipy - Scientific Python

SciPy is a Python-based ecosystem of open-source software for mathematics, science, and engineering, including modules for linear algebra, signal processing and optimization. SciPy builds on the Numpy stack which has similarities to commercial analytical applications such as Matlab.

License: BSD-new license

Usage in Audio-to-Score alignment:

- audio signal processing
- distance and similarity calculations and aggregation

Music21

Music21 is a toolkit for the manipulation and analysis symbolic music initially intended for helping scholars and other active listeners answer questions about music quickly and simply. Its strength is its ability to work with many common formats including MIDI and MusicXML.

License: Released under either the BSD (3-clause) or GNU LGPL license according to your choice.

Usage in Audio-to-Score alignment:

- loading and interpreting MusicXML

Librosa

A python package for music and audio analysis. The primary purpose of librosa is to implement common tools for low- and high-level signal-based music analysis.

License: ISC license

Usage in Audio-to-Score alignment:

- audio feature extraction
- symbolic feature extraction
- calculation of Chroma features

Pretty-Midi

pretty_midi contains utility function/classes for handling MIDI data, so that it is in a format which is easy to modify and extract information from.

License: MIT License

Usage in Audio-to-Score alignment:

- symbolic feature extraction
- calculation of Chroma features

Python MIDI

Python MIDI provides easy means to manipulate MIDI data. This toolkit aims to provide a high level framework that is independent of hardware to manipulate, sequence, record, and playback MIDI. It further provides multi-track aware container, allowing you to manage MIDI events, tempo maps that actively keep track of tempo changes within a track, as well as readers and writers to read and write MIDI tracks to disk.

License: MIT License

Usage in Audio-to-Score alignment:

- output symbolic music as MIDI tracks

8.1.2 Python libraries required for visualisation

Open Computer Vision library for Python (OpenCV)

OpenCV[22] (Open Source Computer Vision) is a library of programming functions for image processing and computer vision. OpenCV was designed for computational efficiency and with a strong focus on real-time applications.

License: BSD license

Usage in Audio-to-Score alignment:

- loading score images
- highlighting measures and notes

[22] http://opencv.org/

Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms.

License: Python Software Foundation license

Usage in Audio-to-Score alignment:

- visualising music features extracted from symbolic and sampled music

8.2 Content requirements

A reliable implementation of a score-following feature can only be realised if both of the following two content related criteria are met:

- Access to high-quality scans of printed sheet music for a particular musical work.
- Access to a music recording of the same musical work as represented in the sheet music

The audio-to-score alignment workflow would then have what it requires to match sheet music print to corresponding music recording. Please note that the two objects do not need to exist in the same collection - one could come from institution X and the other from institution Y.

8.2.1 Technical dependencies

Optical Music Recognition (OMR) is a relatively young research topic compared to Optical Character Recognition (OCR). While recent OCR systems are mature and capable of detecting and extracting text from digitised images of low quality, OMR systems still require higher quality images as input. The following paragraphs summarise the most severe challenges concerning the application of OMR to scans of old and low quality music scores:

General image content restrictions

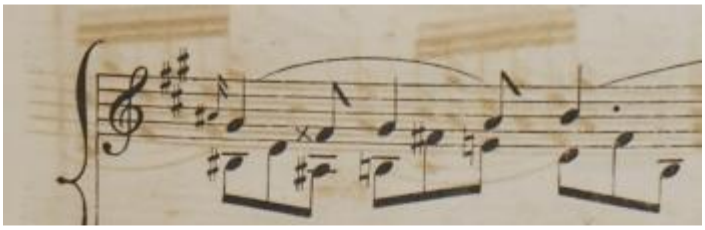
Suitable quality (for OMR) can be considered as images of music sheets using modern staff notation with contemporary fonts and symbols. Staff lines are straight and horizontally aligned. The sheet is cropped and centred to show only the content of the score, having all overlapping space of the scan removed. Content curated and provided by national libraries is usually older. Especially piano or orchestral music printed before 1900 provides many problems. The following content type is currently not or only supported by highly specialised software - most of them were developed in other research projects and require considerable effort to adapt.

- **Handwritten Scores:** Contrary to OCR where some handwritten content can already be recognised, no OMR system is currently capable of processing non-printed sheet music.
- **Early Music Notation:** Specialised versions of OMR are available to process early typographic music. The most advanced system is ARUSPIX which is also evaluated in the Europeana Cloud project. An integrated system providing the capability to process multiple types of music scores is currently not available.

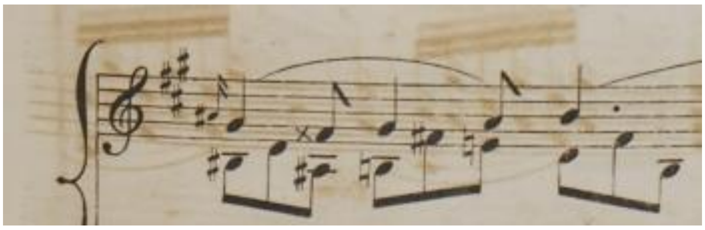
Quality of the paper

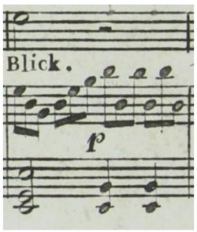
A set of definitions and two models concerning document image quality and degradation are provided by Baird. Balk and Contech summarise the finding of the IMProving ACcess to Text (IMPACT) project, which focused on the development of new approaches to the extraction of text content from historical documents. Thirty-seven characteristics which can affect OCR performance were identified, including bleed-through, stains, page, curl, broken characters, low contrast, skew, and presence of watermarks. All of them have the similar effects on OMR performance.

- **Bleed-through:** caused by degradations of ink and paper, the content of the adjacent page becomes visible. Binarisation is a typical pre-processing step to remove such artefacts. Based on the dimension of the effect, specialised approaches might be required. The provided example depicts the problems of strong bleed-through artefacts. While the binarised image still contains fragments of the undesired content, necessary information such as staff lines, which are fundamental to the OMR extraction process, is destroyed.

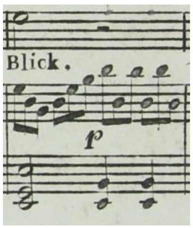


- **Stains, Scratches and Watermarks:** Missing or additional content on music sheets cannot be interpreted correctly on will provide erroneous results.





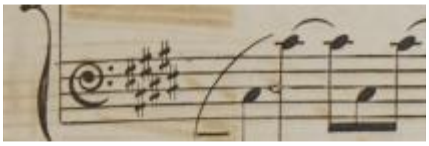
- **Low contrast:** Low contrast becomes a problem in combination with other artefacts such as bleed through. Pre-processing steps to remove these artefacts also degrade the quality of low contrast scores. The example image shows eight notes with bleached ink. Binarisation is expected to remove the inner regions of the notes which makes them look similar to semi notes.



Quality of the scans

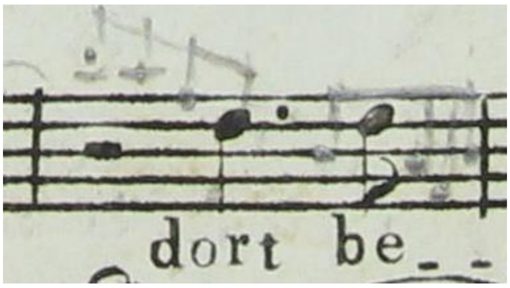
- **Layout detection:** Experiences from evaluating OMR engines in Task 2.4.1 showed that most systems have problems with additional content in scans such as black borders which typically arise at non-page regions of scans. Consequently, digitised scores showing these kinds of artefacts require either manual pre-processing or the implementation of an automatic cropping algorithm.

- **Image rectification:** Images are often distorted non-linearly, especially in bow areas of books. Built-in algorithms to locate lines of text such as implemented in OCR systems are still ongoing research.

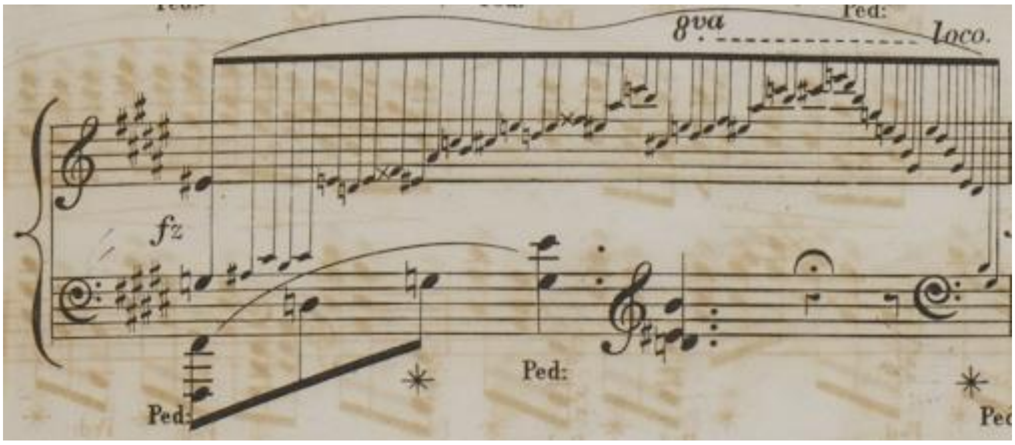


Quality of the scores

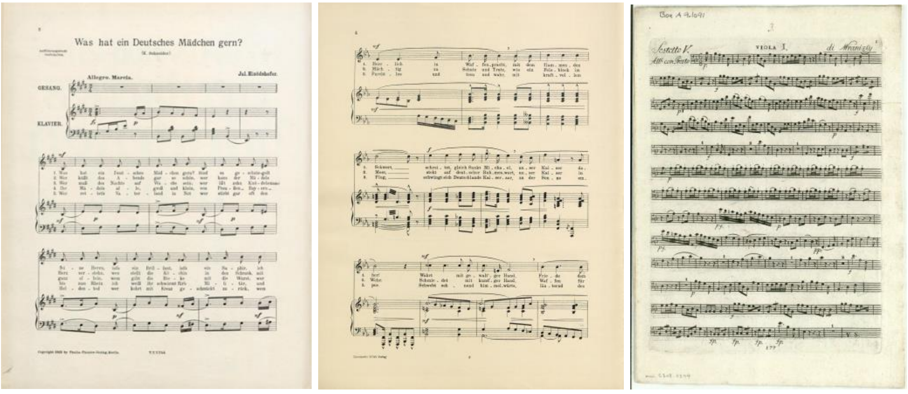
- **Atypical fonts:** OMR systems are optimised to printed music sheets, preferably printed with a standard computer font which has been available since the late 1960s. Older scores for example use different symbols for clefs and also the layout of the notes differs.



- **Manual annotations:** Manual annotations of historically important persons are worth preserving, but in terms of OMR they fall into the same category as stains and scratches.
- **Complexity of the score:** Piano music is often quite complex and dense, which makes it hard for OMR systems. Orchestral scores are often small, resulting in not enough pixels for distinguishing different symbols.



Examples of digitized scores of appropriate quality:



Quality of audio recordings

The problems of sub-optimal audio quality are less severe than those of erroneously extracted music information from images. Urbano et al. [REF 5] analysed the effects of audio quality on Chroma features and concluded that they appear to be robust against variations from different sampling rates, codecs and bitrates of the same audio recording. Mauch and Ewert [REF 6] evaluated the effect of audio degradation on different music information retrieval tasks including audio-to-score alignment. They developed a Matlab toolbox to artificially apply degradation effects to audio files, thus facilitating controlled experiments. Simulated degradation effects included:

- **Live recordings:** Echo of the room and pink noise of the audience
- **Radio Broadcast:** Dynamic Range compression, Speedup (commonly applied to music in mainstream radio)
- **Smartphone Playback:** pink noise
- **Smartphone recording:** Dynamic Range Compression (auto-gain effect), Clipping and pink noise
- **Strong MP3 compression:** 64kpbs
- **Vinyl:** Crackling sounds, wow-and-flutter, light pink noise

The authors based the evaluation on a similar approach as evaluated in Task 2.4.1. They used a simplified setting without optical music recognition. Symbolic music was available in form of high quality MIDI files. For the evaluation the Saarland Music Data set[23] was used, consisting of 50 piano tracks played on a Yamaha Disklavier MIDI piano.

Based on the results of the experiments the authors concluded that the applied audio-to-score alignment method is generally robust against audio degradation effects, except for the categories *Live* and *Smartphone Playback*.

[23] http://resources.mpi-inf.mpg.de/SMD/SMD_MIDI-Audio-Piano-Music.html

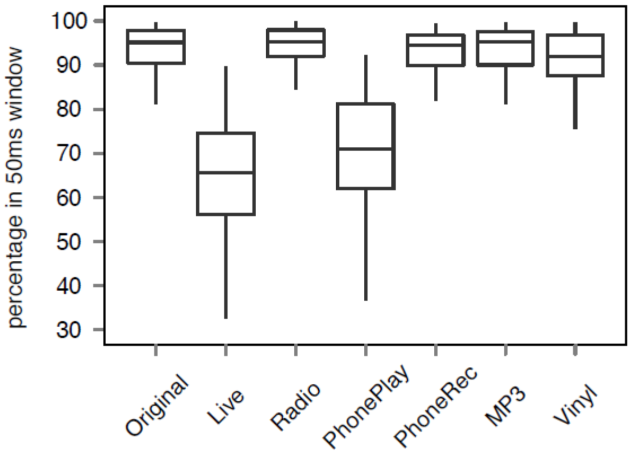


Figure 4. Score-to-audio alignment accuracy under the applied audio degradation effects. The boxes indicate the 1st, 2nd (median) and 3rd quartiles, the whiskers extend to 'the most extreme data point which is no more than 1.5 times the interquartile range' (taken from Footnote 8).

While the latter category might not be of concern for Europeana Sounds content, audio properties described by the *Live* category are more likely to be present. Based on these results a general reliable applicability of the automated mapping approach evaluated in Task 2.4.1 cannot be stated and further adaptations to normalise audio recordings towards their quality should be considered.

Error Categories

The following table lists some issues that are expected to arise frequently using Europeana content for score-following.

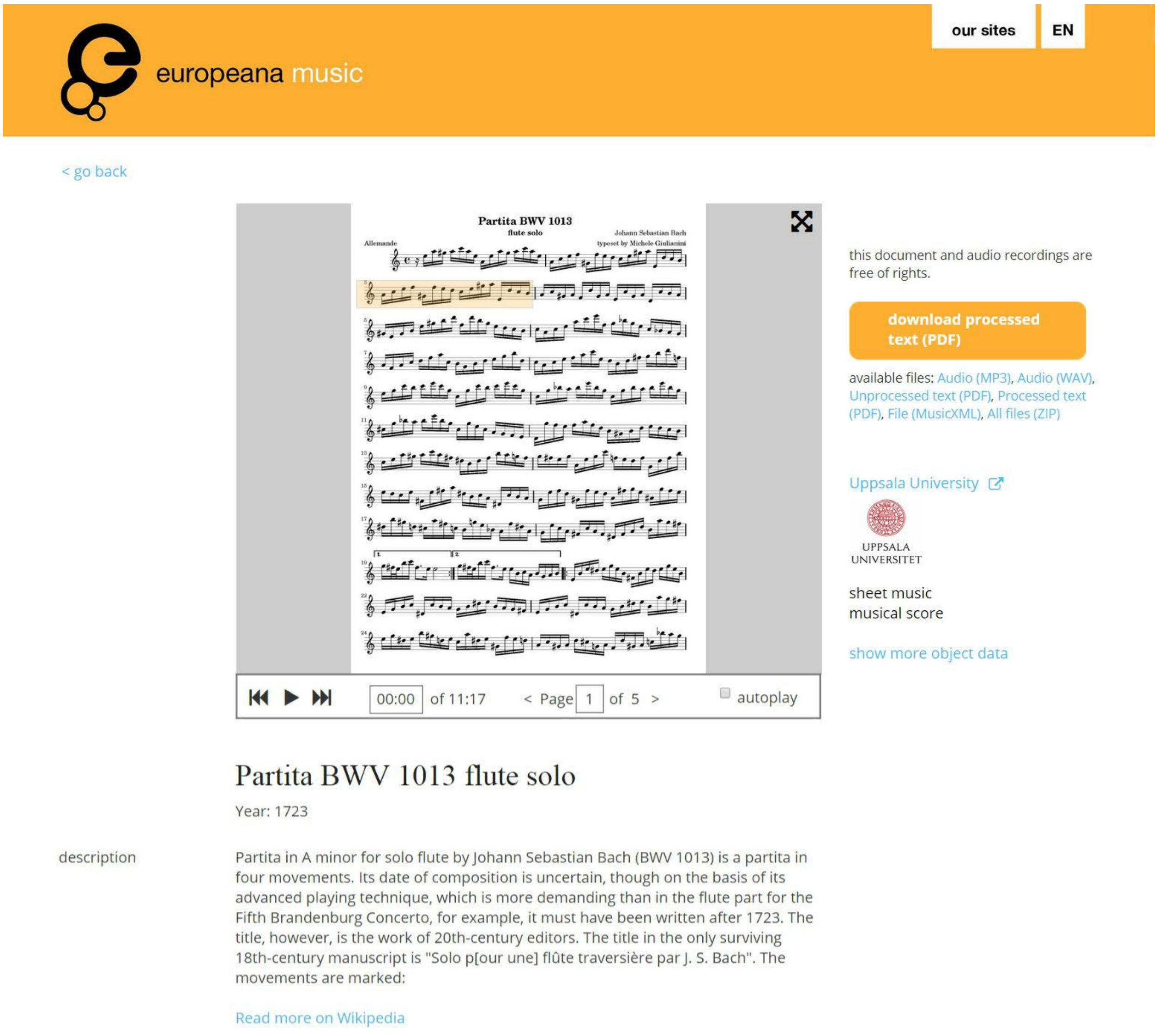
Table 1: Error categories

Type	Description	Consequence	Severity
OMR	Measure boundary not detected	The sum of note values exceeds the expected time signature. This might lead to erroneous alignment and incorrect visualisation.	Middle
	Systems not detected correctly	Systems interprets that the staff lines of a system are independent and should be played in consecutive sequence instead of being played in parallel. This leads to misalignments due to the additional measures.	High
	Repeat sign not detected	Leads to misalignments due to missing measures. If complete score has to be repeated and the sign is missed, the complete audio has to be mapped to half of the measures.	Very high
	Note lengths are misinterpreted	The sum of note values varies from the expected time signature. This might lead to erroneous alignment and incorrect visualization.	Middle
	Note values are misinterpreted	This leads to deviations in the music features and might affect correct alignment. Because the features are normalised, a small amount of	Minor

		misinterpretations can be neglected.	
	Missing staff lines or measures	Leads to misalignments and wrong timing in the visualisation.	Middle
	Additional staff lines or measures	Attempting to map non existing content to recording leads to misalignments.	High
	Wrong time signature	Sum of note lengths within measures does not correspond to the detected time signature. Might result in erroneous results.	High
	Wrong clef or wrong key / accidentals	Leads to wrong calculation of music features. The feature values are shifted. If the clef is globally misinterpreted, the alignment will still work.	Medium
Audio	Constant noise from old records	Affects audio feature calculation and mapping. If noise remains the same for the whole recording the mapping will still work.	Low
Audio	Varying noise	Affects audio feature calculation and mapping. Might lead to wrong mapping.	Middle
	Audio Watermarks	Artificially altered content used to protect property rights. Will lead to wrong mapping.	High
	Fragmented Audio, no overlaps	Such as two sides of a record. Additional effort has to be provided to implement proper handling of audio concatenation and partitioning of the score	Low
	Fragmented Audio with overlaps	Such as magnetic tape recordings with redundant overlaps. Additional effort has to be provided to correctly align the truncated audio pieces to combine them into a complete recording.	High
	Fragmented Audio, missing content	Will lead to misalignments. Additional effort has to be planned to implement methods to identify missing content and correctly annotate these regions in the score images.	High

9 Results

A simple wireframe has been created that shows how a “playable score” could be displayed. The wireframe is in the same medium-fidelity style as in MS20 *Second Audio Channels Prototype*.



this document and audio recordings are free of rights.

available files: Audio (MP3), Audio (WAV), Unprocessed text (PDF), Processed text (PDF), File (MusicXML), All files (ZIP)

Uppsala University

UPPSALA UNIVERSITET

sheet music
musical score

show more object data

Partita BWV 1013 flute solo

Year: 1723

description

Partita in A minor for solo flute by Johann Sebastian Bach (BWV 1013) is a partita in four movements. Its date of composition is uncertain, though on the basis of its advanced playing technique, which is more demanding than in the flute part for the Fifth Brandenburg Concerto, for example, it must have been written after 1723. The title, however, is the work of 20th-century editors. The title in the only surviving 18th-century manuscript is "Solo p[our une] flûte traversière par J. S. Bach". The movements are marked:

Read more on Wikipedia

Figure 5. Screenshot of a wireframe displaying the score-following player incl. download options.

As part of the score-following, the current movement through the musical work played is highlighted in the sheet music. The player is provided with controls both for the audio recording (play, pause, fast forward, fast backward, time code, autoplay) and the sheet music (pagination, full screen).

The various files produced during the feature extraction needed to support score-following are made available for download. While simple, the wireframe hence supports both use cases listed above.

Prototype score following player

An overview of playable prototype examples[24] has been created in which you can play the sound recording with score following.

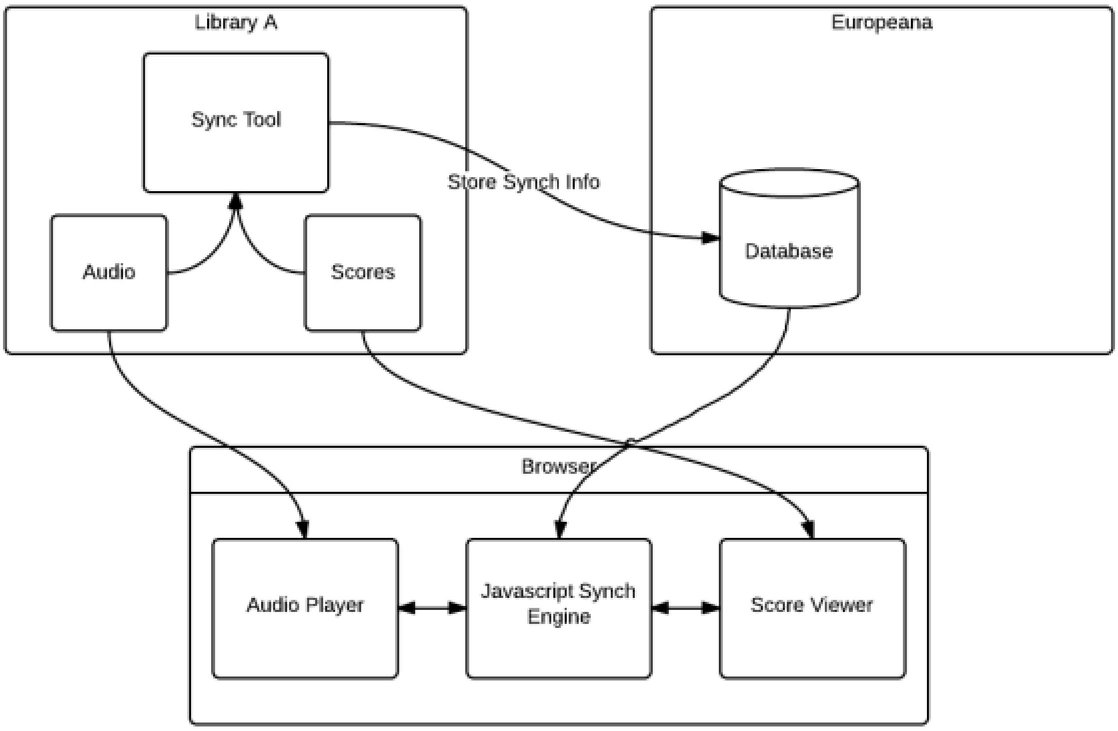
10 Considerations towards integration

While a deployment architecture of the Score Following Pilot has not been formalised, we will make the assumption that the back-end modules of the Score Following Pilot would need to be deployed centrally in the Europeana infrastructure. While it is theoretically possible to install the Score Following Pilot in the collection management systems of all relevant partners in Europeana Sounds, this would require significant development effort for each such partner.

10.1 Integration scenarios

10.1.1 Scenario 1

Partner library A has physical access to the audio recordings and scanned scores. The audio-to-score alignment tool is executed on site and handles all steps (OMR + feature extraction + synchronisation). Results are uploaded and stored in a database hosted by Europeana. This information would only consist of time signatures and coordinates within the images. In this scenario the intermediate files (e.g. the MusicXML output) could be stored, but are not really required.



[24] http://ngcns-demo.ait.ac.at/score_following/index.html

Advantages

- Content resides at the partner's sites, which amends copyright concerns
- Partners intimate knowledge of their own collection can be leveraged
- Collections can be processed in batches.
- Single site of processing
    - reduces points of failure
    - reduces overhead

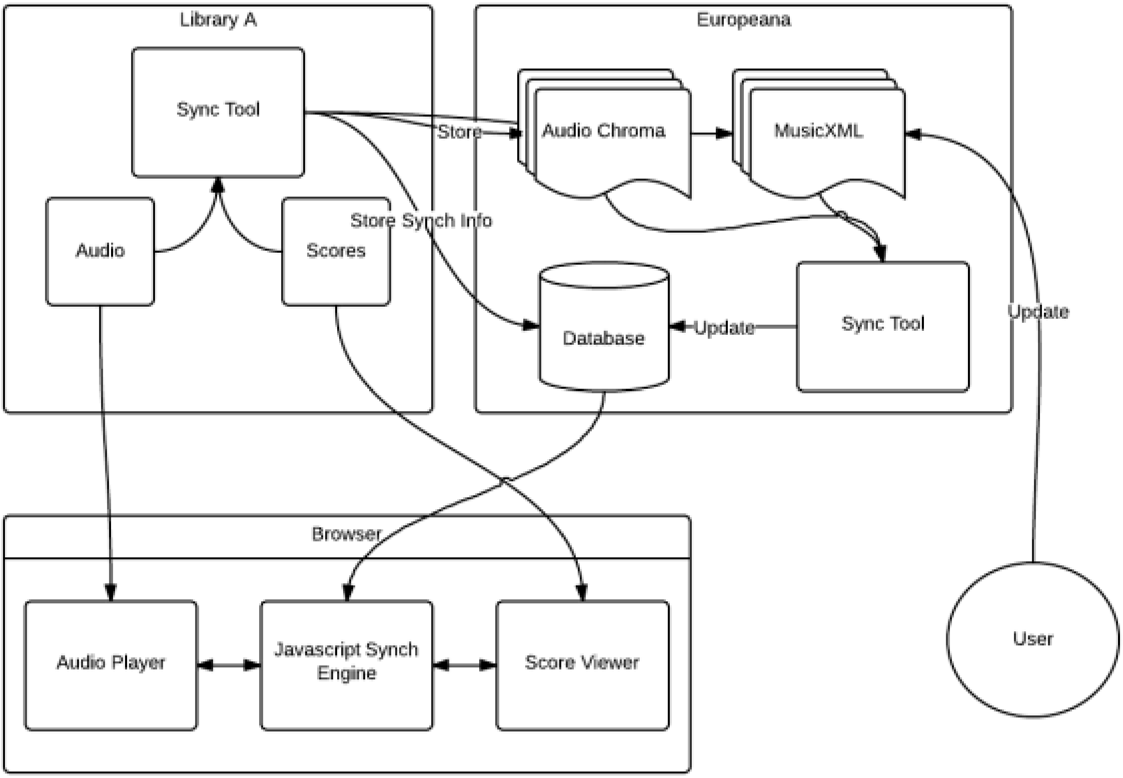
Disadvantages

- Requires that the audio-to-score alignment software is installed at the partner's system.
- May require local adaptations adhering to the prevalent IT infrastructure.
- Requires additional resources and trained personnel.
- Software updates have to be distributed to many sites

10.1.2 Scenario 2

Based on the unsatisfactory results of the OMR systems evaluation, a solely automatic recognition of sheet music is considered to be unreliable. Scenario 2 describes a semi-automatic approach where results of an automatic extraction process can be edited and corrected by online users. In this collaborative process, users experienced in symbolic music editing may download the MusicXML files to correct errors offline and upload the improved version back to the repository.

In this case a central storage facility is required that hosts the MusicXML files, the audio features and a synchronisation service that is triggered when a user uploads a new version. The MIDI features have to be calculated from the updated MusicXML files. Europeana would be an obvious target for hosting such services.



Advantages

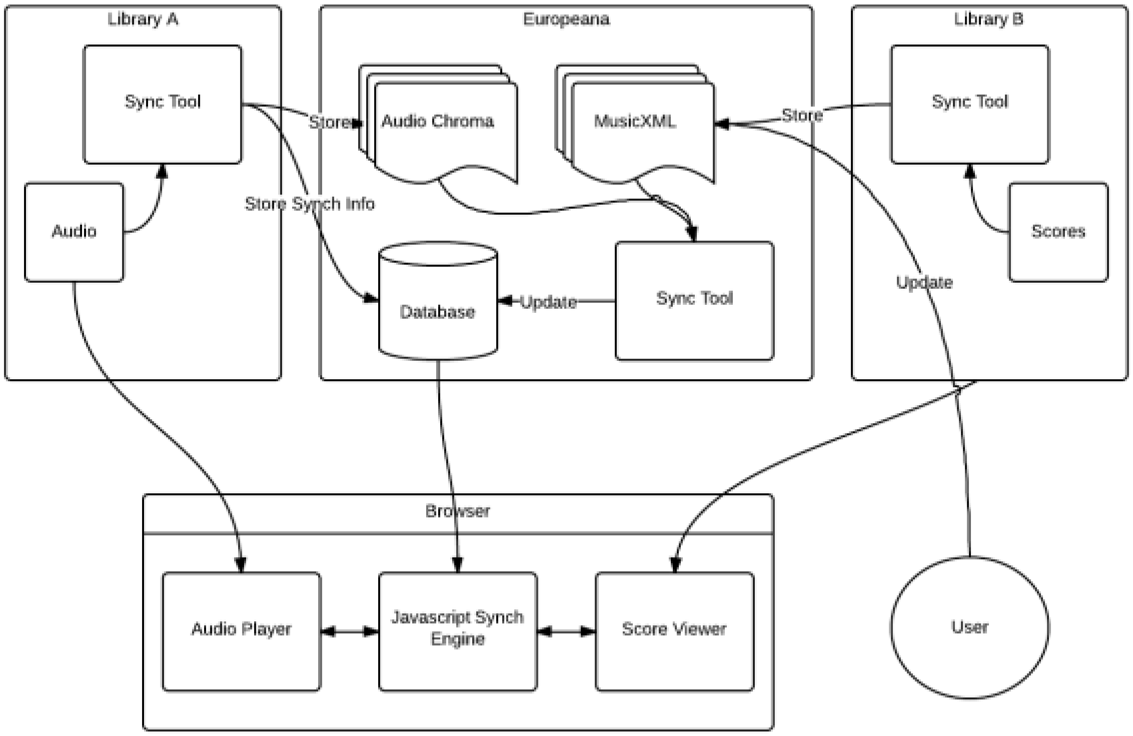
- Users could also make the matching between sheet music object A and music recording B. Something that it is difficult for a computer to do (unless or even if we use machine learning or similar technologies).
- Combines algorithmically created content with quality control by users

Disadvantages

- Sheet music editing software often confuses the page layout of scores extracted from images, often completely reordering measures and pages. Such rearranged versions cannot be back-synched to the source images.
- Software updates have to be distributed to many sites

10.1.3 Scenario 3

A partner library only has access to a single modality - either audio or images. In this case, the libraries have to extract the data required for the alignment process on their site and transmit it to a central repository. Since this is an asynchronous process, mechanisms have to be defined that trigger the synchronisation tool when two appropriate items (MusicXML and audio features) are available. The identification of corresponding items might also constitute a challenging task. In scenario 1 the curators are expected to correctly choose corresponding items. In this scenario it might not be clear to Library A that Library B even has corresponding content nor that it has already been uploaded to the repository.



Advantages

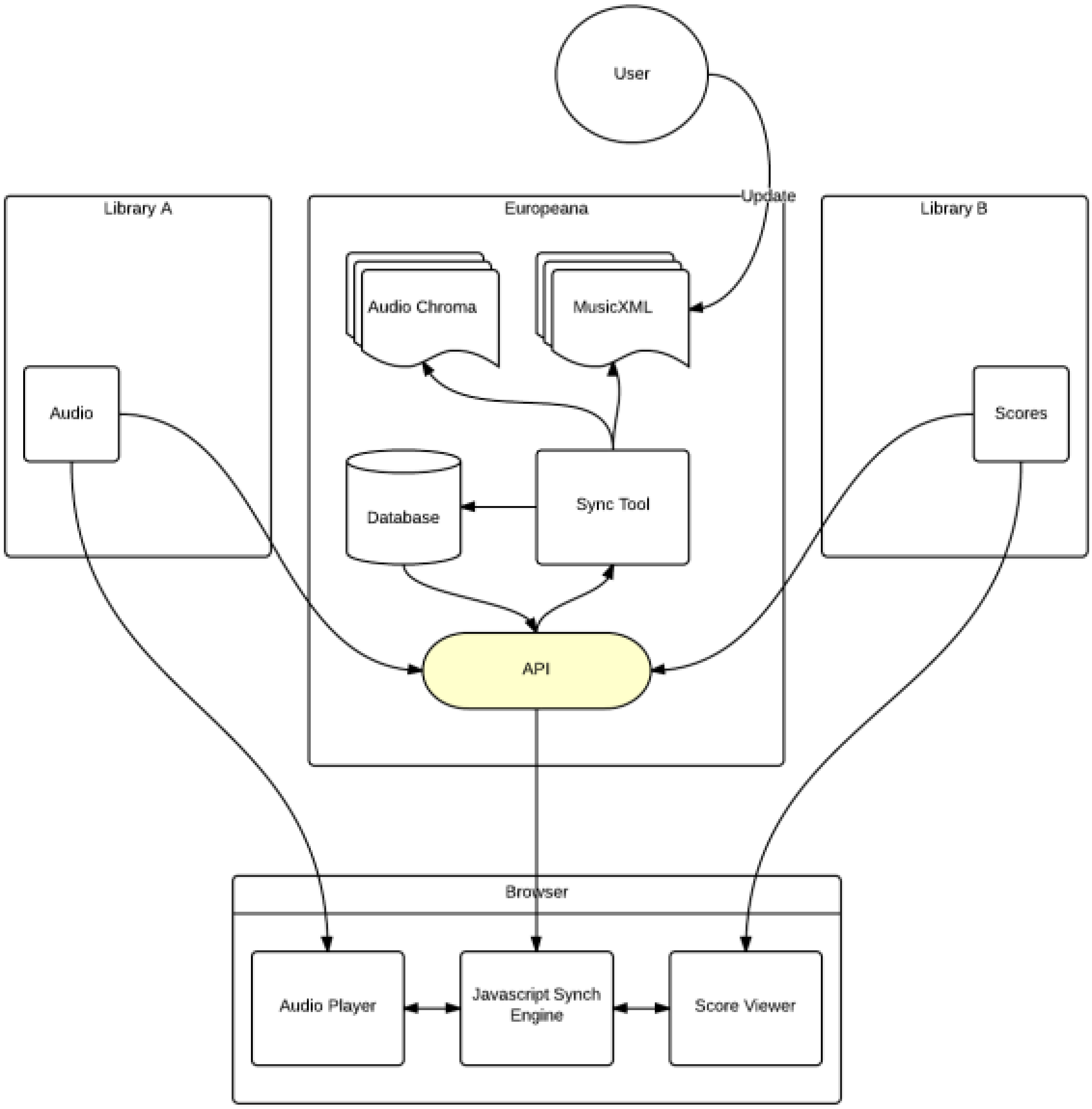
- Content resides at the partners' sites, which amends copyright concerns
- Users could also make the matching between sheet music object A and music recording B. Something that it is difficult for a computer to do (unless or even if we use machine learning or similar technologies).
- Combines algorithmically created content with quality control by users

Disadvantages

- Requires that the audio-to-score alignment software is installed at the partner's system.
- May require local adaptations adhering to the prevalent IT infrastructure.
- Requires additional resources and trained personnel.
- Requires procedures to match asynchronously uploaded content. Semi-automatic approaches should be preferred due to minor reliability of automatic approaches.
- Software updates have to be distributed to many sites

10.1.4 Scenario 4

The Europeana server hosts the synchronisation service which handles feature extraction and storage. External partners can upload their content via the public API.



Advantages

- Software resides at a single site
- Software updates do not need to be distributed.

Disadvantages

- Requires partners to upload content:
  - Raises copyright issues
  - OMR requires high-resolution images, which raises bandwidth concerns
- If substantial changes are made which require recalculation of the features, it is more difficult to re-engage partner institutions.

10.1.5 Scenarios beyond state-of-the-art

Current research ambitions focus Web scenarios. Recent approaches and contributions were presented at the 1st Web Audio Conference in Paris, 2015. Most interesting contributions concerning future scenarios of an audio-to-score alignment system within the Europeana Framework concerned Web based audio feature extraction algorithms, MIDI editors and visualisation engines. These contributions facilitate the relocation of the required data processing to the user, thus transforming it into a crowd-sourcing task. Recently, a Javascript implementation of an OMR system was announced [REF 9], though a date of availability is yet unknown.

Advantages

- Crowd-Sourcing approach
- Lessened copyright issues:
    - Content can be processed within the providing Web page, no download required.

Disadvantages

- Technologies are in an early stage of development
- Availability of Web based OMR might take several years
- May require the user to install a browser plugin

11 Summary

The intention behind Task 2.4.1 was to evaluate the applicability of state-of-the-art score-following technologies to content provided by Europeana partner organizations and its integration within the Europeana technical infrastructure.

The task was approached through a prototypical implementation of a score-following system. First, necessary components were identified. Second, available open-source components were investigated. Based on this a Python based prototypical implementation of an audio-to-score alignment system was created, which output was used to create HTML5 based interfaces to visualize the calculated alignment. This score-following Web interface was used to demonstrate and evaluate the objectives and obstacles of the chosen approach.

12 References

Ref 1	Krottmaier, H., Kurth, F., Steenweg, T., Appelrath, H.J., Fellner, D.: PROBADO - A Generic Repository Integration Framework . In: Proceedings of the 11th European Conference on Digital Libraries. (September 2007)
Ref 2	Kurth, Frank, et al. "A framework for managing multimodal digitized music collections." Research and advanced technology for digital libraries. Springer Berlin Heidelberg, 2008. 334-345.
Ref 3	Orio, Nicola, Serge Lemouton, and Diemo Schwarz. "Score following: State of the art and new developments." Proceedings of the 2003 conference on New interfaces for musical expression. National University of Singapore, 2003.
Ref 4	Müller, Meinard. *Information retrieval for music and motion*. Vol. 2. Heidelberg: Springer, 2007.
Ref 5	Urbano, Julián, et al. "What is the effect of audio quality on the robustness of MFCCs and chroma features." *International Society for Music Information Retrieval Conference*. 2014.
Ref 6	Mauch, Matthias, and Sebastian Ewert. "The Audio Degradation Toolbox and Its Application to Robustness Evaluation." *ISMIR*. 2013.
Ref 7	Rebelo, A., Fujinaga, I., Paszkiewicz, F., Marcal, A. R., Guedes, C., & Cardoso, J. S. (2012). Optical music recognition: state-of-the-art and open issues. *International Journal of Multimedia Information Retrieval*, *1*(3), 173-190.
Ref 8	Bainbridge, David, and Tim Bell. "The challenge of optical music recognition." *Computers and the Humanities* 35.2 (2001): 95-121.
Ref 9	Charalampos, Saitis, Andrew Hankinson, and Ichiro Fujinaga. "Correcting large-scale OMR data with crowdsourcing." *Proceedings of the International Workshop on Digital Libraries for Musicology*, 88–90. London, UK: 2014.

Appendix A: Terminology

A project glossary is provided at: http://pro.europeana.eu/web/guest/glossary.

Additional terms are defined below:

Term	**Definition**
AIT	Austrian Institute of Technology
APEX	Archives Portal Europe network of excellence
EC-GA	Grant Agreement (including Annex I, the Description of Work) signed with the European Commission
OCR	Optical Character Recognition
OMR	Optical Music Recognition
PMB	Project Management Board
TEL	The European Library