Visualization of Composer Relationships Using Implicit Data Graphs

Christoph Niese[1], Tatiana von Landesberger[1], and Arjan Kuijper[1,2](✉)

[1] Technische Universität Darmstadt, Darmstadt, Germany
[2] Fraunhofer IGD, Darmstadt, Germany
arjan.kuijper@igd.fraunhofer.de

Abstract. Relationships between classical music composers are known due to explicit historic material, for instance the friendship between Joseph Haydn and Wolfgang Amadeus Mozart, as well as the influence of the latter on Ludwig van Beethoven. While Haydn and Mozart were critics of each others work, Mozart and Beethoven probably never met in person. In spite of that there is an impact on especially the early music of Beethoven. While relationships between well-known composers like the mentioned ones are investigated, it can also be of historic interest to know the roles less-known composers played. Some of them might have a part in a famous persons work but were not further analyzed given the fact that there have been many composers and no hints given to researchers indicating which person would be worth studying. In this work we develop an approach to visually hint possible relationships among a large number of composers. Detailed historic knowledge is not taken into account; the hints are only based on the composer works as well as their lifetimes in order to guess directions of influence.

1 Introduction and Motivation

There are and have been many music composers living on this planet. All of them naturally got influenced by other representatives of their profession, be it while their time at the conservatory or later in their work life. This simple observation leads to questions that are of academic interest:

- Who influenced the famous composers like Wolfgang Amadeus Mozart and Ludwig van Beethoven?
- Had composers who are nowadays nearly forgotten a major impact?

Those questions have been answered by music theorists. However, those persons of course usually investigate where they think new insights can be gained. This implies:

- Famous composers and their relationships are well researched.
- Less-known composers gain less or no attention.

S. Yamamoto (Ed.): HIMI 2016, Part II, LNCS 9735, pp. 300–312, 2016.
DOI: 10.1007/978-3-319-40397-7_29

There are always ordinary persons who are just doing their job fine, who usually are in the majority. But then there seem to be a few individuals that are outstanding, who seem to press forward mankind in what they do, like Mozart did. This has been expressed in the Great Man Theory [2]. However, Herbert Spencer noted an important fact on this:

> If it be a fact that the great man may modify his nation in its structure and actions, it is also a fact that there must have been those antecedent modifications constituting national progress before he could be evolved. Before he can re-make his society, his society must make him [18].

To Mozart this society may have included his father Leopold and his friend Joseph Haydn most probably among others. Some relationships are that popular that they are general knowledge that is taught in schools, like the just mentioned ones. But what if Mozart once went to a concert of an infamous composer and heard a tune he liked that much he build a major work on it? Or what if he even liked a lesser-known composers style that much he finally permanently adopted it in his own? Maybe it would change the view on Mozart totally. Besides being made up by us, this would not be limited to Mozart. We just use him as he seems to us the commonly most well-known composer. Nevertheless he and others may be connected in manners unknown today. Imagine you do not know anything on any composer and get the task to find all relevant relationships Mozart had. You have two possibilities to start with:

Explicit: Consult written or oral sources that state who he used to meet.
Implicit: Analyze and compare Mozarts and all other composers work to find commodities. Those may indicate that the respective persons either met or knew the work of the other.

It would be cumbersome to read all texts ever written by, to, and on Mozart. Besides you would have to locate those sources. Any seemingly irrelevant person could have known and written down something of interest in a diary. But at least you would have definitive statements. Analyzing all works by all composers would be even more cumbersome. Besides, on one hand, found commodities may be random occurrences or may only serve as hints but not as proofs. On the other hand, however, you may find links that are nowhere to be found in verbal texts. And a computer may help you. Text can be analyzed automatically, but as stated, the relevant text has to be found first. The musical work of composers, however, is more easily available. Of course, much work is lost or incomplete, but we think it is overall easier to grasp. Musical scores are much more formalized constructs than texts are. Therefore processing by machines is more straightforward than automatically analyzing text. In this paper, an automatic approach is developed that can assist you in the task of finding relationships between composers. It gives an overview and hints indicating points of interest so you can solve the following tasks:

- Examine: You have a certain composer in mind and want to find relationships to other composers you did not know of.
- Browse: You have no composer in mind and just want to browse and being lead to directions you did not think of yourself.

Full details can be found in [14].

2 Related Work

The strength of computers is to process large amounts of data. This has been used by others to analyse musical work. Besides there also exists work that has other approaches but leads to the same direction we want to go to.

The project Peachnote by Viro [21] is backed up by a large database of music n-grams. Based on them, similarity between music is measured. While the projects wants to be something like Google Books but for music, based on the technology the author also created an application that visualises an overview of similarities.

This visualisation, however, is work-centric. You specify a work and as a result you get similar works by the same or by other composers. On the one hand you can find similar composer, though only based on a single piece. On the other hand however you need to have an idea to start with, you cannot just browse and look on hints the application gives you [5].

The Classical Music Navigator[1] lets you browse composers and works. For each composer there are entries for influences by and to other composers. It is based on historic knowledge, does not analyse scores automatically and is therefore limited and what humans already know. Furthermore there is no visualisation, so the big picture cannot easily be seen.

The Map of Mozart [13] clusters by similarity into subregions of a predefined area [20]. This again is work centric. Additionally, if you would apply this visualisation approach to composers, you could only use one measurement of similarity. We, however, want to apply more.

Much work has been done on music querying, like [11,22]. Then there is work that processes on physical music signals, but not on notes, like [3,17]. However, they are not composer-centric. For an overview of work related to visualizing music collections based on metadata, see the thesis of Holm [6].

3 Approach

We see finding relevant composer relationships as a top-down process of three steps:

1. General overview: From a large amount of composers and possible relationships you have to pick those you are interested in.

[1] http://people.wku.edu/charles.smith/music/index2.htm.

2. Specific overview: You look closer at the chosen composers and relationships in order to keep or dismiss them.
3. Specific research: The items you kept lead to an in-depth research with historic texts, original scores and actual musical understanding and experience.

While the last step should be conducted manually by a music theorist, the first two can be supported by an automatic approach which we try in the following sections.

3.1 General Overview

Throughout the history of music theory, multiple concepts of measuring music similarity have been developed. We want my approach to theoretically support an unlimited number of those, i.e. it should be extensible and not limited to one kind of measurement like it is the case for Peachnote. we call a measuring concept analyser. An analyser basically represents a function that takes a composer as input and returns a vector type as an output. Between such vector types a distance can be measured. Due to the fact that we have composers that can be connected to multiple other composers in multiple ways at once, we think the best method of visualisation is a node link diagram that consists of vertices and edges [8, 10]. A composer then is a represented as a vertex, a relationship by an edge [9].

Edge Representation. An edge for a certain analyser exists if the distance between the vector types resulting from the analysis of two composers does not exceed a certain threshold. This threshold is freely selectable by the user. As multiple edges between the same two composers can be confusing, we will merge those edges. Each edge as a colour that is derived from the analyser that was used to create it. Merging edges results in a single edge that is striped in the colours of the original edges. The standard colours of the analysers have been chosen in a way that they are easily to distinguish. From a hue circle, the maximum distances between colours have been chosen [19].

Edge Direction. Usually older people influence younger ones due to their experience. We want to reflect this by utilising directed edges which are represented by arrows from one vertex to another. The direction is determined by the following rules:

- If composer As death occurred earlier than composer Bs birth, the edge is drawn from A to B. In this case it is impossible that B influenced A.
- If composer Bs death occurred earlier than composer As birth, the edge is drawn from B to A. In this case it is impossible that A influenced B.
- If composer As birth occurred earlier than composer Bs birth, the edge is also drawn from A to B.
- Otherwise the edge is drawn from B to A.

Only the first two choices are guaranteed to be correct, as a living composer cannot influence a dead one. The other choices can, but do not need to be correct. However, since the graphs purpose is to provide a rough overview semi-correct edges can be tolerated. Furthermore, since we have a directed graph, every edge needs to have a direction. This is one of the two places where meta data is used, which always only happens for visualisation, not analysis purposes.

Vertex Highlighting. For a small graph this is sufficient and if you already know which composer you want to investigate, it may even be for a large graph. However, usually there are just to many edges. Therefore possible nodes of interest should be exposed to the user. We do this by offering highlighting dependent on the degrees of a node. The indegree is the number of edges that lead to a vertex from other vertices. The outdegree is the number of edges going from this vertex to other vertices. A high indegree means a composer has been influenced by many others, a high outdegree means a composer has influenced many others. The higher a degree is in respect to the total number of vertices in the graph, i.e. the more people a composer influenced or has been influenced by, the more highlighting will be applied.

Filtering. To further ease the navigation in an edge-heavy graph, as soon as a composer has been selected, all edges that are not in or outgoing ones of any selected composer are hidden. This is especially useful if you already know which composer you are interested in. You can directly select him and only see the relevant edges. This applies in addition to the filtering by threshold. Furthermore composer vertices can be hidden by choice.

Graph Layout. The last thing to note on the graph at this point is the layout. As the standard layout we choose a circle which proves to be simple and efficient. The vertices are arranged in a circle, leaving much empty space in a centre that can be filled by the edges. However, since lifetime metadata is present, it is used to provide additional layouting possibilities. Composer vertices can be arranged by birth century, birth decade, death century, and death decade. This can be especially useful if you are trying to follow temporal lines of influence. Depending on the granularity you wish, centuries or decades can be selected.

3.2 Specific Overview

If two composers are selected for further examination, the vector types the various analysers calculated for these composers need be compared to provide further insights. This will again done in a visual way. However, there are four questions up to this point:

1. What are the vector types analysers calculate?
2. How do analysers calculate those vector types?

3. How is distance between those vector types measured?
4. What is a good way to visually compare two vector types?

These questions are subject to the next section.

4 Analysis and Visualisation

In this chapter we present the analysers we created to analyse composers. For each analyser, the vector type it returns is defined as well as how one measures the distance between two. Finally there is a visualisation for each vector type that is used for comparison.

4.1 Music

As music is the subject of analysis we first want to introduce some basic concepts of music theory before defining the analysis approaches. There are seven natural tones which are in the English language identified by the letters C, D, E, F, G, A, B, which are represented by the white keys. Five semitones are represented by the black keys which can be created by manipulating the natural tones with an accidental. Those accidentals are ♯, b, ♯♯, bb and shift a natural tone by 1, −1, 2, −2 semitone steps respectively. Apparently the keys are repeating with the same letters. Therefore the letters need to be indexed, leading to the concept of an octave. Examples are C♯0, F1, B2. All combinations of natural tones and accidentals now have a unique identifier and can be called notes. Without the octave, the 12 different types of keys represent the pitch classes C, C♯, D, D♯, E, F, F♯, G, G♯, A, A♯, B. If you press a key, i.e. play a note, you chose how long you hold and how strong you press. This way you can influence the loudness of a tone, which is formalised in musical literature as the dynamic indication usually ranging from ppp (very silent) to fff (very loud). In this work we call this combination of note, duration and dynamic indication a note instance. Note instances are often produced at the same time. Those are grouped as so-called chords. For simplicity, chords can also contain only one note in this work. A sequence of chords played by a certain instrument is called a voice. One voice or multiple voices playing alongside form a movement. A collection of movements forms a piece, a musical composition. A person who has usually written multiple pieces is called a composer.

4.2 Favourite Analysers

The favourite note analyser simply counts all notes in all voices of all movements in all pieces by a composer. The favourite pitch class analyser and the favourite dynamic indication analyser do the same for pitch classes and dynamic indications. The item that is counted most often is then considered a composers favourite one. The distance between two notes a and b obviously is a simple subtraction, as well as the distance between two dynamic indications a and b:

abs(a − b). The distance between two pitch classes a and b can be measured by a modular operation [16]. This is because to the fact that those classes are arranged circular due to the lack of octave indexing: min((a − b + 12)mod12, (b − a + 12)mod12). Visually comparing two simple items requires nothing more than displaying them side by side.

4.3 Histogram Analyzers

Where the favourite analysers only determine the one favourite item by a composer, the histogram anaylsers use a very similar query to setup a histogram of all items that have been used by the composer. In a histogram objects are merged into classes and associated with the total numbers of occurrences. For example, pitch classes could be the classes and the total numbers are the numbers of times pitch classes can be found in a piece.

To measure the distance between two histograms we chose the cosine distance [4]. Its calculation requires two vectors of equal length and structure. Therefore we first determine all classes appearing in both histograms via a union operation. Then the two vectors are built upon these classes. The count values directly come from the original histograms. If a class does not exist there, the default value 0 is used.

In the visualisation, for each item two bars exist, one for each composer that is currently selected. They are normalized, meaning their lengths are relative to each histograms maximum. This allows comparison of distribution of values, not the actual counts. Nevertheless, those are also displayed at the sides.

The note histogram visualisation has the additional feature of displaying the notes that are not shared (i.e. one composer has a count of 0) are displayed at the top and bottom. This way the notes that actually have their counterparts can be compared.

4.4 Melody Analyzer

The melody of a movement is a sequence of single notes. In a piano work this would usually be notes played by the right hand whereas in orchestral works the melody is usually played by a group of string instruments, but not necessarily. To determine the melody we follow the approach of Lu et al. [12]: If the melody track is marked in the MIDI file, we choose it. If not, very short tracks and unlikely ones (percussion) are removed and the remaining ones are merged. The chosen or merged track is broken down to only the topmost notes of each chord.

A melody can be represented by a sequence of rectangles where a rectangles width is derived from a notes duration whereas its height is derived from the pitch. The distance between two such representations can be calculated by determining the difference of the total areas of both melodies [1]. Refinements in form of area minimisation by shifting the melodies horizontally and vertically as well as smartly scaling them are not applied by us. As we want to compare too many works the calculation should not bee too complex. We do, however, scale the rectangle sequences horizontally so they have an equal length and divide vertically by the minimum height. Each melody curve then has the same minimum and length allowing simple comparison of the courses. The analysis of a movement results in a collection of sequences of rectangles in which one is contained for each movement.

The rectangles can be seen as the extent of a set. Therefore the Jaccard index [15] can be used the measure the distance between two rectangle sequences. The needed intersection of two rectangle sequences is calculated like one would calculate the area between two functions, i.e. determining intersection points and solving multiple integrals. However, due to the noncurved structure, usual all-purpose integration is not necessary. The union of the rectangles can be determined by choosing the maximum area of two rectangles of equal width and summing all up.

We decided to let the distance between two rectangle sequence collections be the average of all pairwise contained rectangle sequences. As this whole approach is geometric the visualisation is basically simple. We show the areas of difference and draw coloured lines so one can track the melodies of both composers.

4.5 Data Sources

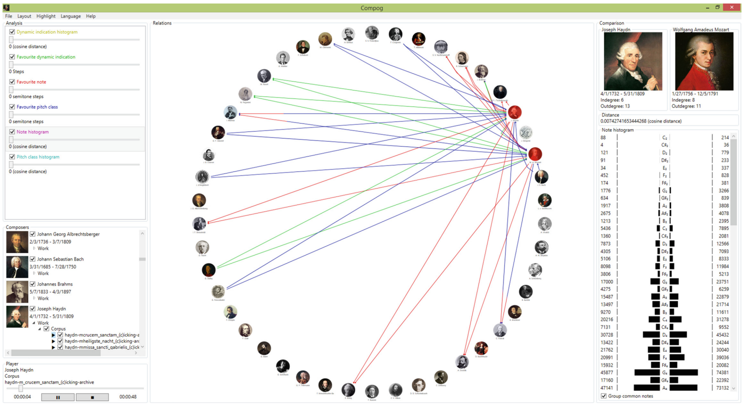
The resources we process come from three sources:

1. Kunst der Fuge (http://www.kunstderfuge.com) we chose MIDI as the file format from which to obtain musical data. It is a common format containing note information that can easily be parsed. The Kunst der Fuge archive contains contains more than 17,000 MIDI files.
2. Wikimedia Commons (http://commons.wikimedia.org) For a nice visualisation we need portraits of the composers. We extracted them from this collection.
3. Wikipedia (http://www.wikipedia.org) To draw directed edges we need the lifespans of each composer. We looked them up on this site. They do not have to be absolutely correct, so this source is sufficient.

4.6 Application Overview

Figure 1 represents a typical situation shortly after starting the program, loading composers and activating some controls.

On the top, the application features the typical menu bar that can be found in many programs. It contains several items and sub items. The import option is for importing data from the file system. A directory is expected that contains subdirectories which represent composers. Each subdirectory's name will be directly used as a composers name. Inside there can be an image file (PNG or JPG) featuring a composers portrait that is used for visualisation. The image files name should have the format YYYYMMDD-YYYYMMDD.FFF where YYYY is a year, MM a month, DD a day and FFF a file ending. Thus the file name indicates the life time of a composer. Subdirectories of composer directories represent pieces written by a him. The directory names equal the piece names.



☑ Dynamic indication histogram
0 (cosine distance)
☑ Favourite dynamic indication
0 Steps
☑ Favourite note
0 semitone steps
☑ Favourite pitch class
0 semitone steps
☑ Note histogram
0 (cosine distance)
☑ Pitch class histogram
0 (cosine distance)

Composers
☑ Johann Georg Albrechtsberger
2/3/1736 - 3/7/1809
Work
☑ Johann Sebastian Bach
3/31/1685 - 7/28/1750
Work
☑ Johannes Brahms
5/7/1833 - 4/3/1897
Work
☑ Joseph Haydn
4/1/1732 - 5/31/1809
Work
☑ Corpus
☑ haydn-mcrucem_sanctam_(c)icking-a
☑ haydn-mheiligste_nacht_(c)icking-arc
☑ haydn-mmissa_sancti_qabrielis_(c)ick

Player
Joseph Haydn
Corpus
haydn-m_crucem_sanctam_(c)icking-archive
00:00:04 00:00:48

Joseph Haydn
4/1/1732 - 5/31/1809
Indegree: 6
Outdegree: 13

Wolfgang Amadeus Mozart
1/27/1756 - 12/5/1791
Indegree: 8
Outdegree: 11

Distance
0.0074274165344268 (cosine distance)

Note histogram

88	C_2	214
4	$C\sharp_2$	36
121	D_2	779
91	$D\sharp_2$	233
34	E_2	337
452	F_2	828
174	$F\sharp_2$	381
1776	G_2	3266
634	$G\sharp_2$	839
1917	A_2	3808
2675	$A\sharp_2$	4078
1213	B_2	2395
5436	C_3	7895
1360	$C\sharp_3$	2081
7873	D_3	12566
4305	$D\sharp_3$	7093
5106	E_3	8333
8098	F_3	11984
3806	$F\sharp_3$	5213
17000	G_3	23751
4275	$G\sharp_3$	6259
15487	A_3	22879
13497	$A\sharp_3$	21714
9270	B_3	11611
20216	C_4	31278
7131	$C\sharp_4$	9552
30728	D_4	45432
13422	$D\sharp_4$	24244
21762	E_4	30040
20991	F_4	39036
15932	$F\sharp_4$	20082
45877	G_4	74381
17160	$G\sharp_4$	22392
47141	A_4	73132

☑ Group common notes

Fig. 1. Application overview

Inside a piece directory there are MIDI files representing movements or parts of the piece. Their names should equal the movement names. If you do not want to sort files by pieces, you could just create one folder and put all files into it. Open, save, and save as refer to the programs file format which is XML. XML files contain all data from the import directory, even the raw MIDI data and the portrait image, as a single file for convenient use. Furthermore analysis results are stored which are directly created at import. This is why importing takes much longer than opening an XML file. Further menu items include functionality for changing the graphs layout, the highlighting mode of vertices, and the language.

List of composers. The composers to be analysed can be reviewed and filtered here. Their respective works that will be analysed can be selected by whole and by movement, and the latter can be listened to with the integrated media player. If you click on a portrait, the composer is selected.

List of analysers. The analysers to be used to evaluate each composer and his respective pieces can be selected here. Each list entry also offers the possibility to set the threshold distance between two composers to filter edges. The analyser you have selected in this list is used for the detailed result comparison view.

Graph. The graph serves as the centre of the application. It is the visual overview of composers indicating similarities between them. Vertices and Edges are clickable in order to select two composers for further analysis. Zooming and translating is possible.

Comparison. The details of why two composers have been connected can be reviewed here. The commodities and differences can be seen as well as the calculated distance and the vertex degrees of the selected composers.

5 Evaluation

This penultimate chapter presents evaluation against known influences.

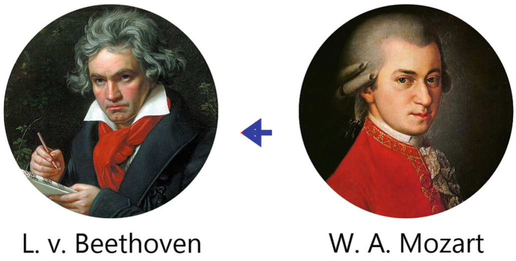
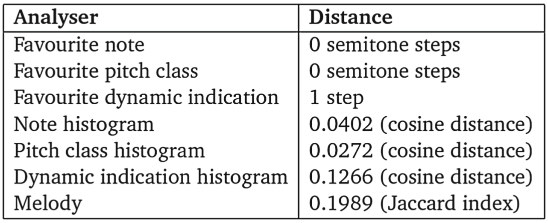


Fig. 2. Graph with only Mozart and Beethoven



Analyser	Distance
Favourite note	0 semitone steps
Favourite pitch class	0 semitone steps
Favourite dynamic indication	1 step
Note histogram	0.0402 (cosine distance)
Pitch class histogram	0.0272 (cosine distance)
Dynamic indication histogram	0.1266 (cosine distance)
Melody	0.1989 (Jaccard index)

Fig. 3. Results for piano concertos by Mozart and Beethoven

5.1 Piano Concertos by Mozart and Beethoven

Ludwig van Beethoven got inspired by Wolfgang Amadeus Mozarts Piano Concerto No. 24 when writing his own Piano Concerto No. 3 [7]. At the very beginning of the pieces you will notice similar themes. We created a special graph you can see in Fig. 2 for this that only contains the two composers and their respective works. Clicking on the edge between them and iterating through the analyser list, the values presented in Fig. 3 can be extracted.

According to the analysers the pieces as a whole are very similar: Although Beethovens work is louder the most occurring notes and pitch classes are the same. Furthermore, the two note histograms have a small cosine distance that is even underscored by the one of the pitch class histograms. One probably would examine this further if the connection was not already known.

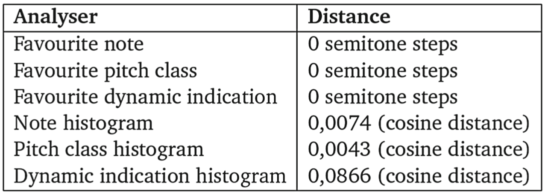
5.2 Whole Work by Haydn and Mozart

As mentioned in the beginning, Haydn and Mozart were friends that influenced each other. Analysis results of this relationship are shown in Fig. 4.

Due to the very small distances the analysers confirm that there is a connection between the two men.

5.3 Browsing

If we do not know what we are looking for we can nevertheless just start the program and load composers. The situation is as shown in Fig. 5 left. We decide



Analyser	Distance
Favourite note	0 semitone steps
Favourite pitch class	0 semitone steps
Favourite dynamic indication	0 semitone steps
Note histogram	0,0074 (cosine distance)
Pitch class histogram	0,0043 (cosine distance)
Dynamic indication histogram	0,0866 (cosine distance)

Fig. 4. Results for whole work by Haydn and Mozart

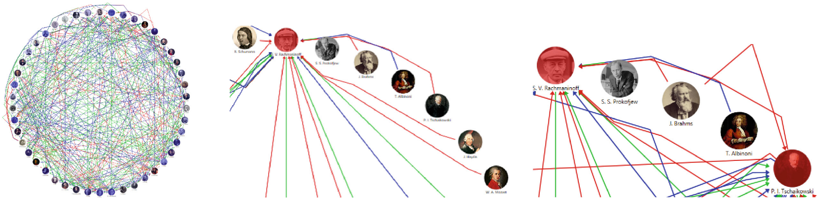
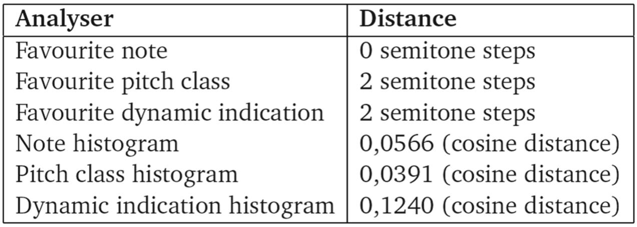


Fig. 5. Browsing, steps 1 to 3.



Analyser	Distance
Favourite note	0 semitone steps
Favourite pitch class	2 semitone steps
Favourite dynamic indication	2 semitone steps
Note histogram	0,0566 (cosine distance)
Pitch class histogram	0,0391 (cosine distance)
Dynamic indication histogram	0,1240 (cosine distance)

Fig. 6. Results for the comparison of Rachmaninoff and Tschaikowski

to click on Rachmaninoff because he is highlighted in a deeper blue than other composers. Now the situation has changed to the middle image. We see an edge coming from Tschaikowski and because we like Tschaikowski we decide to click on him. Now both composers are selected as shown in the right image. We analyse the distances and get the values shown in Fig. 6.

The results indicate a close relationship. If we look Rachmaninoff up in the Classical Music Navigator, we can read he has been heavily influenced by Tschaikowski which confirms the results.

5.4 Limitations

Much work cannot easily be found in the MIDI format. So even some works of famous composers are unavailable, not speaking of those of unknown ones. This seems to be the price of this trivially machine-readable format. Besides the issue that we cannot verify the connection between more work we know it is related, there is a much larger one: Influences of unknown composers cannot be found if the program has no data on them. You may have noted that results from the melody analyser is only available in the first evaluation. This is due to performance issues. Too many comparisons are needed to run on a decent end user device, so we excluded it most of the time. Also, we cannot load all 884 composers that are available to us, as the performance is too much of an issue. Therefore we used a smaller set of less than 50 composers.

5.5 Conclusions

In this work we developed an approach to assist a music theorist in finding unknown relationships between composers. We created a concept of music analysis, graph visualisation and details visualisation so an expert can follow three steps on the way to new insights. The approach resulted in a software application.

To solve the data amount problem, the largest online source of music should be concerned: Scanned sheet music, which is for example collected in the Petrucci Music Library. This then evokes the need of image analysis and translating into a machine-readable representation. Sheets never scanned or even never found unfortunately are inaccessible of course leaving the dream of a complete graph being a dream. It is unlikely that a fat client can handle all this data. Therefore a thin client with raw server power as a back end should be considered and the program accordingly rewritten. This also leads to new analysis possibilities. Where the melody analyser is critical because of the pairwise distance calculation, N-Grams that are used in Peachnote are simply too processing intensive. However they seem to work very well and the program could benefit from them. In general, there are always more analysers that could be added. A person who has academic music theory background could be consulted to determine what exactly is needed and to know whether the layout options are sufficient. Concerning visual output one could think of a better way of edge drawing. Both the routing and the colour mixing is not the optimum.

References

1. Aloupis, G., Fevens, T., Langerman, S., Matsui, T., Mesa, A., Nuñez, Y., Rappaport, D., Toussaint, G.: Algorithms for computing geometric measures of melodic similarity. Comput. Music J. **30**(3), 67–76 (2006)
2. Carneiro, R.L.: Herbert spencer as an anthropologist. J. Libertarian Stud. **2**, 1712 (1981)
3. Charbuillet, C., Peeters, G., Barton, S., Gouet-Brunet, V.: A fast algorithm for music search by similarity in large databases based on modified symetrized Kullback Leibler divergence. In: Content-Based Multimedia Indexing (CBMI 2010), pp. 1–6 (2010)
4. Connor, R., Moss, R.: A multivariate correlation distance for vector spaces. In: Navarro, G., Pestov, V. (eds.) SISAP 2012. LNCS, vol. 7404, pp. 209–225. Springer, Heidelberg (2012)
5. Engelke, T., Becker, M., Wuest, H., Keil, J., Kuijper, A.: MobileAR browser - a generic architecture for rapid AR-multi-level development. Expert Syst. Appl. **40**(7), 2704–2714 (2013)

6. Holm, J.: Visualizing music collections based on metadata: Concepts, user studies and design implications. Technical report, Tampere University of Technology (2012). https://dspace.cc.tut.fi/dpub/handle/123456789/21161
7. Kinderman, W.: Reviews of books: Beethoven forum, ii. Music Lett. **77**(1), 124–126 (1996)
8. von Landesberger, T., Bremm, S., Kirschner, M., Wesarg, S., Kuijper, A.: Visual analytics for model-based medical image segmentation: opportunities and challenges. Expert Syst. Appl. **40**(12), 4934–4943 (2013)
9. von Landesberger, T., Görner, M., Rehner, R., Schreck, T.: A system for interactive visual analysis of large graphs using motifs in graph editing and aggregation. In: Vision, Modeling, and Visualization 2009, pp. 331–340 (2009)
10. von Landesberger, T., Kuijper, A., Schreck, T., Kohlhammer, J., van Wijk, J.J., Fekete, J., Fellner, D.W.: Visual analysis of large graphs: state-of-the-art and future research challenges. Comput. Graph. Forum **30**(6), 1719–1749 (2011)
11. Liu, C.C., Hsu, J.L., Chen, A.: An approximate string matching algorithm for content-based music data retrieval. In: IEEE International Conference on Multimedia Computing and Systems, 1999, vol. 1, pp. 451–456 (1999)
12. Lu, L., You, H., Zhang, H.J.: A new approach to query by humming in music retrieval. IEEE International Conference on Multimedia and Expo, ICME 2001, pp. 595–598 (2001)
13. Mayer, R., Lidy, T., Rauber, A.: The map of mozart. In: 7th International Conference on Music Information Retrieval, ISMIR 2006, pp. 351–352 (2006)
14. Niese, C.: Visualization of composer relationships using implicit data graphs. Technical report, Darmstadt University of Technology (2015)
15. Niwattanakul, S., Singthongchai, J., Naenudorn, E., Wanapu, S.: Using of Jaccard coefficient for keywords similarity. Lect. Notes Eng. Comput. Sci. **2202**(1), 380–384 (2013)
16. Rahn, J.: Basic Atonal Theory. Schirmer, New York (1980)
17. Schnitzer, D.: Mirage - high-performance music similarity computation and automatic playlist generation. Vienna University of Technology, Austria, Technical report (2007)
18. Spencer, H.: The Study of Sociology. Appleton, New York (1896)
19. Stone, M.: Choosing colors for data visualization. Technical report (2006). http://www.perceptualedge.com/library.php
20. Tatu, A., Zhang, L., Bertini, E., Schreck, T., Keim, D., Bremm, S., von Landesberger, T.: Clustnails: visual analysis of subspace clusters. Tsinghua Sci. Technol. **17**(4), 419–428 (2012)
21. Viro, V.: Peachnote: music score search and analysis platform. In: 12th International Society for Music Information Retrieval Conference, ISMIR 2011, pp. 359–362 (2011)
22. Welsh, M., Borisov, N., Hill, J., von Behren, R., Woo, A.: Querying large collections of music for similarity. Technical report UCB/CSD-00-1096, EECS Department, University of California, Berkeley (2000)