COMPUTATIONAL METHODS FOR THE ANALYSIS OF MUSICAL STRUCTURE

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF MUSIC
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Craig Stuart Sapp
May 2011

This dissertation is online at: http://purl.stanford.edu/br237mp4161

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Julius Smith, III, Primary Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Christopher Chafe

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Eleanor Selfridge-Field

Approved for the Stanford University Committee on Graduate Studies.

Patricia J. Gumport, Vice Provost Graduate Education

Contents

List of Tables

List of Figures

Chapter 1

Overview

Music is an art form which is realized in time. This dissertation presents computational methods for examining the temporality of music at multiple time-scales so that both short-term surface features and deeper long-term structures can be studied and related to each other. The methods are applied in particular to musical key analysis (Chapters 2–4) and also adapted for use in performance analysis (Chapters 5–6). The essential methodology is to examine all sequential time-scales within a piece using some analytic process and then arrange a summary of the analytic results into a maximally overlapped arrangement. Chapter 2 defines a two-dimensional plotting domain for displaying musical features at all possible time-scales which forms a basis for further analysis methods. The resulting structures in the plots can be examined subjectively as a navigational aid in the music as illustrated in Chapters 3 and 5. They can also be used to extract musically relevant information as discussed in Chapters 4 and 6.

1.1 Computational analysis of musical keys

The multi-scale analysis methods were developed specifically to examine the behavior of computational algorithms for identifying musical keys and especially to show how they behave within different parts of a composition. Most tonal compositions are described as being in a particular key—often included in the work's title. Other secondary key regions are left unmentioned, however, unless a detailed analysis of the music is being written.

Chapters 2 and 3 focus on applications using the Krumhansl-Schmuckler key finding algorithm (KS algorithm) in structural key analysis. This algorithm can be found defined mathematically at the start of Chapter 3, and it is suitable for examining the musical key in a wide range of musical styles and time-scales. It is a low-complexity algorithm which correlates pitch-class histograms collected from a musical sample against expected scale-degree prototype histograms in either a major or minor key. These key prototypes for the KS algorithm were initially taken from perceptual experiments; however, several alternative weightings subsequently have been proposed as improvements for use in the algorithm. These key prototype scale-degree histogram (scale-degree weight) improvements are examined in detail in the next few sections.

The KS algorithm is particularly suitable for key-finding applications because it can process any type of musical texture. Since it collapses all musical content into a single pitch-class histogram in order to determine the key, it will work equally well for music which is monophonic—containing only a single melodic line—or for the most complex musical textures in which chords are elaborated in time. Key-finding algorithms based solely on the definition of key via chord sequences will have difficulty parsing music other than didactic examples or clearly presented or pre-identified chords.

1.1.1 Scale-degree weights for the KS algorithm

Chapter 4, Table 4.1 lists the original Krumhansl-Kessler (KK) scale-degree weightings for major and minor keys used in the KS algorithm as well as three additional prototype weight-sets subsequently proposed. All three of the more recent prototypes are extracted from musical scores, which is in contrast to the KK weights which are adapted from psychological experiments rating the suitability of test tones following a musical sequence. The three additional weight-sets which are examined in detail in the next section are briefly described in the following list.

- AE = Aarden-Essen weights. Scale-degree histograms derived from intermediate note counts generated from monophonic songs in the Essen Folksong Collection.[1] (Also “Aa” abbreviation used in some figures).

- BB = Bellman-Budge weights. Scale-degree histograms derived primarily from chord frequency distributions in 100 works of classical music from the 17th and 18th centuries.

- KP = Kostka-Payne weights. Scale-degree histograms derived from short harmony examples in a music-theory textbook (and workbook).

By extension, other scale-degree weight-sets can be generated by extracting pitch histograms from other musical repertories. A fifth set of weights developed as a control case is also given in Chapter 4, Table 4.1, where they are labeled as being “low complexity”

[1] http://kern.ccarh.org/browse?l=/essen

or "simple" since they are generated from only two concepts: (1) assign all scale-degrees in the key to 1 and all pitch-classes outside the key to 0. Since this by itself will cause modal confusion between relative major and (natural) minor keys which have the same key signature (such as C major and A minor), an additional value of 1 is added to the tonic and dominant scale-degrees (or the final and reciting tone in generalized modal keys).

1.1.2 Krumhansl-Schmuckler algorithm evaluation

The KS algorithm performs well when the underlying music is in a single key; however, it cannot be used alone to detect the presence of two (or more) keys within a segment of music. Since all temporal information is removed before identifying the key, the difference between highly chromatic music and multiple key regions can be negligible. Chapter 4 discusses the identification problems which occur when the notes in one key are combined with another's before applying the KS algorithm, and how to identify key modulation boundaries for separating these key regions.

In order to find key modulations, it is important to start with key prototype weights which give as accurate a local key measurement as possible. As a basic comparison of the KS algorithm using the different weights for finding the musical key, consider the following test of applying the algorithm to 120 compositions—all of the J.S. Bach Well-Tempered Clavier (WTC), Books I & II (both preludes and fugues), as well as Chopin's op. 28 preludes. These five collections contain cycles of compositions in all 24 major and minor keys, so highlighting each key is a primary compositional purpose for these works.

Results of applying the KS algorithm to each of these works as a whole are summarized in Table 1.1 for each weight-set. Interestingly, the simple weight-set has the lowest total error rate, misidentifying 4 out of the 120 compositions, followed closely in performance by the three weights derived from musical data: AE weights missing 6/120, KP weights missing 8/120 and BB missing 9/120 key assignments, and all performing better than the original KK weights, which misidentify 12/120 tonic keys in the test set of compositions.

Errors generated by the algorithm usually result in choosing a key closely related to the actual tonic key. Ten of the identification errors in Table 1.1 are caused by matching to the dominant key instead of the actual tonic key (marked "dom" in the table, plus the

	J.S. Bach, The Well-tempered Clavier, Book I, preludes					Book I, fugues					Book II, preludes					Book II, fugues					Chopin preludes op. 28					
key	AE	BB	KK	KP	S	AE	BB	KK	KP	S	AE	BB	KK	KP	S	AE	BB	KK	KP	S	AE	BB	KK	KP	S	*key*
C maj																										C maj
c min												rel		rel												c min
C♯maj								dom														para	dom	para	dom	C♯maj
c♯min																										c♯min
D maj																										D maj
d min																										d min
E♭maj																										E♭maj
e♭min																										e♭min
E maj																					para	para	para	para		E maj
e min							rel		rel																	e min
F maj	rel	rel	rel	rel	rel					rel	rel												dom			F maj
f min												rel														f min
F♯maj																										F♯maj
f♯min																										f♯min
G maj													dom					dom								G maj
g min																										g min
A♭maj																										A♭maj
g♯min																										g♯min
A maj																rel		dom								A maj
a min							rel		rel													D	D			a min
B♭maj			dom																		e♭	e♭		para	para	B♭maj
b♭min																										b♭min
B maj													dom			rel										B maj
b min																		f♯	rel							b min
errors	1	1	2	1	1	0	2	1	2	1	1	2	2	1	0	2	0	3	1	0	2	4	4	3	2	

cumulative errors: AE: 6/120 (5.0%) BB: 9/120 (7.5%) KK: 12/120 (10.0%) KP: 8/120 (6.7%) S: 4/120 (3.3%)

Table 1.1: KS algorithm results by weight-set for 120 compositions (using notes from entire composition in the analysis). Only errors are marked, with "dom" = dominant identified as the key; "rel" = relative key; "para" = parallel key.

F♯ minor identification in the B minor fugue of WTC, Book II). This is a reasonable error since there is only one pitch-class difference between a tonic key and its dominant key, such as F♮ in C major changing to F♯ in G major. Seventeen of the errors are misidentifications of the parallel key, such as identifying C major when C minor is the tonic. This is an understandable confusion when major and minor tonic key regions are both present in the piece. The Bach WTC compositions contain only these two types of errors, while the more harmonically complex Chopin preludes contain an additional four identification errors where the subdominant key was misidentified as the tonic key of the composition. Again, this is a closely related key to the tonic with only one pitch-class difference between the keys. Therefore, only four types of errors are listed in Table 1.1: misidentification of the tonic as the dominant key, subdominant key, parallel key, or relative key. This observation is utilized in Chapter 4 for refinement of local-key analyses to improve the identification accuracy of modulation boundaries between keys using the KS algorithm.

Of course, this type of test using the entire composition is not completely fair, since internal secondary key material will confuse the algorithm when applying it to an entire composition. Treating this as a relative test for comparing the different weights' performances, however, does give a basic indication of their accuracy. The KS algorithm could not correctly identify the key of only one work—the F-major prelude from WTC, Book I—with any of the tested weight-sets; otherwise, no more than two errors occur for any piece in the WTC compositions using the five different weight-sets. The Chopin preludes are more harmonically complex, so all weights except one fail to identify the actual tonic key in three preludes (but a different weight-set is responsible for the correct answer in each case).

1.1.3 Scale-degree prototype evaluations by keyscape

The keyscape plots defined in Chapter 2 can be used to conduct a more detailed analysis of the performance of the proposed prototype weights for use in the KS algorithm. Figure 1.1 gives schematic descriptions of three types of evaluation tests which can be done to compare the effectiveness of different key prototype weights.

The first test illustrated schematically in Figure 1.1 is the same as the test conducted in Table 1.1. Here it is called the "Top-level Test" because all of the notes in a composition are used as input into the KS algorithm, generating a single pitch-class histogram used to determine the key. The test schematics in Figure 1.1 are derived from one of the plotting domains described in Chapter 2 which generates a triangular plot. The apex of the triangle represents an analysis of the work when all of its notes are processed by the KS algorithm at once. This point represents a two-dimensional parameter with the *x*-axis (horizontal) coordinate value being the *center* of the region of music being processed, and the *y*-axis (vertical) coordinate value being the *duration* of the musical selection being input (or "*window-size*"). As alluded to previously, this test will typically include data noise from secondary keys and is the main reason why an incorrect key identification can be generated by the KS algorithm.

A complementary test to the top-level one is a low-data test illustrated in the middle of Figure 1.1. This test is labeled the "Leading-Edge Test", since it involves accumulating more and more notes starting at the beginning of the composition until the correct answer is given by the algorithm, thus measuring how many notes are needed to identify the key. This test is covered in Chew [2] and Krumhansl [7], so it is not covered here. This sort of test evaluates how the KS algorithm behaves under low-data conditions, which is not an ideal test for algorithms that require distribution data to perform an analysis (such as the KS algorithm). This test probably has strengths and weaknesses comparable to the top-level test.

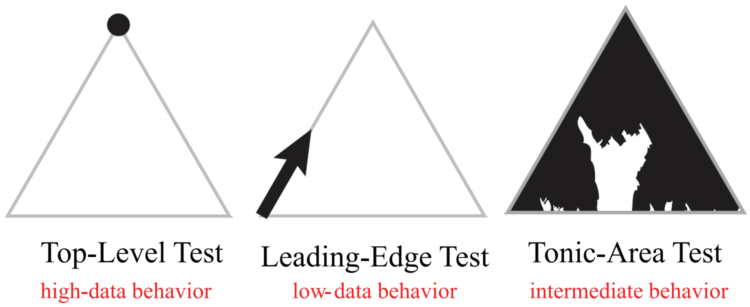


Figure 1.1: Three possible evaluation tests for weights used in the KS algorithm shown in terms of the keyscape plotting domain defined in Chapter 2.

A third, intermediate form of test is labeled the "Tonic-Area Test" in Figure 1.1. In this test, all possible sequential segmentations of the music from short musical snippets of the entire composition are considered, and the fraction of segments which gives the assigned key for the composition is measured. For example, the dark area in the figure may represent identification of the tonic key of a composition while the white region may indicate one or more secondary keys present in the middle of the composition. The tonic-area test reports the total area of the triangular plotting region which is covered by the tonic key identification, and the weight-set which finds the greatest amount of tonic key is viewed as the best set.

Figures 1.2 through 1.6 illustrate the raw analysis plots for the tonic-area test using the same 120 Bach and Chopin compositions as in the top-level test. Six plots are shown for each composition, with "SS" being the simple weight key prototypes abbreviation. The sixth plot for each composition, labeled "M", is a majority-answer plot. In these plots, keys are indicated if at least three of the five test weight-sets agree on the same answer at the same position in the music; otherwise, black is used in the majority plot to indicate disagreement between the sets (possibly indicating an ambiguous key region in the music). The green-colored regions in the plots indicate that the identified key was the desired tonic key label of the composition. Table 1.2 lists the coverage area of this green color (tonic key) in each of the 720 evaluation plots. Light blue regions in the plots indicate the dominant key, yellow for the subdominant, red for the mediant, purple for the submediant, orange for the supertonic, and dark blue for the supertonic. Darker hues indicate minor keys.

In order to numerically compare the performance of the weights for each composition independently (so that overly simple or complex compositions do not bias the results), the values from Figure 1.2 are normalized for each composition by the weight-set which gives the highest tonic-area coverage for each particular composition. A summarization of the final results of the key area test is shown in Figure 1.3. Major and minor key performances for each weight-set are reported separately as well as combined.

For major keys, the KP (Kostka-Payne) weights perform very well. In more than half of the 60 major-key compositions, they detect a higher amount of tonic key throughout the piece than any of the other weights, and they have the lowest variability in their answers as indicated by having the lowest standard deviation between normalized key area percentages. The BB (Bellman-Budge) weights also perform better than the simple weights used as a control set. But neither the AE (Aarden-Essen) nor the KK (Krumhansl-Kessler) weight-set performs as well in the tonic-area test as the simple weights, with the KK weights performing significantly lower—only averaging 58% area coverage compared to the best weight-set. The majority result performs insignificantly poorer than the simple weights, with about the same reliability as the simple weights, since the standard deviation of the normalized tonic-areas over all major-key compositions is about as low as for the simple weights.

Visual inspection of the keyscapes in Figures 1.2 through 1.6 demonstrates that the poor performance quality of the KK weights is due to a strong preference for the dominant key. For example, the blue regions in the tonic-area test plots for the KK weights are almost always more prominent than in plots generated by the other weight-sets. A clear example

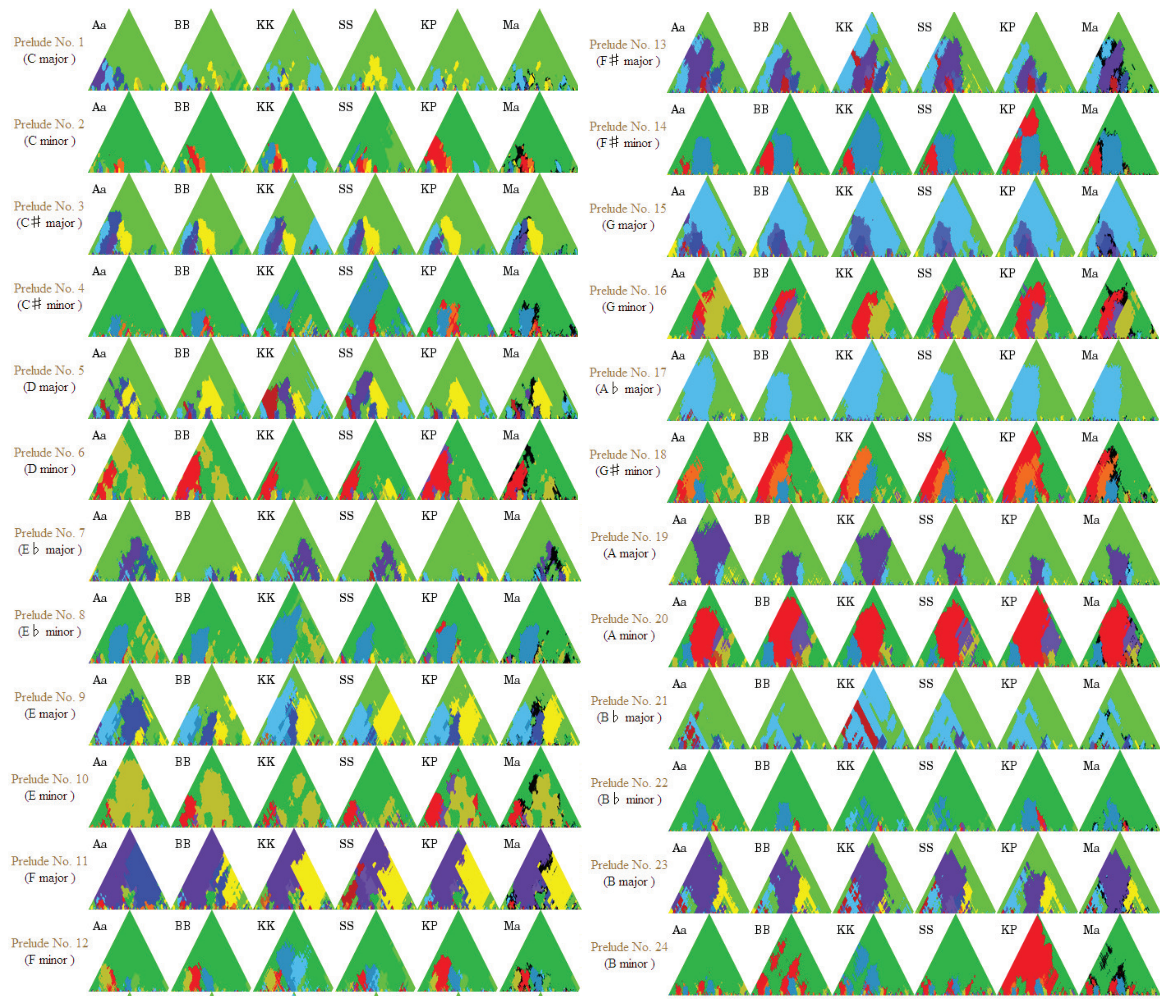


Figure 1.2: Functional keyscapes for J.S. Bach: WTC, Book I preludes.

of this dominant-biased property of the KK weights can be seen in fugue no. 7 in E♭ major in Figure 1.5. For this composition, all other weight-sets have a small dominant region near the middle of the piece, while the KK weights over-saturate the plot towards the dominant.

Much of the preference towards the dominant key is controlled by the ratio of the tonic to dominant scale-degree weights in the prototypes. If this ratio is greater than one, such as with the KK prototypes, the weights will tend to be more sensitive to the dominant key. If the ratio is less than one, such as with the AE prototypes, then the weights will tend to be sensitive to the subdominant—or in other words, sensitive in the opposite direction in

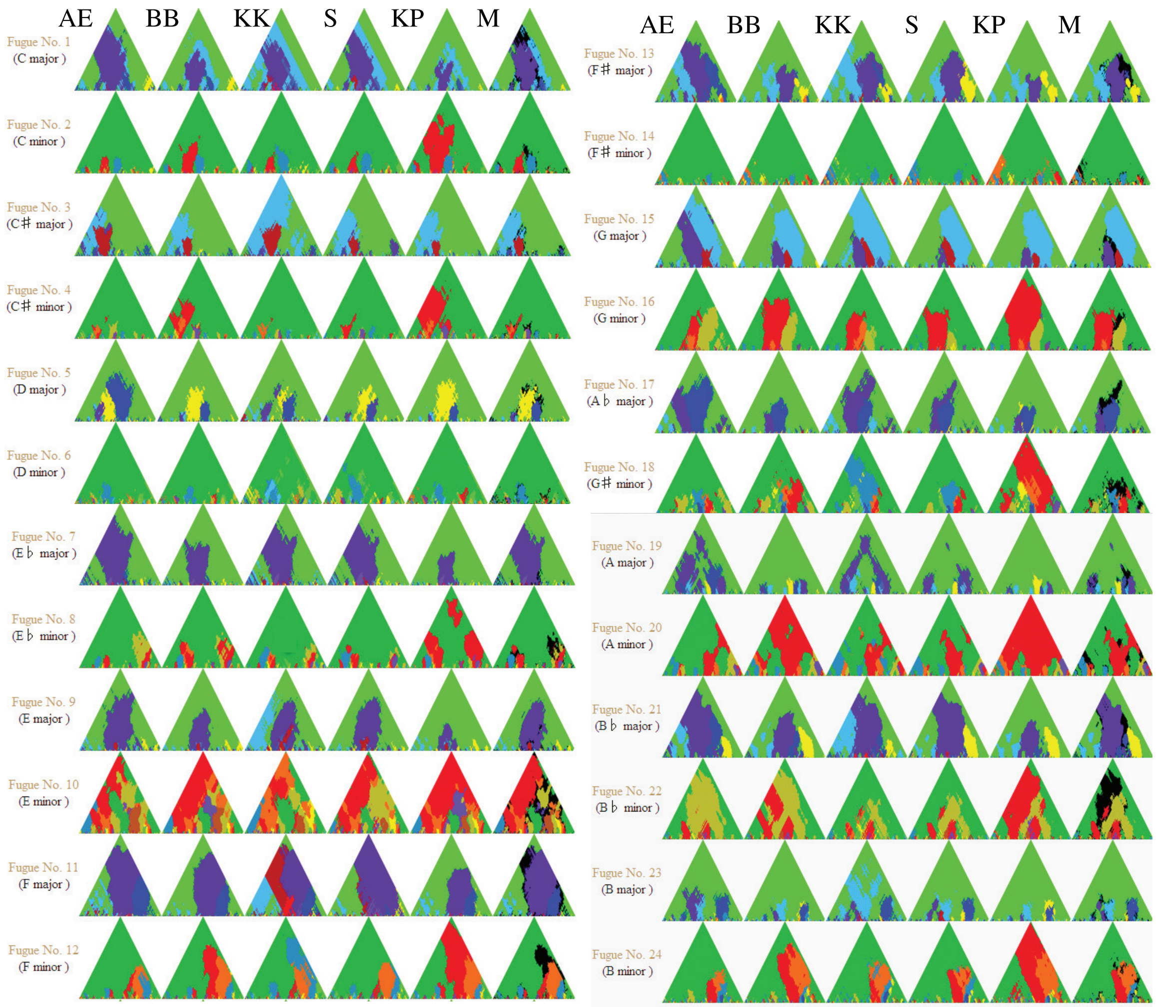


Figure 1.3: Functional keyscapes for J.S. Bach: WTC, Book I fugues.

the Circle of Fifths. This is one reason that the simple weights perform best on the top-level test, since it will not be particularly biased towards either dominant or subdominant secondary key regions, and the tonic has a better chance of appearing at the top of the plot. Visual inspection of the plots in the previous figures shows that the simple weights are slightly biased towards the dominant key, probably caused by non-optimal secondary ratios between scale-degrees other than the tonic and dominant.

For minor keys, only the AE weights perform better than the simple weights. In particular, the KP weights perform very poorly when compared to their near-perfect performance

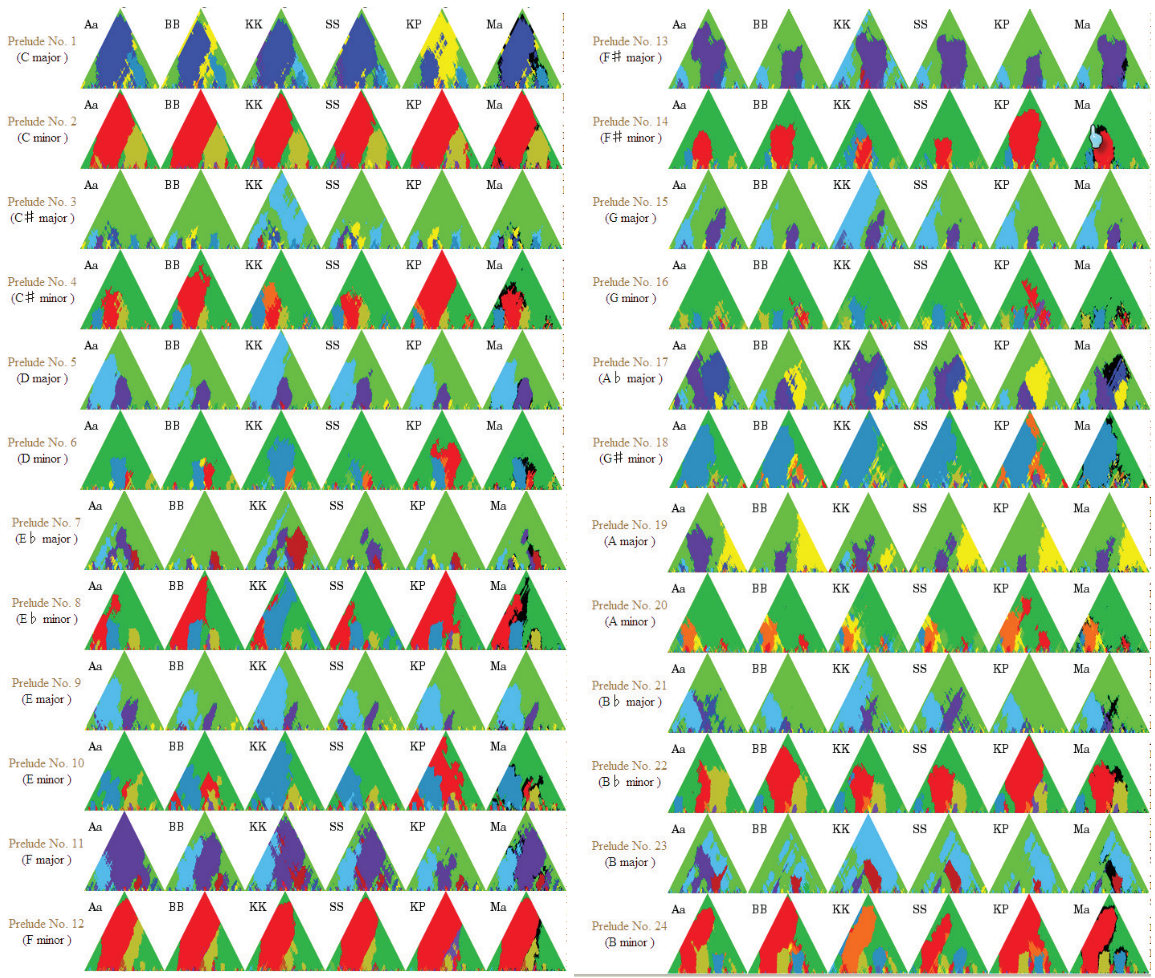


Figure 1.4: Functional keyscapes for J.S. Bach: WTC, Book II preludes.

on the major keys. The KK weights perform much better with the minor-key compositions, at about the same level as the BB weights. Again, the majority results perform slightly poorer than simple weights in minor keys.

Visual inspection of the keyscapes reveals why the KP weights perform poorly in minor keys—they are biased towards the relative major key. In other words, the red regions in the minor plots for the KP weights are almost always larger than those of other weight-sets. As an example, consider Chopin's prelude no. 8 in F♯ minor in Figure 1.6, where the red region representing the relative major is much larger in the KP plot than in the plots generated by other weights.

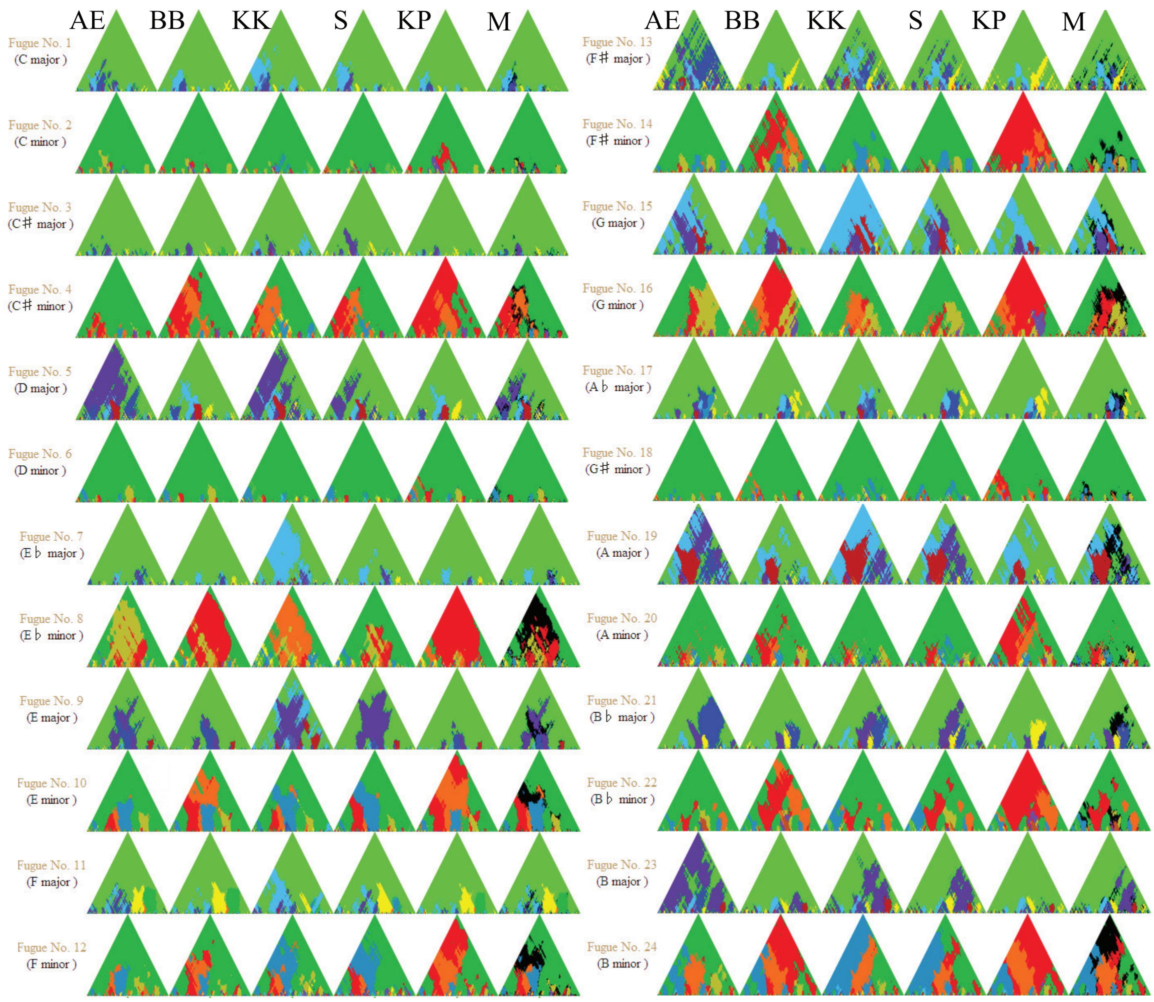


Figure 1.5: Functional keyscapes for J.S. Bach: WTC, Book II fugues.

For the results considering all 120 compositions, including both major and minor keys, the BB weights perform slightly better than the simple weights, with a mean normalized score of 0.86 compared to 0.85 for the simple weights. However, considering the standard deviation of 0.14 in the variability of normalized scores across all compositions, this is not a very significant difference. The AE weights perform about the same as the simple weights. The KP weights have a poor overall performance mainly due to their bias towards the relative major when in minor keys.

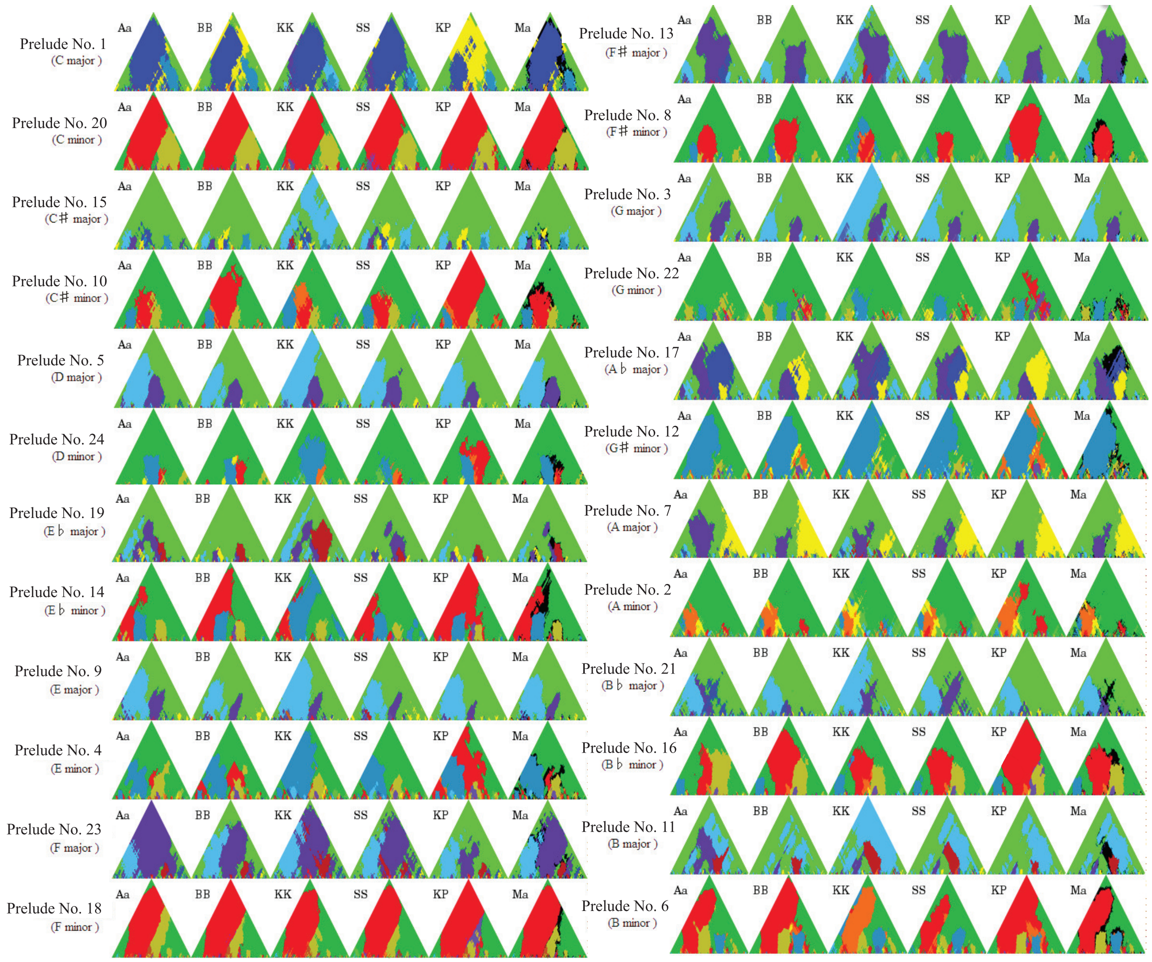


Figure 1.6: Functional keyscapes for F. Chopin's 24 preludes, op. 28.

Even though all three score-derived weight-sets perform noticeably better than the original KK weights, none of them can perform much better than the simple weight-set, except for the KP weights in major keys. From the top-level test where the simple weights perform the best, and the key-area test where only the BB weights perform (insignificantly) better than the simple weights, it can be surmised that all similar weight-sets derived from musical scores will perform at about the same level as these three examples. The simple weights are close to the maximum performance which can be expected from the KS algorithm. Any significant improvement would require additional complexity, such as dynamic switching

	J.S. Bach, The Well-tempered Clavier — Book I preludes						Book I fugues						Book II preludes						Book II fugues						Chopin preludes op. 28					
key	AE	BB	KK	KP	S	M	AE	BB	KK	KP	S	M	AE	BB	KK	KP	S	M	AE	BB	KK	KP	S	M	AE	BB	KK	KP	S	M
C maj	74	74	70	82	74	80	43	59	32	80	43	46	20	23	26	26	24	22	85	88	77	92	83	87	84	83	37	87	85	86
c min	87	85	76	74	65	81	88	81	82	36	83	83	14	11	18	10	16	13	90	91	90	72	90	90	83	86	80	83	77	82
C♯maj	63	67	50	68	68	67	65	73	31	75	67	69	79	82	34	81	72	79	92	91	85	94	88	92	39	20	2	24	29	23
c♯min	86	81	64	73	37	75	89	78	88	37	88	87	65	47	55	29	60	56	84	51	57	10	61	57	80	85	39	81	72	80
D maj	64	63	50	68	54	62	60	77	74	76	79	75	48	62	34	61	53	54	30	76	37	83	70	67	54	77	41	78	72	71
d min	47	51	75	56	70	62	84	86	71	79	73	82	75	76	56	59	77	73	90	90	89	77	89	89	55	55	32	45	43	46
E♭maj	64	86	67	88	76	76	46	71	50	88	52	56	65	86	49	84	78	78	90	91	46	89	85	89	72	73	60	77	74	73
e♭min	56	72	33	64	71	65	78	75	80	26	83	76	61	43	37	27	57	47	38	25	30	9	65	31	59	60	59	52	42	55
E maj	38	51	28	39	41	39	53	74	46	89	62	65	52	68	45	65	66	62	70	83	37	92	57	72	11	13	8	14	41	12
e min	34	46	61	48	71	49	26	11	26	7	13	13	60	51	34	26	59	51	73	46	62	19	59	57	48	65	49	67	81	62
F maj	3	7	6	7	8	6	24	44	12	67	26	31	10	33	12	59	25	24	67	73	67	79	74	73	82	84	38	77	65	77
f min	79	79	53	70	71	74	77	61	56	19	70	62	26	28	36	24	33	29	75	66	51	25	54	57	68	71	48	69	67	68
F♯maj	41	57	27	59	43	45	34	63	35	69	50	50	40	65	28	72	50	51	40	79	56	79	70	69	85	86	75	89	84	87
f♯min	73	62	46	45	64	60	90	87	82	36	84	85	70	64	63	46	76	67	82	42	76	3	80	74	44	65	19	55	32	46
G maj	18	23	16	21	25	21	32	52	32	63	53	51	62	74	27	70	65	67	43	69	8	77	56	55	61	53	46	63	58	58
g min	46	56	51	51	40	48	65	52	71	23	68	62	78	76	70	64	73	74	56	23	59	10	71	49	31	46	37	47	30	38
A♭maj	45	65	35	57	64	56	44	78	47	81	71	70	32	57	30	55	43	44	78	81	77	78	77	79	23	24	34	33	51	32
g♯min	59	46	46	38	54	50	71	64	50	10	72	65	40	38	25	26	32	33	93	88	85	22	87	89	56	58	54	52	55	56
A maj	40	75	48	78	74	71	53	86	63	89	78	79	50	59	67	69	60	62	19	65	11	71	37	38	86	87	61	85	84	86
a min	34	22	38	18	29	27	69	27	57	4	66	55	73	72	56	56	62	67	77	67	79	22	75	75	16	24	17	20	16	21
B♭maj	74	76	20	69	58	69	29	54	28	72	37	39	58	73	41	75	58	66	58	81	63	79	70	72	22	24	27	31	30	28
b♭min	81	84	78	78	79	82	37	32	70	24	65	40	52	33	51	23	50	45	79	38	73	9	61	64	67	72	60	49	69	66
B maj	25	40	31	52	38	37	70	84	54	84	77	79	45	65	20	61	50	52	25	82	59	88	66	72	94	94	68	93	90	92
b min	84	69	66	25	83	74	78	53	68	7	71	67	41	28	35	21	51	35	53	15	23	5	36	21	68	83	77	74	66	73

Table 1.2: Percent coverage of the tonic in the keyscape plots (colored green) in Figures 1.2 through 1.6.

between weight-sets in the key-identification process, or requiring more sequential information in the calculations, which will come at the cost of generality in the key-finding algorithm.

1.2 Keyscape applications in music analysis

Chapter 4 presents post-processing techniques of the raw key-analysis data used to create the keyscape plots in order to improve portions of the plot where the key-finding algorithm fails to find the best musical answer. The primary difficulty in localizing key regions is that using pitch distributions to determine a key will give weak answers at key modulation boundaries. For closely related keys this ambiguity is magnified (there is little difference between a tonic key and its dominant in terms of both pitch-class content and pitch-class distributions).

$\boxed{}$ = best result ⁝ ⁝ = better than simple weights

	weight set:	AE	BB	KK	KP	SS	Majority
Major keys:	mean	0.70	0.90	0.58	$\boxed{0.96}$	0.83	0.82
	median	0.73	0.96	0.57	$\boxed{1.00}$	0.88	0.88
	sd	0.22	0.14	0.23	$\boxed{0.11}$	0.15	0.16
Minor keys:	mean	$\boxed{0.91}$	0.81	0.81	0.56	0.87	0.84
	median	$\boxed{0.99}$	0.88	0.84	0.61	0.93	0.87
	sd	0.14	0.21	0.17	0.30	0.14	$\boxed{0.13}$
Major and Minor combined:	mean	0.81	$\boxed{0.86}$	0.69	0.76	0.85	0.83
	median	0.91	$\boxed{0.93}$	0.72	0.90	0.90	0.87
	sd	0.21	0.18	0.23	0.30	0.14	$\boxed{0.14}$

Table 1.3: Tonic-area test results based on Bach WTC Books I & II as well as Chopin preludes, op. 28 (120 compositions—60 in major keys and 60 in minor keys).

After post-processing of the keyscape to remove likely local key-analysis errors, the plots are more suitable for interpreting the key regions present in the music. As an example, consider the following commentary on the harmonic structure for the three piano sonatas in C minor by Ludwig van Beethoven. Even though they are labeled as being in C minor, each movement will have its own secondary key regions, and the primary key for each movement is not necessarily C minor.

The keyscape plots provide a quick overview for examining this harmonic structure. Each keyscape in Figures 1.7 through 1.9 represents a movement in each sonata, with the first two sonatas containing three movements and the last sonata only two movements. An interesting common aspect of these three sonatas is that very little of the dominant key (G major, represented by light blue) is present throughout all movements.

1.2.1 Beethoven piano sonata no. 5

The outer movements of sonata no. 5, (op. 10, no. 1) shown in Figure 1.7 are in C minor, while the inner movement is in the submediant key of A♭ major. These key designations are represented at the top of the keyscape plots and also form the background structure of the plot. Secondary key regions are superimposed along the time-line at the bottom of the plots on top of these backgrounds formed by the primary key.

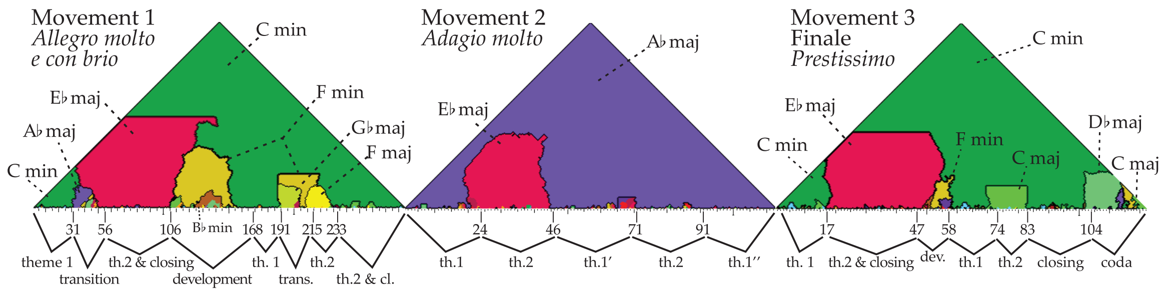


Figure 1.7: L. van Beethoven. Piano sonata no. 5, op. 10/1, composed *c.*1795–97).

Both the first and last movements of this sonata show the prototypical harmonic structure of a sonata form. Both movements start in the tonic key of C minor and then modulate to the relative key of E♭ major (shown in red) for the exposition of the second theme. Following the second theme is a development section which contains short contrasting key-regions, followed by a recapitulation in the tonic key.

The sonata form of the first movement is the more elaborate of the two outer movements, with a transition between the first and second themes which is mostly represented by the key of A♭ major—serving as the submediant of C minor as well as the subdominant of E♭ major. The development section transitions between many key areas, as is to be expected; however, the key-finding algorithm says that at a deeper level, the development section is functioning as the minor subdominant of C minor (i.e., F minor).

The transition from the first theme to the second theme in the recapitulation contains two unusual characteristics. First, this transition starts with a modulation from C minor to G♭ major, which is the most distant modulation possible, having a tonic movement of a tritone. The musical key then drops a semi-tone to F minor until finally arriving at F major at the start of the recapitulation of the second theme. A recapitulation of the second theme into the subdominant key (F major) is unusual, since the second theme is expected to be repeated in either the major or minor variant of the tonic key (C minor). However, Beethoven extends the sonata-allegro form at this point and subsequently repeats the second theme in the expected key of C minor, perhaps to mirror the subdominant relationship of A♭ to E♭ which occurs in the exposition's transition into the second theme.

A noticeable difference in the keyscape plots between the first and last movements is the lack of a transition between the first and second themes in both the exposition and in the recapitulation of the last movement. In addition, the recapitulation of the first theme occurs immediately at the modulation back to the tonic from a shorter development section, while in the first movement the modulation to the tonic key anticipates the recapitulation before the end of the development section, foreshadowing the return of the first theme. The recapitulation of the second theme is treated more conventionally in the third movement, being transposed into the parallel tonic of C major, which retains the major modality of this theme. However, the closing material after the second theme, which was originally in E$\flat$ major, is subsequently presented in the minor tonic key. An unusual coda closes the sonata with another distant key modulation to D$\flat$, which reflects the minor-second drop in the key from G$\flat$ major to F major in the transition to the second theme recapitulation in the first movement.

The middle movement has an overall structure of ABABA. The melody of the A section remains mostly unchanged in each A section but the elaboration of the harmony changes each time. The key of the movement is A$\flat$ major, which functions as the submediant of C minor. The harmonic construction of this slow movement is simple, helping to project a calmness and song-like character in contrast to the dramatic key modulations found in the outer movements. The only notable harmonic feature is that the second occurrence of the B section is repeated in the tonic key of the movement, modulating out of the dominant key used in its first statement. A cadence on the dominant in the first two A sections is changed to the tonic key in the third and final A section of the movement. This dominant cadence transitions into the dominant key of the first B section, while in the second appearance of the B section it is used to modulate to the tonic key.

An interesting pan-movement structure in this sonata is the use of E$\flat$ major for the secondary theme of all three movements (red regions in the plots). For the outer movements, E$\flat$ major functions as the relative major of C minor, while for the inner movement it functions as the dominant of A$\flat$ major. Also, in all three movements this secondary E$\flat$ material is transposed to the tonic when it returns later in the movement.

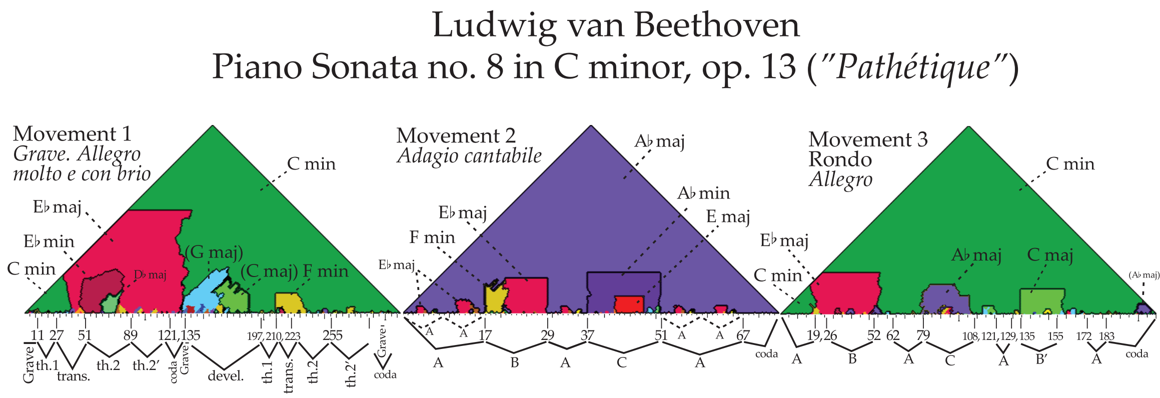


Figure 1.8: L. van Beethoven. Piano sonata no. 8, op. 13 ("Pathétique"), composed 1797–98.

1.2.2 Beethoven piano sonata no. 8

Figure 1.8 displays the internal key structure for Beethoven's 8th piano sonata (op. 13). This sonata also has the typical sonata form for its first movement, and the second theme is again in the relative key of E♭ major. Another distinctive similarity to sonata no. 5 is the use of the submediant key of A♭ major for the middle movement, and E♭ major is used as the main secondary key region in each movement.

An unusual aspect of the sonata form in the first movement is that the second theme is first presented in E♭ minor rather than the relative major, which is more commonly used for a sonata in a minor key. The development section starts with short sequences in various tonal areas, each 5–10 measures long, which the key-finding algorithm merges into the dominant (G major) at larger time-scales. The development section then gradually shifts into the tonic key through C major before the recapitulation of the first theme. The recapitulation of the second theme is handled more traditionally with a direct statement in C minor, although there is a gradual transition into the second theme from F minor.

The second movement is in song form, with episodes between the repeats of the A section being harmonically more elaborate than the treatment found in the second movement of sonata no. 5. The B section starts in F minor (the submediant of A♭ major, which is in turn the submediant of C minor) and then modulates after one phrase into E♭ major, the dominant of A♭. The C section is presented in the parallel key of A♭ minor which has its own internal key structure, modulating into E major—another distant key modulation by an augmented fifth reminiscent of the tritone motions found in sonata no. 5. The outer sections labeled A in the middle movement contain two repetitions of the theme, which consists of a period of two phrases, with the first phrase cadencing on the dominant (E♭ major). This is visible in the plot as a small reddish region which occurs twice in the first and third A sections, and once in the second A section.

The third movement is in a rondo form. This form is clearly seen in the plot, with the multiple A sections alternating with other material in different keys. This movement has harmonic structures which reflect those of the other movements. The second and third movements both consist of sections alternating between the tonic key and other keys. The return of the B material in the major tonic key in measures 135–155 in the third movement also mirrors the recapitulation of the second theme in the first movement, in which the original second-theme material transposes from E♭ major into the minor tonic key.

1.2.3 Beethoven piano sonata no. 32

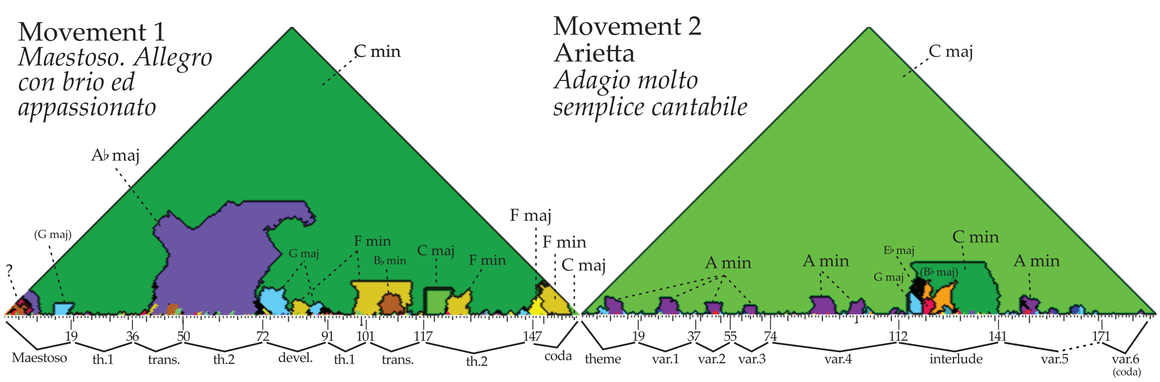


Figure 1.9: L. van Beethoven. Piano sonata no. 32, op. 111, composed 1821–22.

Sonata no. 32, shown in Figure 1.9, is the last piano sonata composed by Beethoven and consists of only two movements instead of the usual three. The texture and motivic treatments of each movement are quite complex, yet they both retain classical harmonic forms that are also recognizable in the keyscape plots of the previously mentioned sonatas.

The first movement is in sonata-allegro form, with the second theme now in the atypical key of the submediant (A♭ major) rather than the relative major (E♭), as would be expected in a minor-keyed sonata—perhaps as a compensation for the lack of a middle movement, which was placed in the submediant key in the other two C minor piano sonatas. Like the previous two sonatas, the recapitulation contains significant material in the subdominant (F minor). In addition, the transition between the first and second themes in the recapitulation passes through the subdominant of the subdominant (B♭ minor), and the coda section at the end of the piece functions as a large IV-I cadence to the final chord in C major. Use of keys containing more flats than C minor (F minor, B♭ minor, and A♭ major) conveys a somber/dark mood. Conversely, when going in the opposite direction in the Circle of Fifths, keys generally have a brighter character by instead raises diatonic pitch-classes (e.g., F♮ to F♯ from C major to G major).

The opening material of the *maestoso* section is an interesting case which causes confusion for the KS key-finding algorithm for several reasons. In terms of chords, the opening two measures of the first movement clearly defines the key of C minor with the chord sequence vii°_{7}/V, V, i, (V_6). As well as emphasizing a secondary chord (vii°_{7}/V) outside of the primary key, this sequence contains the only occurrence of the tonic triad (C minor) within the first ten measures of the music, thus avoiding the tonic and dominant scale degrees necessary in the KS algorithm for defining a key. Chromaticism throughout the opening measures also confounds the methodology of the KS algorithm. The three fully diminished chords found at the start of the first, third, and fifth measures contain all 12 chromatic pitches, although the diminished chord in m. 5 only implies G and B♭ before breaking the sequence pattern to form a B♭ minor chord. In addition, most of the chords in measures 6–10 are generated by chromatically altering one pitch at a time, similar to the harmonic techniques used in F. Chopin's prelude in E minor, op. 28, no. 4.

The second movement has a variations form, with each variation having the same harmonic structure as the theme, starting and ending in C major with a middle section modulating to the implies (A minor). In variation 4, the repeats are written out as the variational procedures become freer, so two A-minor regions are visible in the plot at measures 74–112. After the fourth variation, a brief foray into C minor interrupts the variation form. This key region is entered via the dominant, then the relative major (E♭), and exits with a direct modulation back to the parallel major (C) for a return to the primary thematic material in measure 141. The coda section at the end of the movement has similar properties to the A sections in the middle movement of sonata no. 5, because it minimizes the secondary key near the end of the piece to focus instead on closing the movement in the tonic key. The presence of the A-minor key-region, which was more predominant in the variations, has now been minimized in the coda section starting at m. 171 in favor of greater emphasis of the tonic key as the movement comes to a final cadence.

1.3 Performance analysis

A second general application of the scape-plotting domain is given in Chapters 5 and 6, where it is used for musical performance analysis. In these chapters, different performances of the same composition are correlated with each other in terms of the tempo per beat as well as loudnesses sampled at the beats. Whereas the key analysis technique discussed in the previous section compares actual and expected pitch-class histograms, the correlation of performance data compares arbitrary performances to each other. The "timescapes" in Chapter 5 show the nearest-neighbor performances for a particular recording at every possible timescale in exactly the same manner as the best-identified key is displayed in keyscapes. Each colored region is now the most similar performance rather than the identified key.

However, there is a significant difference in the underlying meaning of similarity between the keyscapes and timescapes. In terms of key analysis there is a fixed set of pitch-class patterns for each key, while in performance analysis there are no absolute prototypes—only other performances can be used as a basis of comparison. Chapter 6 describes a process for generating a numerical similarity score between two performances of the same work. The score is based on how well two performances match each other when compared to a set of performances identified as least similar, resulting in a more consistent similarity measurement across compositions than simple correlation can produce.

1.3.1 Data entry

Audio recordings provide the source data for the numerical comparisons between performances in Chapters 5 and 6. The time of each beat in a recording is extracted in a semi-automatic process illustrated in Figure 1.10. An audio editing program called *Sonic Visualiser*[2] was used to tap to the beats in real time while the music was playing. This inserts beat markers in the audio display window which can be adjusted to remove the reaction delay of the listener tapping to the music.

Several audio analysis plugins for *Sonic Visualiser* were developed specifically for correcting the tap responses of the data encoder so that the timings of the pianist could be more accurately observed and extracted.[3] Figure 1.10 illustrates the most important of these plugins:

- *SpectralReflux*: This plugin identifies note onsets in percussive timbres (such as piano). It is based on the spectral flux technique [6] with enhancements to remove spurious peaks following a note onset to improve efficiency with visual correction of the data and increased sensitivity at low sound levels.

- *HarmonicSpectrogram*: This plugin displays the raw output of the harmonic product spectrum as a spectrogram. [5] This process was originally used to identify monophonic pitch in vocal timbres, but it functions well to highlight the melodic line in homophonic music such as the Chopin mazurkas by removing redundant information generated by harmonics, which perceptually clutter the view in a normal Fourier-based spectrogram.

- *PowerCurve*: The SpectralReflux plugin does not compensate for noise caused by scratches in old records. Therefore when extracting timing data from old recordings, the PowerCurve plugin is also used to help localize note onsets by examining the slope of the smoothed power values extracted from the audio data.[4] The smoothing process hides clicks and pops (very short bursts of noise). And by smoothing the

[2] http://www.sonicvisualiser.org
[3] http://sv.mazurka.org.uk/download
[4] http://sv.mazurka.org/MzPowerCurve

raw power measurements in both the forward and reverse directions, the slope of the smoothed power curve will contain peaks centered at note onsets.

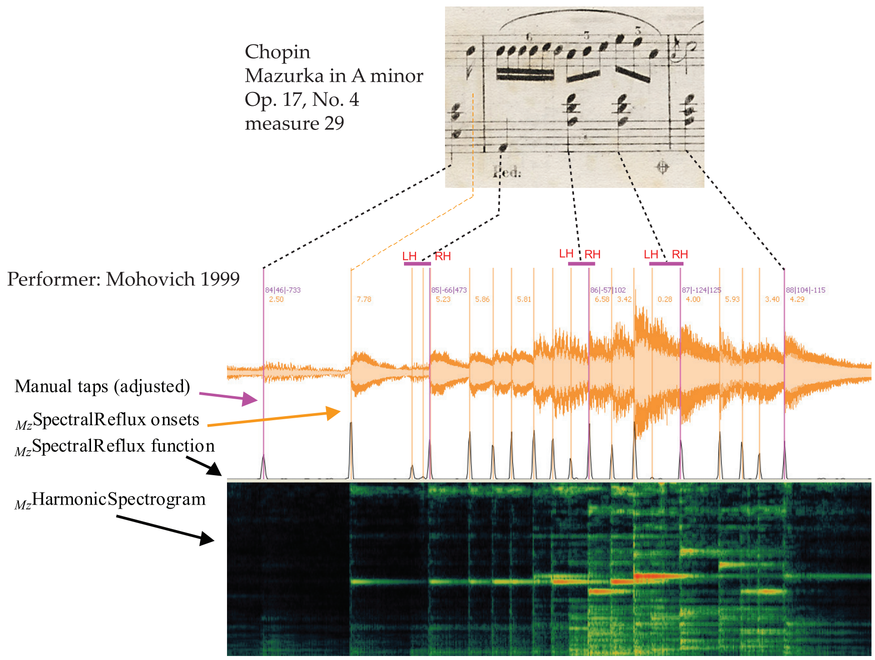


Figure 1.10: Screen-shot in *Sonic Visualiser* of the analytic tools used for data extraction of the beat tempo, along with part of the musical score matching to the visible portion of the audio.

In Figure 1.10, the relationship between the score and a screen-shot taken of the corresponding section of a recorded performance in *Sonic Visualiser* gives an overview of the performance details which can be extracted from a good-quality recording. Points in time representing beats are marked with dashed lines above the orange-colored audio waveform linking to the score. *Sonic Visualiser* allows for analytic data to be superimposed on this audio waveform display. The black curve underneath the waveform is a continuous onset detection function (sampled every 10 ms) which is generated by the SpectralReflux plugin, and which is very useful for accurately finding note onsets in piano music. The vertical orange lines rising through the center of each peak in the continuous onset function are the identified note onset times as defined by these peaks. The purple vertical lines marking the beat locations were initially tapped in real time while listening to the performance, and then subsequently adjusted to coincide with the correct note onset.

Once the performance has been tapped to in real time, the tap times and the note-onset times which were automatically extracted from the audio data are input into an online program called *tapsnap*,[5] which moves the tap event to the nearest onset time if one is found within a certain timing tolerance. The manually corrected tap times are then loaded back into *Sonic Visualiser* and proof-audited to ensure that the tap events are attached to the correct note onsets. When chord tones generate multiple note-onsets for a single beat location in the score (as seen in Figure 1.10), the onset time of the melody note was chosen, since the expressive timing being studied is more likely to occur in the melodic line than in the harmonic accompaniment. The final data set extracted from each recording used for the analyses in Chapters 5 and 6 consists of a list of absolute times within the recording at which beats occur. The absolute beat times are then converted into time differences between beats (often called IOI's, or inter-onset intervals, in music-perception literature). These timing differences are converted into tempo values expressed in beats per minute according to the equation:

$$\text{tempo} = \frac{60}{\text{timing difference in seconds}} \tag{1.1}$$

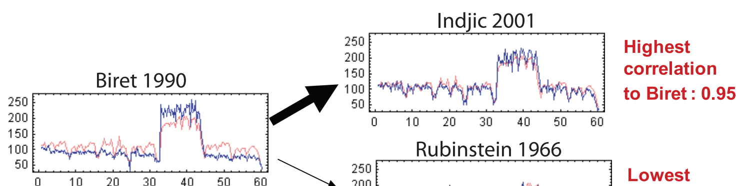
and referred to as the *beat tempo*. Tempo values are used instead of delta times for performance comparisons in order to minimize the outlier behavior of long pauses in the music when calculating correlation values, although this is a small effect. These final beat-tempo values are the basis for numerical measurements made for comparing multiple performances with each other.

Note that there is much more performance information extractable from the audio data than is addressed in Chapters 5 and 6. For example, each peak in the onset function occurring between the beats represents rhythmic events falling off the beat in the score. The first offbeat in the example is a sixteenth-note, which is played as if it were an eighth-note, since it falls halfway between the adjacent beats. The triplets in the second beat of the measure are played at the same rate as the sextuplets in the first beat.

[5] http://mazurka.org.uk/tapsnap

A particularly interesting performance feature seen in Figure 1.10 is that the left-hand chords and the right-hand melody notes are not played simultaneously (note the LH/RH marks above the waveform). In all cases, the left-hand notes are played before the right-hand notes. This is not (usually) caused by sloppiness of the pianist, but rather is a Gestalt principle used to maximize perceptual independence of the voices. [1] In other words, the melody is of primary importance and must be played louder than the accompaniment. If the left-hand chords were played exactly at the same time as the melody note or slightly after, their note-attacks would be obscured, and the musical texture would sound thinner. By playing the left-hand notes slightly ahead of the melody note, the left-hand notes can be played very quietly and unobtrusively while still being clearly audible. Polish, and particularly Russian, pianists employ this hand-offset technique in Chopin's mazurkas.

1.3.2 Performance correlation



		Br	Ch	Fl	In	Lu	R8	R6	Sm	Un
Biret	1.									
Brailowsky	0.92	1.								
Chiu	0.81	0.81	1.							
Friere	0.83	0.86	0.86	1.						
Indjic	0.95	0.91	0.86	0.88	1.					
Luisada	0.85	0.84	0.81	0.84	0.88	1.				
Rubinstein 1938	0.62	0.66	0.76	0.73	0.66	0.67	1.			
Rubinstein 1966	0.5	0.55	0.74	0.7	0.59	0.61	0.77	1.		
Smith	0.55	0.65	0.67	0.74	0.63	0.56	0.62	0.59	1.	
Uninsky	0.86	0.85	0.89	0.89	0.9	0.89	0.75	0.69	0.64	1.

Figure 1.11: Beat-tempo correlation matrix for 10 performances of Chopin's mazurka, op. 68/3. Top of figure shows the best and worst tempo-curve correlation matches to Biret's tempo-curve. Red line in plots is the average tempo by beat for all performances.

The top of Figure 1.11 shows three beat-tempo plots for Chopin's mazurka in F major, op. 68, no. 3. Underneath these plots is a Pearson correlation matrix which compares the similarity of performers' tempo-curves to each other. When correlated to itself a value of

1.0 is produced for each tempo-curve. This indicates that it is maximally similar to itself, as can be seen in the diagonal edge of the matrix. The correlation between Biret's performance and others is given in the first column. The highest correlation value to Biret is 0.95 with Indjic, and the lowest correlation is to Rubinstein at 0.50. Note that the red line in the plots shows the average tempo-curve for all performances, and it is useful for comparing between performances. For example, Biret plays slower than the average performer throughout most of the mazurka, but faster than most performers in the middle section marked *più vivo* in the score. In contrast, Rubinstein plays most of the mazurka much faster than the average performance, but slightly slower than average in the *vivo* section. Indjic plays quite close to the average performance tempo throughout the mazurka, and this neutral interpretation has interesting applications. [3]

The correlation values give a basic indication of the similarity between the tempo-curves of two performances. For example, visual inspection demonstrates that the tempo-curve of Indjic looks more similar to Biret's tempo-curve than does Rubinstein's. In some fields (such as epidemiological research) correlation values are used to try to identify causes and effects, such as the efficacy of a medication. However, a search for direct relationships is not being done in Chapters 5 and 6. The focus is on observed similarities between performances, which can be further analyzed as discussed in the last section of this chapter. Chapter 3 briefly points out that Pearson correlations and Discrete Fourier Transforms are essentially equivalent forms of similarity measurements. The application of correlation in Chapters 5 and 6 is related to the use of Fourier analysis in audio signal processing. For Discrete Fourier Transforms, an audio signal is correlated using the dot-product to a set of harmonically related sinusoids.

Correlation values by themselves are difficult to interpret. For example, Biret and Indjic tempo-curves have a correlation of 0.95, which is a higher similarity caused by the simplicity of the mazurka which has a large-scale tempo structure influenced by the *vivo* section to which all pianists respond to in varying degrees. Chapter 6 presents a method which gives a more consistent similarity measurement across different compositions, regardless of their expressive simplicity or complexity.

1.3.3 Nearest-neighbor performance display

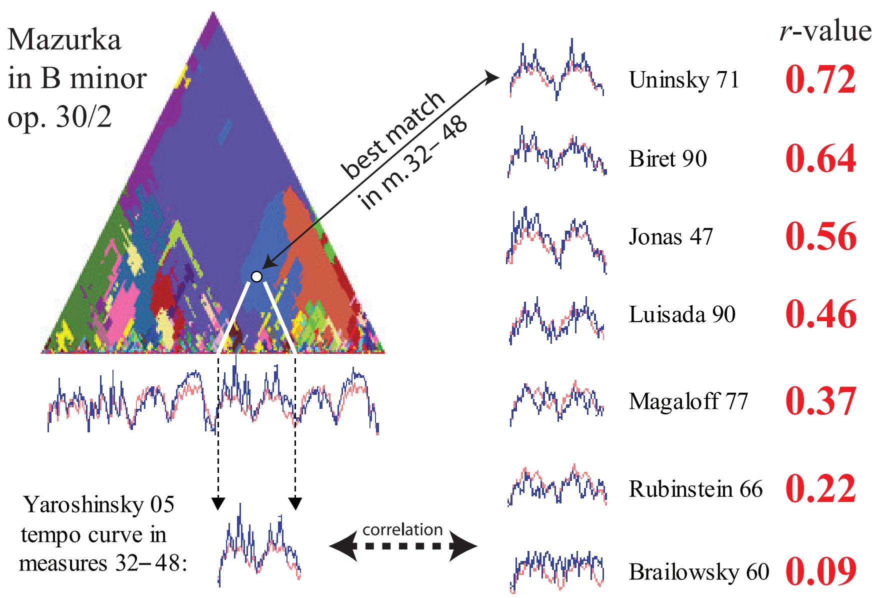


Figure 1.12: Nearest-neighbor plots for Yaroshinsky's 2005 performance of Chopin's mazurka in B minor, op. 30/2, with example correlation calculations to other performances at one point in plot.

Comparing correlation values at all sequential segmentations of tempo-curves, a *time-scape* can be generated in a manner similar to the keyscape plots discussed in the previous section. In this case, each plot represents highest similarity between a two performances when compared to a set of different performances. Instead of marking the best key for a segment of the music, each point in the timescape plot represents the particular performance which generates the highest correlation to the target performance at the given position and time-scale within the composition. Figure 1.12 illustrates such a performance timescape. One of the points in the plot is highlighted. The tempo data values represented by this point are extracted from the full tempo-curve and correlated with the corresponding points in all of the other performances. The performance producing the best match (Uninsky in this case) is then represented in the plot by an arbitrary color. Chapter 5 describes these types of plots in more detail.

This kind of analysis of the performance data can also be extended to dynamics data. Also, It can be applied to filtered versions of performance data to look in detail at either phrasing (low-frequency features) or accentuation (high-frequency features).[6] Figure 1.13 shows nearest-neighbor scape plots for multiple feature types. Raw tempo and dynamic features extracted from the recordings are low-pass filtered to generate smoothed tempo and dynamic curves. This procedure primarily extracts phrasing information from the tempo-curves. The residual features are the original features with the smoothed data subtracted, leaving high-frequency features which relate musically to accentuation—temporal accents from tempo-curves or dynamics accents from dynamics curves. The last three scape plots in Figure 1.13 show the nearest-neighbor performances when an equal mix of tempo and dynamic features are considered.

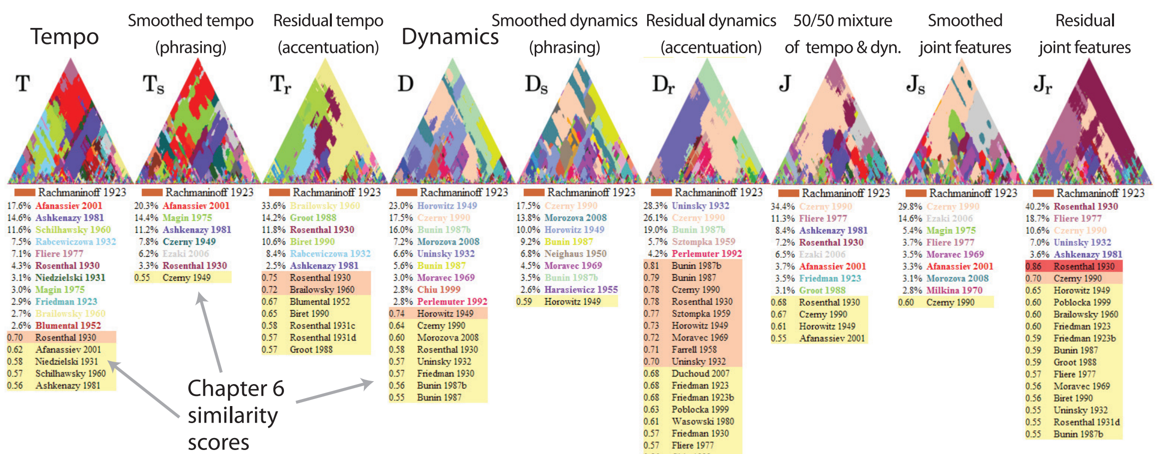


Figure 1.13: Nearest-neighbor plots for Rachmaninoff's 2005 performance of Chopin's mazurka in C♯ minor, op. 63/3, for tempo and dynamic features, tempo and dynamic sub-features, and an equal mixture of tempo and dynamic features.

Two sets of numeric values are shown under each plot in the figure. The first set of values reported as percentages is a list of the most representative nearest-neighbors, and indicate the amount of color-coded regions in the plot which represent that performance. More useful sets of numbers are given beneath these percentages, and these are Chapter-6 similarity scores. The tempo-curve for Rachmaninoff is most similar to Rosenthal's, with a

[6] see http://mazurka.org.uk/ana/pcor-all/mazurka63-3-noavg

similarity score of 0.70 between the two, which is roughly interpreted to mean "moderately similar" (orange background behind score). Four other pianists have similarity scores for the full-tempo feature are broadly categorized as "slightly similar" (yellow background): Afanassiev (Russian), Niedzielski (Polish), Schilhawsky (Austrian), Ashkenazy (Russian). The tempo-curve similarity to Rosenthal (Polish pianist and pupil of Karol Mikuli who was in turn a pupil of Frédéric Chopin) occurs primarily in the temporal accentuation (0.75 similarity score) and not in the tempo phrasing (0.47 similarity score). Likewise, Rosenthal also has moderate similarity to Rachmaninoff's dynamics accentuation (0.78 similarity). Rosenthal's accentuation similarity to Rachmaninoff becomes very similar when both the residual tempo and dynamics features are compared at the same time.

Again, this is not to say that performance similarity is the result of conscious imitation. A more appropriate analogy would be to think of performance style as analogous to a linguistic accent, or a proximity effect. Two people living in the same region will tend to speak similarly to each other, not because they directly hear each other speaking, but rather their speaking patterns are influenced by other people with whom they come in contact. Figure 1.17 shows an initial attempt to group pianists by their expressive performance "accents."

1.3.4 Forensic applications

An early unintended application of the nearest-neighbor performance timescapes was to uncover surreptitious copying of previously released commercial recordings under another pianist's name. [8] Timescapes will typically show close similarities between different recordings by the same pianist. The right side of Figure 1.14 demonstrates such a case. In the two timescape plots for recordings by Uninsky, the other recording by Uninsky appears as the best match at most time-scales, particularly on the longer ones. At shorter time-scales, different pianists will occasionally be more similar than his other recording due to random effects. However, the pair of plots on the left side of Figure 1.14 show a puzzling property: the two pianists, Eugen Indjic and Joyce Hatto, play so similarly to each other at every possible timescale that they cover each others timescapes nearly completely. This behavior is physically impossible, since the same pianist cannot play with such repeated accuracy. Further inspection of the audio revealed that the two recordings are time-stretched and filtered versions of each other in order to generate a superficial impression of different recordings.

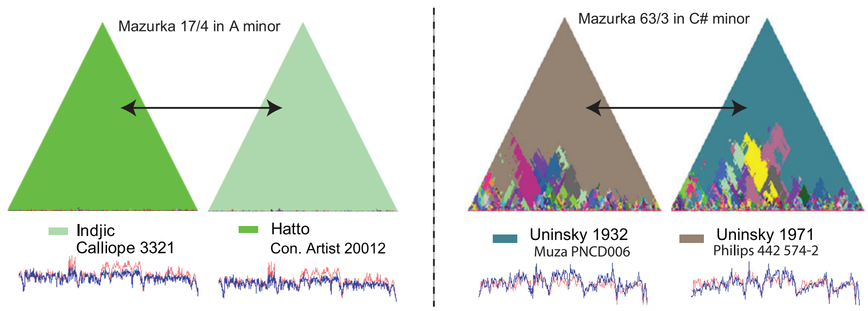


Figure 1.14: Example timescapes for Hatto/Indjic and Uninsky 1932/1971.

While the popular press has focused on the subsequent Hatto story due to its impressive scale of 100 recordings apparently generated by a reclusive ill pianist, other forged and likely forged recordings were created by Hatto's husband, William Barrington-Coupe (Barry) over the years. The Hatto story primarily covers the years 2003–2007 when the extensive number of performances by Hatto were released by her husband's small record label, Concert Artist/Fidelio. However, Ernst Lumpe provided to the author a 1993 cassette-tape release of Hatto's performances of the Chopin mazurkas which also turns out to be the same recordings which Concert Artist re-released in 2005 and which were copied from the same recordings by Indjic. This earlier duplication indicates a much longer history of performance borrowing in Concert Artist releases.

Prior to the mass-production stage, Barrington-Coupe had conducted research to see if people can actually identify a performer based solely on their recording:

> At a moment when she [Hatto] wasn't in the room, Barry offered to play an excerpt of a more recent recording of Hatto's: Rachmaninoff's Third Piano Concerto. He placed a cassette in the tape deck and they listened to its dizzying cadenza (the longer of two versions that Rachmaninoff wrote). Lumpe expressed admiration, only to be told, "Well, I've fooled you a little. I made a little joke. That's not actually Joyce, it's Andrei Gavrilov"—a former winner of the International Tchaikovsky Competition. Barry smiled. "But now here's

> Joyce," he said, and he put in another cassette, a different performance of the same cadenza. It was a lighthearted, insignificant jest, Lumpe thought, hardly a test or a ruse. [8]

Barrington-Coupe had released recordings under pseudonyms several decades earlier, which is probably the source of his interest in performance authenticity:

> In 1960, after the collapse of a company he was involved in called Saga Films, he created the Lyrique record label with Marcel Rodd, who had a record-pressing factory. It was then that Coupe first began to release records by artists under different pseudonyms - a practice not unheard of at the time. 'The repertoire was from the variety of master tapes now in Rodd's tape library,' wrote Ted Perry, one of Coupe's former colleagues in an unpublished autobiography. 'It was also, possibly, from some of Coupe's own tapes since he always seemed to have a lot of recorded material of unknown, not to say dubious, provenance.' [8]

Sergio Fiorentino (1927–1998) is another pianist who was posthumously tainted by the activities of the Concert Artist label. Figure 1.15 shows a list of mazurkas supposedly performed by Fiorentino on CACD 9200-2 (2003), recorded in the 1960's. However, this CD contains copies of commercially available recordings by at least three other pianists, which were initially uncovered by comparing timescapes in one of the mazurkas. Subsequent evaluation using more automated methods of audio similarity measurement identified the other two performers represented in the album.[7] It is possible that the remaining 15 mazurkas on the CD are actually by Fiorentino, since he was a recording artist for Concert Artist/Fidelio in the 1960's.

The 1993 cassette release of the Hatto mazurkas is interesting for another reason. A set of Chopin mazurkas which were released in the 1980's by Concert Artist/Fidelio and re-released in 2005 on CACD 91802 are performed by the famous French pianist Alfred Cortot (1877–1962). According to the liner notes, these mazurkas "were obviously recorded at diverse locations and dates presumably in the period 1950–1952." This is similar to the recording information provided in the Fiorentino mazurka CD liner notes: "Recorded at various venues, Hamburg, Paris, London, Hornsey, Greenwich & Guilford during the

[7] Developed by Michael Casey (Goldsmiths College, University of London / Dartmouth College): http://omras2.doc.gold.ac.uk/software/audiodb

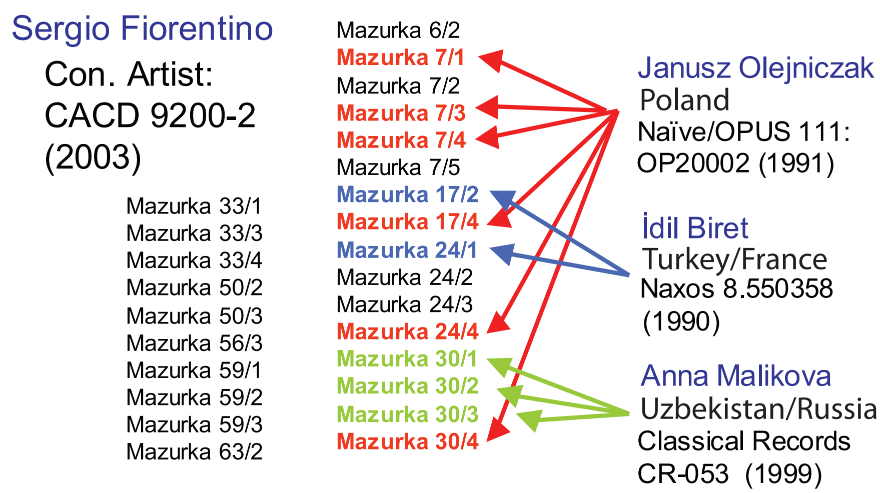


Figure 1.15: Three pianists "borrowed" for the Concert Artist label release of Chopin mazurkas performed by Sergio Fiorentino.

period July 1958 to February 1966." These varied recording conditions provide a useful explanation to allay suspicions caused by variability in the recording quality across pieces on the same album.

Cortot did not commercially release any Chopin mazurkas, yet the same recording label which produced the Hatto hoax offered the complete Chopin mazurkas performed by Cortot. The Turkish pianist, İdil Biret, who studied with Cortot even mentions these recordings on her website:

> A most important happening during this period was the discovery in succession of some extraordinary Chopin recordings Biret had not known about by three great pianists; performances which greatly inspired her. First came the complete 51 Mazurkas by Alfred Cortot on three cassettes bought privately in the UK. These were recorded by Cortot in the late 1950s and never released for unknown reasons.[8]

The end of Chapter 6 discusses one of these Cortot recordings, Chopin's mazurka in B minor, op. 30, no. 2. A 75% complete fragment of this mazurka was recorded by Cortot as part of a master class in the late 1950's released by Sony Classical (S3K89698) in 2005.

[8]From the webpage http://www.idilbiret.org/ENG/IBe13.htm with the quote first appearing no later than 2006 and still present in 2011.

According to the similarity metric developed in Chapter 6, the master-class Cortot recording and the Concert-Artist recording are quite dissimilar. Out of 36 recordings compared for this mazurka, the Cortot recordings were nearly always the least similar to each other in both tempo and dynamic features (Chapter 6, Table 6.5). This can be compared to the expected high similarity other performers generate between their performances as summarized in Chapter 6, Table 6.4. So, it is at least unlikely that mazurka op. 30, no. 2 on the Concert Artist label was recorded by Cortot. Given the history of mislabeling performances by Concert Artist as well as the lack of a clear provenance for the Cortot mazurka recordings, it is unlikely that many, if any, of the rest of the mazurkas were actually recorded by Cortot.

1.3.5 Performance similarity

Timescapes were not intended to catch performance copies such as those of Hatto and Fiorentino. These exact or nearly exact copies can be uncovered by less complicated processes. Rather, timescapes were intended to examine similarities between performances by the same pianists, between students and teachers, and to enable studies of performance-style changes over time.

Figure 1.14 (right side) shows what a typical timescape plot looks like for the same performers over time. In that figure, the two recordings of Uninsky (1910–1972) are separated by nearly 40 years, yet his performance interpretation remains fairly consistent over an entire career which began in Russia and ended in Texas.

Another timescape pair, between a student and teacher, is given in Figure 1.16. These two performances demonstrate the possibility of direct style transfer between pianists. The similarity between these two performers is unlikely to be caused exclusively by verbal prescriptions. It probably involves aural transfer of performance style via either live performance or recordings. Visual inspection of the two tempo-curves underneath each timescape plot shows that the main difference between their performances is that Nezu performs with a more pronounced mazurka meter and a faster tempo in the last quarter of the mazurka.

After graduating from the Tokyo National University of Fine Arts and Music, Nezu studied piano on the post-graduate level at the Academy of Music in Bydgoszcz (Poland)

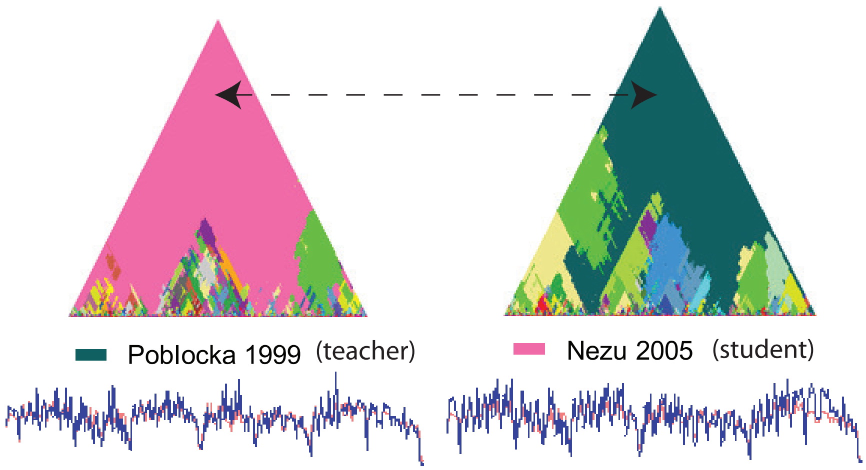


Figure 1.16: Beat-tempo similarity between student and teacher, comparing against 30 performances of Chopin mazurka, op. 24/2.

with Poblocka. Unlike linguistic accents, this style similarity occurs quite late in the performance learning process (unless perhaps Nezu had been listening to the Poblocka mazurka recordings before studying with her). In contrast to the Uninsky case (and other professional recording artists who maintain a fairly consistent performance interpretation over time), this may indicate that performance style becomes fixed in the later learning stages and is either plastic or ill-defined in earlier levels of musical development.

1.3.6 Future work

1.3.7 Performance clustering

The primary intent of the similarity metric defined in Chapter 6 is to be able to relate different performances to each other numerically so that further cross-composition analysis can characterize performance styles in broader contexts. The similarity metric in Chapter 6 is shown to be three to four times more accurate than plain correlation at identifying two performances by Anton Rubinstein out of a larger unlabeled collection of performances of the same piece when using a third performance by Rubinstein as the target performance. Likewise, other pianists tend to perform tempo—and to a lesser extent, dynamics—more similarly to themselves than to other pianists over time. This demonstrates a continuity of personal interpretation. Some pianists, such as Rubinstein, show more variability in their performance style. Others, such as Uninsky, produce performances that are consistent in both tempo and dynamics over many decades.

One complication in comparing performance features is that the extractable surface features are composites of many local and global musical intentions. These independent sub-features are difficult to separate. For example, Principal Component Analysis decomposition of tempo-curves results in separate sub-features that maximize the most variable portions of the data; however, these sub-features do not concentrate evidence of the uniqueness of a performer according to the evaluation method used with Rubinstein in Chapter 6. Instead, in terms of PCA decomposition, the uniqueness of a performer is widely distributed throughout all or many components. Likewise, filtering tempo-curves into low (phrasing) and high (accentuation) frequency components does not identify the performer as well as the unfiltered tempo or dynamics curves as discussed in Chapter 6.

Some musical sub-features may also dominate over others in similarity measurements. For example, Brailowsky always performs mazurkas with a strong metric pattern, consisting of a short first beat and longer second and/or third beat. Therefore, when other pianists match to him, this primarily means that they also play with a strong mazurka meter throughout their performance. This similarity may or may not be interesting and of course is limited to the mazurka genre. On one hand, knowing how strongly a performer applies this metrical pattern can be useful to relate performances to each other. But on the other hand, if this metrical pattern overpowers other equal or more important musical features, then a strong similarity measurement caused only by metrical strength is more likely to be random in terms of broader significance in performance style.

Putting aside all of the difficulties in interpreting the similarities between different performances, it is possible to look for patterns between performances of the same piece by using the similarity metric derived in Chapter 6. The tempo similarity network shown in Figure 1.17 demonstrates a prototype for such analyses. In the figure, each black dot represents a particular performance recording. Spatial layout of the performances is arbitrary, and only the colored lines between them convey the similarity information, with a color mapping to numerical similarity ranges given in the lower left. Abbreviations give the first three letters of the performer's surname and the date of the performance.[9]

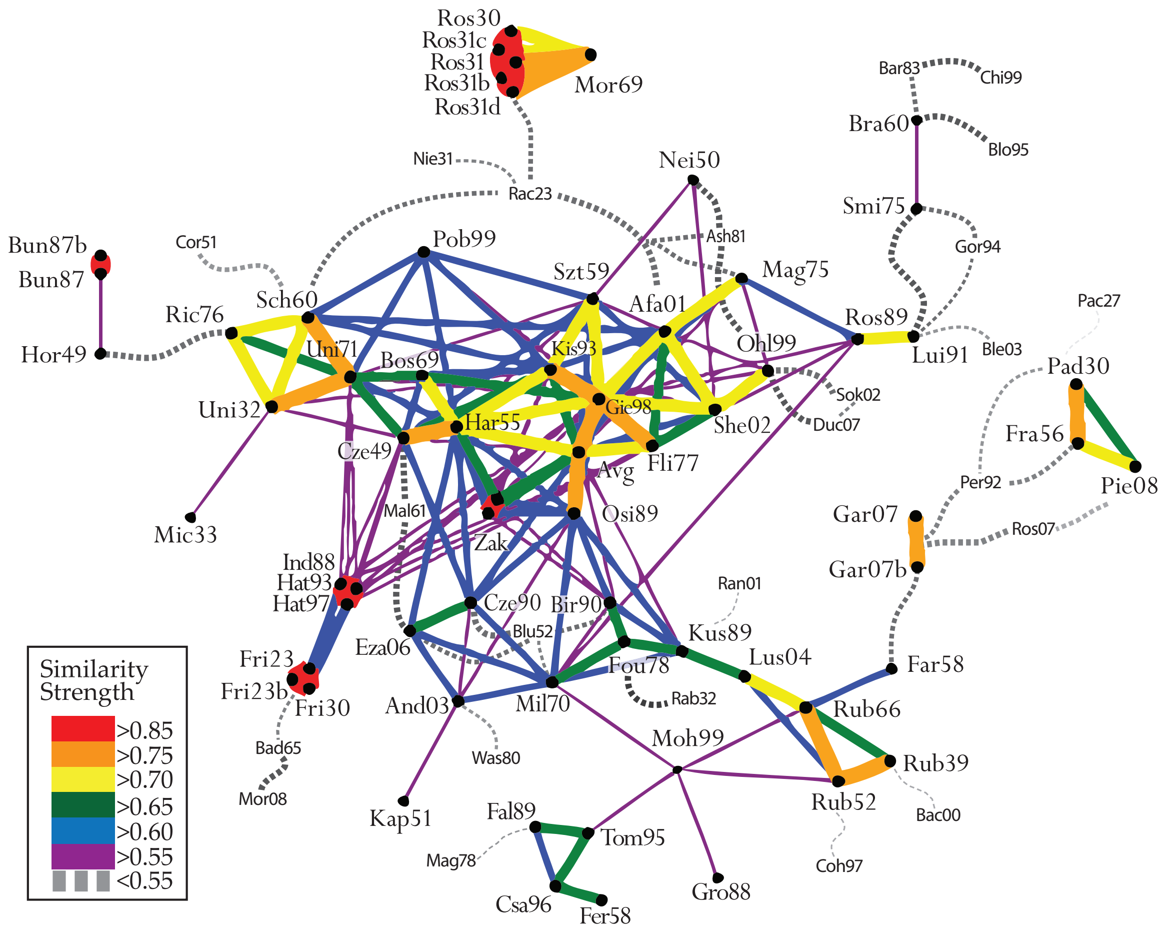


Figure 1.17: Tempo similarity network for performances of Chopin's mazurka in C♯ minor, op. 63/3.

Stronger connections are given shorter lines to place neighboring performances close to each other. Red-colored connections are highly similar and represent either the exact same performance, or a very similar separate performance by the same performer. When a performance has no particular similarity to any other performance, a single dashed gray line connects it to its most (weakly) similar performance. These performances are more

[9]see http://mazurka.org.uk/info/discography for expansions of the abbreviations.

individual in their tempo-curve interpretation of the mazurka as compared to other performances of the same mazurka, and any similarity to other performers is somewhat arbitrary and difficult to distinguish from chance.

Several performance groupings emerge from the plot. Figure 1.18 illustrates six general groupings determined manually. The most coherent group is labeled “Hungarian”, which contains three Hungarian pianists (Falvay, Ferenczy and Csalog) along with a performer from the neighboring country of Slovenia (Tomšič). A weakly defined southern European group includes two Italian piano students (Pie08, Ros07) along with a Spanish pianist (García Casas) and a French pianist (François). These are grouped with the famous statesman/pianist Paderewski of Poland, whose performance is not found to be particularly similar to any Polish or Russian pianists in this mazurka.

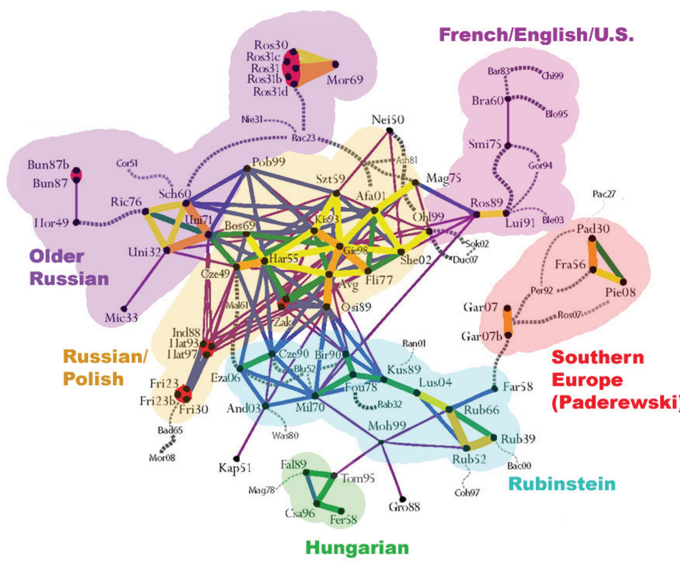


Figure 1.18: Tempo similarity network for performances of Chopin mazurka in C♯ minor, op. 63/3.

The average tempo of all performances (Avg) lies in the largest group which shows the greatest cohesion. This group can be described broadly as containing most Polish/Russian pianists in the network, and perhaps represents the core performance tradition for this work. A weakly defined group consisting mostly of Russian pianists can be seen on the left edge of the main Polish/Russian group. An interesting pairing between pianists is that between the Rosenthal 1930/1 and Moravec 1969 recordings, which may indicate that Moravec had prior exposure to the Rosenthal recording.

Two other satellite groupings are set off from the main cluster. One is a group consisting predominantly of French (Lui91, Ble03), English (Smi75), and North American (Blo95, Chi99, Ohl99) pianists, as well as a Brazilian (Bar83). The other satellite group is best described as being connected via the performance style of Anton Rubinstein. It contains a wide range of pianists, possibly representative of an international style for this mazurka. The pianists come from America (Kus89, Ran01, Lus04, Kap51), Poland (And03, Cze90, Rab32–a student of Rubinstein), the Netherlands (Gro88), Japan (Eza06), and Turkey (Bir90). It also includes widely migrating pianists Blu52 (Poland, France, Brazil), Mil70 (Russia, France, England), and Fou78 (China, Poland, England).

1.3.8 Phrase analysis

The scape-plotting domain is useful for other structural analysis applications in music. A brief application to phrase analysis in musical performances is provided here. In Chapters 4 and 5, performances are compared directly against each other via correlation in order to search directly for similarities between performances. The following application instead compares synthetic functions to features extracted from the performances. In this case, the synthetic function is an arch shape. It roughly matches the phrasing structure in tempo-curves, but any shape of interest could be used. For example, a ramp function can be used in phrase analysis to study the acceleration and deceleration properties of phrases.[10] Hybrid applications which compare similarity of phrasing between performances could be done by correlating performances first to phrases, and then comparing the extracted phrase information between performances.

Figure 1.19 illustrates a use of the scape-plotting domain which functions very similar to wavelet transforms. The figure shows two different performances of Chopin's mazurka in B minor, op. 30, no. 2. The tempo-curve for each performance is shown at the bottom of the figure and demonstrates a significant difference in performance interpretation. The

[10] Arch and ramp correlation plots can be generated online at
http://sv.mazurka.org.uk/software/online/scape

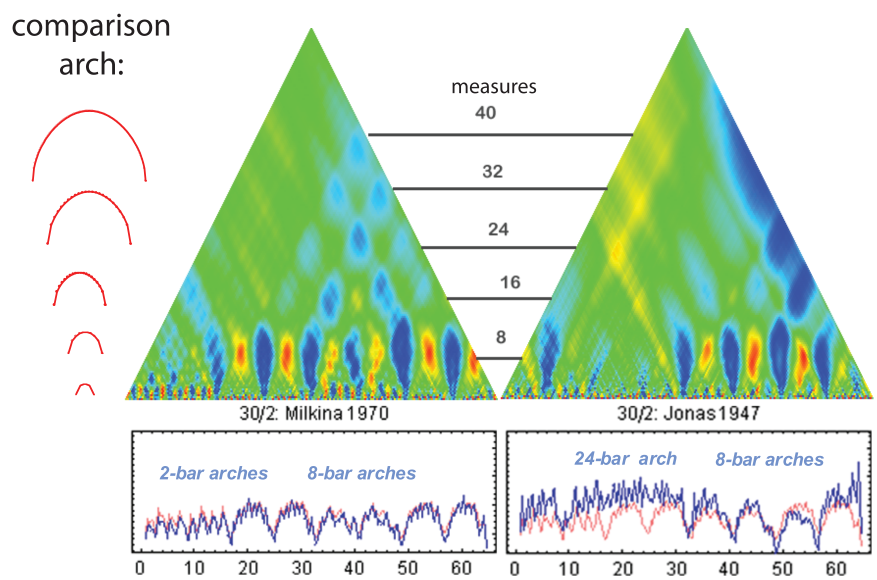


Figure 1.19: Arch correlation plots for two tempo-curves extracted from two different performances of Chopin's mazurka in B minor, op. 30/2.

tempo plot for Milkina shows a performance where the tempo-curve very closely matches the average tempo taken from a large number of performances (shown in red in her tempo-curve plot), while the Jonas tempo-curve on the right shows a large deviation from the average performance (also shown in red).

In order to identify the arch shape of phrases in the tempo-curves, the tempo data from the plots are correlated against prototype arches at all time scales in a manner identical to the multi-timescale properties of wavelet analysis. The arch is stretched to match the length of the portion of the tempo-curve with which it is being correlated. The scape plot shows how well the tempo-curve matches small arches at the bottom of the plot and large arches at the top of the plot. The resulting correlation values for each position and timescale in the data are plotted in the triangular plots above the tempo-curves, with red representing high correlation values between the tempo-curve and arch-shape, while blue represents low correlation (inverted arches) and green indicates no correlation (flat tempo-curve).

In the Milkina scape plot, the tempo-curve fits best to two-bar arches in the first 16 measures, and the rest of the music fits best with 8-bar arches. Measures 32–48 show arches at both the 2-bar and 8-bar levels. Weak groupings of the phrases into 16-bars phrases can be seen by the stronger extension of the blue regions to higher levels in measures 16, 32, and 48 as compared to measures 8, 24, 40 and 56.

The Jonas plot shows very different tempo-phrasing organization for the music. Instead of grouping the music by 2 measures in the beginning, a weak 8-bar arch is the most prominent. Then a 24-bar arch with no significant arching at smaller time-scales occurs in measures 9–24, followed by a more conventional 8-bar phrasing for the rest of the composition. The 24-bar arch can also be seen to encompass neighboring phrases to generate a slightly weaker arch structure at the 32- and 42-bar levels.

An interesting similarity between the two plots can be seen at the 40-bar level, where both performances have a positive arch in the first half of the music and a negative arch in the second half of the performance. The Jonas arch pattern at this time scale is more apparent in the underlying tempo plot, with a shape similar to one cycle of a sine wave visible in the tempo-curve.

Application of this plotting method for phrase identification can also be applied to other musical features, such as dynamics. In [4], example plots for both tempo phrasing and dynamic phrasing are compared against each other to study the timing offset (or lack of difference) between tempo and dynamics arches within musical performances.

References

[1] Albert S. Bregman. *Auditory Scene Analysis: The Perceptual Organization of Sound*. MIT Press, 1990.

[2] Elaine Chew. *Towards a Mathematical Model of Tonality*. PhD thesis, MIT, 2000.

[3] Nicholas Cook and C. Sapp. Une pure coïncidence?: Joyce Hatto et les Mazurkas de Chopin. *L'étincelle*, Nov. 2007. IRCAM: Paris, France. http://www.charm.rhul.ac.uk/projects/p2_3_2.html.

[4] Nicholas Cook *et al.* *The Cambridge Companion to Recorded Music*. Cambridge University Press, 2009.

[5] Patricio de la Cuadra, A. Master, and C. Sapp. Efficient pitch detection techniques for interactive music. *Proceedings of the International Computer Music Conference (ICMC)*, pages 403–406, 2001.

[6] Simon Dixon. Onset detection revisited. *Proceedings of the 9th International Conference on Digital Audio Effects (DAFx'06)*, 18–20 Sept. 2006.

[7] Carol Krumhansl. *Cognitive Foundations of Musical Pitch, Oxford Psychology Series No. 17*, pages 81–96. Oxford University Press, 1990.

[8] Mark Singer. Fantasia for piano: Joyce Hatto's incredible career. *New Yorker*, pages 66–81, 17 Sep. 2007.

Chapter 2

Harmonic Visualizations of Tonal Music

ABSTRACT

Multi-timescale visualization techniques for displaying the output from key-finding algorithms are presented in this chapter. The horizontal axis of the key graphs represents time in the score, while the vertical axis represents the duration of an analysis window used to select music for the key-finding algorithm. Each analysis window result is shaded according to the output key's tonic pitch. The resulting diagrams can be used to compare differences between key-finding algorithms at different time scales and to view the harmonic structure and relationships between key regions in a musical composition.

2.1 Motivation

A tonal composition is usually described as being in a particular key, such as Brahms Symphony no. 4 in E minor, or Beethoven's Piano Sonata in F minor, op. 57; however, rarely does a piece of music maintain a single key center throughout its entirety. The *key* of a piece typically starts and ends the piece, but other key centers are used somewhere in the middle of a piece to give form to the music. A simple stereotypical tonal piece might start in the tonic key, modulate to the dominant, and then return to the tonic key by the end of the piece.

Currently available key-finding algorithms are not very sensitive to identifying modulations, and if given a selection of music which contains several modulations, the algorithms can only identify what the most likely key is for the entire selection. If there are two key areas in a selection of music, then the algorithm hopefully assigns the stronger key the best score while the other key area hopefully is assigned the second best score. A key identification error may occur if an entire piece of music is presented to a key-finding algorithm, since the secondary key areas may overpower the starting/ending key or bias the primary key towards a closely related key, such as the dominant or relative minor key areas.

One of the best algorithms for determining the key in a region of music is the Krumhansl-Schmuckler (KS) key-finding algorithm based on probe-tone ratings generated from experimental results. [3] The KS algorithm is implemented in the `key` program contained in the Humdrum Toolkit for musical analysis. [2] Applying the KS algorithm to an entire

			KS algorithm analyses	
BWV		*notated key*	*prelude*	*fugue*
846	1.	C major	C major	C major
847	2.	C minor	C minor	C minor
848	3.	C♯ major	C♯ major	G♯ major
849	4.	C♯ minor	C♯ minor	C♯ minor
850	5.	D major	D major	D major
851	6.	D minor	D minor	D minor
852	7.	E♭ major	E♭ major	E♭ major
853	8.	E♭/D♯ min.	E♭ minor	D♯ minor
854	9.	E major	E major	E major
855	10.	E minor	E minor	E minor
856	11.	F major	D minor	F major
857	12.	F minor	F minor	F minor

Table 2.1: KS algorithm results when applied to entire J.S. Bach Well-Tempered Clavier, Book I compositions compared to actual tonic keys of the music. Identification errors are underlined.

piece, Table 2.1 lists the analyzed keys for the first half of the first book of J.S. Bach's Well-Tempered Clavier using the `key` program. The Well-Tempered Clavier is an excellent source of test material for testing key-finding algorithms, because each set of prelude and fugue in the collection are in a different key, starting in C major and then progressing chromatically through all 24 major and minor keys.

Table 2.1 points out two common errors generated by key-finding algorithms in general. The first error is in fugue no. 3 where the KS algorithm identifies the dominant rather than the correct tonic key of C♯ major. The second error occurs in the eleventh prelude where the relative minor is identified rather than the key of F major.

These two errors are primarily due to more than one key-area being present in the analyzed music, causing slight offsets in the tonic key weightings such that a closely related key becomes more likely for the algorithm than the actual key. A fifth-relation error can occur if the secondary key areas are predominantly all above or below the tonic key in the circle of fifths. The second error is a modality error. The correct key signature was identified, but the distribution of notes in the music was such that the tonic was incorrectly identified.

Of course, the key identification errors in the two compositions from the Well-Tempered Clavier could be fixed by only applying the algorithm to the first and last parts of the music, since these sections are more likely to contain the tonic key. However, side-stepping the issue in this manner creates other problems: (1) How much of the beginning and ending of the piece should be examined? and (2) What if the composition starts in one key and ends in another? The true problem to solve is how to identify correctly regions of stable key centers and regions of modulation in a piece of music.

If a more detailed view of a composition's key structure is desired beyond the key of the piece, then a moment-by-moment view of the key centers in the piece will give a much more detailed description of the piece. Krumhansl proposes applying a sliding analysis window to the notes in a piece to generate localized key measurements. This gives a good overview of the key relationships in the music, but results can be sensitive to the analysis window size. A problem with this sliding window technique is that the global importance of a local key is not apparent in the local context. A music theorists would assign a sequence of key centers to a piece of music using both the global and local characteristics of the music. If a region is difficult to assign a key label, then music outside the local region will be considered. It is very difficult to have a computer generate a reasonably accurate sequence of musically relevant key centers.

Key visualization techniques described in the following section avoid the problem of choosing a fixed analysis window duration by instead using all possible analysis window durations. The interpretation of the local key problem is partially solved with these display methods because the height to which a key region survives in a diagram demonstrates the relative strength of that key region. Strong modulations are represented by large vertical structures, while tonicizations are represented by smaller vertical structures.

It is also possible to view the behavior between various key-finding algorithms at different analysis window sizes and to see how they interact around regions of modulation. Key-algorithms can then be compared using the diagram methods below to see how the different algorithms handle the same music at various time scales. Numerous computational algorithms for identifying keys using computers have been proposed since the 1960s. See work and surveys on key-finding algorithms by Temperley (1997) [7], Chew (2000) [1] and Shmulevich and Yli-Harja (2000) [5]. Sleator and Temperley make the source code for their harmonic analysis programs available on the web [6].

diatonic pitch class	*color*	*root*	(R, G, B)	*root*	(R, G, B)
		C♭	(36, 255, 0)	G♭	(54, 200, 218)
		C	(0, 255, 0)	G	(63, 191, 255)
E	red	C♯	(9, 246, 36)	G♯	(63, 177, 255)
B	orange	D♭	(63, 109, 255)	A♭	(118, 41, 255)
F	yellow	D	(63, 95, 255)	A	(127, 31, 255)
C	green	D♯	(73, 86, 255)	A♯	(145, 27, 219)
G	blue	E♭	(237, 4, 36)	B♭	(255, 109, 0)
D	indigo	E	(255, 0, 0)	B	(255, 127, 0)
A	violet	E♯	(255, 18, 0)	B♯	(255, 145, 0)
		F♭	(255, 237, 0)	F♯	(218, 255, 0)
		F	(255, 255, 0)		

Table 2.2: Sample RGB color mappings for key tonics.

2.2 Diagram types

2.2.1 Key-to-color mappings

To display data from key-finding algorithms in a compact visual manner, each key is mapped to a different color. The principle key-to-color mapping being used is shown in Table 2.2. The colors of the rainbow are mapped onto the circle of fifths collapsed to the seven diatonic pitches. For example, the key C is assigned the color green. Ascending the circle of fifths yields G (blue), D (indigo blue), A (purple), E (red). Going the opposite direction in the circle of fifths takes you through the rainbow colors in the opposite direction: F (yellow), B♭ (orange), E♭ (red).

This color mapping is designed to be intuitively easy to navigate, since closely related keys map into closely related colors. The only drawback of this mapping is that there are only 7 colors while there are 12 (or more) pitch classes—notice that E and E♭ have the same red color. Since enharmonic keys are located far apart in key space, this mapping is unambiguous for most tonal music and should only cause confusion in rare cases. A monochromatic color mapping has been used in this chapter for printing in black-and-white, but it is possible to view example figures from this chapter in the *Tonal Landscape Gallery* [4] or dynamically generated from musical data on the kernscores website.[1]

[1]http://kern.ccarh.org, http://kern.humdrum.org

Of course, the choice of a color mapping from key to color is arbitrary, and different color mappings may be suitable for different types of music. For tonal music, the relationship between keys is predominantly based on the circle of fifths; therefore, arranging the circle of fifths onto the rainbow is advantageous. Many other types of color mappings are possible, including:

- indicating major/minor modalities with brightness. For example, C major could be light green and C minor could be dark green.
- indicating sharp keys brighter, and flat keys darker to unambiguous enharmonic keys, for example C♭ could be dark green and C♯ could be light green.
- using hue instead of the rainbow to maximize distance between keys in color space.
- using a monochromatic mapping for optimal display in black and white (as well as for color-blind people).
- displaying music in the key colors of its synesthetic composer.

It is also possible to label the tonic of a composition in a certain color, and then label all other key areas in relation to the tonic. But for this system to work, you would need to guarantee that the correct key has been chosen as the tonic key.

2.2.2 Type 1: discrete time/discrete roots

The first type of diagram for key display divides a piece of music into successively smaller analysis window units as outlined in Figure 2.1. The top level of the diagram analyzes all of the notes in the selection to generate a key identification. The second level of the diagram then splits the music into two equal parts and generates a key for the music in each half, and so on. Each successively lower level of the diagram divides the music into a greater number of equal-sized analysis windows which get continually smaller in duration.

The lowest level will then divide the musical selection into N parts. The duration of the analysis windows in this level are total-time/N time units wide. Typically N is set to the number of beats in the music, but any smaller or larger value of N can be used as well.

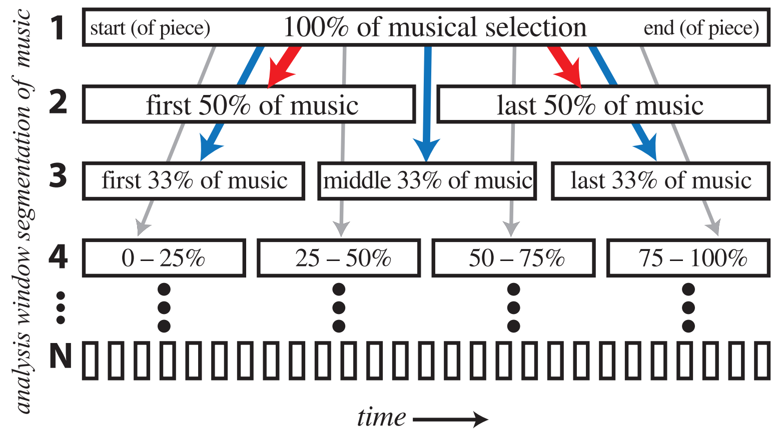


Figure 2.1: Analysis window configuration which generates type-1 plots.

When using a value of N equal to the value of beats in the music, the bottom level of the diagram displays chord roots, and the top levels represent the strong key areas present in the piece.

If the N levels of the vertical axis are displayed with equal heights, the lower levels overpower the higher levels as demonstrated on the left side of Figure 2.2. For example, the ratio between the first and second levels at the top of the diagram is a ratio of 1/2, or a 50% decrease in window duration. The ratio between the 100th and 101st levels is 100/101 or a 1% decrease in duration. The higher the number of divisions, the less change there is between analysis window sizes, and therefore less importance should be given to the visual impact between levels.

To correct for the emphasis on larger division levels, the vertical axis can be adjusted

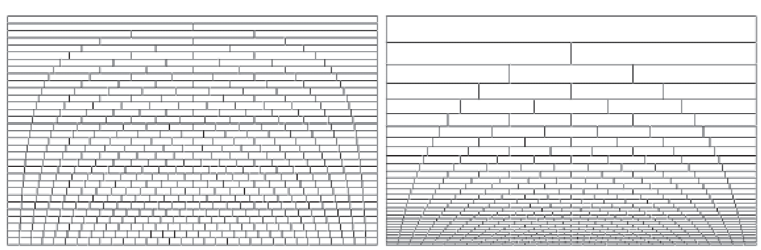


Figure 2.2: Type-1 plot analysis window layout. Left: linear vertical scale; Right: logarithmic vertical scale.

with a logarithmic scaling. A strong logarithmic scaling weights the upper levels of the diagrams too much, so good scalings which equally emphasize the top level and bottom levels of the diagrams is generated by this example scaling:

$$m = M\left(1 - \frac{\log(n+1)}{\log(N+1)}\right) \tag{2.1}$$

Where N is the total number of analysis window levels, n is the current level indexed from 0, M is the number of vertical pixels in the picture, and m is y-axis maximum pixel location of level n with respect to the bottom of the picture. The right side of Figure 2.2 displays this type of mapping where the areas of the window blocks are closer in proportion to the change in window duration between levels.

Figure 2.3 displays a type-1 key diagram for the first movement of W.A. Mozart's Divertimento no. 4, K. 439b (1783; transposed to C major). The horizontal axis displays the music from the start (left) to the ending (right) of the piece. The vertical axis displays the analysis results for small window durations at the bottom of the picture and larger analysis window durations at the top of the picture. Pitch labels identify important key/chord regions in the piece. Notice that the key of C forms the dominant shaded area of the picture which coincides with the identification of this piece being in the key of C major.

The bottom portion of Figure 2.3 represents individual chords. As the analysis windows get larger towards the top of the picture, the chords merge with adjacent chords. Some chords get absorbed quickly into adjacent chords, particularly if they are remotely related to the surrounding chords. By the middle of the vertical axis, the strongest chords start to represent key areas. Some key areas have a short duration and are absorbed into stronger keys areas higher up in the diagram. The top portion of the figure shows the major key regions of the piece which are primarily the keys of C major and G major.

The divertimento movement is clearly in C major, since the upper left and right portions of the diagram are in C. The figure also shows a strong region of G major in the second quarter of the piece. There is a distinction between the F modulation near the middle of the piece and the temporary modulation (tonicization) of F closer to the end of the piece. Tonicizations do not extend towards the top of the piece as high as true modulations into a key area.

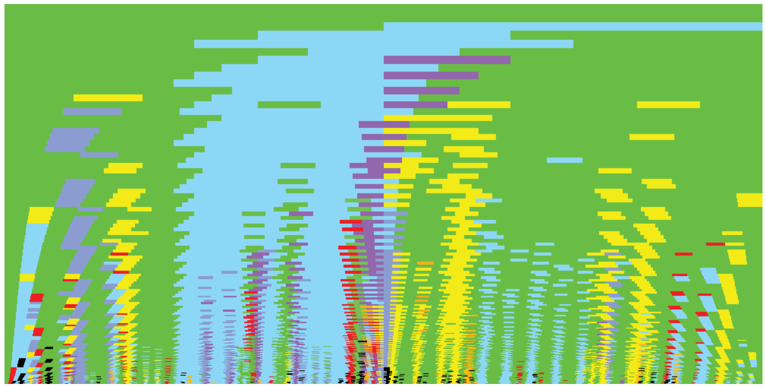


Figure 2.3: Mozart divertimento K. 439b, mvmt. 1: type-1 plot with logarithmic vertical scale.

The striped bands for the key of A in the middle of the movement demonstrate weakness in the particular key-finding algorithm being used. The A key region is probably a compromise in the algorithm between an overlapping region of D minor and F major. Another problem is that the F key area contains a region of C inside. This C region is due to dominant seventh chords in the key of F, so the C regions should be expected to be smaller and incorporate into F faster than the plot shows.

2.2.3 Type 2: continuous time/discrete roots

The primary drawback of type-1 key diagrams is that quantization errors increase in the higher analysis window levels where there are fewer divisions to represent the entire piece. Therefore, a second type of diagram is presented in this section which gives equal resolution at all time scales. Instead of coloring the entire analysis window duration with the key color, a single pixel centered in the middle of the analysis window is drawn. Note that computation time for type-2 plots is about 30 to 50 times greater than for type-1 plots.

Figure 2.4 gives an overview of the windowing method used to create type-2 key diagrams. The top level of the diagram contains an analysis window that can hold the entire piece of music. For that window, a single pixel is displayed at the top of the diagram. To generate the lower parts of the diagram, the analysis windows continually get smaller. For each window duration, the analysis window is slid continuously over the entire piece of

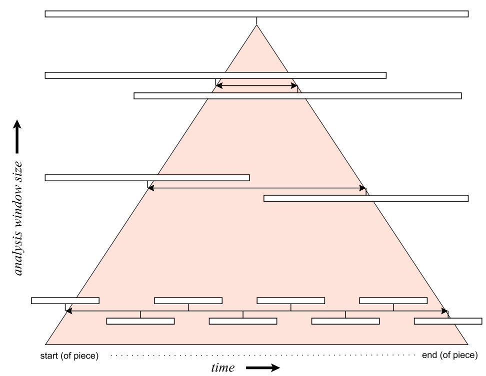


Figure 2.4: Type-2 analysis window arrangement.

music, placing the result of the key-finding algorithm at the center point of the analysis window. Typically, the analysis windows of the lowest level in the diagram contains one beat of the music. The placement of the analysis output at the center of the window generates a characteristic triangular shape which makes the diagram easy to distinguish from type-1 plots.

Increased resolution is a big advantage of type-2 plots over type-1 plots. Figure 2.5 now shows the Mozart divertimento movement with a very nicely formed region in the key of G. Also, the development regions near the center of the movement are easier to distinguish, since they are less fragmented.

Just as there was a problem with the vertical scale in type-1 plots which show too much of the lower time-scales, Figure 2.5 displays too much of the higher time-scales. Note that the second level of a type-1 plot is now found midway on the vertical axis in a type-2 plot. The unaltered form of the type-2 plots is equivalent to a strong logarithmic scaling of the type-1 plots. Therefore, a more balanced view of the top and bottom levels can be achieved by scaling the vertical axes logarithmically, as shown in Figure 2.6.

2.2.4 Type 3: continuous time and roots

Most key-finding algorithms calculate the likelihood of every possible key and then assign the key with the highest score as the best key to fit the music. For example, Table 2.3 lists

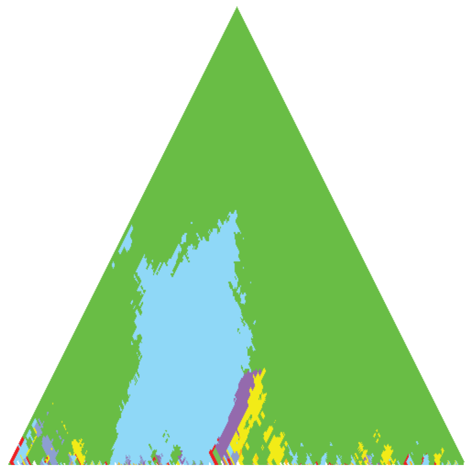


Figure 2.5: Divertimento type-2 plot with linear vertical scale (comp. to Figure 2.3).

the KS algorithm scores of each possible key for the entire Mozart divertimento movement. The third category of key diagram described in this chapter takes these secondary key probabilities into account to generate color-interpolated pictures of key based on the general form of type-2 plots.

r-value	*key*	*r-value*	*key*	*r-value*	*key*	*r-value*	*key*
0.945	C major	0.403	G minor	-0.068	E♭ major	-0.420	G♯ minor
0.770	G major	0.163	D minor	-0.158	E major	-0.436	F♯ minor
0.665	E minor	0.110	D major	-0.160	A major	-0.450	B♭ minor
0.481	C minor	0.063	F minor	-0.260	G♯ major	-0.557	C♯ major
0.479	A minor	0.027	B♭ major	-0.321	C♯ minor	-0.646	D♯ minor
0.417	F major	0.019	B minor	-0.415	B major	-0.650	F♯ major

Table 2.3: KS algorithm *r*-value scores of each possible key for the Mozart divertimento, sorted from most likely (highest *r*-value) to least likely key.

Figure 2.7 plots algorithmic score weightings of the four primary key centers taken from the midsection of Figure 2.6. In this figure the relative strengths of each key score at any given moment in the music is plotted. Type-1 and -2 key diagrams will only display the most likely key at any given moment, which in this case is the sequence: C, G, A, F, and then a return to C. What is lacking in these visualizations is an indication of how certain the algorithm is in its choice of key.

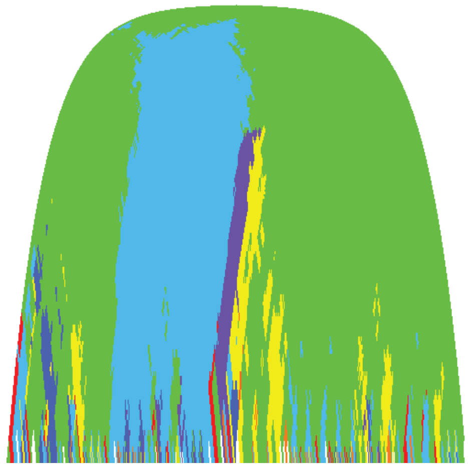


Figure 2.6: Mozart divertimento: type-2 plot with logarithmic vertical scale (compare with Figure 2.5.) The vertical scale is approximately the same as in Figure 2.3.

Two specific measurements can be extracted from information shown in Figure 2.7 to give a more continuous key diagram. (1) Clarity: The higher the score, the better the hypothesis key fits the music. Regions of musical stability are likely places to find a clear key center. For example, the recapitulation into C major in the first movement of the divertimento coincides with the highest score for C major. (2) Ambiguity: regions of development such as in the middle of the divertimento consist of closely scored keys. The algorithm has difficulty in choosing the best key in this case, because modulations happen so quickly compared to the more stable exposition and recapitulation. Key identification errors are more likely in regions of high ambiguity.

Experiments in displaying just the clarity or ambiguity yield promising diagrams which might prove useful in music analysis; however, key scores are difficult to interpret and relate to each other over different time scales. Figure 2.8 displays a plot of clarity between the best and second best keys. Notice that the short development region in the middle of the movement is clearly indicated by the dark band rising vertically in the center of the plot. Plots of ambiguity look similar to the regular best-key plots because they usually outline the borders between key areas.

Interpolating the colors of the best and second best keys gives a nice continuous diagram. Figure 2.9 shows a plot of the second best keys for the divertimento. Notice that

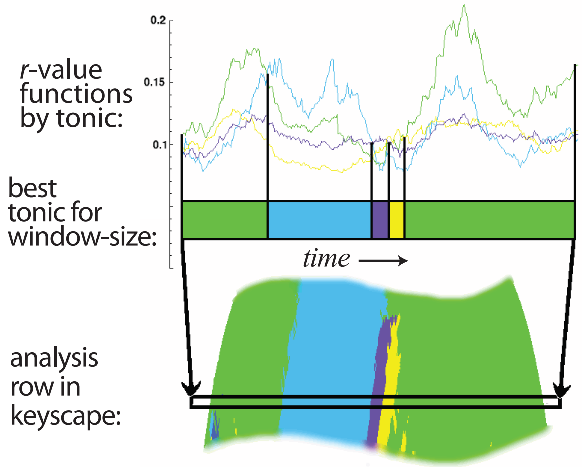


Figure 2.7: Continuous nature of some key-finding algorithms key scores. Bottom part of figure is a midrange zoom-in on Figure 2.6. Top part of figure shows the continuous key-finding scores for important key regions highlighted in the lower figure.

in this piece, the second best key is usually tonic in a dominant section, and the dominant in a tonic section, although there is a strong tendency towards the subdominant during the beginning of the piece and close to the final cadence.

Figure 2.10 is an interpolation between the best and second best keys at each point in Figure 2.6 and Figure 2.9. For each analysis window size, the maximum clarity between the best and second best score is used to normalize the interpolation: at the point of maximum score separation, the best key is displayed fully in its own color. At points during the piece where the best and second best keys trade places (at 100% ambiguity), the color assigned to the diagram is halfway between the colors of the two key centers. Notice that the higher levels of secondary key regions (G and F) in Figure 2.10 blend smoothly into the primary key of C major.

The interpolation for Figure 2.10 was done linearly in the RGB color space. Interpolation in another color space, such as HSI (Hue-Saturation-Intensity) may be possible, but producing visually pleasant diagrams with this type of interpolation has been difficult to control. An HSI interpolation could be used to distinguish between modulations by fifths from other types of more distant key modulations. The more distant the modulation, the higher would be the saturation of the colors in the modulation region.

Figure 2.8: Plot of clarity regions indicate higher certainty of correctness by the key-finding algorithm and darker regions indicate little difference in the scores between best and second best keys.

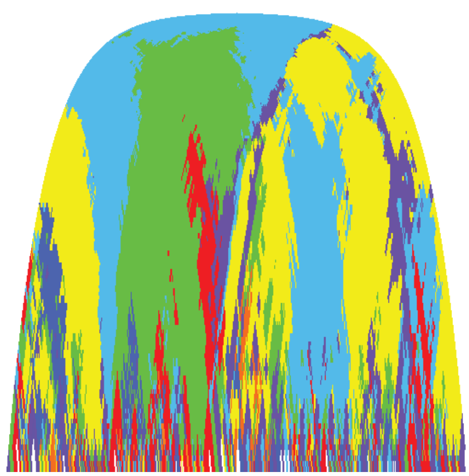


Figure 2.9: Mozart divertimento: second-best keys. Compare to Figure 2.6 which shows the best roots.

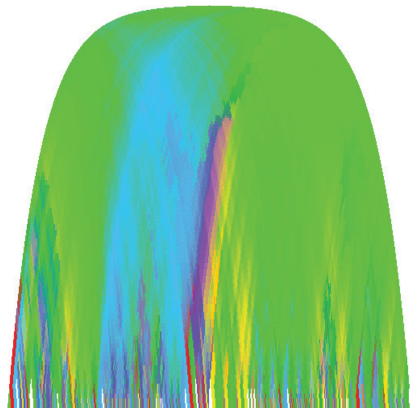


Figure 2.10: Mozart divertimento: type-3 plot interpolating between best root (Figure 2.6) and second best root (Figure 2.9).

2.3 Applications

2.3.1 Comparing key-finding algorithms

The key diagrams described in the previous section were developed for two purposes: (1) examine the interaction between key regions in a piece of music, and (2) compare the behavior of different key-finding algorithms. Most of the plots in this chapter use an algorithm being developed for finding the roots of chords. This algorithm matches the output of the KS algorithm for the most part, although the calculations to derive the analysis are completely different.

Figure 2.11 shows a type-2 plot of the Mozart divertimento movement which can be compared directly to Figure 2.6. Good features: (1) definition of C major at start of piece is clearer than in Figure 2.6, (2) dominant key areas in the piece are C and G, (3) incorrect A key region in middle of the piece is less pronounced, (4) more solid F key region in middle of the piece, (5) nicely behaved boundary between G and C key areas. Not-so-good features: (1) E minor key tendencies near end of the movement (modal error from G major), and (2) incorrect identification of key regions just before F major area.

A slightly stronger emphasis on the dominant key can also be seen in Figure 2.11 as compared to Figure 2.6. The large G major structure proceeds higher up in the diagram than it does in Figure 2.6 and the fragmented G region around the 2/3 point in Figure 2.11

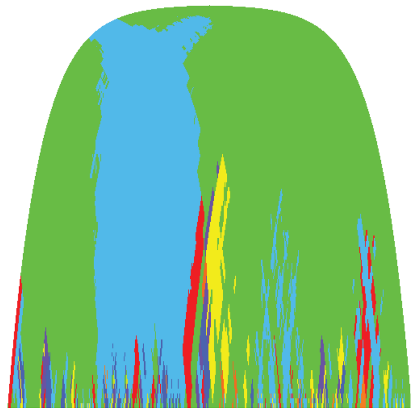


Figure 2.11: Mozart divertimento: using the KS algorithm. (compare to Figure 2.6).

is much less pronounced in Figure 2.6. Thus, it can be inferred that the KS algorithm will be more prone to identifying the dominant key of the correct key than the algorithm used to generate Figure 2.6.

Figure 2.11 can also be used to illustrate the difficulty of extracting a good sequence of modulations in a composition as mentioned in the Motivation section. The Mozart divertimento is a very simple and clear tonal piece. The movement contains a sequence of 5 key regions: (C, G, D, F, C), with the following segmentation boundaries:

key area	*bars*	*length*
C major	1 – 15	15 bars
G major	16 – 28	13 bars
D minor	29 – 32	4 bars
F major	33 – 38	6 bars
C major	39 – 63	25 bars

The D minor section is very brief and could be considered part of the F major region. D minor serves as a transition between the keys of G and F by switch roles from a dominant key to a relative minor key. Now notice that this sequence of keys is not present on any horizontal line in Figure 2.11. Therefore, no fixed analysis window size in the KS algorithm will give the correct sequence according to my human-based harmonic analysis. Figure 2.6 fairs a little better with a wide region in the middle of the diagram containing the sequence (C, G, A, F, C), and the algorithm admits that it is having problems identifying a key in the D minor region (see Figure 2.8).

Further improvements to key-finding algorithms can be accomplished by using the visualization techniques presented in this chapter as an evaluation tool. With key diagrams, the strengths and weakness in a particular algorithm are much easier to detect, since a large number of analyses are needed to make a single picture. In particular, improvements to the handling of modulation areas should yield better boundaries between key-regions.

2.4 Music analysis

The key diagrams presented in this chapter have numerous potential applications in harmonic analysis. In music theory training, the visualization maps of the harmony can assist students in understanding musical structures which are difficult to explain. For example, the diagrams could help in understanding the difference between a modulation and a tonicization. Also, the plots can be an objective tool to explain the conflicting interpretations of music theorists.

Another type of music analysis application may be the identification of harmonic form, and style. Highly tonal music such as the Mozart divertimento, used as an example in this chapter, form clearly defined high-level key structures in the diagrams. Baroque and Romantic era music generally contain more elaborate key relations and therefore have a tendency to contain more detail in the higher levels of the diagrams as compared to Classical music.

The diagrams for J.S. Bach’s prelude and fugue in C minor from WTC 1, displays an interesting contrast. The preludes diagram contains smooth key center boundaries at the higher levels, while the fugues diagram contains key centers with much more fragmented boundaries. This is to be expected, since Bach preludes usually consist of one melody and accompaniment, while fugues are more focused on the linear aspects of melody rather than on harmony.

As a contrast to tonal music, Figures 2.12 and 2.13 display samples of non-tonal music. The plot structure for tonal music at larger time-scales will tend to simplify, but non-tonal music and extended tonality will tend to have complicated boundaries higher up in the plot. This is due to both high ambiguity and low clarity: when the music is not in any particular

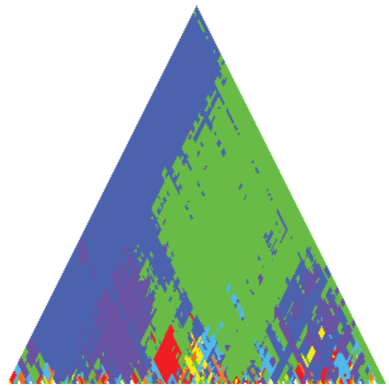


Figure 2.12: Anton Webern, op. 27, mvmt. 1. Twelve-tone music (compare to Figure 2.5 and Figure 2.13.).

key, the score between the best and second-best answer from a key-finding algorithm will be small (high ambiguity), and the absolute value of the scores will be much lower than for tonal music (low clarity).

The Webern composition shown in Figure 2.12 uses a post-tonal technique where all 12 pitch classes are used in sequence before being repeated. The twelve-tone technique is used intentionally to destroy any sense of key throughout the composition. The key boundaries remain complex throughout the height of the plot, with two unrelated tonal centers of C and D oscillating as the best answer. This indicates that the ambiguity between these two key regions is higher, with not much difference between the numeric scores for both keys.

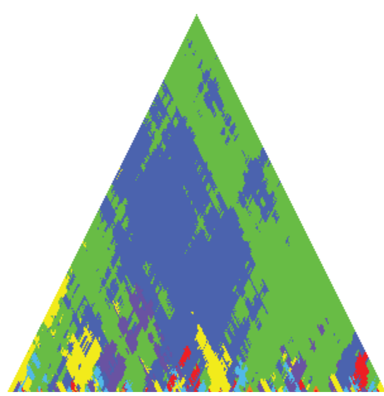


Figure 2.13: 13th-century motet attributed to Petrus de Cruce (compare to Figure 2.5 and Figure 2.12.

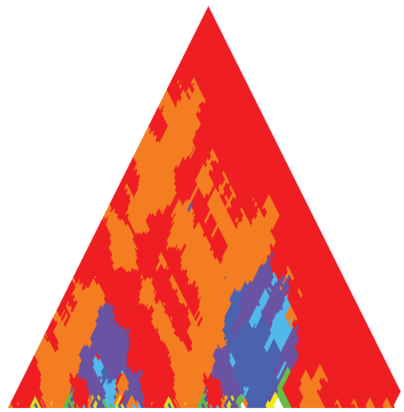


Figure 2.14: Alexander Scriabin. Prelude in E minor, op. 11, no. 4.

Figure 2.12 only shows the best keys, when plotting a type-3 diagram of the twelve-tone piece, the keys areas become even less pronounced than in this type-2 plot.

Figure 2.13 shows a piece of music from the other side of western-music history. This figure displays a type-2 key plot for a Medieval motet from the 13th century—several hundred years before the development of functional harmony in the early 17th century. Even though the piece uses a diatonic pitch set, it is remarkably similar in character to the Webern example. Key boundaries are fragmented and follow diagonal paths more than vertical paths. Also similar to the Webern example, the top level key centers switch between C and D, which are not closely related keys.

As a final historical contrast, consider Figure 2.14 which diagrams the key profiles in a prelude written by the late Romantic composer Alexander Scriabin, which is an example of extended functional harmony. The key boundaries at the higher levels are fragmented, but the primary key is clearly E minor. The top-level key centers in this diagram are closely related (tonic and dominant). The lower level key centers of A and D are arranged in a vertical manner similar to the key regions in the Mozart divertimento.

2.5 Summary

It is more interesting to know the interaction between key areas in a piece of music than just knowing the key-label for an entire composition or selection of music. With the harmonic maps generated from these visualizations of key-finding algorithms, the interactions and relations between keys become easier to examine.

The plots presented in this chapter generate fascinating visualizations of harmony. Aside from looking nice, they can be used to compare the behavior of different key-finding algorithms. Also, they are a good starting point from which to develop more robust algorithms that can accurately detect key modulations in a musical sense rather than in a computational sense.

Many variations on the basic harmonic visualizations presented in this chapter are yet to be explored. For example, all of the plots presented in this chapter use analysis windows which weight musical pitches in the analysis window equally. Varying the importance of a pitch inside the analysis windows by using a triangular or exponential window might yield interesting results.

References

[1] Elaine Chew. *Towards a Mathematical Model of Tonality*. PhD thesis, MIT, 2000.

[2] David Huron. *The Humdrum Toolkit: Software for Music Research*. http://csml.som.ohio-state.edu/Humdrum.

[3] Carol Krumhansl. *Cognitive Foundations of Musical Pitch*, Oxford Psychology Series No. 17, pages 81–96. Oxford University Press, 1990.

[4] Craig Sapp. *Tonal Landscape Gallery*. http://ccrma.stanford.edu/˜craig/keyscape.

[5] Ilya Shmulevich and Olli Yli-Harja. Localized key-finding: Algorithms and applications. *Music Perception*, 17(4):531–544, 2000.

[6] Daniel Sleator and David Temperley. *The Melisma Music Analyzer*. http://www.link.cs.cmu.edu/music-analysis.

[7] David Temperley. An algorithm for harmonic analysis. *Music Perception*, 15(1):31–68, 1997.

Chapter 3

Visual Hierarchical Key Analysis

ABSTRACT

Tonal music is often conceived of as progressing through a sequence of key regions, usually starting and ending in the tonic key, with a journey away from the tonic key somewhere in the middle of the piece. This chapter presents a visual method of displaying the musical key structure of a composition in a single picture. The hierarchical plots can also show the relative strength of these key regions and how they develop out of the chordal substrate of the music.

3.1 Computational key identification

Consider the melody shown in Figure 3.1:

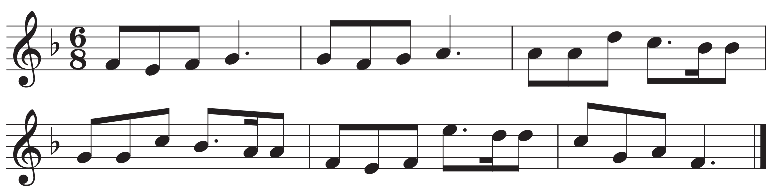


Figure 3.1: A short song (Liebes-A-B-C by August Pohlenz, 1827).

In what musical key is this melody? For a trained musician, it is easy to decide that the music is in the key of F major. This is due to the key signature containing one flat, the first note being an F, and the last note being an F as well. However, these are only superficial features used to determine the key of the music by eye. Suppose that someone listens to this melody without access to the score? Even in this case, musicians should easily be able to tell that the music is in a major key and that the first and last notes are the tonic of the key.

Now suppose that, in an attempt to hide the fact that the music is in F major, the music is altered by changing the key signature as well as the first and last notes, as demonstrated in Figure 3.2:

Has the key changed? At a cursory glance, it might appear that the music is in G minor because the last note is G, the longest note in the first measure is G, and the key signature matches that of G minor. Only the first and the last few notes have been changed,

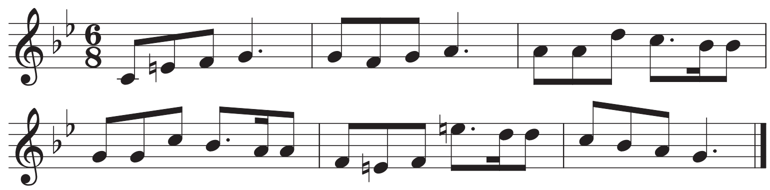


Figure 3.2: The same melody as in Figure 3.1, but with a slightly altered beginning and ending.

while the key signature has just been changed to suggest G minor is the key of the music. Nonetheless, the key of the song has not changed. Listening to the last note of the song, a non-musician might perceive the end of the music as unresolved, and the last note might seem unstable, since it is expected to go down one more step. For a trained musician, the last note should sound like the second degree of a major scale, which is an unusual note on which to end a song.

So what is a more robust method of determining the key of a song? One method could be to look at interval patterns that are more likely to occur in one key rather than another. For example, the pattern F-E-F-G that starts the original version of the song can only occur in the two major keys of C and F. Another way of looking at this note sequence is by intervals: the sequence first goes down one half-step, then up one half-step and then up a whole-step. This can only occur in two places in a major key: on the scale-degree pattern 1-7-1-2 or the pattern 4-3-4-5. Taken in isolation, the scale-degree pattern 1-7-1-2 is much more common in music, so a musician should quickly suspect that the first measure of the original version of the song is in the key of F major (with the scale-degree pattern 1-7-1-2) rather than in C major (with the scale-degree pattern 4-3-4-5). This is a good method for determining the perception of a musical key in a melody; however, it is difficult to apply to more complicated music that contains harmonic accompaniment along with the melody, since the music might be more difficult to break down into linear sequences of interval patterns.

Another key-finding method, which works well for polyphonic music, was developed by Carol Krumhansl and Mark Schmuckler in the 1980s. [2, ch. 4] In their key-finding algorithm, the relative amount of pitches in the music is compared to a prototypical pattern of pitches expected in major and minor keys. For example, there obviously should be more pitches in a song that are scale-degree notes of the key rather than non-scale notes. If a piece of music is in C major, one would expect lots of pitches in the scale of C major (C, D, E, F, G, A, and B) and not expect many notes outside of the key: (C♯, D♯, F♯, G♯, and A♯).

With knowledge about the number of notes in each pitch-class in a song, a key can be determined by correlation with a prototypical pattern for a major (or minor) key. Correlation is a mathematical means of determining how similar two patterns are to each other. Figure 3.3 shows a sequential pattern which will be used to find example correlations:

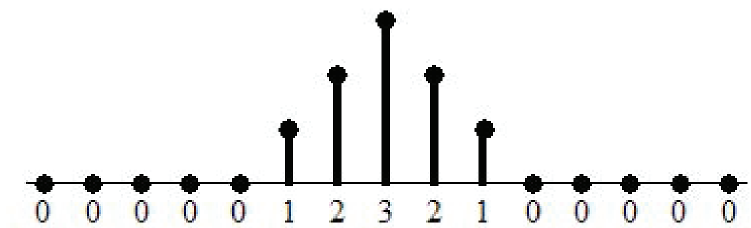


Figure 3.3: A sample pattern sequence to be matched.

How similar is the pattern in Figure 3.3 to the following pattern in Figure 3.4?

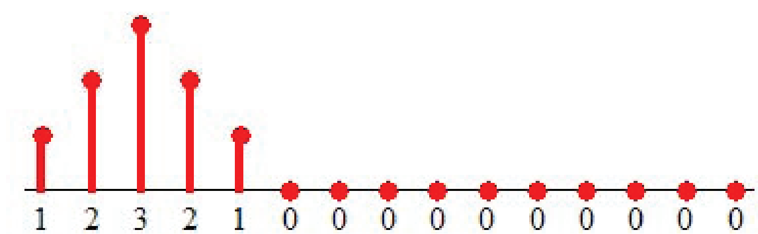


Figure 3.4: A sequence to be compared with Figure 3.3

To determine the correlation between these two patterns, multiply each corresponding element in each sequence, and then add the resulting values of the sequence together. Example calculations for the correlation between Figure 3.3 and Figure 3.4 are summarized below:

	0	0	0	0	0	1	2	3	2	1	0	0	0	0	0
×	1	2	3	2	1	0	0	0	0	0	0	0	0	0	0

0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 **= 0**

Since the calculations yield a value of zero, there is no correlation between these two patterns. This might seem strange, since to the eye, they do seem similar as both have the same basic shape. However, the correlation value does not consider shifting the patterns, and is only looking point-by-point between the two patterns when comparing them. Now suppose that the second pattern is shifted slightly to the right so that there is a little bit of overlap between the patterns, in this case the correlation is a little larger:

```
  0   0   0   0   0   1   2   3   2   1   0   0   0   0   0
× 0   1   2   3   2   1   0   0   0   0   0   0   0   0   0
  ---------------------------------------------------------
  0 + 0 + 0 + 0 + 0 + 1 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0  = 1
```

In this case there is slightly more correlation (similarity) between the two patterns, but not very much. At what point is the correlation between the original pattern and the shifted pattern the greatest? Obviously when the peak of the shifted pattern is aligned with the first pattern:

```
  0   0   0   0   0   1   2   3   2   1   0   0   0   0   0
× 0   0   0   0   0   1   2   3   2   1   0   0   0   0   0
  ---------------------------------------------------------
  0 + 0 + 0 + 0 + 0 + 1 + 4 + 9 + 4 + 1 + 0 + 0 + 0 + 0 + 0  = 19
```

For our purposes in determining the musical key of a song, it is useful to plot all of the correlation values for each shift of the matching pattern. For example, Figure 3.5 displays a plot of the 11 correlation values generated by shifting the second pattern (Figure 3.4) gradually from the left against the original pattern (Figure 3.3).

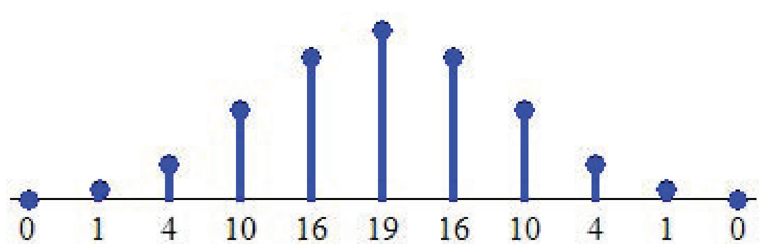


Figure 3.5: Correlation values for shifted version of Figure 3.4 compared to Figure 3.3.

How do correlation measurements apply when finding the key of a song? First, count all of the pitch classes (pitch names without octave information) to generate a pitch-class histogram. For example, when counting by sixteenth-note durations, there are 7 duration units of C in the original song from Figure 3.1; 0 duration units of C♯; 5 duration units of D in the song, etc., as shown in the histogram in Figure 3.6.

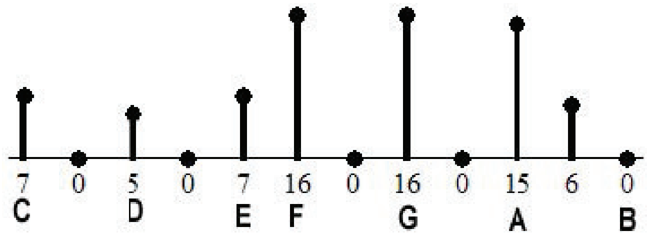


Figure 3.6: Pitch-duration histogram for the music from Figure 3.1 in 16th-note durations.

This histogram of the pitch-classes in the original song (weighted by their duration) can now be correlated with the patterns typical for those in a major key. What are those values? There are many ways to derive a distribution of pitches for a major key, but the basic Krumhansl-Schmuckler (KS) key-finding algorithm in particular uses probe-tone weights from perceptual experiments as the major scale pattern, shown in Figure 3.7.

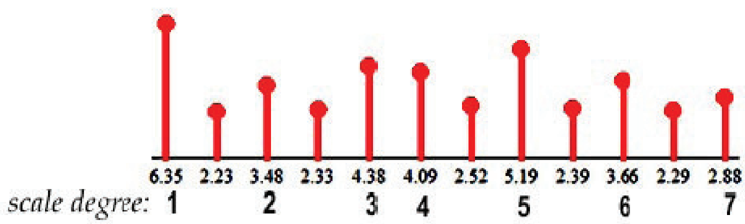


Figure 3.7: Krumhansl's probe-tone weightings for major key contexts.

Note that the scale-degree positions have higher weighting values than do the non-scale-degree values. If the same correlation calculations are now done as in Figure 3.5 (but wrapping values that extend to the left back onto the right side of the sequence), the correlation values for each shifted version of the major pattern will be generated as shown in Figure 3.8.

Since the highest correlation value (322) is generated when the song's pitch histogram and the major-key profile occurs when the tonic note is on F, the musical key that best fits the music is F major.

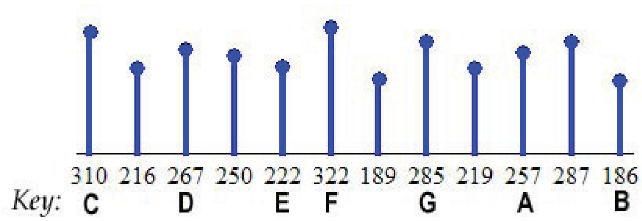


Figure 3.8: Correlation values for rotated versions of an expected major scale pitch distribution (Figure 3.7) when compared to the histogram of pitch-classes from a real sample of music (Figure 3.6).

The previous discussion demonstrates the essence of key-finding by correlation. The mathematical description of the correlation calculations being done for the above example key-finding calculations is

$$\text{key}_k = \arg\max_k \sum_n e_{n,k} d_n \tag{3.1}$$

where e is the major (or minor) key profile (such as the one in Figure 3.7); d is the histogram of pitches in the actual music (such as that in Figure 3.6); n is an index of the pitch classes; k is the shifting of the scale profile onto the various pitch classes; and N is 12, the number of pitch classes. The full practical implementation of the Krumhansl-Schmuckler algorithm normalizes the correlation values into the range between +1.0, and -1.0, where +1.0 means maximally correlated, and -1.0 maximally uncorrelated, using the normalizations in the following normalized correlation equation:

$$\text{key}_k = \arg\max_k \frac{\sum_n (e_{n,k} - \mu_e)(d_n - \mu_d)}{\sqrt{\sum_n (e_{n,k} - \mu_e)^2 \sum_n (d_n - \mu_d)^2}} \tag{3.2}$$

where μ_e is the arithmetic mean (average) of the e measured pitch profile, and d is the arithmetic mean of the d histogram. In the field of statistics the values calculated with this normalization are called r-values. The r-value is a correlation coefficient that indicates the strength of a relationship between data sets (pitch histograms and key profiles in our case). An r-value of +1.0 means that the data sets are very strongly related; an r-value of 0 means that there is no relation between data sets. The r-value normalization of the correlation

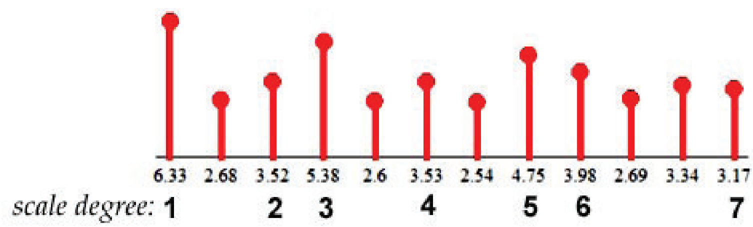


Figure 3.9: Krumhansl's probe-tone weightings for minor-key contexts.

allows comparisons between different key patterns, such as the probe-tone profile for minor keys in the Krumhansl-Schmuckler key-finding algorithm (Figure 3.9).

3.2 Musical key modulation

Correlation is a good technique for finding the key in a selection of music that entirely or predominantly remains in the same key, but when more that one key is present in the music, the meaning of the correlation between a pitch histogram and a prototypical key profile is less reliable and can be difficult to interpret. The correlation technique enumerates the possible keys in the musical selection, but it cannot by itself identify cases when there are supposed to be two or more keys present in a musical sample.

Take, for example, the sixth variation from Franz Schubert's 13 Variations on a Theme by Anselm Hüttenbrenner in A major, D. 576 (1817) which starts out in the key of F♯ minor and ends in the key of A major (Figure 3.10). Even if you are not a trained musician, you should still be able to hear the character of the music change half-way through the variation as the music transitions from a sadder, melancholy feeling into a happier, optimistic feeling half-way through the variation.

If a pitch histogram of the entire variation is given as input to the Krumhansl-Schmuckler key-finding algorithm, it will determine that the best key for the music is in A major, with an r-value of 0.86. F♯ minor (r=0.78) is the second-best choice, followed by C♯ minor (r=0.62), and E major (r=0.57). The key-finding algorithm identifies A major as the best key for the music, but it cannot indicate the internal key structure of the music except by giving F♯ minor, which has equal importance to A major in the variation, as the second best key to fit the entire variation.

Figure 3.10: Franz Schubert's 13 Variations on a Theme by Anselm Hüttenbrenner in A minor, D. 576 (1817), variation no. 6.

Now suppose that the music of the variation is split into two halves, so that key of each half can be examined separately via the Krumhansl-Schmuckler key-finding algorithm. In this case the algorithm will find the correct key assignments for each half. The first half will be identified as F♯ minor (r=0.80); the second half will be identified as A major (r=0.90).

If the music is further split into a sequence of four four-measure segments, the KS key-finding algorithm will identify the key sequence: C♯ minor (r=0.67), F♯ minor (r=0.87), E major (r=0.77), A major (r=0.84). This key analysis does not make precise musical sense, but it does describe the basic movement from dominant to tonic key regions in each section (C♯ to F♯) and (E to A). In this case, not enough information is present in the pitch histograms to give the same answer for the first and second halves of each key region, so the Krumhansl-Schmuckler algorithm is looking at a slightly lower level of music resolution than the actual key, and it displays the overall chord structure from dominant to tonic for each key.

In the first case, where all of the music for the variation was used to estimate the key, only one key in the music was identified because the Krumhansl-Schmuckler analysis window was too large and included two key regions. The algorithm chose A major, perhaps only because there are more note-heads in the second half of the piece, or because of the algorithms tendency to emphasize mediant key areas more than submediant key areas when using the standard probe-tone weightings. In the second case, the algorithm correctly identified the two keys in the music when the music was cut into two equal parts, exactly proscribing the two key areas in the variation. In the third case, the duration of music input into the Krumhansl-Schmuckler algorithm calculations was too small, and a sub-key description of the music was given as output.

Note that if too much music is analyzed at once, fewer important keys are suppressed; if too little music is analyzed at once, the chordal structure of the music is really being analyzed instead of the key structure. So what is the proper duration of music to give as input to the Krumhansl-Schmuckler key-finding algorithm?

3.3 Key analysis plots

This author's approach is to examine all possible segmentations of the music. To accomplish this, a two-dimensional plot was devised to display the analysis results from thousands of key identifications generated by the Krumhansl-Schmuckler key-finding algorithm, or any other computational key-finding algorithm.

Figure 3.11 shows the layout of the plot domain used to display the key analyses from a computational key-finding algorithm. The horizontal axis represents time in the music, from the start of the piece at the far left side through the end of the piece on the far right side. The vertical axis represents the duration of music given to the key-finding algorithm. For example, the top of the plot represents the entire duration of the composition, while the bottom of the plot represents some minimum time duration, such as one beat in the music. The duration of music represented on the vertical axis is also called the analysis windows size.

To generate data for plot, an analysis window is selected on the vertical axis and centered at a time in the composition represented on the horizontal axis. For example, the point in Figure 3.11 labeled first half of composition, represents the key analysis of the first half of the composition. The key analysis for that window is displayed in the plot in the middle of the analysis windows duration in the music. For the case of the Schubert variation, the first half of the music is found to be in F♯ minor, so that point in the plot will be labeled F♯ minor. Likewise, the point marked second half of composition will be labeled as being in

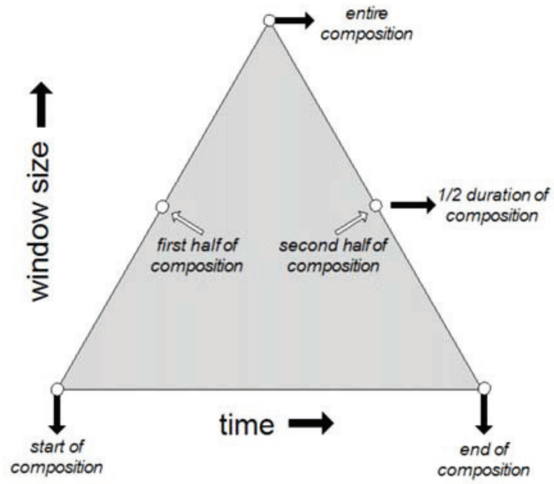


Figure 3.11: The significance of the horizontal and vertical axes in a key-analysis plot with interpretations of a few points in the plot.

A major. To get the intervening points, an analysis window equal to 1/2 of the composition gradually slides through the music from start to finish; an analysis result is labeled in the plot at the midpoint time location of the window in the music on the horizontal axis.

Every point in the plotting region represents a key analysis done with two parameters: (1) the duration of the analysis window into the music; and (2) the center-point in time of the analysis window. The plotting region is in the shape of a triangle because each analysis-window size represented on the vertical axis is centered at a specific time in the composition. For example, the top of the plot can only have one point, since there is only one way of centering an analysis window of the entire piece (at the half-way point in the composition).

Since every point in the plotting domain shown in Figure 3.11 is a separate key analysis, it would be difficult to display analysis results in textual form. Instead, each key analysis result is assigned a color. What color to assign each of the 24 major/minor keys is an arbitrary decision. For the following plots, the coloring convention given in Figure 3.12 is used, where the diatonic circle of fifths is mapped onto the colors of the rainbow. This allows harmonic relations of a fifth, which are the most important relations in tonal harmony, to be represented by closely related colors. Chromatic alterations of the diatonic

Figure 3.12: Key-to-color mappings used in this paper. Diatonic pitches are arranged in a sequence of fifths on the colors of the rainbow. Chromatic alterations of diatonic pitches generate slight color shifts, and modality is indicated by brightness.

pitches are given as slight variations in the color of the natural pitch classes, as indicated by the accidentals below the diatonic E colors in Figure 3.12. Major and minor keys can be distinguished by brightness as also illustrated in Figure 3.12.

Using the coloring scheme in Figure 3.12, key analyses of the music in Figure 3.10 are displayed in the plot shown in Figure 3.13. Note that the two black dots in Figure 3.13 indicate the first and second halves of the variation. The Krumhansl-Schmuckler algorithm correctly identifies the two keys of the variation at these points. The first half of the music is in F♯ minor, which in the plot is colored in a dark greenish-yellow color. The second half of the music is in A major, colored in purple. Also notice that the colors of the plot in the immediate vicinity of these two points also agree with the key-analyses at the two given points, which shows there is some tolerance for the presence of non-key materials in the analysis window in the key-finding algorithm. Other colors in the plot are due to temporary key regions and/or chord roots, since the bottom of the plots represent small-scale key features in the music.

Figure 3.14 shows the same plot as Figure 3.13 and illustrates the input data used for the key-finding algorithm at various points in the plot. These types of plots are given the nickname keyscapes, since they are analogous to landscape paintings. The bottom of the plots represent small-scale key features such as chords that are similar to the foreground in a landscape painting. The top of each plot represents the large-scale key of the composition, which is similar to the background in a landscape painting. The various vertical positions in the plots are also related to the concept of foreground, middleground, and background in Schenkerian analysis, which is a form of tonal music analysis developed in the early 20th century. Musical keyscape plots, to some extent, serve as an objective form of Schenkerian

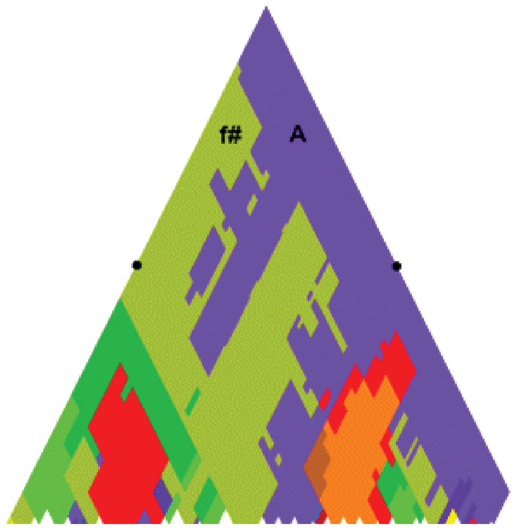


Figure 3.13: Linear keyscape plot of the music shown in Figure 3.10. The key regions of F♯ minor and A major are labeled with their tonic letters. The black dots in the plot indicate the analysis window for the first half of the music (left side) which is F♯ minor, and the last half of the music (right side) which is in A major.

analysis. [4]

The keyscape plots were originally conceived for comparing the properties of various key-finding algorithms at various time resolutions. For example, Aarden [1] has measured a different set of major/minor key correlation profiles from musical scores, which can be used in place of the Krumhansl probe-tone profiles in the correlation algorithm. Figure 3.15 shows Aarden's major and minor profiles used to find the key by correlation calculations. Using the major/minor profiles from Figure 3.15, a keyscape plot for the Aarden profiles is shown in Figure 3.16. In this plot the F♯ minor half of the music is identified as stronger than the A major half, since the top of the plot is colored for F♯ minor. The two primary key regions do have the same correct analyses of F♯ minor and A major in both plots at the appropriately marked points, but in the presence of modulation between keys, both differ in their analyses due to their different correlation weights.

Included with this chapter is an animation graphic that interpolates the key profile weights between those of Krumhansl and Aarden. The regions of the plot that remain in a single color (and thus in a single key) can be interpreted as being more stable. Regions

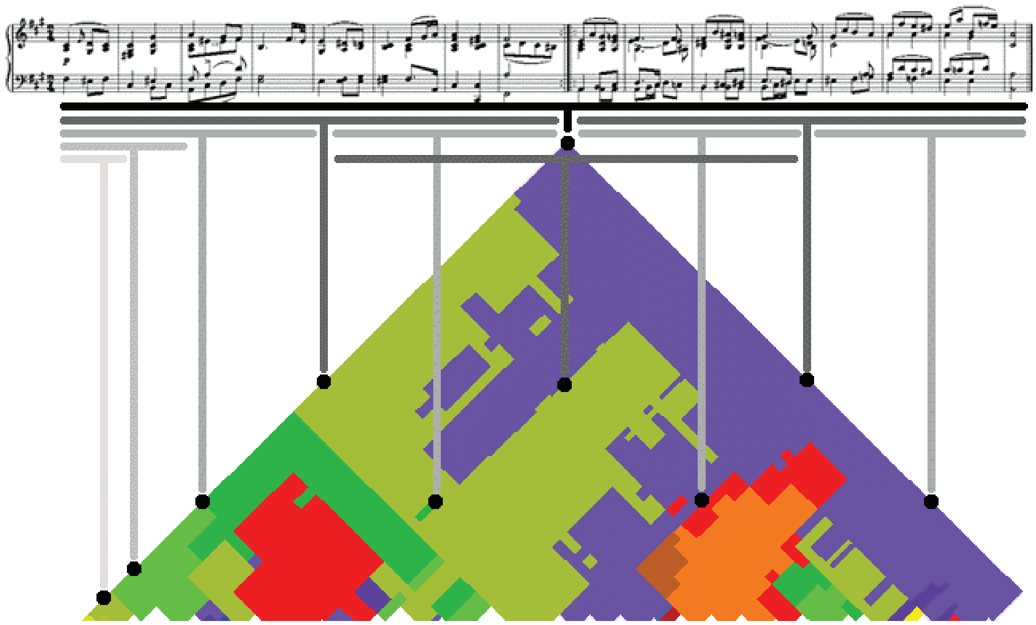


Figure 3.14: Keyscape plot of the music from Figure 3.13 and the relationship of various points in the plot to music from Figure 3.10.

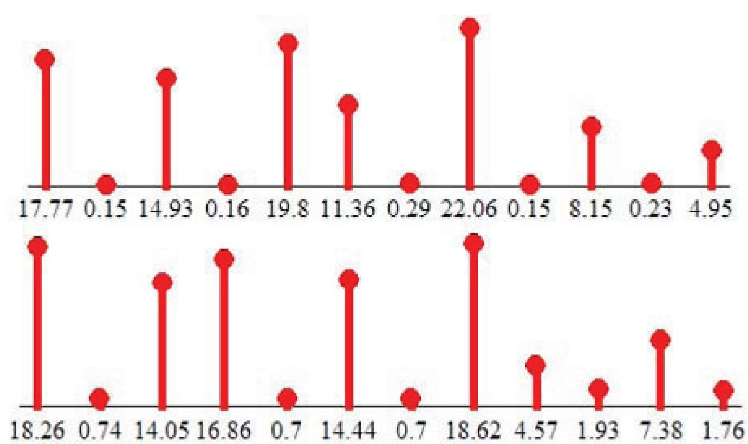


Figure 3.15: Aarden’s correlation profiles for major (top) and minor (bottom) keys.

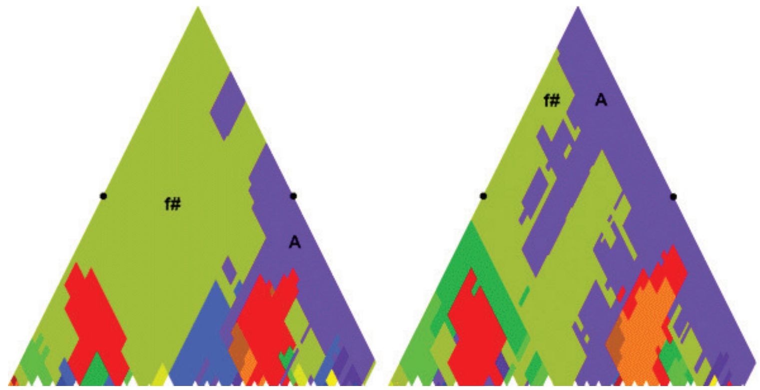


Figure 3.16: Plot generated from Aarden weightings (left) compared to Krumhansl weightings (right) from Figure 3.13. Black dots in each plot indicate the two key regions of F♯ minor and A major in the music

of the plot that shift colors during the morphing between the two profiles are less certain to be given the correct key assignment by either set of weights, and could even indicate the presence of a modulation boundary between adjacent key regions.

Any key-finding algorithm can be input into the plot, not just those based on key-profile correlations. For example, Figure 3.17 plots the output from a root-finding algorithm that is independent of key-profiles. This algorithm also gives the correct key answers for the first and last halves of the variation. The Krumhansl-Schmuckler key-finding algorithm is reasonably accurate at finding the key despite the amazing fact that all notes in the analysis windows are jumbled together into a single histogram, just as if all the notes were played at once in a single large chord. Another key-finding algorithm that is sensitive to the ordering of the notes and chords was developed by Temperley [7], and may be input into the keyscape plotting format in the future.

Other types of descriptive information from music can be displayed in the keyscape plots. For example, ambiguity and clarity measurements derived from computational key-analysis algorithms can be used. [5] And the basic plotting domain from the keyscape pictures has been used to analyze musical features other than the keys. [6]

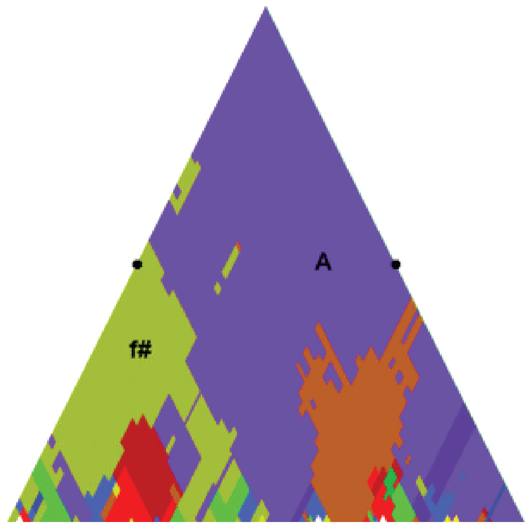


Figure 3.17: Plot of the same music as in Figure 3.16, but using an unrelated root-identification algorithm.

3.4 Tonal structure in music

Most pieces of western classical music do not remain in one key throughout the entire composition. The typical format is to start in the tonic, go off into other keys, and then return to the initial key at the end of the piece. Older musical styles (say before 1800) are typically more conservative in the range of keys covered in a single composition, while later styles of classical music tend to get as far away from the initial key as possible (particularly in the music of the late 19th and early 20th centuries). With the keyscape plots, it is possible to get an overview of the harmonic structure of a piece of music and a rough estimate of the musical style in which the music was composed.

Figure 3.18 shows the musical example used in this section: Divertimento no. 4, K 439b, mvmt. 1 composed by W.A. Mozart. The form of this piece is an abbreviated sonata form. The first half of the music contains the first theme in C major (the tonic key) and the second theme in G major (the dominant key). The second half of the music starts with a development section that is intentionally ambiguous in terms of key. The development section starts with a phrase that is not really in any particular key but can be assigned to D minor for convenience. This is followed by two phrases in F major. The recapitulation section

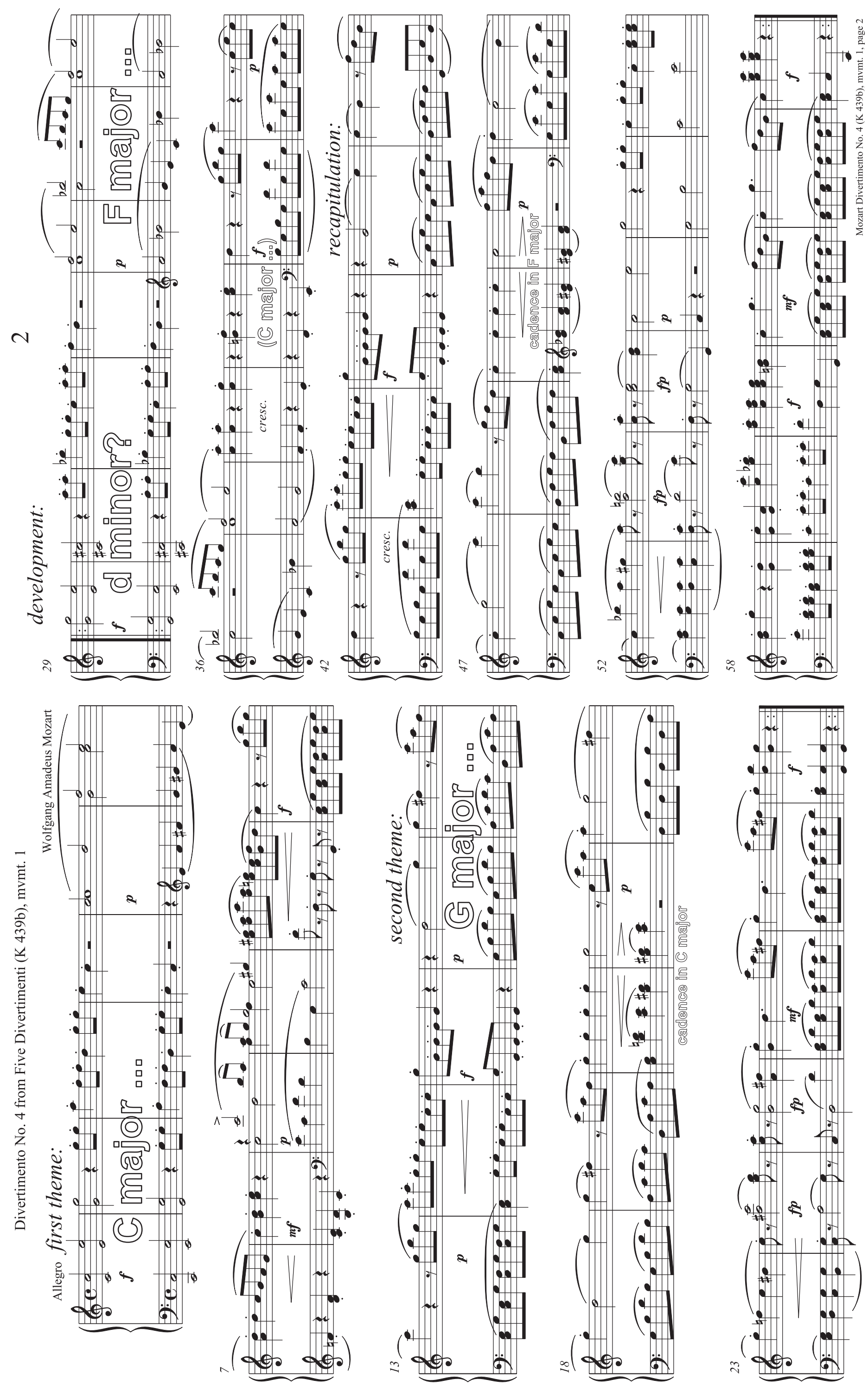


Figure 3.18: W.A. Mozart's Divertimento no. 4, K 49b (1783), mvmt. 1.

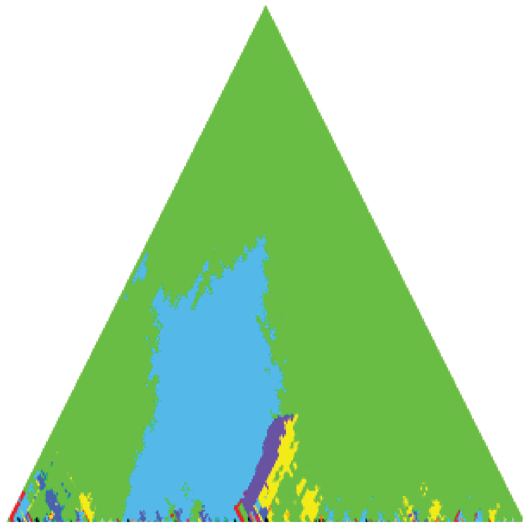


Figure 3.19: Linear keyscape plot of the music found in Figure 3.18.

follows the development and repeats the second theme from the first half of the music; however the theme has been transposed from the dominant key into the tonic, as is traditionally done in the sonata form.

The music can best be described as going through the key sequence C-G-(d)-F-C. Summarizing the relative importance of the key regions is usually not done carefully in traditional music theory, but one simple way of describing the importance of each key is to indicate the number of measures that each key region lasts: C[15]-G[13]-d[4]-F[7]-C[24]. This gives a rough overview of the relative importance of each key region; however, this is similar to giving a table of numbers rather than a plot of numbers. The keyscape plots can be used to show the relative importance of key regions in a composition. Figure 3.19 displays a keyscape plot of the Mozart divertimento. Notice that the predominant color in the plot is bright green, which represents C major, the tonic key in the divertimento. The green section of the plot goes from the beginning of the music on the far left through to the end of the music on the far right, which indicates that the music starts and ends in C major. The very top of the plot is also colored green, which indicates that the key-finding algorithm determined that the entire composition as a whole was in C major.

The second most important key in the music is G major which is the key of the second theme in the introduction. This key is represented by light blue in the plot. Light blue is also the second most prominent color in the plot, agreeing with the music analysis of composition. The keys of the development are less important. This is shown by the violet and yellow sections on the lower right side of the light blue region in the plot. These color regions are smaller due to the key regions in the development section being shorter and less important than the key regions in the introduction. Notice that the development key regions do not go as high in the plot as the G major (light blue) region, since they are not as prominent in the music. Likewise, G major is not as prominent as C major represented by the green region, which reaches all the way to the top of the plot. The various colored regions at the bottom of the plot represent individual chord roots that merge into the weaker and then stronger key regions in the music. Thus the keyscape plots display the harmonic structure of the music in a hierarchical manner that is similar to Tree-Notation reductions of music found in A Generative Theory of Tonal Music. [3]

Up to this point, the keyscape plots have been presented in a triangular format, but it is sometimes useful to display them with a rounded top, as found in Figure 3.20. Each row in a triangular form of the plot represents a window size that increases/decreases at a constant arithmetic rate. For example, if there are one thousand rows in a plot and the plot represents a piece of music with one thousand beats, the analysis window size for each row going upwards in the plot gets one beat longer on each row. Notice that the relative size of the window grows at a slower rate for larger windows. For example, the top row is about 0.1% larger than the row underneath it (1000 beats compared to 999 beats); however, the bottom row is 50% smaller than the row above it (1 beat compared to 2 beats). The plot in Figure 3.20 is rounded on the top because the size of each row increases at the same rate in a geometric progression rather than in an arithmetic progression. Therefore, the vertical scale of the plot in Figure 3.20 is logarithmic with respect to the size of the analysis window.

Since all levels of the music are weighted equally on a logarithmic scale in Figure 3.20, it gives a more perceptually accurate view of the harmonic structure of the music. The triangle pictures are useful for viewing the large-scale key structure of a piece, or for short pieces such as the Schubert variation where the small-scale structure is also visible due to the short duration of the music. Figure 3.21 shows the Krumhansl and Aarden weighted plots of the divertimento movement. Looking at these plots, the Krumhansl weights can

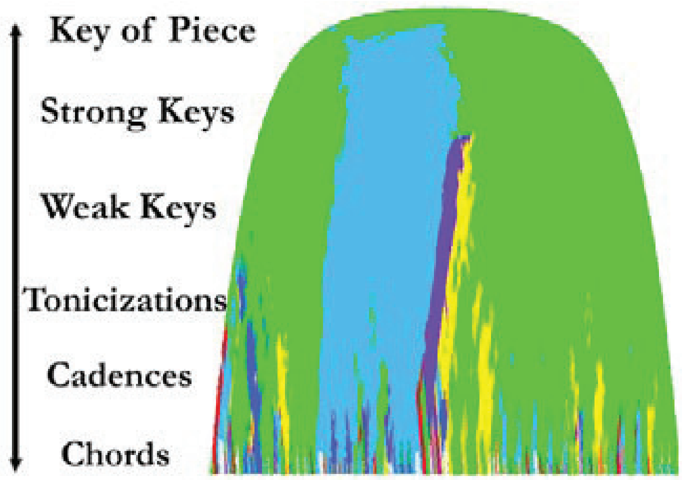


Figure 3.20: Mozart divertimento with a logarithmic scaling of the analysis window size on the vertical axis.

be seen to over-emphasize the dominant key (G major; light blue) at the expense of the subdominant key (F major; yellow). The Aarden weightings come closer to the music-theory expectations for the movement, since the sub-dominant key area in the development (yellow regions) is more prominent; however, the dominant region suffers a bit since it is only about as large as the briefer sub-dominant key region.

3.5 Examples from various styles of music

3.5.1 Analysis of the Prelude from J.S. Bach's Cello Suite, BWV 1007

Figure 3.22 shows the keyscape analysis of the prelude from J.S. Bach's cello suite, BWV 1007, which was composed around 1720. Both keyscape plots agree that the music starts and ends in the key of G major (light blue) with a significant region of the dominant (medium blue) in the middle of the piece. The Aarden weightings over-emphasize the sub-mediant chord (red), which is probably not that noticeable to human listeners.

The Aarden-weight plot in Figure 3.22 has a nice feature: at the top of the plot there is a small region of light blue which represents the key of G major. This means that when the entire composition is given to the algorithm, it identifies the key as G major, even though

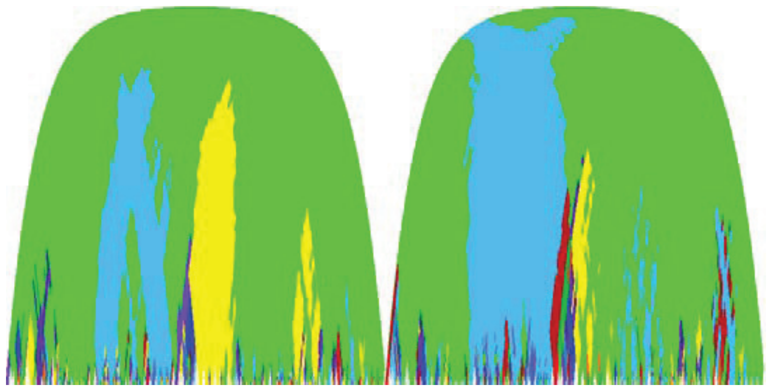


Figure 3.21: Mozart divertimento with a logarithmic vertical scale comparing Aarden weights on the left and Krumhansl weights on the right.

Figure 3.22: Keyscape plots from the prelude from J.S. Bach's cello suite, BWV 1007. Aarden weights used on the left and Krumhansl weights on the right.

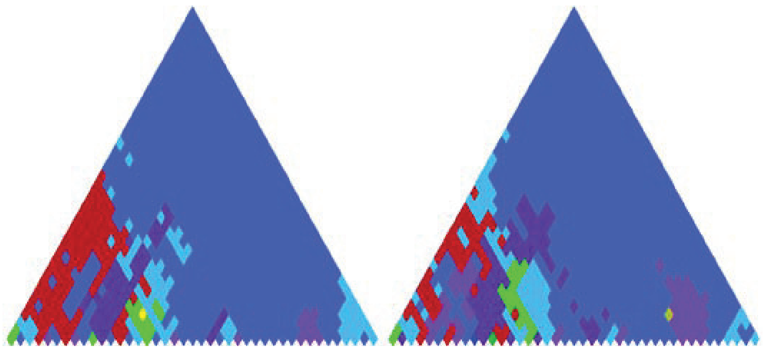


Figure 3.23: Keyscape plots for a chordal reduction of the prelude also used in Figure 3.22. Aarden weights used on the left and Krumhansl weights on the right.

more dominant (medium blue) chords are present in the music. On the other hand, the Krumhansl weightings will identify D major as the key of the piece when given the notes of the entire movement, most likely due to an over-emphasis of the dominant key region.

The prelude has a fair number of non-harmonic tones, and since it is for a solo instrument, the music is sparse. Figure 3.23 shows comparable keyscape plots, with the music reduced to the basic chord sequence from each measure, which removes non-harmonic notes from the analysis. Both plots remain similar to the plots for the full version of the music, although the Krumhansl-weight plot does give more emphasis to the submediant (red).

3.5.2 Analysis of Johann Pachelbel's Canon in D major

By examining the plots in Figure 3.24 we can see that the music to Pachelbel's Canon is indeed in the key of D major, the key represented by the medium-blue color. According to the plots, there are absolutely no secondary key regions in the piece, since the key of D major arises directly from the underlying chords which are continually repeated.

In this case it is more interesting to display a plot of the music for every two bars (Figure 3.25), since the canon is a round that cycles every two bars. Each repetition has the same harmonic structure based on the baseline D-A-B-F♯-G-D-G-A. There is some variation in the harmonic structure, which is made evident by the greenish-yellow regions

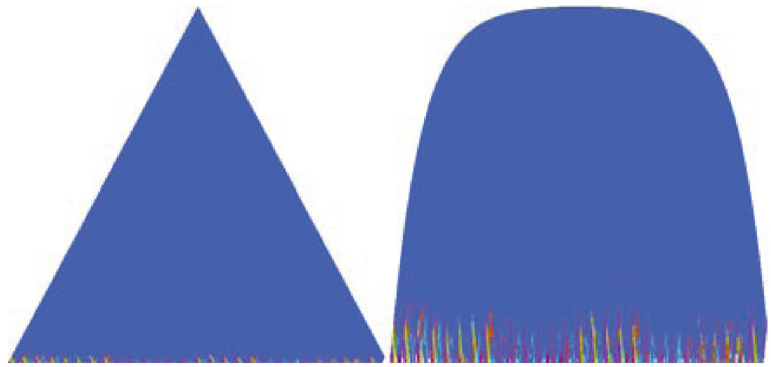


Figure 3.24: Keyscape plots for Pachelbel's Canon in D Major in both linear (left) and logarithmic (right) vertical scaling, using Krumhansl weightings.

representing a chord of F♯ minor in the plots of Figure 3.25. Sometimes, the greenish-yellow regions are not present in a segment, such as the 9th through 14th repetitions on the second row of the figure. In these cases the chord with a root on F♯ is changed into a chord with a root on D.

3.5.3 Analysis of Samuel Barber's Adagio for Strings

Figure 3.26 shows the keyscape plots for Samuel Barber's Adagio for Strings, written in 1936. The composition is in B♭ minor, which is indicated in the plots by the orange color extending to the top of the plots. The climax of the piece is visible as a red rectangle at the bottom of the plot near the end of the piece. This rectangle represents the key of F♭ major, although red is usually the color for the E pitch classes (E and F♭ are the same key on a piano). Interestingly, F♭ is the most distant key from B♭, since the interval between them is a tritone, but F♭ is played closely after the clearest presentation of a B♭ minor chord in the piece (measures 50–52).

The harmonic climax coincides with a large registral shift in the music. The lines below each plot in Figure 3.26 are plots of the notes present in the score. The horizontal axis represents time, just as in the keyscape plots. The vertical axis represents pitch, from low to high, and color is used to indicate the different instrumental parts. Notice that starting around the middle of the composition, the parts all start to rise gradually in pitch until they

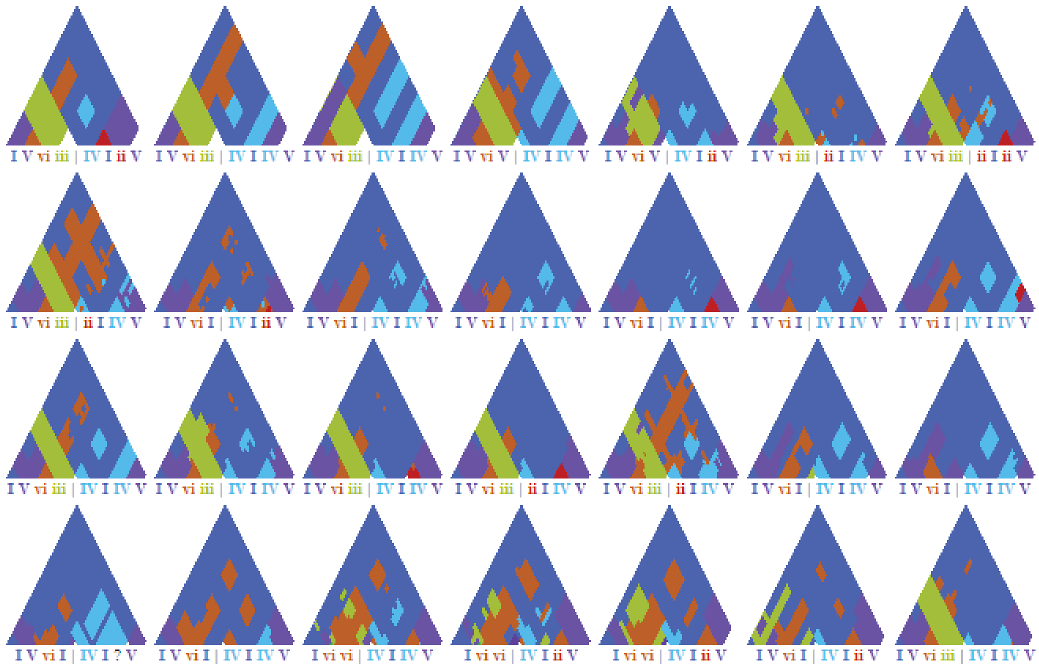


Figure 3.25: Keyscape plots for every two bars of Pachelbel's Canon. Harmonic analyses are displayed underneath each plot, colored according to the chord root.

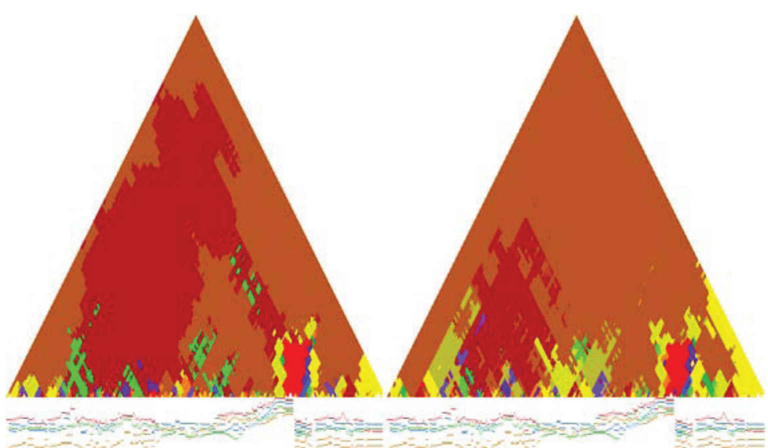


Figure 3.26: Keyscape plots for Samuel Barber's Adagio for Strings. Aarden weights used on the left and Krumhansl weights on the right. Underneath each plot is a piano-roll notation of the music, where the vertical axis represents pitch from low to high.

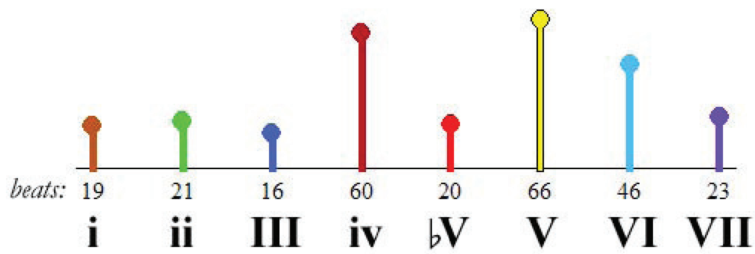


Figure 3.27: Relative durations of prominent chord sonorities present in the Adagio for Strings. Colors represents chord roots which match the key colors in Figure 3.26.

suddenly drop back to the starting register at about 75% of the way through the piece at the same time as the harmonic climax in the music.

Harmonically, the Adagio for Strings is an unusual composition. While the music is in B♭ minor, and both keyscape plots in Figure 3.26 agree with this analysis, the tonic chord of B♭ minor is used much less often than expected in a tonal piece. The opening of the piece starts with a single B♭ note, followed by a iv_7 chord and then a V chord, giving a weak introduction to the key of the piece. Another unusual tonal device is ending the piece in the dominant instead of the tonic, which can be seen in the yellow colored region on the bottom right of the plots.

In the piece, the tonic chord is avoided as much as possible. For example, during the entire piece, the full tonic chord is only played 19 of the 564 quarter-note durations in the piece (3.3% of the total duration). On the other hand, the subdominant (dark red in the plots) and the dominant (yellow) are each played about three times more often than the tonic chord, as demonstrated in Figure 3.27. In a typical tonal composition, more emphasis on the tonic chord and less on the subdominant chord would be expected.

3.5.4 Analysis of Anton Webern's Variations for Piano, op. 27, first movement

The first movement of Anton Webern's Piano Variations, op. 27 (1936) is given as a final example of keyscape pictures. Anton Webern (1883–1945) was an Austrian composer and a student of Arnold Schoenberg, the developer (around 1920) of twelve-tone music. In twelve-tone music, all twelve chromatic pitches of the octave are played in sequence before

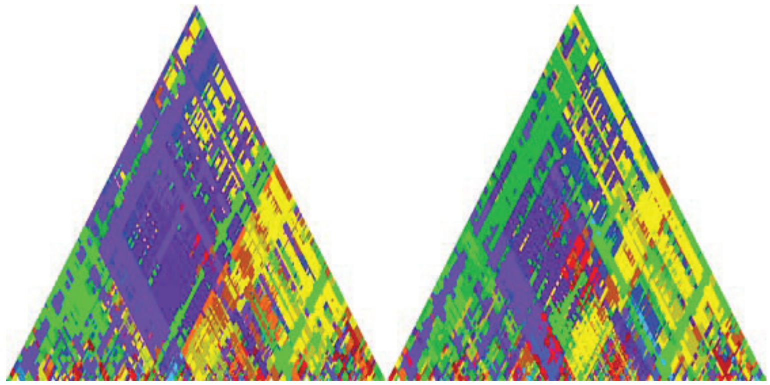


Figure 3.28: Keyscape plots for Anton Webern's Variations for Piano, op. 27, mvmt. 1. Aarden weights used on the left and Krumhansl weights on the right.

any notes are repeated. The goal is to destroy all perception of key and tonality: since no particular note is favored in the music, there can be no tonal center in the music.

Figure 3.28 shows keyscape plots of the first movement of the Piano Variations. In these plots, it is easy to see the esthetic of destroying a tonal center. As the analysis window becomes larger from bottom to top in the plot, the color regions remain fragmentary. No overall key region becomes dominant in the large-scale structure of the music. This is, of course, intentional in twelve-tone music. Previous example keyscape plots in the chapter were derived from tonal music, and there has always been a simplification in the key structure towards the musics top-level organization.

Not only are the higher-level regions of the keyscape plot fragmentary, but distant key regions are displayed adjacent to each other. The pitch-to-color mapping based on the circle of fifths also helps to demonstrate the lack of key in this example. Distantly related key regions are represented by contrasting colors. For example, violet and yellow are opposite colors representing key regions of A and F respectively; green and red represent musical regions C and E.

References

[1] Bret Aarden. *Dynamic Melodic Expectancy*. PhD thesis, School of Music, Ohio State University, 2003.

[2] Carol Krumhansl. *Cognitive Foundations of Musical Pitch, Oxford Psychology Series No. 17*, pages 81–96. Oxford University Press, 1990.

[3] Fred Lerdahl and R. Jackendoff. *A Generative Theory of Tonal Music*. MIT Press, 1983.

[4] Eugene Narmour. Beyond schenkerism: The need for alternatives in music analysis. 1977.

[5] Craig Sapp. Harmonic visualizations of tonal music. *Proceedings of the International Computer Music Conference*, pages 423–430, 2001.

[6] Rodrigo Segnini and C. Sapp. Scoregram: Displaying gross timbre information from a score. *Proceedings of the International Symposium on Computer Modeling and Retrieval*, 2005.

[7] David Temperley. *The Cognition of Basic Musical Structure*. MIT Press, 2002.

Chapter 4

Key modulation identification in tonal music

ABSTRACT

Techniques for identifying key modulation points in tonal music are described in this paper. Starting with a localized field of independent overlapping key measurements at multiple time-scales and positions throughout a musical score, key-regions are identified and further processed to improve accuracy at region boundaries by compensating for confusion between closely related keys. After processing key measurements to minimize errors at confusion boundaries, a hierarchical identification of key-regions is done, resulting in localization of key modulation boundaries within the music.

4.1 Localized key identification

The process of identifying key modulation boundaries in tonal music described in this paper covers three general processes: (1) local key identifications at various time-scales within a piece, (2) compensation for confusion boundaries between keys with key reassignments based on emergent properties, and (3) discrete hierarchical assignments of key-regions which generate the final modulation boundaries.

As an initial step in identifying modulation boundaries, the Krumhansl-Schmuckler key-finding algorithm (KS algorithm), is used to measure keys of localized segments of the music. Any key-finding algorithm would do, but since the KS algorithm does not require *a priori* knowledge of chords within the music before determining the key, this algorithm can handle a wide range of musical textures from monophonic to homophonic to polyphonic including complex temporal elaborations of chords, such as arpeggiation or Alberti bass figuration. This algorithm is a low-complexity method for measuring the musical key in a selection of music by correlating pitch-class histograms extracted from the music against prototype histograms (frequency distributions) expected for a typical major or minor key. [7]

In particular, a normalized form of correlation is used in the KS algorithm, often called Pearson product-moment correlation coefficient. Given two ordered sequences of input data, the Pearson correlation value (often called the r-value) is defined as:

$$r(x,y) = \frac{\sum_n (x_n - \bar{x})(y_n - \bar{y})}{\sqrt{\sum_n (x_n - \bar{x})^2 \sum_n (y_n - \bar{y})^2}} \tag{4.1}$$

where x and y are paired sequences of numbers and n is an index into the sequences for summation. The KS algorithm calculates a correlation value for each pairing of the measured pitch-class content in a musical selection with an expected pitch-class distributions for each candidate key. The comparison which yields the highest r-value is assigned to be the key for the music. The KS algorithm can be summarized in the equation:

$$\text{key}_k = \arg\max_k \; r(x, y_k) \tag{4.2}$$

where x is a histogram of pitch-classes present in the music (typically weighted by duration), and y_k is the set of prototype probability weights transposed into the 12 major and 12 minor keys starting on each chromatic pitch. The variable k is an index into the list of all keys being tested and "arg max" reports the k-th key which produces the highest correlation value with the measured pitch-class histogram.

In order to compare the KS algorithm to unnormalized forms of correlation, it is useful to express it in terms of z-scores, which are normalized forms of sequences that set the mean value of a sequence to 0 and the standard deviation to 1:

$$z_x = \frac{x - \bar{x}}{\sigma} \tag{4.3}$$

where $\bar{x}$ is the average, and σ is the population standard deviation:

$$\sigma = \sqrt{\frac{1}{N} \sum_n (x_n - \bar{x})^2} \tag{4.4}$$

with N being the number of elements in the sequence. Thus, the KS algorithm becomes:

$$\mathrm{key}_k = \arg\max_k \frac{1}{N} \sum_n z_{x,n}\ z_{y_k,n} \tag{4.5}$$

Note that the $1/N$ factor can be dropped from the equation without affecting the results.

The Discrete Fourier Transform has a similar form, but uses unnormalized correlation (dot product) to compare two sequences. A spectrum is generated by multiplying a signal with a set of harmonically-related complex sinusoids and summing:

$$X_k = \sum_n x_n\ e^{-\frac{2\pi j}{N}kn} \tag{4.6}$$

By analogy, the KS algorithm can be thought of as identifying the peak in a *key spectrum*, although the basis vectors y_k (musical keys) are not orthogonal. [5] In addition, the z-score form of the KS algorithm can be compared directly to an earlier key-identification algorithm developed by Gabura which uses dot-product correlation: [3]

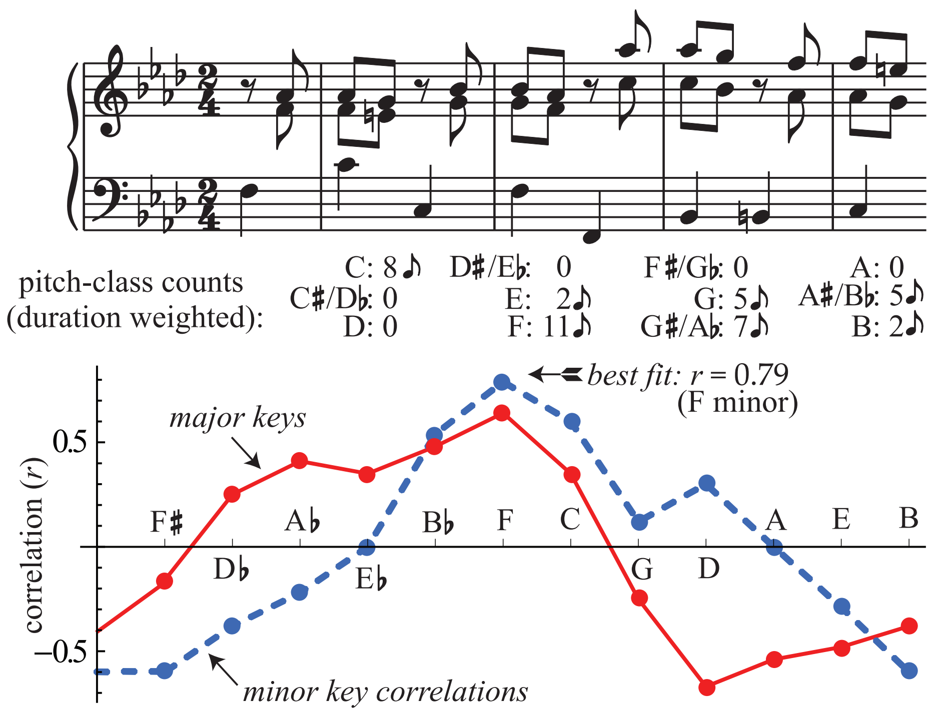
$$\mathrm{key}_k = \arg\max_k \sum_n x_n\ y_{k,n} \tag{4.7}$$

This algorithm is suitable for identifying the best major key or the best minor key, but without the normalization of Pearson correlation, choosing between the best major and best minor keys is problematic. [2]

Figure 4.1 gives an example application of the KS algorithm to four bars of music. First, a histogram of the pitch-classes is extracted from the music. This histogram is then correlated with expected histograms for each possible major and minor key. Example prototypes for these two key modes are given underneath the extracted histogram which are transposed into all keys. F minor generates the highest correlation value and is thus identified as the key of the musical selection.

4.1.1 Key prototype pitch-class weights

The original prototype frequency distributions for major and minor keys used in the KS algorithm were based on probe-tone ratings by human subjects collected by Krumhansl &



pitch-class histogram: $x = (8, 0, 0, 0, 2, 11, 0, 5, 7, 0, 5, 2)$

major key prototype: $y_{\mathrm{M}} = (2, 0, 1, 0, 1, 1, 0, 2, 0, 1, 0, 1)$

minor key prototype: $y_{\mathrm{m}} = (2, 0, 1, 1, 0, 1, 0, 2, 1, 0, 1, 0)$

Figure 4.1: Example KS algorithm calculations for bars 1–4 of J.S. Bach, BWV 881, with F minor identified as the best key of the musical selection.

scale degree	*semi-tones above tonic*	KK	AE	KP	BB	LC
		Major Key Weights				
1	0	6.35	17.77	0.748	16.80	2
	1	2.23	0.15	0.060	0.86	0
2	2	3.48	14.93	0.488	12.95	1
	3	2.33	0.16	0.082	1.41	0
3	4	4.38	19.80	0.670	13.49	1
4	5	4.09	11.36	0.460	11.93	1
	6	2.52	0.29	0.096	1.25	0
5	7	5.19	22.06	0.715	20.28	2
	8	2.39	0.15	0.104	1.80	0
6	9	3.66	8.15	0.366	8.04	1
	10	2.29	0.23	0.057	0.62	0
7	11	2.88	4.95	0.400	10.57	1
		Minor Key Weights				
1	0	6.33	18.26	0.712	18.16	2
	1	2.68	0.74	0.084	0.69	0
2	2	3.52	14.05	0.474	12.99	1
3	3	5.38	16.86	0.618	13.34	1
	4	2.60	0.70	0.049	1.07	0
4	5	3.53	14.44	0.460	11.15	1
	6	2.54	0.70	0.105	1.38	0
5	7	4.75	18.62	0.747	21.07	2
6	8	3.98	4.57	0.404	7.49	1
	9	2.69	1.93	0.067	1.53	0
7	10	3.34	7.38	0.133	0.92	1
	11	3.17	1.76	0.330	10.21	0

Table 4.1: Key prototype weights for use in the KS algorithm.

Kessler (KK column in Table 4.1). [7] Several proposals have since been made for alternate key prototype weights for use in the KS algorithm which are listed in Table 4.1, all derived from musical scores. The Aarden-Essen (AE) weights are generated from monophonic songs of the Essen Folksong Collection. [1] The Kostka-Payne (KP) weights are extracted from short harmonic examples as selected by Brian Pardo. [6,11] The Bellman-Budge (BB) weights are extracted from chord frequency distributions from 100 classical works. [2, 4] Other pitch-class prototype weightings could be extracted from any repertory or model. The last column in Table 4.1 list a proposed set of low-complexity weights which can be used as a baseline for performance of these or other weights, and which compensate for the primary confusion boundaries between keys illustrated in Figure 4.4.

4.2 Keyscape plots

By itself, the KS algorithm cannot identify more than one key at a time. The algorithm performs well when applied to musical segments within a single key; however if more than one key is present in the music being analyzed, the algorithm will only be able to choose one of them or occasionally a third irrelevant key caused by an unusual boundary case. At modulation boundaries, two keys will have the same *r*-value in the KS algorithm. [10, Fig. 7] Accurate boundaries are difficult to localize since the textural density of the music may not remain constant, causing drift in the observed modulation boundaries from their true locations in the music. Therefore, comparing secondary *r*-values within a musical segment cannot reliably hint to the presence or localization of modulation boundaries between two keys in the KS algorithm. In particular, highly chromatic music will naturally generate low *r*-values in a narrow range, and *r*-values of closely related keys will follow the tonic key's *r*-value (i.e., *non-orthogonal* basis vectors). The absence of modulation boundaries is easier to determine. Key identifications with high *clarity* (large *r*-values) coupled with low *ambiguity* (a large difference between the top two estimated key *r*-values) yield a high confidence that the correct key has been identified and that the likelihood of a modulation boundary within the musical segment is low. [10, *ibid.*]

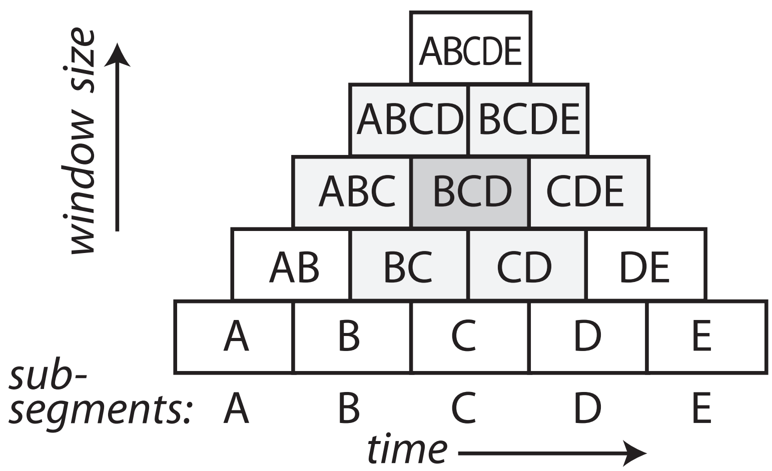


Figure 4.2: “Scape” plotting domain, with time on the horizontal axis and analysis window size on the vertical axis. Each domain cell is centered at the midpoint in the analysis window along the horizontal axis.

As a first step in the identification of modulation boundaries with the KS algorithm, or other similar algorithms which cannot identify modulations by themselves, a plotting method shown in Figure 4.2 can be used to display independent key analyses for all possible groupings of short musical segments. [10] In this plotting domain, all possible *n*-gram groupings of the sequential sub-segments (A, B, C, D, E in the example) are arranged in two dimensions such that each cell in the plot has six neighboring cells which vary by only one operation. For example, the highlighted cell represents the key analysis for the minimal segments (B, C, D) considered as a composite segment. The cells to the left and right on the row represent a shift of one unit to the left or right in the data respectively. For the two cells underneath BCD, an underlying segment either at the beginning or ending of the sequence is dropped, while on the row above, an extra segment is added to the beginning or ending of the BCD segment.

This plotting domain shows all possible contiguous segmentations of the music which can be used as a basis for examining the behavior of the KS algorithm at all possible time-scales and positions within the music. When displaying key-analysis data in this plotting domain, the plot is nicknamed a *keyscape* since it is analogous to a land*scape* painting where the foreground at the bottom of the picture displays small-scale localized features, while the top of the picture show large-scale distant features. Keyscape plots are useful for comparing differences between key-profile weights described in the previous section. For example, Figure 4.3 shows keyscapes for three different key-profile weights used with the KS algorithm from Table 4.1. The music represented in the plots is Chopin's prelude in C major, op. 28, no. 1. When applying the KS algorithm to the entire piece, represented by the top point in each plot, all weight sets agree that the piece is in C-major (colored green and labeled "tonic"). Note that the KK weights have significant difficulty in correctly identifying the key of the music at smaller segmentations except when the ending of the piece is included (right edge), since these weights are biased towards the dominant key. The AE weights are biased slightly towards the subdominant (probably due to the presence of dominant key-regions in the music used to generate prototype major/minor pitch-class histograms). It is for this reason that near the end of the piece, the AE weights detect a region in F major which is stronger than with the other two weight-sets. The KP weights probably give the most accurate view of this piece; however, these weights are

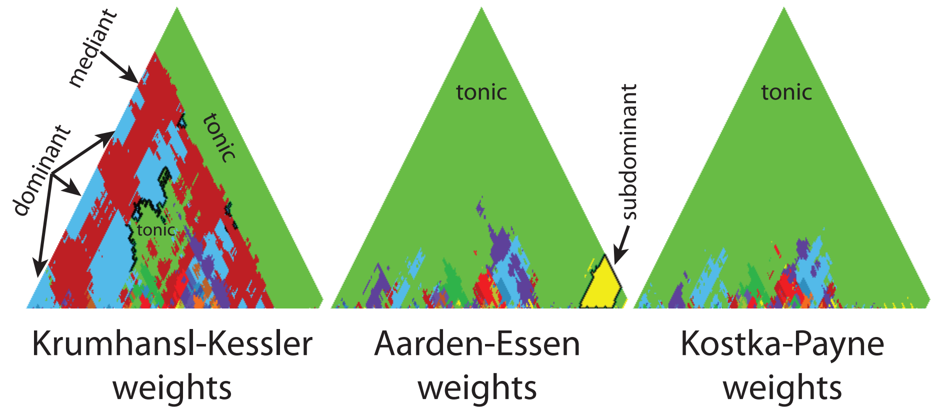


Figure 4.3: Comparison of the key-identification behavior for various key prototype weights (Chopin op. 28/1).

biased towards the relative major, a case not represented in the example (i.e., they have a preference for major keys).

These example keyscape plots show the importance of starting with a good set of key-prototype weights when trying to identify key modulation boundaries. In general the AE weights perform well in both major and minor keys except for the slight over-emphasis of the subdominant. However, the KK weights are strongly biased towards the dominant and are not optimal for identifying modulation boundaries. The KP weights perform very well for major keys, but will not perform as well with minor key-regions due to a relative-major bias. The BB weights perform about the same as the AE and KP weights with no particular bias towards neighboring keys, but will occasionally give unusual results from the interactions between keys. The low-complexity weights perform well at large time-scales but are somewhat dominant-biased.

4.3 Post-processing of raw key estimates

After regions in a keyscape plot have been identified from sets of adjacent cells having the same best-key analysis, the borders of these regions, which represent key modulations, can be improved. Borders of these regions also represent sections of the music where the KS algorithm has difficulty identifying a best key because at least two candidate keys have the same *r*-value. Therefore, any small perturbation of the prototype weights will shift the border between these two key-regions and thus move the measured modulation point between the keys. Starting with the raw key identifications generated by the KS algorithm, further processing can be done to improve key analysis results by applying methods which remove likely analysis error based on broader trends in the data.

The techniques described in this section can also be viewed as an examination of the emergent properties of the independent KS algorithm key identifications. [8] Individual cells in the keyscape plot group together to form key-regions, and the boundaries of these regions can be adjusted to better match a more musically relevant perception of key-regions and their resulting modulation boundaries.

4.3.1 Confusion boundaries

Not all boundaries between key-regions are equal. Most key identification errors occur at two boundary categories: (1) the tonic/dominant boundary, where the pitch-class content differs by only one note between the keys, and (2) the modal boundary, where the pitch-class content of the keys is the same and the distributions of pitch-classes distinguish the keys.

The Tonic/dominant (or conversely the subdominant/tonic) boundary is the closest relation between keys, and therefore causes the greatest identification confusion. The top portion of Figure 4.4 illustrates this boundary problem, where, for example, the keys of C and G major differ by only a single note (F♮ in C major or F♯ in G major). In addition, the pitch-class distributions expected for C and G major are nearly identical. For example, when the pitch-class prototype histograms for each key are arranged by the Circle of Fifths, they roughly look like highly overlapped Gaussian distributions.

The relative modal boundary is the next most confusing key relationship between keys. Figure 4.4 shows the relative major/minor pairing of C major and A (natural) minor. Both of these keys contain the same pitch-classes and can only be distinguished by different pitch-class distributions in the KS algorithm. For example, the pitch-class *a* is very common in A minor, but not as common in C major. It is interesting to note that functional

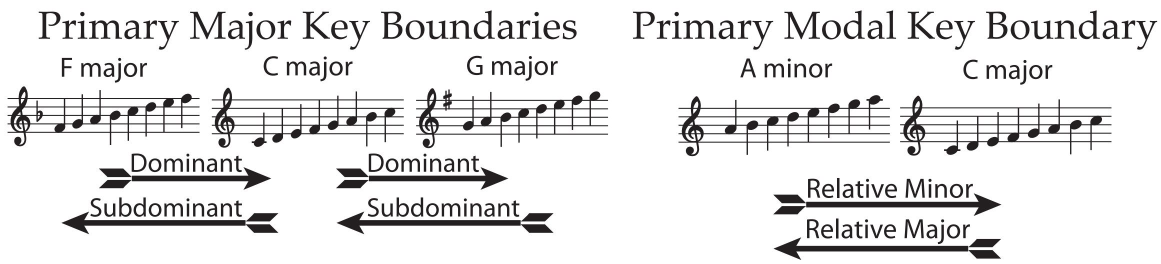


Figure 4.4: Nearest-neighbor key boundaries in functional tonality. Major keys related by a fifth differ by one pitch class. Relative major and (natural) minor keys share the same pitch classes.

harmony uses only two modes, and other types of modal boundaries do not need to be considered in functional harmony.

These two primary kinds of boundary confusion between closely related keys are indicated schematically in Figure 4.5, along with the parallel modal boundary, which is usually the next most confounding aspect of key-relationships. These boundaries relate a tonic key to its most similar neighboring keys, and these relationships have been used to create two-dimensional and toroidal arrangements of keys. [7, 9]

Most key-identification algorithms that do not first identify chord sequences and instead use statistical methods will have difficulties at these boundaries, but they can perform better at boundaries between less similar keys, since the pitch-class content and pitch-class distributions will be more easily separable. For the following processing techniques, it is important to note that each confusion-boundary is independent. Only one boundary causes confusion at a time, and corner cases where three keys intersect at a confusion boundary

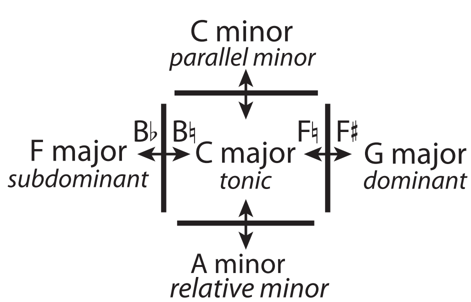


Figure 4.5: Primary confusion boundaries for key of C major: (1) Circle-of-Fifth shift by one pitch class and (2) modal shifts using same pitch classes.

are much less common. Thus only the top two keys identified by the KS algorithm are used to improve region boundaries in the keyscape plot in the following techniques.

4.3.2 Spurious region identification

The first step in post-processing raw key identifications is to determine whether or not contiguous best key-regions attach to the bottom of the keyscape plotting domain. Regions which are attached to the bottom have a reasonable chance of representing true key-regions; otherwise, note that it is difficult to have a musical key suddenly appearing at a higher time-scale without the presence of a tonic triad or dominant seventh chord in the music underneath it to define the key.

Any key-region which is not attached to the bottom of the plotting domain is marked as invalid and blanked out. These floating regions are very likely to be incorrectly identified, and instead they probably represent disjoint smaller-scale key-regions which are arranged in such a way that they become focused at longer time-scales further up in the plot. The region marked with an '*a*' in Figure 4.6 demonstrates this sort of error. Another type of error which can occur is a third key-region cropping up when two or more key-regions mix together in the input histogram as marked by '*b*' in Figure 4.6. In particular, note that the KS algorithm removes all sequential information from the music when an analysis is done, so mixing distant keys together or highly chromatic music will tend to cause this sort of problem.

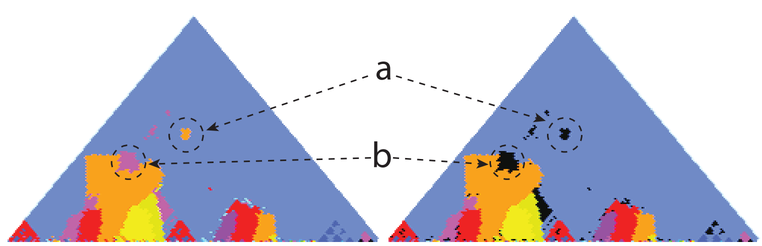


Figure 4.6: Removal of tonal regions not attached to bottom of plot. Two general cases: (a) weak high-level confluence of the same tonal region at smaller time-scales, and (b) spurious tonal region created from a mixture of unrelated keys.

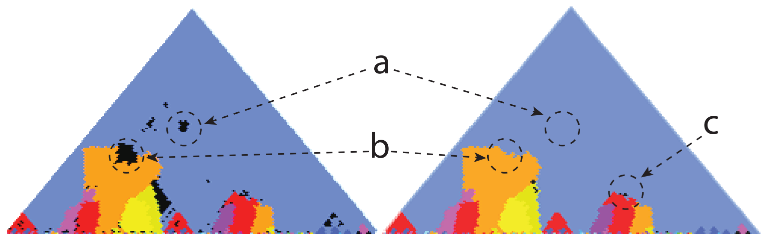


Figure 4.7: Filling process in spurious regions with second-best keys identified by the key-finding algorithm along the region border.

4.3.3 Spurious region filling

After floating key-regions have been invalidated, an attempt is made to fill in these spurious regions with the second best key estimations generated by the key-finding algorithm. In order to determine whether or not to select the second-best key, that key-region must be present along the boundary of the blanked out region. An easy method to satisfy this requirement is to replace best-key identifications with second-best measurements in the invalidated regions, and then apply the spurious region identification step again. Figure 4.7 demonstrates this process applied to Chopin's prelude in D major, op. 28, no. 4. Notice that the regions marked '*a*' are now completely filled in with the second-best key identification. Nearly all of the regions like that marked '*b*' could be filled, leaving a small undefined section. Similarly, the region marked '*c*' could not be completely filled with second-best key results. Regions left undefined at this point probably generate a poor sense of key, since the two preceding answers from the key-finding algorithm are likely to be incorrect.

4.3.4 Region trimming

After removing spurious regions from the plot, all remaining regions will be attached to the bottom of the plot (or will be marked as undefined regions). These regions can be enumerated from left to right along the bottom as illustrated in Figure 4.8. A process named *trimming* can be done whenever a previously enumerated region returns in the sequence. This repetition of a region indicates that an intermediate set of regions is completely constrained by the bounding region. Any portions of these bounded key-regions extending beyond the encompassing border are likely to be identified incorrectly. These expanded

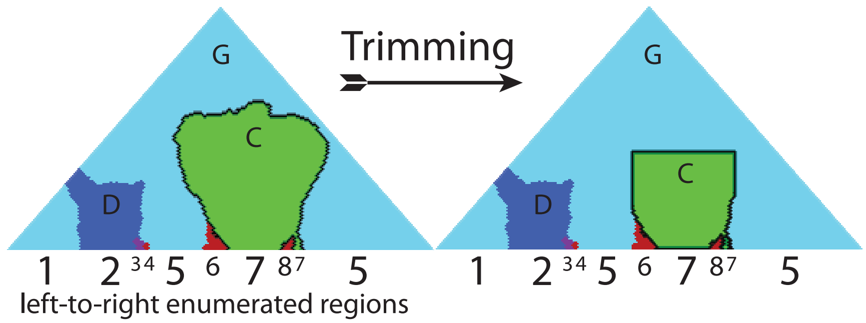


Figure 4.8: Process of trimming regions with regions enumerations along the bottom of the plot from left to right.

sections of the bounded regions are trimmed by replacing the key identification in these areas with the key of the encompassing region regardless of its identification rank according to the key-finding algorithm, as shown on the right side of Figure 4.8.

4.3.5 Tonal biasing

In a vast majority of tonal compositions, the primary key is presented at the start of the work, and this same key returns at the end, with secondary keys present between these two regions. To compensate for strong medial key-regions in the region-trimming process, a preference can be given for initial and terminal key-regions by examining the left and right edges of the keyscape plot. If both the bottom left edge and the bottom right edge of the plot share the same key identification, then this key can be presumed to be the overall key of the composition, and a bridge between these non-adjacent regions on the keyscape can be used to force a connection between them along the edge of the plot as illustrated in Figure 4.9. In this case, regions 1 and 5 from Figure 4.8 are now connected, allowing region 2 to be trimmed. The tonal biasing process may not be suitable for compositions which do not start and end in the same key, a notable example being Chopin's prelude in A minor, op. 28, no. 2, which starts in E minor and progresses to G major, and then to D major, until finally arriving at A minor in bar 15—over half-way through the piece.

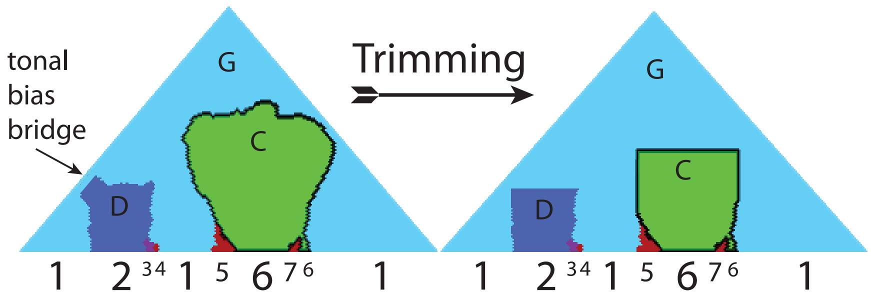


Figure 4.9: Process of connecting regions along the right (and/or left) edges of the plot which are identified as the tonic key of the music.

4.3.6 Tunneling

When a small-scale region represents the same key as a large-scale region with some intermediate key-region separating them, a technique named *tunneling* is useful to join the two regions together and split the intermediate key-region into two smaller regions. Tonal biasing can be considered a special case of the tunneling technique. Key prototype weights or textural changes in the music can emphasize secondary keys more than the primary key, and as a result, these secondary key-regions may join to form a single region. The tunneling technique corrects for this problem in the raw key estimates.

Figure 4.10 shows the tunneling process being applied to Canzon 12, libro primo (1634) by G. Frescobaldi. An intermediate region in G major divides large-scale and small-scale regions of C major. In order to tunnel, the algorithm searches in the intermediate region for a vertical path between the small- and large-scale regions such that the second-best key along the path matches the key of the two regions which are trying to connect. In Figure 4.10, the second-best keys are displayed on the top right. Note that the second-best key in the intermediate G-major region remains C major (between the two regions being tested for a connection). Therefore, a path linking the two C-major regions is created (bottom left). This allows for the G-major region split by the tunneling process to be trimmed into more musically reasonable presentations of G major within this C-major composition.

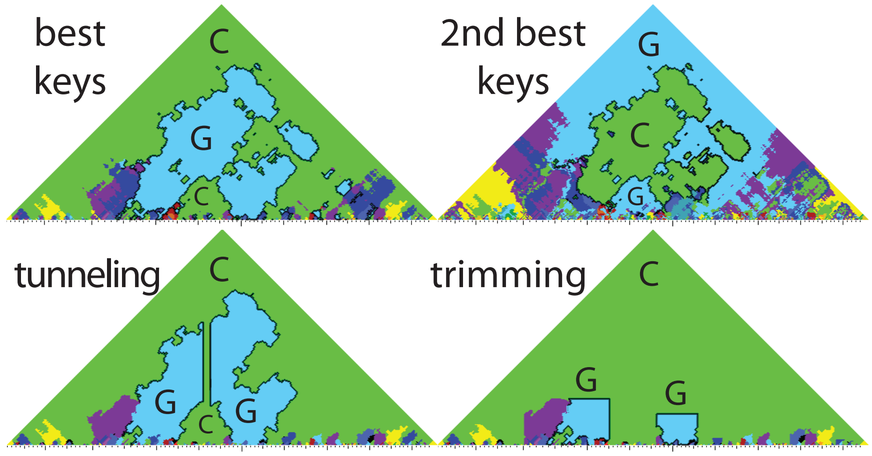


Figure 4.10: Tunneling process: small C major region in middle of music is connected to global C region through G major region via uninterrupted bridge of second-best key with the G region.

4.4 Key-region discretization

Once region boundaries of independent key measurements have been adjusted using methods described in the previous section, the task of identify musically-relevant keys and their modulation boundaries is relatively straight-forward. Note from the example plots in Figure 4.8 and Figure 4.10 that wide, and hence more important, key-regions reach higher into the plotting domain. The final step is to follow the left and right edges of key-region boundaries to the bottom of the plot and to use these intersection points as the locations of modulation boundaries within the music. The modulation points of larger key-regions will be determined first, and they are allowed to wander into smaller key-regions via second-best key identifications.

The first step in identifying these modulation points is to enumerate the key-regions from top to bottom in the plot. This is done to roughly rank the keys by importance. When two or more regions have the same height in the plot, the left-to-right ordering of the regions is used as a tie-breaker to sort the priority of the regions.

Next, the connecting points of the left and right sides of each key region at the bottom of the plotting domain are refined. When searching along the edges of a key-region towards the connecting point at the bottom of the plot, both the best and second-best key values

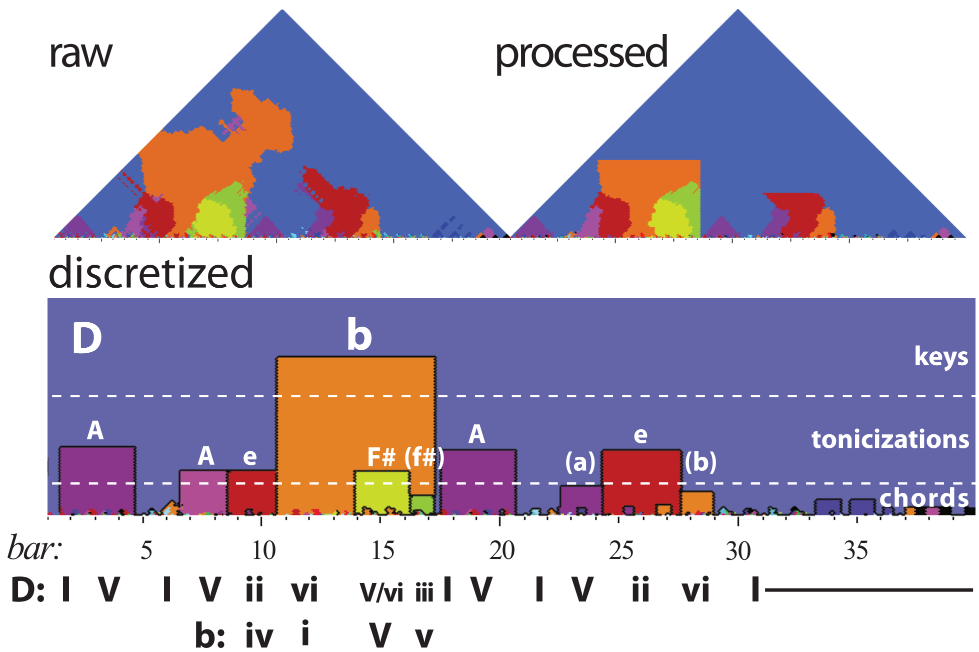


Figure 4.11: Process of extracting discrete key and modulation boundaries regions from raw, then processed local key identifications.

in cells are used, provided that earlier enumerated (larger) regions are excluded. In other words, a key-region can expand outwards to the left or right into smaller key-regions if the second-best key within these smaller regions matches the one currently being tested. This enables a more important key-region to engulf less important key-regions if they are sufficiently related. For example, the B-minor section from bars 11–16 in Figure 4.11 is able to pass through the F♯-minor region on its right edge. Thus, the B-minor section can expand from 3 bars to a more musically accurate 6 bars by including its dominant (F♯). Once the left and right connection points of a key-region have been identified, the algorithm progresses to the next enumerated region in the plot and performs the same process of locating the region's attachment points at the bottom of the plot.

The bottom of Figure 4.11 shows the final result of this modulation-point identification displayed graphically underneath the two initial stages of local-key identification and boundary refinements. When determining modulation boundaries, there is a continuum between keys, tonicizations (temporary keys), and chords. The cut-off levels between these key categorizations will be dependent on the music, but a rough guideline is that regions larger than a phrase (typically four to eight bars) are keys while smaller regions are tonicizations.

In Figure 4.11, the key-duration level is set so that there are two keys within the music. The global key for the music is in D major (which matches the key assignment given by the composer), and there is a secondary key of B minor in bars 11–16. Tonicizations and elongated chords are also labeled in the plot, and they show that the B-minor region is entered through its minor subdominant and exited through its minor dominant. The pattern of key-regions in this example reveals the structure of the piece, which is a section from bars 1–16 being repeated, with the second occurrence de-emphasizing B minor and transitioning to a coda in the tonic key.

4.5 Conclusions

The KS algorithm (or other low-complexity key-finding algorithms) can be used to identify the key in a wide range of textures and styles of tonal music. By itself the algorithm cannot accurately identify modulation boundaries, but with the analysis of independently measured results from the algorithm, borders between key-regions can be refined to the point that reasonably robust modulation boundaries can be assigned. These methods can be used in a complementary manner with bottom-up chordal methods of defining key, and they can provide reference points for identifying chords themselves. In addition, further refinements of key-prototype weights could be optimized by evaluating hand-annotated modulation points against automatically extracted modulation locations using these methods.

References

[1] Bret Aarden. *Dynamic Melodic Expectancy*. PhD thesis, School of Music, Ohio State University, 2003.

[2] Héctor Bellman. About the determination of key of a musical excerpt. *Proceedings of the International Symposium on Computer Modeling and Retrieval*, pages 187–203, 2005.

[3] James Gabura. Music style analysis by computer. *The Computer and Music*, 1970.

[4] Budge Helen. *A Study of Chord Frequencies Based on the Music of Representative Composers of the Eighteenth and Nineteenth Centuries*. PhD thesis, Columbia University, 1943.

[5] Özgür İsmirli. Cyclic-distance patterns among spectra of diatonic sets. *Tonal Theory for the Digital Age, Computing in Musicology*, 15:11–23, 2008.

[6] S. Kostka and D. Payne. *Tonal Harmony*. McGraw-Hill, 4th ed. edition, 2000.

[7] Carol Krumhansl. *Cognitive Foundations of Musical Pitch,* Oxford Psychology Series No. 17, pages 81–96. Oxford University Press, 1990.

[8] Robert Laughlin. *A Different Universe: Reinventing Physics from the Bottom Down*. Basic Books, 2005.

[9] Hendrik Purwins, B. Blankertz, and K. Obermayer. Toroidal models in tonal theory and pitch-class analysis. *Tonal Theory for the Digital Age,* Computing in Musicology, 15:73–98, 2008.

[10] Craig Sapp. Harmonic visualizations of tonal music. *Proceedings of the International Computer Music Conference*, pages 423–430, 2001.

[11] David Temperley. *Music and Probability*, page 85. MIT Press, 2007.

Chapter 5

Comparative analysis of multiple musical performances

ABSTRACT

A technique for comparing numerous performances of an identical selection of music is described. The basic methodology is to split a one-dimensional sequence into all possible sequential sub-sequences, perform some operation on these sequences, and then display a summary of the results as a two-dimensional plot; the horizontal axis being time and the vertical axis being sub-sequence length (longer lengths on top by convention). Most types of time-wise data extracted from performances can be compared with this technique, although the current focus is on beat-level information for tempo and dynamics as well as com-mixtures of the two. The primary operation used on each sub-sequence is correlation between a reference performance and analogous segments of other performances, then selecting the best correlated performances for the summary display. The result is a useful navigational aid for coping with large numbers of performances of the same piece of music and for searching for possible influence between performances.

5.1 Introduction

In the Mazurka Project[1] conducted at CHARM along with Nicholas Cook and Andrew Earis, we have collected over 3,000 recorded performances for 49 of Chopin's mazurkas—on average over 80 performances for each mazurka. Keeping track of differences and similarities between numerous performances is difficult when comparing recordings heard weeks, months or even years apart. And remembering the distinguishing features of 80 individual performances of a composition would be taxing on anyone's memory. Often the surface acoustics of a performance (such as reverb, microphone placement, piano model, recording/playback noise) are more noticeable and memorable than the actual performance, so identifying related performances solely by ear can sometimes be difficult.

A written score contains only the most basic of expressive instructions. The composer relies on the performer to interpret the work according to implicit rules as well as the written instructions. The unwritten rules of a composition are transmitted aurally between performers as well as passed down from teacher to student. These performance conventions

[1] `http://mazurka.org.uk`

can apply to specific pieces, genres, composers or entire time periods. Performances may involve combining interpretations from several sources, such as teachers or other admired pianists; or conversely, it could be a reaction against convention.

To help in the exploration of influences between performances, basic descriptions of tempo and dynamics are extracted from each performance of a work which can then be correlated against each other. A single global similarity measurement for this data could miss interesting smaller-scale structures. Therefore, the following plots were developed which display the closest performance to the reference at all possible timescales.

In the most interesting variation of the plot, each performance is assigned a color, and when a particular performance is most similar to the reference, its color is filled in the corresponding point in the plot. As a result of looking at all time spans, patterns of color emerge which can give clues to the relative importance of other performances to the reference performance of the plot.

5.2 Raw Data

Two types of data are used for comparative analysis: beat duration and loudness. There are many other facets of performance which are being ignored, such as individual note timings, voicing, pedaling, and articulation. However, tempo and overall loudness level at the beats are easier to extract from audio data than many other expressive features and form a reasonable expressive baseline.

Both tempo and loudness data are extracted beat by beat throughout a performance, and the data can be plotted against the sequence of beats as illustrated in Figure 5.1. While the data is extracted by beat from the performances for this paper, we are also working on extracting individual note times and dynamics (including off-beats as well as hand synchrony). Such fine-grained performance information may prove useful in characterizing similarities or differences between performances.

Beat durations are extracted by first recording taps in real-time while listening to a performance in an audio editor called Sonic Visualiser developed at the Centre for Digital

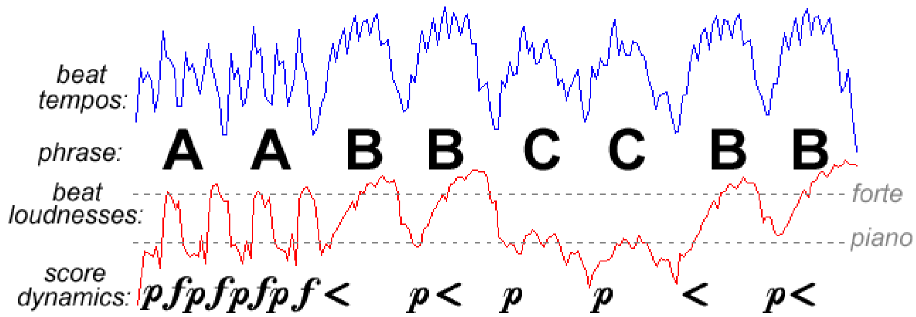


Figure 5.1: Average tempo and dynamic graphs for 35 performances of mazurka in B minor, op. 30/2.

Music at Queen Mary, University of London.[2] The resulting taps are not aligned precisely to true beat onsets in mazurkas due to a lag in response by the listener—typically with a standard deviation of 60–80 ms (compared to about 30 ms for following a steady tempo). Therefore, audio analysis plugins are used to assist in adjusting the taps onto the exact attack times of notes played on the beats.[3] By repeating data entry for the same performance in an independent manner, the alignment error is reduced to a standard deviation of around 11 ms. Defining a data error as a difference in beat localization by more than 50 ms, the measured data-entry error rate was about 1% for recordings made after 1980 and 3% for recordings in good condition from the early 1920's.

At timing resolutions around 10 ms, defining beat location can become difficult in piano music, particularly due to attack-time differences between the left and right hands (hand synchrony). In these cases, the best procedure is to define the beat location in a consistent manner in the analogous places in each performance. Since the melody usually contains more expressive timing, it is useful to define the beat as the time at which the melody note is played rather than using the less-expressive accompaniment.

For comparisons of musical dynamics between performances, a smoothed version of the raw power calculated for the audio signal every 10 ms is sampled at each beat location.

[2] `http://www.sonicvisualiser.org`
[3] `http://sv.mazurka.org.uk`

The raw power in decibels in a sample of audio is given by the equation:

$$\text{raw power} = 10\ \log_{10}\left(\frac{1}{N}\sum_n x_n^2\right) \tag{5.1}$$

where N is the number of audio-samples in sequence x being considered. The raw power measurements are then smoothed with an exponential smoothing filter described by the following difference equation:

$$y[n] = \alpha\, x[n] + (1-\alpha)\, y[n-1] \tag{5.2}$$

where α is a constant set to 0.2 in the case of 44100 Hz audio data with power measurements made every 10 ms. The exponential smoothing filter is applied twice to the raw power data: once in the forward direction and once in the time-reversed direction. This keeps the smoothed data centered at its original time location. To extract a loudness level for a particular beat in the audio, the smoothed power value about 70 ms after that onset is used—to compensate for a loss of high-frequency information in the smoothed data which delays the maximum amplitude location of note attacks.

5.3 Analysis Tools

5.3.1 Correlation

Normalized correlation, or *Pearson* correlation, is defined in Equation 5.3. This form of correlation yields values in the range from -1.0 to $+1.0$, with 1.0 being an exact match, and 0.0 indicating no predictable relation between the sequences being compared.

$$r(x,y) = \frac{\sum_n (x_n - \bar{x})(y_n - \bar{y})}{\sqrt{\sum_n (x_n - \bar{x})^2\ \sum_n (y_n - \bar{y})^2}} \tag{5.3}$$

where x and y are number sequences of the same length; $\bar{x}$ and $\bar{y}$ are average values of each number sequences x and y.

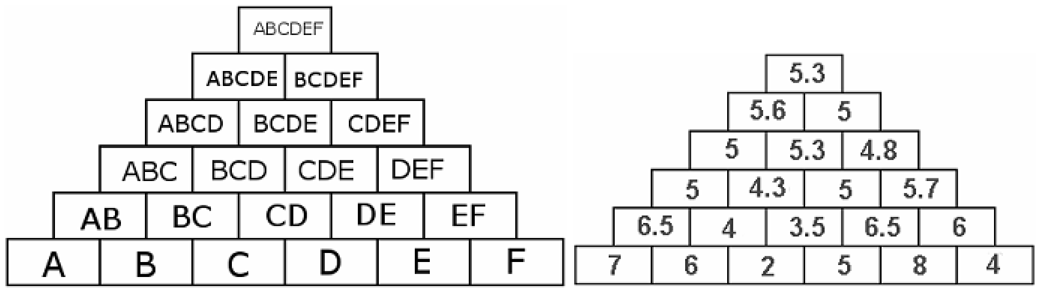


Figure 5.2: Scape plotting domain (left) and an example application of averaging in each cell (right), where the original data sequence is (7,6,2,5,8,4).

Correlation is a useful way to measure the similarity between two shapes such as comparing different performers tempo and dynamic curves as shown in Figure 5.1.

5.3.2 Scape plot

Correlation values are difficult to interpret in isolation, so the following plotting method is one way of presenting the data in a more human-readable format. Scape plots take their name from the word *landscape* since they show small-scale features analogous to the foreground in a picture, as well as large-scale features similar to the background. And like a painting, the interesting parts of the scape plot usually lie somewhere in the middle-ground.

Consider a simple example illustrated in Figure 5.2. A musical performance consists of six beats which are labeled: A, B, C, D, E, and F. These six beats can be chopped up into 21 unique sub-sequences (*n*-grams). Firstly, the elements can be considered in isolation. Next they can be grouped by sequential pairs: AB, BC, CD, DE, EF. Then by threes: ABC, BCD, CDE, DEF; by fours: ABCD, BCDE, CDEF; by fives: ABCDE, BCDEF; and finally one sequence covering the entire performance: ABCDEF. All of these possible sub-sequences of the basic six-beat performance, can be arranged on top of each other to form the arrangement shown in Figure 5.2.

Originally the scape plotting method was designed for structural analysis of harmony in musical scores ([2] and [3]). However, it has also been applied to audio-based harmony analysis [1] and timbral analysis [4].

5.4 Comparative Performance Scapes

What operation is done in each cell of a scape plot is arbitrary. The plot on the right in Figure 5.2 shows the application of averaging in each cell. In the following subsections, the calculation for each cell is done using the following steps:

- Choose one performance to be the reference for a particular plot.
- For each cell in the scape plot, measure the correlation between the reference performance and all other performances, then make note of the performance which yields the highest correlation value.
- Color the cell with a unique hue assigned to that highest-correlating performance.

Note that the actual correlation values are thrown away in this variation of the scape plot. This is primarily because the plots would become too complex and confusing if it were kept (for example displayed as gray-scale mask on the indexed performance colors). Other plot variants may display raw correlation values such as one that correlates half-sine arches to performance data for identifying phrasing structure.

5.4.1 Timescapes

Figure 5.3 demonstrates a pair of similar performances found in the set for mazurka in C major, op. 24, no. 2. Mutual best matching seen in this figure indicates a strong link between two performances and is less likely to be caused by chance. However, other structures seen in this figure are more likely to be random links to other performances with no interesting relationships. The total area covered in a plot by a particular performance is also an indication of significance, but less so than mutual similarity between two particular performances. In this case the performance on the left contains an area of 76% from another particular performance, and that performance in turn contains 58% by area of the original performance. Who was influenced by whom cannot be deduced from the plots. They only show that there is a strong relationship between the two performances in this case. Clues as to what is going on can be gleaned from the fact that the performance on the

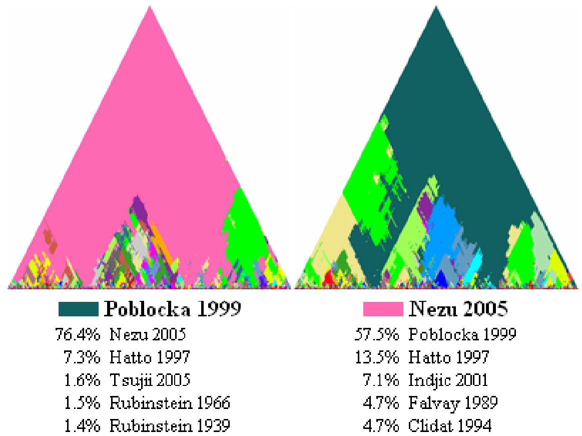


Figure 5.3: Timescapes for two performances of mazurka in C major, op. 24/2, showing teacher/student pairing, each showing large regions of best-correlation to each other (out of 35 performances).

left was recorded in 1999 and the one on the right in 2005; also the performer on the right did post-graduate studies with the performer represented on the left.

It is often useful to include the average of all performances in the collection of a piece of music being analyzed so that minor and random relationships between performances are hidden by the similarity to the average performance which is usually quite strong. Figure 5.4 demonstrates the effect of including the average performance along with the other real performances (compare to Figure 5.3).

In all five mazurkas examined comprehensively so far, all performers for which we have multiple recordings of show very strong relations to each other, regardless of the amount of time between the recordings. In Figure 5.5, three recordings of Arthur Rubinstein are displayed—an early, middle and late career sampling covering a time period of 25 years. In each case, the closest performance to the reference is another Rubinstein performance.

5.4.2 Dynascapes

Beat-level tempo is fairly unique to each performer, and when there is a strong mutual similarity between performers, it is usually not likely to be a coincidence. For dynamics (beat-level amplitude measurements in this case), the uniqueness is less pronounced due in

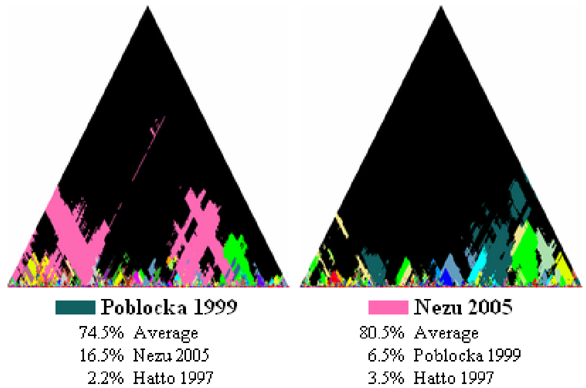


Figure 5.4: Same performances as in Figure 5.3, but with the average of all performances included (black).

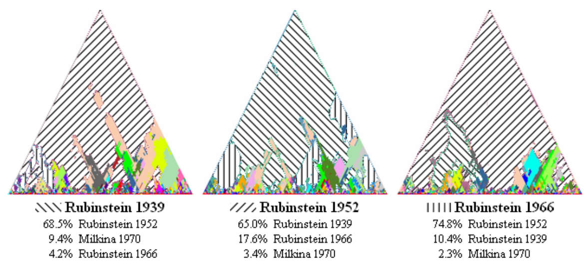


Figure 5.5: Timescapes for three performances of mazurka in B minor, op. 30/2. showing early, middle and late career performances by Arthur Rubinstein.

part to the composer writing basic loudness guides such as *forte* or *piano* in the composition or data extraction accuracy. Dynamics (as extracted in this study) are less unique to a single individual performer, and a greater likelihood of random patterns make the plots more difficult to interpret than when using tempo data. Also, it is possible that tempo expressivity is more static between performances, while loudness is easier to consciously control.

However, Figure 5.6 shows some nice mutually similar dynascapes for the same performer, recorded almost 40 years apart. In this case, the performer is closest to his dynamic interpretations in these two performance than to any of the other 58 performance of the same work which were examined. Also consider that the performances were recorded in very different technological eras, the first in the time of 78 rpm records, while the later one in the 33.3 rpm era.

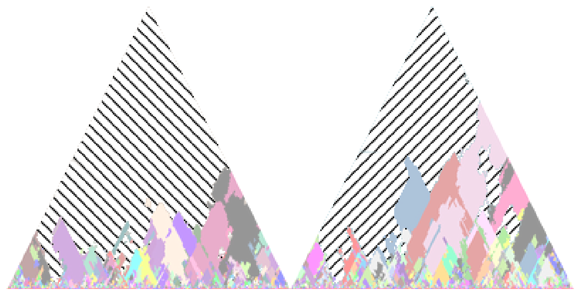


Figure 5.6: Two dynascapes of mazurka in C♯ minor, op. 63/3, showing early/late career pairing of performers.

5.4.3 Scape plots of parallel feature sequences

For Pearson correlation calculations, the ordering of the data is not significant as long as the sequence order is identical for both performances. But to generate multi-feature scape plots with a structure similar to the single-data forms, the independent values are interleaved in the correct time order so that the structure in the scape plot remains analogous to the single-sequence plots. To combine tempo and dynamics for comparison between performers, the time series of each feature are interleaved. Here are examples of two data sequences for tempo and dynamics to be mixed:

$$t = (t_1, t_2, t_2, t_4, ..., t_n) \tag{5.4}$$

$$d = (d_1, d_2, d_2, d_4, ..., d_n) \tag{5.5}$$

To mix them together with equal strength, create another sequence of joint features which interleaves tempo and dynamic values:

$$J = (J_{t,1}, J_{d,1}, J_{t,2}, J_{d,2}, ..., J_{t,n}, J_{d,n}) \tag{5.6}$$

To minimize the effect of mixing unrelated data in such a manner for the correlation calculations, the standard deviation and mean of the two sets of data should be equivalent. In this case the tempo values are left unchanged since they contain more performance

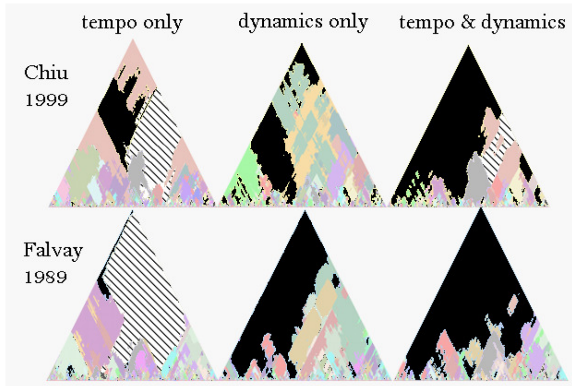


Figure 5.7: Tempo, dynamics and joint data plots. Black regions indicate mutual best matches. Striped region indicates a third performer common to both.

information to start with:

$$J_{t,n} = t_n \tag{5.7}$$

while the loudness sequence's standard deviation and mean are adjusted to match that of the tempo sequence:

$$J_{d,n} = s_t \left(\frac{d_n - \bar{d}}{s_d} \right) + \bar{t} \tag{5.8}$$

where s_x means the standard deviation of a sequence x, and $\bar{x}$ represents the mean value of a sequence x. The joint sequence can either be created globally, or locally based on the sub-sequence data (the latter would not work well at small timescales).

Figure 5.7 demonstrates the benefit of finding a performance match which is probably not random. When only time data is compared, there is little direct matching between the two performances. Comparing dynamics alone gives a stronger match between the performances, but is difficult to ascertain if the match is relevant due to the limited range for dynamics between performances. However, when both time and dynamic data are processed in parallel into a scape plot, the match between the performance becomes clear, and is likely to show a direct relation between the performance rather than a random occurrence.

5.5 Conclusions and Future Work

Significance of correlation measurements is difficult to assess in performance data since it is hard to statistically model a performer. So the precise meanings of the color patterns which emerge are not easy to pin down. Scape plots are a step towards identifying significant relations and can show where in a performance similarities are occurring.

The most difficult aspect of the plots is determining how relevant the best matches between performances are. Large patches of color do seem to be more significant, but not always. In particular, if a patch of color starts from a point and widens as it rises in a plot, it is most likely due to chance. Mutual best-matches between performers seems to be a good indication of significance, and sharp boundaries between color regions also tend to indicate more significant matches.

Tempo data in particular can be a superposition of several types of performance features. In mazurkas, for example, the low-frequency tempo component (phrasing) can be controlled independently by the performer from the high-frequency mazurka metrical pattern (where the first beat is typically shorter than the other two) and time accentuation of notes. Thus, it would be useful to identify and extract single performance features and compare them in isolation as well as in composite.

References

[1] Emilia Gómez and Jordi Bonada. Tonality visualisations of polyphonic audio. *Proceedings of the International Computer Music Conference*, 2005.

[2] Craig Sapp. Harmonic visualizations of tonal music. *Proceedings of the International Computer Music Conference*, pages 423–430, 2001.

[3] Craig Sapp. Visual hierarchical key analysis. *Computers in Entertainment*, 3(4), Oct. 2005.

[4] Rodrigo Segnini and C. Sapp. Scoregram: Displaying gross timbre information from a score. *Proceedings of the International Symposium on Computer Modeling and Retrieval*, 2005.

Chapter 6

Hybrid numeric/rank similarity metrics for musical performance analysis

ABSTRACT

This paper describes a numerical method for examining similarities among tempo and loudness features extracted from recordings of the same musical work and evaluates its effectiveness compared to Pearson correlation. Starting with correlation at multiple timescales, other concepts such as a performance "noise-floor" are used to generate measurements which are more refined than correlation alone. The measurements are evaluated and compared to plain correlation in their ability to identify performances of the same Chopin mazurka played by the same pianist out of a collection of recordings by various pianists.

6.1 Introduction

As part of the Mazurka Project at the AHRC Centre for the History and Analysis of Recorded Music (CHARM)[1], almost 3,000 recordings of Chopin mazurkas were collected to analyze the stylistic evolution of piano playing over the past 100 years of recording history, which equates to about 60 performances of each mazurka. The earliest collected performance was recorded on wax cylinders in 1902 and the most recent posted as home-made videos on YouTube. Table 6.1 lists 300 performances of five mazurkas which will be used for evaluation later in this paper since they include a substantial number of recordings with extracted tempo and loudness features.

Mazurka		*Performances*	
Opus	*Key*	*Collected*	*Processed*
17/4	A minor	93	63
24/2	C major	63	63
30/2	B minor	60	33
63/3	C♯ minor	88	88
68/3	F major	51	50

Table 6.1: Collection of musical works used for analysis.

[1]http://www.charm.rhul.ac.uk

For each of the processed recordings, beat timings in the performance are determined using the *Sonic Visualiser* audio editor[2] for markup and manual correction with the assistance of several vamp plugins.[3] Dynamics are then extracted as smoothed loudness values sampled at the beat positions. [3] Feature data will eventually be extracted from all collected mazurkas in the above list, but comparisons made in Section 3 are based on the processed performance counts in Table 6.1. Raw data used for analysis in this paper is available on the web.[4]

Figure 6.1 illustrates extracted performance feature data as a set of curves. Curve 6.1a plots the beat-level tempo which is calculated from the duration between adjacent beat timings in the recording. For analysis comparisons, the tempo curve is also split into high- and low-frequency components with linear filtering.[5] Curve 6.1b represents smoothed tempo which captures large-scale phrasing architecture in the performance (note there are eight phrases in this example). Curve 6.1c represents the difference between Curves 6.1a and 6.1b which is called here the desmoothed tempo curve, or the residual tempo. This high-frequency tempo component encodes temporal accentuation in the music used by the performer to emphasize particular notes or beats. Mazurka performances contain significant high-frequency tempo information, since part of the performance style depends on a non-uniform tempo throughout the measure—the first beat usually shortened, while the second and/or third beat are lengthened. Curve 6.1d represents the extracted dynamics curve which is a sampling of the audio loudness at each beat location.

Other musical features are currently ignored here, yet are important in characterizing a performance. In particular, pianists do not always play left- and right-hand notes together, according to aural traditions, although they are written as simultaneities in the printed score. Articulations such as legato and staccato are also important performance features but are equally difficult to extract reliably from audio data. Nonetheless, tempo and dynamic features are useful for developing navigational tools which allow listeners to focus their attention on specific areas for further analysis.

[2] `http://www.sonicvisualiser.org`
[3] `http://sv.mazurka.org.uk/download`
[4] `http://mazurka.org.uk/info/excel`
[5] The filtering method is available online at `http://mazurka.org.uk/software/online/smoother` .

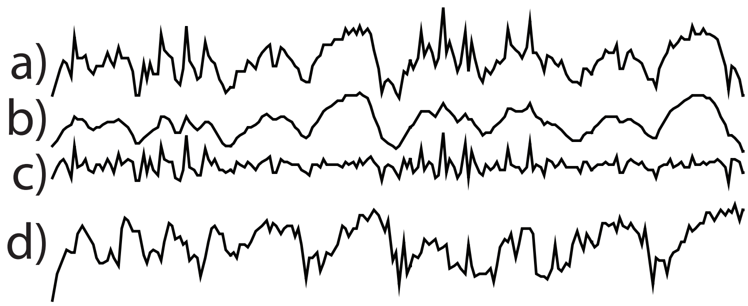


Figure 6.1: Extracted musical features from a recording of Chopin's mazurka in B minor, op. 30/2: a) tempo between beats; b) smoothed tempo; c) residual tempo ($c = a - b$); and d) beat-level dynamics.

6.2 Derivations and Definitions

Starting with the underlying comparison method of correlation (called S_0 below), a series of intermediate similarity measurements (S_1, S_2, and S_3) are used to derive a final measurement technique (S_4). Section 3 then compares the effectiveness of S_0 and S_4 measurements in identifying recordings of the same performer out of a database of recordings of the same mazurka.

6.2.1 Type-0 Score

As a starting point for comparison between performance features, Pearson correlation, often called an r-value in statistics, is used:

$$\text{Pearson}(x,y) = \frac{\sum_n (x_n - \bar{x})(y_n - \bar{y})}{\sqrt{\sum_n (x_n - \bar{x})^2 \sum_n (y_n - \bar{y})^2}} \tag{6.1}$$

This type of correlation is related to dot-product correlation used in Fourier analysis, for example, to measure similarities between an audio signal and a set of harmonically related sinusoids. The value range for Pearson correlation is -1.0 to $+1.0$, with 1.0 indicating an identical match between two sequences (exclusive of scaling and shifting), and the value 0.0 indicating no predictable linear relation between the two sequences x and y.

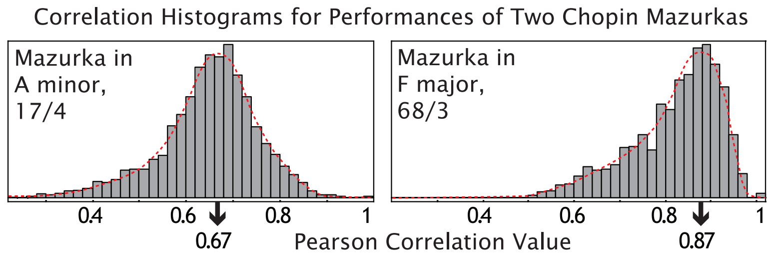


Figure 6.2: Different compositions will have different expected correlation distributions between performances.

Correlation values between extracted musical features typically have a range between 0.20 and 0.97 for different performances of mazurkas. Figure 6.2 illustrates the range of correlations between performances in two mazurkas. Mazurka op. 17/4 is a more complex composition with a more varied interpretation range, so the *mode*, or most-expected value, of the correlation distribution is 0.67. Mazurka op. 68/3 is a simpler composition with fewer options for individual interpretations so the mode is much higher at 0.87.

These differences in expected correlation values between two randomly selected performances illustrate a difficulty in interpreting similarity directly from correlation values. The correlation values are consistent only in relation to a particular composition, and these absolute values cannot be compared directly between different mazurkas. For example, a pair of performances which correlate at 0.80 in mazurka op. 17/4 indicates a better than average match, while the same correlation value in mazurka 68/3 would be a relatively poor match. In addition, correlations at different timescales in the same piece will have a similar problem, since some regions of music may allow for a freer interpretation while other regions may have a more static interpretation.

6.2.2 Type-1 Score

In order to compensate partially for this variability in correlation distributions, scapeplots were developed which only display nearest-neighbor performances in terms of correlation at all timescales for a particular reference performance. [3] Examples of such plots created for the Mazurka Project can be viewed online.[6]

[6] `http://mazurka.org.uk/ana/pcor-perf`

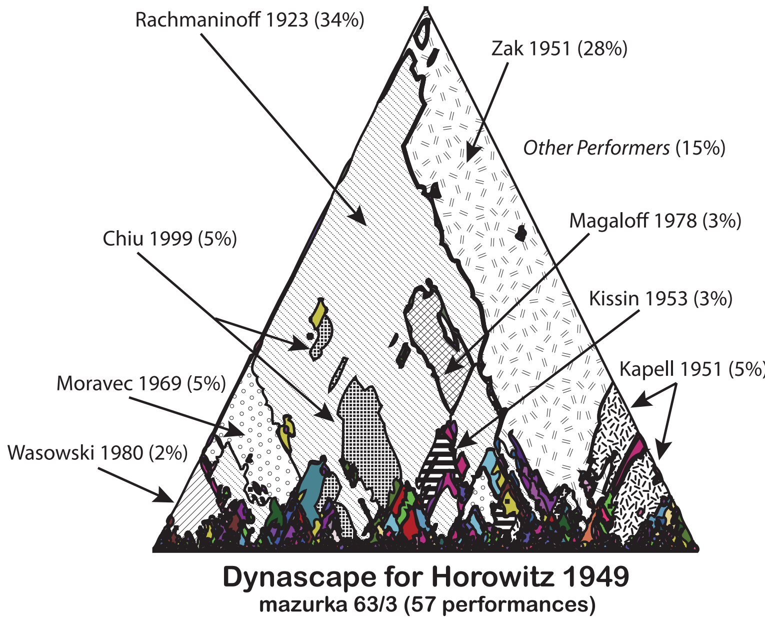


Figure 6.3: Scapeplot for the dynamics in Horowitz's 1949 performance of mazurka, op. 63/3, with the top eight matching performances labeled.

The S_1 score is defined as the fraction of area each target performance covers in a reference performer's scapeplot. Figure 6.3 demonstrates one of these plots, comparing the dynamics of a performance by Vladimir Horowitz to 56 other recordings. In this case, Rachmaninoff's performance of the same piece matches better than any other performance, since his performance covers 34% of the scape's plotting domain. At second best, Zak's performance matches well towards the end of the music, but covers only 28% of the total plotting domain. Note that Zak's performance has the best correlation for the entire feature sequence (S_0 score) which is represented by the point at the top of the triangle. The S_1 scores for the top eight matches in Figure 6.3 are listed in Table 6.2. There is general agreement between S_1 and S_0 scores since five top-level correlation matches also appear in the list.

6.2.3 Type-2 Score

Scape displays are sensitive to the *Hatto effect*: if an identical performance to the reference, or query, performance is present in the target set of performances, then correlation values

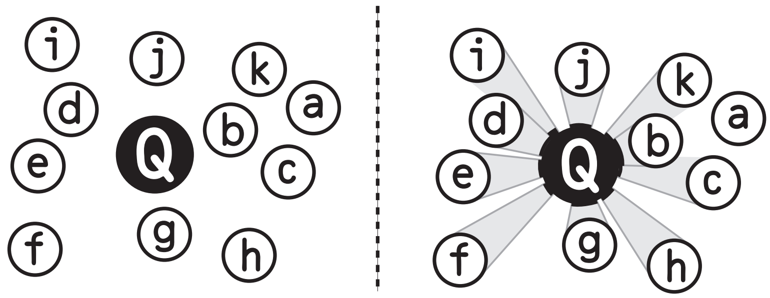


Figure 6.4: Schematic of nearest-neighbor matching method used in timescapes.

at all time resolutions will be close to the maximum value for the identical performance, and the comparative scapeplot will show a solid color. All other performances would have an S_1 score of approximately 0 regardless of how similar they might otherwise seem to the reference performance. This property of S_1 scores is useful for identifying two identical recordings, but not useful for viewing similarities to other performances which are hidden behind such closely neighboring performances.

One way to compensate for this problem is to remove the best match from the scape plot in order to calculate the next best match. For example, Figure 6.4 gives a rough schematic for how scapeplots are generated. Capital 'Q' represents the query, or reference, performance, and lower-case lettered points represent other performances. The scapeplot basically looks around in the local neighborhood of the feature space and displays closest matches as indicated by lines drawn towards the query on the right side of Figure 6.4. Closer performances will tend to have larger shadows on the query, and some performances can be completely blocked by others, as is the case for point "*a*" in the illustration.

An S_2 score measures the coverage area of the most dominant performance which is assumed to be most similar to the reference performance. This nearest of the neighbors is then removed from the search database, and a new scapeplot is generated with the remaining performances. Gradually, more and more performances will be removed which allows for previously hidden performances to appear in the plot. For example, point "*a*" in Figure 6.4 will start to become visible once point "*b*" is removed. S_2 scores and ranks are independent. As fewer and fewer target matches remain, the S_2 scores will increase towards 1.0 while the S_2 ranks decrease towards the bottom match.

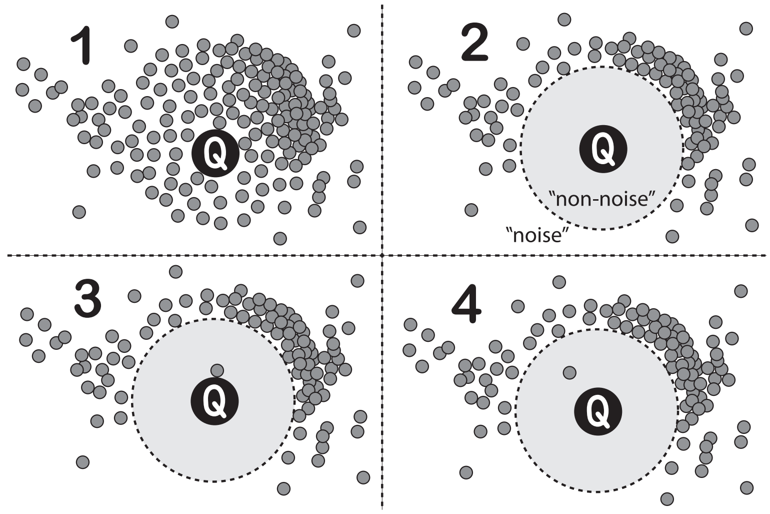


Figure 6.5: Schematic of the steps for measuring an S_3 score: (1) sort performances from similar to dissimilar, (2) remove most similar performance to leave noise floor, (3) & (4) insert more similar performances one-by-one to observe how well they can occlude the noise-floor.

In Figure 6.3, Rachmaninoff is initially the best match in terms of S_1 scores, so his performance will be removed and a new scapeplot generated. When this is done, Zak's performance then represents the best match, covering 34% of the scapeplot. Zak's performance is then removed, a new scapeplot is calculated, and Moravec's performance will have the best coverage at 13%, and so on. Some of the top S_2 scores for Horowitz's performance are listed in Table 6.2.

6.2.4 Type-3 Score

Continuing on, the next best S_2 rank is for Chiu, who has 20% coverage. Notice that this is greater than Moravec's score of 13%. This demonstrates the occurrence of what might be called the lesser Hatto effect: much of Moravec's performance overlapped onto Chiu's region, so when looking only at the nearest neighbors in this manner, there are still overlap problems. Undoubtedly, Rachmaninoff's and Zak's performances mutually overlap each other in Figure 6.3 as well. Both of them are good matches to Horowitz, so it is difficult to determine accurately which performance matches “best” according to S_2 scores since they are interfering with each others scores and are both tied at 34% coverage.

In order to define a more equitable similarity metric and remove the Hatto effect completely, all performances are first ranked approximately by similarity to Horowitz using either S_0 values or the rankings produced during S_2 score calculations. Performances are then divided into two groups, with the poorly matching half defined as the performance “noise-floor” over which better matches will be individually placed.

To generate an S_3 score, the non-noise performances are removed from the search database as illustrated in step 2 of Figure 6.5, leaving only background-noise performances. Next, non-noise performances are re-introduced separately along with all of the noise-floor performances and scapeplot is generated. The coverage area of the single non-noise performance represented in the plot is defined as its S_3 similarity measurement with respect to the query performance.

This definition of a performance noise-floor is somewhat arbitrary but splitting the performance database into two equal halves seems the most flexible rule to use, and is used for the evaluation section later in this paper. But the cut-off point could be a different percentage, such as the bottom 75% of ranked scores, or an absolute cut-off number. In any case, it is preferable that the noise floor does not appear to have any favored matches, and should consist of uniform small blotches at all timescales in the scapeplot representing many different performers as is the example shown in Figure 6.6 (top left part of the figure). While Rachmaninoff and Zak have equivalent S_2 scores, Rachmaninoff’s performance is able to cover 74% of the noise-floor, while Zak’s is only able to cover 64%.

6.2.5 Type-4 Score

Type-3 scores require one additional refinement in order to be useful since performances are not necessarily evenly distributed in the feature space. The plots used to calculate the S_3 scores are still nearest-neighbor rank plots, so the absolute numeric distances between performances are not directly displayed. Unlike correlation values between two performances, S_3 scores are not symmetric: the score from A to B is not the same value as from B to A. It is possible for an outlier performance to match well to another performance closer

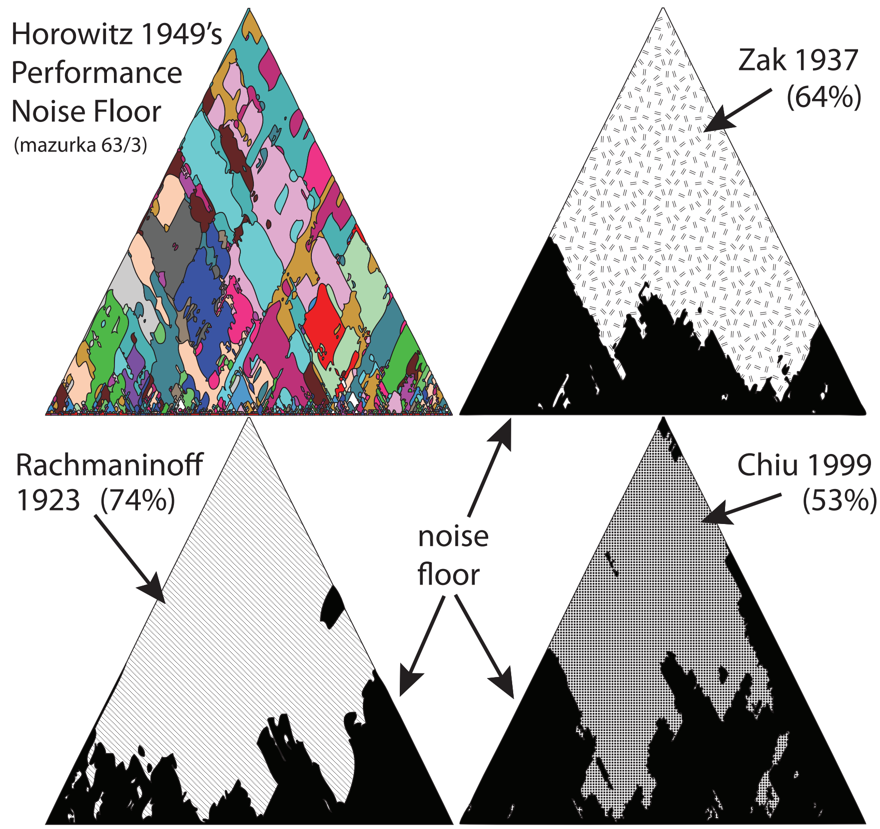


Figure 6.6: Dynascapes for Horowitz's performance of mazurka, op. 63/3. Top left is a plot of the noise-floor performances, and the other three plots separate include one of the top matching performances which can cover most of the noise-floor.

Target	S_0	R_0	S_1	S_2	S_3	S_{3r}	S_4	R_4
Rac23	0.60	3	0.34	0.34	0.74	0.82	0.78	1
Zak37	0.64	1	0.28	0.34	0.64	0.60	0.62	2
Mor69	0.59	4	0.05	0.13	0.57	0.54	0.55	3
Chi99	0.49	20	0.05	0.20	0.53	0.54	0.53	4
Kap51	0.51	17	0.05	0.08	0.24	0.17	0.20	22
Mag78	0.62	2	0.03	0.27	0.59	0.37	0.47	7
Kis93	0.52	15	0.03	0.09	0.44	0.23	0.32	11
Was80	0.58	5	0.02	0.11	0.41	0.55	0.47	6

Table 6.2: Scores and rankings for sample targets to Horowitz's 1949 performance of mazurka, op. 63/3.

to the average performance just because it happens to be facing towards the outlier, with the similarity just being a random coincidence.

Therefore, the geometric mean is used to mix the S_3 score with the reverse-query score (S_{3r}) as shown in Equation 6.4.

$$S_3 = A \Rightarrow B \text{ measurement} \tag{6.2}$$

$$S_{3r} = A \Leftarrow B \text{ measurement} \tag{6.3}$$

$$S_4 = \sqrt{S_3\, S_{3r}} \tag{6.4}$$

The arithmetic mean could also be used, but the geometric mean is useful since it penalizes the final score if the type-3 and its reverse scores are not close to each other. For example, the arithmetic mean between 0.75 and 0.25 is 0.50, while the geometric mean is lower at 0.43. Greatly differing S_3 and S_{3r} scores invariably indicate a poor match between two performances, with one of them acting as an outlier to a more central group of performances.

Table 6.2 shows several of the better matches to Horowitz's performance in mazurka, op. 63/3, along with the various types of scores that they generate. S_0 is the top-level correlation between the dynamics curves, and R_0 is the corresponding similarity rankings generated by sorting S_0 values. Likewise, S_4 and R_4 indicate the final proposed similarity metric, and the resulting rankings generated by sorting these scores.

Query	Target	Tempo			T_s			T_d			Dynamics			TD		
		R_0	R_3	R_4	R_0	R_3	R_4	R_0	R_3	R_4	R_0	R_3	R_4	R_0	R_3	R_4
Rub39	Rub52	2	3	2	3	8	8	3	2	1	6	8	8	3	3	2
Rub39	Rub66	1	1	1	1	1	1	2	3	2	4	4	2	1	2	1
Rub52	Rub39	6	9	2	27	31	12	3	3	2	28	23	6	12	9	2
Rub52	Rub66	2	2	1	3	2	1	2	2	1	2	2	1	2	2	1
Rub66	Rub39	3	4	2	2	3	1	3	6	5	15	8	6	5	5	3
Rub66	Rub52	1	2	1	3	2	2	2	2	1	1	3	2	1	2	1
Ranking Averages (by feature)	R_0	2.5			6.5			2.5			9.3			4.0		
	R_3		3.5			7.8			3.0			8.0			3.8	
	R_4			1.5			4.2			2.0			4.2			1.7
Overall Averages:		$R_0 = 4.97$						$R_3 = 5.32$						$R_4 = 2.70$		

Table 6.3: Rankings for mazurka op. 17/4 Rubinstein performances. Shaded numbers indicate perfect performance of a similarity metric.

6.3 Evaluation

When evaluating similarity measurement effectiveness, a useful technique with a clear ground-truth is to identify recordings by the same performer mingled among a larger collection of recordings. [5] Presumably pianists will tend to play more like their previous performances over time rather than like other pianists. If this is true, then better similarity metrics should match two performances by the same pianist more closely to each other than to other performances by different pianists.

6.3.1 Rubinstein performance matching

Arthur Rubinstein is perhaps the most prominent interpreter of Chopin's compositions in the 20th century, and luckily he has recorded the entire mazurka cycle three times during his career: (1) in 1938–9, aged 51; (2) in 1952–4, aged 66, and (3) in 1966, aged 79.

Table 6.3 lists the results of ranking his performances to each other in mazurka op. 17/4 where the search database contains an additional 60 performances besides the three by Rubinstein. The first column in the table indicates which performance was used as the query (Rubinstein's 1939 performance, for example, at the top of the first row). The *target* column indicates a particular target performance which is one of the other two performances by Rubinstein in the search database. Next, five columns list three types of rankings for comparison. The five columns represent four different extracted features as illustrated in

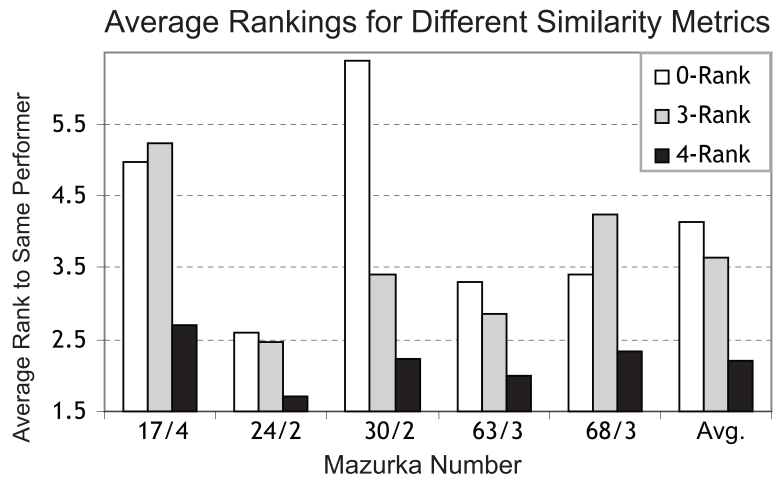


Figure 6.7: Ability of metrics to identify the same performer in a larger set of performances, using 3 performances of Rubinstein for each mazurka. (Lower rankings indicate better results.)

Figure 6.1, plus the *TD* column which represents a 50/50 admixture of the tempo and dynamics features.

For each musical feature, three columns of rankings are reported. R_0 represents the rankings from the S_0 scores; R_3 being the type-3 scoring ranks, and R_4 resulting from sorting the S_4 similarity values. In these columns, a "1" indicates that the target performance was ranked best in overall similarity to the query performance, "2" indicates that it is the second best match, and so on (see the search database sizes in Table 6.1). In the ranking table for mazurka op. 17/4 performances of Rubinstein, the shaded cells indicate perfect performance matches by a particular similarity metric where the top two matches are both Rubinstein. Note that there is one perfect pair of matches in all of the R_0 columns which is found in the full-tempo feature when Rubinstein 1939 is the target performance. No columns for R_3 contain perfect matching pairs, but about 1/2 of the R_4 columns contain perfect matches: all of the full-tempo R_4 rankings are perfect, and a majority of the desmoothed tempo and joint tempo/dynamics rankings are perfect. None of the metrics contain perfect matching pairs for the dynamics features. This is perhaps due to either (1) the dynamics data containing measurement noise (due to the difficultly of extracting dynamics data from audio data), or (2) Rubinstein varying his dynamics more over time than

Mazurka	*Query*	*Target*	*T* R_0	*T* R_4	T_s R_0	T_s R_4	T_d R_0	T_d R_4	*D* R_0	*D* R_4	*TD* R_0	*TD* R_4
17/4	Cze49	Cze49b	1	1	1	1	1	1	3	1	1	1
17/4	Cze49b	Cze49	1	1	1	1	1	1	7	1	1	1
63/3	Fri23	Fri30	1	1	1	1	1	1	1	1	1	1
63/3	Fri30	Fri23	1	1	1	1	1	1	1	1	1	1
17/4	Hor71	Hor85	2	1	13	1	2	1	1	1	2	1
17/4	Hor85	Hor71	1	1	1	1	1	1	1	1	1	1
30/2	Fou78	Fou05	2	1	1	1	13	17	2	2	2	1
30/2	Fou05	Fou78	1	1	1	1	2	6	3	2	2	1
63/3	Uni32	Uni71	1	1	1	1	5	1	1	1	1	1
63/3	Uni71	Uni32	1	1	1	1	1	1	1	1	1	1

Table 6.4: Performer self-matching statistics.

his tempo features, or a combination of these two possibilities.

Figure 6.7 shows the average rankings of Rubinstein performances for all extracted features averaged by mazurka. The figure shows S_4 scores are best at identifying the other two Rubinstein performances for all of the five mazurkas which were used in the evaluation. Typically S_4 gives three to four times better rankings than the S_0 values according to this figure. S_3 scores (used to calculate S_4 scores), are slightly better than plain correlation, but sometimes perform worse than correlation in some mazurkas.

Figure 6.8 evaluates the average ranking effectiveness by musical feature, averaged over all five mazurkas. Again S_4 scores are always three to four times more effective than plain correlation. S_3 scores are approximately as effective as S_0 rankings for full and smoothed tempo, but perform somewhat better on residual tempo and dynamics features, probably by minimizing the effects of sudden extreme differences between compared feature sequences caused by noisy feature data.

6.3.2 Other performers

Rubinstein tends to vary his performance interpretation more than most other pianists. Also, other performers may tend to emulate his performances, since he is one of the more prominent interpreters of Chopin's piano music. Thus, he is a difficult case to match and is a good challenge for similarity metric evaluations.

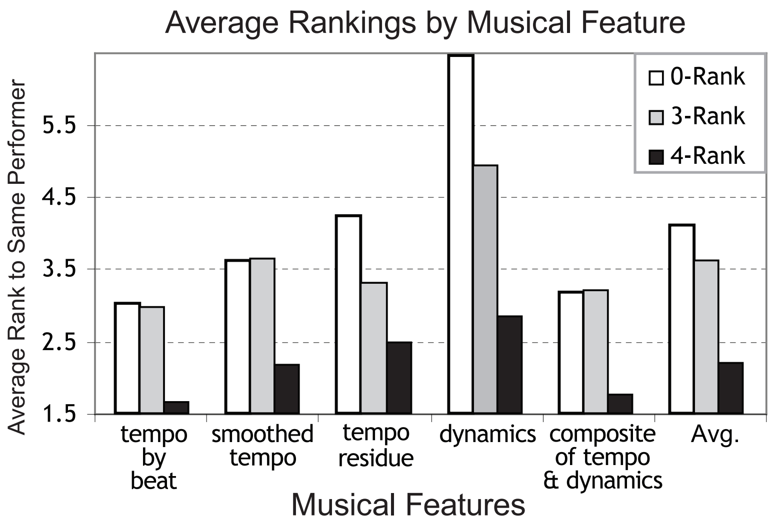


Figure 6.8: Ranking effectiveness by extracted musical features, using three performances of Rubinstein for each mazurka. (Lower values indicate better results.)

This section summarizes the effectiveness of the S_0 and S_4 similarity metrics in identifying other pianists found in the five selected mazurkas for which two recordings by the same pianist are represented in the data collection (only Rubinstein is represented by three performances for all mazurkas).

Table 6.4 presents ranking evaluations for performance pairs in a similar layout to those of Rubinstein found in Table 6.3. In all except two cases (for Fou) the S_4 metrics perform perfectly in identifying the other performance by the same pianist. Top-level correlation was able to generate correct matches in 75% of the cases. An interesting difference between the two metrics occurs when Hor71 is the query performance. In this case S_0 yields a rank of 13 (with 12 other performance matching better than his 1985 performance), while S_4 identifies his 1985 performance as the closest match.

Fou's performance pair for mazurka op. 30/2 is also an interesting case. For his performances, the phrasing portion of the full- and smoothed-tempo features match well to each other, but the tempo residue does not. This is due to a significant change in his metric interpretation: the earlier performance has a strong mazurka metric pattern which consists of a short first beat, followed by a lengthened second or third beat in each measure. His 2005 performance greatly reduces this effect, and beat durations are more uniform throughout the measure in comparison to his 1978 performance.

Finally, it is interesting to note the close similarity between Uninsky's pair of performances listed in Table 6.4. These two performances were recorded almost 40 years apart, one in Russia and the other in Texas. Also, the first was recorded onto 78 RPM monophonic records, while the later was recorded onto 33-1/3 RPM stereo records. Nonetheless, his two performances indicate a continuity of performance interpretation over a long career.

6.4 Application

As an example application of the derived similarity metrics, two performances of mazurka op. 30/2 performed by Alfred Cortot are examined in this section. One complete set of his mazurka performances can be found on commercially release recordings from a 1980's-era issue on cassette tape "recorded at diverse locations and dates presumably in the period of 1950–1952." [1] These recordings happen to be issued by the same record label as the recordings of Joyce Hatto, which casts suspicion on other recordings produced on that label. [4]. A peculiar problem is that no other commercial recordings exist of Cortot playing any mazurka, let alone the entire mazurka cycle.

In 2005, however, Sony Classical (S3K89698) released a 3-CD set of recordings by Cortot played during master classes he conducted during the late 1950's, and in this set, there are six partial performances of mazurkas by Cortot where he demonstrates how to play mazurkas to students during the class. His recording of mazurka op. 30/2 on these CDs is the largest continuous fragment, including 75% of the entire composition, stopping two phrases before the end of the composition.

Table 6.5 lists the S_4 scores and rankings for these two recordings of mazurka op. 30/2, with the Concert Artist's rankings on the left, and the Sony Classical rankings to the right. The five different musical features listed by column in previous tables for Rubinstein and other pianists are listed here by row. For each recording/feature combination, the top three matches are listed, along with the ranking for the complimentary Cortot recording.

Note that in all cases, the two Cortot recordings match very poorly to each other. In two

feature	Cortot ConA R4		S4	Cortot Sony R4		S4
T	1.	Rangell 2001	0.65	1.	Luisada 1990	0.40
	2.	Uninsky 1971	0.47	2.	Ferenczy 1956	0.31
	3.	Yaroshinsky 2005	0.44	3.	Rubinstein 1966	0.27
	...			...		
	31.	Cortot Sony	0.04	33.	Cortot ConA	0.04
T_s	1.	Pobłocka 1999	0.60	1.	Lushtak 2004	0.27
	2.	Luisada 1990	0.60	2.	Milkina 1970	0.18
	3.	Mohovich 1999	0.56	3.	Fou 2005	0.18
	...			...		
	26.	Cortot Sony	0.07	24.	Cortot ConA	0.07
T_d	1.	Rangell 2001	0.79	1.	Luisada 1990	0.53
	2.	Uninsky 1971	0.72	2.	Rubinstein 1966	0.49
	3.	Yaroshinsky 2005	0.57	3.	Rubinstein 1939	0.47
	...			...		
	32.	Cortot Sony	0.03	35.	Cortot ConA	0.03
D	1.	Brailowsky 1960	0.38	1.	Fliere 1977	0.46
	2.	Biret 1990	0.25	2.	Sofronitsky 1960	0.42
	3.	Milkina 1970	0.24	3.	Ashkenazy 1981	0.40
	...			...		
	33.	Cortot Sony	0.06	32.	Cortot ConA	0.06
TD	1.	Rangell 2001	0.40	1.	Avg. performance	0.16
	2.	Pobłocka 1999	0.37	2.	Indjic 1988	0.16
	3.	Yaroshinsky 2005	0.35	3.	Rubinstein 1952	0.15
	...			...		
	35.	Cortot Sony	0.03	34.	Cortot ConA	0.03

Table 6.5: Comparison of Cortot performances of mazurka op. 30/2 (m. 1–48).

cases, the worst possible ranking of 35 is achieved (since 36 performances are being compared in total). Perhaps Cortot greatly changes his performances style in the span of 6 years between these two recordings late in his life, although data from Tables 6.3 and 6.4 would not support this view since no other pianists has significantly alter all musical features at once, and only Fou significantly changes one musical feature between performances.

Therefore, it is likely that this particular mazurka recording on the Concert Artist label was not actually performed by Cortot. Results from further investigation of the other five partial mazurka performances on the Sony Classical recordings would help to confirm or refute this hypothesis, but the other examples are more fragmentary, making it difficult to extract reasonable amounts of recital-grade performance material. In addition, no performer in the top matches for the Concert Artist Cortot performance match well enough to have likely recorded this performance, so it is unlikely that any of the other 30 or so performers being compared to this spurious Cortot performance is the actual source for this particular mazurka recording.

6.5 Future Work

Different methods of combining S_3 and $S_{3\mathrm{r}}$ scores, such as measuring the intersection between plot areas rather than measuring the geometric mean to calculate S_4 should be examined. When the S_3 and $S_{3\mathrm{r}}$ scores are very different, this usually indicates that one performance (with the higher S_3 score) is an outlier performance. This performance is therefore more likely to be more similar to another more centrally grouped performance at random, so perhaps even taking the minimum of the S_3 and $S_{3\mathrm{r}}$ scores is a valid choice. The concept of a noise floor when comparing multiple performances is useful for identifying features which are common or rare, and allows similarity measurements to be more consistent across different compositions, which may aid in the identification of pianists across *different* musical works. [6]

Further analysis of the layout of the noise-floor as seen in Figure 6.6 might be useful in differentiating between directly and indirectly generated similarities between performances. For example in this figure, Rachmaninoff's performance shows more consistent similarity towards smaller-scale features, which may indicate a direct influence on Horowitz's performance style. Zak's noise-floor boundary in Figure 6.6 may demonstrate an indirect similarity, such as a general school of performance.

Since the similarity measurement described in this paper works well for matching the same performer in different recordings, and examination of student/teacher similarities may be done. The analysis techniques described here should be applicable to other types of features, and may be useful with other underlying similarity metrics besides correlation. For example, It would be interesting to extract musical features first from the data with other techniques such as Principle Component Analysis [2] and use this derived feature data for characterizing the similarities between performances in place of correlation.

References

[1] *Chopin: The Mazurkas. Alfred Cortot, piano.* Concert Artist compact disc CACD 91802, 2005.

[2] Bruno H. Repp. A microcosm of musical expression: I. Quantitative analysis of piansts' timing in the initial measures of Chopin's Etude in E major. *Journal of the Acoustical Society of America*, 104:1085–1100, 1998.

[3] Craig Sapp. Comparative analysis of multiple musical performances. *Proceedings of the 8th International Conference on Music Information Retrieval*, pages 497–500, 2007.

[4] Mark Singer. Fantasia for piano: Joyce Hatto's incredible career. *New Yorker*, pages 66–81, 17 Sep. 2007.

[5] Efstathios Stamatatos and G. Widmer. Automatic identification of music performers with learning ensembles. *Artificial Intelligence*, 165(1):37–56, 2005.

[6] Gerhard Widmer, S. Dixon, W. Goebl, E. Pampalk, and A. Tobudic. In search of the Horowitz factor. *AI Magazine*, 24(3):111–130, 2003.