Key Profile Optimisation for the Computational Modelling of Tonal Centre

by

Hendrik Johannes Vermeulen

Thesis presented in partial fulfilment of the requirements for the degree Master of Philosophy in Music Technology at Stellenbosch University

Supervisor: Mr. Theo Herbst
Faculty of Arts and Social Sciences

December 2012

Declaration

By submitting this thesis/dissertation electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

HJ Vermeulen

December 2012

Acknowledgements

The author gratefully acknowledges the contributions of the following individuals and institutions:

- My study leader, Theo Herbst, for introducing me to music technology, for unwavering encouragement and support throughout this project and for inspirational contributions to the academic content of the research.
- My wife Gene and daughters Vicki-Mari and Liezl-Mari for their encouragement, support and understanding. Also for proofreading of the final document.
- The members of the Measurement and Verification team and the post-graduate Energy Management laboratory at Stellenbosch University for their generous encouragement, support and friendship.
- Stellenbosch University for providing excellent research facilities and support services.

Table of Contents

1 Project motivation and project description 21
1.1 Introduction 21
1.2 Project motivation 21
1.3 Project description 25
1.3.1 Research objectives 25
1.3.2 Research methodology 25
1.4 Thesis structure 27
2 Literature review 28
2.1 Introduction 28
2.2 The physical and mathematical foundations of tone and timbre 29
2.2.1 Introduction 29
2.2.2 Mathematical properties of periodic waveforms 29
2.2.3 Frequency domain properties of musical instruments 31
2.3 Musical scales, tuning and temperament 31
2.3.1 Introduction 31
2.3.2 The Pythagorean scale 32
2.3.3 Just intonation scales 34
2.3.4 The harmonic series scale 35
2.3.5 Tempered scales 36
2.3.5.1 Introduction 36
2.3.5.2 Mean-tone temperament 37
2.3.5.3 Well temperament 37
2.3.5.4 Equal temperament 39
2.4 Experimental investigations of tonality cognition 40
2.4.1 Overview 40
2.4.2 The tonal hierarchy 40
2.4.3 Tonal closure 42
2.4.4 Tonal relationships in musical context 42
2.4.5 Tonal centre in the context of chords 43
2.4.6 Two-tone sequences in tonal context 44
2.5 Durational accent in music perception analysis 44
2.6 Pitch perception 47

2.7 Key-finding algorithms 47
2.7.1 Introduction 47
2.7.2 The shape-matching key-finding algorithm proposed by Longuet-Higgins and Steedman 48
2.7.3 The template-based key-finding algorithm proposed by Krumhansl and Schmuckler 49
2.7.4 The modified Krumhansl and Schmuckler key-finding algorithms proposed by Temperley 52
2.7.5 The modified Krumhansl and Schmuckler algorithm proposed by Huron and Parncutt 54
2.7.6 The preference rule key-finding algorithm proposed by Temperley 55
2.8 Geometrical models of tonal relationships 57
2.8.1 Introduction 57
2.8.2 Geometrical models based on music theory 58
2.8.2.1 Introduction 58
2.8.2.2 The Tonnetz 59
2.8.2.3 The charts of regions proposed by Schoenberg 60
2.8.2.4 The melodic map 61
2.8.2.5 The harmonic map 62
2.8.3 Geometrical models based on tonality cognition experiments 62
2.8.3.1 Introduction 62
2.8.3.2 Multidimensional scaling map proposed by Krumhansl and Kessler 63
2.9 The MIDI toolbox 63
2.10 The standard MIDI file format 64
2.10.1 Introduction 64
2.10.2 File structures 64
2.10.3 Chunk structures 65
2.10.3.1 Header chunk structure 65
2.10.3.2 Track chunk structure 66
2.10.4 Meta events 67
2.10.4.1 Key signature meta event 67
2.10.5 Time signature and tempo meta events 67
2.10.6 Note events 68
2.10.7 Suitability of the standard MIDI file format for music notation and analysis 69
2.11 Direct search methods for parameter estimation 70
2.11.1 Overview 70
2.11.2 Direct search methods 70
3 Mathematical Formulation of Computational Music Analysis Algorithms 72

3.1 Overview 72
3.2 Statistical properties 72
3.2.1 Introduction 72
3.2.2 Statistical representations of note durations 73
3.2.2.1 Introduction 73
3.2.2.2 Distribution of note durations 74
3.2.2.3 Distribution of dyad note durations 75
3.2.3 Statistical representations of pitch intervals 76
3.2.3.1 Introduction 76
3.2.3.2 Distribution of pitch intervals 77
3.2.3.3 Distribution of pitch interval sizes 80
3.2.3.4 Distribution of pitch interval directions 81
3.2.3.5 Distribution of dyad pitch intervals 82
3.2.4 Statistical representations of pitch classes 83
3.2.4.1 Introduction 83
3.2.4.2 Distribution of pitch classes 84
3.2.4.3 Distribution of dyad pitch classes 86
3.3 Tonality analysis 87
3.3.1 Introduction 87
3.3.1.1 Implementation of the Krumhansl and Schmuckler key-finding algorithm 87
4 Program implementation and validation 91
4.1 Overview 91
4.2 Unified Modelling Language 92
4.3 Program structure 92
4.3.1 Overview 92
4.3.2 Program environmental options 92
4.3.2.2 File path parameters and user options 93
4.3.2.3 Graphical object parameters and user options 94
4.3.3 Score definition 95
4.3.4 Score visualisation and auditioning 97
4.3.5 Statistical analysis 98
4.3.5.1 Overview 98
4.3.5.2 Distributions of note durations parameters and user options 98
4.3.5.3 Distributions of pitch intervals parameters and user options 99

4.3.5.4 Distributions of pitch classes parameters and user options 100
4.3.6 Key-finding analysis 101
4.4 Implementation of the score selection and partitioning strategy 103
4.5 Implementation of the direct search key profile estimation routine 105
4.5.1 Overview 105
4.5.2 Algorithm topology 106
4.5.3 Cost function definitions 107
4.5.3.1 Introduction 107
4.5.3.2 Cost function A 108
4.5.3.3 Cost function B 108
4.5.3.4 Cost function C 109
4.6 Program validation 110
4.6.1 Overview 110
4.6.2 Statistical analysis of note durations 110
4.6.2.1 Distribution of note durations 110
4.6.2.2 Distribution of dyad note durations 111
4.6.3 Statistical analysis of pitch intervals 112
4.6.3.1 Distribution of pitch intervals 112
4.6.3.2 Distribution of pitch interval sizes 115
4.6.3.3 Distribution of pitch intervals directions 116
4.6.3.4 Distribution of dyad pitch intervals 117
4.6.4 Statistical analysis of pitch classes 119
4.6.4.1 Distribution of pitch classes 119
4.6.4.2 Distribution of dyad pitch classes 122
4.6.5 Tonality analysis 124
4.6.5.1 Implementation of the Krumhansl and Schmuckler and Temperley key-finding algorithms 124
5 Case studies and results 132
5.1 Introduction 132
5.2 Statistical analysis 132
5.2.1 Overview 132
5.2.2 Preludes of Bach's Well-tempered Clavier Book I 133
5.2.2.1 Distributions of note durations 133
5.2.2.2 Distributions of pitch classes 133

5.2.3 Courante from Bach's Cello Suite in C major 134
5.2.3.1 Distributions of note durations 134
5.2.3.2 Distributions of pitch intervals 135
5.2.3.3 Distributions of pitch classes 137
5.2.4 Gavotte from Bach's French Suite No. 5 in G major (BWV 816) 139
5.2.4.1 Distributions of note durations 139
5.2.4.2 Distributions of pitch classes 139
5.3 Performance evaluation of the Krumhansl and Schmuckler key-finding algorithm 140
5.3.1 Overview 140
5.3.2 Key-finding case study results 142
5.3.3 Key profile optimization using direct search methods 148
5.3.3.1 Overview 148
5.3.3.2 Key profile estimation results for the full scores of the 24 preludes of Bach's Well-tempered Clavier Book I 151
5.3.4 Key estimation results for the first four notes of the 24 preludes of Bach's Well-tempered Clavier Book I 155
5.3.5 Key estimation results for the Courante from Bach's Cello Suite in C major 160
5.3.6 Key estimation results for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816) 165
5.3.6.2 Performance evaluation of the optimised key profiles 169
5.3.7 Pitch class hierarchy properties of the optimised key profiles 175
6 Conclusions and recommendations 179
6.1 Conclusions 179
6.1.1 Overview 179
6.1.1.1 Development of a Matlab application with a Graphical User Interface (GUI) for analysing the symbolic music 179
6.1.1.2 Investigating the optimal combination of pitch class representation and key profile template for key-finding algorithms 180
6.1.1.3 Deriving a key profile template using direct search methods 181
6.1.2 Recommendations 183

Appendix A 190
Musical excerpts and pitch strings used by Brown 190
Appendix B 192
Key-finding performance of the Krumhansl and Schmuckler algorithm for the first four notes of the forty-eight preludes of the Well-Tempered Clavier composed by Bach 192

Appendix C 193

Key-finding performance of the Krumhansl and Schmuckler algorithm for the first four notes of the twenty-four preludes composed by Shostakovich 193

Appendix D 194

Key-finding performance of the Krumhansl and Schmuckler algorithm for the first four notes of the twenty-four preludes composed by Chopin 194

Appendix E 195

Comparison of the key-finding performance of the Krumhansl and Schmuckler and the Longuet-Higgins and Steedman algorithms for the forty-eight fugues of the Well-Tempered Clavier composed by Bach 195

Appendix F 196

Key-finding performance of the Krumhansl and Schmuckler algorithm for the twenty-four fugues composed by Shostakovich 196

Appendix G 197

Key modulation tracking using the Temperley algorithm for the Gavotte from Bach's French Suite No. 5 in G major *(BWV 816)* 197

Appendix H 198

Overview of Unified Modelling Language 198

Appendix I 200

Supplementary statistical analysis results for the test corpus 200

Appendix J 206

Key-finding results of the Krumhansl and Schmuckler algorithm 206

Appendix K 211

Pitch class coefficients for the Krumhansl and Kessler, Temperley and optimised key profiles 211

List of Figures

Fig. 2.1 Time-domain representation of a sinusoidal waveform. ... 30
Fig. 2.2 Line spectrum representation of a periodic waveform. ... 31
Fig. 2.3 Circle of fifths representation of Pythagorean tuning. ... 33
Fig. 2.4 Intervallic relationships defined by the first eight harmonic components of the harmonic series (Rameau, 1971). ... 36
Fig. 2.5 Circle of fifths representations of quarter-comma and sixth-comma mean-tone temperaments (Adapted from Sethares, 2005. ... 37
Fig. 2.6 Circle of fifths representation of Rameau's well temperament (Rameau, 1971). ... 38
Fig. 2.7 Circle of fifths representations of Werckmeister III and Werckmeister VI well temperaments (Adapted from Sethares, 2005). ... 38
Fig. 2.8 Circle of fifths representation of Kirnberger III well temperament (Adapted from Sethares, 2005). ... 38
Fig. 2.9 Circle of fifths representation of equal temperament tuning. ... 39
Fig. 2.10 Major and minor mode key profile coefficients proposed by Krumhansl and Kessler (Krumhansl, 1990) ... 41
Fig. 2.11 Major and minor key psychological key profiles proposed by Krumhansl and Kessler (Krumhansl, 1990). ... 42
Fig. 2.12 Durational accent as a function of saturation duration for a fixed accent index of 2. ... 46
Fig. 2.13 Durational accent as a function of accent index for a fixed saturation duration of 0.5. ... 46
Fig. 2.14 Major and minor shapes proposed by Longuet-Higgins and Steedman (1971). ... 49
Fig. 2.15 Major and minor mode key profile coefficients proposed by Temperley (1999). ... 52
Fig. 2.16 Comparison of the Krumhansl and Kessler and the Temperley key profiles for the major mode. ... 53
Fig. 2.17 Comparison of the Krumhansl and Kessler and the Temperley key profiles for the minor mode. ... 53
Fig. 2.18 Circle of fifths pitch spelling model (Temperley, 1999;2002) ... 55
Fig. 2.19 Line of fifths pitch spelling model (Temperley, 1999; 2002). ... 56
Fig. 2.20 Tonal pitch class key profiles proposed by Temperley (1999; 2002). ... 56
Fig. 2.21 Circle of fifths representation of consonant and dissonant intervals. ... 58
Fig. 2.22 Tonal substructure and intervallic relationships contained in the Tonnetz. ... 59
Fig. 2.23 Example of major and minor key triad relationships expressed in the Tonnetz. ... 60
Fig. 2.24 Full Tonnetz. ... 60
Fig. 2.25 Schoenberg's charts of major and minor key regions illustrated for C major and A minor respectively (Schoenberg, 1969; Krumhansl, 1990) ... 61
Fig. 2.26 The melodic map proposed by Lakner (1960). ... 61
Fig. 2.27 The harmonic map proposed by Shepard (1982a). ... 62
Fig. 2.28 Multidimensional scaling map proposed by Krumhansl and Kessler (1982). ... 63
Fig. 2.29 Standard MIDI file structures. ... 65
Fig. 2.30 Standard MIDI file chunk structures. ... 65
Fig. 2.31 Maximum permissible delta-time represented as a four byte variable length value. ... 67
Fig. 3.1 Hierarchy of statistical analysis options addressed in the software implementation. ... 72
Fig. 3.2 Standard note durations in beats. ... 73
Fig. 3.3 Standard note durations and bin centres on a base 2 logarithmic scale. ... 73
Fig. 4.1 High-level topology of the environmental parameters and user options data structure. ... 93
Fig. 4.2 Topology of the default file path parameters and user options data structure. ... 93
Fig. 4.3 Topology of the graphic objects parameters and user options data structure. ... 94
Fig. 4.4 Topology of the main score suite data structure. ... 96
Fig. 4.5 Topology of the score array data structure. ... 97
Fig. 4.6 Topology of the MIDI sequence data structure. ... 98
Fig. 4.7 Topology of the parameters and user options data structure for the distributions of note durations. ... 98
Fig. 4.8 Topology of the parameters and user options data structure for the distributions of pitch intervals. ... 99
Fig. 4.9 Topology of the parameters and user options data structure for the distributions of pitch classes. ... 100
Fig. 4.10 High-level topology of the key-finding algorithm parameters and user options data structure. ... 101

Fig. 4.11 *Topology of the parameters and user options data structure for the distribution of pitch classes applied in the key-finding algorithm.* *102*
Fig. 4.12 *Topology of the parameters and user options data structure for the key-finding algorithm.* *103*
Fig. 4.13 *Score partitioning using the single window approach.* *104*
Fig. 4.14 *Score partitioning using the sliding window approach.* *104*
Fig. 4.15 *Score partitioning using the expanding window approach.* *104*
Fig. 4.16 *Topology of the parameters and user options data structure for the partitioning window.* *105*
Fig. 4.17 *Flow diagram of the key profile estimation algorithm.* *107*
Fig. 4.18 *Test score for the statistical analysis of note durations.* *110*
Fig. 4.19 *Distribution of note durations for the test score shown in Fig. 4.18.* *111*
Fig. 4.20 *Normalised distribution of note durations for the test score shown in Fig. 4.18.* *111*
Fig. 4.21 *Distribution of dyad note durations for the test score shown in Fig. 4.18.* *112*
Fig. 4.22 *Normalised distribution of dyad note durations for the test score shown in Fig. 4.18.* *112*
Fig. 4.23 *Test score for statistical analysis of pitch intervals.* *113*
Fig. 4.24 *Unweighted distribution of pitch intervals for the test score shown in Fig. 4.23.* *113*
Fig. 4.25 *Normalised unweighted distribution of pitch intervals for the test score shown in Fig. 4.23.* *113*
Fig. 4.26 *Weighted distribution of pitch intervals for the test score shown in Fig. 4.23.* *114*
Fig. 4.27 *Normalised weighted distribution of pitch intervals for the test score shown in Fig. 4.23.* *114*
Fig. 4.28 *Unweighted distribution of pitch interval sizes for the test score shown in Fig. 4.23.* *115*
Fig. 4.29 *Normalised unweighted distribution of pitch interval sizes for the test score shown in Fig. 4.23.* *115*
Fig. 4.30 *Weighted distribution of pitch interval sizes for the test score shown in Fig. 4.23.* *116*
Fig. 4.31 *Normalised weighted distribution of pitch interval sizes for the test score shown in Fig. 4.23.* *116*
Fig. 4.32 *Unweighted distribution of pitch interval directions for the test score shown in Fig. 4.23.* *117*
Fig. 4.33 *Weighted distribution of pitch interval directions for the test score shown in Fig. 4.23.* *117*
Fig. 4.34 *Unweighted distribution of dyad pitch intervals for the test score shown in Fig. 4.23.* *118*
Fig. 4.35 *Normalised unweighted distribution of dyad pitch intervals for the test score shown in Fig. 4.23.* *118*
Fig. 4.36 *Weighted distribution of dyad pitch intervals for the test score shown in Fig. 4.23.* *119*
Fig. 4.37 *Normalised weighted distribution of dyad pitch intervals for the test score shown in Fig. 4.23.* *119*
Fig. 4.38 *Test score for statistical analysis of pitch classes.* *120*
Fig. 4.39 *Unweighted distribution of pitch classes for the test score shown in Fig. 4.38.* *120*
Fig. 4.40 *Normalised unweighted distribution of pitch classes for the test score shown in Fig. 4.38.* *121*
Fig. 4.41 *Weighted distribution of pitch classes for the test score shown in Fig. 4.38.* *121*
Fig. 4.42 *Normalised weighted distribution of pitch classes for the test score shown in Fig. 4.38.* *122*
Fig. 4.43 *Unweighted distribution of dyad pitch classes for the test score shown in Fig. 4.38.* *123*
Fig. 4.44 *Normalised unweighted distribution of dyad pitch classes for the test score shown in Fig. 4.38.* *123*
Fig. 4.45 *Weighted distribution of dyad pitch classes for the test score shown in Fig. 4.38.* *124*
Fig. 4.46 *Normalised weighted distribution of dyad pitch classes for the test score shown in Fig. 4.38.* *124*
Fig. 4.47 *Score for the first half of the courante from Bach's cello suite in C major.* *125*
Fig. 4.48 *Key progression calculated with the Krumhansl and Schmuckler algorithm, using the Krumhansl and Kessler key profiles and Krumhansl key correlation formula, for measures 1 to 40 of the Courante from Bach's Cello Suite in C major.* *126*
Fig. 4.49 *Pitch class distribution calculated with the parameters listed in Table 4.13 for measure 39 of the Courante from Bach's Cello Suite in C major.* *127*
Fig. 4.50 *Key correlation coefficients calculated with the Krumhansl and Schmuckler algorithm and the parameters listed in Table 4.13 for measure 39 of the Courante from Bach's Cello Suite in C major.* *128*
Fig. 4.51 *First measure of "Yankee Doodle" (Temperley, 1999).* *128*
Fig. 4.52 *Key correlations calculated with the software application for the first measure of "Yankee Doodle" using the Krumhansl and Kessler key profiles and the Krumhansl key correlation formula.* *129*

Fig. 4.53 *Pitch class distribution calculated with the software application for the first measure of "Yankee Doodle" using linear durational weighing in beats.* 129

Fig. 4.54 *Key progression calculated with the Krumhansl and Schmuckler algorithm, using the Temperley key profiles and key correlation formula, for measures 1 to 40 of the Courante from Bach's Cello Suite in C.* 130

Fig. 4.55 *Key progression calculated with the Krumhansl and Schmuckler algorithm, using the key profiles, key correlation formula and flat pitch class vector proposed by Temperley, for measures 1 to 40 of the Courante from Bach's Cello Suite in C.* 131

Fig. 5.1 *Normalised distribution of note durations for the 24 preludes of Bach's Well-tempered Clavier Book I.* 133

Fig. 5.2 *Normalised unweighted distribution of pitch classes for the 24 preludes of Bach's Well-tempered Clavier Book I.* 133

Fig. 5.3 *Normalised distributions of pitch classes for the 24 preludes of Bach's Well-tempered Clavier Book I: Comparison of unweighted, linear durational weighing and durational accent weighing (* $\tau = 0.5 \text{ and } I_a = 2.0$*).* 134

Fig. 5.4 *Normalised distribution of note durations for the Courante from Bach's Cello Suite in C major.* 134

Fig. 5.5 *Normalised distribution of dyad note durations for the Courante from Bach's Cello Suite in C major.* 135

Fig. 5.6 *Normalised unweighted distribution of pitch intervals for the Courante from Bach's Cello Suite in C major.* 135

Fig. 5.7 *Normalised unweighted distribution of dyad pitch intervals for the Courante from Bach's Cello Suite in C major.* 136

Fig. 5.8 *Normalised unweighted distribution of pitch interval sizes for the Courante from Bach's Cello Suite in C major.* 136

Fig. 5.9 *Normalised unweighted distribution of pitch interval directions for the Courante from Bach's Cello Suite in C major.* 137

Fig. 5.10 *Normalised distributions of pitch intervals for the Courante from Bach's Cello Suite in C major: Comparison of unweighted, linear durational weighing and durational accent weighing (*$\tau = 0.5 \text{ and } I_a = 2.0$*).* 137

Fig. 5.11 *Normalised unweighted distribution of pitch classes for the Courante from Bach's Cello Suite in C major.* 138

Fig. 5.12 *Normalised unweighted distribution of dyad pitch classes for the Courante from Bach's Cello Suite in C major.* 138

Fig. 5.13 *Normalised distributions of pitch classes for the Courante from Bach's Cello Suite in C major: Comparison of unweighted, linear durational weighing and durational accent weighing (*$\tau = 0.5 \text{ and } I_a = 2.0$*).* 139

Fig. 5.14 *Normalised distribution of note durations for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816).* 139

Fig. 5.15 *Normalised unweighted distribution of pitch classes for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816).* 140

Fig. 5.16 *Normalised pitch class distributions for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816): Comparison of unweighted, linear durational weighing and durational accent weighing (* $\tau = 0.5 \text{ and } I_a = 2.0$*).* 140

Fig. 5.17 *Hierarchy of case studies conducted with the Krumhansl and Schmuckler key-finding algorithm.* 141

Fig. 5.18 *Comparison of the key-finding performance of the Krumhansl and Kessler key profile grouped according to the test score.* 144

Fig. 5.19 *Comparison of the key-finding performance of the Krumhansl and Kessler key profile grouped according to the weighing model.* 144

Fig. 5.20 *Comparison of the key-finding performance of the Temperley key profile grouped according to the test score.* 145

Fig. 5.21 *Comparison of the key-finding performance of the Temperley key profile grouped according to the weighing model.* 145

Fig. 5.22 *Key-finding performance of the flat pitch class weighing model for the Krumhansl and Kessler and the Temperley key templates for all test scores.* 146

Fig. 5.23 *Number of pitch classes as a function of window size in measures for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816). 146*
Fig. 5.24 *Key-finding performance of the histogram pitch class weighing model for the Krumhansl and Kessler and the Temperley key templates for all test scores. 147*
Fig. 5.25 *Key-finding performance of the linear durational pitch class weighing model for the Krumhansl and Kessler and the Temperley key templates for all test scores. 148*
Fig. 5.26 *Key-finding performance of the durational accent pitch class weighing model for the Krumhansl and Kessler and the Temperley key templates for all test scores. 148*
Fig. 5.27 *Hierarchy of case studies conducted for the key profile estimation investigation. 149*
Fig. 5.28 *Key profile estimation performances for a key resolution of 0.1% for the full scores of the 24 preludes of Bach's Well-tempered Clavier Book I. 152*
Fig. 5.29 *Key profile estimation performances for a key resolution of 1.0% for the full scores of the 24 preludes of Bach's Well-tempered Clavier Book I. 152*
Fig. 5.30 *Key profile estimation performances for a key resolution of 10% for the full scores of the 24 preludes of Bach's Well-tempered Clavier Book I. 152*
Fig. 5.31 *Best scoring unconstrained major mode key profiles with linear durational weighing for the full scores of the 24 preludes of Bach's Well-tempered Clavier Book I. 153*
Fig. 5.32 *Best scoring unconstrained minor mode key profiles with linear durational weighing for the full scores of the 24 preludes of Bach's Well-tempered Clavier Book I. 154*
Fig. 5.33 *Best scoring constrained major mode key profiles with linear durational weighing for the full scores of the 24 preludes of Bach's Well-tempered Clavier Book I. 154*
Fig. 5.34 *Best scoring constrained minor mode key profiles with linear durational weighing for the full scores of the 24 preludes of Bach's Well-tempered Clavier Book I. 155*
Fig. 5.35 *Key profile estimation performances for a key resolution of 0.1% for the first four notes of the 24 preludes of Bach's Well-tempered Clavier Book I. 156*
Fig. 5.36 *Key profile estimation performances for a key resolution of 1.0% for the first four notes of the 24 preludes of Bach's Well-tempered Clavier Book I. 157*
Fig. 5.37 *Key profile estimation performances for a key resolution of 10.0% for the first four notes of the 24 preludes of Bach's Well-tempered Clavier Book I. 157*
Fig. 5.38 *Best scoring unconstrained major mode key profiles with linear durational weighing for the first four notes of the 24 preludes of Bach's Well-tempered Clavier Book I. 158*
Fig. 5.39 *Best scoring unconstrained minor mode key profiles with linear durational weighing for the first four notes of the 24 preludes of Bach's Well-tempered Clavier Book I. 158*
Fig. 5.40 *Best scoring constrained major mode key profiles with linear durational weighing for the first four notes of the 24 preludes of Bach's Well-tempered Clavier Book I. 159*
Fig. 5.41 *Best scoring constrained minor mode key profiles with linear durational weighing for the first four notes of the 24 preludes of Bach's Well-tempered Clavier Book I. 159*
Fig. 5.42 *Key profile estimation performances for a key resolution of 0.1% for the Courante from Bach's Cello Suite in C major. 161*
Fig. 5.43 *Key profile estimation performances for a key resolution of 1.0% for the Courante from Bach's Cello Suite in C major. 161*
Fig. 5.44 *Key profile estimation performances for a key resolution of 10.0% for the Courante from Bach's Cello Suite in C. 162*
Fig. 5.45 *Best scoring unconstrained major mode key profiles with linear durational weighing for the Courante from Bach's Cello Suite in C major. 162*
Fig. 5.46 *Best scoring unconstrained minor mode key profiles with linear durational weighing for the Courante from Bach's Cello Suite in C major. 163*
Fig. 5.47 *Best scoring constrained majormode key profiles with linear durational weighing for the Courante from Bach's Cello Suite in C major. 164*

Fig. 5.48 *Best scoring constrained minor mode key profiles with linear durational weighing for the Courante from Bach's Cello Suite in C major.* 164
Fig. 5.49 *Key profile estimation performances for a key resolution of 0.1% for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816).* 166
Fig. 5.50 *Key profile estimation performances for a key resolution of 1.0% for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816).* 166
Fig. 5.51 *Key profile estimation performances for a key resolution of 10.0% for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816).* 167
Fig. 5.52 *Best scoring unconstrained major mode key profiles with linear durational weighing for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816).* 167
Fig. 5.53 *Best scoring unconstrained minor mode key profiles with linear durational weighing for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816).* 168
Fig. 5.54 *Best scoring constrained major mode key profiles with linear durational weighing for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816).* 168
Fig. 5.55 *Best scoring constrained minor mode key profiles with linear durational weighing for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816).* 169
Fig. 5.56 *Performances of the optimised, Krumhansl and Kessler and Temperley key profiles for a key resolution of 0.1% with the flat pitch class weighing model.* 170
Fig. 5.57 *Performances of the optimised, Krumhansl and Kessler and Temperley key profiles for a key resolution of 0.1% with the linear durational pitch class weighing model.* 171
Fig. 5.58 *Performances of the optimised, Krumhansl and Kessler and Temperley key profiles for a key resolution of 0.1% with the durational accent pitch class weighing model.* 171
Fig. 5.59 *Performances of the optimised, Krumhansl and Kessler and Temperley key profiles for a key resolution of 1.0% with the flat pitch class weighing model.* 172
Fig. 5.60 *Performances of the optimised, Krumhansl and Kessler and Temperley key profiles for a key resolution of 1.0% with the linear durational pitch class weighing model.* 172
Fig. 5.61 *Performances of the optimised, Krumhansl and Kessler and Temperley key profiles for a key resolution of 1.0% with the durational accent pitch class weighing model.* 173
Fig. 5.62 *Performances of the optimised, Krumhansl and Kessler and Temperley key profiles for a key resolution of 10.0% with the flat pitch class weighing model.* 174
Fig. 5.63 *Performances of the optimised, Krumhansl and Kessler and Temperley key profiles for a key resolution of 10.0% with the linear durational pitch class weighing model.* 174
Fig. 5.64 *Performances of the optimised, Krumhansl and Kessler and Temperley key profiles for a key resolution of 10.0% with the durational accent pitch class weighing model.* 175
Fig. 5.65 *Comparison of major mode key profiles: Krumhansl and Kessler, Temperley, unconstrained Gavotte with a key resolution of 1.0% and unconstrained Courante with a key resolution of 10.0%.* 176
Fig. 5.66 *Comparison of minor mode key profiles: Krumhansl and Kessler, Temperley, unconstrained Gavotte with a key resolution of 1.0% and unconstrained Courante with a key resolution of 10.0%.* 177

Fig. A1 *Musical excerpts and pitch strings used by Brown (1988).* 191
Fig. G1 *Key modulation tracking using the Temperley algorithm for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816) (Temperley 1999).* 197
Fig. H1 *Hierarchy of UML diagrams* 198
Fig. I1 *Distribution of note durations for the 24 preludes of Bach's Well-tempered Clavier Book I.* 200
Fig. I2 *Distribution of pitch classes for the 24 preludes of Bach's Well-tempered Clavier Book I.* 200
Fig. I3 *Distribution of note durations for the Courante from Bach's Cello Suite in C major.* 201
Fig. I4 *Distribution of dyad note durations for the Courante from Bach's Cello Suite in C major.* 201
Fig. I5 *Distribution of pitch intervals for the Courante from Bach's Cello Suite in C major.* 202
Fig. I6 *Distribution of dyad pitch intervals for the Courante from Bach's Cello Suite in C major.* 202

Fig. I7 Distribution of pitch intervals sizes for the Courante from Bach's Cello Suite in C major. 203
Fig. I8 Distribution of pitch intervals directions for the Courante from Bach's Cello Suite in C major. 203
Fig. I9 Distribution of pitch classes for the Courante from Bach's Cello Suite in C major. 204
Fig. I10 Distribution of dyad pitch classes for the Courante from Bach's Cello Suite in C major............................... 204
Fig. I11 Distribution of note durations for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816)........ 205
Fig. I12 Distribution of pitch classes for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816). 205

List of Tables

Table 2.1 *Pythagorean chromatic scale with the wolf tone at $F^{\#}$ (Wikipedia, the free encyclopaedia, 2011c).* 33
Table 2.2 *Just intonation chromatic scale derived with harmonic ratios of two, three and five (Adapted from Sethares, 2005).* 34
Table 2.3 *Comparison of the justness of the major and minor thirds and the fifth for two just intonation scales (Adapted from Sethares, 2005).* 35
Table 2.4 *Spectral scale based on four and five octaves of the harmonic series (Sethares, 2005).* 36
Table 2.5 *Equal temperament chromatic scale compared to just intonation (Adapted from Wikipedia, the free encyclopaedia, 2011b).* 39
Table 2.6 *Major and minor mode key profile coefficients proposed by Krumhansl and Kessler (Krumhansl, 1990).* .. 41
Table 2.7 *Major and minor mode psychological key profile coefficients proposed by Krumhansl and Kessler (Krumhansl, 1990).* 41
Table 2.8 *Performance evaluation strategies applied for the Krumhansl and Schmuckler key-finding algorithm.* 51
Table 2.9 *Major and minor mode key profile coefficients proposed by Temperley (1999).* 52
Table 2.10 *Major and minor mode tonal pitch class key profile coefficients proposed by Temperley (1999; 2002).* 56
Table 2.11 *Time division format specification for standard MIDI files (The International MIDI Association, 1996).* 66
Table 2.12 *Standard SMPTE frame rates (Rees, 2001).* 66
Table 2.13 *Key signature meta event data structure.* 67
Table 2.14 *Time signature meta event data structure.* 68
Table 2.15 *MIDI note pitch range in relation to the ISO octave map.* 68
Table 2.16 *Note on and note off event data structures.* 69
Table 3.1 *Note durations for determining the histograms of note durations.* 74
Table 3.2 *Interval sizes implemented for the statistical analysis of intervals.* 77
Table 3.3 *Pitch classes implemented for the statistical analysis of pitch classes.* 83
Table 4.1 *Parameter definitions for the score suite data structure shown in Fig. 4.4.* 96
Table 4.2 *Parameter definitions for the score array data structure shown in Fig. 4.5.* 96
Table 4.3 *Parameter definitions for the MIDI sequence data structure shown in Fig. 4.6.* 97
Table 4.4 *Parameter definitions for the data structure shown in Fig. 4.7 for the distributions of note durations.* 99
Table 4.5 *Parameter definitions for the data structure shown in Fig. 4.8 for the distributions of pitch intervals.* 100
Table 4.6 *Parameter definitions for the data structure shown in Fig. 4.9 for the distributions of pitch classes.* 101
Table 4.7 *Parameter definitions for the data structure shown in Fig. 4.11 for the distribution of pitch classes applied in the key-finding algorithm.* 102
Table 4.8 *Parameter definitions for the data structure shown in Fig. 4.12 for the key-finding algorithm.* 103
Table 4.9 *Parameter definitions for the data structure shown in Fig. 4.16 for the partitioning window.* 105
Table 4.10 *Cumulative durational weights for the pitch classes contained in the test score shown in Fig. 4.38 for a playback tempo 100 BPM, Saturation Duration (τ)=0.5 and Accent Index (I_a)=2.* 120
Table 4.11 *Cumulative durational weights for the dyad pitch classes contained in the test score shown in Fig. 4.38 for a playback tempo 100 BPM, Saturation Duration (τ)=0.5 and Accent Index (I_a)=2.* 122
Table 4.12 *Key-finding results for measures 1 to 40 of the Courante from Bach's Cello Suite in C major (Temperley, 1999).* 126
Table 4.13 *Algorithm parameters for evaluation of the Krumhansl and Schmuckler algorithm, using the Krumhansl and Kessler key profiles and Krumhansl key correlation formula.* 127
Table 4.14 *Comparison of the key correlation values published by Temperley (1999) and the software application for the first measure of "Yankee Doodle". The results were obtained using the Krumhansl and Kessler key profiles and Krumhansl key correlation formula.* 128

Table 4.15 *Algorithm parameters for evaluation of the Krumhansl and Schmuckler algorithm, using the Temperley key profiles and key correlation formula.* *130*
Table 4.16 *Algorithm parameters for evaluation of the Krumhansl and Schmuckler algorithm, using the key profiles, key correlation formula and flat pitch class vector proposed by Temperley.* *131*
Table 5.1 *Naming conventions adopted for the case studies conducted for the Krumhansl and Schmuckler key-finding algorithm.* *142*
Table 5.2 *Performance of the Krumhansl and Schmuckler key-finding algorithm for the full scores of the 24 preludes of Bach's Well-tempered Clavier Book I.* *143*
Table 5.3 *Performance of the Krumhansl and Schmuckler key-finding algorithm for the first four notes of the 24 preludes of Bach's Well-tempered Clavier Book I.* *143*
Table 5.4 *Performance of the Krumhansl and Schmuckler key-finding algorithm for the Courante from Bach's Cello Suite in C major.* *143*
Table 5.5 *Performance of the Krumhansl and Schmuckler key-finding algorithm for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816).* *143*
Table 5.6 *Pattern search parameter values used in the key profile estimation cases studies.* *150*
Table 5.7 *Unconstrained key profile estimation performance percentage for the full scores of the 24 preludes of Bach's Well-tempered Clavier Book I.* *151*
Table 5.8 *Constrained key profile estimation performance percentage for the full scores of the 24 preludes of Bach's Well-tempered Clavier Book I.* *151*
Table 5.9 *Correlations for the major mode key profiles shown in Fig. 5.31.* *153*
Table 5.10 *Correlations for the minor mode key profiles shown in Fig. 5.32.* *154*
Table 5.11 *Correlations for the major mode key profiles shown in Fig. 5.33.* *155*
Table 5.12 *Correlations for the minor mode key profiles shown in Fig. 5.34.* *155*
Table 5.13 *Unconstrained key profile estimation performance percentage for the first four notes of the 24 preludes of Bach's Well-tempered Clavier Book I.* *156*
Table 5.14 *Constrained key profile estimation performance percentage for the first four notes of the 24 preludes of Bach's Well-tempered Clavier Book I.* *156*
Table 5.15 *Correlations for the unconstrained major mode key profiles shown in Fig. 5.38.* *158*
Table 5.16 *Correlations for the unconstrained minor mode key profiles shown in Fig. 5.39.* *158*
Table 5.17 *Correlations for the constrained major mode key profiles shown in Fig. 5.40.* *159*
Table 5.18 *Correlations for the constrained minor mode key profiles shown in Fig. 5.41.* *160*
Table 5.19 *Unconstrained key profile estimation performance percentage for the* Courante *from Bach's* Cello Suite in C major. *160*
Table 5.20 *Constrained key profile estimation performance percentage for the* Courante *from Bach's* Cello Suite in C major. *161*
Table 5.21 *Correlations for the unconstrained major mode key profiles shown in Fig. 5.45.* *163*
Table 5.22 *Correlations for the unconstrained minor mode key profiles shown in Fig. 5.46.* *163*
Table 5.23 *Correlations for the constrained major mode key profiles shown nFig. 5.47.* *164*
Table 5.24 *Correlations for the constrained minor mode key profiles shown in Fig. 5.48.* *164*
Table 5.25 *Unconstrained key profile estimation performance percentage for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816).* *165*
Table 5.26 *Constrained key profile estimation performance percentage for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816).* *166*
Table 5.27 *Correlations for the unconstrained major mode key profiles shown in Fig. 5.52.* *167*
Table 5.28 *Correlations for the unconstrained minor mode key profiles shown in Fig. 5.53.* *168*
Table 5.29 *Correlations for the constrained major mode key profiles shown in Fig. 5.54.* *169*
Table 5.30 *Correlations for the constrained minor mode key profiles shown in Fig. 5.55.* *169*
Table 5.31 *Optimised key profiles with the best average key-finding scores for all test cases.* *169*
Table 5.32 *Key-finding performances of the optimised, Krumhansl and Kessler and Temperley key profiles for a key resolution of 0.1% with flat weighing.* *170*

Table 5.33 Key-finding performances of the optimised, Krumhansl and Kessler and Temperley key profiles for a key resolution of 0.1% with linear durational weighing. ... 170

Table 5.34 Key-finding performances of the optimised, Krumhansl and Kessler and Temperley key profiles for a key resolution of 0.1% with durational accent weighing. ... 171

Table 5.35 Key-finding performances of the optimised, Krumhansl and Kessler and Temperley key profiles for a key resolution of 1.0% with flat weighing. ... 172

Table 5.36 Key-finding performances of the optimised, Krumhansl and Kessler and Temperley key profiles for a key resolution of 1.0% with linear durational weighing. ... 172

Table 5.37 Key-finding performances of the optimised, Krumhansl and Kessler and Temperley key profiles for a key resolution of 1.0% with durational accent weighing. ... 173

Table 5.38 Key-finding performances of the optimised, Krumhansl and Kessler and Temperley key profiles for a key resolution of 10.0% with flat weighing. ... 173

Table 5.39 Key-finding performances of the optimised, Krumhansl and Kessler and Temperley key profiles for a key resolution of 10.0% with linear durational weighing. ... 174

Table 5.40 Key-finding performances of the optimised, Krumhansl and Kessler and Temperley key profiles for a key resolution of 10.0% with durational accent weighing. ... 174

Table 5.41 Comparison of major mode pitch class hierarchies for the Krumhansl and Kessler, Temperley, unconstrained Gavotte with a key resolution of 1.0% and unconstrained Courante with a key resolution of 10.0% profiles. ... 176

Table 5.42 Comparison of minor mode pitch class hierarchies for the Krumhansl and Kessler, Temperley, unconstrained Gavotte with a key resolution of 1.0% and unconstrained Courante with a key resolution of 10.0%. ... 178

Table 6.1 Performance of the Krumhansl and Schmuckler key-finding algorithm for the full scores of the 24 preludes of Bach's Well-tempered Clavier Book I. ... 182

Table 6.2 Performance of the Krumhansl and Schmuckler key-finding algorithm for the first four notes of the 24 preludes of Bach's Well-tempered Clavier Book I. ... 182

Table 6.3 Performance of the Krumhansl and Schmuckler key-finding algorithm for the Courante from Bach's Cello Suite in C major. ... 182

Table 6.4 Performance of the Krumhansl and Schmuckler key-finding algorithm for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816). ... 182

Table 6.5 Average key-finding performances of the optimised, Krumhansl and Kessler and Temperley key profiles for key resolutions of 0.1%, 1.0% and 10.0% with linear durational weighing. ... 182

Table B1 Key finding performance of the Krumhansl and Schmuckler algorithm for the first four notes of the forty-eight preludes of the Well-tempered Clavier composed by Bach (Krumhansl, 1990). ... 192

Table C1 Key-finding performance of the Krumhansl and Schmuckler algorithm for the first four notes of the twenty-four preludes composed by Shostakovich (Krumhansl, 1990). ... 193

Table D1 Key-finding performance of the Krumhansl and Schmuckler algorithm for the first four notes of the twenty-four preludes composed by Chopin (Krumhansl, 1990). ... 194

Table E1 Comparison of the key-finding performance of the Krumhansl and Schmuckler and the Longuet-Higgins and Steedman algorithms for the forty-eight fugues of the Well-tempered Clavier composed by Bach (Krumhansl, 1990). ... 195

Table F1 Key-finding performance of the Krumhansl and Schmuckler algorithm for the twenty-four fugues composed by Shostakovich (Krumhansl, 1990). ... 196

Table J1 Key-finding results of the Krumhansl and Schmuckler algorithm for the full scores of the 24 preludes of Bach's Well-tempered Clavier Book I. ... 206

Table J2 Key-finding results of the Krumhansl and Schmuckler algorithm for the first four notes of the 24 preludes of Bach's Well-tempered Clavier Book I. ... 207

Table J3 Key-finding results for the Krumhansl and Schmuckler algorithm for the Courante from Bach's Cello Suite in C major. 208
Table J4 Key-finding results of the Krumhansl and Schmuckler algorithm for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816). 210
Table K1 Pitch class coefficients for the Krumhansl and Kessler (Krumhansl, 1990) and the Temperley (1999) key profiles. 211
Table K2 Pitch class coefficients for the optimised key profiles based on the Gavotte from Bach's French Suite No. 5 in G major (BWV 816) and the Courante from Bach's Cello Suite in C major. 212

Abstract

Tonality cognition incorporates a number of diverse and multidisciplinary aspects, including music cognition, acoustics, culture, computer-aided modelling, music theory and brain science. Current research shows growing emphasis on the use of computational models implemented on digital computers for music analysis, particularly with reference to the analysis of statistical properties, form and tonal properties. The applications of these analytical techniques are numerous, including the classification of genre and style, Music Information Retrieval (MIR), data mining and algorithmic composition.

The research described in this document focuses on three aspects of tonality analysis, namely music cognition, computational modelling and music theory, particularly from the perspectives of statistical analysis and key-finding. Mathematical formulations are presented for a number of computational algorithms for analysing the statistical and tonal properties of music encoded in symbolic format. These include algorithms for determining the distributions of note durations, pitch intervals and pitch classes for statistical analysis and for template-based key-finding for tonal analysis. The implementation and validation of these computational algorithms on the Matlab software platform are subsequently discussed.

The software application is used to determine whether a more optimal combination of pitch class weighing model and key profile template for the template-based key-finding algorithm can be derived, using the 24 preludes from Bach's *Well-tempered Clavier Book I*, the *Courante* from Bach's *Cello Suite in C major* and the *Gavotte* from Bach's *French Suite No. 5 in G major (BWV 816)* as test material. Four pitch class weighing models, namely histogram weighing, flat weighing, linear durational weighing and durational accent weighing, are investigated. Two prominent key profile templates proposed in literature are considered, namely a key profile derived from tonality cognition experiments and a key profile based on classical music theory principles. The results show that the key-finding performances of all the combinations of the pitch class weighing models and existing key profile templates depend on the nature of the test material and that none of the combinations perform optimally for all test material.

The software application is subsequently used to determine whether a more optimal key profile template can be derived using a pattern search parameter estimation algorithm. This investigation was conducted for diverse sets of search conditions, including unconstrained and constrained key profile coefficients, different pitch class weighing models, various key resolutions and different search algorithm parameters. Using the same sample material as for the key-finding evaluations, the investigation showed that a more optimal key profile, compared to existing profiles, can be derived. In comparing the average key-finding scores for all of the test material, the optimised profiles outperform the existing profiles substantially. The optimised key profiles introduce new pitch class hierarchies where the supertonic and the subdominant rate higher at the expense of the mediant in the major profile to improve the tracking of key modulations.

Opsomming

Kognitiewe tonaliteit behels ′n aantal uiteenlopende en multidissiplinêre aspekte, insluitende musiek, akoestiek, kultuur, rekenaargesteunde modelering, musiekteorie en breinwetenskap. Huidige navorsing toon toenemende klem op die gebruik van berekenende modelering wat op digitale rekenaars geimplimenteer is vir musiekanalise, veral met verwysing na die analise van statistiese eienskappe, vorm en tonale eienskappe. Die aanwending van hierdie analitiese tegnieke is veelvoudig, insluitende die klassifikasie van genre of styl, onttrekking van musiekinformasie, dataversameling en algoritmiese komposisie.

Die navorsing wat in hierdie dokument beskryf word fokus op drie aspekte van tonaliteit analise, naamlik musiekkognisie, berekenende modelering en musiekteorie, veral vanuit die perspektiewe van statistiese analise and toonsoortsoek. Wiskundige formulerings word aangebied vir ′n aantal berekeningalgoritmes vir die analise van die statistiese en tonale eienskappe van musiek wat in simboliese formaat ge-enkodeer is. Hierdie sluit algoritmes in vir die bepaling van die verspreidings van nootlengtes, toonintervalle en toonklasse vir statistiese analise en vir templaat-gebaseerde toonsoortsoek vir tonale analise. Die implementering en validering van hierdie berekeningalgoritmes op die Matlab programmatuur platvorm word vervolgens bespreek.

Die programmatuur toepassing word vervolgens gebruik om te bepaal of ′n meer optimale kombinasie van toonklas weegmodel en toonsoortprofiel templaat vir die templaat-gebaseerde toonsoortsoek algoritme afgelei kan word, deur gebruik te maak van Bach se *Well-tempered Clavier Book I*, die *Courante* van Bach se *Cello Suite in C major* en die *Gavotte* van Bach se *French Suite No. 5 in G major (BWV 816)* as toetsmateriaal. Vier toonklas weegmodelle, naamlik histogram weging, plat weging, lineêre duurtyd weging en duurtyd aksent weging, word ondersoek. Twee prominente toonsoortprofiel template uit die literatuur word oorweeg, naamlik ′n toonsoortprofiel wat van tonaliteit kognisie eksperimente afgelei is en ′n toonsoortprofiel gebaseer op klassieke musiekteoretiese beginsels. Die resultate wys dat die toonsoortsoek prestasies van al die kombinasies van die toonklas weegmodelle en bestaande toonsoortprofiel template afhang van die aard van die toetsmateriaal en dat geen van die kombinasies optimaal presteer vir alle toetsmateriaal nie.

Die programmatuur toepassing word vervolgens aangewend om vas te stel of ′n meer optimale toonsoortprofiel afgelei kan word deur gebruik te maak van ′n patroonsoek parameterestimasie algoritme. Hierdie ondersoek is uitgevoer vir uiteenlopende stelle soektoestande, insluitende onbeperkte en beperkte toonsoortprofiel koëffisiënte, verskillende toonklas weegmodelle, ′n verskeidenheid toonsoort resolusies en verskillende soekalgoritme parameters. Deur gebruik te maak van dieselfde toetsmateriaal as vir die toonsoortsoek evaluerings, toon die ondersoek dat ′n meer optimale toonsoortprofiel, in vergelyking met bestaande profiele, afgegelei kan word. In ′n vergelyking van die gemiddelde toonsoortsoek prestasie vir al die toetsmateriaal, presteer die ge-optimeerde profiele aansienlik beter as die bestaande profiele. The ge-optimeerde toonsoortprofiele lei tot nuwe toonklas hiërargiee waar die supertonikum en die subdominant hoër rangposissies beklee ten koste van die mediant in die majeur profiel, ten einde die navolg van toonsoort modulasies te verbeter.

1 Project motivation and project description

1.1 Introduction

It is generally accepted that the term *music* has more often than not defied attempts at formal definition, both historically and in the modern era. Although many popular definitions have been offered, such as "*organized sound*" (Goldman, 1961), the term has proven to be of a fluid nature, often changing with historical context, as can be expected for a concept inherently defined in terms of perception rather than fundamental science. These sentiments are well expressed by Nattiez (1990) in the statement "*there is no single and intercultural universal concept defining what music might be*". It follows that the term *music analysis* inherits similar characteristics, particularly in the sense that it embodies an extremely wide range of concepts, all related through a complex and often loosely-defined hierarchy that involves constituents of a similar, often vaguely defined nature. The concept of tonality is one such example.

Historically, the concept of tonality has been defined and redefined repeatedly by music theorists and historians. Rameau (1971), in his *Treatise on Harmony*, established the most significant initial foundations for the theory of tonal music, and thus for the concept of tonality. Based on the physics of vibrating strings and the mathematics of whole numbers, Rameau derived the fundamental relationships between the tones of a scale and the tonic note or root. These relationships, expressed in terms of intervallic quantities and ratios, continue to form the basis for the concept of harmony and the associated concepts of consonance and dissonance.

The term tonality originates from the term *tonalité* used early in the nineteenth century, when the concept was associated primarily with the primary triad chords, tonic, subdominant and dominant. A more advanced definition of tonality, namely the "*set of relationships, simultaneous or successive, among tones of the scale*", was subsequently proposed by Fétis in 1844 (Judd, 1998). This early definition recognised the importance of both melody and harmony in defining tonality. Of particular importance in the interpretation proposed by Fétis, is the recognition of the importance of personal interpretation, and thus culture, in defining the concept of tonality.

In the modern era, the study of tonality benefitted hugely from the efforts of the post-war musicologist Dahlhaus. Dahlhaus (1990) documented and analysed the development of the relationships between tonality and composition from the intervallic approach used in the middle ages and renaissance period to the modern approach favouring chord progressions.

The most recent research contributions in the field of tonality clearly highlight the complex and multidisciplinary nature of tonality. In a comprehensive review of the current status of tonality cognition, Krumhansl (2004) addresses six diverse aspects that feature significantly in modern research efforts in this field, namely music cognition, acoustics, culture, computational modelling, music theory and brain science. The project described in this document incorporates three of these aspects, namely music cognition, computational modelling and music theory.

1.2 Project motivation

Unlike the concept of *tonality*, the concept of a *musical work* represented in symbolic format through some form of musical notation system has evolved as an established and well-defined entity. Modern score notation principles incorporate the following two types of information:

- *Deterministic musical elements:* These include elements such as the pitch and duration of notes in temporal context, time signatures, key signatures, etc.
- *Expressive information:* This refers to information that is open to interpretation by the performer, including tempo, dynamics and articulation.

The digital era has given rise to the development of a number of sophisticated, feature-rich software applications for producing and managing musical scores. Prominent examples include dedicated notation programs such as Sibelius, Finale, Lilypond and MuseScore and music production software such as Pro Tools and Cubase. These applications generally store score information in proprietary application-specific formats, thereby limiting the accessibility and usefulness of the encoded material. Most applications, however, have the ability to exchange information through the standard Musical Instrument Digital Interface (MIDI) file format, which has become an industry-standard for sharing music notation information (The MIDI Manufacturers Association, 1996; Good, 2001a).

The MIDI format was originally designed for real-time sequencing and playback rather than notation. As a result, some notational information is lost in the encoding process. Many alternative formats for encoding music notational information, such as Standard Music Description Language (SMDL) and the Notation Interchange File Format (NIFF), have been proposed (Sloan, 1997; Grande, 1997). The SMDL and NIFF formats, however, still suffer from serious deficiencies as an interchange format, particularly for the purpose of music analysis, and were never widely adopted (Good, 2001a; Good and Actor, 2003; Good, 2006). These concerns have been addressed in recent years by the development of MusicXML, which is a dedicated XML-based music notation file format designed specifically for the interchange of information between different music notation programs (Good, 2001b; Castan, Good and Roland, 2001). MusicXML Version 1.0 was released in 2004, with the current version, i.e. version 3.0, released in August 2011. Support for MusicXML as an interchange format expanded rapidly in the years following its introduction (Good and Actor, 2003; Kuipers and Good, 2004). It is now supported by most professional music notation software applications, including Finale, Sibelius, Encore, Nuendo, Score, MuseScore and many more (Recordare, 2011).

Despite shortcomings for the production of sheet music, the MIDI file storage format remains a viable storage and exchange option for computer-aided music analysis (Good, 2001a). This is particularly true in view of the extensive collection of MIDI-encoded music available through websites such as Classical Archives (Classical Music on Archives, 2011) and the online library of public-domain music, Musopen (Musopen, 2011). The MIDI format also has the benefit that much of the historical research on methodologies and algorithms for music analysis based on symbolic sources has been applied to music encoded in MIDI format. A prime example of this is the MIDI toolbox, which represents a collection of functions for analysing and visualising MIDI files in the Matlab environment (Eerola and Toiviainen, 2004a; 2004b). The functionality implemented in the MIDI Toolbox represents the departure point and frame of reference for the work conducted for this research project, i.e. the development of a user-friendly Matlab application for the statistical and tonal analysis of music encoded in MIDI format.

The deterministic content of notated music offers an excellent basis for some aspects of music analysis using digital computers, particularly with reference to the analysis of statistical properties, form and tonality. The applications of these analytical techniques are numerous, including the classification of genre and style, Music Information Retrieval (MIR), data mining and algorithmic composition.

The MIDI toolbox encodes a significant part of the modern analytical techniques developed for the analysis of music encoded in symbolic format, particularly with reference to statistical properties, tonality, meter and melody (Eerola and Toiviainen, 2004a; 2004b). It represents a valuable research tool, particularly with reference to the following:

- The toolbox makes it possible to extend the existing body of case studies to include a wider selection of subject material, thereby representing a more comprehensive range of genres and musical content.

- The toolbox facilitates performance comparisons of different methodologies and algorithms for different categories of subject material. This is particularly useful for research efforts aimed at improving the existing algorithms and developing new methodologies.

In its present form, the MIDI toolbox is a collection of Matlab m-files that is available free under the GNU General Public License from the website of the University of Jyväskylä (Midi Toolbox, 2011). Although it represents an excellent research resource, a number of practical considerations impose limitations on its usefulness for many students and researchers. These include the following:

- Using the routines offered by the toolbox effectively for music analysis requires a reasonably high level of competency in programming Matlab. This is particularly true when dealing simultaneously with multiple scores and in keeping track of the parameter values associated with analysis options. The development of a Graphical User Interface (GUI) has been proposed by the developers of the MIDI toolbox (Eerola and Toiviainen, 2004b), but has not been realised to date.

  The code for some of the methodologies not implemented in the MIDI toolbox, such as the preference rule key-finding algorithm proposed by Temperley (1999), is also available online. As in the case of the MIDI toolbox, the user requires programming skills, with the difference that the code is implemented in the C programming language rather than in Matlab.
- The toolbox is accompanied by a comprehensive users guide with a fairly detailed reference list to assist potential users (Eerola and Toiviainen, 2004a). Some applications however, require in-depth scrutiny of the Matlab code to determine exactly how the original research material was interpreted in coding the associated algorithms and thus how to interpret the results.
- The current implementation of the MIDI toolbox offers considerable scope for increased flexibility and extended functionality in some respects. These include examples such as the following:
  - Cases arise where the flexibility and versatility offered by the implementation can be improved by presenting the user with a well-designed set of parameter options to adjust, e.g. the ability to bypass the hard-coded incorporation of durational accent (Huron and Parncutt, 1993) in determining statistical properties such as the various distributions of pitch intervals and pitch classes.
  - Some prominent analysis algorithms are not currently included, such as the key-finding algorithms proposed by Temperley (1999).

Many of the algorithms encoded in the MIDI toolbox have been applied mainly for performance evaluation of the algorithms themselves, rather than for analysis and research of the broader corpus of available music. By addressing the above considerations, it is believed that practical application of the available algorithms and methodologies for computation music analysis can be extended considerably beyond the limited number of case studies performed to date.

The focus of this project is on computational tonality analysis, particularly from the perspective of key-finding and determining tonal centre. Tonality analysis using algorithms implemented on digital computers have gained significant importance in recent years for applications such as data mining, music classification, etc. These algorithms include the following:

- The key-finding algorithm proposed by Longuet-Higgins and Steedman (1971).
- The key profile template-based key-finding algorithm proposed by Krumhansl and Schmuckler (Krumhansl, 1990).
- The modified Krumhansl and Schmuckler key-finding algorithms proposed by Huron and Parncutt (1993).

- The modified Krumhansl and Schmuckler key-finding algorithms proposed by Temperley (1999).
- The preference rule algorithm for key-finding and tracking key modulations proposed by Temperley (1999).
- The spiral array approach proposed by Chew (2001).

The algorithm proposed by Longuet-Higgins and Steedman (1971) applies a shape-matching algorithm to a two-dimensional geometrical model of the relationships between pitch classes while the algorithm proposed by Chew (2001) determines the key based on a three-dimensional spiral model of the relationships between pitch classes. The four remaining algorithms determine the key from the correlation between a statistical measure of the pitch classes in the sample material and a key profile template. This give rise to the following considerations with regard to the two parameters used in these algorithms:

- *Statistical representation of the pitch classes in the sample material*:

  The original Krumhansl and Schmuckler algorithm (Krumhansl, 1990) weighs the occurrence of pitch classes by note duration. Huron and Parncutt (1993) proposed a modified weighing scheme, introducing the concept of durational accent whereby the durational weights are modified to reflect the effects of echoic memory. Temperley (1999) applied a flat weight in the sense that the distribution of pitch classes reflects the presence rather than the cumulative duration or frequency of occurrence of a pitch class in the sample material, similar to the approach used in the Longuet-Higgins and Steedman (1971) shape-matching algorithm.

- *Key profile*:

  The original Krumhansl and Schmuckler algorithm (Krumhansl, 1990) and the modified algorithm proposed by Huron and Parncutt (1993) use psychological key profile templates derived by Krumhansl and Kessler (1982) through experimental tonality cognition studies with human listeners. Temperley (1999) proposed modified key profiles derived through the application of classical music theoretical principles. These are used in the modified Krumhansl and Schmuckler algorithms and the preference rule algorithm devised by Temperley. The weights assigned to the individual pitch classes in the Krumhansl and Kessler and the Temperley key profiles share a number of hierarchical properties, but also exhibit significant differences in some respects.

The above overview shows that the two parameters, namely the durational weights assigned in the statistical representation of the pitch classes in the sample material and the key profile template have been modified extensively along the research timeline reflected by the various contributions. Not all combinations of the various proposals have been evaluated. This gives rise to the following research questions:

- Given the various durational weighing schemes and key profiles that have been proposed, can a more optimal combination, compared to the limited number of combinations evaluated in the individual research contributions, be identified?
- Given the significant differences between the proposed key profile templates, can a more optimal pitch class template be derived using modern parameter estimation techniques?

The above overview raises a number of issues with regards to the current status of efforts to develop algorithms and implementations for the computational modelling of tonality and key-finding in particular, thereby giving rise to the project description presented in the next section.

1.3 Project description

1.3.1 Research objectives

The project background and discussions presented in section 1.2 give rise to the following research objectives:

- The development of a user-friendly Matlab application with a Graphical User Interface (GUI) and versatile parameter options hierarchy for analysing the statistical and tonality properties of symbolic music:
- An investigation to determine whether a more optimal combination, compared to the combinations evaluated to date, of the pitch class representation of the sample material and key profile template can be derived.
- An investigation to determine whether a more optimal key profile template can be derived using parameter estimation techniques such as direct search methods.

The main focus of the software development is on implementation of the most prominent key profile template-based key-finding algorithms. The statistical analysis component, however, is included because most of the statistical algorithms, particularly with reference to note durations and pitch classes, are required as part of the key-finding processes. Furthermore, modern computational music analysis algorithms rely increasingly on statistical indicators of note durations, pitch intervals and pitch classes (Eerola and Toiviainen, 2004b). Emerging trends in tonality analysis include the statistical properties of pitch intervals as demonstrated by Madsen and Widmer (2007).

1.3.2 Research methodology

The first research objective involves the development of a Matlab implementation with a GUI for the existing statistical and tonality analysis algorithms. The research plan for this phase is summarised by the following task list:

- *Conduct a literature review:*

  The focus of this literature study is as follows:

  - The physical, mathematical and music theoretical foundations of musical scales, tuning and temperament.
  - The historical and modern developments in tonality cognition, key-finding algorithms and computational music analysis in general.
  - The MIDI toolbox implementation, particularly with the view to make an assessment of the functionality implemented in the toolbox for the statistical and tonality analysis of symbolic music.
  - MIDI technology, particularly with reference to the suitability of the MIDI file format for the symbolic representation of music for analysis purposes.
  - Parameter estimation algorithms, particular direct search methods such as the pattern search algorithm.

- *Formal mathematical formulation of the algorithms implemented in the analysis routines*:

  The algorithms as implemented in a resource such as the MIDI toolbox are based on research contributed by numerous researchers over an extended period of time. In some cases, contributions by different researchers at different times are combined in implementing a particular algorithm. The algorithms are generally described in verbal terms in the relevant research literature, which results in a certain degree of uncertainty with regards to exactly how the algorithms were actually implemented. Formal mathematical formulations represent unambiguous guidelines of exactly how the algorithms are to be implemented in the proposed software application. In addition, these formulations assist in interpreting the results obtained using the algorithms. The formulations will be derived through close scrutiny of the relevant research publications and the Matlab and C code available online.

- *Design of a suitable program structure for the proposed implementation:*

  This aspect is important in view of the design objective of achieving a versatile implementation that facilitates flexible use of the parameter options and algorithms developed to date. The design of suitable data structures for the score information and parameter options are of particular importance in this regard.

- *Implementation of the analysis algorithms and development of a suitable Graphical User Interface (GUI) to access these algorithms:*

  Most of analysis algorithms require a number of parameter values, while routines such as the key-finding algorithms use other algorithms, e.g. the distribution of pitch classes algorithm. It follows that a highly flexible and structured user interface that facilitate menu-driven control of the various algorithm parameters is essential.

- *Program validation:*

  It is important to validate the mathematical formulations and actual implementations of the analysis algorithms, particularly in view of the fact that not all algorithms are described in unambiguous terms in literature. This will be achieved using specially designed test material and test cases published in literature.

The second objective involves evaluation of various untried combinations of algorithmic parameter settings for the key-finding algorithms, with the objective of determining whether a more optimal combination can be identified. This research plan for this phase includes the following:

- *Identification of suitable sample material:*

  Results published in literature for the various algorithms show that the performance depends strongly on the nature of the test material. Some algorithms deliver good results for a particular test case, but perform poorly for a different test case.

- *Select a number of appropriate test scenarios and conduct the relevant performance evaluation simulations:*

  The test scenarios, i.e. parameter combinations, must be chosen sensibly given the properties of the sample material and properties of the algorithm in question.

The third objective involves the application of optimization methods to search for an improved key profile template to use with the template-based key-finding algorithms. The research plan for this phase includes the following:

- *Identification of an appropriate cost function and optimization algorithm:*

  The choice of optimization algorithm depends on the properties of the cost function. In this regard, for instance, it is important to take cognizance of the fact that many popular optimization algorithms require a differentiable cost function.

- *Identification of suitable sample material:*

  The sample material should represent as wide a range of keys and key modulations as possible, with reliable key references from music theoretical sources. Examples of suitable candidates, often used in the performance evaluation of the key-finding algorithms, include the 48 preludes fugue subjects of Bach's *Well-Tempered Clavier*, the 24 preludes of Shostakovich and the 24 preludes of Chopin.

- *Conduct optimization exercises to identify a more optimal key profile template:*

Given the numerous parameters settings that affect the performance of an optimization algorithm, a carefully designed approach is required to manage the numerous permutations that can arise.

- *Evaluate the performance of the optimised key profile candidates using suitable score material.*

The above research methodology represents a fairly detailed plan and also reflects an appropriate temporal order for the main research tasks contained in the project.

1.4 Thesis structure

The remainder of this document is structured as follows:

- *Chapter 2: Literature review:*

  The relevant literature is reviewed. The review includes musical scales, tuning and temperament, modern interpretations and computational models of tonality cognition, key-finding algorithms, geometrical models of tonal relationships, the MIDI toolbox Matlab implementation, the standard MIDI file format and an overview of direct search methods for solving optimization problems.

- *Chapter 3: Mathematical formulation of computational music analysis algorithms:*

  The computation algorithms implemented in the software application are described in mathematical terms. These include two categories, namely the statistical analysis algorithms and the tonality analysis or key-finding algorithms. The statistical analysis algorithms include the various distributions of note durations, pitch intervals and pitch classes, while the key-finding algorithms include the Krumhansl and Schmuckler algorithm (Krumhansl, 1990), the variations thereof proposed by Huron and Parncutt (1993) and Temperley (1999) and the preference rule key-finding algorithm proposed by Temperley (1999). Finally, the experimental algorithm for deriving optimal key profiles using direct search methods is discussed.

- *Chapter 4: Program implementation:*

  The software design and implementation of the analysis algorithms in Matlab is discussed. The program structure and functionality implemented in the application is reviewed with reference to the various user-definable options for environmental and computational parameters, and the data structures used in the implementation are presented. The functionality offered by the Graphical User Interface (GUI) is discussed.

- *Chapter 5: Program validation and results:*

  The validation procedures and results for the various analysis algorithms implemented in the program are discussed. Results are subsequently presented for the case studies conducted with the application to address the research questions proposed in the project description.

- *Chapter 6: Conclusions and recommendations:*

  Final conclusions and recommendations for further work are presented.

2 Literature review

2.1 Introduction

The scope of the literature review conducted for the purposes of this project is determined by the project objectives outlined in the project description given in section 1.3. The objective of the literature review is to provide a theoretical foundation for the development and implementation of the analytical methodologies and computational algorithms incorporated in the software implementation. This gives rise to the following considerations pertaining to the subject matter relevant to the literature study:

- *Physical and mathematical foundations of tone and timbre:*

  Tonality cognition is defined in the context of the perception of the consonant properties of stable pitches rendered successively, i.e. melody, or simultaneously, i.e. harmony. It is therefore important to establish the physical and mathematical properties of periodic waveforms such as the sound waves produced by musical instruments. The Fourier series representations of such waveforms embody the fundamental mathematical relationships that also feature in the relationships between the intervals contained in musical scales. In the context of tonality perception, it is important to take cognizance of the effects of timbre in designing and interpreting the results of tonality cognition experiments.

- *Musical scales, tuning and temperament:*

  The musical scale provides the set of pitch classes used in all computational tonality cognition models and key-finding algorithms. Although tuning and temperament does not feature directly in the tonality analysis algorithms for symbolic music, it is an important factor in tonality analysis based on sound signals. It is also acknowledged that temperament, as is also true for timbre, has to be considered in designing and interpreting the results of tonality cognition experiments. This is motivated on the basis that scale structure and the associated intonation, tuning and temperament system cannot be divorced from the human perception of the consonance of the musical intervals. Tonality cognition experiments, e.g. the experiments conducted by Krumhansl and Kessler (1982), constitute the basis for the psychological key profiles used in one of the most prominent key-finding algorithms for symbolic music, namely the Krumhansl and Schmuckler (Krumhansl, 1990) key-finding algorithm.

  While Krumhansl recognises the importance of timbre in designing the sonic stimuli used in the probe-tone experiments, the possible effects and implications of different tuning and temperament systems on the results have not been addressed.

- *Experimental investigations of tonality perception:*

  Modern computational models of tonality analysis are based on the results of experimental investigations to determine how tonal relationships are perceived by human listeners. These studies have given rise to methodologies such as the probe-tone technique to determine how tone is perceived in the presence of tonal contexts defined by scales, cadences and chords. Tonality cognition has also been investigated with reference to perceptual influences such as durational accent and pitch perception.

- *Computational models of tonality cognition:*

  A number of computational models for tonality cognition have been proposed in literature. These are generally based on mathematical representations of the tonal relationships between pitches, pitch intervals, chords and keys. Key-finding algorithms for determining tonal centre and key modulations are typically based on some form of tonality cognition model. The most prominent computational models for tonality cognition will be reviewed and categorised with the view to identify the key-finding algorithms most suitable for incorporation into the proposed software application.

- *Geometrical models of tonal relationships:*

  The tonal relationships associated with the computational models for tonality cognition are often expressed in the form of geometrical models that map the relationships between pitches, pitch intervals, chords and keys to a two-dimensional or three-dimensional spatial representation. These models then form the basis for visualising tonal centre and key modulations. In the MIDI toolbox, for instance, visual representations of the tonal centre are presented in the form of Self Organising Maps (SOMs). The geometrical models associated with the most prominent computational models of tonality cognition will be reviewed.

- *Functionality and implementation details of the Midi toolbox:*

  The MIDI toolbox represents a comprehensive implementation on the Matlab software platform of the most important algorithms and models reported in literature for analysing music represented in symbolic format. The functionality offered by the MIDI toolbox includes the analysis of statistical properties, tonality, metrical structure and melodic properties. This functionality is reviewed in the literature review.

- *The standard MIDI file format from the perspective of the representation of musical score information:*

  The implementation is required to import MIDI data from MIDI files and create a data structure for analysis. This data structure must extract all relevant music notational elements from the imported file. In order to develop an understanding of the limitations of the standard MIDI file format and the implications thereof for music analysis, it is necessary to have a thorough understanding of the file structure and data encoding principles implemented by the MIDI file specification.

2.2 The physical and mathematical foundations of tone and timbre

2.2.1 Introduction

The fundamental musical concepts of tone and timbre are firmly rooted in the physical and mathematical properties of periodic waveforms. From these properties arise also the higher level concepts such as musical scales, consonance and dissonance, harmony, tuning systems and all other aspects related to tonality. Much of the early developments in relating physics, mathematics and music are attributed to the Ionian philosopher Pythagoras, who studied the relations between the length of the string and the pitch of the musical note produced by it. Pythagoras is credited with the discovery that the pitch intervals between harmonious frequencies relate to numerical ratios of whole numbers.

2.2.2 Mathematical properties of periodic waveforms

The harmonic properties of musical intervals and scales in the context of tonal music derive from the physical and mathematical properties of repetitive waveforms, as manifested by the sound waves produced by physical instruments. The mathematical representation and analysis of periodic functions of time, as representations of the dynamic and oscillatory sonic time-domain waveforms produced by musical instruments, are therefore essential for understanding the musical concepts of intervals, scales and timbre.

The sinusoidal function of time, shown in Fig. 2.1, constitutes one of the most fundamental phenomena in physics. In the context of sound, it represents the fundamental dynamic behaviour of physical entities such as vibrating strings, oscillating air columns, the modes of vibration of solid bodies, etc. Mathematically, through Fourier analysis, a time-domain representation can be mapped to a frequency domain representation defined in terms of sinusoidal functions. The sinusoidal waveform $f(t)$ shown graphically in Fig. 2.1 is defined by the relationship

$$f(t) = A\sin(\omega t + \theta) \tag{2.1}$$

where A denotes the peak amplitude, ω denotes the frequency in radians per second, t denotes time in seconds and θ denotes the phase angle in radians of the waveform. The radian frequency ω is defined by the relationship

$$\omega = 2\pi f \tag{2.2}$$

with

$$f = \frac{1}{T} \tag{2.3}$$

where f denote the frequency in cycles per second [Hz] and T denote the period in seconds.

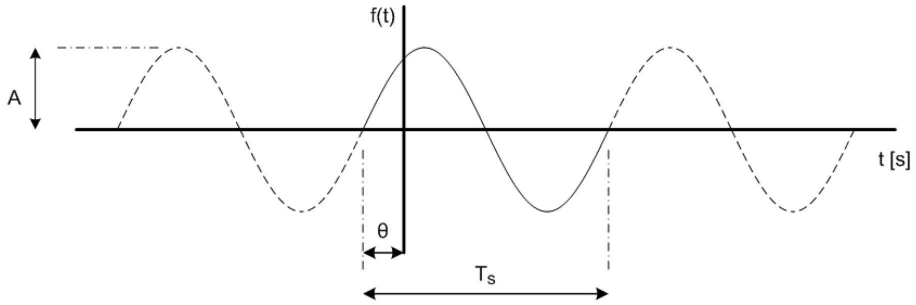


Fig. 2.1 Time-domain representation of a sinusoidal waveform.

A physical musical instrument playing a stable pitch produces a complex, yet periodic, sonic waveform rather than a simple sinusoidal waveform. Fourier analysis allows such a complex periodic waveform to be expressed mathematically as the sum of a finite or infinite series of sinusoidal components through the relationship

$$f(t) = \sum_{n=1}^{N} A_n \sin(\omega_n t + \theta_n) \tag{2.4}$$

with

$$\omega_n = 2\pi f_n \tag{2.5}$$

and

$$f_n = nf_1 \tag{2.6}$$

where A_n denotes the peak amplitude, ω_n the frequency in radians per second, f_n the frequency in Hz and θ_n the phase angle in radians of the n^{th} harmonic component. In musical terminology, the harmonic components of a sonic waveform are known as the overtones or partials of the waveform. The harmonic frequency spectrum defined by (2.4) assumes the form of a line spectrum consisting of evenly spaced discrete frequency components as shown by the amplitude spectrum shown in Fig. 2.2. The harmonic phase information can be represented similarly by a discrete phase spectrum.

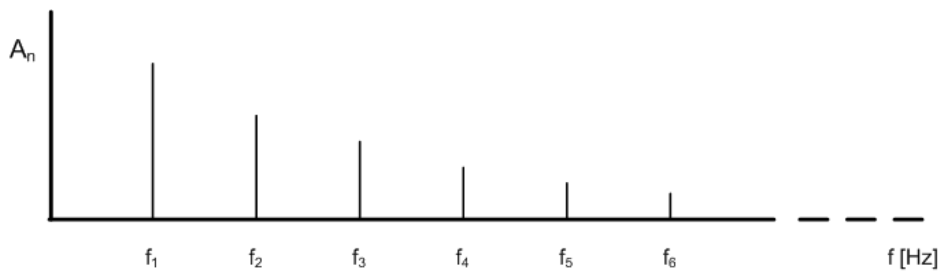


Fig. 2.2 Line spectrum representation of a periodic waveform.

2.2.3 Frequency domain properties of musical instruments

The concepts of consonance and dissonance and the structures of musical scales and tuning practices derive from the spectral properties of the sonic waveforms produced by musical instruments. It follows that these spectral properties are also fundamental to music theoretical concepts such as melody and harmony and psycho-acoustical concepts such as timbre and tonality. The frequency spectrum of a sound stimulus represents the primary characteristic, together with waveform, sound pressure, frequency location of the spectrum and the temporal characteristics, that define the timbre of the stimulus, i.e. "*that attribute of sensation in terms of which a listener can judge that two sounds having the same loudness and pitch are dissimilar*" (American Standards Association, 1960). Sethares (2005) concluded that just intonation and the Western equal tempered scale derive from the harmonic spectra and timbre of Western instruments.

2.3 Musical scales, tuning and temperament

2.3.1 Introduction

The Greek philosopher Pythagoras is generally credited with the initial investigations into the relationships between music and mathematics. In exploring the consonance properties of the sounds produced by vibrating strings, Pythagoras discovered that the sections of a subdivided string produce the most consonant sounds if the lengths of the resulting string sections form numerical ratios of whole numbers.

The fundamental frequency f of a vibrating string is defined by the relationship

$$f = \frac{1}{2L}\sqrt{\frac{T}{\mu}}\ \text{[Hz]} \tag{2.7}$$

where L denotes the length of the string in m, T denotes the tension of the string in Newton and μ denotes the linear mass of the string in kg/m. Equation (2.7) shows that the frequency of the fundamental mode of vibration is inversely proportional to the length of the string. Mathematically, the ratio of the fundamental frequencies f_1 and f_2 produced by two strings with lengths L_1 and L_2 respectively, for equal tension and linear mass, is given by the relationship

$$\frac{f_1}{f_2} = \frac{L_2}{L_1}. \tag{2.8}$$

Thus, if the ratio of the string lengths is expressed as a ratio of whole numbers, the same applies for the ratio of the fundamental frequencies. It is the latter entity that is perceived as harmonious by the listener. All musical scale structures and tuning systems are inherently derived from the consonant nature of pitches related by ratios of whole numbers.

The musical scale, i.e. a discrete series of notes or pitch elements, represents the building blocks of tonal music. Rather than absolute pitch, it is the relative pitch of a note compared to the other members of the scale set that defines the characteristics of individual notes in the context of tonality. In musical terminology, a scale refers to a set of pitch classes or notes arranged in ascending or descending order of pitch, with the distance between two successive pitches designated as a scale step. A scale is defined by the following two fundamental characteristics:

- *The number of pitch classes contained in the scale:*

  The pentatonic (five notes), hexatonic (six notes), heptatonic (seven notes) and octatonic (eight notes) classes are common examples of the number of pitch classes.

- *The intervals represented by the pitch classes:*

  The scale structures used in classical Western compositions include the seven-tone diatonic scale, seven-tone melodic and harmonic minor scales and twelve-tone chromatic scale.

In the context of tonality, a pitch class is characterised by the ratio of the frequency of the pitch class to the frequency of the tonic. The intervallic relationships between two pitch classes are likewise characterised by the ratio of the frequencies of the pitch classes. For many scales and temperaments, these ratios assume the form of a ratio of whole numbers. It is also customary to express the intervals between pitch classes in cents, which is derived by dividing the equal tempered semitone logarithmically into 100 equal parts. It is therefore the 1200th root of 2, a ratio approximately equal to (1:1.0005777895).

The '*cents-value*' of an interval ratio R is calculated using the relationship

$$cents = 1200 \frac{\log_{10} R}{\log_{10} 2} = 1200 \log_2 R. \tag{2.9}$$

Historically, the construction of musical scales is firmly rooted in the science of acoustics and the mathematics of numbers. The classical analytical approach used to study the properties of musical scales, adopted initially by Pythagoras and subsequently by Helmholtz (1954), is based on the harmonious relationships between the sounds produced by vibrating strings with lengths that form numerical ratios of positive integer numbers. Classical and modern music theory is therefore based on ratios and intervals rather than absolute pitch or frequency. This is particularly evident in the theories of consonance and harmony contributed by Pythagoras, Rameau (1971) and Helmholtz (1954).

The number of pitch classes contained in the scales in common use varies, depending on the historical and cultural context. This research described in this document focuses on tonality in the context of Western music practice. The treatment of scale structures will consequently focus predominantly on the Western 12-tone chromatic scale.

2.3.2 The Pythagorean scale

The Pythagorean 12-tone chromatic scale, generally credited to Pythagoras, is the oldest Western scale structure. It is derived from the highly consonant ratios 2:1 and 3:2, i.e. the octave and perfect fifth intervals respectively in music terminology, by raising or lowering the first degree of the scale repeatedly by the ratios of 3:2 or 2:3 respectively, and using downward or upward octave transpositions by the ratios 1:2 and 2:1 respectively. Mathematically, the intervals formed by the pitch classes with the tonic can be represented by the relationship

$$F_{mn} = 2^m \left(\frac{3}{2}\right)^n \tag{2.10}$$

where m and n denote integer numbers. Completing the circle of fifths, however, does not return the interval to the unison. This is due to the fact that the mathematical relationship

$$\left(\frac{3}{2}\right)^n = 2^m \tag{2.11}$$

has no integer solution (Sethares, 2005). Raising the tonic through twelve steps of fifths, i.e. $(3/2)^{12}$, yields a ratio of $\frac{531441}{4096}$, which transforms to $\frac{531441}{524288}$ through octave transformations. This ratio, known as the Pythagorean comma, is approximately equal to 1.0136 or about one quarter of a semitone. The problem is resolved by detuning one of the fifths by an appropriate amount, giving rise to the well-known *wolf* tone in Pythagorean tuning. Table 2.1 summarises the pitch ratios and numerical processes for deriving a Pythagorean chromatic scale with $F^{\#}$ designated as the *wolf* tone (Sethares, 2005). Fig. 2.3 represents the Pythagorean tuning system as a circle of fifths representation.

Table 2.1 Pythagorean chromatic scale with the wolf tone at $F^{\#}$ (Wikipedia, the free encyclopaedia, 2011c).

Scale degree	Mathematical expression	Ratio	cents	Comments
C	$2^0 (3/2)^0$	1/1	0	Tonic
$C^{\#}$	$2^3 (3/2)^{-5}$	256/243	90	Five fifths down, three octaves up
D	$2^{-1} (3/2)^2$	9/8	204	Two fifths up, one octave down
$D^{\#}$	$2^2 (3/2)^{-3}$	32/27	294	Three fifths down, two octaves up
E	$2^{-2} (3/2)^4$	81/64	408	Four fifths up, two octaves down
F	$2^1 (3/2)^{-1}$	4/3	498	One fifth down, one octave up
$F^{\#}$	$2^{-3} (3/2)^6$	729/512	612	Six octaves up, three octaves down
G	$2^0 (3/2)^1$	3/2	702	One fifth up
$G^{\#}$	$2^3 (3/2)^{-4}$	128/81	792	Four fifths down, three octaves up
A	$2^{-1} (3/2)^3$	27/16	906	Three fifths up, one octave down
$A^{\#}$	$2^2 (3/2)^{-2}$	16/9	996	Two fifths down, two octaves up
B	$2^{-2} (3/2)^5$	243/128	1110	Five fifths up, two octaves down
C	$2^1 (3/2)^0$	2/1	1200	One octave up

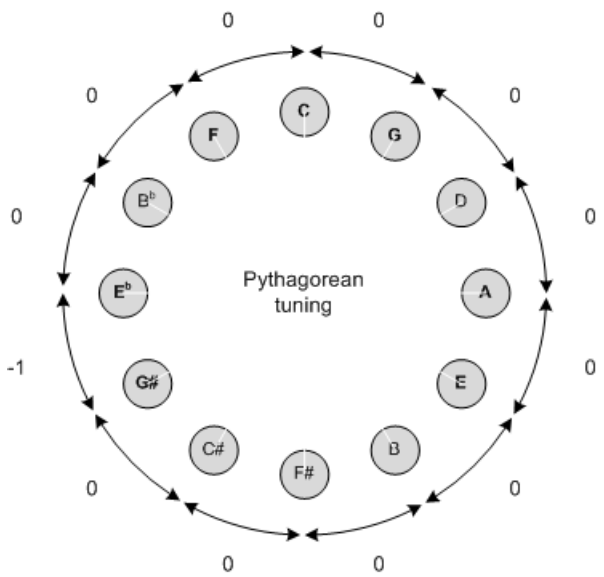


Fig. 2.3 Circle of fifths representation of Pythagorean tuning.

The Pythagorean scale was widely used and is well suited for music treating fifths as consonances and thirds as dissonances, e.g. classical music written before the 16th century. This is due to the relatively complex ratios of 81:64 and 32:27 for the major and minor thirds respectively (Wikipedia, the free encyclopaedia, 2011c). Although other scales have superseded the Pythagorean scale in music practice, Pythagoras established the foundation for structured music by defining the musical scale in terms of precise mathematical relationships. Modern tempered scales such as such as the mean-tone, well-tempered and equal-tempered scales derive directly from the

Pythagorean scale. The development of the Pythagorean scale has had a profound influence on musical education, the development of musical instruments and composers such as Mozart and Beethoven (Ferrara, 1991).

2.3.3 Just intonation scales

Just intonation scales are constructed based on the following two requirements:

- The frequencies of the scale notes must be related by ratios of whole numbers. Based on this requirement, the Pythagorean scale also qualifies as an example of a just intonation scale.
- The number ratios should consist of relatively small numerical values, especially for the third and fifth intervals. The main objective of this requirement is to improve the consonance of the major and minor third intervals, compared to whole-number ratios such as 81/64 and 32/27 as used in the Pythagorean scale. Based on this requirement, the Pythagorean scale is not normally classified as a just intonation scale.

The requirement that the frequency of the scale note forms a ratio of whole numbers with the tonic effectively ensures that the members of the scale are also members of a harmonic series. The scale notes are derived using harmonic ratios such as the second, third and fifth orders. Table 2.2 shows an example where the harmonic ratios are limited to the second, third and fifth orders, known as 5-limit just intonation (Sethares, 2005). Mathematically, the resulting intervals can be represented by the relationship

$$F_{mnp} = 2^m 3^n 5^p \tag{2.12}$$

where m, n and p denote integer numbers. Based on the mathematical formulation of just intonation scales, the Pythagorean scale qualifies as a 3-limit just intonation scale.

Table 2.2 Just intonation chromatic scale derived with harmonic ratios of two, three and five (Adapted from Sethares, 2005).

Scale degree	Mathematical expression	Ratio	cents	Comments
C	1	1/1	0.00	Tonic
$C^{\#}$	$2^4\ 3^{-1}\ 5^{-1}$	16/15	111.73	One third down, one fifth down, four octaves up
D	$2^{-3}\ 3^2\ 5^0$	9/8	203.91	Two thirds up, three octaves down
$D^{\#}$	$2^1\ 3^1\ 5^{-1}$	6/5	315.64	One third up, one fifth down, one octave up
E	$2^{-2}\ 3^0\ 5^1$	5/4	386.31	One fifth up, two octaves down
F	$2^2\ 3^{-1}\ 5^0$	4/3	498.04	One third down, two octaves up
$F^{\#}$	$2^{-5}\ 3^2\ 5^1$	45/32	590.22	Two thirds up, one fifth up, five octave s down
G	$2^{-1}\ 3^1\ 5^0$	3/2	701.96	One third up, one octaves down
$G^{\#}$	$2^3\ 3^0\ 5^{-1}$	8/5	813.69	One fifth down, three octaves up
A	$2^0\ 3^{-1}\ 5^1$	5/3	884.36	One third down, one fifth up
$A^{\#}$	$2^4\ 3^{-2}\ 5^0$	16/9	996.09	Two thirds down, four octaves up
B	$2^{-3}\ 3^1\ 5^1$	15/8	1088.27	One third up, one fifth up, three octaves down
C	$2^1\ 3^0\ 5^0$	2/1	1200.00	One octave up

Table 2.3 compares the intervallic properties of two just intonation scales with reference to the justness of the major and minor thirds and the fifth. For the first scale, the table identifies the notes that form a just major third and just fifth respectively with the notes located four and seven scale steps above. For the second scale, the table identifies the notes that form a just minor third and just fifth respectively with the notes located three and seven scale steps above.

Just intonation scales are key specific, giving rise to dissonant intervals in keys that are distant from the tonic on which the scale is based. This naturally presents a problem for music that modulates extensively. However, just intonation enjoys growing support for the inherent highly consonant intervallic properties (The Just Intonation Network, 2011). Retuning and adaptive tuning are considered as solutions for the modulation issue. Just tone intonation are not limited to twelve-tone scales, but has given rise to alternative scale structures such as the 43-tone scale proposed by Partch

(1974). Partch investigated the theory and application of just intonation scales extensively, including the development of a dedicated family of instruments to take advantage of the 43-tone scale structure.

Table 2.3 Comparison of the justness of the major and minor thirds and the fifth for two just intonation scales (Adapted from Sethares, 2005).

Scale degree	Scale 1				Scale 2			
	Ratio	cents	Just major third	Just fifth	Ratio	cents	Just minor third	Just fifth
C	1/1	0.00	✓	✓	1/1	0.00	✓	✓
$C^{\#}$	16/15	111.73	✓	✓	25/24	70.67	✓	
D	9/8	203.91	✓		10/9	182.40	✓	✓
$D^{\#}$	6/5	315.64	✓		6/5	315.64		
E	5/4	386.31		✓	5/4	386.31	✓	✓
F	4/3	498.04	✓	✓	4/3	498.04	✓	✓
$F^{\#}$	45/32	590.22			45/32	590.22		
G	3/2	701.96	✓	✓	3/2	701.96		
$G^{\#}$	8/5	813.69	✓	✓	8/5	813.69		✓
A	5/3	884.36		✓	5/3	884.36	✓	✓
$A^{\#}$	16/9	996.09		✓	16/9	996.09		✓
B	15/8	1088.27		✓	15/8	1088.27		✓
C	2/1	1200.00	✓	✓	2/1	1200.00	✓	✓

2.3.4 The harmonic series scale

Scales based on the harmonic series, also denoted as spectral scales (Sethares, 2005), were investigated extensively by Rameau (1971). The frequency components of the harmonic series exhibit ratio of whole numbers relationships in the sense that the ratio of the frequencies of any two harmonic components can be expressed as a ratio of whole numbers, e.g.

$$\frac{f_3}{f_2} = \frac{3}{2} \tag{2.13}$$

where f_2 and f_3 denote the frequencies of the second and third harmonic components respectively. Based on the consonant properties of frequency ratios defined by whole numbers, the harmonic series thus represents a natural basis for the definition of a scale.

Fig. 2.4 shows the intervallic relationships defined by the first eight frequency components, i.e. first three octaves, of a harmonic series with A as root or tonic. The figure shows how the harmonic number ratios 2/1, 4/2 and 8/4 define the octave, 3/2 and 6/4 define the perfect fifth and 5/4 defines the major third. The associated scale contains the pitch classes defined by the intervals 1/1, 3/2, 5/4, 7/4 and 2/1. Extending the number of octaves to four and five gives rise to the scales shown in Table 2.4.

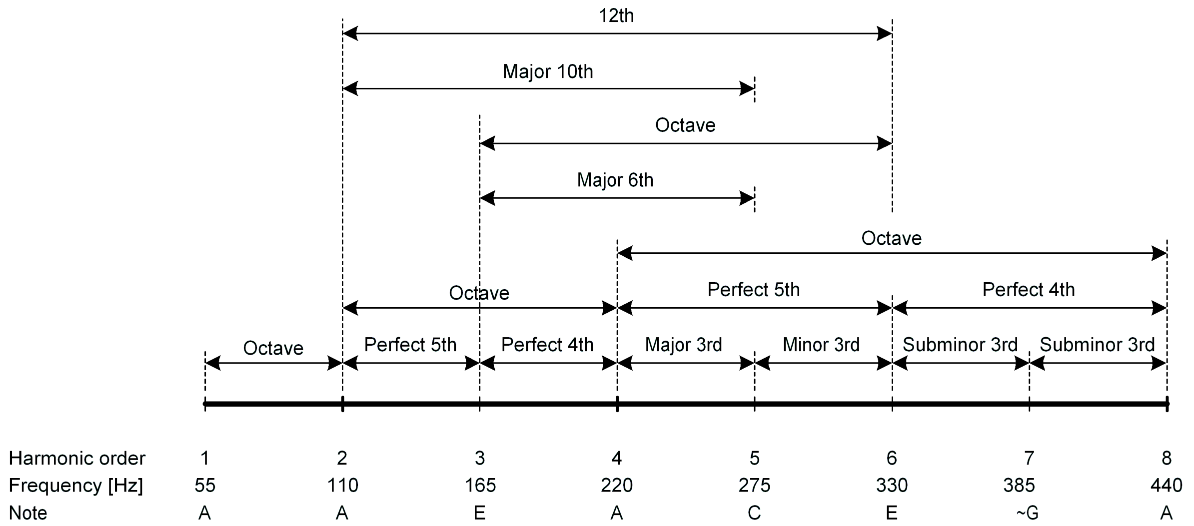


Fig. 2.4 Intervallic relationships defined by the first eight harmonic components of the harmonic series (Rameau, 1971).

Table 2.4 Spectral scale based on four and five octaves of the harmonic series (Sethares, 2005).

Four octaves			**Five octaves**		
Scale degree	**Ratio**	**cents**	**Scale degree**	**Ratio**	**cents**
1	1/1	0.00	1	1/1	0.00
2	9/8	203.91	2	17/16	104.96
3	5/4	386.31	3	9/8	203.91
4	11/8	551.32	4	19/16	297.51
5	3/2	701.96	5	5/4	386.31
6	13/8	840.53	6	21/16	470.78
7	7/4	968.83	7	11/8	551.32
8	15/8	1088.27	8	23/16	628.27
9	2/1	1200.00	9	3/2	701.96
			10	25/16	772.63
			11	13/8	840.53
			12	27/16	905.87
			13	7/4	968.83
			14	29/16	1029.58
			15	15/8	1088.27
			16	31/16	1145.04
			17	2/1	1200.00

2.3.5 Tempered scales

2.3.5.1 Introduction

Temperament refers to the adjustment of individual notes of twelve-tone scale structures to achieve a compromise between the perfect intervals of just intonation and improved possibilities for modulation between different keys. Tempered scales are typically derived from the Pythagorean scale by distributing the Pythagorean comma between the intervals represented on the circle of fifths. Three categories of tempered scales will be considered briefly, namely mean-tone temperament, well temperament and equal temperament.

2.3.5.2 Mean-tone temperament

Mean-tone temperament evolved from the contributions of the fifteenth-century contemporary music theorists Francisco de Salinas and Gioseffo Zarlino (Sadie, 1980). The mean-tone tuning system, used predominantly during the early Renaissance and Baroque periods, enjoys renewed interest due to the work of modern composers such as György Ligeti, who collaborated with Stockhausen in the genre of electronic music, and Douglas Leedy.

The main objective of mean-tone temperaments is to achieve a combination of perfect thirds and acceptable fifths, i.e. optimization of the consonance of triad cords, for a cluster of central keys (Sethares, 2005). This is achieved at the expense of bad thirds and fifths in remote keys. Fig. 2.5 demonstrates the redistribution of the Pythagorean comma in the circle of fifths for two mean-tone temperaments, namely the quarter-comma and sixth-comma mean-tone temperaments. The most common mean-tone temperament, i.e. the quarter-comma temperament attributed to Zarlino (Sadie, 1980), reduces eleven of the perfect fifth 3/2 intervals by one quarter of the Pythagorean comma, and compensates by increasing the remaining interval by seven quarters of the comma. The sixth-comma temperament, possibly due to de Salinas (Sadie, 1980), reduces eleven of the perfect fifth intervals by one sixth of the Pythagorean comma, and increases the remaining fifth by five sixths of the comma.

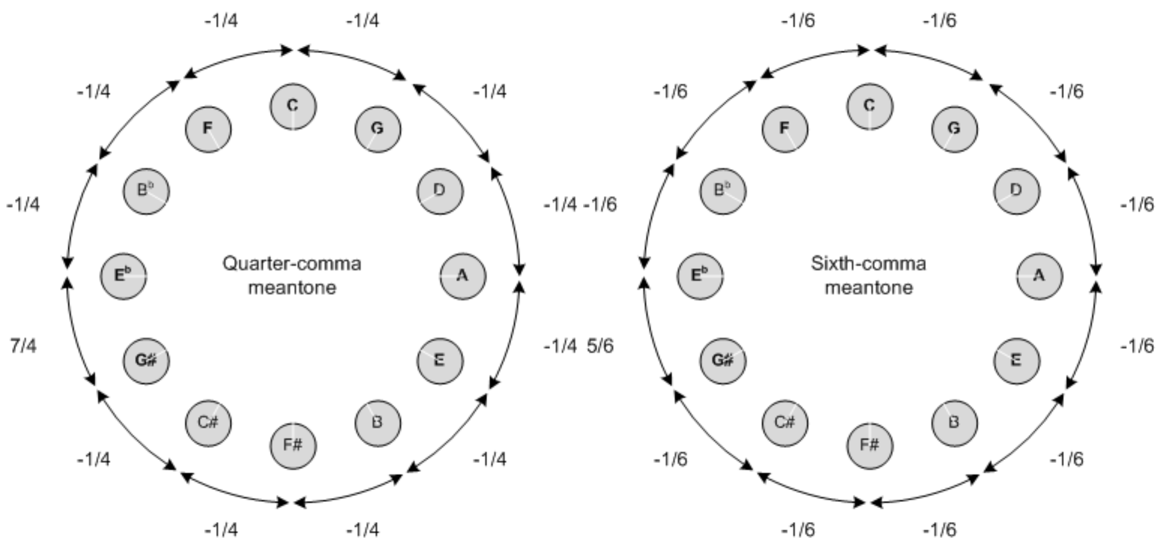


Fig. 2.5 Circle of fifths representations of quarter-comma and sixth-comma mean-tone temperaments (Adapted from Sethares, 2005.

2.3.5.3 Well temperament

Well temperament is distinguished by the fact that the tempered fifth intervals are adjusted unevenly, i.e. to different sizes, in such as manner that none of the keys have very impure intervals. As a result of the uneven adjustments, however, the intervals associated with the keys vary from key to key. Consequently, each key exhibits a unique tone colour or intonation. Bach seeked to demonstrate this property in composing the *Well-Tempered Clavier* (Sethares, 2005; Barnes, 1979). The *Well-Tempered Clavier* features prominently in the performance assessments of a number of key-finding algorithms, including the algorithms proposed by Krumhansl and Schmuckler (Krumhansl, 1990), Chew (2001) and Temperley (1999). Examples of well temperaments include Werckmeister temperament proposed by Andreas Werckmeister, French Temperament Ordinaire, Neidhardt, Kirnberger and the Vallotti temperament invented by Francesco Antonio Vallotti and Young. The circle of fifths representations shown in Fig. 2.6, Fig. 2.7 and Fig. 2.8 for the well temperaments attributed to Rameau (1971), Werckmeister (Sethares, 2005) and Kirnberger (Sethares, 2005) respectively, clearly reflect the uneven fifths that characterise well temperament.

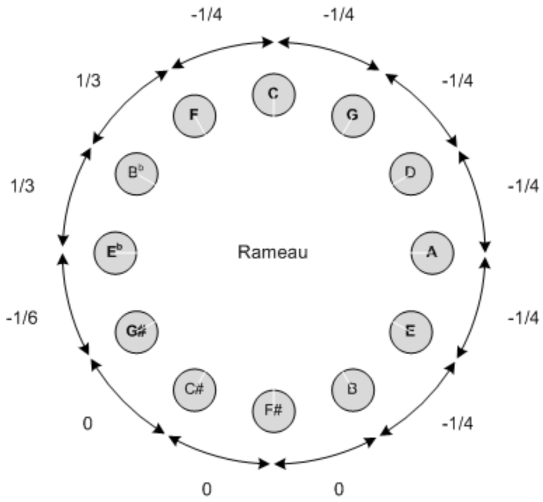


Fig. 2.6 Circle of fifths representation of Rameau's well temperament (Rameau, 1971).

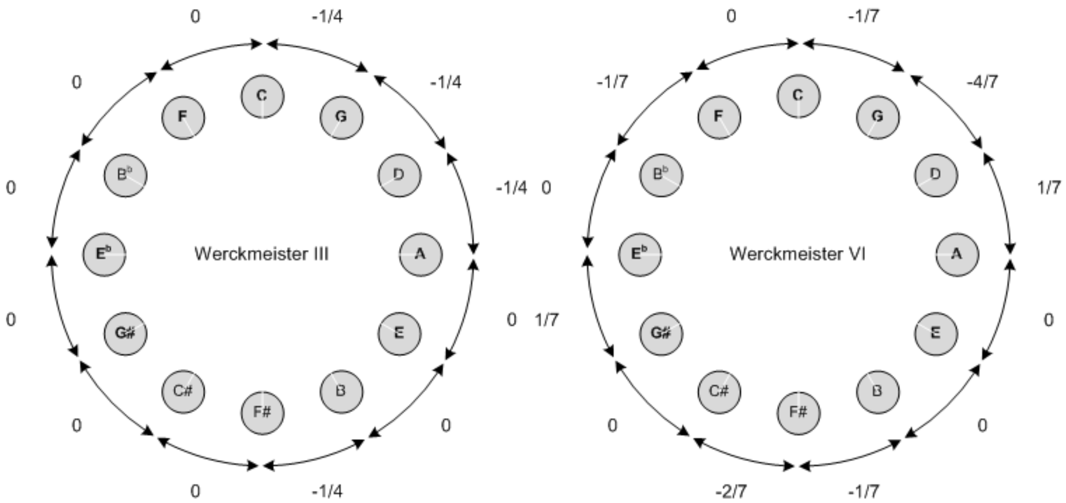


Fig. 2.7 Circle of fifths representations of Werckmeister III and Werckmeister VI well temperaments (Adapted from Sethares, 2005).

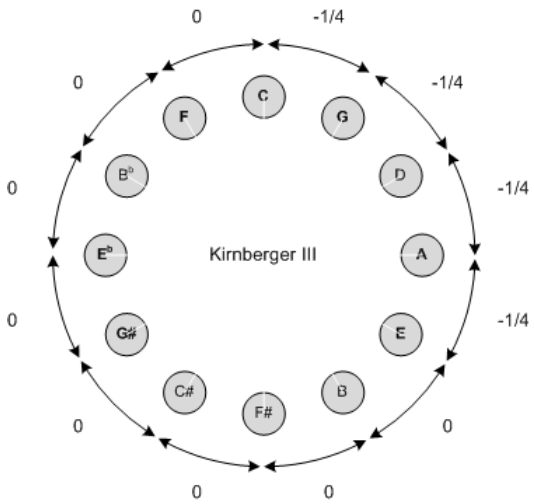


Fig. 2.8 Circle of fifths representation of Kirnberger III well temperament (Adapted from Sethares, 2005).

2.3.5.4 Equal temperament

The main objective of equal temperament is to achieve perfect equality between the intervals associated with the keys of the 12-tone chromatic scale. This guarantees unlimited modulation freedom between the keys. In practice, this translates to equal intervallic distances between the successive notes, as shown in Fig. 2.9, and the requirement that the scale must repeat at the octave. Mathematically, these requirements are expressed by the relationship

$$s = \sqrt[12]{2} \tag{2.14}$$

where s represents the distance between the successive chromatic notes. Equal temperament inevitably implies that the just intervals inherited from the Pythagorean scale, e.g. the perfect fifths, must all be detuned as shown in Table 2.5.

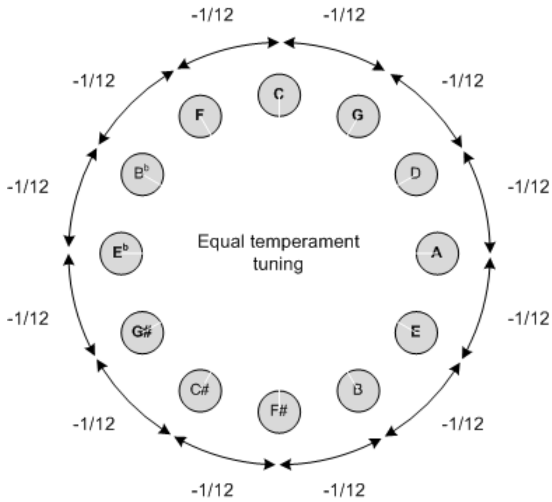


Fig. 2.9 Circle of fifths representation of equal temperament tuning.

Table 2.5 Equal temperament chromatic scale compared to just intonation (Adapted from Wikipedia, the free encyclopaedia, 2011b).

Scale degree	**Mathematical expression**	**Ratio**	**cents**	**Detune compared to just intonation [cents]**
Unison C	$2^{0/12} = 1$	1/1	0	0.00
Minor second C#	$2^{1/12} = \sqrt[12]{2}$	16/15	100	−11.73
Major second D	$2^{2/12} = \sqrt[6]{2}$	9/8	200	−3.91
Minor third D#	$2^{3/12} = \sqrt[4]{2}$	6/5	300	−15.64
Major third E	$2^{4/12} = \sqrt[3]{2}$	5/4	400	+13.69
Perfect fourth F	$2^{5/12} = \sqrt[12]{32}$	4/3	500	+1.96
Augmented fourth F#	$2^{6/12} = \sqrt{2}$	45/32	600	+17.49
Perfect fifth G	$2^{7/12} = \sqrt[12]{128}$	3/2	700	−1.96
Minor sixth G#	$2^{8/12} = \sqrt[3]{4}$	8/5	800	−13.69
Major sixth A	$2^{9/12} = \sqrt[4]{8}$	5/3	900	+15.64
Minor seventh A#	$2^{10/12} = \sqrt[6]{32}$	16/9	1000	+31.17
Major seventh B	$2^{11/12} = \sqrt[12]{2048}$	15/8	1100	+11.73
Octave C	$2^{12/12} = 2$	2/1	1200	0.00

2.4 Experimental investigations of tonality cognition

2.4.1 Overview

The initial investigations and theories on harmony and tonality in tonal music adopted an analytical approach emphasising physical, acoustical and mathematical properties rather than cognitive considerations (Rameau, 1971). In the theories advanced by Rameau (1971), Helmholtz (1954) and Terhardt (1977; 1978), this approach relied predominantly on the concept of the fundamental bass, particularly with regards to the tonality of chords. Subsequent efforts to introduce the listener's perception focused on incorporating the analysis and intuitions of a trained theorist in the analytical approach (Lerdahl and Jackendoff, 1983). Lerdahl and Jackendoff applied linguistic principles to develop a structural representation of music that includes aspects such as temporal grouping, meter, time-span considerations and harmonic tension and relaxation.

Following the experimental work of Shepard (1964), it is now widely accepted that *"the perception of pitch could not be described adequately along a single psychological dimension"* (Butler, 1989). Modern theories on tonality perception and key-finding recognise the actual perceptions of a wider class of listeners and seek to *"explore relationships between analytic systems and the measured responses of a variety of listeners"* (Brown, 1988). As a consequence, models of tonality perception based on experimental results obtained with listeners have since evolved substantially in terms of complexity.

A considerable number of investigations with the view to research the tonality perception of listeners have been conducted over the past two decades. These will be reviewed here, focusing on the results that have impacted on the computational modelling of tonality cognition.

2.4.2 The tonal hierarchy

The development of the probe tone technique (Krumhansl and Shepard, 1979) as a means of evaluating listener's perceptions of tone, given particular tonal context, represents an important contribution to the experimental approaches subsequently used in perception studies. In the initial application, the technique involved establishing the tonal context of C major by sounding an incomplete ascending or descending major scale in C, followed by one of the twelve chromatic tones in the following octave, termed the probe tone. A total of 24 listeners, grouped in three groups according to musical experience and training, rated how well the probe tone completed the scale on a rating scale of 1 to 7. The scale ascended and descended from C3 and C5 respectively, with the last tonic left missing. Although the results exhibited significant differences in the average ratings returned by the individual groups, the outcomes confirmed the following general trends:

- The tonic achieved the highest rating, with the tonic immediately following the pitch class of the context scale rated higher than the tonic at the other end of the probe tone octave.
- The other pitch classes of the major triad, namely the 3rd and 5th achieved the next highest ratings.
- The remaining pitch classes of the diatonic scale achieved the next highest rating, followed by the non-diatonic pitch classes.

The pitch class hierarchy obtained in the Krumhansl and Shepard study is expected based on conventional music theory, including the theories of consonance and dissonance emanating from the physics and mathematics of acoustical signals as contained in the theories of Rameau (1971) and Helmholtz (1954). In a subsequent study, Krumhansl and Kessler (1982) made a further contribution in developing pitch-class profiles derived from empirical data representing the cognitive responses of human listeners. In this study, subjects rated the appropriateness of diatonic probe tones sounded after various key-defining introductions. The experimental procedures applied in the Krumhansl and Kessler experiments can be summarised as follows (Krumhansl, 1990):

- The participants were selected to have a minimum of five years of formal music instruction in the form of instrumental and/or voice lessons. The average figure for the group exceeded this minimum significantly. In order to emphasise perceptive response rather than theoretical knowledge, the participants had no or very limited formal music theory training.
- Complex tones, consisting of sinusoidal components distributed at octave intervals over a five-octave range, were used in order to minimise the effects of differences in pitch height between the key-defining sequences and the probe tones. This shows the relevance of timbre in tonality cognition experiments.
- The key-defining contexts consisted of complete scales, i.e. tonic to tonic scales, tonic triads or I chords and three different chord cadences, i.e. IV V I, VI V I and II V I.

Table 2.6 lists the key profile coefficients for the major and minor modes subsequently proposed by Krumhansl and Kessler (1982), while Fig. 2.10 displays the key profiles graphically. Table 2.7 lists the psychological key profile coefficients for the major and minor modes proposed by Krumhansl and Kessler, while Fig. 2.11 displays the key profiles graphically.

The pitch class hierarchies proposed by Krumhansl and Shepard (1979) and Krumhansl and Kessler (1982) represent quantitative models based on listener perceptions that have found application in numerous research studies on the computational modelling of tonality cognition (Temperley, 1999; Huron and Parncutt, 1993).

Table 2.6 Major and minor mode key profile coefficients proposed by Krumhansl and Kessler (Krumhansl, 1990).

Degree	**I**	**ii**	**II**	**iii**	**III**	**IV**	**v**	**V**	**vi**	**VI**	**vii**	**VII**
Note	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
Major key	0.39	0.14	0.21	0.14	0.27	0.25	0.15	0.32	0.15	0.22	0.14	0.18
Minor key	0.38	0.16	0.21	0.32	0.15	0.21	0.15	0.28	0.24	0.16	0.2	0.19

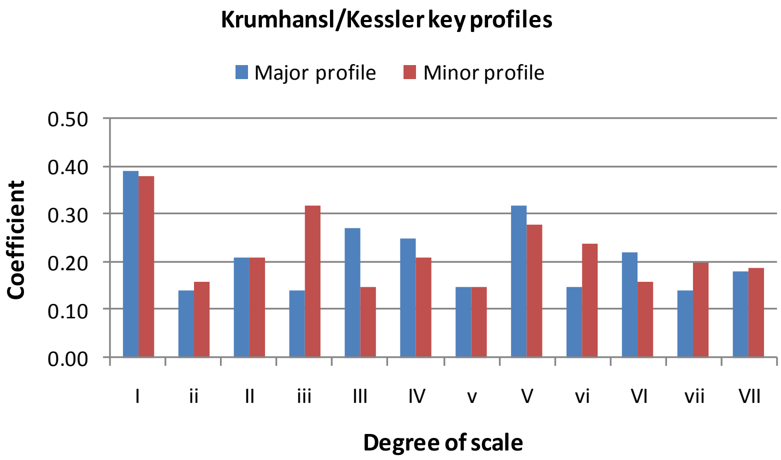


Fig. 2.10 Major and minor mode key profile coefficients proposed by Krumhansl and Kessler (Krumhansl, 1990).

Table 2.7 Major and minor mode psychological key profile coefficients proposed by Krumhansl and Kessler (Krumhansl, 1990).

Degree	**I**	**ii**	**II**	**iii**	**III**	**IV**	**v**	**V**	**vi**	**VI**	**vii**	**VII**
Note	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
Major key	6.35	2.23	3.48	2.33	4.38	4.09	2.52	5.19	2.39	3.66	2.29	2.88
Minor key	6.33	2.68	3.52	5.38	2.60	3.53	2.54	4.75	3.98	2.69	3.34	3.17

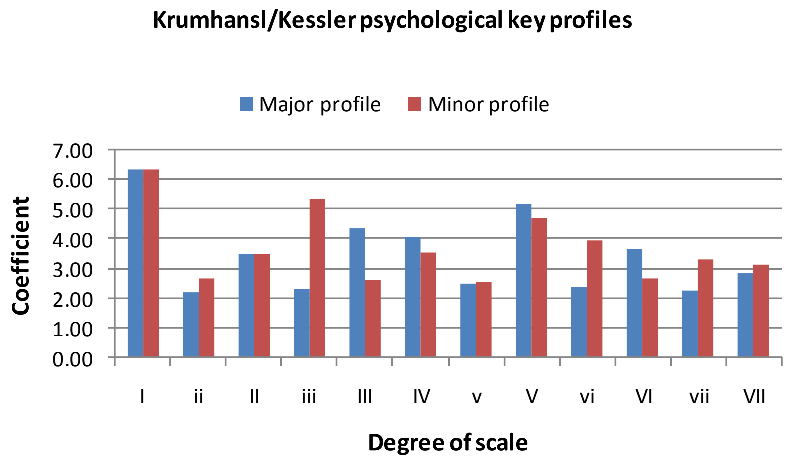


Fig. 2.11 Major and minor key psychological key profiles proposed by Krumhansl and Kessler (Krumhansl, 1990).

2.4.3 Tonal closure

The tone profile hierarchy approach to tonality represents a static approach in the sense that the temporal context, i.e. the sequential context created by tone progressions, is not represented. Contemporary literature reports on a number of experiments that investigates this temporal aspect. In the experiment conducted by Cook (1987), listeners judged the *"tonal closure or sense of completion"* of passages selected from performed piano works or movements, as well as modified versions thereof that end in a key other than the original tonic. The results of the study can be summarised as follows (Cook, 1987; Huron and Parncutt, 1993):

- *Passages shorter than 90 seconds:* Listeners were able to detect when the tonal centre associated with the end of the passage differed from the tonal centre associated with the beginning of the passage.
- *Passages longer than 2 minutes:* Listeners were unable to detect when the tonal centre associated with the end of the passage differed from the tonal centre associated with the beginning of the passage, leading Cook to conclude that the long-term key memory is quite weak in the absence of absolute pitch.

The results obtained by Cook (1987) confirmed that the *"more recent sonorities have a greater influence on key determination than sonorities from the distant past"* (Huron and Parncutt 1993), thereby introducing the concept of echoic memory.

2.4.4 Tonal relationships in musical context

Brown (1988) conducted an experiment using 42 musically educated listeners to investigate tonality perception with the view to consider *"both formal analytic systems and musical listeners' responses to tonal relationships in musical contexts"*. The musical material for the experiments was derived by extracting a total of nine short excerpts representing non-modulating diatonic and chromatic harmonic progressions from piano pieces. The pitch class content of each of these excerpts was then manipulated according to a complex set of guidelines, including rules such as avoiding the duplication of pitch-classes in the stimuli, with the view to derive three pitch strings that represent the following cases:

- The pitch string evokes the same tonal centre as the original excerpt.
- The pitch string evokes a different tonal centre compared to the original excerpt.
- The pitch string evokes an ambiguous tonal centre.

The original excerpts and pitch strings devised by Brown for the experiment is presented in Appendix A (Brown, 1988). The following results for the excerpt labelled 5M in Appendix A, taken from movement three of Schubert's *Sonata in D Major, D. 664*, illustrate the nature of the results obtained in the study (Butler, 1989):

- A total of 86% of the listeners identified the tonal centre of the excerpt correctly as D major.
- For the pitch string designed to evoke the same tonal centre as the original excerpt, an increased percentage of 95% identified the tonal centre as D major.
- For the pitch string designed to evoke a different tonal centre compared to the original excerpt, namely G major in this case, 41% persisted with D major as the tonal centre while 45% identified G major as the tonal centre.
- For the pitch string designed to evoke an ambiguous tonal centre, 45% identified D major as the tonal centre. Seven other pitches, however, were also nominated, including three that are not represented in the tonal pattern.

Based on the results obtained in the study, Brown (1988) identified two approaches to the perception of tonality by human listeners, namely a structural approach and a functional approach. These approaches serve as a good framework for interpreting the hierarchy of modern tonality perception models, and can be summarised as follows:

- The structural approach involves determining the key of a musical passage based on the cumulative properties of the pitch content, i.e. the prevalence of pitch-classes and the durations thereof. Although distant pitches on the timeline are regarded as less significant due to the decay of sensory memory, the local ordering of pitch classes is not considered in the structural approach (Huron and Parncutt, 1993). The geometrical models proposed by Shepard (1982a; 1982b), Krumhansl (1979; 1986) and Krumhansl and Shepard (1979) are all examples of the structural approach to the modelling of tonality. The pitch class template, or tonal hierarchy, introduced by Krumhansl and Kessler (1982) represents the most well-known example of the structural approach in modern tonality cognition studies. Krumhansl (1990) demonstrated that tonal centre or key can be determined effectively using correlations of distributions of pitch class durations with key profile templates determined experimentally. Despite the successes achieved with the tonal hierarchy theory, however, it is accepted that it is *"not sufficiently sensitive to the perceptual activity of identifying, confirming, and revising one's cognitive awareness of pitch relationships from one musical moment to the next"* (Butler, 1989).
- The functional approach, in contrast to the structural approach, emphasises the sequential or contextual properties of the material, i.e. the intervallic properties of the pitch-class transitions, while pitch-class content is considered of lesser significance. This approach is represented by the contributions of Butler (1983; 1989) and Brown (1988), who showed that the order of pitches classes impacts significantly on the perception of tonality by human listeners (Huron and Parncutt, 1993).

Experimental results and the models of tonality perception and key-finding proposed to date indicate that both approaches are of importance in tonality perception and that these approaches should be regarded as complementary rather than contradictory in the quest for improving the modelling of tonality perception (Huron and Parncutt, 1993).

2.4.5 Tonal centre in the context of chords

In an investigation to test listener perception of tonal centre or key in the context of chords or chord progressions, Krumhansl and Kessler (Krumhansl, 1990) designed a collection of ten chord progressions to use with the probe-tone technique. These chord progressions, each consisting of a succession of nine triad chords, were designed according to the following guidelines:

- The first three chords form a cadence in the initial key, with the first chord II or IV, followed by V and ending in I.
- The fourth chord is a diatonic triad in the initial key.
- The fifth chord is a diatonic triad in the both the initial and final key.
- The sixth chord is a diatonic triad in the second key.
- The last three chords form a cadence in the final key, ending with a V-I progression.

The chord progressions fall in two categories, i.e. progressions considered consistent with a single key and progressions considered to modulate from an initial key to another final key. The test procedure involved sounding a section of the chord progression, starting with the first chord, followed by one of the diatonic probe tones. Listeners rated how well the probe tone fits with the preceding chord. The process was repeated for all twelve probe tones and all possible sections, each consisting of one to nine successive chords, of the ten chord progressions. Listener ratings were compared with the major and minor psychological key profiles obtained in an earlier study (Krumhansl and Kessler, 1982).

2.4.6 Two-tone sequences in tonal context

The tone ratings obtained by Krumhansl (1990) in the probe-tone experiments showed asymmetrical values for identical intervals. In order to investigate the effects of temporal order on the tone ratings, Krumhansl devised an experiment where six musically trained listeners were presented with a tonal context followed by two probe tones. Listeners were required to judge on a seven point scale how well the second tone relates to the first tone in the context provided. The following contexts were used:

- Ascending and descending scales in C major.
- Ascending and descending melodic scales in C minor.
- An I-IV-V-I chord cadence in $F^{\#}$ major.
- An i-iv-V-i chord cadence in $F^{\#}$ minor.

All possible probe-tone combinations of the twelve pitch classes were tested. For probe-tone combinations ending with the same tone, the results correlated well with the tonal hierarchy found by Krumhansl and Kessler (1982). This shows a preference for intervals that end on tones that rate high in the Krumhansl and Kessler tonal hierarchy, irrespective of the pitch class of the first tone. Further analysis indicated that the temporal-order preferences of the listeners also depend on the tonal hierarchy (Krumhansl, 1990). Krumhansl concluded as follows: *"Perceived pitch relations are a joint function of certain invariant structural features"* and *"a contextually-dependent component - the tonal hierarchy"* (Krumhansl, 1990).

2.5 Durational accent in music perception analysis

The concept of durational accent features prominently in the realm of music perception. The statistical analysis algorithms implemented in the MIDI toolbox, for instance, applies durational accent in determining the distributions of pitch intervals and pitch classes. Parncutt (2003) identifies two classes of durational accent, namely pre-durational and post-durational accent, where pre-durational accent is associated with the concept of echoic memory while post-durational accent is associated with the concept of working memory. Echoic store or echoic memory, a term introduced by Neisser (1967), is a neurological concept related to the short-term retention or buffering of auditory information before it is categorised and subjected to further processing and was originally applied in the field of speech processing (Massaro, 1970). Echoic memory is regarded as independent of conscious attention and is typically assigned durations in the range 0.5 to 2 seconds, while working memory is associated with conscious attention and is assigned durations of the order of several seconds. Durational accent, as used in the remainder of this document, refers to the pre-durational accent associated with echoic memory.

Quantitative models for deriving a measure of the durational accent, or importance from a music perception perspective, of note events are rooted in research aimed at determining how musical rhythms and metrical structure are perceived by listeners. A number of researchers developed models for the perception of rhythm and metrical structure, including Longuet-Higgins and Steedman (1971), Longuet-Higgins and Lee (1982), Povel and Essens (1985), Lee (1991) and Rosenthal (1992). This was followed by a quantitative model to account for the perceptual significance of a pulse sensation, termed pulse salience (Parncutt, 1994).

Parncutt's model embodies the concept that the salience of pulse sensations depends on both tempo and rhythmic pattern. The model is based on experimental data obtained for a number of listeners subjected to a set of rhythms with the view to investigate the effects of rhythmic pattern and tempo on periodic grouping (Parncutt, 1994). Listeners were required to tap a beat pattern in response to a stimulus represented by a rhythmic pattern played at a particular tempo. Six rhythmic patterns were used in the experiment, namely pulse, waltz, march, swing, skip and cross. Each pattern was evaluated for six logarithmically spaced playback tempos, ranging from 50 to 400 events per minute, and the responses of the listeners were recorded for analysis. In an accompanying experiment, listeners were tested to investigate the perception of metrical accent.

Based on the experimental results obtained, Parncutt developed a quantitative model for predicting metrical accent from pulse salience. The model assigns a weight, termed phenomenal accent (Lerdahl and Jackendoff, 1983), to each event in the rhythmic cycle. The phenomenal accents are determined by a number of parameters, including the Inter-Onset Interval (IOI) between events, loudness, timbre, pitch and combinations thereof (Parncutt, 1994). The IOI is represented in the model by a durational accent value and is regarded as the most important contributor to phenomenal accent (Povel and Okkerman, 1981). Based on the argument that "*durational accent increases with IOI for small values of IOI and saturates as IOI approaches and exceeds the duration of the echoic store (auditory sensory memory)*", Parncutt developed the following mathematical relationship between durational accent D_a and the IOI of successive events:

$$D_a = \left(1 - e^{-\frac{IOI}{\tau}}\right)^{I_a} \qquad (2.15)$$

where τ denotes the *saturation duration* in seconds and I_a denotes the *accent index* (Parncutt, 1994). The *saturation duration* is assumed to be proportional to the echoic store while the *accent index* accounts for the minimum IOI that can be perceived by the listener. Calculations aimed at obtaining a good fit between predictions using Parncutt's model with the experimental results obtained in the study, suggest typical values of 0.5s and 2 for the *saturation duration* and *accent index* respectively (Parncutt, 1994). Fig. 2.12 shows the variation of durational accent with *saturation duration* for a fixed *accent index* of 2. Fig. 2.13 shows the variation of durational accent with *accent index* for a fixed *saturation duration* of 0.5. The curves reflect the following:

- Lower *saturation duration* values give rise to higher durational accents for short IOIs and saturated durational accents of the order of unity for longer IOIs.
- Higher *saturation duration* values give rise to lower durational accents for short IOIs and saturated durational accents for longer IOIs.
- Lower *accent index* values give rise to higher durational accents for short IOIs and saturated durational accents for longer IOIs.

- Higher *accent index* values give rise to lower durational accents for short IOIs and saturated durational accents for longer IOIs.

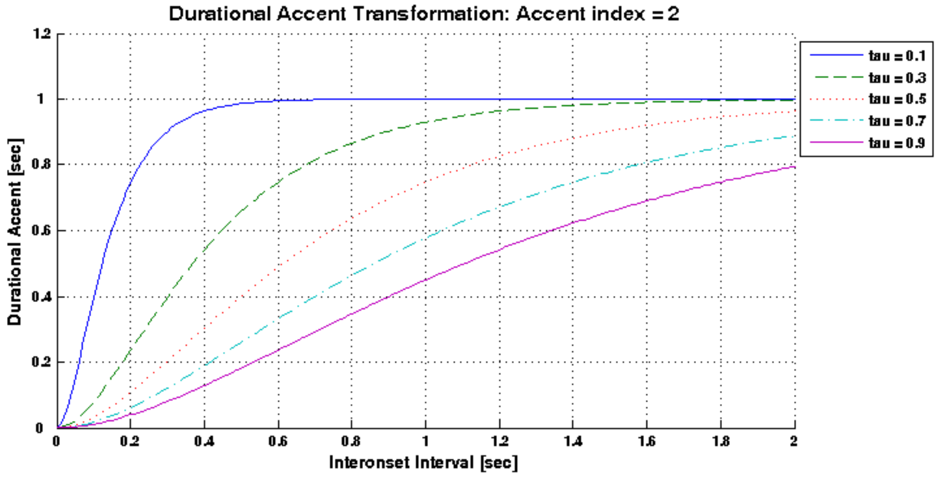


Fig. 2.12 Durational accent as a function of saturation duration for a fixed accent index of 2.

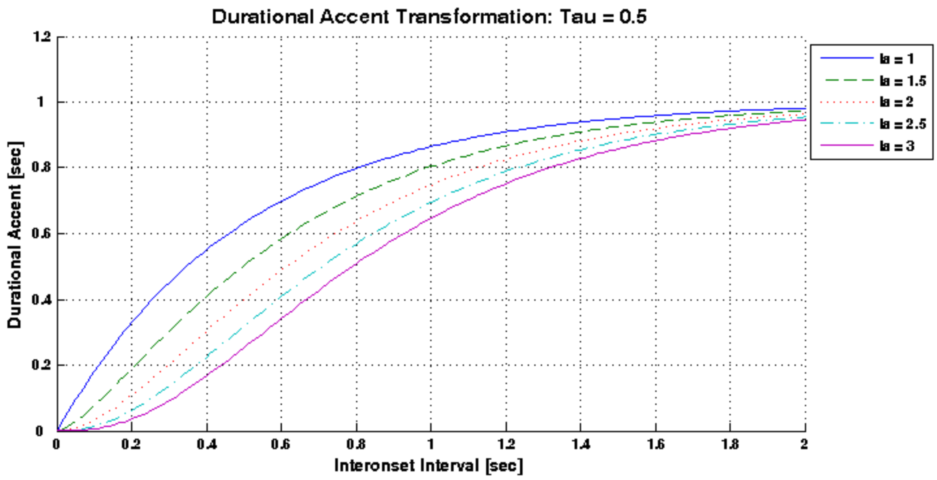


Fig. 2.13 Durational accent as a function of accent index for a fixed saturation duration of 0.5.

Decreasing the *saturation duration* accentuates the importance of changes in IOI for shorter IOI values while reducing it for longer IOIs. Increasing the *saturation duration* reduces the importance of changes in IOI for both shorter and longer IOI values.

Durational accent has been accepted as an important parameter for quantifying pulse salience in the context of perceptual models of rhythm and metrical structure. It has also been incorporated as an integral part of tonality cognition models following the investigations by Cook (1987), who published experimental evidence of a link between the perception of tonality and memory. Huron and Parncutt (1993) have subsequently shown that the accuracy of the Krumhansl and Schmuckler key-finding algorithm (Krumhansl, 1990) can be improved significantly by incorporating the effects of echoic memory.

The link between durational accent and tonality cognition is incorporated in the algorithms implemented for statistical analysis in the MIDI toolbox by weighing the occurrences of pitch intervals and pitch classes by durational accent using Parncutt's formula. In all of these cases, the durational accent is calculated based on the durations of note events rather than the IOIs of successive events.

2.6 Pitch perception

The ability to perceive pitch and the mechanisms involved in pitch perception ultimately establishes the boundaries for the higher order process of tonality perception, i.e. interpreting pitch class content and temporal intervallic information in musical context with reference to considerations such as consonance and dissonance, harmony and tonal centre. The model of pitch perception proposed by Terhardt (1979) represents an important advance in efforts to quantify the psycho acoustical effects associated with complex acoustical signals. Huron and Parncutt (1993) offered a good hierarchical explanation of Terhardt's model, identifying three stages in the analysis process:

- *Analysis of the input signal to determine the audibility of the different frequency components:*

  This process incorporates three aspects, namely the effect of masking of the audibility of a spectral component by the other spectral components present, the saturation effect experienced for the audibility of a pure tone for increased Sound Pressure Level (SPL) and the variation in sensitivity of listeners to the pitch of a pure tone. The process yields a *spectral pitch weight* that reflects the relative audibility or relative salience for each pure tone component in the spectrum (Terhardt, 1979; Huron and Parncutt, 1993).

- *Harmonic analysis of the frequency components with the view to identify complex tones and determine the corresponding fundamental frequencies:*

  The complex tones are identified by means of a pattern recognition approach using a harmonic pitch-pattern template. For each complex tone, the process yields a *virtual pitch* representing the fundamental of the associated harmonic series. Each virtual pitch is assigned a weight that reflects how well the complex tone fits the harmonic series template.

- *Combination of the pitch components to yield a single spectrum that represents the subjective experience of the listener:*

  This process yields a *"composite portrait of all pitches which could be perceived"* (Huron and Parncutt, 1993) composed on the basis of the highest pitch weights assigned in stage two.

Terhardt's model is designed to operate on acoustical signals, hence the requirement to perform spectral analysis as part of the process. It can, however, also be applied to symbolic music analysis, as shown by Huron and Parncutt (1993) for the pitch class profile model of tonality perception proposed by Krumhansl and Kessler (1982).

2.7 Key-finding algorithms

2.7.1 Introduction

Key-finding algorithms play a central role in tonality cognition, particularly with reference to determining tonal centre and key modulations. Early key-finding algorithms are based purely on music theoretical considerations while the more recent algorithms make use of experimental results of cognitive studies in tonality perception. The body of historical research material on tonal analysis is vast and extends over the entire history of music analysis, reflecting a progression involving the principles of physics, formal music theoretical analysis and, more recently, cognitive considerations to varying degrees. The following key-finding models and methodologies, however, are of particular importance for this study:

- *The shape-matching algorithm proposed by Longuet-Higgins and Steedman (1971)*:

  Longuet-Higgins and Steedman devised a procedural or preference-rule model (Temperley, 1999) that gravitates to a resolution by eliminating certain key options.

- *The key profile template-based algorithm proposed by Krumhansl and Schmuckler (Krumhansl, 1990)*:

  The Krumhansl and Schmuckler key-finding algorithm identifies the most likely key based on the correlation between a vector of weighted pitch classes, derived using linear weighing of the note durations of the pitch classes in the sample material, and a vector of psychological key profiles derived from tonality perception experiments.

- *The modified Krumhansl and Schmuckler algorithms proposed by Temperley (1999)*:

  Temperley introduced two variations of the Krumhansl and Schmuckler key-finding algorithm. These variations involved parametric rather than algorithmic modifications that can be categorised as follows:

  - A modified set of key profiles and a simplified key correlation formula.
  - A modified set of key profiles, simplified key correlation formula and flat, i.e. unweighted, vector of pitch classes.

- *The modified Krumhansl and Schmuckler algorithm proposed by Huron and Parncutt (1993)*:

  Huron and Parncutt introduced pitch saliency and echoic memory to the pitch classes hierarchy used in the Krumhansl and Schmuckler algorithm.

- *The preference rule algorithm proposed by Temperley (1999)*:

  Temperley proposed a new algorithm that combines aspects of the Longuet-Higgins and Steedman procedural approach with the Krumhansl and Schmuckler structural approach.

The remainder of this section briefly summarises the most relevant characteristics of the above approaches. The presentation also aims to impose a relational hierarchy on the models and methodologies in view of the objective of developing a software application that is representative of the current computational models of tonality analysis.

2.7.2 The shape-matching key-finding algorithm proposed by Longuet-Higgins and Steedman

In an important early contribution to the development of an automatic key-finding methodology Longuet-Higgins and Steedman (1971) proposed a shape-matching algorithm based on the two-dimensional harmonic network space. This space is characterised by horizontal intervals of perfect fifths and vertical intervals of major thirds between the keys. The major and minor modes are represented by the shapes shown in Fig. 2.14 for C Major and A minor respectively, while the key is determined by the location of the shape in the network. By regarding the key locations as diatonic scale tones, pitch events are mapped successively to the harmonic network space. For each successive event the shapes, and thus the corresponding keys, that do not cover the event are eliminated. The key is identified by repeating the process until a single key remains. The algorithm makes provision for the following special cases:

- *Multiple solutions:* In this case the tonic-dominant rule is invoked. This rule makes use of the assumption that the first event of the progression is likely to be the tonic or dominant degree of the actual key. If a multiple solution is reached, the tonic rule is applied, thereby selecting the key of which the first tone is the tonic, provided that this key is one of the multiple candidates. If not, the dominant rule is applied, thereby selecting the key of which the first note is the dominant, provided that this key is one of the candidates.
- *Elimination of all keys*: In this case the algorithm returns to the previous stage and the tonic-dominant rule is invoked.

Longuet-Higgins and Steedman evaluated the performance of the above key-finding algorithm for the forty-eight fugues of the *Well-tempered Clavier* (WTC) composed by Bach. The results can be summarised as follows:

- The shape-matching component by itself, i.e. with invoking the tonic-dominant rule, of the algorithm converged to the correct solution for twenty-six of the fugues, representing a success rate of 54.2%.
- The shape matching process eliminated all keys in five cases. The correct designated keys were identified in all of these cases by invoking the tonic-dominant rule.
- The shape-matching process yielded multiple solutions in seventeen cases. The correct designated keys were also identified in all of these cases by invoking the tonic-dominant rule.

Overall, the algorithm yielded a success rate of 100% for the fugue subjects of Bach's *Well-tempered Clavier*. However, the results suggest that the algorithm will perform relatively poorly for material where the tonic-dominant rule fails.

The shape-matching algorithm proposed by Longuet-Higgins and Steedman (1971) relies on an unweighted or flat pitch class template. This implies that all pitches belonging to the diatonic scale are weighted equally, while non-diatonic notes are allocated a weight of zero. This is in contrast to the more advanced algorithms proposed subsequently, such as the weighted approach proposed by Krumhansl and Schmuckler (Krumhansl, 1990), which furthermore assigns non-zero weights to the non-diatonic pitch-class elements. Some aspects of the Longuet-Higgins and Steedman algorithm, such as the flat pitch class template, have found application in recent models, e.g. the preference rule model proposed by Temperley (1999).

E	B	$F^{\#}$	$C^{\#}$	$G^{\#}$	$D^{\#}$	$A^{\#}$	$E^{\#}$	$B^{\#}$
C	G	D	A	E	B	$F^{\#}$	$C^{\#}$	$G^{\#}$
A_b	E_b	B_b	F	C	G	D	A	E
F_b	C_b	G_b	D_b	A_b	E_b	B_b	F	C
D_{bb}	A_{bb}	E_{bb}	B_{bb}	F_b	C_b	G_b	D_b	A_b

C Major

E	B	$F^{\#}$	$C^{\#}$	$G^{\#}$	$D^{\#}$	$A^{\#}$	$E^{\#}$	$B^{\#}$
C	G	D	A	E	B	$F^{\#}$	$C^{\#}$	$G^{\#}$
A_b	E_b	B_b	F	C	G	D	A	E
F_b	C_b	G_b	D_b	A_b	E_b	B_b	F	C
D_{bb}	A_{bb}	E_{bb}	B_{bb}	F_b	C_b	G_b	D_b	A_b

A Minor

Fig. 2.14 Major and minor shapes proposed by Longuet-Higgins and Steedman (1971).

2.7.3 The template-based key-finding algorithm proposed by Krumhansl and Schmuckler

Krumhansl and Schmuckler (Krumhansl, 1990) developed a key-finding algorithm based on a weighted pitch-class template, as opposed to the flat pitch-class template employed by Longuet-Higgins and Steedman (Temperley, 1999). The algorithm is based on the Krumhansl and Kessler (1982) psychological key profile hierarchy given in Table 2.7 and Fig. 2.11. The algorithm can be summarised as follows:

- The durations, specified in terms of the number of beats, of individual pitch events in the subject material are mapped cumulatively to the twelve pitch-classes. This implies that the absolute pitch or octave in which the events are sounded is disregarded. This yields a vector $\mathbf{D} = [d_i]$, where $i = 1 \text{ to } 12$, representing the cumulative durations for each of the twelve pitch-classes.

- By circular shifting of the key profiles given for C major and C minor in Table 2.7, a psychological key profile $\mathbf{K} = \left[k_j \right]$ where $j = 1$ to 24 is derived, representing the twenty-four keys and modes.
- The correlation coefficient τ_j between the vector of cumulative pitch-class durations **D** and the j[th] key profile vector is $\mathbf{K}_j$ calculated. This yields a value for each of the twenty-four possible key and mode combinations. The correlation coefficient τ_j reflects a measure of the strength with which the j[th] key is represented by the distribution of pitch-class durations of the sample material. Mathematically, the process can be represented by the relationship

$$\tau_j = \frac{\sum_{i=1}^{12} \left(d_i - \overline{d}\right)\left(k_{ij} - \overline{k_j}\right)}{\sqrt{\sum_{i=1}^{12} \left(d_i - \overline{d}\right)^2 \sum_{i=1}^{12} \left(k_{ij} - \overline{k_j}\right)^2}} \tag{2.16}$$

  where $\overline{d}$ and $\overline{k_j}$ denote the means of d_i and k_{ij} respectively for $i = 1$ to 12.

Krumhansl (1990) evaluated the performance of the proposed key-finding algorithm extensively. The evaluation strategy considered a range of compositions by different composers, including the 48 preludes of Bach's *Well-tempered Clavier*, the 24 preludes of Shostakovich, the 24 preludes of Chopin and the 48 fugue subjects of Bach's *Well-tempered Clavier*. The performance results were compared with the findings obtained by other researchers, including Cohen (1977), Longuet-Higgins and Steedman (1971) and Holtzman (1977). Various implementation strategies were considered, such as applying the algorithm for a short initial segment of the test material, applying the algorithm for a progressively increasing segment of the material and applying the algorithm on a measure-by-measure basis with the view to track key strength along the temporal axis. Table 2.8 summarises the main aspects of the performance evaluation strategies applied by Krumhansl.

The performance results of the Krumhansl and Schmuckler algorithm for the various case studies listed in Table 2.8 can be summarised as follows (Krumhansl, 1990):

- *Key-finding performance for the first four notes of the forty-eight preludes of Books I and II of the Well-Tempered Clavier composed by Bach:*

  The algorithm yielded the designated key for forty-four of the preludes, thereby identifying the designated key with a success rate of 91.7%. Krumhansl (1990) shows that the algorithm assigns closely related keys, according to the key regions charted by Schoenberg (1969), in the four incorrect cases. The parallel major of the designated key is assigned in two cases, and the relative minor of the dominant of the designated key is assigned in the remaining two cases. The results of this study are summarised in Appendix B.

  Krumhansl (1990) further compared the performance of the key-finding algorithm with results obtained by Cohen (1977). Cohen evaluated the ability of listeners to identify the designated key of twelve preludes selected from Book I of the *Well-Tempered Clavier*. Listeners were presented with the first four tones from each piece, counting tones sounded simultaneously as a single tone. The Krumhansl and Schmuckler algorithm outperformed Cohen's listeners significantly, yielding a success rate of 100% compared to the average success rate of 75% returned by listeners.

- *Key-finding performance for the first four notes of the twenty-four preludes composed by Shostakovich:*

The algorithm yielded the designated key for 17 of the twenty-four preludes, thereby identifying the designated key with a success rate of 70.8%. For the seven incorrect results, the keys identified by the algorithm were relatively close in terms of the Schoenberg maps to the designated keys. Confusion of major and minor keys represented four of the incorrect results. In two of the remaining incorrect cases, the input vector contained only one or two different notes, i.e. very limited information. Krumhansl related the cause of the error in the final incorrect case to relatively long third and fifth scale degrees in the input vector, causing the algorithm to assign the parallel major of the designated minor key. The results of this study are summarised in 6.1.2Appendix C.

- *Key-finding performance for the first four notes of the twenty-four preludes composed by Chopin:*

The algorithm yielded the designated key for 11 of the twenty-four preludes, thereby identifying the designated key with a success rate of 48.8%. In the majority of the incorrect cases, the algorithm identified the correct key region. Krumhansl (1990) points out that the tonality of the preludes composed by Chopin is much more ambiguous than is the case with the preludes of Bach and Shostakovich. This partly explains the much weaker performance achieved for Chopin's compositions. The results of this study are summarised in Appendix D.

- *Key-finding performance in terms of the number of tones required to identify the designated key for the forty-eight fugues of Books I and II of the Well-Tempered Clavier composed by Bach:*

This test involved determining the number of tones required by the Krumhansl and Schmuckler algorithm and the Longuet-Higgins and Steedman shape-matching algorithm respectively to identify the designated key for the forty-eight fugues of Bach's *Well-tempered Clavier*. The results are summarised in Appendix E. The Krumhansl and Schmuckler algorithm generally required fewer steps to find the key compared to the Longuet-Higgins and Steedman algorithm, i.e. an average of 5.1 steps versus 9.42 steps. The Longuet-Higgins and Steedman algorithm invoked the tonic-dominant in twenty-two cases, and identified the correct designated key successfully in each case. This represents an overall success rate of 100%. The Krumhansl and Schmuckler algorithm invoked the tonic-dominant rule in four cases, and identified the correct designated key successfully in three of these. This represents forty-four of the preludes, thereby identifying the designated key with a success rate of 91.7%. Krumhansl (1990) showed that the algorithm assigns closely related keys in the four incorrect cases. The parallel major of the designated key is assigned in two cases, and the relative minor of the dominant of the designated key is assigned in the remaining two cases.

Table 2.8 Performance evaluation strategies applied for the Krumhansl and Schmuckler key-finding algorithm.

	Methodology	**Subject**	**Comparison**	**Success rate**
1	Algorithm is applied to the initial four-note segment of the passage.	Bach: 48 preludes from the Well-tempered Clavier	12 preludes from Book I, Listener responses presented by Cohen (1977)	91.7%
		Shostakovich: 24 preludes	No comparison	70.8%
		Chopin: 24 preludes	No comparison	45.8%
2	The algorithm is applied to an initial section of the passage, which is increased in length on a note by note basis until the entire subject is included. The point, number of tones, at which the correct key is derived is determined.	Bach: 48 fugues from the Well-tempered Clavier	Longuet-Higgins and Steedman (1971) Holtzman (1977)	
		Shostakovich: fugues	No comparison	
3	Algorithm is applied on a measure-by-measure basis.	Bach: Prelude no. 2	Analysis performed by music theory experts.	

2.7.4 The modified Krumhansl and Schmuckler key-finding algorithms proposed by Temperley

Temperley (1999) conducted a thorough evaluation of the Krumhansl and Schmuckler key-finding algorithm and subsequently proposed modifications to the key profile coefficients and the mathematical definition of the correlation function. The main performance assessment was conducted on a measure-by-measure basis for the first half, i.e. first 40 measures, of the *Courante* from Bach's *Cello Suite in C major (BWV 1009)*. Using classical music theoretical principles applied to each measure in isolation, Temperley derived a key set consisting of one or more keys deemed likely candidates for the tonal centre of each of the measures. These then served as a benchmark with which to compare the keys estimated with the algorithms.

Temperley found that the Krumhansl and Schmuckler algorithm identified the key correctly for 13 out of 40 measures, yielding a success rate of 27 out of 40, i.e. 67.5%, which is considerably lower compared to the success rate reported by Krumhansl for the first four notes of the Bach preludes. Temperley concluded that the discrepancy in performance between the results for the Bach preludes and the Bach *Courante* is due to the fact that a significant number of the preludes begin by *"outlining or elaborating a tonic triad"* compared to the wider *"variety of melodic and harmonic situations"* provided by the Bach *Courante* (Temperley, 1999). Based on a music theoretical analysis of the measures where the Krumhansl and Schmuckler algorithm predicted the key incorrectly for the Bach *Courante*, Temperley proposed modified key profile coefficients for the Krumhansl and Schmuckler key-finding algorithm. Table 2.9 lists the modified key profile coefficients proposed by Temperley for the major and minor modes of the diatonic scale, while Fig. 2.15 displays the key profiles graphically. Fig. 2.16 and Fig. 2.17 compare the key profiles proposed by Krumhansl and Kessler (1982) with the modified profiles proposed by Temperley for the major and minor modes respectively.

Table 2.9 Major and minor mode key profile coefficients proposed by Temperley (1999).

Degree	**I**	**ii**	**II**	**iii**	**III**	**IV**	**v**	**V**	**vi**	**VI**	**vii**	**VII**
Note	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
Major key	5.0	2.0	3.5	2.0	4.5	4.0	2.0	4.5	2.0	3.5	1.5	4.0
Minor key	5.0	2.0	3.5	4.5	2.0	4.0	2.0	4.5	3.5	2.0	1.5	4.0

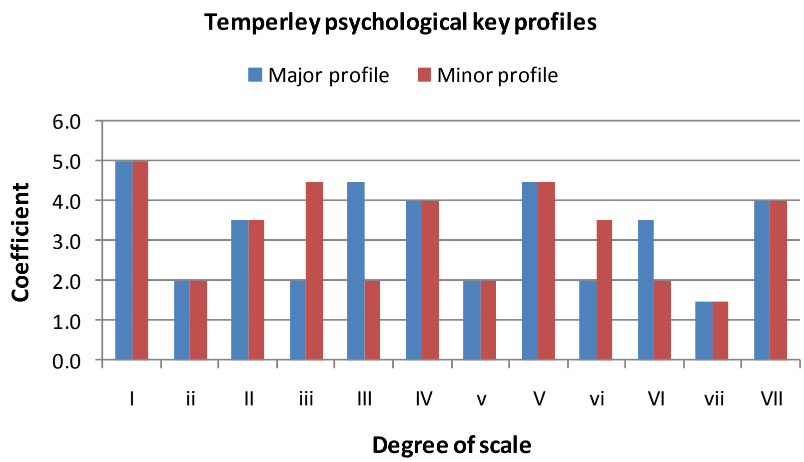


Fig. 2.15 Major and minor mode key profile coefficients proposed by Temperley (1999).

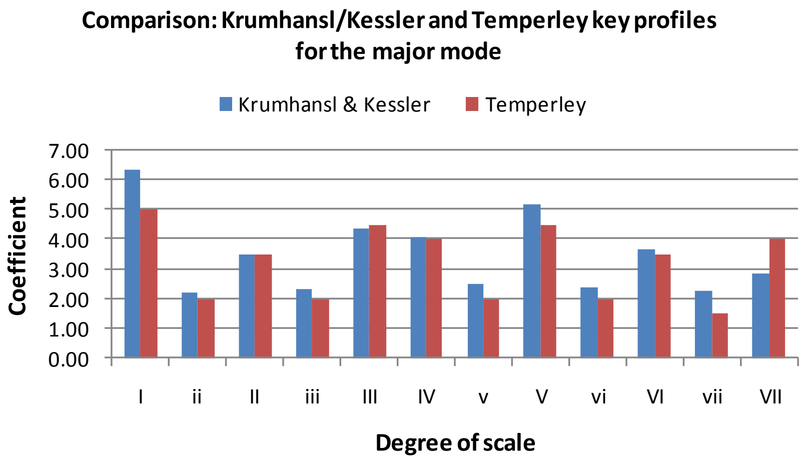


Fig. 2.16 Comparison of the Krumhansl and Kessler and the Temperley key profiles for the major mode.

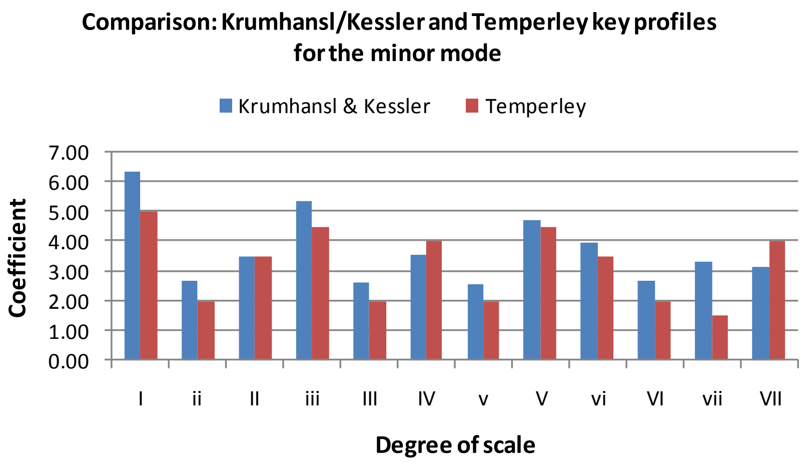


Fig. 2.17 Comparison of the Krumhansl and Kessler and the Temperley key profiles for the minor mode.

The changes introduced by Temperley can be summarised as follows (Temperley, 1999):

- Diatonic steps reflect higher values compared to chromatic steps, with values of at least 3.5. Temperley assumed the harmonic minor scale for the minor mode.
- Chromatic degrees, with the exception of vii, are assigned an equal value of 2.0.
- Degree vii is assigned a low value of 1.5 in order to accommodate the impact of the dominant seventh.

Fig. 2.16 and Fig. 2.17 also show that Temperley reduced the weight of the tonic and the fifth and increased the weight of degree VII for both modes compared to the values originally proposed by Krumhansl and Kessler.

Temperley also proposed the following simplified relationship for determining the correlation coefficient τ_j between the note durations vector $\mathbf{D} = \left[d_i\right]$ and the key profile vector for the j$^{\text{th}}$ key $\mathbf{K}_j = \left[k_{ij}\right]$

$$\tau_j = \sum_{i=1}^{12} d_i k_{ij} \qquad (2.17)$$

As in the case with the Krumhansl and Schmuckler algorithm, the algorithm chooses the key or keys with the highest key correlation score. The revised key profile coefficients and modified key-profile formula yielded an improved performance for the Bach *Courante*, with a success rate of 33.5 out of 40, i.e. 83.75%, compared to the 67.5% obtained with the original Krumhansl/Kessler key profiles and algorithm. In determining the success rate, the algorithm is awarded a score of 0.5 if it chooses two keys, and one of them is judged as correct.

Based on detailed analysis of the results obtained with the modified Krumhansl and Schmuckler key-finding algorithm, particularly for the measures where the algorithms failed to assign the correct key, Temperley concluded that the best approach to a template-based key finding algorithm may be found in using a combination of the weighted Krumhansl and Schmuckler template and the flat Longuet-Higgins and Steedman template. This gave rise to a new algorithm that combines principles from both the "*flat-input/flat-key*" Longuet-Higgins and Steedman shape-matching algorithm and the "*weighted-input/weighted-key*" Krumhansl and Schmuckler key profile algorithm to deliver a "*flat-input/weighted-key*" approach (Temperley, 1999). The new Temperley algorithm can be summarised as follows:

- The pitch class distribution is determined by assigning a weight of one if a pitch class is present and a weight of zero if a pitch class is absent.
- The modified key profiles proposed by Temperley, as shown in Table 2.9 and Fig. 2.15, are used.
- The Krumhansl and Schmuckler key-finding algorithm is implemented with the modified key correlation formula proposed by Temperley, given by (2.17).

Temperley applied the algorithm on a measure by measure basis to the first 40 measures of the Bach *Courante*, achieving key estimates that are fully compatible with the theoretical benchmark assessments for all measures. The overall assessment acknowledges, however, that this algorithm is mostly suited for short sections of score material viewed in isolation, i.e. without taking the context induced by previous and subsequent sections into consideration.

2.7.5 The modified Krumhansl and Schmuckler algorithm proposed by Huron and Parncutt

Huron and Parncutt (1993) proposed improvements to the Krumhansl and Schmuckler algorithm using the following approach:

- *Incorporating the perceptual effects of pitch saliency:*

  The effects of pitch saliency are incorporated to account for listener's psycho acoustical perception of the pitch content of the subject material. The methodology makes use of the model of pitch perception developed by Terhardt (1979) and Terhardt, Stoll and Seewann (1982a; 1982b).

- *Incorporating the perceptual effects of echoic memory:*

  The effects of echoic memory are incorporated to account for the results obtained by Cook (1987), that suggests that recent sonorities rate perceptually higher than more distant sonorities (Huron and Parncutt, 1993). The methodology involves the application of a weighing factor to the note durations, based on the model of sensory decay developed by Parncutt.

Huron and Parncutt (1993) evaluated the performance of the improved model for stimuli representing both the structural and functional scenarios of tonality perception. For the structural scenario, model performance was evaluated with and without pitch salience, while the half-life value associated with the echoic memory was treated as a parameter that was optimised to achieve the best correlation between the model output and the Krumhansl and Kessler key profile coefficients. In comparing the results obtained with and without pitch salience, Huron and Parncutt concluded that incorporation of sensory memory decay improves the model in the sense that the model predictions exhibit better correlation with the listener perceptions recorded by Krumhansl. The performance of the model is further improved by incorporating the psycho acoustical model of pitch salience.

For the functional scenario, model performance was evaluated for the pitch strings devised by Brown (1988), using an echoic memory decay half-life of 1.0 second and complex tones with durations of 0.5 seconds. For a given input stimulus, the model delivers a correlation value for each of the 12 major and 12 minor keys represented by the Krumhansl and Kessler major and minor mode profiles, which were reduced to 12 by summing the major and minor mode values for each pitch class. This is necessary because Brown's listener responses did not distinguish between major and minor modes. Thus, while the incorporation of echoic memory decay and pitch salience improves the key-finding performance of the original Krumhansl and Schmuckler algorithm, in comparing the model predictions with listener responses obtained by Brown, Huron and Parncutt (1993) concluded that the model fails to predict the listener responses for the pitch strings used in Brown's study.

Overall, Huron and Parncutt (1993) concluded that "*tonality perception is determined by both structural and functional factors*" and that "*perception of musical key may also be influenced by future expectations*".

2.7.6 The preference rule key-finding algorithm proposed by Temperley

Based on an extensive evaluation of the Krumhansl and Schmuckler key-finding algorithm, including the parametric and computational modifications thereof, Temperley (1999; 2002) concluded that the key template approach, whilst an important component, cannot on its own account for the tonal properties of a musical score. Based on this assessment, using a music theoretical rather than a cognitive departure point, Temperley devised a new model that combines aspects of the Longuet-Higgins and Steedman procedural approach with the Krumhansl and Schmuckler preference rule approach. The algorithm retains the concept of a pitch class template, but makes provision for two different representations of the pitch classes, namely Natural Pitch Classes (NPCs) and Tonal Pitch Classes (TPCs).

The NPC approach, also used in the Krumhansl and Schmuckler key-finding algorithms, is based on the well-known circle of fifths shown in Fig. 2.18 and does not allow for different spellings of the same pitch. The TPC approach, proposed by Temperley, is based on the line of fifths shown in Fig. 2.19 and assigns pitch classes according to a preference rule system.

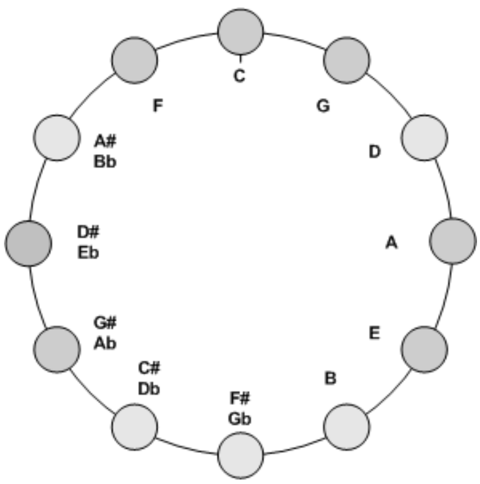


Fig. 2.18 Circle of fifths pitch spelling model (Temperley, 1999;2002)

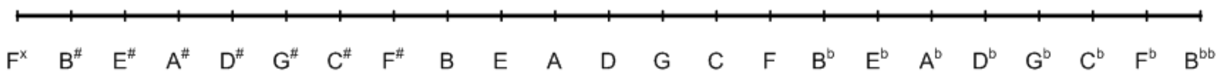


Fig. 2.19 Line of fifths pitch spelling model (Temperley, 1999; 2002).

It allows for multiple spellings of the same pitch, depending on the tonal context. Temperley argues that the TPC offers a distinct advantage for key-finding because different spellings of the same NPC pitch class can be assigned different key profile weights, as shown in Table 2.10 and Fig. 2.20 for the major and minor modes.

Table 2.10 Major and minor mode tonal pitch class key profile coefficients proposed by Temperley (1999; 2002).

Pitch class	**Fb**	**Cb**	**Gb**	**Db**	**Ab**	**Eb**	**Bb**	**F**	**C**	**G**	**D**	**A**
Major key	1.50	1.50	1.50	2.00	2.00	2.00	1.50	4.00	5.00	4.50	3.50	3.50
Minor key	1.50	1.50	1.50	2.00	3.50	4.50	1.50	4.00	5.00	4.50	3.50	2.00
Pitch class	**E**	**B**	**F#**	**C#**	**G#**	**D#**	**A#**					
Major key	4.50	4.00	2.00	1.50	1.50	1.50	1.50					
Minor key	2.00	4.00	2.00	1.50	1.50	1.50	1.50					

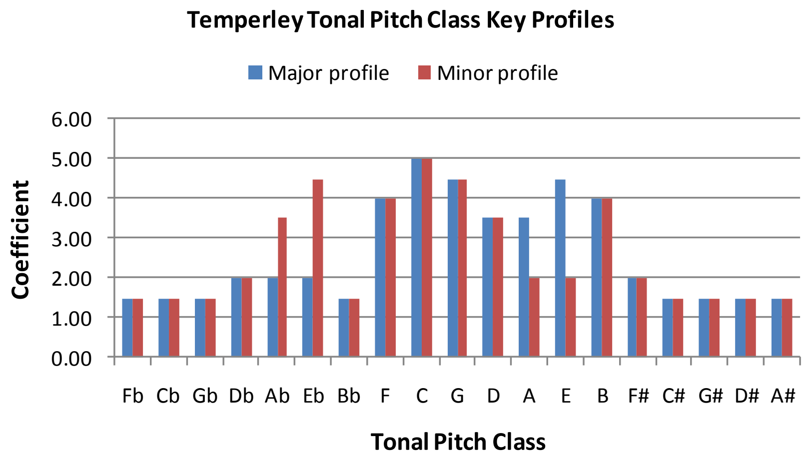


Fig. 2.20 Tonal pitch class key profiles proposed by Temperley (1999; 2002).

In broad terms, the algorithm can be summarised as follows:

- The score material is partitioned along the temporal timeline. Temperley admits that the choice of partition length is a matter of arbitrary choice and a subject for further research, but suggests and applies the score measure as a suitable option.
- Key correlation values are derived for each key for each partition in isolation using the Krumhansl and Schmuckler algorithm. This procedure makes use of the following:
  - Key profiles and a flat pitch class vector using a Tonal Pitch Class (TPC) representation.
  - The simplified correlation formula given by (2.17).
- Key scores are calculated for each partition, based on the key correlation values calculated for the partition and a penalty that is imposed if the key differs from the key derived for the preceding section.
- A final key estimate is derived for each partition from the key scores calculated for the successive key combinations.

The following general comments apply for the new Temperley algorithm:

- Temperley admits that the two parameters required by the program, i.e. the partition or section length and the value of the change penalty, can be problematic in the sense that optimal values depends to some extend on the nature of the sample material.

- Due to the partitioning methodology, the algorithm is suitable for tracking key modulations.
- The actual implementation of the algorithm is described rather vaguely in literature. From inspection of the C language code made available by Temperley as part of the Melisma project, it is clear that the algorithm is also quite complicated.

Temperley (1999) evaluated the performance of the new key-finding algorithm for a range of subject material, including the 48 fugue subjects from Bach's *Well-tempered Clavier*, the *Gavotte* from Bach's *French Suite No. 5 in G major (BWV 816)* and a corpus of data from the workbook (Kostka and Payne, 1995b) and instructor's manual (Kostka, 1995) of the textbook *Tonal Harmony* (Kostka and Payne, 1995a). The results of the performance tests can be summarised as follows:

- *Fugue subjects from Bach's Well-tempered Clavier:*

  The algorithm chose a correct single key for 42 out of the 48 fugues. In two cases, ties were recorded, with the correct key as one of the ties. If half a point is awarded for the ties, this yields 43 correct keys out of 48, i.e. 89.6%. The algorithm tracked the modulation of two of the six modulating fugues correctly, tracked to the second key for the third modulating piece and assigned a single incorrect key for the remaining three fugues.
- *Gavotte from Bach's French Suite No. 5 in G major (BWV 816):*

  Temperley evaluated the ability of the algorithm to track key modulations for the *Gavotte* from Bach's *French Suite No. 5 in G major (BWV 816).* Based on the results shown in Appendix G, Temperley concluded that the algorithm tracks the key modulations correctly if compared against analytical results based on music theoretical principles.
- *Kostka and Payne corpus*:

  The algorithm was used to analyse 46 excerpts from the Kostka and Payne corpus. In terms of Temperley's methodology of scoring some of the ambiguous results, such as dual keys assigned by Kostka and Payne and the handling of pivot chords, the algorithm achieves a correct result in 87.4% of a total of 896 segments or partitions. The algorithm also tracked the 40 key modulations identified in the Kostka and Payne analysis.

The material contained in the Kostka and Payne corpus represents a rigorous test and the excellent performance of the Temperley key-finding algorithm for this corpus establishes it as a prominent methodology in this field. The need to assign appropriate values for the segment length and change penalty parameters, however, detract from the overall quality of the algorithm. In this regard, it is noteworthy that the results listed above were obtained using significantly different change penalties, namely 6.0 for the Bach fugues and 12.0 for the Kostka and Payne corpus.

2.8 Geometrical models of tonal relationships

2.8.1 Introduction

Geometrical models of tonal relationships, designed with the view to represent the tonal relationships between pitch classes, chords and keys, feature prominently in literature. These models generally assume the form a geometrical structure in the Euclidean plane or three-dimensional Euclidean space so that the relative positions of pitch classes, chords or keys relate to "*closeness*" in the tonal sense. In the three-dimensional structures, pitch classes, chords and keys are typically positioned on geometrical surfaces or manifold forms such as helices and toroids.

Geometrical models of tonal relationships represent an approach where tonality perception is defined in terms of static hierarchal relationships between pitch classes, intervals, chords and keys. These models do not take cognizance of the temporal context of these entities as they occur in musical progressions. The latter aspect is investigated in the experimental studies conducted by Cook (1987) and Brown (1988).

Computational models of tonality cognition and key-finding algorithms have close relationships with geometrical models representing tonal relationships. The key-finding algorithms often rely on tonal relationships that can be represented in the form of a geometrical model. The results achieved, especially in tracking key modulations, are also often interpreted and/or represented using geometrical structures. Krumhansl (1990), for instance, devoted considerable effort to express the tonal relationships implied by the psychological pitch class profiles in geometrical form. Similarly, the MIDI toolbox features the use of Self Organising Maps (SOMs) to reflect the temporal movement of tonal centre (Eerola and Toiviainen, 2004a). Some key-finding algorithms, such as the methodology proposed by Chew (2000, 2001), are based entirely on tonal relationships expressed in the geometrical domain. For the purposes of this review, the models will be categorised as follows:

- Models based on classical analytical theories of consonance and harmony.
- Models based on experimental data of tonality perception.

Geometrical models can be further categorised with regard to whether a model aims to encode the relationships between pitch classes, chords or keys. The number of different models proposed in literature is quite high. The focus in this review will be on models that relate to the topology and interpretation of the modern computational models of tonality perception, particularly with reference to the two-dimensional models that can be used to represent tonal centre on a plane. Some of the earlier geometrical models, however, form the basis of or are closely related to these.

2.8.2 Geometrical models based on music theory

2.8.2.1 Introduction

Early examples of the geometrical models of tonal relationships include Heinichen's circle of key relationships and Weber's charts of major and minor key relations (Purwins, Blankertz and Obermayer, 2007). The circle of fifths for the chromatic scale shown in Fig. 2.21 represents a well-known example of the two-dimensional geometrical approach to express the tonal relationships between pitch classes. The model encodes the consonance or dissonance associated with the intervals of the chromatic scale relative to the root, in the sense that consonant intervals such as the fifth are located close to the root while dissonant intervals such as the tri-tone are located distant from the root.

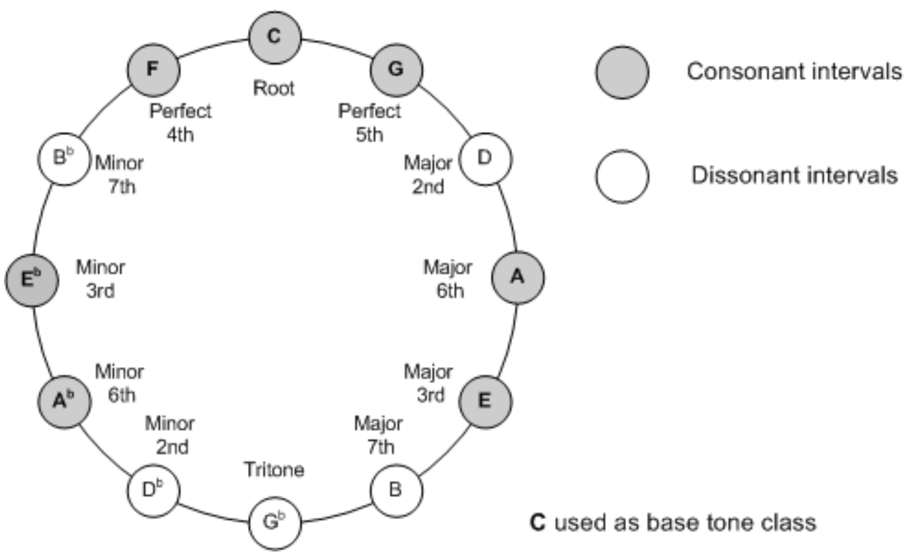


Fig. 2.21 Circle of fifths representation of consonant and dissonant intervals.

2.8.2.2 The Tonnetz

The Tonnetz or tone-network of pitch classes has been extensively employed by researchers such as von Oettingen (1866) to map harmonic relationships and motion in the tonal space, particularly with reference to modulation between keys. It features prominently in the classical works of Helmholtz (1954) and Schoenberg (1969), more recent research such as the neo-Riemannian theory of harmony revived by Lewin (1987) and the theories of tonal space advanced by Gollin (1998), Chew (2000) and Lerdahl (2001). The Tonnetz have been represented in numerous forms, a few of which is of interest to this study. The main properties of the Tonnetz can be summarised as follows:

- Fig. 2.22 shows the basic two-dimensional substructure of pitch class relationships and the associated intervallic relationships between any single root pitch class and all adjacent pitch classes contained in the Tonnetz structure. Pitch classes in the horizontal lines from left to right reflect a positive interval of a fifth. Pitch classes in the diagonal lines from the lower left to upper right reflect a positive interval of a major third while pitch classes in the diagonal lines from the lower right to the upper left reflect a negative interval of a minor third.
- Fig. 2.23 shows the major and minor triad relationships for the major and minor keys contained in the Tonnetz for the diatonic scale in C. The shaded upright triangles relate the IV, I and V major triads while the inverted shaded triangles relate the ii, vi and iii minor triads.
- The two-dimensional substructure shown in Fig. 2.22 can be expanded horizontally and vertically by appropriately appending identical substructures at the horizontal and vertical edges to yield the full Tonnetz shown in Fig. 2.24. The left and right edges and lower and upper edges respectively of the full Tonnetz are circularly adjacent, which allows it to be represented three-dimensionally as a torus for equal temperament tuning systems.

The original Tonnetz lattice structures in the Euclidean plane have given rise to numerous developments, including the two-dimensional and three-dimensional neo-Riemannian representations (Cohn, 1997) and the three-dimensional structure proposed by Gollin (1998) for representing tetrachord classes. The relationships contained in the various Tonnetz representations have been formulated and analysed extensively in formal mathematical terms.

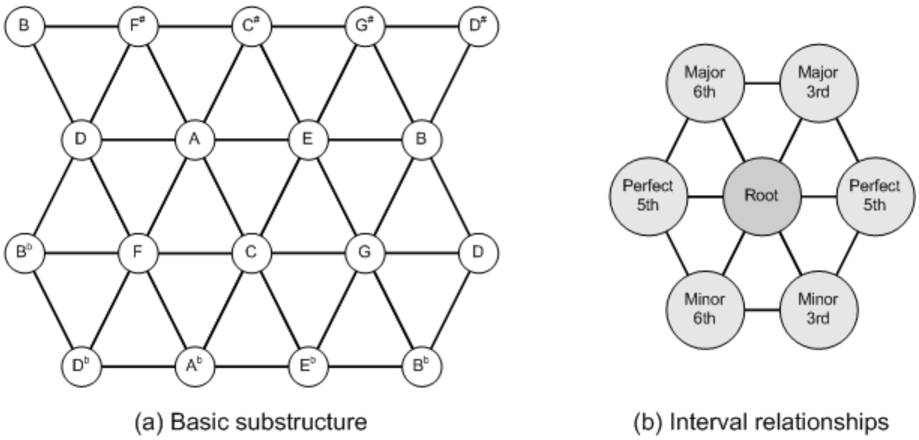


Fig. 2.22 Tonal substructure and intervallic relationships contained in the Tonnetz.

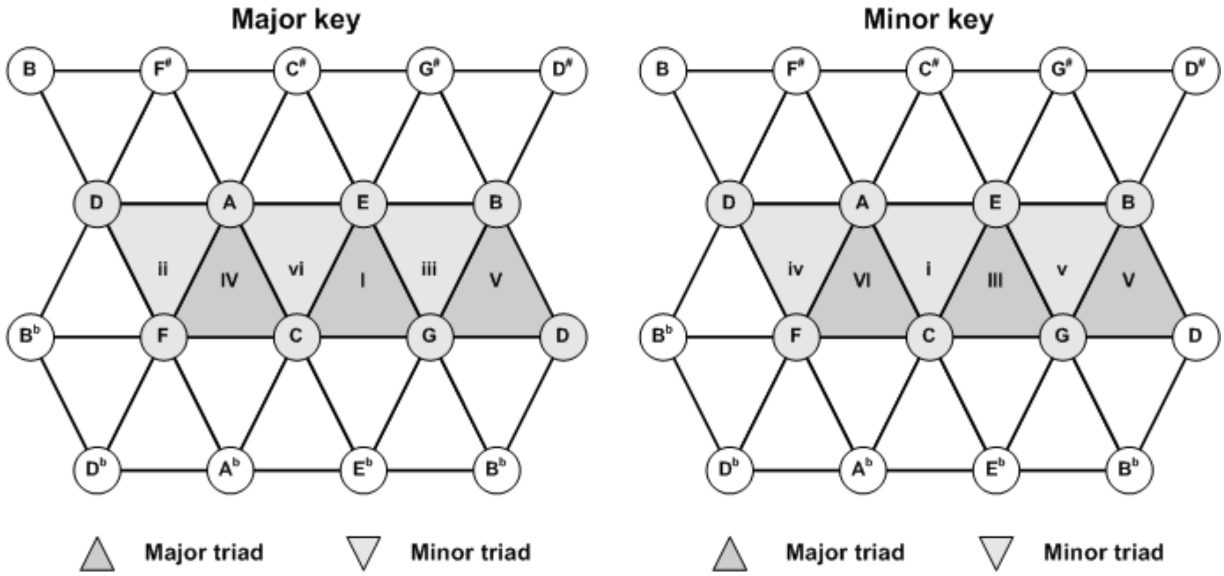


Fig. 2.23 Example of major and minor key triad relationships expressed in the Tonnetz.

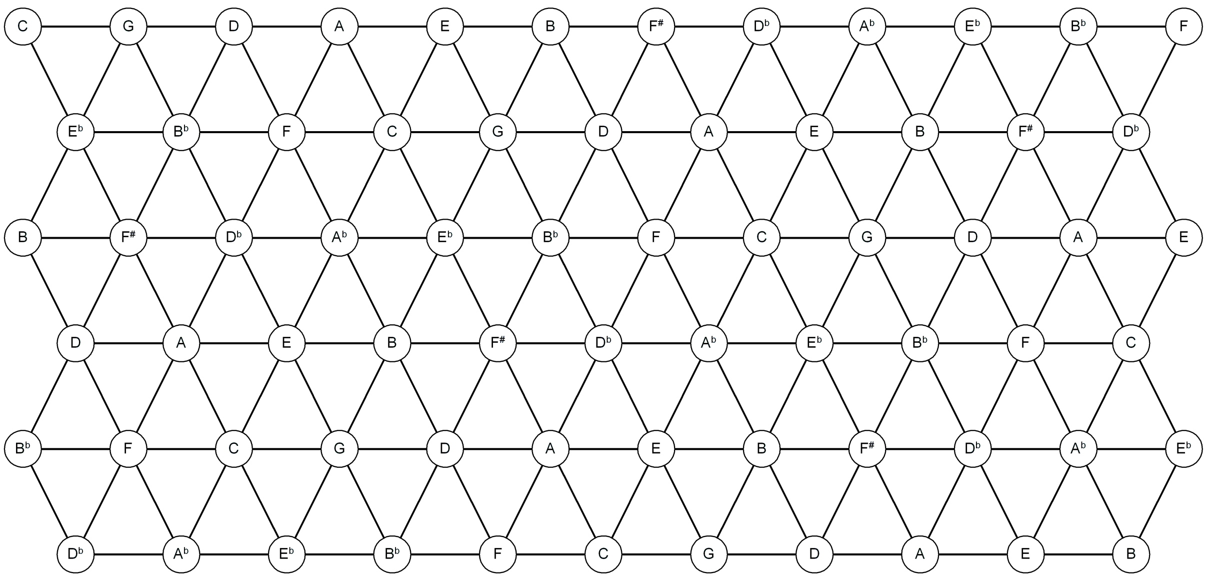


Fig. 2.24 Full Tonnetz.

Although the conventional Tonnetz geometry encodes tonal relationships in space, it does not constitute a quantitative model of tonality in the sense that the distances between pitch classes represent numerical measures of tonal proximity. Such representations evolved from the more recent studies of tonality cognition.

2.8.2.3 The charts of regions proposed by Schoenberg

The chart of key regions proposed by the Austrian and later American composer Schoenberg (1969) represents one of the historical geometrical topologies of tonal relationships that also feature in the context of recent computational models of tonality perception. The structures represent the tonal relationships between major and minor keys in the Euclidean plane, but can be represented using the toroid form. The charts of major and minor key regions, of which a sub region is shown in Fig. 2.25, reflect the circle of fifths in the vertical dimension while the horizontal dimension is arranged according to the relationships of relative and parallel minors or majors relative to a central tonic. It is based on Schoenberg's principle of monotonality, which states that "*every digression from the tonic is considered to be still within the tonality, whether directly or indirectly, closely or remotely* *related. In other words, there is only one tonality in a piece, and every segment formerly considered as another tonality is only a region, a harmonic contrast within that tonality...subordinate to the central power of [its] tonic.*" (Schoenberg, 1969). The tonal distance between the tonic and a region are defined in terms of the number of common notes, such that more notes in common translates to a closer relationship while less notes in common translates to a more distant relationship.

The inter-key relations found in the Schoenberg charts also appear in the geometrical structures derived by Krumhansl and Kessler (1982) from psycho acoustical experiments. It is interesting that Schoenberg is also credited with the atonal twelve-tone technique of composition that lacks a tonal centre in the sense that the composer strives to equalise the weight of all tones in the composition.

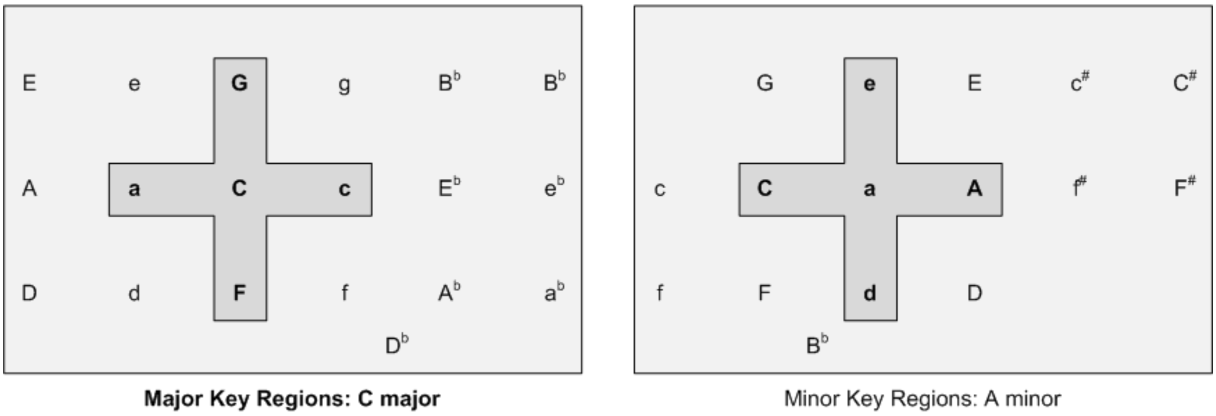


Fig. 2.25 Schoenberg's charts of major and minor key regions illustrated for C major and A minor respectively (Schoenberg, 1969; Krumhansl, 1990)

2.8.2.4 The melodic map

Lakner (1960) proposed the two-dimensional structure of tonal relationships shown in Fig. 2.26, subsequently named the *melodic map* by Shepard (1982a) due to the prominent place of scale relationships in the structure.

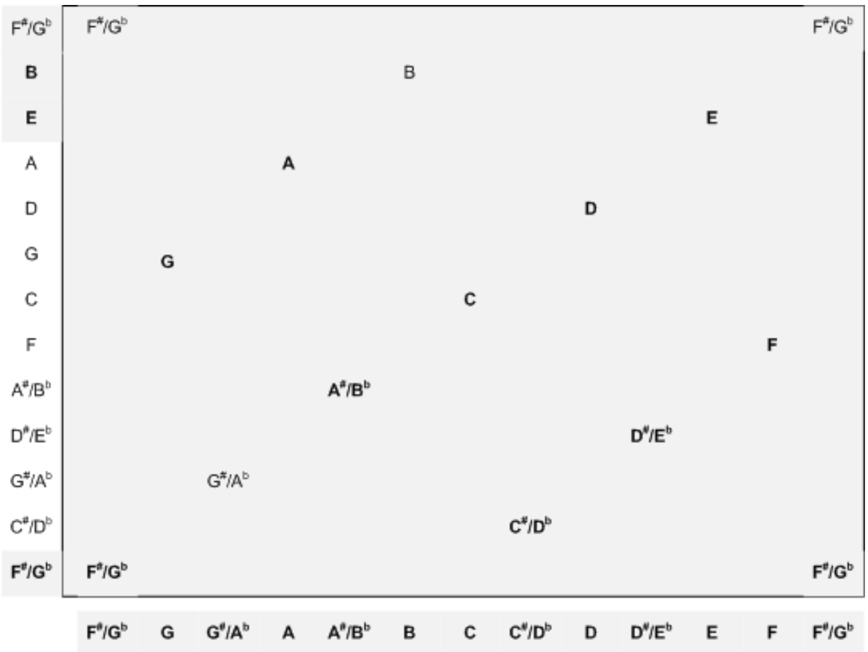


Fig. 2.26 The melodic map proposed by Lakner (1960).

The horizontal dimension reflects the position of the tones in the so-called chroma circle (Krumhansl, 1990), while the vertical dimension is associated with the position in the circle of fifths. The diagonals alternately express major second and minor third intervals. The structure gives rise to a toroidal surface in four dimensions.

2.8.2.5 The harmonic map

Shepard (1982a) proposed a two-dimensional structure for representing the pitch classes in such a manner that the tones related to any other tone by major thirds, minor thirds and perfect fifths are located in close spatial proximity, while major and minor seconds are located relatively distant. This structure, derived from the double helix model and named the *harmonic map* by Shepard, is shown in Fig. 2.27.

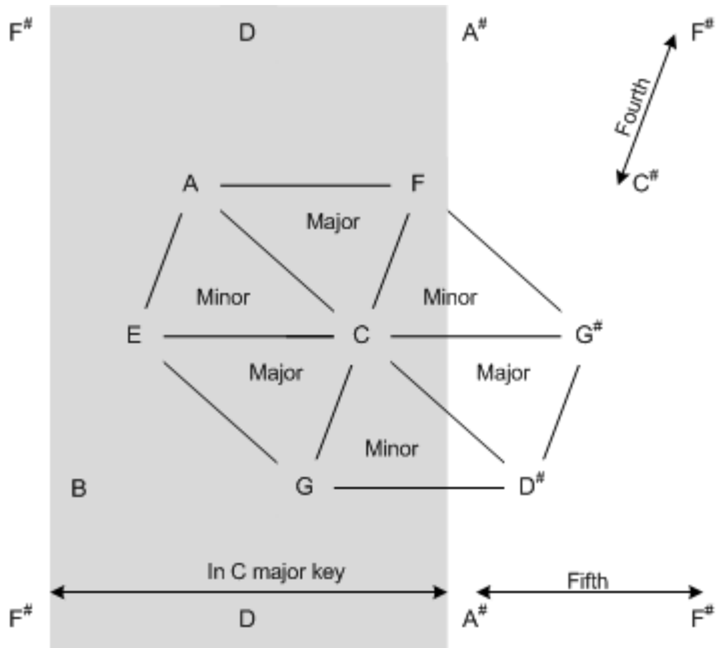


Fig. 2.27 The harmonic map proposed by Shepard (1982a).

The harmonic map also has the property that the tones that form major and minor triad chords with a particular tone are clustered around that tone, giving rise to major and minor chord regions as shown in Fig. 2.27. It is significant that a major chord region is bordered by the associated relative and parallel minor chord regions. Based on the tonal relationships encoded in the structure, Shepard concluded that the harmonic map is well suited for representating harmonic relations.

2.8.3 Geometrical models based on tonality cognition experiments

2.8.3.1 Introduction

Krumhansl and Shepard (Krumhansl and Shepard, 1979; Krumhansl, 1979; 1986; Shepard, 1982a; 1982b) are credited with the development of the cognitive-structural approach to music cognition. Shepard focused on developing geometrical models to represent the hierarchical relations between tones (Shepard, 1982a; 1982b). These include a simple helix form, a double helix, a four-dimensional toroid and a five-dimensional toroidal helix. As is common in computational geometrical models of tonal relationships, closer relationships between pitch classes and/or chord formations are mapped to closer proximities in the geometrical structure. Krumhansl (1979; 1986) and Krumhansl and Shepard (1979) continued this approach with an alternative geometrical model, based on the shape of a cone. These are significant in the sense that the models encode the relationships obtained in cognitive experiments with triad chords in various tonal contexts.

2.8.3.2 Multidimensional scaling map proposed by Krumhansl and Kessler

Krumhansl and Kessler (1982) used the key profiles summarised in Table 2.6 and Fig. 2.10 to derive a measure of interkey distance between each pair of the possible 12 major and 12 minor keys. This interkey distance is calculated as the correlation between the key profiles associated with the pair of keys. Using a mathematical technique known as *nonmetric multidimensional scaling* to represent the keys in multidimensional space, Krumhansl and Kessler found a good fit using four dimensions. This gave rise to a structure where the keys are located on the surface of a toroid such that the Euclidian distance between the key coordinates reflects the measure of similarity or closeness.

Fig. 2.28 represents the toroidal surface as a two-dimensional plane. In the horizontal dimension, interkey distances reflect major thirds. The diagonal lines from the bottom left to the upper right alternate between lines of major and minor keys arranged according to the circle of fifths. The spatial representation shown in Fig. 2.28 forms the basis for the Self Organising Map (SOM) approach used in the visualization of key relationships and tonal centre (Kohonen, 1995; Toiviainen and Krumhansl, 2003; Toiviainen, 2008). The key map shown in Fig. 2.28 exhibits a number of similarities with Schoenberg's charts of major and minor key regions shown in Fig. 2.25.

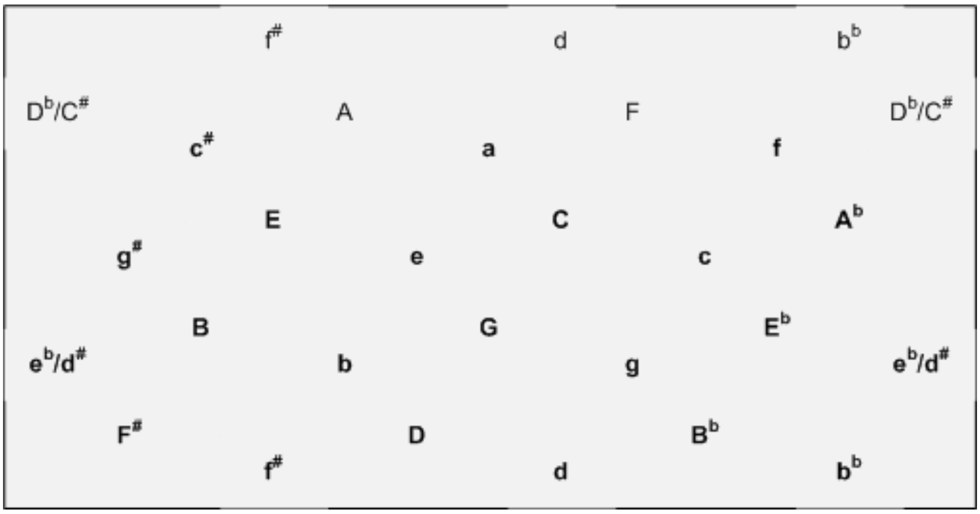


Fig. 2.28 Multidimensional scaling map proposed by Krumhansl and Kessler (1982).

2.9 The MIDI toolbox

The MIDI toolbox is a suite of software routines for the analysis and characterization of symbolic music data using digital computers (Eerola and Toiviainen, 2004b). The toolbox is implemented in Matlab and was developed at the Department of Music of the University of Jyväskylä, Finland. The source data is imported from standard MIDI files and the suite contains some support functions for the visualising and playback of the imported musical content. Eerola and Toiviainen list the applications of the toolbox as data-mining, modelling music, processing musical data for perceptual experiments and as a teaching aid in music cognition (Eerola and Toiviainen, 2004b). The functionality offered for music analysis can be categorised as follows:

- *Statistical analysis:*

  Statistical analysis include the distributions of note durations, dyad note durations, interval sizes, absolute interval sizes, interval directions, dyad intervals, pitch-classes and dyad pitch-classes (Eerola and Toiviainen, 2004b). The entropy of a particular distribution can also be determined.

- *Tonality analysis:*

Tonality analysis includes key estimations and major or minor mode estimations, using the Krumhansl and Schmuckler key-finding algorithm with the Krumhansl and Kessler (1982) key profiles.

- *Meter-related analysis:*

  Meter-related analysis include determining metrical hierarchy (Lerdahl and Jackendoff, 1983), the duration accent of the events (Parncutt, 1994), the measure of durational variability of events (Grabe and Low, 2002), meter estimation (Brown, 1993) and tempo estimation.

- *Melodic analysis:*

  Melodic analysis include melodic range, melodic complexity (Eerola and North, 2000), melodic originality (Simonton, 1984), degree of melodiousness (Euler, 1739), melodic accent salience (Thomassen, 1982), melodic attraction (Lerdahl, 1996), measure of phenomenal accent synchrony (Eerola and Toiviainen, 2004b), melodic motion as a mobility (Von Hippel, 2000), implication-realization principles by Narmour (1990), melodic tessitura based on deviation from median pitch height (Von Hippel, 2000) and melodic distance.

It is clear from the above summary that the MIDI toolbox implements a significant representation of the analytical techniques developed in recent years for the analysis of symbolic music data, including techniques that involve the modelling of music perception. The capabilities of the MIDI toolbox have been demonstrated for applications such as melodic segmentation, meter-finding, key-finding and determining melodic contour and similarity (Eerola and Toiviainen, 2004b).

2.10 The standard MIDI file format

2.10.1 Introduction

The suitability of the standard MIDI file storage format for music notational and analysis purposes is determined by the data storage specifications associated with this standard. The following aspects are of importance for the purposes of this project:

- *The file structure:* The software application is required to import musical data from standard MIDI files and translate this data to a format suitable for analysis.
- *The relevant meta event data:* Meta event data that encode music notational information such as key signature, time signature and tempo information is essential for fully qualifying the musical information. The nature of the data structures and temporal deployment of this information in the MIDI source file can impose limitations on the functionality and accuracy that can be achieved in an analysis.
- *The note event data*: The note events represent the actual musical events. Limitations in representing the note events in the MIDI source file, such inadequate resolution and range of quantities such as time and pitch translate directly into limitations in the analysis.

2.10.2 File structures

The Musical Instrument Digital Interface (MIDI) specification (The MIDI Manufacturers Association, 1996) represents a very detailed standard in the sense that it embodies all aspects such a specification, including musical aspects such as instrument definitions, data transmission protocol, interface hardware specifications, file storage specifications, etc. In view of the emphasis on score representation and analysis rather than music sequencing and synthesis, this project is mostly concerned with the file storage considerations.

The standard MIDI file format as specified in the Standard MIDI File Format Specification Ver. 1.1 (The International MIDI Association, 1996) is an industry standard for the storage and interchange of symbolic music information (Good, 2001a). The MIDI file structure consists of a collection of units denoted as chunks, as shown in the hierarchy presented in Fig. 2.29. Two types of chunks are allowed for, namely header chunks and track chunks. A header chunk encodes information applicable to the entire MIDI file while track chunks encode sequential MIDI data for up to 16 MIDI channels. The MIDI file specification make provision for three file formats, namely type 0, type 1 and type 2. A type 0 file consists of only two chunks, i.e. a header chunk and one track chunk. Type 1 and type 2 files consist of one header chunk and up to 16 track chunks. The track chunks of a type 1 format are interpreted as independent tracks, while the track chunks of a type 2 format are interpreted as independent patterns rather than independent tracks.

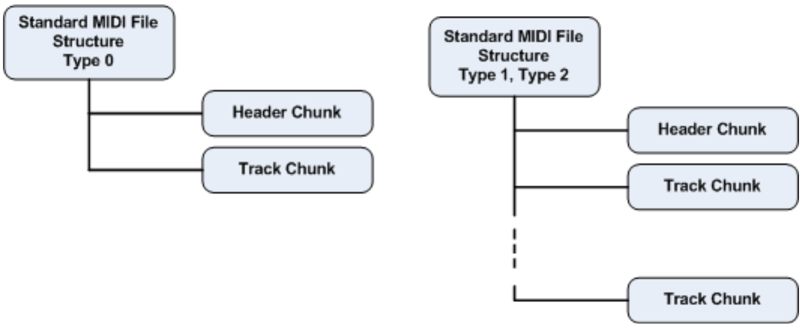


Fig. 2.29 Standard MIDI file structures.

2.10.3 Chunk structures

2.10.3.1 Header chunk structure

Fig. 2.30 shows the structure of the header chunk specified in the standard MIDI file specification (The International MIDI Association, 1996). The chunk is headed by a 4-character chunk type identifier and a 32-bit length specification, which encodes the number of bytes in the chunk. The type identifier for the header chunk consists of the ASCII characters "*MThd*". The chunk length specification excludes the eight bytes required for the chunk type and length information and is formatted with Most Significant Byte (MSB) in the first position.

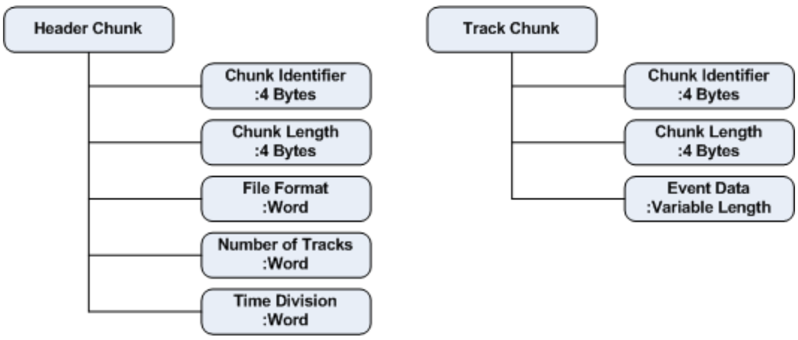


Fig. 2.30 Standard MIDI file chunk structures.

The header chunk encodes three parameters, all stored in word format, namely the *file format*, *number of tracks* and *time division* specification for the file. The *file format* parameter specifies the type of MIDI file, i.e. type 0, type 1 or type 2. The *number of tracks* parameter specifies the total number of tracks contained in the file.

Table 2.11 summarises the time division format specification for standard MIDI files. The *time division* parameter specifies the nature of the delta-time attribute associated with the musical events contained in the track chucks. It is a crucial parameter that is used to convert the track event delta times into real time. This parameter is encoded at bit level and allows for two types of time formats, namely metrical time format and time-code based formats.

Table 2.11 Time division format specification for standard MIDI files (The International MIDI Association, 1996).

Time division format	Description	Bit 15	Bit 14 – bit 0	
Metrical time	Delta-time is specified as the number of ticks per quarter-note	0	Bit 14 – bit 0	
			Number of delta time ticks in a quarter-note	
Negative SMPTE	Delta-time is specified in negative SMPTE format	1	Bits 14 - 8	Bits 7 - 0
			Number of frames per second (Stored as a negative number in two's compliment form).	Frame resolution

The type of time format is defined by bit 15 in the *time division* parameter, where a bit value of 0 denotes a metrical time format and a bit value of 1 denotes a Society of Motion Picture and Television Engineers (SMPTE) time format.

In the case of metrical time format, bit 14 to bit 0 defines the number of delta time ticks in a quarter-note. This allows a maximum parameter value of

$$2^{15} - 1 = 32767, \tag{2.18}$$

which yields an excellent upper limit for the time resolution that can be achieved.

In the case of the SMPTE time-code based format, the *time division* parameter word is further subdivided in a frame rate and frame resolution specification, encoded in bit 14 to bit 8 and bit 7 to bit 0 respectively. The frame resolutions in general use are 4 (MIDI Time Code resolution), 8, 10, 80 (bit resolution), or 100. Table 2.12 summarises the frame rates, application areas and bit encoding for the standard SMPTE time-codes (Rees, 2001).

Table 2.12 Standard SMPTE frame rates (Rees, 2001).

Frame rate	Comments	Bits 14 - 8
24 frames/s	Required for film work, rarely used for audio.	-24
25 frames/s	European Broadcasting Union (EBU) standard. Used for video through Europe and Australia. Related to a mains frequency of 50 Hz. Used for PAL or SECAM colour TV system.	-25
29 frames/s	Drop frame standard American colour television standard. Used for video work in America and Japan. Related to a mains frequency of 60 Hz. Used for NTSC television standard. The number of frames per second is 29.97, i.e. not an integer. An approximation is required. Two frames counts are dropped at the start of every minute, except for every tenth minute.	-29
30 frames/s	Used for audio in US. Also used with the Sony 1630 format for CD mastering	-30

2.10.3.2 Track chunk structure

Fig. 2.30 shows the structure of the track chunks specified in the standard MIDI file specification (The International MIDI Association, 1990). The chunk is headed by a 4-character chunk type identifier and a 32-bit length specification, which encodes the number of bytes in the chunk. The type identifier for track chunks consists of the ASCII characters "*MTrk*". The chunk length specification excludes the eight bytes required for the chunk type and length information and is formatted with Most Significant Byte (MSB) in the first position.

The header is followed by the event data that represents the musical events in the track. All events are represented by a delta-time parameter, an event type parameter and some additional event-specific data. The delta-time parameter is stored as a Variable Length Value (VLV) that specifies the temporal position of the event relative to previous event in the track. A VLV consists of a variable number of bytes, where each byte is structured as a continuation bit in the Most Significant Bit (MSB) position followed by a 7 bit data value. A continuation bit of one indicates that the next byte is also part of the VLV, while a continuation bit of zero indicates that the current byte is the final byte as shown for the 4-byte VLV in Fig. 2.31. The standard MIDI file specification allows a maximum VLV length of 4 bytes. This yields a maximum data value consisting of 28 bits.

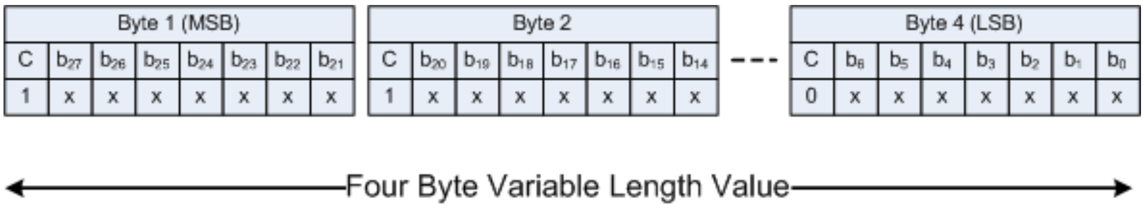


Fig. 2.31 Maximum permissible delta-time represented as a four byte variable length value.

A delta-time of zero indicates that the event has the same temporal position as the previous event. In the case of the first event in the track, the associated delta-time defines the time interval before the occurrence of the first event. All events are encoded with a delta-time value, including events that are by nature not affected by the timeline position in the track. The latter should be associated with a delta-time value of zero and be located in the beginning of the event data in the track.

Delta-times are relative rather than absolute time values. The time unit associated with a delta-time value is determined by the *time division* parameter defined in the header chunk and the *tempo* events defined as part of the track.

Two categories of events are associated with the track chunks, namely meta events and note events. The MIDI standard makes provision for 15 type of meta events. Three of these meta events are of crucial importance for music notation applications, namely the *key signature*, *time signature* and *tempo* events, and all MIDI files should include this information.

2.10.4 Meta events

2.10.4.1 Key signature meta event

Table 2.13 summarises the definition of the *key signature* meta event. The *key signature* data allows the number of sharps or flats to be defined, together with a specification of whether the *key signature* refers to a major or minor key.

Table 2.13 Key signature meta event data structure.

Parameter	Parameter	Data type	Description
Metadata header	Identifier	Byte	Meta event identifier: FFh
	Type	Byte	Time signature type identifier: 59h
	Length	Byte	Meta event data length in bytes: 02h
Key signature data	Sharps/flats	Byte	Specifies the number of sharps or flats: Positive number: Number of sharps Negative number: Number of flats
	Scale	Byte	Denotes the scale mode: 00h: Major key, 01h: Minor key

2.10.5 Time signature and tempo meta events

In the absence of a valid value, however, the MIDI specification states that the *time signature* and *tempo* defaults to 4/4 and to 120 beats per minute (BPM) respectively.

The location of this metadata in the track chunk structure is depends on the *format* of the MIDI file. A type 0 file with a single multi-channel track should have these meta-events at least at the beginning of the track. A type 1 file with multiple tracks should have these meta-events in the first track. A type 2 file with multiple temporally independent patterns should have initial time signature and tempo information for each of the patterns.

Table 2.14 summarises the definition of the *time signature* meta event. The time signature data defines the time signature in accordance with conventional notation, i.e. a numerator and denominator, where the numerator can be expressed as an integer power of 2. This time signature does therefore not allow for irrational measure lengths such as 4/3. The metronome data parameters allow MIDI clock signals to be interpreted differently from the default of 24 MIDI clocks per quarter note with four 32nd notes per quarter note.

Table 2.14 Time signature meta event data structure.

Parameter	**Parameter**	**Data type**	**Description**
Metadata header	Identifier	Byte	Meta event identifier: FFh
	Type	Byte	Time signature type identifier: 58h
	Length	Byte	Meta event data length in bytes: 04h
Time signature data	Numerator	Byte	Numerator of the time signature
	Denominator	Byte	Denominator of the time signature. Expressed as a negative power of 2. Quarter note is expressed as 2, eighth note is expressed as 3, etc.
Metronome data	Metronome pulse	Byte	Number of MIDI clocks per metronome click. The default is 24 clocks per quarter note.
	32nd notes/quarter note	Byte	Number of notated 32nd notes in 24 MIDI clocks. The default value is 8. Can be used to change the MIDI timing defaults.

The MIDI standard makes provision for tempo changes in the sense that tempo maps can be stored as *tempo* metadata events throughout the track chunks, provided that certain provisions and recommendations are adhered to.

2.10.6 Note events

The standard MIDI file stores the temporal musical information, e.g. individual notes, as events rather that objects. This is due to the original drivers for developing MIDI technology, i.e. real-time music synthesising using stored musical data. Each individual note is defined by two events, namely a *note on* and a *note off* event, stored in the MIDI file using the data structures summarised in Table 2.16.

Table 2.15 MIDI note pitch range in relation to the ISO octave map.

Octave	**Note numbers**											
	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
-1	0	1	2	3	4	5	6	7	8	9	10	11
0	12	13	14	15	16	17	18	19	20	21	22	23
1	24	25	26	27	28	29	30	31	32	33	34	35
2	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59
4	60	61	62	63	64	65	66	67	68	69	70	71
5	72	73	74	75	76	77	78	79	80	81	82	83
6	84	85	86	87	88	89	90	91	92	93	94	95
7	96	97	98	99	100	101	102	103	104	105	106	107
8	108	109	110	111	112	113	114	115	116	117	118	119
9	120	121	122	123	124	125	126	127				

The pitch parameter allows for a range of 128 individual pitch values, where note number 60 denotes middle C, also designated as C4 in the International Organization for Standardization (ISO) octave numbering system. Although the MIDI standard does not specify octave numbers, the MIDI pitch numbers thus translate to the octave designations specified in the ISO standard as shown in Table 2.15. The MIDI range thus exceeds the range specified in the ISO standard. It can actually represent pitch values associated with some instruments that sound outside the ISO standard range, e.g. the Boardwalk Hall auditorium organ, which extends down to C-1, i.e. one octave below C0 (Wikipedia, the free encyclopaedia, 2011a).

Table 2.16 Note on and note off event data structures.

Event	Parameter	Data type	Description
Note on event	Event type	Nibble	Note on: 08h
	MIDI channel	Nibble	MID channel number: 00h – 0Fh
	Pitch	Byte	Note pitch: 00h – 7Fh
	Velocity	Byte	Note velocity: 00h – 07h
Note off event	Event type	Nibble	Note on: 09h
	MIDI channel	Nibble	MID channel number: 00h – 0Fh
	Pitch	Byte	Note pitch: 00h – 7Fh
	Velocity	Byte	Note velocity: 00h – 07h

The absolute position and duration of an individual note have to be deduced from the delta-times of preceding events and the delta-times between the associated *note on* and *note off* events respectively. It follows that resolution and accuracy of note position and duration are determined by the overall properties of the delta-times used in the MIDI file. This has important implications for music notation and analysis based on MIDI data in the sense that inappropriate *time division* and *tempo* specifications can render some analysis results invalid, irrespective of the quality of the algorithms employed for the analysis.

2.10.7 Suitability of the standard MIDI file format for music notation and analysis

A number of conclusions with regard to the suitability of MIDI files for music analysis can be derived from the specifications of the standard MIDI file outlined above. Two categories of considerations are of interest, namely the resolutions that can be achieved for musical events and representation of music notational information in the MIDI file. These can be summarised as follows:

- The *key signature* is encoded as a meta event that allows for all the major and minor key possibilities. The individual pitch classes designated as sharps or flats are not specified and the MIDI standard does not distinguish between enharmonic keys (Good, 2001b).
- The *time signature* is encoded as a meta event with a numerator and a denominator specified as a negative power of two. Each can be specified with an accuracy of 7 bits, which exceeds the requirements of music notation. Due to the numerator definition as a power of two, the MIDI standard does not allow for irrational measure lengths such as 4/3.
- The temporal position of a note event is not encoded as an absolute position, but in terms of a delta-time that specifies the relative position with reference to the preceding event. In music notational context, the resolution of a note position with reference to the bar demarcations are determined jointly by cumulative value of the preceding delta-times and the metrical time division resolution specified in the file header. Delta-time is represented as a VLVs with a maximum resolution of 28 bits while the time division is specified with a resolution of 15 bits. The resolution and range that can be achieved with this combination exceed the requirements of music notation and analysis.
- The range of pitch values that can be represented in the MIDI file extends from C-1 to G9, which exceeds the range expressed in the ISO standard. MIDI does not distinguish between enharmonic notes such as $C^{\#}$ and D^{b}.
- Note durations are not specified as discrete note duration types, but are determined by the delta-times between *note on* and *note off* events. The resolution that can be achieved for note durations is the same as the resolution of the temporal positions of note events, which exceeds the requirements for music notation and analysis.
- The MIDI format does not incorporate the concept of staves. This is regarded as a shortcoming.

From a music notation and analysis perspective, the event-orientated approach of the MIDI file format is less optimal compared to the object-orientated approach employed by most score notation software applications and interchange formats such as MusicXML (Good, 2001a). MIDI files, however, remain a highly viable source for music analysis, mainly due to the following:

- The standard MIDI file standard is a well-defined and open format that has matured to an industry standard supported by all prominent music production and music notation applications as an interchange format.
- An extensive and growing collection of material is readily available in the public domain (Good, 2001b). The fact that MIDI is the dominant control interface standard used for musical instruments and instrumentation for purposes of recording and synthesising ensures that the collection of material available through MIDI files will continue to grow.
- Historically, much of the research in music analysis has made extensive use of symbolic information encoded using the MIDI file format (Eerola and Toiviainen, 2004a).
- The MIDI standard is widely supported by musical instruments, recording equipment and recording software (Kuipers and Good, 2004). It follows that the MIDI file format, which has been designed primarily as a storage format for the MIDI messaging protocol, is particularly well suited for music analysis applications that focus on material in symbolic format representing actual performances.

The rapid growth in the implementation of MusicXML by the music publication industry (Kuipers and Good, 2004), however, dictates that the MIDI format should be complemented by MusicXML as an interchange format for symbolic music analysis applications.

2.11 Direct search methods for parameter estimation

2.11.1 Overview

One of the research objectives for this project involves an investigation of the application of parameter estimation techniques to determine whether it is possible to derive a more optimal key profile for use with the Krumhansl and Schmuckler key-finding algorithm. Key-finding involves determining a key or set of candidate keys for a particular sample of score material. This process, by nature, gives rise to a discontinuous output in the sense that the correctness of the key is defined in terms of discrete criteria or formulations such as applied by Krumhansl and Schmuckler (Krumhansl, 1990) and Temperley (1999). It follows that the cost functions derived from these results will also be of a discontinuous nature. This reality effectively eliminates all parameter estimation algorithms using gradient methods or methods relying on higher order derivatives of the cost function as candidates for the key profile optimization exercise. This leaves direct search algorithms as the only alternative.

2.11.2 Direct search methods

Direct search methods represent a class of algorithms for optimising cost functions that are not continuous or differentiable. The algorithms minimise a cost function or objective function $f(x)$ of a parameter vector x, i.e.

$$\min_{x} \left(f_x\right). \qquad (2.19)$$

The Matlab optimization toolbox offers three such direct search algorithms, namely the Generalised Pattern Search (GPS), the Generating Set Search (GSS) and the Mesh Adaptive Search (MADS) algorithms. Direct search algorithms typically evaluate the cost function at a number of points in the vicinity of the point defined by the current parameter vector and determine whether points with lower cost function values exist. If so, the point with the lowest cost function value becomes the new current point.

The methodology whereby new locations, known as mesh positions, for evaluation are defined depends on the particular search algorithm. It is determined by two considerations, namely the search pattern and the mesh scaling properties. The pattern is defined by the number of independent parameters to be estimated and the positive basis set used. Maximal basis, with 2N vectors and minimal basis, with N+1 vectors, are two commonly basis sets also implemented in Matlab (Mathworks, 2009).

Both the GPS algorithm and the GSS algorithm define the mesh points by adding a pattern of fixed direction vectors defined in the multi-dimensional parameter space to the departure point. When the parameter set is subjected to linear constraints and the current point is located near a constraint boundary, however, the GSS algorithm uses the methodology proposed by Kolda et al (2006) to achieve a more efficient implementation. The MADS algorithm assign mesh points by adding a random set of vectors to the departure point.

The pattern search algorithms as implemented in Matlab employ a polling methodology. A poll involves determining the cost function at the mesh points defined by the algorithm. When the *Complete Poll* parameter associated with the algorithm is off, the algorithm computes the cost function only until a point with a cost function value less than the current point is found. This is considered a successful poll, in which case polling is stopped and the new point becomes the current point. When the *Complete Poll* parameter is set, the algorithm computes the cost function at all mesh points. If the mesh point with the smallest cost function value is smaller than the cost function value associated with the current point, the poll is successful and that mesh point becomes the new current point. After a successful poll the mesh size is expanded, i.e. multiplied by an *Expansion Factor* greater than unity. Following an unsuccessful poll, the mesh is contracted, i.e. multiplied by a *Contraction Factor* less than unity. New mesh points are determined and the polling process is repeated.

The Matlab implementation of the pattern search algorithm also allows for the following:

- *Imposing lower and upper boundaries on the estimated parameters:* This is useful feature for this research investigation as it allows the estimated key profile coefficients to be limited to a normalised range between 0 and 1.
- *Imposing linear and non-linear constraints on the estimated parameters:* This is important from the perspective that the estimated key profile coefficients can be forced to adhere to a numerical hierarchy that is consistent with music theoretical principles.

Overall, the direct search functionality offered by the Matlab optimization toolbox is well suited for the objective of determining an improved key profile template for the Krumhansl and Schmuckler algorithm.

3 Mathematical Formulation of Computational Music Analysis Algorithms

3.1 Overview

Most of the algorithms proposed in literature for computational music analysis are defined in non-mathematical terms. This potentially gives rise to a situation where various definitions and interpretations of such algorithms can be in use. Furthermore, the algorithms implemented in the MIDI toolbox are not explicitly defined in the associated documentation, requiring interpretation of the Matlab code for clarification. In view of the fact that it is one of the objectives of this project to eliminate the need for coding and interpretation of program code in using the software tools, the most important algorithms implemented for the purposes of this project will be defined formally in mathematical terms.

3.2 Statistical properties

3.2.1 Introduction

Fig. 3.1 shows a graphical representation of the hierarchy of statistical analysis options implemented in the software application.

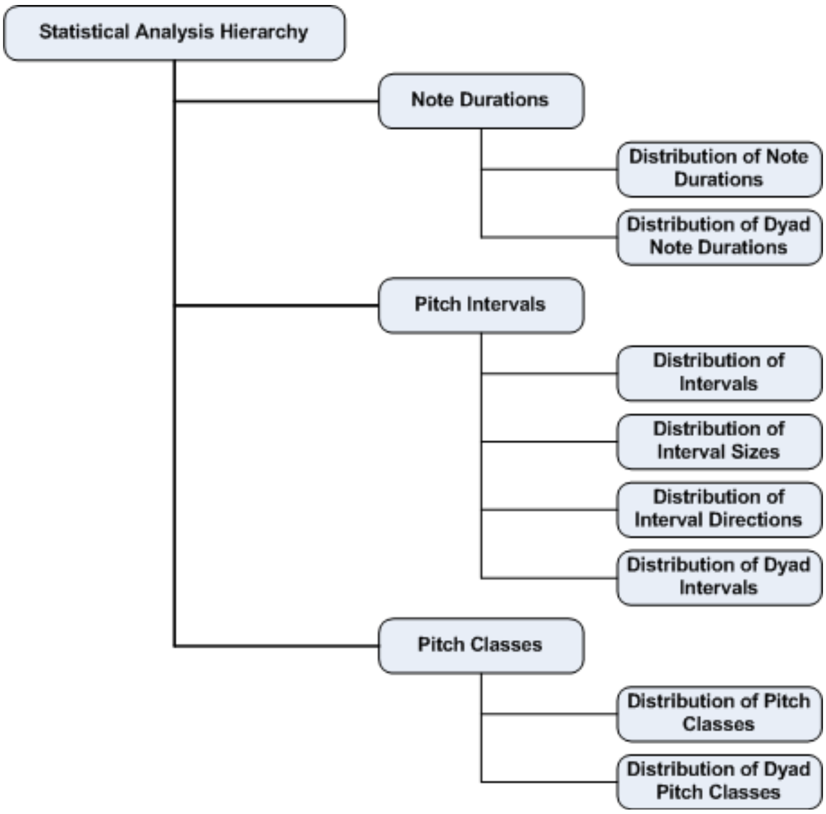


Fig. 3.1 Hierarchy of statistical analysis options addressed in the software implementation.

The statistical analysis hierarchy shown in Fig. 3.1 include three categories of information, namely note durations, pitch intervals and pitch classes. For each category, the statistical properties of a number of feature vectors associated with the category are determined in the form of distributions or histograms.

3.2.2 Statistical representations of note durations

3.2.2.1 Introduction

Fig. 3.2 shows a graphical representation of the durations of the standard note values, i.e. $\frac{1}{16}$, $\frac{1}{8}$, $\frac{1}{4}$, etc., as well as single dotted notes, in the range from a sixty-fourth note (one sixteenth of a beat) to a longa (sixteen beats). It is clear that the durations of successive note values follow a nonlinear logarithmic relationship. Thus, for the purpose of defining a histogram of note durations, whether for single note events or dyad note event pairs, the histogram bin centres are also defined on a logarithmic scale with base 2 as shown in Table 3.1. This approach is also employed in the MIDI toolbox, where the note durations are limited to the range from a sixteenth note to a whole note. This range has been extended in the current implementation to range from a sixty-fourth note to a longa.

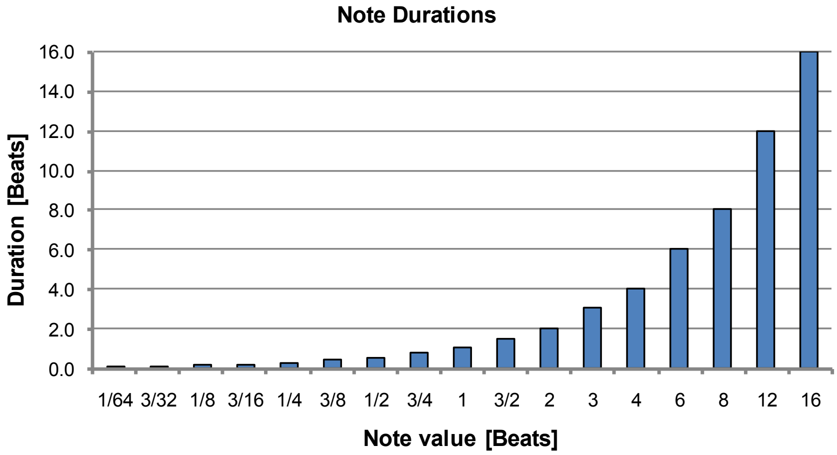


Fig. 3.2 Standard note durations in beats.

Fig. 3.3 displays the note durations and bin centres given in Table 3.1 graphically on a logarithmic scale with a base of 2. The bin centres for the dotted note values do not coincide exactly with the corresponding note durations, unlike the bin centres for standard notes. This is a direct result of the logarithmic relationship between the standard note durations and the linear relationship applicable for extending note durations through the dotted convention.

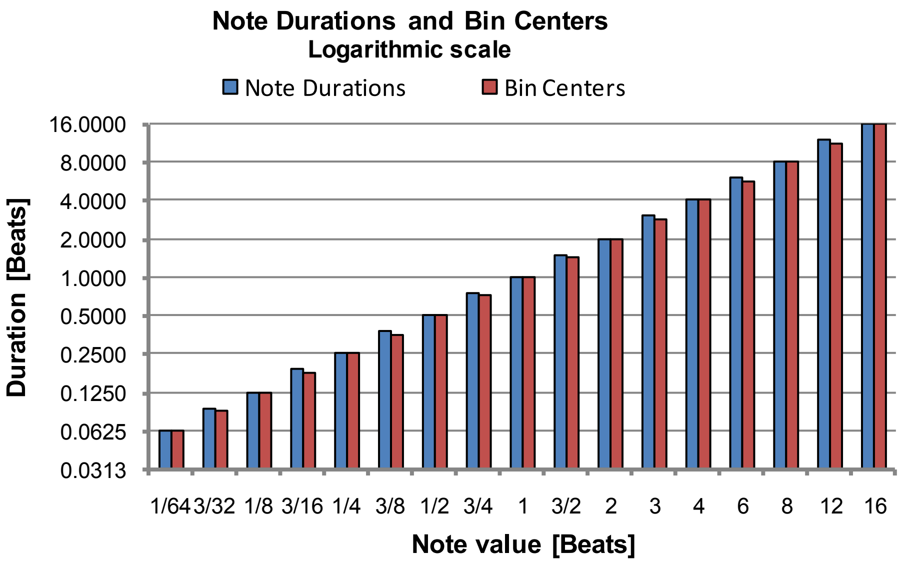


Fig. 3.3 Standard note durations and bin centres on a base 2 logarithmic scale.

Table 3.1 Note durations for determining the histograms of note durations.

Note value		**Note Durations**			**Histogram Bin Centres**	
		Duration (D) [Beats]		**$2\log_2 D$**	**Duration (D)** [Beats]	**$2\log_2 D$**
1	Sixty-fourth note	$^1/_{16}$	0.06250	-8.000000	0.06250	$2\log_2 \frac{1}{16} = -8$
2	Dotted sixty-fourth note	$^3/_{32}$	0.09375	-6.830075	0.08839	$2\log_2 \frac{\sqrt{2}}{16} = -7$
3	Thirty-second note	$^1/_8$	0.12500	-6.000000	0.12500	$2\log_2 \frac{1}{8} = -6$
4	Dotted thirty-second note	$^3/_{16}$	0.18750	-4.830075	0.17678	$2\log_2 \frac{\sqrt{2}}{8} = -5$
5	Sixteenth note	$^1/_4$	0.25000	-4.000000	0.25000	$2\log_2 \frac{1}{4} = -4$
6	Dotted sixteenth note	$^3/_8$	0.37500	-2.830075	0.35355	$2\log_2 \frac{\sqrt{2}}{4} = -3$
7	Eighth note	$^1/_2$	0.50000	-2.000000	0.50000	$2\log_2 \frac{1}{2} = -2$
8	Dotted eighth note	$^3/_4$	0.75000	-0.830075	0.70711	$2\log_2 \frac{\sqrt{2}}{2} = -1$
9	Quarter note	1	1.00000	0	1.00000	$2\log_2 1 = 0$
10	Dotted quarter note	$^3/_2$	1.50000	1.169925	1.41421	$2\log_2 \sqrt{2} = 1$
11	Half note	2	2.00000	2.000000	2.00000	$2\log_2 2 = 2$
12	Dotted half note	3	3.00000	3.169925	5.65685	$2\log_2 2\sqrt{2} = 3$
13	Whole note	4	4.00000	4.000000	4.00000	$2\log_2 4 = 4$
14	Dotted whole note	6	6.00000	5.169925	11.31371	$2\log_2 4\sqrt{2} = 5$
15	Double whole note	8	8.00000	6.000000	8.00000	$2\log_2 8 = 6$
16	Dotted double whole note	12	12.00000	7.169925	2.82843	$2\log_2 8\sqrt{2} = 7$
17	Longa	16	16.00000	8.000000	16.00000	$2\log_2 16 = 8$

The software implementation allows for the following statistical representations of note durations information:

- Distribution of note durations.
- Distribution of dyad note durations.

A distribution can be normalised with reference to the cumulative sum of the values contained in the distribution.

The remainder of this section discusses the algorithms implemented for determining the above distributions.

3.2.2.2 Distribution of note durations

The distribution of note durations is defined as a histogram of the note durations in beats. Mathematically, using set notation, the algorithm for deriving the distribution of note durations can be summarised as follows:

- *The set representing the durations in beats of the note events in the score is defined:*

  This yields the set

  $$D = \{\mathrm{d_i}\} \qquad (3.1)$$

  where d_i denotes the duration in beats of the i^{th} note event.

- *The subset of non-zero durations are extracted from D:*

  This yields the set

$$D^{'} = \left\{ \mathrm{d}_j^{'} \in D \middle| d_j^{'} > 0 \right\} \tag{3.2}$$

where d_j denotes the duration in beats of the j[th] note event with non-zero duration.

- *The durations are transformed to a logarithmic scale:*

  This yields the set

  $$D^{"} = \left\{ \mathrm{d}_j^{"} \middle| \mathrm{d}_j^{"} = 2\log_2 \mathrm{d}_j^{'} \ , \mathrm{d}_j^{'} \in D^{'} \right\}. \tag{3.3}$$

- *The set of histogram bins for analysis is defined:*

  The note durations given in Table 3.1 yields the following set of logarithmic histogram bin centres as a basis for the various distributions of note durations:

  $$B = \left\{ -8,-7,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7,8 \right\}. \tag{3.4}$$

  The analysis is performed for a subset of B reflecting the desired range of analysis. This subset is defined by imposing lower and upper limits, denoted by B_L and B_U respectively, on the elements of B. This yields the set

  $$B^{'} = \left\{ b_k^{'} \in B \middle| B_L \le b_k^{'} \le B_U \right\} \tag{3.5}$$

  where $b_k^{'}$ denotes the centre of the k[th] bin.

- *A histogram of the logarithmic durations contained in $D"$ is derived:*

  The distribution of note durations represents the number of occurrences of the durations contained in $D^{"}$ for each of the bins contained in $B^{'}$. Mathematically, it is represented by the relationship

  $$H = \left\{ h_k \right\} \tag{3.6}$$

  where h_k denotes the number of durations registered for bin $b_k^{'}$, i.e.

  $$h_k = \left| \left\{ d_j^{"} \in D^{"} \middle| b_k^{'} - 0.5 < d_j^{"} \le b_k^{'} + 0.5 \right\} \right|. \tag{3.7}$$

The histogram can be normalised with respect to the total number of the durations contained in the histogram, using the relationship

$$h_k = \frac{h_k}{\sum_{h_k \in H} h_k}. \tag{3.8}$$

3.2.2.3 Distribution of dyad note durations

The distribution of dyad note durations is defined as a two-dimensional histogram of the note durations in beats of dyads or pairs of successive notes. In order to define such successive note pairs unambiguously, the score is required to be monophonic. Mathematically, using set notation, the algorithm for deriving the distribution of note durations can be summarised as follows:

- *The set $D^{"}$ representing the logarithmic non-zero durations in beats of the note events in the score as defined by (3.1) to (3.3) is derived.*
- *The set $B^{'}$ representing the histogram bins for analysis as defined by (3.4) and (3.5) is derived.*

- *The set representing pairs of successive note durations is derived:*

  This yields the set

  $$D''' = \{d'''_m\} = \{(d'''_{m1}, d'''_{m2})\} = \{(d''_j, d''_{j+1}) | d''_j, d''_{j+1} \in D''\} \tag{3.9}$$

  where d_j and d_{j+1} represent two successive durations in D''.

- *The set representing the histogram bin combinations for analysis is derived:*

  This yields the set

  $$B'' = \{b''_{k1,k2}\} = \{(b''_{k1}, b''_{k2}) | B''_{k1}, B''_{k2} \in B'\} = B' \times B'. \tag{3.10}$$

- *A histogram of the pairs of durations contained in D''' is derived:*

  The distribution of dyad note durations represents a count of the number of occurrences of the pairs of successive durations contained in D''' for each pair of bin centres contained in B''. Mathematically, it is represented by the relationship

  $$H = \{h_{k1,k2}\} \tag{3.11}$$

  where $h_{k1,k2}$ denotes the number of successive duration pairs registered for bins combination (b''_{k1}, b''_{k2}), i.e.

  $$h_{k1,k2} = \left| \left\{ \begin{array}{l} (d''_j, d''_{j+1}) | b''_{k1} - 0.5 < d''_{j1} \le b''_{k1} + 0.5, b''_{k2} - 0.5 < d''_{j+1} \le b''_{k2} + 0.5; \\ d''_j, d''_{j+1} \in D''; b''_{k1}, b''_{k2} \in B'' \end{array} \right\} \right|. \tag{3.12}$$

The histogram can be normalised with respect to the total number of duration pairs contained in the histogram, using the relationship

$$h_{k1,k2} = \frac{h_{k1,k2}}{\sum_{h \in H} h}. \tag{3.13}$$

3.2.3 Statistical representations of pitch intervals

3.2.3.1 Introduction

The distributions of pitch intervals describe the statistical properties of the pitch intervals between pairs or dyads of successive notes. In order to define the successive note pairs unambiguously, the score is required to be monophonic. This implies that a melodic line or single voice must be extracted from a polyphonic score before analysis. Table 3.2 summarises the interval sizes, i.e. absolute values of the intervals in semitones, implemented in the MIDI toolbox and the software application project.

The software implementation allows for the following statistical representations of pitch intervals:

- Distribution of pitch intervals.
- Distribution of pitch interval sizes.
- Distribution of pitch interval directions.
- Distribution of dyad pitch intervals.

Table 3.2 Interval sizes implemented for the statistical analysis of intervals.

Interval	**Interval size [Semitones]**	**Interval**	**Interval size [Semitones]**
Unison (**P1**)	0	Perfect fifth (**P5**)	7
Minor second (**m2**)	1	Minor sixth (**m6**)	8
Major second (**M2**)	2	Major sixth (**M6**)	9
Minor third (**m3**)	3	Minor seventh (**m7**)	10
Major third (**M3**)	4	Major seventh (**M7**)	11
Perfect fourth (**P4**)	5	Perfect eighth (**P8**)	12
Diminished fifth (**d5**)	6		

The software application allows for both unweighted and weighted calculations of the above distributions of intervals. These are defined as follows:

- *Unweighted distributions:*

  An unweighted distribution represents a measure of the occurrence of the relevant statistical measure of the intervals, i.e. intervals, interval sizes, interval directions or dyad intervals. The MIDI toolbox does not provide for the unweighted interval distributions. The software implementation allows for the following two measures of the frequency of occurrence:

  - A cumulative distribution where the total count or histogram is derived.
  - A non-cumulative distribution where the occurrence is logged only once.

- *Weighted distributions:*

  A weighted distribution represents the cumulative sum of the weighted durations associated with the statistical measure, i.e. intervals, interval sizes, interval directions or dyad intervals. Two weighing options are implemented, namely linear weighing as implemented by Krumhansl and Schmuckler (Krumhansl, 1990) and weighing according to durational accent (Parncutt, 1994).

A distribution can be normalised with reference to the cumulative sum of the values contained in the distribution.

The remainder of this section discusses the algorithms implemented for determining the above distributions.

3.2.3.2 Distribution of pitch intervals

The unweighted and weighted distribution of pitch intervals are defined as a measure of the occurrences or the sum of the weighted durations respectively of each interval targeted in the analysis for the score material subjected to the analysis. These distributions take cognizance of the interval direction, which implies that positive and negative pitch transitions are categorised separately. Mathematically, using set notation, the algorithm for deriving the unweighted and weighted distributions of pitch intervals can be summarised as follows:

- *The set representing the pitch values of the note events in the score is defined:*

  This yields the set

  $$P = \{p_{\mathrm{i}}\} \tag{3.14}$$

  where p_i denotes the MIDI pitch value of the i^th^ note event. The MIDI representation of pitch values implies the relationship

  $$p_i \in ¢ \,|\, 0 \le p_i \le 255. \tag{3.15}$$

- *The set representing the pitch intervals between successive events is defined:*

  This yields the set

$$P' = \{p'_j\} = \{p_{i+1} - p_i \mid p_i, p_{i+1} \in P\} \tag{3.16}$$

where p'_j denotes the jth interval, i.e. the interval between the ith and (i+1)th pitch values in semitones.

- *The set representing the durations of the note events in the score is defined:*

  This yields the set

  $$D = \{d_i\} \tag{3.17}$$

  where d_i denotes the duration in beats or seconds of the ith note event.

- *The set representing the unweighted or weighted durations of the note events in the score is derived:*

  This yields the set

  $$D' = \{d'_i\} \tag{3.18}$$

  where d'_i denotes the weighted duration of the ith note event. The weighted duration d'_i is determined by the applicable durational weighing option. The following cases are relevant:

  - *No durational weighing:*

    $$d'_i = 0.5 \tag{3.19}$$

    This option is implemented in the software application, but not in the MIDI toolbox.

  - *Linear durational weighing according to the individual note durations:*

    $$d'_i = d_i \tag{3.20}$$

    where d_i denotes the duration in beats or seconds of the ith note event. This weighing option is used in the Krumhansl and Schmuckler key-finding algorithm as implemented in the MIDI toolbox.

  - *Durational weighing according to durational accent:*

    $$d'_i = \left(1 - e^{-\frac{d_i}{\tau}}\right)^{I_a} \tag{3.21}$$

    where d_i denotes the duration in seconds of the ith note event, τ denotes the saturation duration and I_a denotes the accent index (Parncutt, 1994).

- *The set representing the weighted durations associated with the pitch intervals contained in P' is derived:*

  This yields the set

  $$D'' = \{d''_j\} = \{d'_i + d'_{i+1} \mid d'_i, d'_{i+1} \in D'\} \tag{3.22}$$

  where d''_j denotes the weighted duration for the jth interval.

- *The set representing the pitch intervals for analysis is defined:*

Adding a sign to the interval sizes defined in Table 3.2 to take cognizance of the directions of the transitions, i.e. upwards or downwards, yields the following set of intervals as a basis for the analysis:

$$B = \{-12,-11,-10,-9,-8,-7,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7,8,9,10,11,12\} \quad (3.23)$$

where negative and positive interval values indicate downwards and upwards pitch transitions respectively. The analysis is performed for a subset of B reflecting the desired range of analysis. This subset is defined by imposing lower and upper limits, denoted by B_L and B_U respectively, on the elements of B. This yields the set

$$B^{'} = \left\{ b_k^{'} \in B \,\middle|\, B_L \le b_k^{'} \le B_U \right\}. \quad (3.24)$$

where $b_k^{'}$ denotes the kth interval targeted in the analysis.

- *The distribution of pitch intervals is derived:*

  The noncumulative unweighted distribution of pitch intervals reflects a value of one for each of the pitch intervals contained in $B^{'}$ that is present in $P^{'}$ and zero for all others. Mathematically, it is represented by the relationship

  $$H = \{h_k\} \quad (3.25)$$

  where

  $$h_k = \begin{cases} 0 & b_k^{'} \notin P^{'} \\ 1 & b_k^{'} \in P^{'} \end{cases}. \quad (3.26)$$

  The cumulative unweighted distribution of pitch intervals represents a count or histogram of the occurrence of each of the intervals contained in $B^{'}$ for the pitch intervals contained in $P^{'}$. Mathematically, it is represented by the relationship

  $$H = \{h_k\} \quad (3.27)$$

  where

  $$h_k = \left| \left\{ p_j^{'} \in P^{'} \,\middle|\, p_j^{'} = b_k^{'} \right\} \right|. \quad (3.28)$$

  The weighted distribution of pitch intervals represents the cumulative sum of the weighted durations associated with each of the pitch intervals contained in $B^{'}$ for the pitch intervals contained in $P^{'}$. Mathematically, it is represented by the relationship

  $$H = \{h_k\} \quad (3.29)$$

  where h_k denotes the sum of the weighted durations registered for the kth pitch interval, i.e.

  $$h_k = \sum_{\left\{ d_j^{''} \in D^{''} \,\middle|\, p_j^{'} = b_k^{'} \right\}} d_j^{''}. \quad (3.30)$$

The distribution of pitch intervals can be normalised with respect to the sum of the values contained in the distribution, using the relationship

$$h_k = \frac{h_k}{\sum_{\{h \in H\}} h}. \tag{3.31}$$

3.2.3.3 Distribution of pitch interval sizes

The unweighted and weighted distribution of pitch interval sizes are defined as a measure of the occurrences or sum of the weighted durations respectively of each interval size, i.e. absolute value of the interval, targeted in the analysis for the score material subjected to the analysis. Mathematically, using set notation, the algorithm for deriving the distribution of pitch interval sizes can be summarised as follows:

- *The set representing the absolute values of the pitch intervals between successive events is derived from P defined in (3.14):*

  This yields the set

  $$P' = \left\{p_j'\right\} = \left\{\left|P_{i+1} - P_i\right| \middle| P_i, P_{i+1} \in P\right\} \tag{3.32}$$

  where p_j' denotes the absolute value of the jth interval, i.e. the interval between the ith and (i+1)th pitch values in semitones.

- *The set D'' representing the weighted durations associated with the pitch interval sizes contained in P' as defined by* (3.22) *is derived:*

- *The set representing the pitch intervals for analysis is defined:*

  Only the positive intervals defined in Table 3.2 are relevant for the distribution of pitch interval sizes. This yields the following set of intervals as a basis for the analysis:

  $$\mathbf{B} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}. \tag{3.33}$$

  The analysis is performed for a subset of B reflecting the desired range of analysis. This subset is defined by imposing lower and upper limits, denoted by B_L and B_U respectively, on the elements of B. This yields the set

  $$B' = \left\{b_k' \in B \middle| B_L \le b_k' \le B_U\right\} \tag{3.34}$$

  where b_k' denotes the kth interval targeted in the analysis.

- *The distribution of pitch interval sizes is derived:*

  The noncumulative unweighted distribution of pitch interval sizes reflects a value of one for each of the pitch intervals contained in B' that is present in P' and zero for all others. The distribution is determined as defined by (3.25) and (3.26).

  The cumulative unweighted distribution of pitch interval sizes represents a count or histogram of the occurrence of each of the intervals contained in B' for the pitch intervals contained in P'. The distribution is determined as defined by (3.27) and (3.28).

  The weighted distribution of pitch interval sizes represents the cumulative sum of the weighted durations associated with each of the intervals contained in B' for the pitch intervals contained in P'. The distribution is determined as defined by (3.29) and (3.30).

The distribution of pitch interval sizes can be normalised with respect to the sum of the values contained in the distribution as defined by (3.31).

3.2.3.4 Distribution of pitch interval directions

The unweighted and weighted distribution of pitch interval directions are defined as the difference between the measure of the occurrences or difference between the sums of the weighted durations of positive and negative intervals of the same size respectively, for the interval sizes targeted in the analysis for the score material subjected to the analysis. Mathematically, using set notation, the algorithm for deriving the distribution of pitch interval directions can be summarised as follows:

- *The set $P^{'}$ representing pitch intervals between successive events as defined by (3.14) to (3.16) is derived.*
- *The set $D^{''}$ representing the weighted durations associated with the pitch intervals sizes contained in $P^{'}$ as defined by* (3.22) *is derived:*
- *The set $B^{'}$ representing the pitch intervals for analysis as defined by (3.33) and (3.34) is derived.*
- *The distribution of interval directions is derived:*

  The noncumulative unweighted distribution of pitch interval directions reflects a value of one if the difference between the counts of the occurrences of positive and negative intervals of the same size contained in $B^{'}$ for the pitch intervals contained in $P^{'}$ is not zero and zero for all others. Mathematically, it is represented by the relationship

  $$H = \{h_k\} \tag{3.35}$$

  where

  $$h_k = \begin{cases} 0 & \left|\left\{p_j^{'} \in P^{'} \middle| p_j^{'} = b_k^{'}\right\}\right| - \left|\left\{p_j^{'} \in P^{'} \middle| p_j^{'} = -b_k^{'}\right\}\right| = 0 \\ 1 & \left|\left\{p_j^{'} \in P^{'} \middle| p_j^{'} = b_k^{'}\right\}\right| - \left|\left\{p_j^{'} \in P^{'} \middle| p_j^{'} = -b_k^{'}\right\}\right| \neq 0 \end{cases}. \tag{3.36}$$

  The cumulative unweighted distribution of pitch interval directions represents the difference between the counts of the occurrences of positive and negative intervals of the same size contained in $B^{'}$ for the pitch intervals contained in $P^{'}$. Mathematically, it is represented by the relationship

  $$H = \{h_k\} \tag{3.37}$$

  where

  $$h_k = \left|\left\{p_j^{'} \in P^{'} \middle| p_j^{'} = b_k^{'}\right\}\right| - \left|\left\{p_j^{'} \in P^{'} \middle| p_j^{'} = -b_k^{'}\right\}\right|. \tag{3.38}$$

  The weighted distribution of pitch interval directions represents the difference between the cumulative sums of the weighted durations of positive and negative intervals of the same size contained in $B^{'}$ for the pitch intervals contained in $P^{'}$. Mathematically, it is represented by the relationships

  $$H = \{h_k\} \tag{3.39}$$

  where h_k denotes the difference between the sum of the weighted durations registered for positive and negative values of the kth pitch interval, i.e.

  $$h_k = \sum_{\left\{d_j^{''} \in D^{''} \middle| p_j^{'} = b_k^{'}\right\}} d_j^{''} - \sum_{\left\{d_j^{''} \in D^{''} \middle| p_j^{'} = -b_k^{'}\right\}} d_j^{''}. \tag{3.40}$$

The distribution of pitch interval directions can be normalised with respect to the sum of the values contained in the distribution as defined by (3.31).

3.2.3.5 Distribution of dyad pitch intervals

The unweighted and weighted distribution of dyad pitch intervals are defined as a measure of the occurrences or sum of the weighted durations respectively of dyads or pairs of successive intervals, for all two-element combinations of the intervals targeted in the analysis for the score material subjected to the analysis. In order to define such successive dyad note pairs unambiguously, the score is required to be monophonic. Mathematically, the algorithm for deriving the distribution of dyad pitch intervals can be summarised as follows:

- *The set* $P^{'}$ *representing pitch intervals between successive events as defined by (3.16) is derived.*
- *The set* $D^{''}$ *representing the weighted durations associated with the pitch intervals sizes contained in* $P^{'}$ *as defined by* (3.22) *is derived:*
- *The set* $B^{'}$ *representing the pitch intervals for analysis as defined by* (3.23) *and (3.24) is derived.*
- *The set representing the combinations of pitch intervals for analysis is derived:*

  This yields the set

  $$B^{''} = \left\{b^{''}_{k1,k2}\right\} = \left\{\left(b^{''}_{k1}, b^{''}_{k2}\right) \middle| b^{''}_{k1}, b^{''}_{k2} \in B^{'}\right\} = B^{'} \times B^{'}. \tag{3.41}$$

- *The distribution of dyad intervals is derived:*

  The noncumulative unweighted distribution of dyad pitch intervals reflects a value of one for each pair of intervals contained in $B^{''}$ that is present in the successive pitch intervals contained in $P^{'}$ and zero for all others. Mathematically, it is represented by the relationship

  $$H = \left\{h_{k1,k2}\right\} \tag{3.42}$$

  where

  $$h_k = \begin{cases} 0 & \left\{\left(b^{''}_{k1}, b^{''}_{k2}\right) \middle| \left(b^{''}_{k1}, b^{''}_{k2}\right) \in B^{''}\right\} \notin \left\{\left(p^{'}_i, p^{'}_{i+1}\right) \middle| p^{'}_i, p^{'}_{i+1} \in P^{'}\right\} \\ 1 & \left\{\left(b^{''}_{k1}, b^{''}_{k2}\right) \middle| \left(b^{''}_{k1}, b^{''}_{k2}\right) \in B^{''}\right\} \in \left\{\left(p^{'}_i, p^{'}_{i+1}\right) \middle| p^{'}_i, p^{'}_{i+1} \in P^{'}\right\} \end{cases}. \tag{3.43}$$

  The cumulative unweighted distribution of dyad pitch intervals represents a count or histogram of the number of occurrences of each pair of intervals contained in $B^{''}$ for the pairs of successive intervals contained in $P^{'}$. Mathematically, it is represented by the relationship

  $$H = \left\{h_{k1,k2}\right\} \tag{3.44}$$

  where

  $$h_{k1,k2} = \left|\left\{\left(p^{'}_j, p^{'}_{j+1}\right) \middle| p^{'}_j = b^{''}_{k1}, p^{'}_{j+1} = b^{''}_{k2}; p^{'}_j, p^{'}_{j+1} \in P^{'}; \left(b^{''}_{k1}, b^{''}_{k2}\right) \in B^{''}\right\}\right|.$$

  The weighted distribution of dyad intervals represents the sum of the weighted durations associated with each pair of intervals contained in $B^{''}$ for the pairs of successive intervals contained in $P^{''}$. Mathematically, it is represented by the relationship

  $$H = \left\{h_{k1,k2}\right\} \tag{3.45}$$

where $h_{k1,k2}$ denotes the sum of the durational weights associated with the successive interval pairs registered for intervals combination $\left(b''_{k1}, b''_{k2}\right)$, i.e.

$$h_{k1,k2} = \sum\left(d''_j + d''_{j+1}\right) \middle| p'_j = b''_{k1}, p'_{j+1} = b''_{k2}; d''_j, d''_{j+1} \in D''; p'_j, p'_{j+1} \in P'; \left(b''_{k1}, b''_{k2}\right) \in B''. \quad (3.46)$$

The distribution of dyad pitch intervals can be normalised with respect to the sum of the values contained in the distribution, using the relationship

$$h_{k1.k2} = \frac{h_{k1,k2}}{\sum_{\{h \in H\}} h}. \quad (3.47)$$

3.2.4 Statistical representations of pitch classes

3.2.4.1 Introduction

The statistical representations of pitch classes describe the statistical properties of the pitches contained in the score material. Table 3.3 summarises the base pitch classes implemented in the MIDI toolbox and the current project.

Table 3.3 Pitch classes implemented for the statistical analysis of pitch classes.

Designation	MIDI Designation	MIDI Value	Designation	MIDI Designation	MIDI Value
C	C4	60	$F^{\#}$	$F^{\#}4$	66
$C^{\#}$	$C^{\#}4$	61	G	G4	67
D	D4	62	$G^{\#}$	$G^{\#}4$	68
$D^{\#}$	$D^{\#}4$	63	A	A4	69
E	E4	64	$A^{\#}$	$A^{\#}4$	70
F	F4	65	B	B4	71

The software implementation allows for the following statistical representations of pitch classes information:

- Distribution of pitch classes.
- Distribution of dyad pitch classes.

The software application allows for both unweighted and weighted calculations of the above distributions of pitch classes. These are defined as follows:

- *Unweighted distributions:*

  An unweighted distribution represents a measure of the occurrence of the pitch classes. The MIDI toolbox does not provide for unweighted distributions of pitch class. The software application allows for the following two measures of the frequency of occurrence:

  - A cumulative distribution where the total count or histogram is derived. This represents the distributions used in the Krumhansl and Schmuckler (Krumhansl, 1990) key-finding algorithm.
  - A non-cumulative distribution where the occurrence is logged only once. This represents the distribution used by Temperley (1999) in the modified Krumhansl and Schmuckler key-finding algorithm.

- *Weighted distributions:*

  A weighted distribution represents a cumulative sum of the weighted durations associated with the pitch classes. Two weighing options are implemented, namely linear weighing as implemented by Krumhansl and Schmuckler (Krumhansl, 1990) and weighing according to durational accent (Parncutt, 1994).

A distribution can be normalised with reference to the cumulative sum of the values contained in the distribution.

The remainder of this section discusses the algorithms implemented for determining the above distributions.

3.2.4.2 Distribution of pitch classes

The unweighted and weighted distribution of pitch classes are defined as a measure of the occurrences or sum of the weighted durations respectively of the note events for the score material subjected to the analysis. Mathematically, using set notation, the algorithm for deriving the distribution of pitch classes can be summarised as follows:

- *The set representing the pitch values of the note events in the score is defined:*

  This yields the set

  $$P = \{\mathrm{p_i}\} \tag{3.48}$$

  where p_i denotes the MIDI pitch of the ith note event.

- *For each pitch value contained in P, the offset in semitones of the MIDI value relative to C in the lower octave defined by the MIDI value is defined:*

  This yields the set

  $$P' = \{p_i'\} = \{p_i \mathrm{mod}(12) | p_i \in P\} \tag{3.49}$$

  where p_i' denotes the pitch class in semitones of the ith note event relative to C.

- *The set representing the durations of the note events in the score is defined:*

  This yields the set

  $$D = \{d_i\} \tag{3.50}$$

  where d_i denotes the duration in beats or seconds of the ith note event.

- *The set representing the unweighted or weighted durations of the note events in the score is derived:*

  This yields the set

  $$D' = \{d_i'\} \tag{3.51}$$

  where d_i' denotes the weighted duration of the ith note event. The weighted duration d_i' is determined by the applicable durational weighing option. The following cases are relevant:

  - *No durational weighing:*

    $$d_i' = 1 \tag{3.52}$$

    This option is implemented in the software application, but not in the MIDI toolbox.

  - *Linear durational weighing according to the individual note durations:*

    $$d_i' = d_i \tag{3.53}$$

where d_i denotes the duration in beats or seconds of the ith note event. This weighing option is used in the Krumhansl and Schmuckler key-finding algorithm as implemented in the MIDI toolbox.

- *Durational weighing according to durational accent:*

$$d_i^{'} = \left(1 - e^{-\frac{d_i}{\tau}}\right)^{I_a} \tag{3.54}$$

where d_i denotes the duration in seconds of the ith note event, τ denotes the saturation duration and I_a denotes the accent index (Parncutt, 1994).

- *The set representing the pitch classes for analysis is defined:*

The pitch classes defined in Table 3.3 yields the following set of pitch class values as a basis for the analysis:

$$B = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\} \tag{3.55}$$

where the pitch class values represent the relative interval in semitones between the corresponding pitch class and pitch class C. The analysis is performed for a subset of B reflecting the desired range of analysis. This subset is defined by imposing lower and upper limits, denoted by B_L and B_U respectively, on the elements of B. This yields the set

$$B^{'} = \left\{b_k^{'} \in B \middle| B_L \le b_k^{'} \le B_U\right\} \tag{3.56}$$

where $b_k^{'}$ denotes the kth pitch class targeted in the analysis.

- *The distribution of pith classes is derived:*

The noncumulative unweighted distribution of pitch classes reflects a value of one for each of the pitch classes contained in $B^{'}$ that is present in $P^{'}$ and zero for all others. Mathematically, it is represented by the relationship

$$H = \{h_k\} \tag{3.57}$$

where

$$h_k = \begin{cases} 0 & b_k^{'} \notin P^{'} \\ 1 & b_k^{'} \in P^{'} \end{cases}. \tag{3.58}$$

The cumulative unweighted distribution of pitch classes represents a count or histogram of the occurrence of each of the pitch classes contained in $B^{'}$ for the pitch classes contained in $P^{'}$. Mathematically, it is represented by the relationships

$$H = \{h_k\} \tag{3.59}$$

where

$$h_k = \left|\left\{p_i^{'} \in P^{'} \middle| p_i^{'} = b_k^{'}\right\}\right|. \tag{3.60}$$

The weighted distribution of pitch classes represents the cumulative sum of the weighted durations contained in $D^{'}$ for each of the associated pitch classes contained in $B^{'}$. Mathematically, it is represented by the relationships

$$H = \left\{ h_k \right\} \tag{3.61}$$

where h_k denotes the sum of the weighted durations registered for the k[th] pitch class, i.e.

$$h_k = \sum_{\left\{ d'_j \in D' \middle| p'_j = b'_k \right\}} d'_j. \tag{3.62}$$

The distribution of pitch classes can be normalised with respect to the sum of the values contained in the distribution, using the relationship

$$h_k = \frac{h_k}{\sum_{\{h \in H\}} h}. \tag{3.63}$$

3.2.4.3 Distribution of dyad pitch classes

The unweighted and weighted distribution of dyad pitch classes are defined as a measure of the occurrences or sum of the weighted durations respectively of dyads or pairs of successive pitch classes, for all two-element combinations of the pitch classes targeted in the analysis for the score information subjected to the analysis. In order to define such successive dyad note pairs unambiguously, the score is required to be monophonic. Mathematically, the algorithm for deriving the unweighted distribution of dyad pitch classes can be summarised as follows:

- *The set P' representing the pitch classes of the note events in the score as defined by (3.49) is derived.*
- *The set D' representing the unweighted or weighted durations of the note events in the score as defined by* (3.51) *is derived.*
- *The set B' representing the pitch intervals for analysis as defined by* (3.55) and *(3.56) is derived.*
- *The set representing the combinations of pitch intervals for analysis is derived:*

  This yields the set

$$B'' = \left\{ b''_{k1,k2} \right\} = \left\{ \left(b''_{k1}, b''_{k2} \right) \middle| b''_{k1}, b''_{k2} \in B' \right\} = B' \times B'. \tag{3.64}$$

- *The distribution of dyad pitch classes is derived:*

  The noncumulative unweighted distribution of dyad pitch classes reflects a value of one for each pair of pitch classes contained in B'' that is present in the successive pitch classes contained in P'' and zero for all others. Mathematically, it is represented by the relationships

$$H = \left\{ h_{k1,k2} \right\} \tag{3.65}$$

  where

$$h_k = \begin{cases} 0 & \left\{ \left(b''_{k1}, b''_{k2} \right) \middle| \left(b''_{k1}, b''_{k2} \right) \in B'' \right\} \notin \left\{ \left(p'_i, p'_{i+1} \right) \middle| p'_i, p'_{i+1} \in P' \right\} \\ 1 & \left\{ \left(b''_{k1}, b''_{k2} \right) \middle| \left(b''_{k1}, b''_{k2} \right) \in B'' \right\} \in \left\{ \left(p'_i, p'_{i+1} \right) \middle| p'_i, p'_{i+1} \in P' \right\} \end{cases}. \tag{3.66}$$

  The cumulative unweighted distribution of dyad pitch classes represents a count or histogram of the number of occurrences of each pair of pitch classes contained in B'' for the pairs of successive pitch classes contained in P''. Mathematically, it is represented by the relationship

$$H = \left\{ h_{k1,k2} \right\} \tag{3.67}$$

where

$$h_{k1,k2} = \left| \left\{ \left(p_i^{'}, p_{i+1}^{'} \right) \middle| p_i^{'} = b_{k1}^{"}, p_{i+1}^{'} = b_{k2}^{"}; p_i^{'}, p_{i+1}^{'} \in P^{'}; \left(b_{k1}^{"}, b_{k2}^{"} \right) \in B^{"} \right\} \right|. \tag{3.68}$$

The weighted distribution of dyad pitch classes represents the cumulative sum of the products of the weighted durations associated with each pair of pitch classes contained in $B^{"}$ for the pairs of successive pitch classes contained in $P^{"}$. Mathematically, it is represented by the relationship

$$H = \left\{ h_{k1,k2} \right\} \tag{3.69}$$

where $h_{k1,k2}$ denotes the sum of the products of weighted durations associated with the successive pitch class pairs registered for pitch classes combination $\left(b_{k1}^{"}, b_{k2}^{"} \right)$, i.e.

$$h_{k1,k2} = \sum d_i^{'}.d_{i+1}^{'} \,|\, p_i = b_{k1}^{"}, p_{i+1} = b_{k2}^{"}; d_i^{'}, d_{i+1}^{'} \in D^{'}; p_i^{'}, p_{i+1}^{'} \in P^{'}; \left(b_{k1}^{"}, b_{k2}^{"} \right) \in B^{"}. \tag{3.70}$$

The distribution of dyad pitch classes can be normalised with respect to the sum of the values contained in the distribution, using the relationship

$$h_{k1.k2} = \frac{h_{k1,k2}}{\sum\limits_{\{h \in H\}} h}. \tag{3.71}$$

3.3 Tonality analysis

3.3.1 Introduction

The tonality analysis implemented in the software application focuses on the key–finding algorithm developed by Krumhansl and Schmuckler (Krumhansl, 1990). This algorithm, discussed in section 2.7.3, determines the key or tonal centre of a particular musical passage by correlating the durations of the pitch classes of the test material with the key profiles determined by Krumhansl and Kessler (1982) for the major and minor modes.

3.3.1.1 Implementation of the Krumhansl and Schmuckler key-finding algorithm

The software application offers the following extended functionality of the Krumhansl and Schmuckler key-finding algorithm:

- *Key profiles*: Two options are implemented, namely the Krumhansl and Kessler (1982) psychological key profiles and the modified key profiles proposed by Temperley (1999).
- *Durational weighing*: Four options are implemented, namely weighing according to the frequency of occurrence, flat weighing as proposed by Temperley (1999), linear weighing as implemented by Krumhansl and Schmuckler (Krumhansl, 1990) and weighing according to the durational accent (Huron and Parncutt, 1993).
- *Key correlation formula*: Two options are implemented, namely the formula implemented by Krumhansl and Schmuckler (Krumhansl, 1990) and the modified formula proposed by Temperley (1999; 2002).

Mathematically, using set notation and allowing for the extended options detailed above, the Krumhansl and Schmuckler (Krumhansl, 1990) key-finding algorithm can be summarised as follows:

- *The set representing the MIDI pitches of the note events in the score is defined:*

This yields the set

$$P = \{\mathrm{p_i}\} \tag{3.72}$$

where p_i denotes the MIDI pitch of the ith note event.

- *For each pitch value contained in P, the offset in semitones of the MIDI value relative to C in the lower octave defined by the pitch value is defined:*

  This yields the set

  $$P^{'} = \{p_i^{'}\} = \{p_i \mathrm{mod}(12) | p_i \in P\} \tag{3.73}$$

  where $p_i^{'}$ denotes the pitch in semitones of the ith note event relative to C.

- *The set representing the durations of the note events in the score is defined:*

  This yields the set

  $$D = \{d_i\} \tag{3.74}$$

  where d_i denotes the duration in beats or seconds of the ith note event.

- *The set representing the unweighted or weighted durations of the note events in the score is derived:*

  This yields the set

  $$D^{'} = \{d_i^{'}\} \tag{3.75}$$

  where $d_i^{'}$ denotes the weighted duration of the ith note event. The weighted duration $d_i^{'}$ is determined by the applicable durational weighing option. The following cases are relevant:

  - *No durational weighing:*

    $$d_i^{'} = 1 \tag{3.76}$$

    This option is implemented in the software application, but not in the MIDI toolbox.

  - *Linear durational weighing according to the individual note durations:*

    $$d_i^{'} = d_i \tag{3.77}$$

    where d_i denotes the duration in beats or seconds of the ith note event. This weighing option is used in the Krumhansl and Schmuckler key-finding algorithm.

  - *Durational weighing according to the durational accent:*

    $$d_i^{'} = \left(1 - e^{-\frac{d_i}{\tau}}\right)^{I_a} \tag{3.78}$$

    where d_i denotes the duration in seconds of the ith note event, τ denotes the saturation duration and I_a denotes the accent index (Parncutt, 1994).

- *The set representing the pitch classes for analysis is defined:*

  The pitch classes defined in Table 3.3 yields the following set of pitch class values as a basis for the analysis:

$$B = \{b_k\} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\} \tag{3.79}$$

where each pitch class value represents the relative interval in semitones between the corresponding pitch class and pitch class C.

- *The distribution of pitch classes is derived:*

  The distribution of pitch classes represents the cumulative sum of the durations contained in $D^{'}$ for each of the associated pitch classes contained in $P^{'}$. Mathematically, it is represented by the relationship

$$H = \{h_k\} \tag{3.80}$$

where h_k denotes the sum of the weighted durations registered for the kth pitch class, i.e.

$$h_k = \sum_{\left\{d_j^{'} \in D^{'} \middle| p_j^{'} = b_k\right\}} d_j^{'}. \tag{3.81}$$

- *The sets representing the key profile coefficients of the major and minor modes in C are defined:*

  This yields the sets

$$K^{maj} = \left\{\mathrm{k}_n^{maj} \middle| 1 \le n \le 12\right\} \tag{3.82}$$

and

$$K^{\min} = \left\{\mathrm{k}_n^{\min} \middle| 1 \le n \le 12\right\} \tag{3.83}$$

where k_n^{maj} and $k_n^{\min}$ denote the weights of the nth pitch classes for the major and minor modes respectively.

- The key profiles for the 12 major keys and 12 minor keys are derived. This yields the sets

$$K^{'} = \left\{\mathrm{k}_{\mathrm{mn}}^{'} \middle| 1 \le m \le 24, 1 \le n \le 12\right\} \tag{3.84}$$

where

$$k_{mn}^{'} = k_{(n-m)\mathrm{mod}(12)}^{maj} \ \textit{for}\ 1 \le m \le 12, 1 \le n \le 12 \tag{3.85}$$

and

$$k_{mn}^{'} = k_{(n-m)\mathrm{mod}(12)+1}^{\min} \ \textit{for}\ 13 \le m \le 24, 1 \le n \le 12 \tag{3.86}$$

where $k_{mn}^{'}$ denotes the weight of the nth pitch class of mth key profile. Index values $m = 1$ to 12 correspond to the set of major keys while index values $m = 13$ to 24 correspond to the set of minor keys.

- *The correlation coefficients between the set of weighted durations $D^{'}$ and the set of key profile vectors is derived:*

  This yields the set

$$\Upsilon = \{\tau_m\} \tag{3.87}$$

where τ_m denotes the correlation between the weighted durations profile $D^{'}$ and the m[th] key profile. The value of τ_m is determined by the applicable key correlation formula option. The following cases are implemented:

- *Key correlation according to the Krumhansl and Schmuckler formula (Krumhansl, 1990):*

$$\tau_m = \frac{\sum_{n=1}^{12}\left(d_n - \overline{d}\right)\left(k_{mn} - \overline{k_m}\right)}{\sqrt{\sum_{n=1}^{12}\left(d_n - \overline{d}\right)^2 \sum_{n=1}^{12}\left(k_{mn} - \overline{k_m}\right)^2}} \tag{3.88}$$

where $\overline{d}$ and $\overline{k_m}$ denote the means of d_n and k_{mn} respectively for $n = 1$ to 12.

- *Key correlation according to the Temperley formula (Temperley, 1999):*

$$\tau_m = \sum_{n}^{12} d_n k_{mn} \tag{3.89}$$

Set Υ contains the correlations between the weighted durations profile $D^{'}$ and each of the twenty-four possible key and mode combinations contained in $K^{'}$. The maximum correlation value indicates the most likely key, subject to the following considerations:

It is possible to have more than one key that yields the same maximum correlation value. This is particularly true for the Temperley key profile because of the equal weights assigned to multiple pitch classes, particularly the chromatic pitch classes. In practice, this has the following implications for the implementation of the algorithm:

- Due to numerical rounding effects in calculating the key correlation coefficients, keys that should have identical key correlation scores can end up with slightly different scores.
- The limited note duration resolution that results from the limitations of the MIDI file format can also give rise to discrepancies in key correlation coefficients that should have identical values, or an incorrect magnitude hierarchy for key correlation values that are numerically close.

It follows that an algorithm that identifies the key solely on the basis of the maximum key correlation coefficient obtained can deliver incorrect or incomplete results. The solution, as implemented in the software application, is to introduce the following:

- The most likely key or keys are identified based on a continuous region of key correlation values that is bordered on the upper side by the maximum key correlation value. This approach is supported by the fact that music theoretical analysis often identifies a set of candidate key rather than a single key for a particular passage.
- The option is introduced to quantisize note durations to a predetermined resolution chosen from the options.

4 Program implementation and validation

4.1 Overview

One of the two main objectives of this study is the implementation in Matlab of a menu-driven, user-friendly version of the analysis tools available in the Midi toolbox (Eerola and Toiviainen 2004). The main design objectives adopted for this implementation are as follows:

- A menu-driven Graphical User Interface (GUI).
- A versatile score definition data structure, allowing multiple scores to be analysed simultaneously, supported by flexible file storage, import and export options.
- A versatile score partitioning scheme, allowing multiple partitions of multiple scores to be analysed simultaneously.
- Both graphical and tabulated results interfaces for presenting the analysis results.
- Full implementation of the statistical analysis algorithms defined in chapter 3. These include the following:
  - Distribution of note durations.
  - Distribution of dyad note durations.
  - Distribution of interval sizes.
  - Distribution of absolute interval sizes.
  - Distribution of interval directions.
  - Distribution of pitch classes.
  - Distribution of dyad pitch classes.
- Full implementation of the Krumhansl and Schmuckler key-finding algorithm, with provision for the modifications introduced by Terhardt (1978), Huron and Parncutt (1993) and Temperley (1999).
- Implementation of a key profile estimation algorithm using pattern estimation.

The software application implements a considerable number of algorithms and each of these requires a number of interrelated input parameters. In order to achieve the key objective of designing a versatile, highly configurable application that requires minimum user interaction of a programming nature, it is essential to develop a well-designed hierarchy of analysis options to support the analysis methodologies in proper context.

From a software development perspective, the Matlab platform offers both advantages and disadvantages for the proposed software application. The advantages can be summarised as follows:

- The wide selection of software routines available through the various toolboxes facilitates rapid development for research purposes, using the relatively simple Matlab scripting language. The signal analysis, system identification and optimization toolboxes are particularly valuable for the project.
- The existing Midi toolbox is written in Matlab.
- Modern Matlab is object-orientated with support for GUI windows controls such as a menu system, text input fields, dropdown boxes, check boxes, etc.
- The platform has good graphics support through the object-orientated graph utilities.

The disadvantages can be summarised as follows:

- Compared to other software development platforms such as Delphi and C++, Matlab GUI support is rather rudimentary in the sense that a limited number of windows controls with limited functionality is available.
- The Integrated Development Environment (IDE) offers limited support for code debugging and optimization compared to the more formal development platforms.
- Matlab data storage is very loosely typed. This is a disadvantage from the perspectives of memory optimization, execution speed and error handling.

The Matlab platform was adopted despite the abovementioned disadvantages, mostly in view of the fact that the primary objective of the software implementation is the development of a research tool that can be easily extended over time to incorporate novel analysis algorithms.

The remainder of this chapter discusses the implementation details, including the program topology, navigation options, data structures and implementation of the various analysis algorithms.

4.2 Unified Modelling Language

Some of the software program features and implementation details will be described using Unified Modelling Language (UML) principles (Object Management Group, 2003), of which a structural overview is presented in Appendix H. UML represents a standardised methodology that can be used very effectively for describing the structural and behavioural characteristics of a software system, using well-defined graphical notation techniques to create visual models of the system. It is an international standard, i.e. ISO/IEC 19501, jointly supported by the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) and is maintained by the Objects Management Group (OMG). The current version, i.e. UML 2.3, was released in May 2010.

A full model using all of the above diagrams is beyond the scope of this document. UML is still an incomplete standard in the sense that it does not at present support the modelling of Graphics User Interfaces (GUIs) in a consistent manner (Ambler, 2004). Modified variations of the standard diagrams have been proposed for this purpose, including modified versions of the UML *Collaboration*/*Communication* diagram, UML *Activity* diagram and UML State Machine diagram. The software model presentation will focus on some aspects of the *Component Diagrams* to represent the structural characteristics and on *Activity Diagrams* to represent the functional properties.

4.3 Program structure

4.3.1 Overview

The program implementation is composed of a collection of functional units in the form of Matlab m-files. These are executed through the functionality offered by a set of linked, interactive, menu-driven Graphical User Interfaces (GUIs). The high-level functionality offered by the various GUIs can be categorised in five categories, namely program environmental options, score definition, score visualization and auditioning, statistical analysis and tonality analysis.

4.3.2 Program environmental options

The functional specifications of the software implementation gives rise to a relatively complex requirement in terms of the data structures for representation of the various parameters for maintaining the user environment, data input/output interfaces and analysis methodologies. The topologies associated with these data structures provide considerable insight into the system architecture from a structural and functional perspective and will therefore be presented in considerable detail in this section.

Fig. 4.1 shows the topology of the data structure associated with the main program environmental parameters and user options. The environmental parameters include two subcategories that define the default file path specifications and the attributes that control the formats of the graphical output objects respectively.

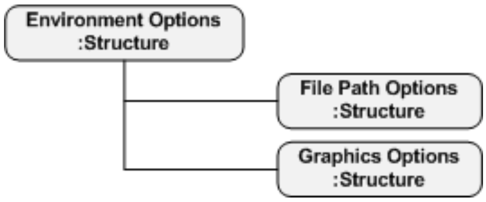


Fig. 4.1 High-level topology of the environmental parameters and user options data structure.

4.3.2.2 File path parameters and user options

Fig. 4.2 shows the topology of the data structures for storing default file path parameters and user options. The shaded elements identify user-defined attributes. The hierarchy features three default file paths, each defined in terms of directory and filename attributes, namely a *Suite* file path, *MIDI* file path and *Output* file path. The following details apply:

- *Suite file path*: The *Suite* file path variable stores the default directory opened by the file browser dialog window for the *Open Score Suite* and *Save As Score Suite* operations.
- *MIDI file path*: The *MIDI* file path variable stores the default directory opened by the file browser dialog window for the *Import Midi Sequence* and *Import Midi Sequence Batch* operations.
- *Output file path*: The *Output* file path variable stores the default directory opened by the file browser dialog window for all export operations, such as exporting analysis results.

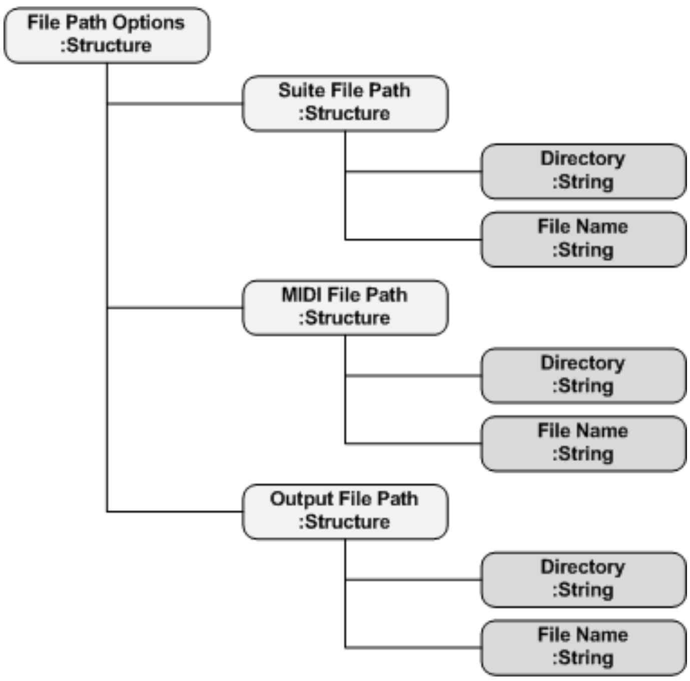


Fig. 4.2 Topology of the default file path parameters and user options data structure.

4.3.2.3 Graphical object parameters and user options

Fig. 4.3 shows the topology of the data structures for storing graphic object parameters and user options, where the shaded elements identify user-defined attributes.

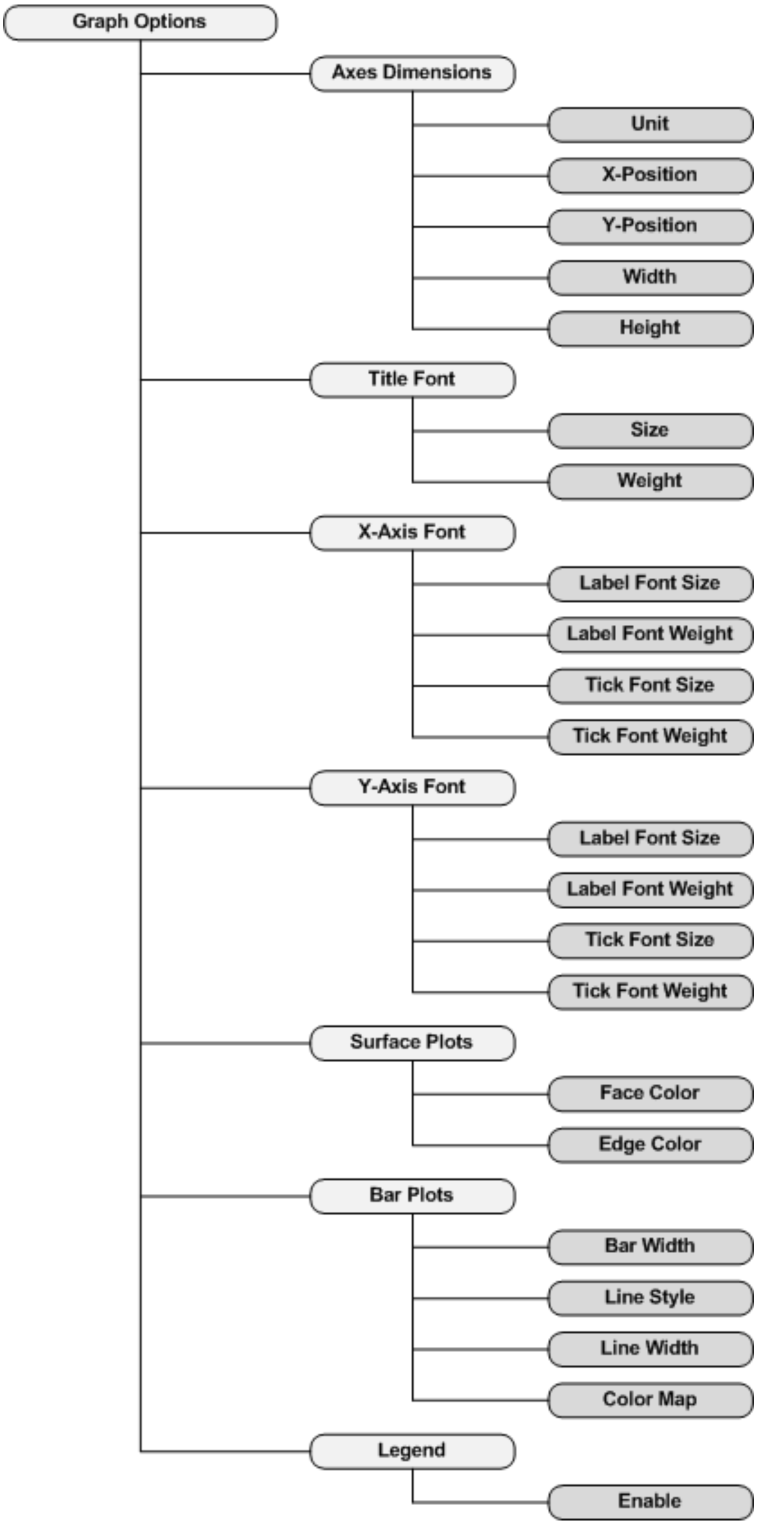


Fig. 4.3 Topology of the graphic objects parameters and user options data structure.

Many of the results generated by the software application are presented in the form of graphs, hence the importance of user-control over the attributes of the graphic objects. The user can control the dimensional properties of graphic axes objects, including the measurement unit, horizontal and vertical position in the parent object and the width and height dimensions. The data structure shows that the user can specify font attributes, including font sizes and font weights, for the title, axis labels and axis ticks of graphic axes objects. The colour schemes of surface and bar graph plots can be specified, as well as bar width, line width and line type of bar graphs. The rendering of legends can be enabled or disabled.

4.3.3 Score definition

The score data is contained in a versatile score structure which makes provision for information such as score annotations, MIDI sequence source file paths, MIDI sequence data, metadata such as reference and estimated key information, score selection flags used by the analysis routines, etc.

The implementation makes provision for the following operations to be performed on the score structure:

- *New score:* Create a new, empty score structure.
- *Close score:* The current score structure is closed.
- *Load score:* Load a score structure from a disk file. Scores are stored in the standard Matlab disk data storage format.
- *Save score:* The current score structure is saved to a disk file with the same filename.
- *Save score as:* The current score structure is saved to a disk file with a new filename.
- *Edit score:* Invoke a GUI for editing the information contained in the current score structure.

The graphical representations given in Fig. 4.4, Fig. 4.5 and Fig. 4.6, in combination, define the topology of the data structure for storing the score suite data. Fig. 4.4 shows the topology implemented for the main score suite data structure, which represents the high-level parameters associated with a score suite, with parameters as defined in Table 4.1. Fig. 4.5 shows the topology implemented for the score array data structure, which represents the parameters associated with an individual score, with parameters as defined in Table 4.2. Fig. 4.6 shows the data structure implemented for the MIDI sequence data structure, which represents the actual MIDI events pertaining to a particular score, with parameters as defined in Table 4.3.

This relatively complex data structure allows the user to construct a versatile data case for analysis in order to facilitate the diverse analysis options offered by the software implementation, particularly with reference to the following:

- Multiple scores can be defined and selected for individual analysis or analysis as a group. This is of particular importance for the key analysis operations, where the ability to conduct the analysis for a set of different scores is essential. For statistical analysis, it is also useful for obtaining the statistical properties for a set of scores as a whole or for comparing the individual results for multiple scores.
- The data structure makes provision for storing the full MIDI representation of the note events, as well as additional metadata such as key information, the monophonic/polyphonic properties of the sequence and the original midi file paths.
- The data structure allows user control of the notations associated with individual scores by editing metadata such as labels and descriptions. For research purposes, this is important for keeping track of the nature of the score data used in a particular analysis exercise.

Table 4.1 Parameter definitions for the score suite data structure shown in Fig. 4.4.

Category	**Subcategory/ Parameter**	**Description**
Suite identification	Label	User-defined name for the score suite.
	Description	User-defined description for the score suite.
Suite file path	Directory	Full directory path to score suite. Used for file operations such as *Suite Save*.
	File name	File name of the score suite. Used for file operations such as *Suite Save*.
Status	Suite status	System variable for indicating whether the suite is empty or holds valid suite information.
	Score status	System variable for indicating whether the score array is empty or holds valid score information.
Score index		System variable for identifying the currently selected score.
Score arrray		Array structure for storing score events

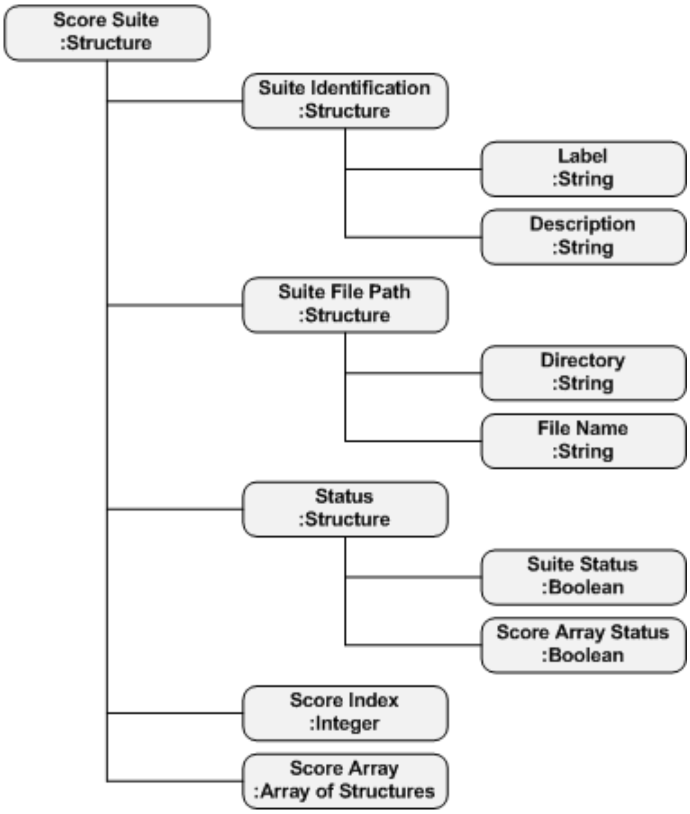


Fig. 4.4 Topology of the main score suite data structure.

Table 4.2 Parameter definitions for the score array data structure shown in Fig. 4.5.

Subcategory/ Parameter	**Subcategory/ Parameter**	**Description**
Score identification	Label	User-defined name for the score.
	Description	User-defined description for the score.
Score source	File path	File path of the imported MIDI sequence.
Midi sequence data		Data structure representing the MIDI data.
Sequence length		Length of the MIDI sequence.
Score key	Reference keys	Reference keys assigned key for the score.
	Estimated keys	Keys assigned by the key analysis algorithm for the score.
Score polyphony		Polyphonic/ monophonic status of the score.
Score status		Selected/deselected status of the score for analysis

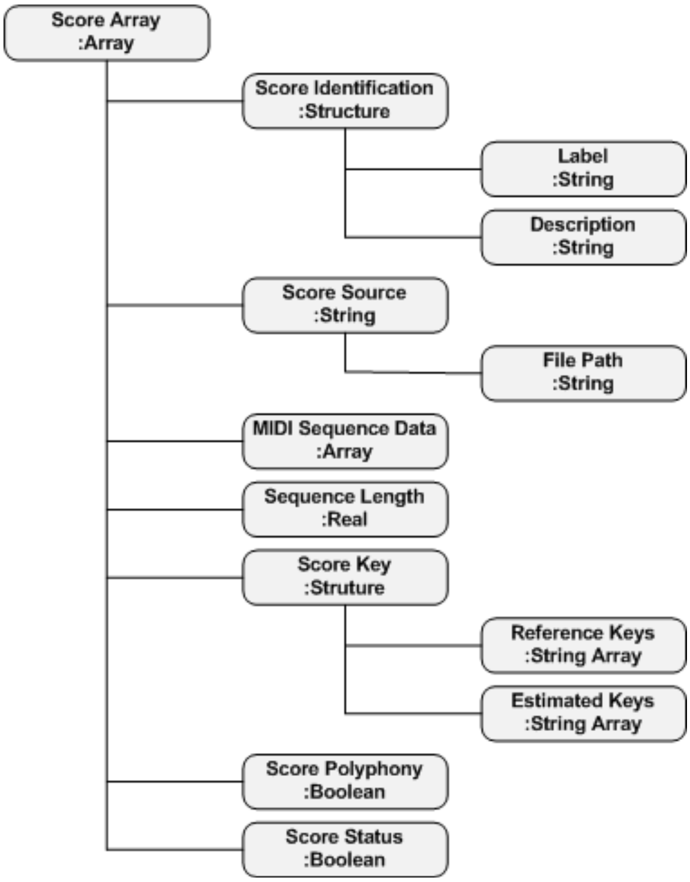


Fig. 4.5 Topology of the score array data structure.

Table 4.3 Parameter definitions for the MIDI sequence data structure shown in Fig. 4.6.

Parameter	**Description**
Timestamp [Beats]	Onset time of a note event in beats from the start of the sequence. Derived from the *Note On* event in the original MIDI sequence.
Duration [Beats]	Duration of a note event in beats. Derived from the *Note On* and *Note Off* events in the original MIDI sequence.
Midi channel	Midi channel associated with a note event.
Midi pitch	Midi pitch associated with a note event.
Midi velocity	Midi velocity associated with a note event.
Timestamp [seconds]	Onset of a note event in seconds from the start of the sequence. Derived from the *Note On* event in the original MIDI sequence.
Duration [seconds]	Duration of a note event in seconds. Derived from the *Note On* and *Note Off* events in the original MIDI sequence.

4.3.4 Score visualisation and auditioning

The application provides very basic functionality for visualising score material and auditioning score material. This functionality can be summarised as follows:

- *Score visualisation:*

  It is useful to view the sequence content of an individual score for diagnostic purposes without having to resort to an external score editing software application. The sequence data contained in the currently selected score can be viewed in pianoroll format.

- *Score auditioning:*

  The sequence data contained in the currently selected score can be played using the standard Windows media player application.

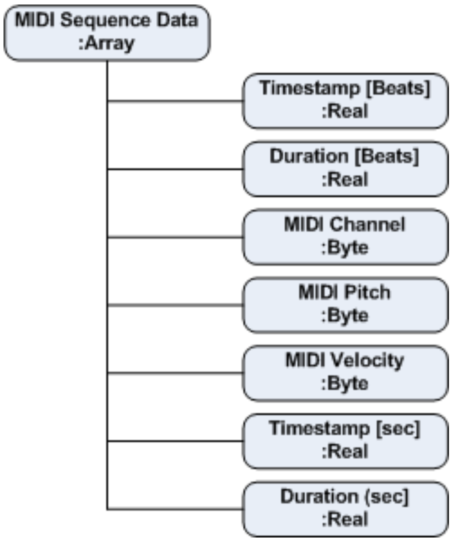


Fig. 4.6 Topology of the MIDI sequence data structure.

4.3.5 Statistical analysis

4.3.5.1 Overview

The statistical analysis functionality includes the algorithms introduced in section 3.2, i.e. various distributions of note durations, pitch intervals and pitch classes. Each of these requires a set of parameters that are implemented in an appropriate data structure to represent algorithm parameters and user options.

4.3.5.2 Distributions of note durations parameters and user options

Fig. 4.7 shows the data structure for the parameters and user options associated with the distributions of note durations algorithms. This same data structure applies for both the distribution of note durations as well as the distribution of dyad note durations. The shaded elements identify user-defined attributes. The individual parameters are defined in Table 4.4.

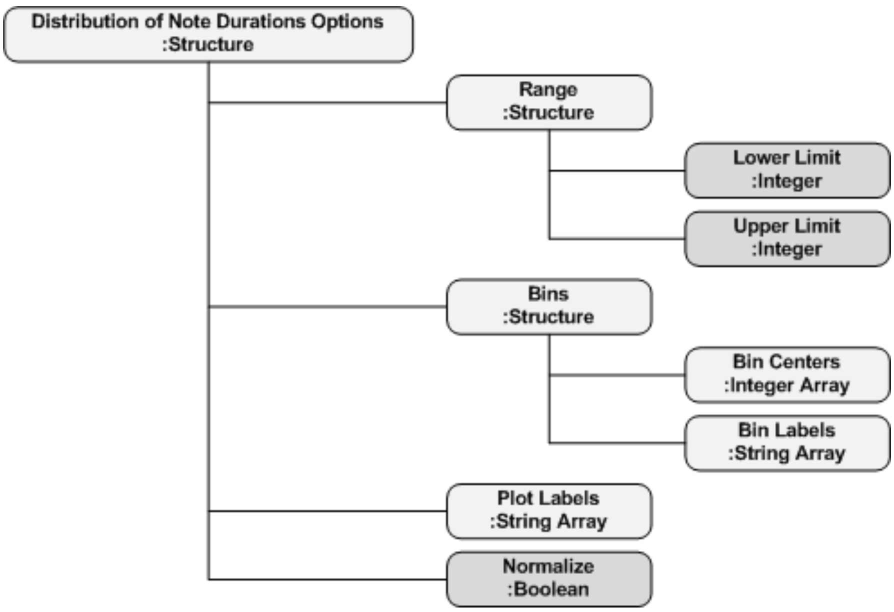


Fig. 4.7 Topology of the parameters and user options data structure for the distributions of note durations.

Table 4.4 Parameter definitions for the data structure shown in Fig. 4.7 for the distributions of note durations.

Category/ Parameter	Subcategory/ Parameter	Description
Range	Lower limit	Lower limit of the note duration bins used in the analysis.
	Upper limit	Upper limit of the note duration bins used in the analysis.
Bins	Centres	Histogram bin centres in the logarithmic domain as defined in Fig. 3.3.
	Labels	Bin labels used in the data export routines.
Plot labels		Duration labels used in the plot routines.
Normalise		Determines whether the distribution must be normalised.

4.3.5.3 Distributions of pitch intervals parameters and user options

Fig. 4.8 shows the data structure for the parameters and user options associated with the distributions of pitch intervals algorithms. This data structure also applies for the distributions of intervals, interval sizes, interval directions and dyad intervals. The shaded elements identify user-defined attributes. The individual parameters are defined in Table 4.5.

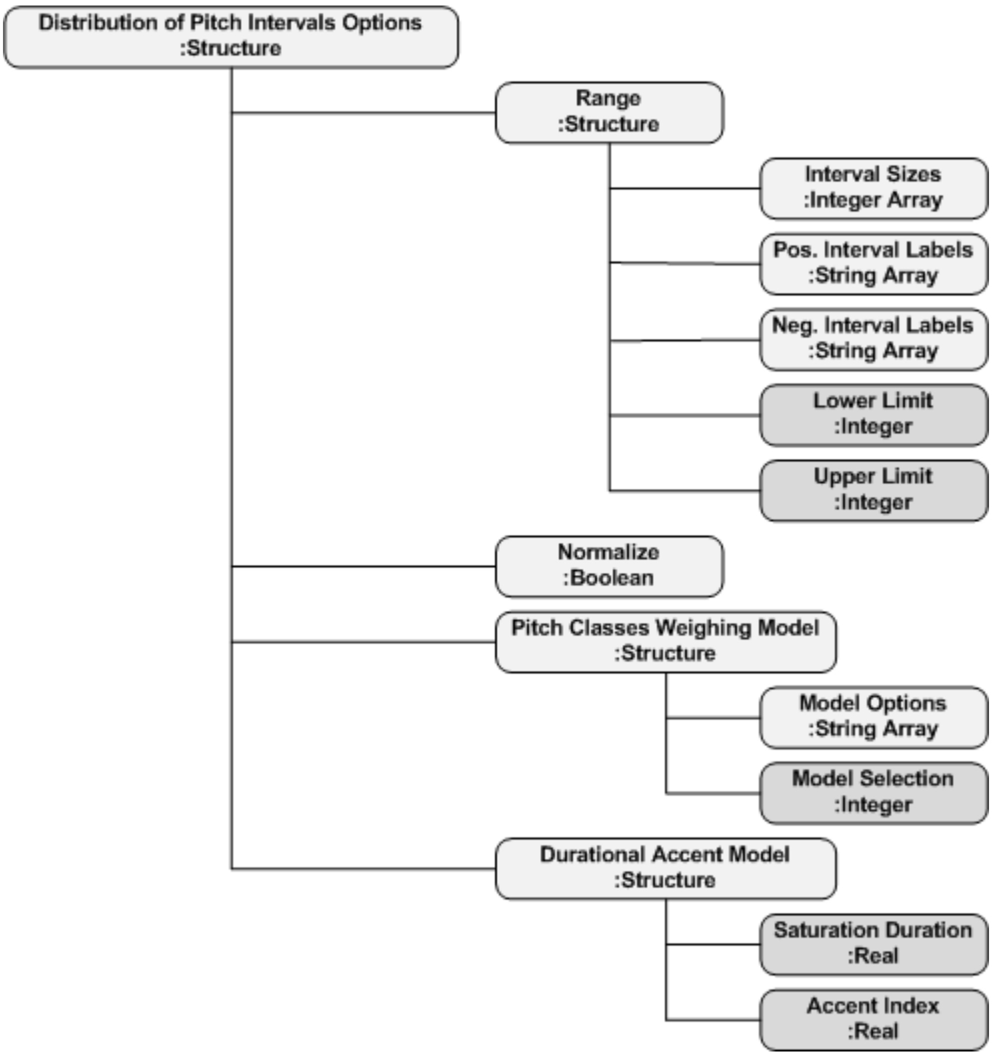


Fig. 4.8 Topology of the parameters and user options data structure for the distributions of pitch intervals.

Table 4.5 Parameter definitions for the data structure shown in Fig. 4.8 for the distributions of pitch intervals.

Category/ Parameter	**Subcategory/ Parameter**	**Description**
Range	Interval sizes	Array of valid interval sizes as defined in Table 3.2.
	Positive interval labels	Array of interval labels for the positive interval sizes.
	Negative interval labels	Array of interval labels for the negative interval sizes.
	Lower limit	Lower limit of the interval size used in the analysis.
	Upper limit	Upper limit of the interval size used in the analysis.
Normalise		Determines whether the distribution must be normalised.
Pitch class weighing model	Model options	Array of options for the pitch class weighing model: {Histogram weighing, Flat weighing, Linear durational weighing, Durational accent weighing}
	Model selection	Identifier of the selected pitch class weighing model.
Durational accent model	Tau	Echoic memory saturation duration (Parncutt, 1994).
	Index	Echoic memory accent Index (Parncutt, 1994).

4.3.5.4 Distributions of pitch classes parameters and user options

Fig. 4.9 shows the data structure for the parameters and user options associated with the distributions of pitch classes algorithms. The shaded elements identify user-defined attributes. The individual parameters are defined in Table 4.6.

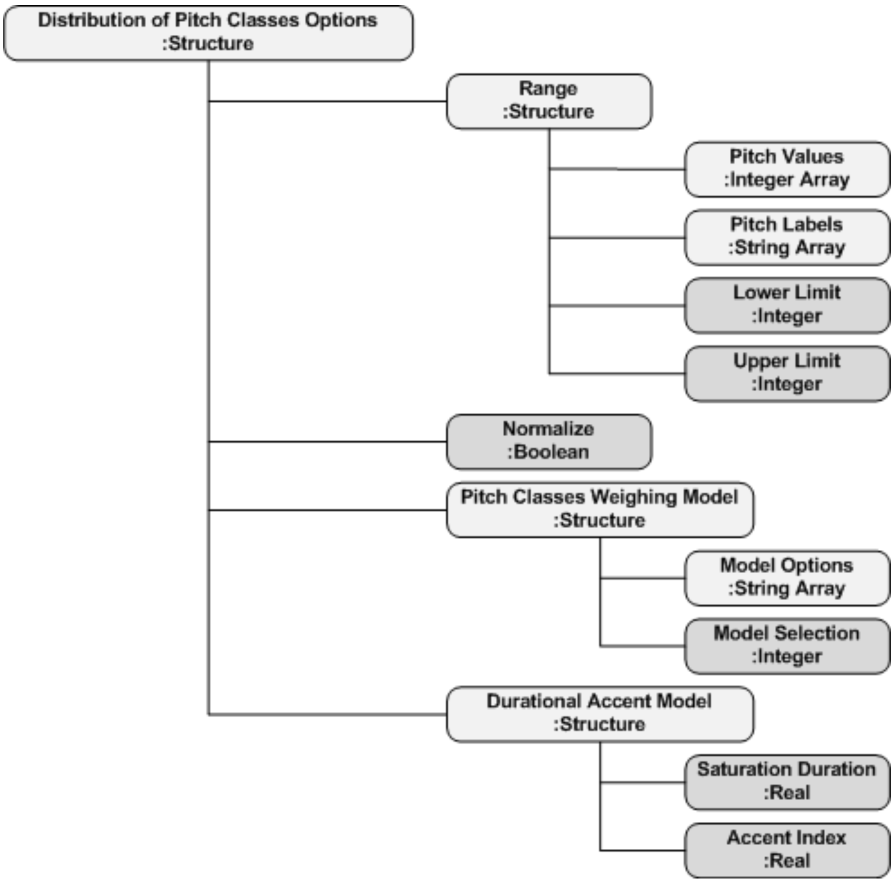


Fig. 4.9 Topology of the parameters and user options data structure for the distributions of pitch classes.

Table 4.6 Parameter definitions for the data structure shown in Fig. 4.9 for the distributions of pitch classes.

Category	**Subcategory/ Parameter**	**Description**
Range	Pitch values	Array of valid pitch values as defined in Table 3.3.
	Pitch labels	Array of pitch labels.
	Lower limit	Lower limit of the pitch value used in the analysis.
	Upper limit	Upper limit of the pitch value used in the analysis.
Normalise		Determines whether the distribution must be normalised.
Pitch class weighing model	Model options	Array of options for the pitch class weighing accent model: {Histogram weighing, Flat weighing, Linear durational weighing, Durational accent weighing}
	Model selection	Identifier of the selected pitch class weighing model.
Durational accent model	Tau	Echoic memory saturation duration (Parncutt,1994).
	Index	Echoic memory accent Index (Parncutt, 1994).

This same data structure applies for both the distribution of pitch classes as well as the dyad distribution of pitch classes.

4.3.6 Key-finding analysis

The software application makes allowance for a number of permutations of the original Krumhansl-Schmuckler key-finding algorithm. These include the following:

- Various weighing models in determining the relevant distribution of pitch classes are implemented, including histogram weighing, flat weighing as applied by Temperley (1999), linear durational accent as applied by Krumhansl and Kessler (1982) and durational accent weighing as applied by Huron and Parncutt (1993).
- Various key profile templates are implemented, including templates based on the pitch class hierarchies proposed by Krumhansl and Kessler (1982), Temperley (1999) and custom profiles entered by the user as part of the research exercise.
- Two different computational strategies for determining the correlation coefficients between a distribution of pitch class and a key profile template, namely the correlation formula used by Krumhansl (1990) and the modified formula used by Temperley (1999).

Fig. 4.10 shows the high level data structure for the parameters and user options associated with the key-finding algorithm. The hierarchy makes provision for two categories of parameters and user options, namely the distribution of pitch classes specifications and the key-finding algorithm specifications.

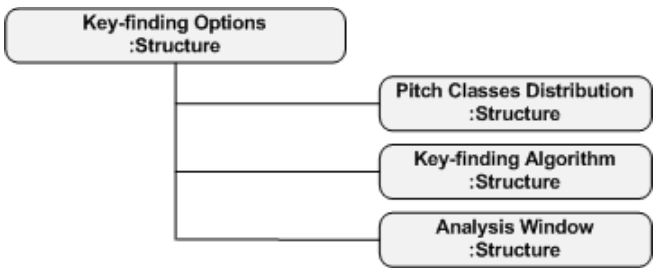


Fig. 4.10 High-level topology of the key-finding algorithm parameters and user options data structure.

Fig. 4.11 shows the data structure for the parameters and user options associated with the distribution of pitch classes applied in the key-finding algorithm. The shaded elements identify user-defined attributes. This structure is identical to the structure shown in Fig. 4.9 for the statistical analysis of pitch classes. The individual parameters are defined in Table 4.7.

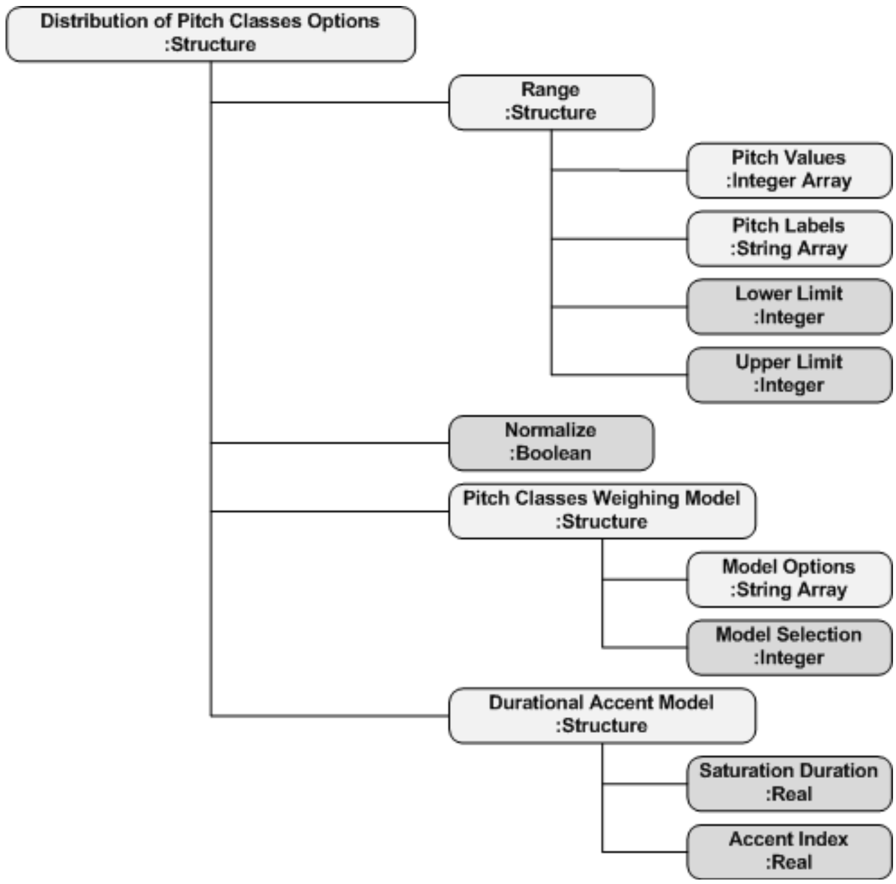


Fig. 4.11 Topology of the parameters and user options data structure for the distribution of pitch classes applied in the key-finding algorithm.

Table 4.7 Parameter definitions for the data structure shown in Fig. 4.11 for the distribution of pitch classes applied in the key-finding algorithm.

Category	**Subcategory/ Parameter**	**Description**
Range	Pitch values	Array of valid pitch values as defined in Table 3.3.
	Pitch labels	Array of pitch labels.
	Lower limit	Lower limit of the pitch value used in the analysis.
	Upper limit	Upper limit of the pitch value used in the analysis.
Normalise		Determines whether the distribution must be normalised.
Pitch class weighing model	Model options	Array of options for the pitch class weighing accent model: {Histogram weighing, Flat weighing, Linear durational weighing, Durational accent weighing}
	Model selection	Identifier of the selected durational accent model.
Durational accent model	Tau	Echoic memory saturation duration (Parncutt, 1994).
	Index	Echoic memory accent Index (Parncutt, 1994).

Fig. 4.12 shows the data structure for the parameters and user options associated with the key-finding algorithm. The shaded elements identify user-defined attributes. The individual parameters are defined in Table 4.8.

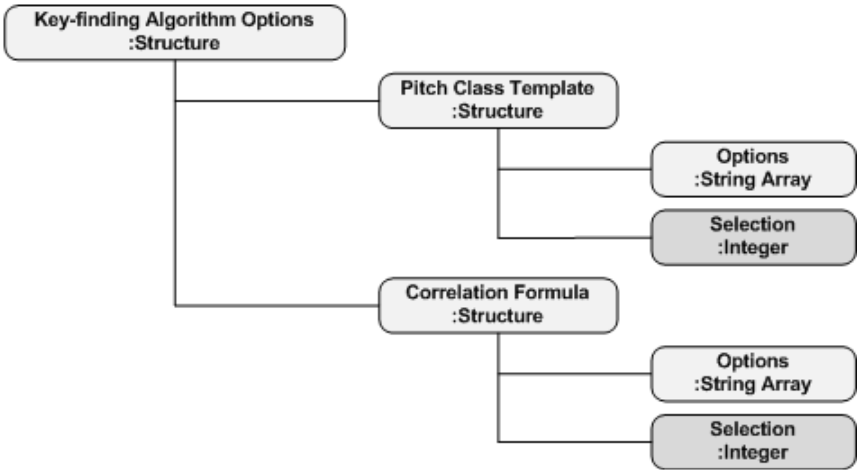


Fig. 4.12 Topology of the parameters and user options data structure for the key-finding algorithm.

Table 4.8 Parameter definitions for the data structure shown in Fig. 4.12 for the key-finding algorithm.

Category	**Subcategory/ Parameter**	**Description**
Key profile template	Options	Array of pitch class template options: {Krumhansl/Kessler, Temperley, Custom}
	Option selection	Identifier of the selected pitch class template.
Correlation formula	Options	Array of correlation formula options: {Krumhansl/Kessler, Temperley}.
	Option selection	Identifier of the selected correlation formula.

4.4 Implementation of the score selection and partitioning strategy

The application implements a versatile methodology for targeting the sequence data contained in the score suite data structure for analysis. The functionality with regards to selecting the target material for analysis can be summarised as follows:

- Multiple scores can be targeted for analysis simultaneously. Scores are selected for analysis by means of a multi-select list box which set the *Score status* flag featured in Fig. 4.4 and Table 4.1 for the selected score. The results can be presented for the individual scores or cumulatively for the entire body of material targeted in the analysis.
- Multiple partitions of the selected scores can be targeted for analysis simultaneously. The partitions are defined through application of a set of window parameters. The results can be presented for the individual partitions or cumulatively for all partitions.

The application makes provision for the following partitioning options:

- *No partitioning:*

  The entire score is targeted.
- *Single window partition:*

  Only the score events located inside the temporal window defined by W_{start} to $W_{start} + W_{size}$ as shown in Fig. 4.13 are targeted in the analysis.
- *Sliding window partitions:*

  The score events located inside the temporal partitions defined by $W_{start} + iW_{increment}$ to $W_{start} + iW_{increment} + W_{size}$, with $0 \le i \le N$ as shown in Fig. 4.14, are targeted individually for each partition in the analysis.

- *Expanding window partitions:*

  The score events located inside the temporal partitions defined by W_{start} to $W_{start} + W_{size} + (i+1)W_{increment}$, with $0 \le i \le N$ as shown in Fig. 4.15, are targeted individually for each partition in the analysis.

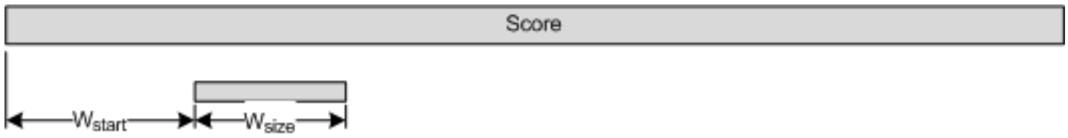


Fig. 4.13 Score partitioning using the single window approach.

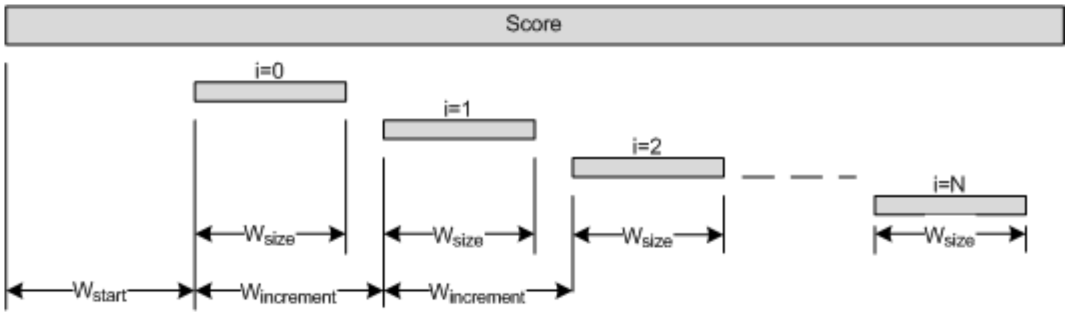


Fig. 4.14 Score partitioning using the sliding window approach.

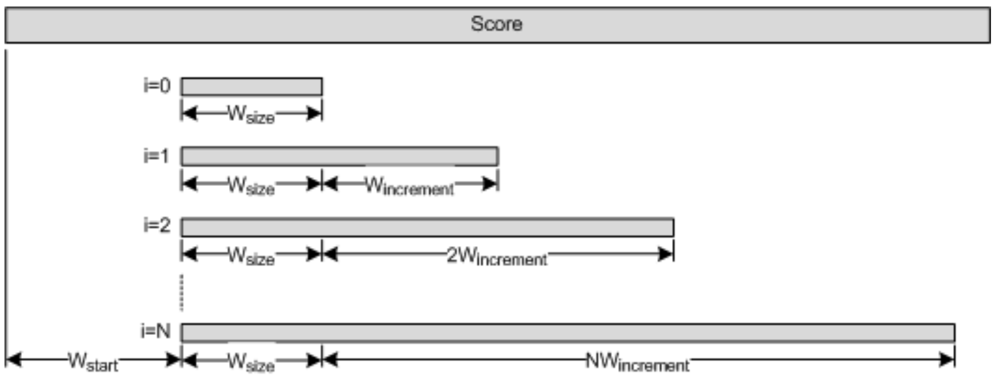


Fig. 4.15 Score partitioning using the expanding window approach.

Three options are implemented for the unit definitions of the window parameters, i.e. W_{start}, W_{size} and $W_{increment}$, namely beats, seconds and number of notes. The partitioning window parameters can be defined in terms of measures by selecting appropriate settings in beats based on the applicable time signature. Fig. 4.16 and Table 4.9 summarise the user options and data structure implemented for the score partitioning scheme.

The score selection and partitioning options implemented in the application deliver a very versatile scheme for applying the statistical and tonality analysis algorithms. This is particularly important for the key-finding and key profile estimation case studies conducted in the investigation. These include the analysis of multiple scores simultaneously, as for the test case of the 24 preludes of Bach's *Well-tempered Clavier*, and the analysis of multiple measures simultaneously for tracking key modulations on a measure by measure basis, as for the test cases of the *Courante* from Bach's *Cello Suite in C* and the *Gavotte* from Bach's *French Suite No. 5 in G major (BWV 816)*.

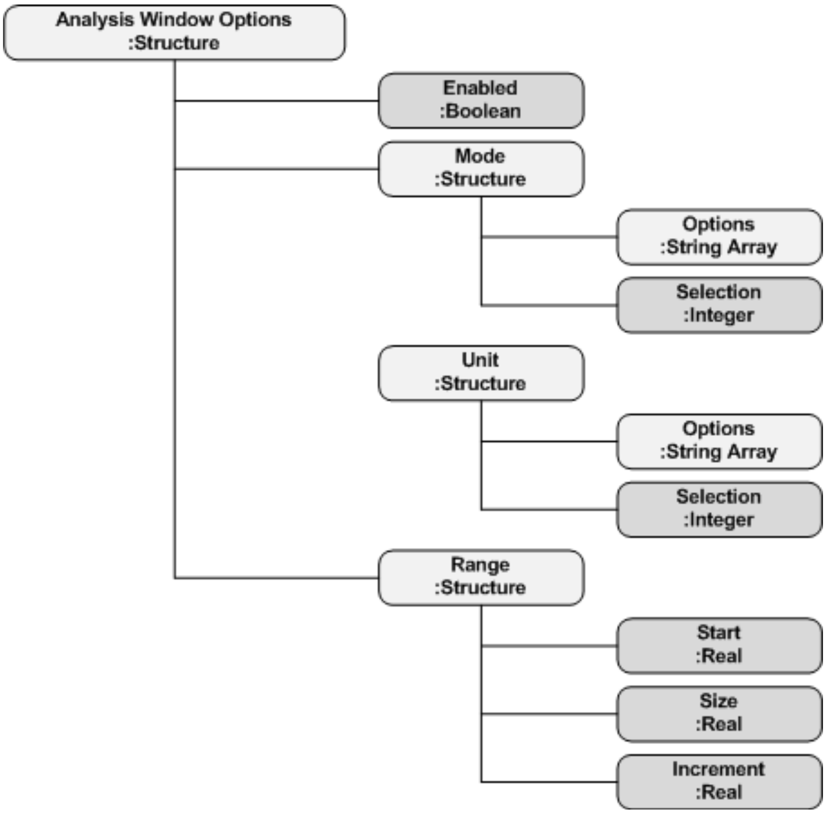


Fig. 4.16 Topology of the parameters and user options data structure for the partitioning window.

Table 4.9 Parameter definitions for the data structure shown in Fig. 4.16 for the partitioning window.

Category	**Subcategory/ Parameter**	**Description**
Enabled		Determines whether the window must be applied. If disabled, the complete MIDI sequence is targeted.
Mode	Mode options	Array of options for the partitioning window mode: {Single, Sliding, Expanding}
	Mode selection	Identifier of the selected mode.
Unit	Unit options	Array of options for the partitioning window unit: {Beats, Seconds, Number of notes}
	Unit selection	Identifier of the selected unit.
Range	Start	Start of the partitioning window.
	Size	Size of the partitioning window.
	Increment	Increment of the start value for the sliding mode window or increment of the window size for the expanding mode window.

4.5 Implementation of the direct search key profile estimation routine

4.5.1 Overview

The final research objective of this investigation involves determining whether it is possible to find a more optimal key profile for the Krumhansl and Schmuckler key-finding algorithm using optimization methods. Therefore, a key profile estimation algorithm was implemented using the direct search methods available through the Matlab optimization toolbox. The implementation was designed to accommodate the following requirements, all of which arise from the nature of the research questions associated with the problem:

- The key profile estimation routine should make full use of the score selection and score partitioning functionality described earlier. This is essential because the selection of the sample material for the optimization process play a major role in the overall success of determining a more optimum key profile.

- The key profile estimation routine should make use of the parameter definition structures already implemented for the Krumhansl and Schmuckler algorithm, such as the selection of the pitch classes weighing methodology, pitch classes normalization setting, key correlation formula, etc. This will ensure maximum versatility in searching for the most optimal combination of key-finding parameters and key profiles.
- The key profile estimation routine should allow for implementation of different cost functions. The mathematical formulation of the cost function represents one of the key research questions, so that flexibility is required in testing different cost functions during the optimization process.
- The key profile estimation routine should also allow for specifying the different parameter options that are associated with the Matlab direct search methods, including parameter constraints, search methods, mesh specifications, polling options, search tolerance, termination criteria, etc. This is important because no *a priory* information currently exists in literature on how the cost functions are likely to behave. As a result, an extensive investigation will have to be conducted to find the most appropriate approach and optimization settings for this type of problem.

4.5.2 Algorithm topology

Fig. 4.17 shows a flow diagram of the key profile estimation algorithm as implemented. The shaded blocks refer to functionality implemented by the direct search algorithm. The algorithm can be summarised as follows:

- A partitioning structure is defined based on the selected score material and the score partitioning parameters defined by the window options structure.
- The score material associated with the current score and partitioning window is extracted and the Krumhansl and Schmuckler key-finding algorithm is applied, taking cognizance of the parameters selected for the distribution of pitch classes, correlation formula, etc.
- The cost function is updated.
- The above two steps are repeated until all partitions for all scores have been processed.
- The direct search algorithm evaluates whether the termination conditions have been satisfied. If so, the routine returns to the calling context for data display and exchange, otherwise the key profile is amended and the cost evaluation is repeated.

The key estimation algorithm returns a significant amount of data and results. These include the partitioning information, distribution of pitch classes for the final key profile, key correlation data for the final key profile, reference key information, estimated key information for all scores and partitions for the final key profile, etc.

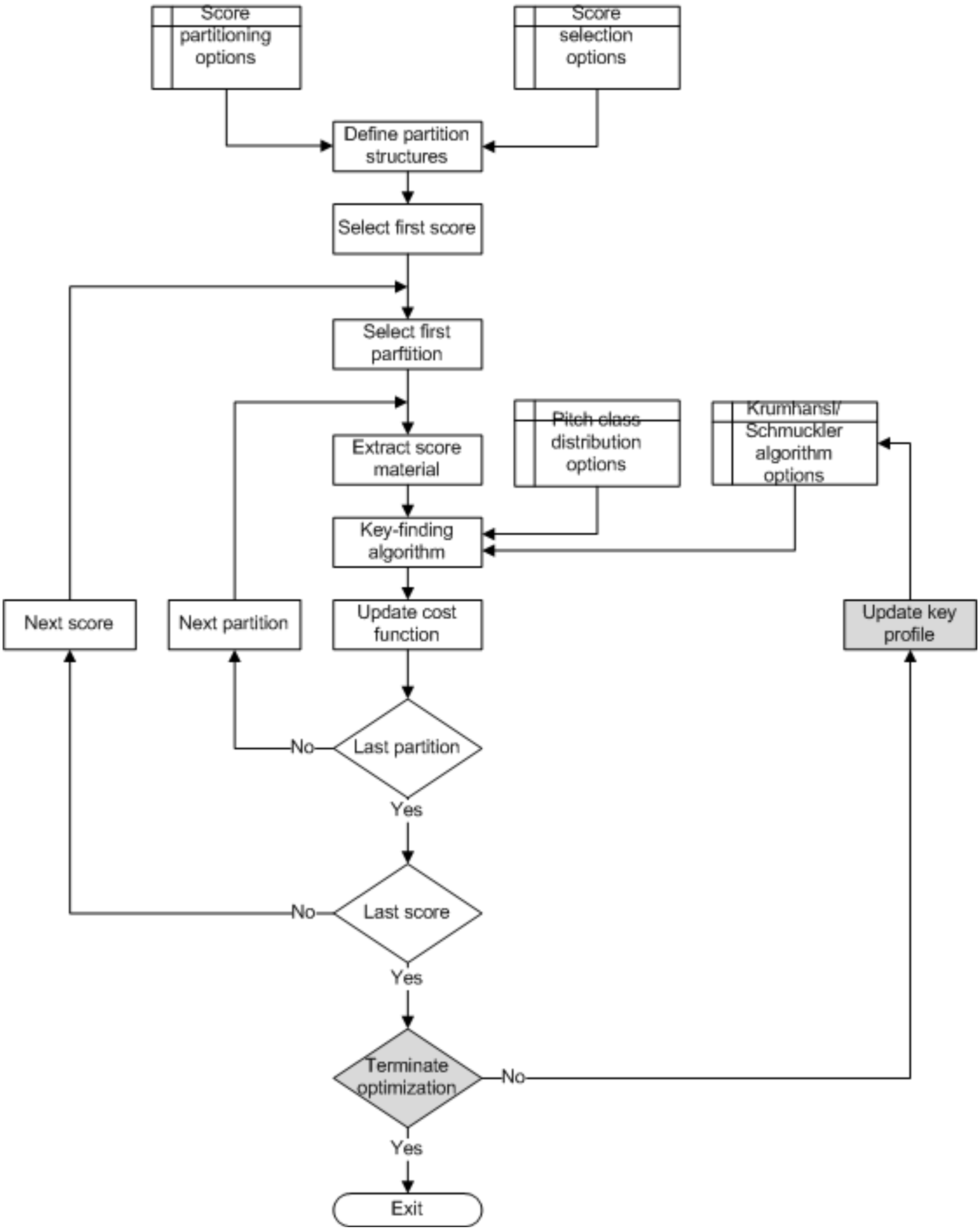


Fig. 4.17 Flow diagram of the key profile estimation algorithm.

4.5.3 Cost function definitions

4.5.3.1 Introduction

The cost function definition is the most important factor in determining the overall success of an optimization exercise. The current implementation makes provision for three different cost functions, which have been designed with the view to evaluate the effects of different approaches with regards to the following:

- The numeral calculation of the key-finding score for a particular partition.

- The effect of including a temporal dimension to the key-finding score by determining the score based on the joint performance for successive partitions rather than for a single partition in isolation. This introduces the penalty principle applied by Temperley (1999) in the sense that a penalty is introduced if the key-finding performance score for successive partitions is not optimal.

The three cost functions are designated as *A*, *B* and *C* in the remainder of this document. A number of variations of these cost functions have also been implemented and tested, including functions that incorporate scaling of the key correlation contributions with the mean values of the key correlations vectors, etc. Due to space considerations, these will not be considered here or in presenting the final case study results.

4.5.3.2 Cost function A

Cost function *A* optimises the cumulative number of key matches, with the key matches determined for the individual partitions in isolation. The function determines the cost C using the relationships

$$C = \frac{1}{\sum_{i=1}^{N}\sum_{j=1}^{N_i} S_{ijk}} \tag{4.1}$$

with

$$S_{ijk} = \begin{cases} \sum_{k=1}^{N_{ij}^E} \frac{1}{N_{ij}^E} & k_{ijk}^E \in K_{ij}^E ; k_{ijk}^E \in K_{ij}^R \\ 0 & k_{ijk}^E \in K_{ij}^E ; k_{ijk}^E \notin K_{ij}^R \end{cases} \tag{4.2}$$

where

i denotes the i^{th} score and N denotes the number of scores,

j denotes the j^{th} partition and N_i denotes the number of partitions of the i^{th} score,

k denotes the k^{th} estimated key and N_{ij}^E denotes the total number of estimated keys returned for the j^{th} partition of the i^{th} score

and

K_{ij}^R and K_{ij}^E denote the set of reference keys and estimated keys respectively for the j^{th} partition of the i^{th} score.

This cost function is highly discontinuous in the sense that it responds in discrete steps defined by the value of $\frac{1}{N_{ij}^E}$.

4.5.3.3 Cost function B

Cost function *B* optimises the cumulative normalised key correlation value for all key matches, with the key matches and key correlations determined for the individual partitions in isolation. The normalization is performed by dividing the key correlation value by the number of keys estimated for the partition. Thus, while the function value reflects the number of matches between the reference and estimated key sets, it also reflects the cumulative key correlation values associated with these matches. The function determines the cost C using the relationships

$$C = \frac{1}{\sum_{i=1}^{N}\sum_{j=1}^{N_i} S_{ijk}} \tag{4.3}$$

with

$$S_{ijk} = \begin{cases} \sum_{k=1}^{N_{ij}^E} \frac{R_{ijk}}{N_{ij}^E} & k_{ijk}^E \in K_{ij}^E ; k_{ijk}^E \in K_{ij}^R \\ 0 & k_{ijk}^E \in K_{ij}^E ; k_{ijk}^E \notin K_{ij}^R \end{cases} \tag{4.4}$$

where R_{ijk} denotes the key correlation coefficient for the k^{th} estimated key returned for the j^{th} partition of the i^{th} score. The rest of the parameters are as defined in section 4.5.3.2.

This cost function is less discontinuous compared to cost function A in the sense that for each discrete contribution made a key match, the function reflects a continuous band of function values due to the optimization of the key correlation contribution.

4.5.3.4 Cost function C

Cost function C optimises the cumulative number of key matches as well as the cumulative key correlation value, with the key matches and key correlations determined cumulatively for two successive partitions. Thus, while the function value reflects the number of matches between the reference and estimated key sets for the successive partitions, it also reflects the cumulative key correlation values associated with these matches. The function determines the score C of the key-finding result using the relationships

$$C = \frac{1}{\sum_{i=1}^{N}\sum_{j=1}^{N_i - 1} S_k} \tag{4.5}$$

where

$$S_j = \begin{cases} \sum_{k_1=1}^{N_{ij}} \sum_{k_2=1}^{N_{i(j+1)}} \left(\frac{R_{ijk_1}^E + R_{i(j+1)k_2}^E}{N_{ij}^E \; N_{i(j+1)}^E} \right) & k_{ijk_1}^E \in K_{ij}^E ; k_{ijk_2}^E \in K_{i(j+1)}^R \\ 0 & k_{ijk_1}^E \in K_{ij}^E ; k_{ijk_2}^E \notin K_{i(j+1)}^R \end{cases} \tag{4.6}$$

where R_{ijk_1} denotes the key correlation coefficient for the k_1^{th} estimated key returned for the j^{th} partition of the i^{th} score and R_{ijk_2} denotes the key correlation coefficient for the k_2^{th} estimated key returned for the $j+1^{th}$ partition of the i^{th} score. The rest of the parameters are as defined in section 4.5.3.2.

Similarly to cost function *B*, this cost function is less discontinuous compared to cost function *A* in the sense that for each discrete contribution made a key pair match from successive partitions, the function reflects a continuous band of function values due to the optimization of the key correlation contribution. The function can be expanded to the general case where any number of successive partitions are scored together.

4.6 Program validation

4.6.1 Overview

The mathematical algorithms presented in chapter 3 and the Matlab implementations thereof were subjected to a rigorous validation process. In the case of the statistical analysis algorithms, this was achieved by designing dedicated test scores that facilitate manual calculations with which to verify the results produced by the software implementation. In the case of the key-finding algorithms, verification by comparison with manual calculations is not feasible. This is due to the complexity of the algorithms and the requirement to also verify correct interpretation of the algorithms against implementations used in the research sources. Validation in this case relies on comparisons between the results produced by the software implementation and results published in literature for appropriate case studies.

The results presented in this chapter also serve to demonstrate the nature of the analysis outputs generated by the application, particularly with regards to the graphical presentations.

4.6.2 Statistical analysis of note durations

4.6.2.1 Distribution of note durations

The algorithms as implemented in Matlab for the distribution of note durations were verified using the test score shown in Fig. 4.18. This test score contains all of the note durations listed in (3.4) monotonically decreasing order.

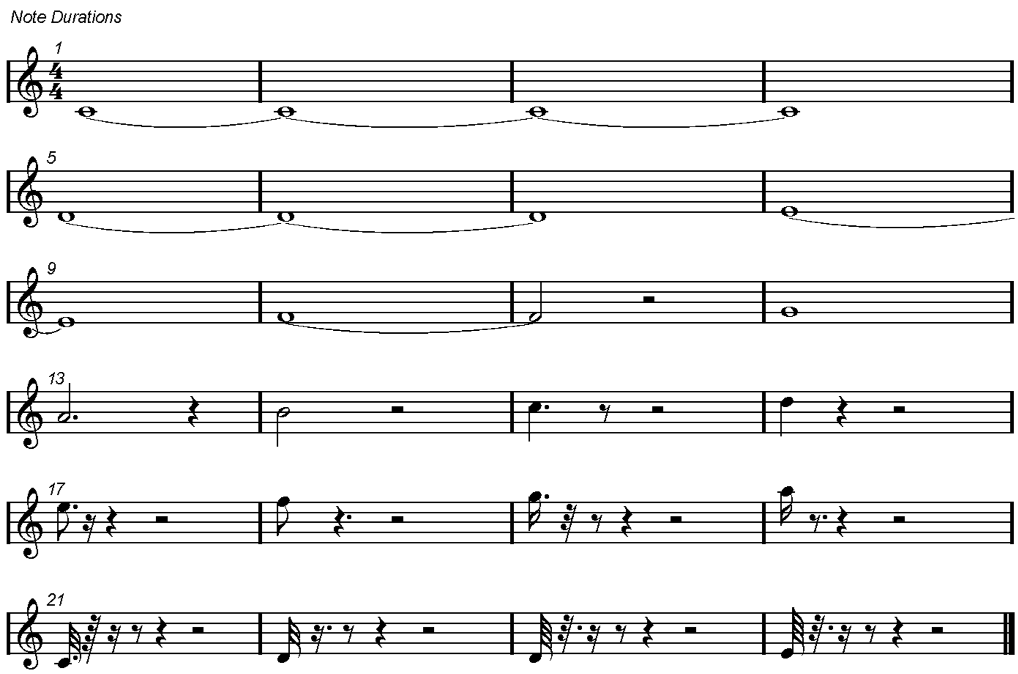


Fig. 4.18 Test score for the statistical analysis of note durations.

Fig. 4.19 shows the distribution of note durations for the test score. The distribution correctly reflects a count of one for all bins.

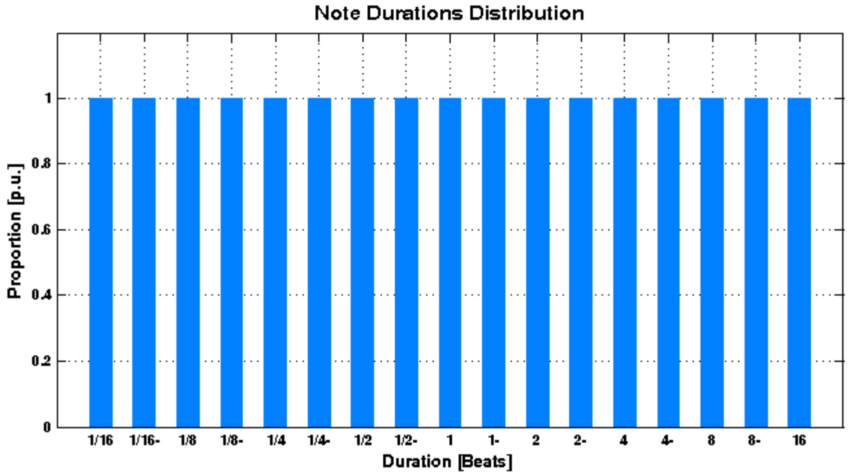


Fig. 4.19 Distribution of note durations for the test score shown in Fig. 4.18.

Fig. 4.20 shows the normalised distribution of note durations for the test score. The distribution correctly reflects a p.u. count of

$$h = \frac{1}{(17)(1)} = 0.0588 \tag{4.7}$$

for all bins.

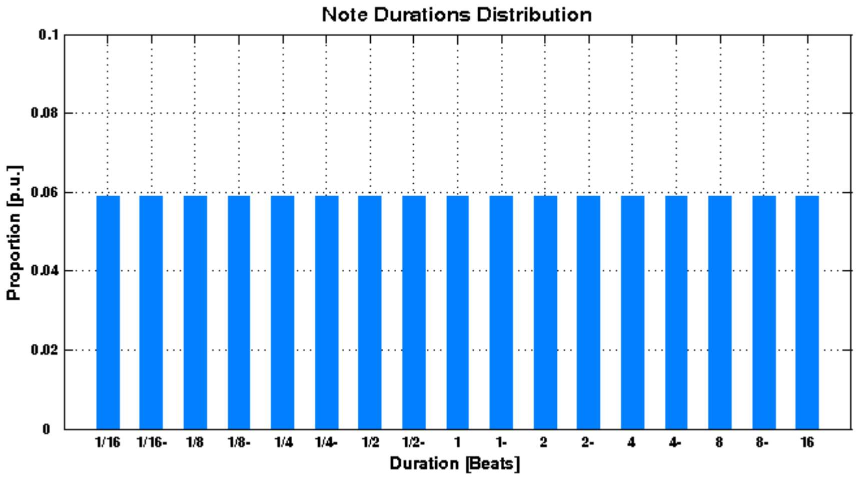


Fig. 4.20 Normalised distribution of note durations for the test score shown in Fig. 4.18.

4.6.2.2 Distribution of dyad note durations

The algorithms as implemented in Matlab for the distribution of dyad note durations were verified using the test score shown in Fig. 4.18. The test score contains all of the note durations listed in (3.4) monotonically decreasing order and gives rise to the following set of successive dyad durations:

$$\left\{ \begin{array}{l} \{16,12\},\{12,8\},\{8,6\},\{6,4\},\{4,3\},\{3,2\},\left\{2,\tfrac{3}{2}\right\},\left\{\tfrac{3}{2},1\right\},\left\{1,\tfrac{3}{4}\right\},\left\{\tfrac{3}{4},\tfrac{1}{2}\right\},\left\{\tfrac{1}{2},\tfrac{3}{8}\right\},\left\{\tfrac{3}{8},\tfrac{1}{4}\right\},\left\{\tfrac{1}{4},\tfrac{3}{16}\right\},\left\{\tfrac{3}{16},\tfrac{1}{8}\right\}, \\ \left\{\tfrac{1}{8},\tfrac{3}{32}\right\},\left\{\tfrac{3}{32},\tfrac{1}{16}\right\} \end{array} \right\} \tag{4.8}$$

Fig. 4.21 shows the distribution of dyad note durations for the test score. The distribution correctly reflects a count of one for the duration combinations listed in (4.8) with counts of zero for all other combinations.

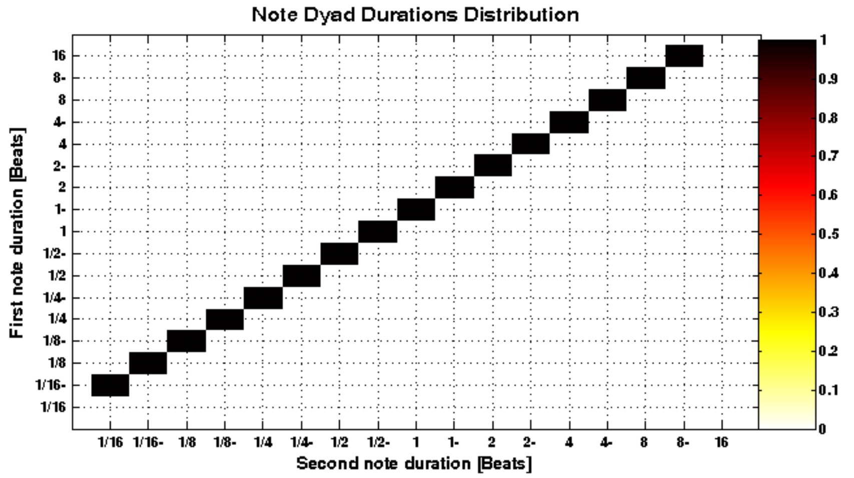


Fig. 4.21 Distribution of dyad note durations for the test score shown in Fig. 4.18.

Fig. 4.22 shows the normalised distribution of dyad note durations for the test score. The distribution correctly reflects a p.u. count of

$$\frac{(1)}{(17)(1)} = 0.0625 \tag{4.9}$$

for the duration combinations listed in (4.8) with counts of zero for all other combinations.

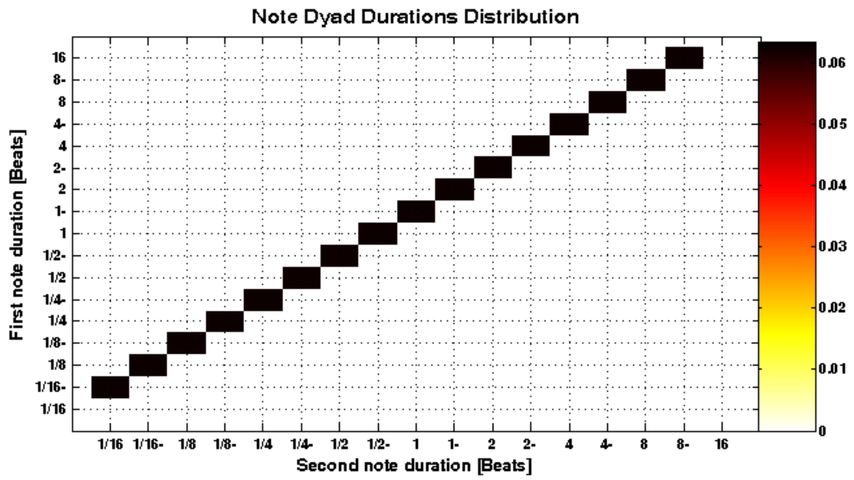


Fig. 4.22 Normalised distribution of dyad note durations for the test score shown in Fig. 4.18.

4.6.3 Statistical analysis of pitch intervals

4.6.3.1 Distribution of pitch intervals

The algorithms as implemented in Matlab for the distributions of pitch intervals were verified using the test score shown in Fig. 4.23, which contains all the pitch intervals listed in (3.23).

Fig. 4.23 Test score for statistical analysis of pitch intervals.

Fig. 4.24 shows the unweighted distribution of pitch intervals for the test score. The distribution correctly reflects a count of one for all intervals.

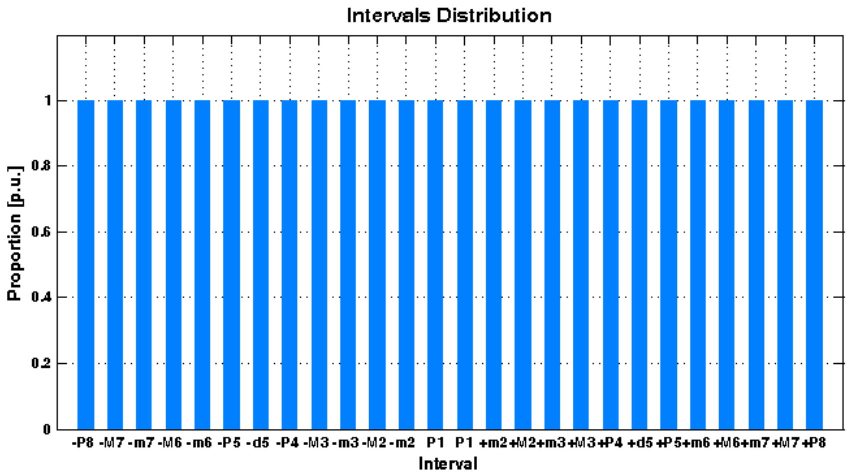


Fig. 4.24 Unweighted distribution of pitch intervals for the test score shown in Fig. 4.23.

Fig. 4.25 shows the normalised unweighted distribution of pitch intervals for the test score. The distribution correctly, for all intervals, shows a p.u. count of

$$h_k = \frac{1}{(26)(1)} = 0.385. \tag{4.10}$$

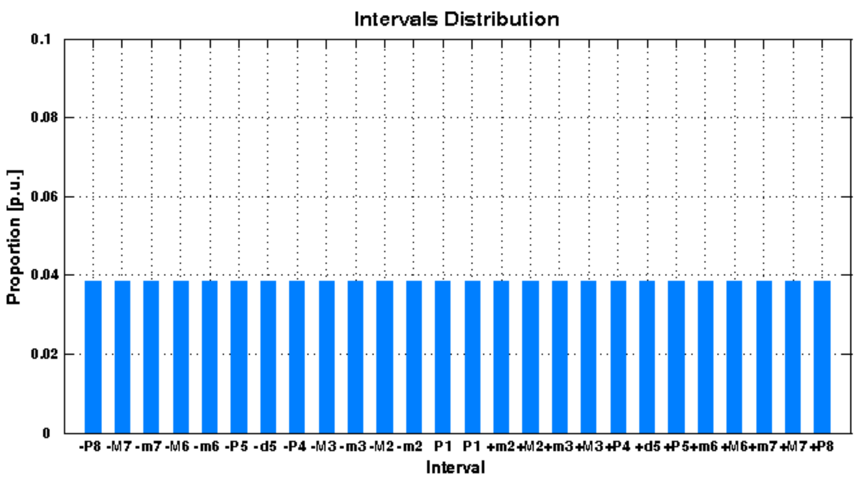


Fig. 4.25 Normalised unweighted distribution of pitch intervals for the test score shown in Fig. 4.23.

Fig. 4.26 shows the weighted distribution of pitch intervals for the test score. The quarter notes in the test score, using a tempo of 120 Beats Per Minute (BPM), translates to durations of 0.5 s. With the recommended default values, i.e. saturation duration of 0.5 and accent index of 2, the durational accent of a quarter note is given by the relationship (Parncutt, 1994)

$$Durational\ Accent = \left(1 - e^{-\frac{0.5}{0.5}}\right)^2 = 0.3996. \tag{4.11}$$

This yields the following result for the weighted duration of two successive quarter notes:

$$Durational\ Weight = 0.3996 + 0.3996 = 0.7992 \tag{4.12}$$

The distribution shown in Fig. 4.26 correctly reflects a cumulative weighted duration of 0.7992 for all intervals.

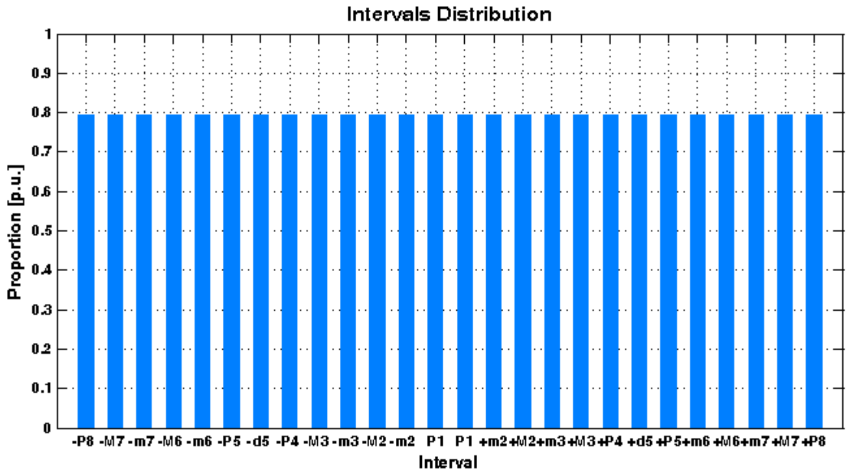


Fig. 4.26 Weighted distribution of pitch intervals for the test score shown in Fig. 4.23.

Fig. 4.27 shows the normalised weighted distribution of pitch intervals for the test score. The distribution correctly, for all intervals, reflects a p.u. cumulative weighted duration of

$$h_k = \frac{0.7992}{(26)(0.7992)} = 0.0385 \tag{4.13}$$

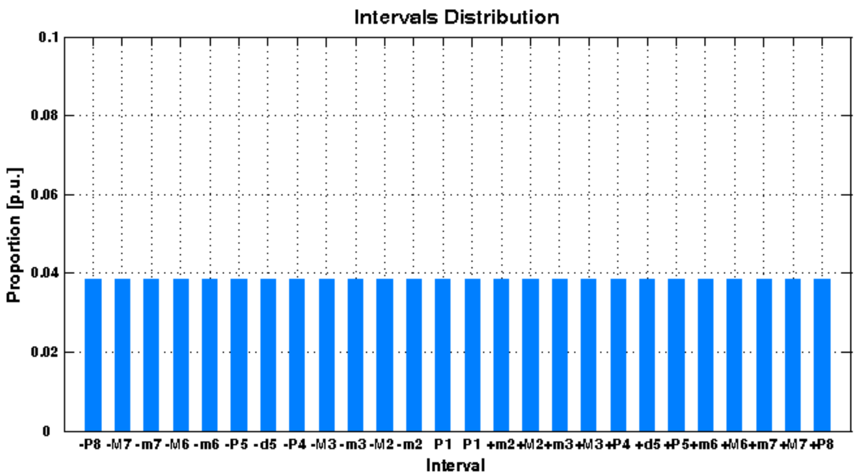


Fig. 4.27 Normalised weighted distribution of pitch intervals for the test score shown in Fig. 4.23.

4.6.3.2 Distribution of pitch interval sizes

The algorithms as implemented in Matlab for the distributions of pitch interval sizes were verified using the test score shown in Fig. 4.23. This test score contains all of the pitch interval sizes listed in (3.33) twice.

Fig. 4.28 shows the unweighted distribution of pitch interval sizes for the test score. The distribution correctly reflects a count of two for all pitch interval sizes.

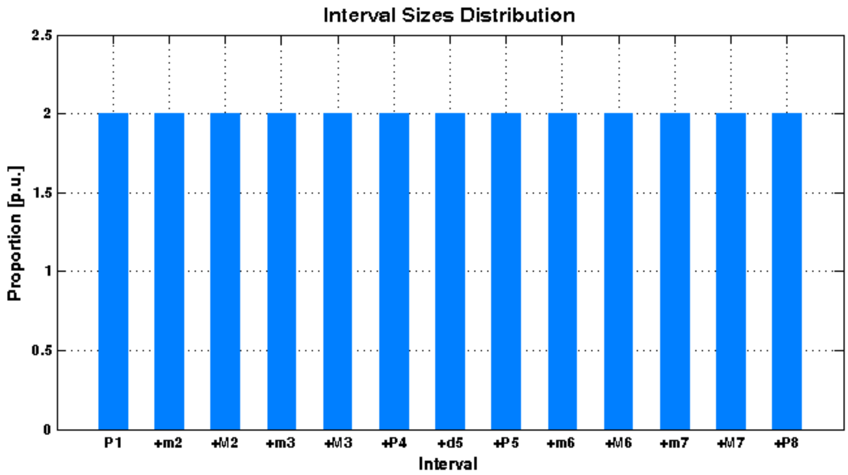


Fig. 4.28 Unweighted distribution of pitch interval sizes for the test score shown in Fig. 4.23.

Fig. 4.29 shows normalised unweighted distribution of pitch interval sizes for the test score. The distribution correctly shows a p.u. count of

$$h_k = \frac{2}{(13)(2)} = 0.0769 \tag{4.14}$$

for all pitch interval sizes.

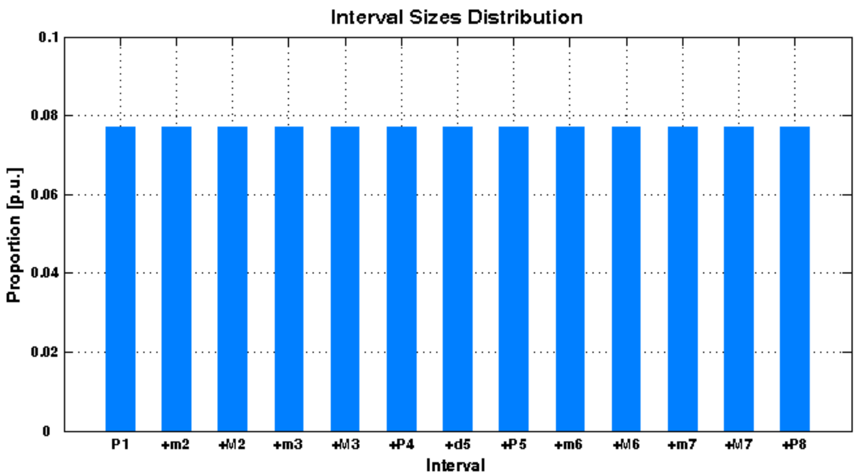


Fig. 4.29 Normalised unweighted distribution of pitch interval sizes for the test score shown in Fig. 4.23.

Fig. 4.30 shows the weighted distribution of pitch interval sizes for the test score. As shown in section 3.2.3.2, two successive quarter notes with a tempo of 120 BPM in the test score translate to a weighted duration of 0.7992. For two occurrences, this yields a cumulative weighted duration of

$$h_k = (2)(0.7992) = 1.5984. \tag{4.15}$$

Fig. 4.30 correctly reflects a cumulative weighted duration of 1.5984 for all pitch interval sizes.

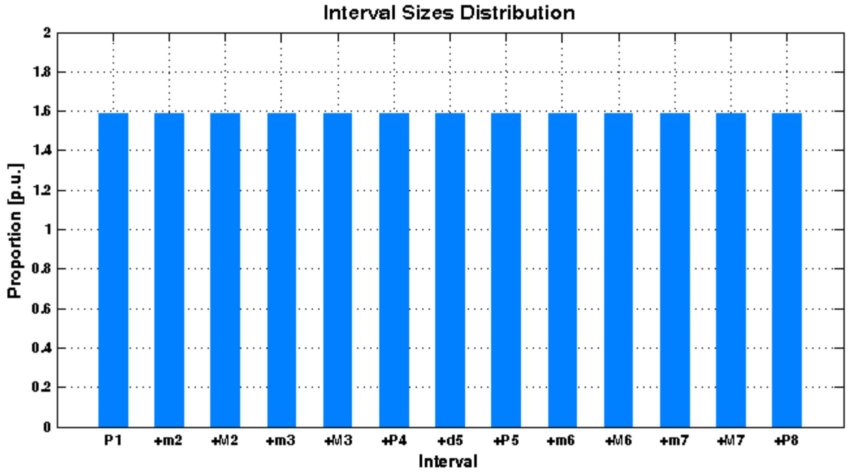


Fig. 4.30 Weighted distribution of pitch interval sizes for the test score shown in Fig. 4.23.

Fig. 4.31 shows the normalised weighted distribution of pitch interval sizes for the test score. The distribution, for all pitch interval sizes, correctly reflects a p.u. cumulative weighted duration of

$$h_k = \frac{1.5984}{(13)(1.5984)} = 0.0769. \tag{4.16}$$

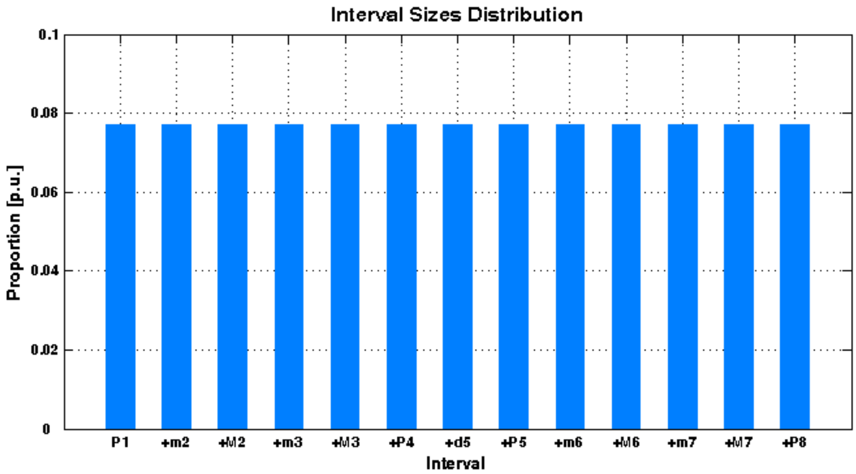


Fig. 4.31 Normalised weighted distribution of pitch interval sizes for the test score shown in Fig. 4.23.

4.6.3.3 Distribution of pitch intervals directions

The algorithms as implemented in Matlab for the distributions of pitch interval directions were verified using the test score shown in Fig. 4.23. This test score contains one positive and one negative transition of each of the pitch intervals listed in (3.33).

Fig. 4.32 shows the unweighted distribution of pitch interval directions for the test score. The distribution correctly reflects a count of zero for all intervals. This is expected as the test score contains an equal number of positive and negative transitions for all interval sizes. The normalised unweighted distribution of pitch interval directions correctly shows a similar result.

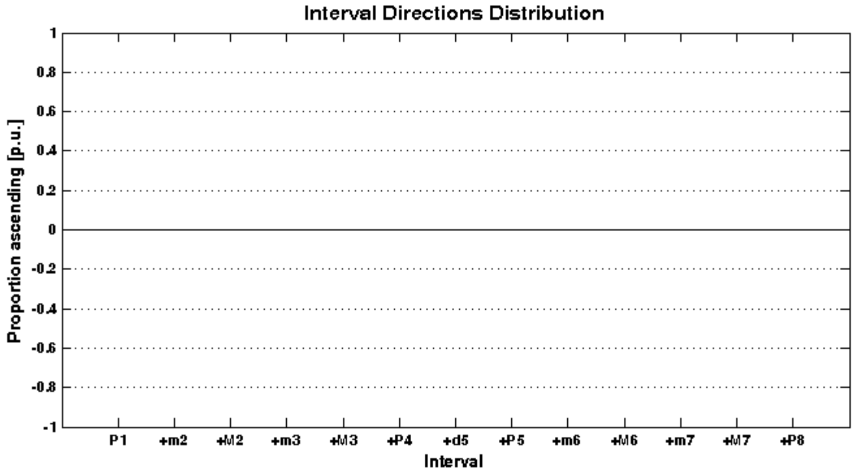


Fig. 4.32 Unweighted distribution of pitch interval directions for the test score shown in Fig. 4.23.

Fig. 4.33 shows the weighted distribution of pitch interval directions for the test score. The distribution correctly reflects a cumulative weighted duration of zero for all interval sizes. This is expected as the test score contains an equal number of positive and negative transitions with equal durational weights for all interval sizes. The normalised weighted distribution of pitch interval directions correctly shows a similar result.

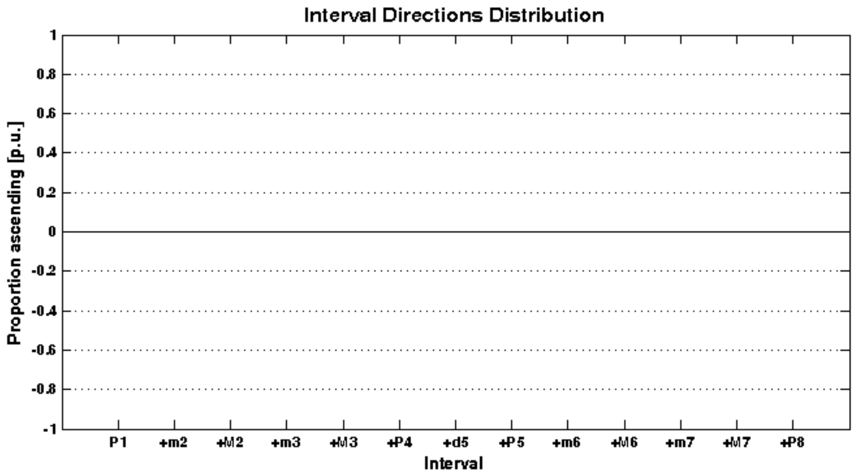


Fig. 4.33 Weighted distribution of pitch interval directions for the test score shown in Fig. 4.23.

4.6.3.4 Distribution of dyad pitch intervals

The algorithms as implemented in Matlab for the distributions of dyad pitch intervals were verified using the test score shown in Fig. 4.23. This test score contains all of the pitch intervals listed in (3.23) and gives rise to the following set of successive dyad intervals:

$$\left\{\begin{array}{l}\{P1,m2\},\{-m2,M2\},\{-M2,m3\},\{-m3,M3\},\{-M3,P4\},\{-P_4,d5\},\{-d5,P5\},\{-P5,m6\},\{-m6,M6\},\{-M6,m7\},\\ \{-m7,M7\},\{-M7,P8\},\{-m7,P8\},\{m2,-m2\},\{M2,-M2\},\{m3,-m3\},\{M3,-M3\},\{P4,-P4\},\{d5,-d5\},\{P5,-P5\},\\ \{m6,-m6\},\{M6,-M6\},\{m7,-m7\},\{M7,-M7\},\{P8,-P8\}\end{array}\right\} \quad (4.17)$$

Fig. 4.34 shows the unweighted distribution of dyad pitch intervals for the test score. This distribution correctly reflects a count of one for the interval combinations listed in (4.17) with counts of zero for all other combinations.

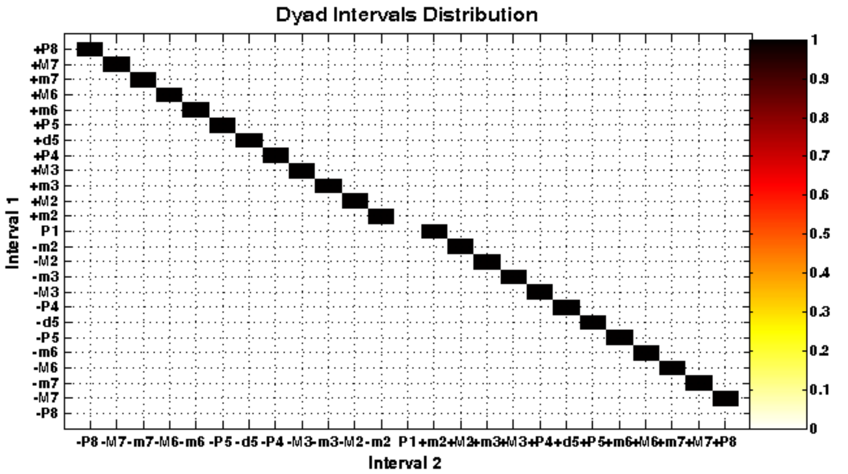


Fig. 4.34 Unweighted distribution of dyad pitch intervals for the test score shown in Fig. 4.23.

Fig. 4.35 shows the normalised unweighted distribution of dyad pitch intervals for the test score. This distribution correctly reflects a p.u. count of

$$\frac{(1)}{(25)(1)} = 0.0400 \tag{4.18}$$

for the interval combinations listed in (4.17) with counts of zero for all other combinations.

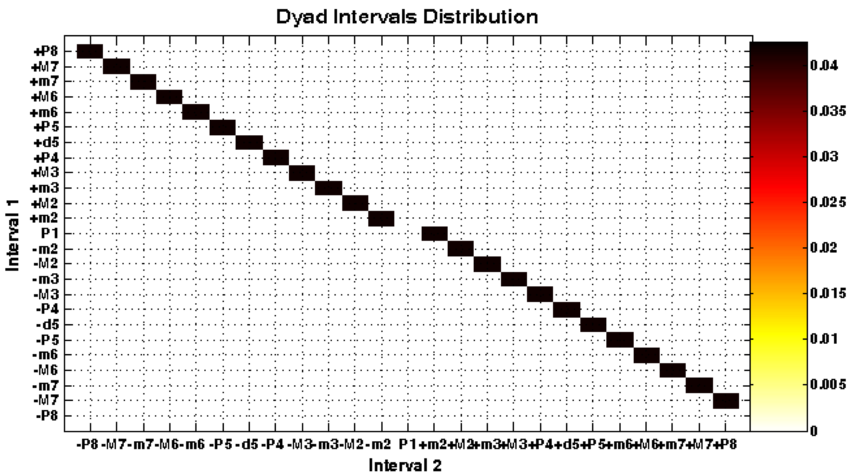


Fig. 4.35 Normalised unweighted distribution of dyad pitch intervals for the test score shown in Fig. 4.23.

Fig. 4.36 shows the weighted distribution of dyad pitch intervals for the test score. As shown in section 3.2.3.2, two successive quarter notes with a tempo of 120 BPM in the test score translate to a durational weight of 0.7992. The weighted duration associated with an interval pair is equal to the sum of the weights of the individual intervals. For a dyad interval count of one, this yields

$$h_k = (1)(0.7992 + 0.7992) = 1.5984. \tag{4.19}$$

The distribution shown in Fig. 4.36 correctly reflects a cumulative weight of 1.5984 for the interval combinations listed in (4.17) with counts of zero for all other combinations.

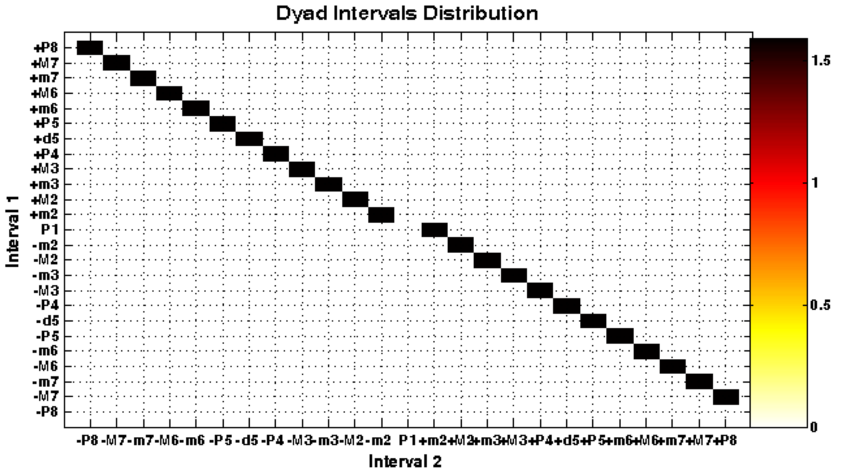


Fig. 4.36 Weighted distribution of dyad pitch intervals for the test score shown in Fig. 4.23.

Fig. 4.37 shows the normalised weighted distribution of dyad pitch intervals for the test score. This distribution correctly reflects a p.u. cumulative weighted duration of

$$h_k = \frac{1.5984}{(25)(1.5984)} = 0.0400 \tag{4.20}$$

for the interval combinations listed in (4.17) with p.u. weights of zero for all other combinations.

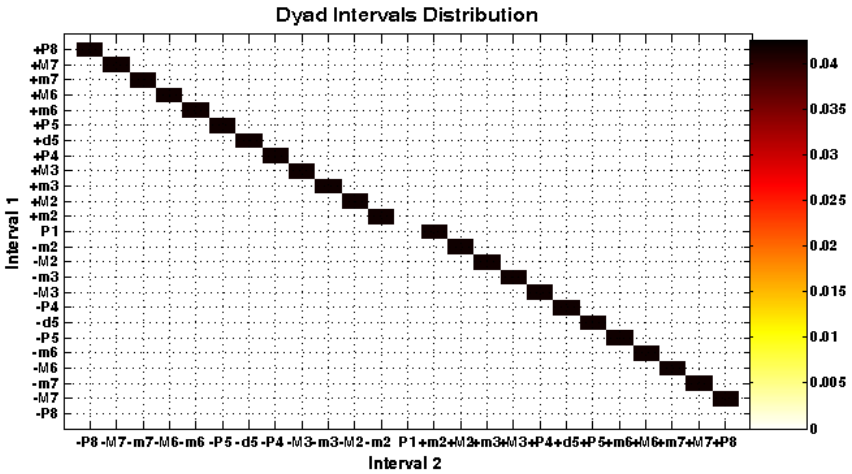


Fig. 4.37 Normalised weighted distribution of dyad pitch intervals for the test score shown in Fig. 4.23.

4.6.4 Statistical analysis of pitch classes

4.6.4.1 Distribution of pitch classes

The algorithms as implemented in Matlab for the distributions of pitch classes were verified using the test score shown in Fig. 4.38. This test score contains all of the pitch classes listed in (3.55) with monotonically decreasing note durations for successive pitch classes. Table 4.10 summarises the note durations and durational weights for the pitch classes contained in the test score for a playback tempo of 100 Beats Per Minute (BPM), Saturation Duration (τ)=0.5 and Accent Index (I_a)=2.

Fig. 4.38 Test score for statistical analysis of pitch classes.

Table 4.10 Cumulative durational weights for the pitch classes contained in the test score shown in Fig. 4.38 for a playback tempo 100 BPM, Saturation Duration (τ)=0.5 and Accent Index (I_a)=2.

Pitch class	**Note duration**	**Cumulative durational weight**	**Normalised cumulative durational weight**
C	Dotted whole note	0.9985	0.1740
$C^{\#}$	Whole note	0.9835	0.1714
D	Dotted half note	0.9458	0.1648
$D^{\#}$	Half note	0.8260	0.1440
E	Dotted quarter note	0.6953	0.1212
F	Quarter note	0.4862	0.0847
$F^{\#}$	Dotted eighth note	0.3497	0.0610
G	Eighth note	0.2011	0.0350
$G^{\#}$	Dotted sixteenth note	0.1290	0.0225
A	Sixteenth note	0.0653	0.0114
$A^{\#}$	Dotted thirty-second note	0.0390	0.0068
B	Thirty-second note	0.0182	0.0032

Fig. 4.39 shows the unweighted distribution of pitch classes for the test score. The distribution correctly reflects a count of one for all pitch classes.

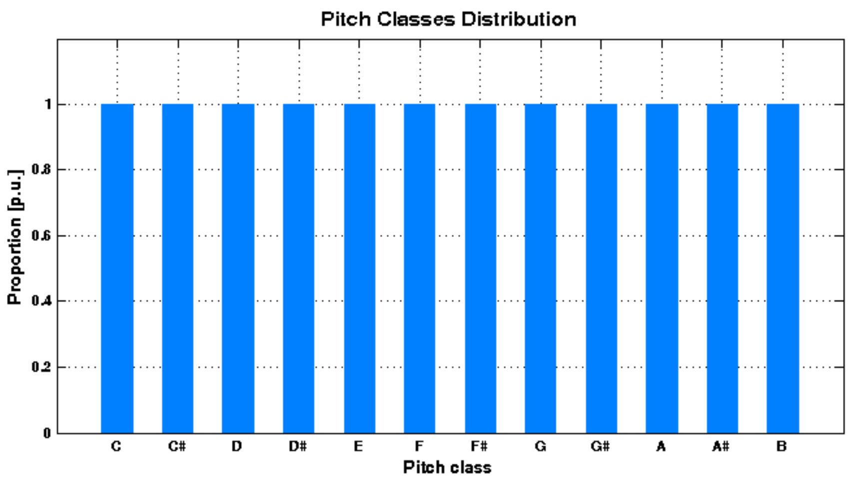


Fig. 4.39 Unweighted distribution of pitch classes for the test score shown in Fig. 4.38.

Fig. 4.40 shows the normalised unweighted distribution of pitch classes for the test score. The distribution correctly shows a p.u. count of

$$h_k = \frac{1}{(12)(1)} = 0.0833 \tag{4.21}$$

for all pitch classes.

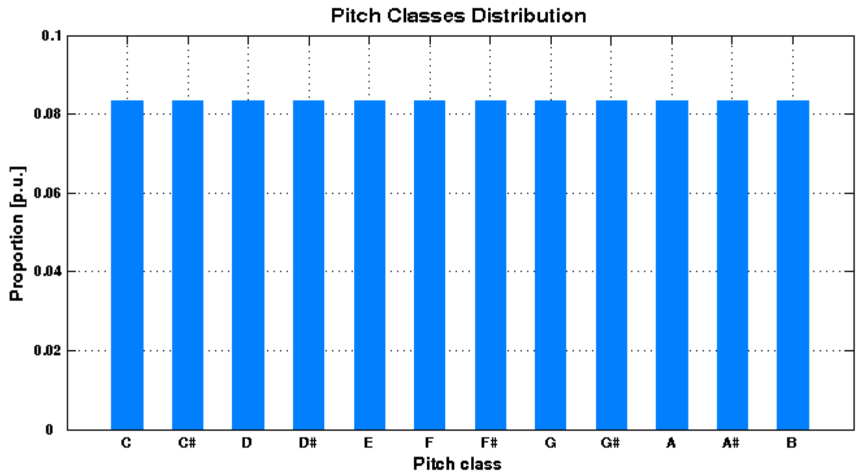


Fig. 4.40 Normalised unweighted distribution of pitch classes for the test score shown in Fig. 4.38.

Fig. 4.41 shows the weighted distribution of pitch classes for the test score. Fig. 4.41 correctly reflects an exponentially decreasing curve for the cumulative durational weights of the pitch classes, varying from 0.9985 for the dotted whole note associated with pitch class C to 0.0182 for the thirty-second note associated with class B in Table 4.10.

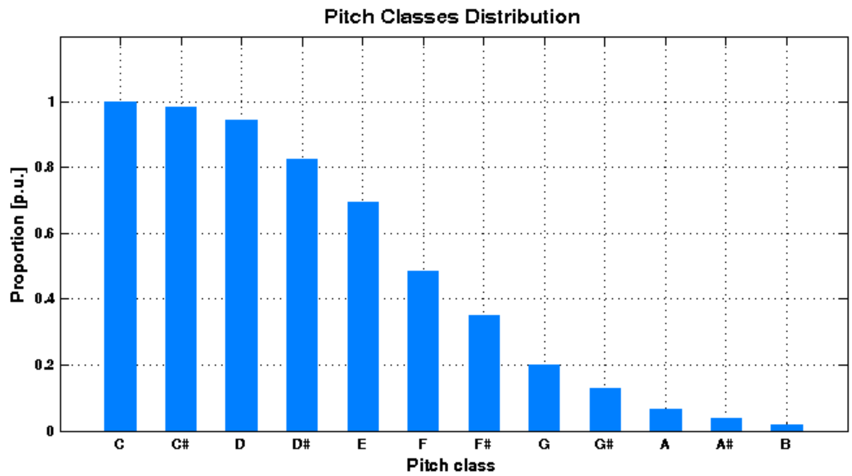


Fig. 4.41 Weighted distribution of pitch classes for the test score shown in Fig. 4.38.

Fig. 4.42 shows the normalised weighted distribution of pitch classes for the test score. The pitch classes listed in Table 4.10 has a total cumulative durational weight of 5.7377. The non-zero normalised cumulative durational weights reflect the results listed in Table 4.10, varying from 0.1740 for the dotted whole note associated with pitch class C to the low value of 0.0032 for the thirty-second note associated with pitch class B.

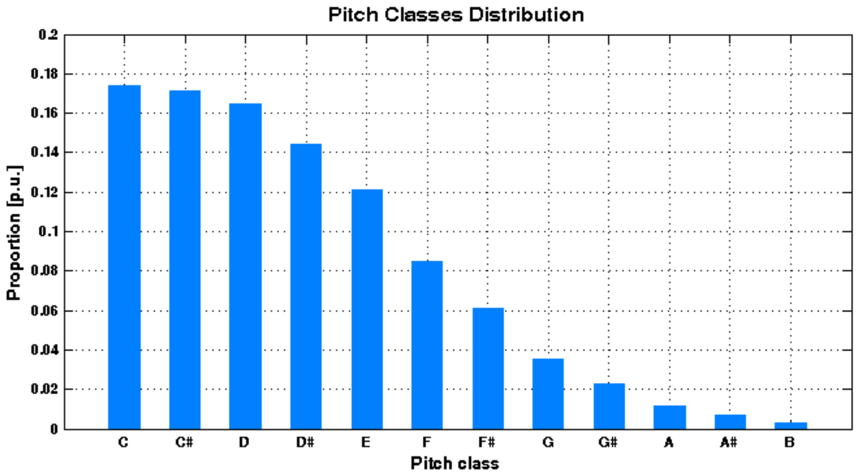


Fig. 4.42 Normalised weighted distribution of pitch classes for the test score shown in Fig. 4.38.

4.6.4.2 Distribution of dyad pitch classes

The algorithms as implemented in Matlab for the dyad pitch class distributions were verified using the test score shown in Fig. 4.38. This test score contains all of the pitch classes listed in (3.55) with monotonically decreasing note durations for the successive pitch classes. It gives rise to the following set of successive dyad pitch classes:

$$\left\{\left\{C,C^{\#}\right\},\left\{C^{\#},D\right\},\left\{D,D^{\#}\right\},\left\{D^{\#},E\right\},\left\{E,F\right\},\left\{F,F^{\#}\right\},\left\{F^{\#},G\right\},\left\{G,G^{\#}\right\},\left\{G^{\#},A\right\},\left\{A,A^{\#}\right\},\left\{A^{\#},B\right\}\right\} \quad (4.22)$$

Table 4.11 summarises the cumulative durational weights for the dyad pitch classes contained in the test score for a playback tempo of 100 Beats Per Minute (BPM), Saturation Duration (τ)=0.5 and Accent Index (I_a)=2.

Table 4.11 Cumulative durational weights for the dyad pitch classes contained in the test score shown in Fig. 4.38 for a playback tempo 100 BPM, Saturation Duration (τ)=0.5 and Accent Index (I_a)=2.

Dyad pitch class	**Note durations**	**Cumulative durational weight**	**Normalised cumulative durational weight**
C, $C^{\#}$	Dotted whole note, Whole note	0.9821	0.2528
$C^{\#}$, D	Whole note, Dotted half note	0.9303	0.2395
D, $D^{\#}$	Dotted half note, Half note	0.7812	0.2011
$D^{\#}$, E	Half note, Dotted quarter note	0.5743	0.1479
E, F	Dotted quarter note, Quarter note	0.3381	0.0870
F, $F^{\#}$	Quarter note, Dotted eighth note	0.1701	0.0438
$F^{\#}$, G	Dotted eighth note, Eighth note	0.0703	0.0181
G, $G^{\#}$	Eighth note, Dotted sixteenth note	0.0259	0.0067
$G^{\#}$, A	Dotted sixteenth note, Sixteenth note	0.0084	0.0022
A, $A^{\#}$	Sixteenth note, Dotted thirty-second note	0.0025	0.0007
$A^{\#}$, B	Dotted thirty-second note, Thirty-second note	0.0007	0.0002

Fig. 4.43 shows the unweighted distribution of dyad pitch classes for the test score. The distribution correctly reflects a count of one for the pitch class combinations listed in (4.22), with counts of zero for all other combinations.

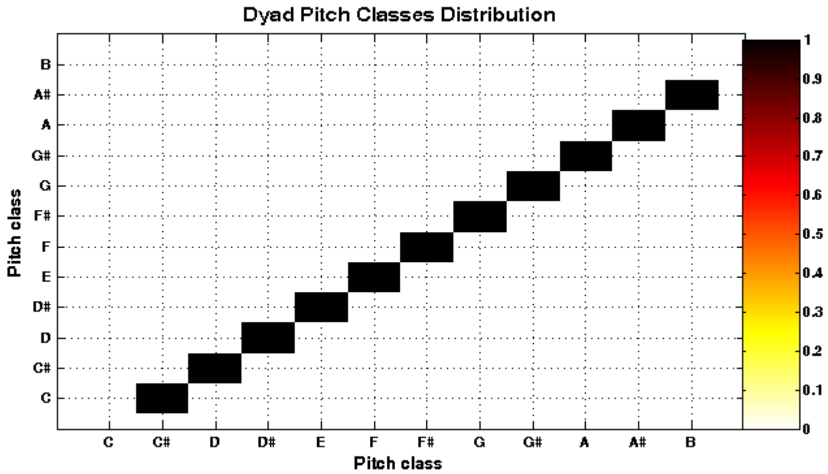


Fig. 4.43 Unweighted distribution of dyad pitch classes for the test score shown in Fig. 4.38.

Fig. 4.44 shows the normalised unweighted distribution of dyad pitch classes for the test score. The distribution correctly reflects a p.u. count of

$$\frac{(1)}{(11)(1)} = 0.0909 \qquad (4.23)$$

for the pitch class combinations listed in (4.22), with counts of zero for all other combinations.

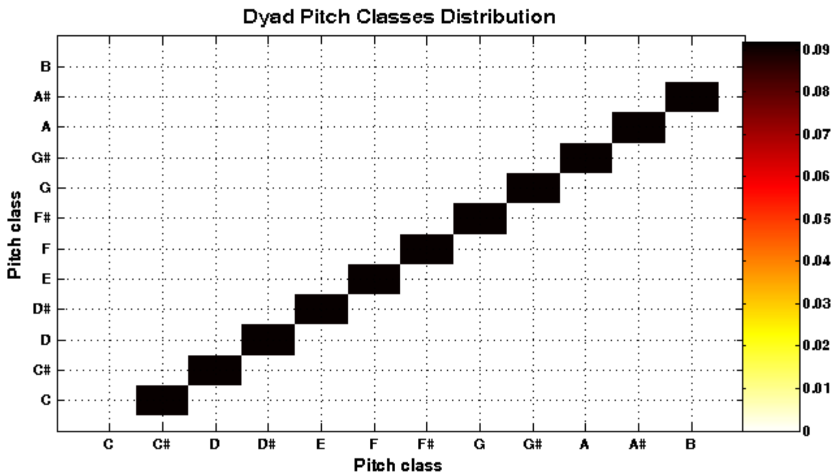


Fig. 4.44 Normalised unweighted distribution of dyad pitch classes for the test score shown in Fig. 4.38.

Fig. 4.45 shows the weighted distribution of dyad pitch classes for the test score. The distribution correctly reflects non-zero cumulative durational weights for the pitch class combinations listed in (4.22) with zero values for all other combinations. The non-zero cumulative durational weights reflects the results listed in Table 4.11, varying from 0.9821 for the dotted whole note and whole note combination of dyad pitch classes C and $C^{\#}$ to the low value of 0.0007 for the dotted thirty-second note and thirty-second note combination of dyad pitch classes $A^{\#}$ and B.

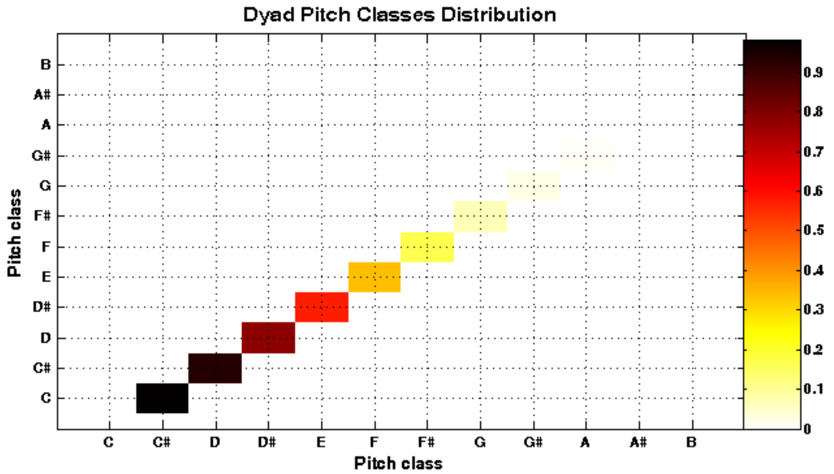


Fig. 4.45 Weighted distribution of dyad pitch classes for the test score shown in Fig. 4.38.

Fig. 4.46 shows the normalised weighted distribution of dyad pitch classes for the test score. The dyad pitch classes listed in Table 4.11 has a total cumulative durational weight of 3.8840. The non-zero normalised cumulative durational weights reflects the results listed in Table 4.11, varying from 0.2528 for the dotted whole note and whole note combination of dyad pitch classes C and $C^{\#}$ to the low value of 0.0002 for the dotted thirty-second note and thirty-second note combination of dyad pitch classes $A^{\#}$ and B.

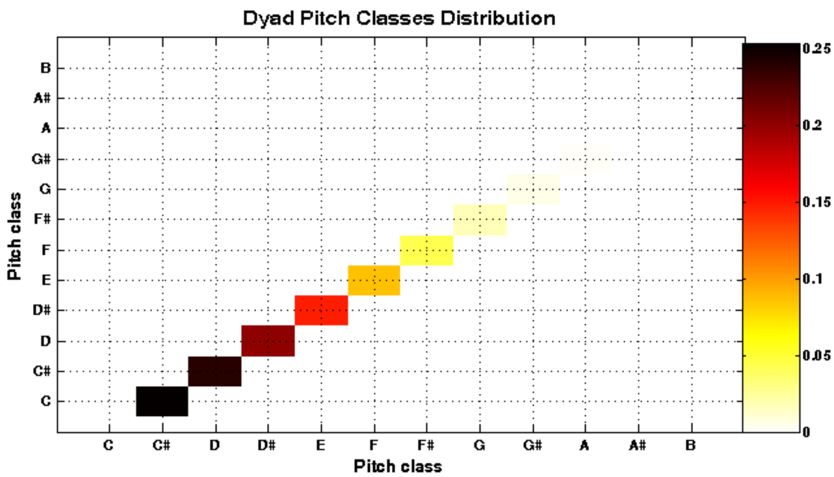


Fig. 4.46 Normalised weighted distribution of dyad pitch classes for the test score shown in Fig. 4.38.

4.6.5 Tonality analysis

4.6.5.1 Implementation of the Krumhansl and Schmuckler and Temperley key-finding algorithms

The algorithms as implemented in Matlab for the Krumhansl and Schmuckler and the modified Temperley key-finding methodologies were verified using the results published by Temperley (1999) for measures 1 to 40 of the *Courante* from Bach's *Cello Suite in C major (BWV 1009)*. This material is particularly well suited for the evaluation procedure because of the following:

- From a tonality perspective, the Bach *Courante* represents a relatively complex and varied source compared to the alternatives found in literature, including the first four notes from the 48 preludes of Bach's *Well-tempered Clavier* used by Krumhansl and Schmuckler (Krumhansl, 1990).

- The results published by Temperley include assessments of the key for each of the individual measures using both the Krumhansl and Schmuckler and the modified Temperley key-finding algorithms.

Fig. 4.47 shows the score for measures 1 to 40 of the *Courante* from Bach's *Cello Suite in C major (BWV 1009)*. Table 4.12 summarises the key-finding results obtained by Temperley (1999), where the columns labelled *K-S* and *TI* denote the key-finding results for the Krumhansl and Schmuckler and the modified Temperley key-finding algorithms respectively.

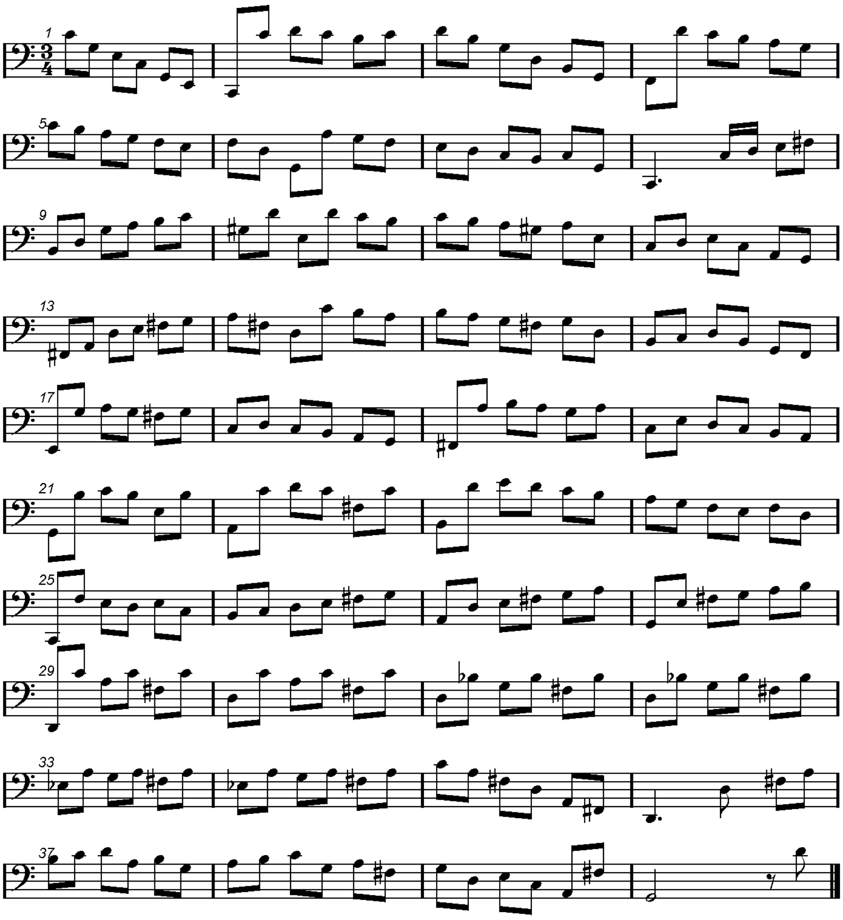


Fig. 4.47 Score for the first half of the courante from Bach's cello suite in C major.

The following comments apply for the results presented in Table 4.12:

- In a number of cases, the Temperley algorithm identifies multiple key options. These include measures 2, 14, 16, 17, 18, 33, 34, 35 and 40, i.e. a total of 9 of 40 or 22.5% of the measures.
- For the measures identified by the shaded areas, the Krumhansl and Schmuckler and the modified Temperley key-finding algorithms yielded different keys. These include measures 4, 10, 14, 16, 19, 22, 23 29, 30, 33 34 and 38, i.e. a total of 12 of 40 or 30% of the measures. These do not include the cases where the Krumhansl and Schmuckler algorithm predicted one of the multiple keys predicted by the Temperley algorithm, i.e. measures 2, 17, 18, 35 and 40, i.e. a total of 5 of 40 or 12.5% of the measures.
- The column notated as *TII* lists the keys determined by Temperley through a musical theoretical analytical approach and also using the proposed preference rule algorithm. Keys in the columns marked *K-S* and *TI* that differ from these reference values are typed in bold. For the Krumhansl and Schmuckler algorithm, these include measures 4, 8, 10, 14, 16, 22, 23, 29, 30, 33, 34 and 35,

i.e. a total of 12 of 40, or 30% of the measures. For the Temperley algorithm, these include measures 8, 14, 22, 29, 30, 33, 34 and 35, i.e. a total of 8 of 40, or 20% of the measures.

Table 4.12 Key-finding results for measures 1 to 40 of the Courante from Bach's Cello Suite in C major (Temperley, 1999).

Measure	K-S	TI	TII	Measure	K-S	TI	TII
1	C	C	C	21	Em	Em	Em
2	C	C/Cm	C	22	**C**	**Am**	G/Gm
3	G	G	G	23	**Bm**	G	G
4	**G**	C	C	24	Dm	Dm	Dm
5	C	C	C	25	C	C	C
6	Dm	Dm	Dm	26	G	G	G
7	C	C	C	27	D	D	D
8	**C**	**C**	G	28	Em	Em	Em
9	G	G	G	29	**C**	**Am**	G/Gm
10	**Bm**	Am	Am	30	**C**	**Am**	G/Gm
11	Am	Am	Am	31	Gm	Gm	Gm
12	C	C	C	32	Gm	Gm	Gm
13	D	D	D	33	**A**	**D**/Em	Em
14	**D**	G/**Am**	G	34	**A**	**D**/Em	Em
15	G	G	G	35	**F#m**	**D/F#m**	G/Gm
16	**G**	C/Cm	C/Cm	36	D	D	D
17	G	G/Em	G/Em	37	G	G	G
18	C	C/G	C/G	38	Am	G	G
19	**A**	D	D	39	C	G	G
20	Am	Am	Am	40	G	G/Gm	G/Gm

Fig. 4.48 shows the key progression calculated with the software application, using the Krumhansl and Kessler key profiles and Krumhansl key correlation formula, for measures 1 to 40 of the *Courante* from Bach's *Cello Suite in C major*. Table 4.13 summarises the algorithm parameter settings used in the estimation.

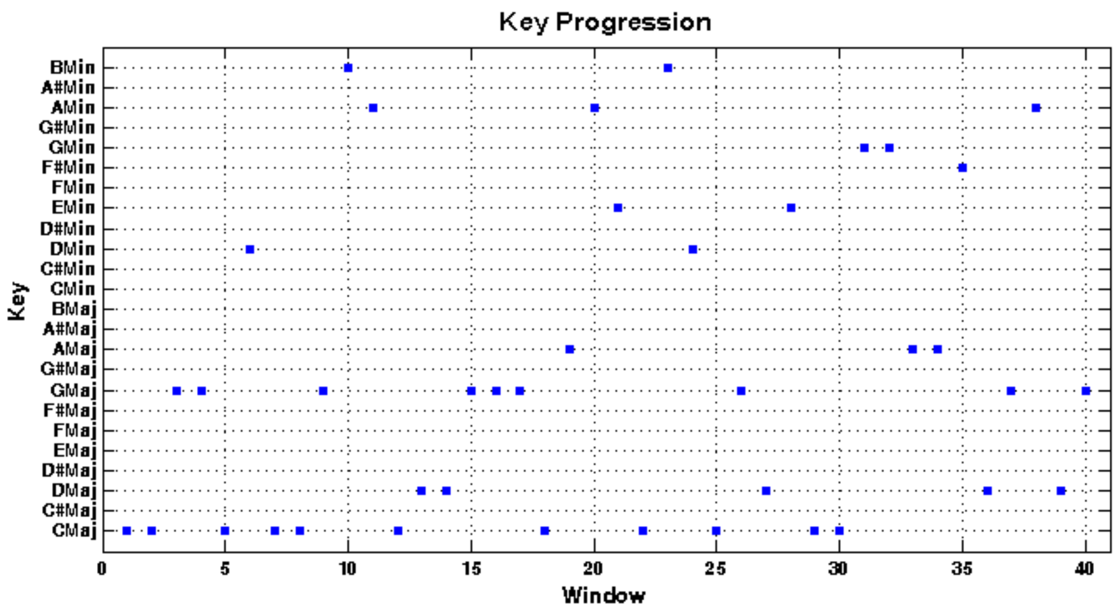


Fig. 4.48 Key progression calculated with the Krumhansl and Schmuckler algorithm, using the Krumhansl and Kessler key profiles and Krumhansl key correlation formula, for measures 1 to 40 of the Courante from Bach's Cello Suite in C major.

Table 4.13 Algorithm parameters for evaluation of the Krumhansl and Schmuckler algorithm, using the Krumhansl and Kessler key profiles and Krumhansl key correlation formula.

Category	Parameter	Value	Category	Parameter	Value
Window	Mode	Sliding	Pitch class distribution	Durational accent model	Linear
	Unit	Beats		Duration quantisize	1/64
	Start	0	Algorithm	Pitch class hierarchy	Krumhansl & Kessler
	Size	3		Correlation formula	Krumhansl & Kessler
	Increment	3		Key resolution	0.1 %

The key progression gives rise to the following conclusions:

- Fig. 4.48 shows that the keys identified for the individual measures correlate exactly with the results given in Table 4.12 for the K-S algorithm, except for measure 39 where the key progression produced by the software implementation shows D_{Maj} instead of C_{Maj}.
- Fig. 4.49 and Fig. 4.50 show the pitch class distribution and key correlation coefficients calculated for measure 39 using the parameters listed in Table 4.13. Keys C_{Maj}, D_{Maj} and G_{Maj} all rate high based on the key correlation coefficients shown in Fig. 4.50. It follows that the discrepancy between the result obtained with the software application and the result reported by Temperley (1999) can possibly be due to the numerical effects discussed in section 3.3.1.1. The score material published by Temperley (1999) shows an error for measure 39, in the sense that it contains 6 quarter notes rather than 6 eight notes as required by the time signature of ¾ and as reflected in alternative sheet music sources. While it is not clear whether this affected the result, it is concluded that the C_{Maj} reported by Temperley is incorrect and that the key estimate of D_{Maj} delivered by the software application is correct.

The results of the above comparison validate the mathematical formulation and Matlab implementation of the Krumhansl and Schmuckler key-finding algorithm using the Krumhansl and Kessler key profiles and Krumhansl key correlation formula.

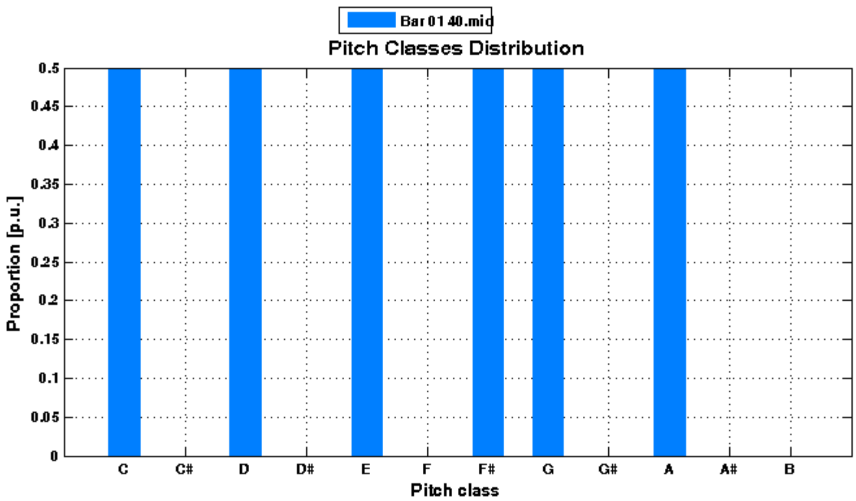


Fig. 4.49 Pitch class distribution calculated with the parameters listed in Table 4.13 for measure 39 of the Courante from Bach's Cello Suite in C major.

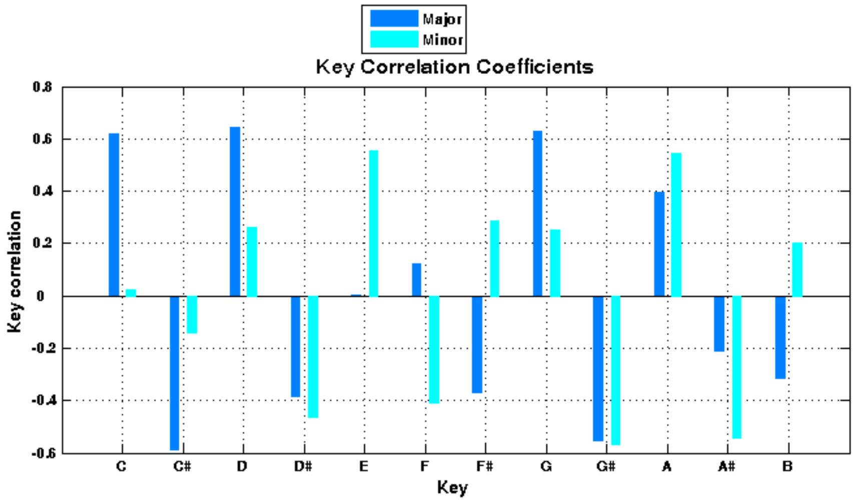


Fig. 4.50 Key correlation coefficients calculated with the Krumhansl and Schmuckler algorithm and the parameters listed in Table 4.13 for measure 39 of the Courante from Bach's Cello Suite in C major.

In an alternative validation comparison, the key correlation values calculated with the software application for the Krumhansl and Schmuckler algorithm, using the Krumhansl and Kessler key profiles and Krumhansl key correlation formula, were compared with results published by Temperley (1999) for the first measure of the familiar *"Yankee Doodle"* shown in Fig. 4.51. Table 4.14 compares the results obtained by Temperley with the results produced by the software application.

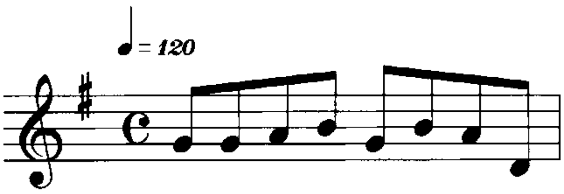


Fig. 4.51 First measure of "Yankee Doodle" (Temperley, 1999).

Table 4.14 Comparison of the key correlation values published by Temperley (1999) and the software application for the first measure of "**Yankee Doodle*****". The results were obtained using the Krumhansl and Kessler key profiles and Krumhansl key correlation formula.***

Key	Temperley	Application	Key	Temperley	Application
C major	0.245	0.2742	C minor	-0.012	-0.0129
C# major	-0.497	-0.5593	C# minor	-.0.296	-0.3324
D major	0.485	0.5432	D minor	0.133	0.1494
Eb major	-0.114	-0.1297	Eb minor	-0.354	-0.3977
E major	0.000	-0.0013	E minor	0.398	0.4471
F major	0.003	0.0026	F minor	-.0.384	-0.4314
F# major	-0.339	-0.3814	F# minor	0.010	0.0117
G major	0.693	0.7770	G minor	0.394	0.4429
Ab major	-0.432	-0.4867	Ab minor	-.0.094	-0.1056
A major	0.159	0.1772	A minor	0.223	0.2512
Bb major	-0.129	-0.1464	Bb minor	-0.457	-0.5129
B major	-0.061	-0.0694	B minor	-.436	0.4907

The two sets of results reflect the same trends and give rise to the same key, i.e. G major, as shown in Fig. 4.52 for the key correlation plot produced by the application, but the numerical values do not correlate. The pitch class distribution coefficients reported by Temperley are half the values shown in Fig. 4.53 for the results obtained with the software application. This is due to the fact that

Temperley used note durations specified in seconds while the application used note durations specified in beats. This does not, however, affect the numerical values due to the normalising nature of the Krumhansl correlation formula.

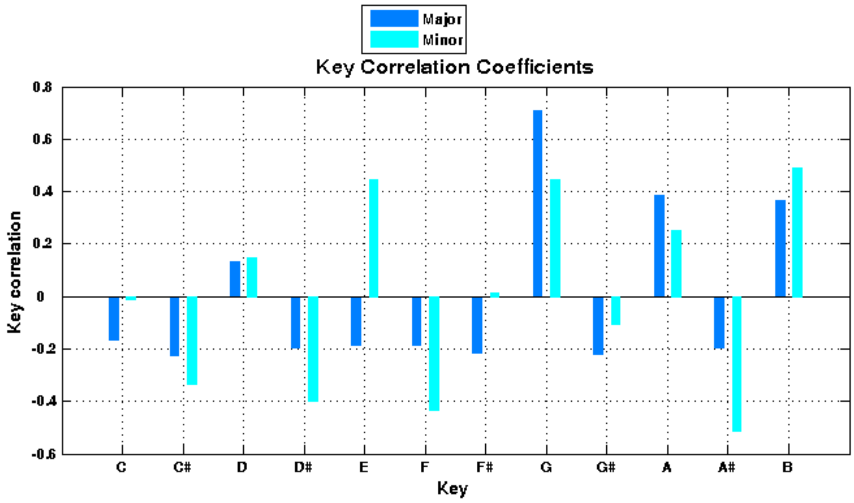


Fig. 4.52 Key correlations calculated with the software application for the first measure of "Yankee Doodle" using the Krumhansl and Kessler key profiles and the Krumhansl key correlation formula.

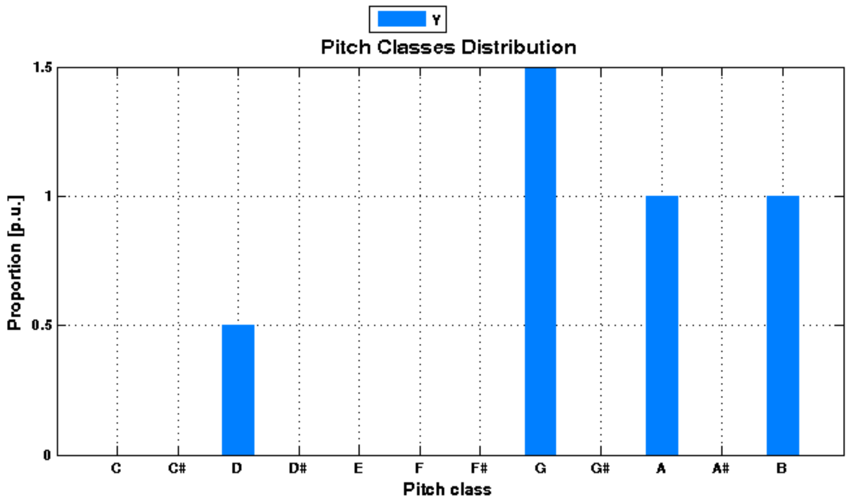


Fig. 4.53 Pitch class distribution calculated with the software application for the first measure of "Yankee Doodle" using linear durational weighing in beats.

Fig. 4.54 shows the key progression calculated with the software application for the Krumhansl and Schmuckler algorithm, using the Temperley key profiles and key correlation formula, for measures 1 to 40 of the *Courante* from Bach's *Cello Suite in C major*. Table 4.15 summarises the algorithm parameter settings used in the estimation. The key progression gives rise to the following conclusions:

- The key progression shown in Fig. 4.54 reflects that the keys identified for the individual measures, including the cases where multiple keys are indicated, correlate exactly with the results given in Table 4.12 for the *TI* algorithm for all measures.
- With regard to the selection of multiple keys, the results obviously depend on the key resolution parameter, i.e. the region of valid key correlation values, used in the calculation. Temperley, however, does not address this issue in his research paper at all (Temperley, 1999). As indicated in Table 4.15, the results in Fig. 4.54, applies for a relatively small arbitrarily chosen key resolution value of 0.1% of the maximum key correlation value achieved in the analysis.

The results for the above comparison validate the mathematical formulation and Matlab implementation of the Krumhansl and Schmuckler key-finding algorithm using the Temperley key profiles and key correlation formula.

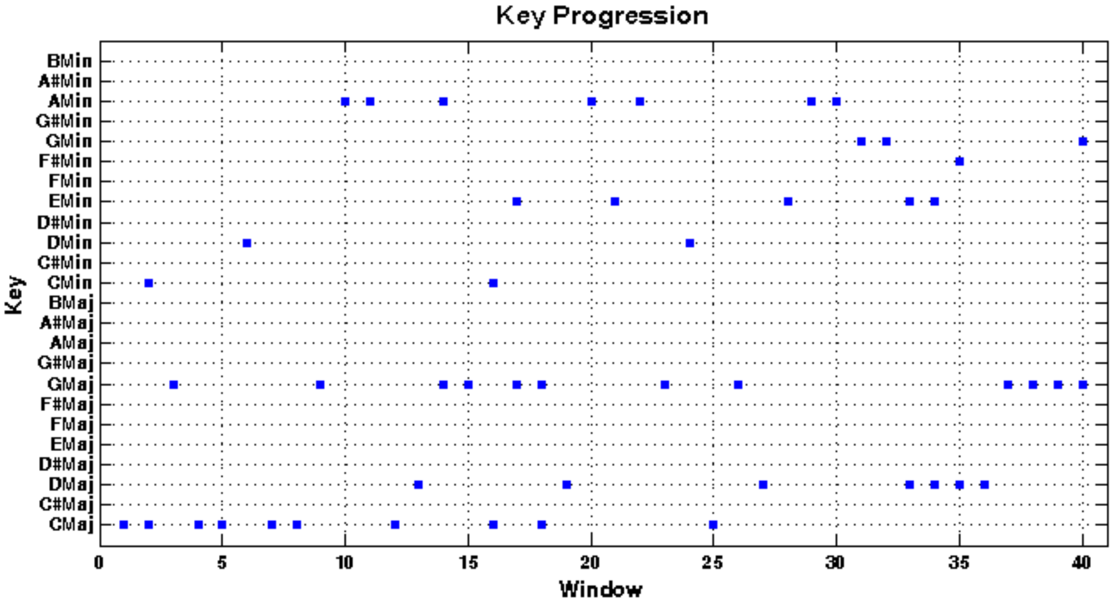


Fig. 4.54 Key progression calculated with the Krumhansl and Schmuckler algorithm, using the Temperley key profiles and key correlation formula, for measures 1 to 40 of the Courante from Bach's Cello Suite in C.

Table 4.15 Algorithm parameters for evaluation of the Krumhansl and Schmuckler algorithm, using the Temperley key profiles and key correlation formula.

Category	Parameter	Value	Category	Parameter	Value
Window	Mode	Sliding	Pitch class distribution	Durational accent model	Linear
	Unit	Beats		Duration quantisize	1/64
	Start	0	Algorithm	Pitch class hierarchy	Temperley
	Size	3		Correlation formula	Temperley
	Increment	3		Key resolution	0.1 %

Fig. 4.55 shows the key progression calculated with the software application for the Krumhansl and Schmuckler key-finding algorithm, using the key profiles, key correlation formula and flat pitch classes input vector proposed by Temperley (1999), for measures 1 to 40 of the Courante from Bach's Cello Suite in C. Table 4.16 summarises the algorithm parameter settings used in the estimation. The key progression gives rise to the following conclusions:

- Fig. 4.55 shows that the keys identified for the individual measures, including the cases where multiple keys are indicated, correlate exactly with the results given in Table 4.12 for the TII algorithm for all measures.
- With regards to the selection of multiple keys, the results obviously depend on the key resolution parameter, i.e. the region of valid key correlation values, used in the calculation. Temperley, however, does not address this issue in his research paper at all (Temperley 1999). As indicated in Table 4.15, the results in Fig. 4.54, applies for a relatively small arbitrarily chosen key resolution value of 0.1% of the maximum key correlation value achieved in the analysis.

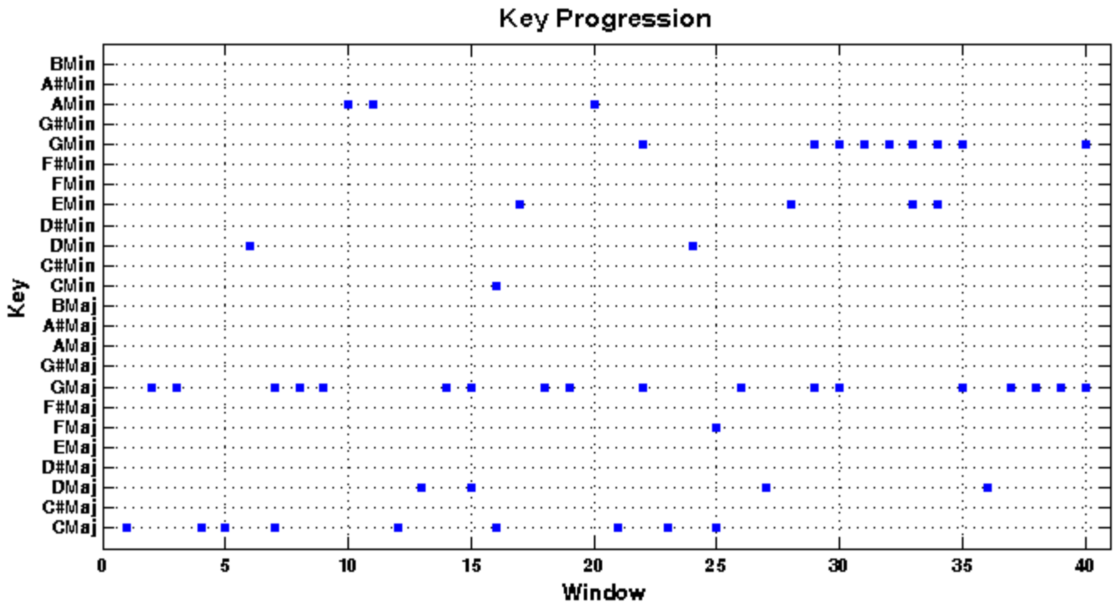


Fig. 4.55 Key progression calculated with the Krumhansl and Schmuckler algorithm, using the key profiles, key correlation formula and flat pitch class vector proposed by Temperley, for measures 1 to 40 of the Courante from Bach's Cello Suite in C.

Table 4.16 Algorithm parameters for evaluation of the Krumhansl and Schmuckler algorithm, using the key profiles, key correlation formula and flat pitch class vector proposed by Temperley.

Category	Parameter	Value	Category	Parameter	Value
Window	Mode	Sliding	Pitch class distribution	Durational accent model	Flat
	Unit	Beats		Duration quantisize	1/64
	Start	0	Algorithm	Pitch class hierarchy	Temperley
	Size	3		Correlation formula	Temperley
	Increment	3		Key resolution	0.1 %

The results for the above comparison validate the mathematical formulation and Matlab implementation of the Krumhansl and Schmuckler key-finding algorithm using the key profiles, key correlation formula and flat pitch class vector proposed by Temperley.

5 Case studies and results

5.1 Introduction

This chapter presents the results achieved with regard to the original project objectives given in section 1.3.1. These are as follows:

- Development of a Matlab application for statistical and tonal analysis of music encoded in symbolic format.
- Evaluation of the Krumhansl and Schmuckler key-finding performance for a number of new combinations of test scores, pitch class representations and key profile templates.
- Determining whether a more optimal key profile template for use with the Krumhansl and Schmuckler key-finding algorithm and variations thereof can be derived using parameter estimation techniques such as direct search methods.

The sample material for the case studies has been selected based on the following considerations:

- In view of the emphasis on key-finding, it is important to make use of material that is rich and diverse from a tonal perspective.
- By selecting material that has also been used in related historical studies, it is possible to interpret the results of this investigation in the context of previous findings.

Based on the literature review, the following pieces have been selected for the case studies:

- The 24 preludes of Bach's *Well-tempered Clavier Book I*.
- The *Courante* from Bach's *Cello Suite in C majo*r.
- The *Gavotte* from Bach's *French Suite No. 5 in G major (BWV 816)*.

5.2 Statistical analysis

5.2.1 Overview

The application was used to derive the distributions of note durations, pitch intervals and pitch classes for the following test cases:

- *The 24 preludes of Bach's Well-tempered Clavier Book I:* The score material in the *Well-tempered Clavier* is polyphonic. The statistical results are therefore limited to distributions of note durations and pitch classes.
- *The Courante from Bach's Cello Suite in C major:* The score material in the *Courante* is monophonic. The full range of statistical results is therefore presented, i.e. distributions of note durations, pitch intervals and pitch classes.
- *The Gavotte from Bach's French Suite No. 5 in G major (BWV 816):* The score material in the *Gavotte* is polyphonic. The statistical results are therefore limited to distributions of note durations and pitch classes.

The importance of the statistical distributions presented for the sample material in this section, particularly with reference to interpretation of the results for the key-finding and key profile estimation algorithms presented in the next section, can be summarised as follows:

- The distributions of note durations represent the diversity of note durations of the score material and thus reflect on the suitability of the material for evaluating the effects of durational weighing in the key-finding algorithms.
- The distributions of pitch classes represent the tonal diversity of the score material and thus reflect on the suitability of the material for evaluating the performance of the key-finding algorithms.

Only the normalised distributions will be presented in the main text. Some additional results, such as the actual histograms, are given in Appendix I.

5.2.2 Preludes of Bach's Well-tempered Clavier Book I

5.2.2.1 Distributions of note durations

Fig. 5.1 shows the normalised distribution of note durations for the 24 preludes of the *Well-tempered Clavier Book I*. The results show that the material favours sixteenth and eighth notes, which together represent about 70% of the note events. All note durations, however, are represented in the distribution.

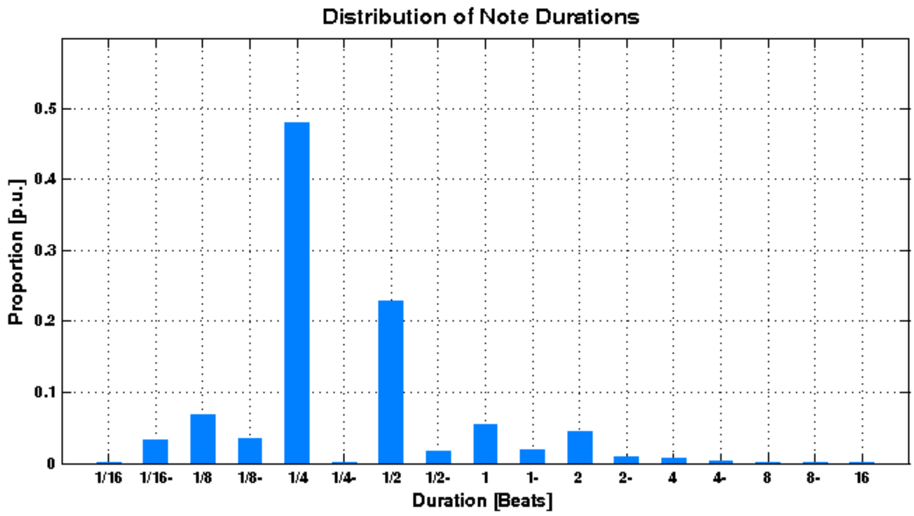


Fig. 5.1 ***Normalised distribution of note durations for the 24 preludes of Bach's Well-tempered Clavier Book I.***

5.2.2.2 Distributions of pitch classes

Fig. 5.2 shows the normalised unweighted distribution of pitch classes for the 24 preludes of the *Well-tempered Clavier Book I*. As expected, based on the fact that the material represents all 24 keys, the distribution exhibits an even spread, i.e. within approximately 10%, over all pitch classes.

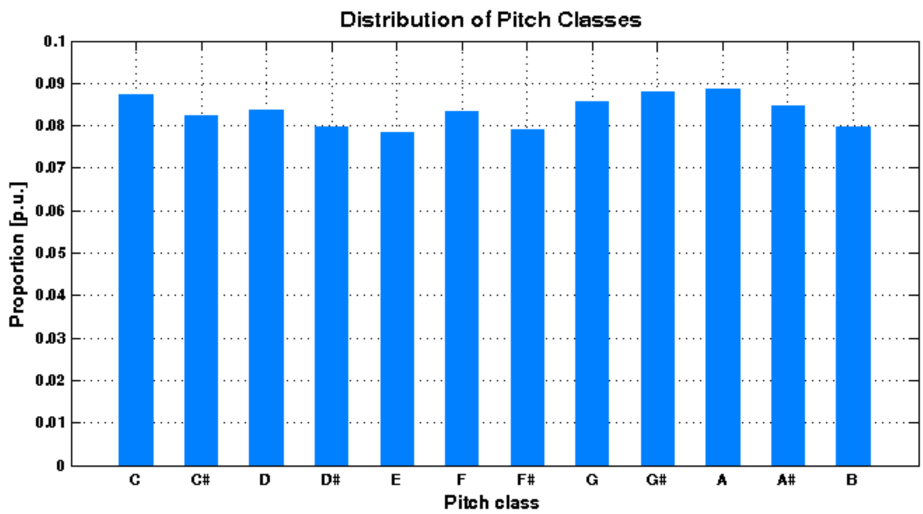


Fig. 5.2 ***Normalised unweighted distribution of pitch classes for the 24 preludes of Bach's Well-tempered Clavier Book I.***

Fig. 5.3 compares the distributions of pitch classes for the 24 preludes of the *Well-tempered Clavier Book I* for the unweighted, linear durational weighing and durational accent weighing models. The distributions show that the introduction of durational weighing can increase or decrease the values recorded for the unweighted histogram. Furthermore, the effect of durational weighing for a particular pitch class is not always in the same direction, i.e. the weight of the pitch class relative to the unweighted case can be increased by one of the models and decreased by the other model.

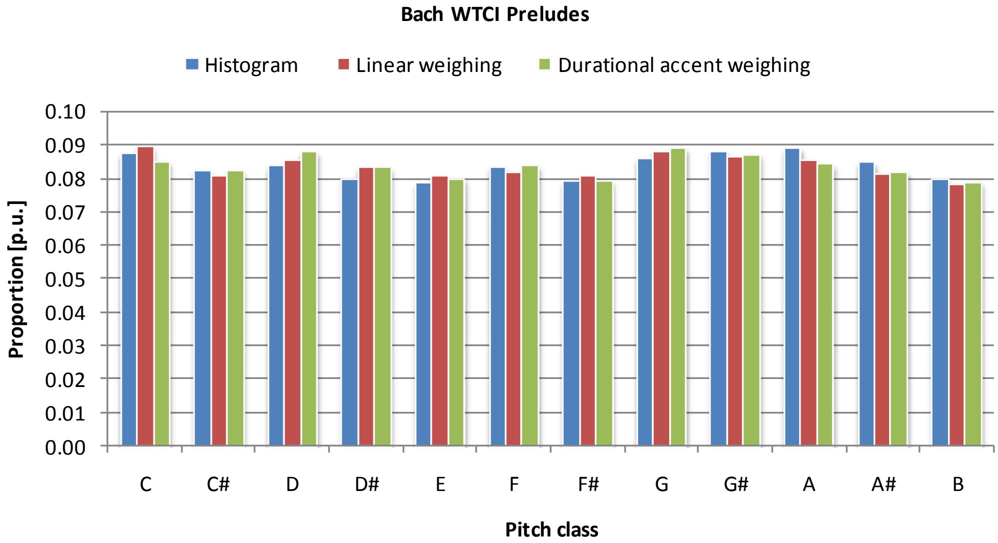


***Fig. 5.3** Normalised distributions of pitch classes for the 24 preludes of Bach's Well-tempered Clavier Book I: Comparison of unweighted, linear durational weighing and durational accent weighing (* $\tau = 0.5$ and $I_a = 2.0$*).*

5.2.3 Courante from Bach's Cello Suite in C major

5.2.3.1 Distributions of note durations

Fig. 5.4 and Fig. 5.5 show the normalised unweighted distributions of note durations and dyad note durations respectively for the *Courante* from Bach's *Cello Suite in C major*.

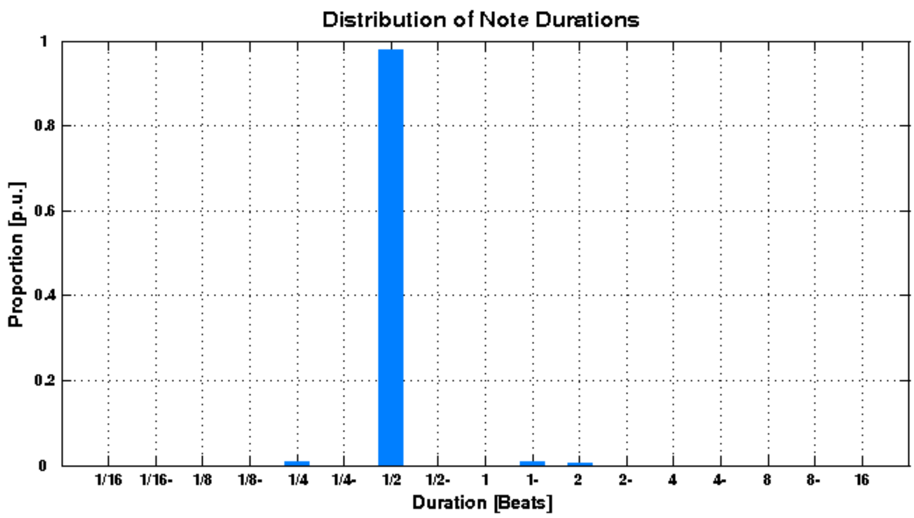


***Fig. 5.4** Normalised distribution of note durations for the Courante from Bach's Cello Suite in C major.*

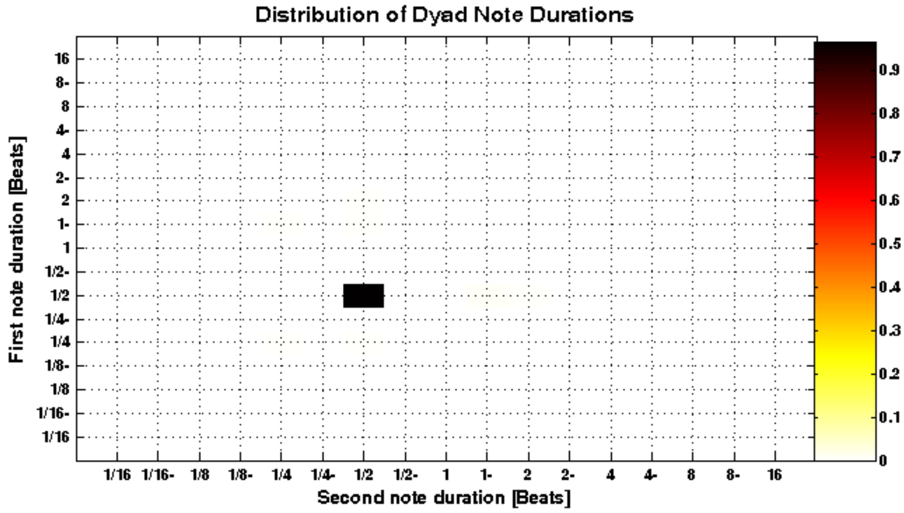


Fig. 5.5 Normalised distribution of dyad note durations for the Courante from Bach's Cello Suite in C major.

The distributions reflect the fact that the score material contains predominantly eight notes and is clearly not suitable for determining whether durational weighing improves the performance of key-finding algorithms. In this regard, it is noteworthy that Temperley (1999) used the *Courante* in comparing the performance of his unweighted algorithm with the weighted algorithm proposed by Krumhansl and Schmuckler (Krumhansl, 1990).

5.2.3.2 Distributions of pitch intervals

Fig. 5.6 and Fig. 5.7 show the normalised distributions of pitch intervals and dyad pitch intervals respectively for the *Courante* from Bach's *Cello Suite in C major*. The results show that the minor 2nd, major 2nd and minor 3rd intervals feature prominently.

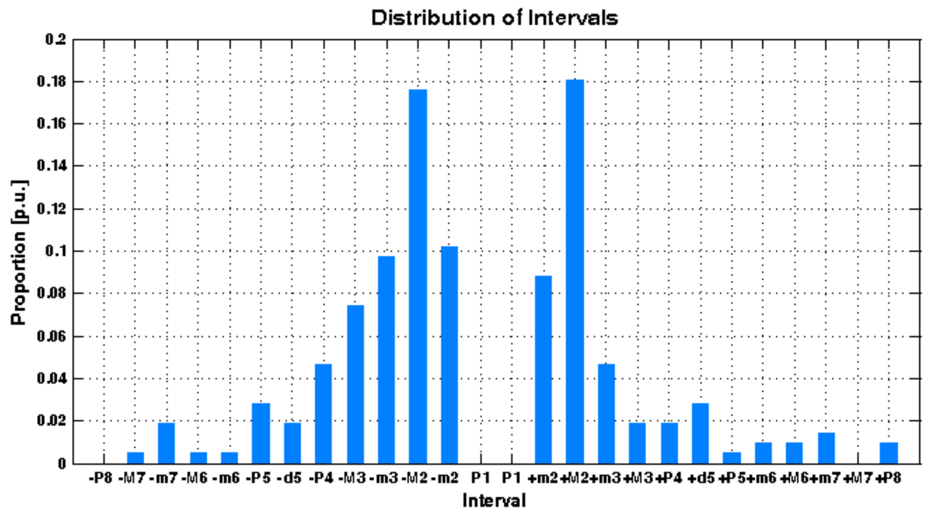


Fig. 5.6 Normalised unweighted distribution of pitch intervals for the Courante from Bach's Cello Suite in C major.

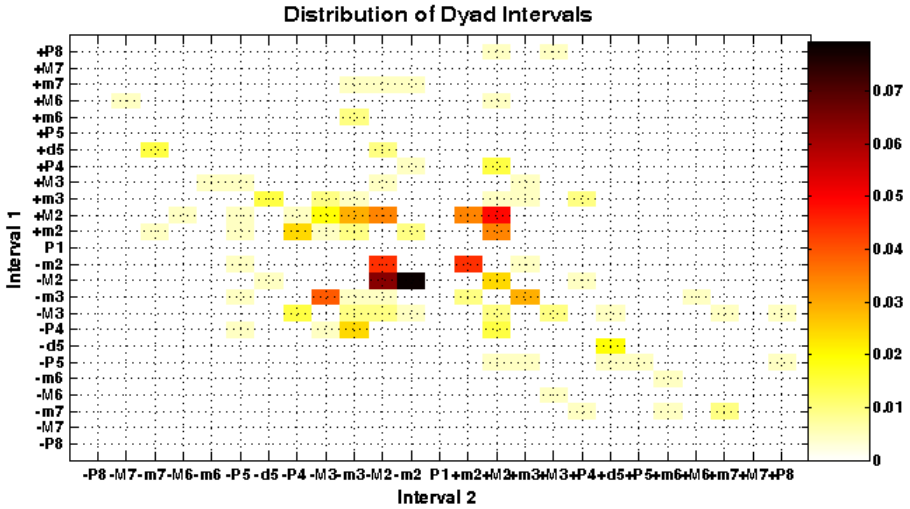


Fig. 5.7 ***Normalised unweighted distribution of dyad pitch intervals for the Courante from Bach's Cello Suite in C major.***

Fig. 5.8 shows the normalised distribution of interval sizes, confirming the importance of the minor 2nd, major 2nd and minor 3rd intervals in the melodic progressions.

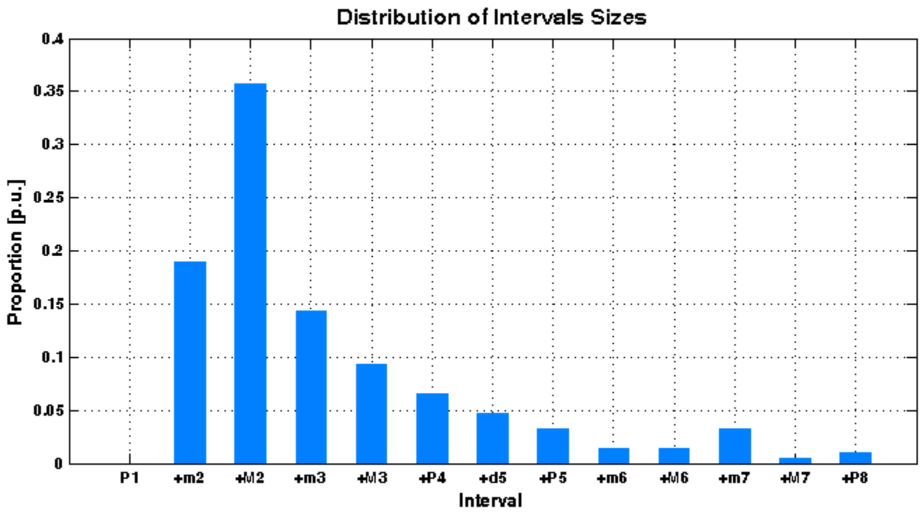


Fig. 5.8 ***Normalised unweighted distribution of pitch interval sizes for the Courante from Bach's Cello Suite in C major.***

Fig. 5.9 shows the normalised distribution of interval directions. The value for the major 2nd interval is almost zero, thus indicating equal movement in the ascending and descending directions. The major 3rd and perfect 5th intervals reflect high proportions in the descending direction. The histograms are given in Appendix I.

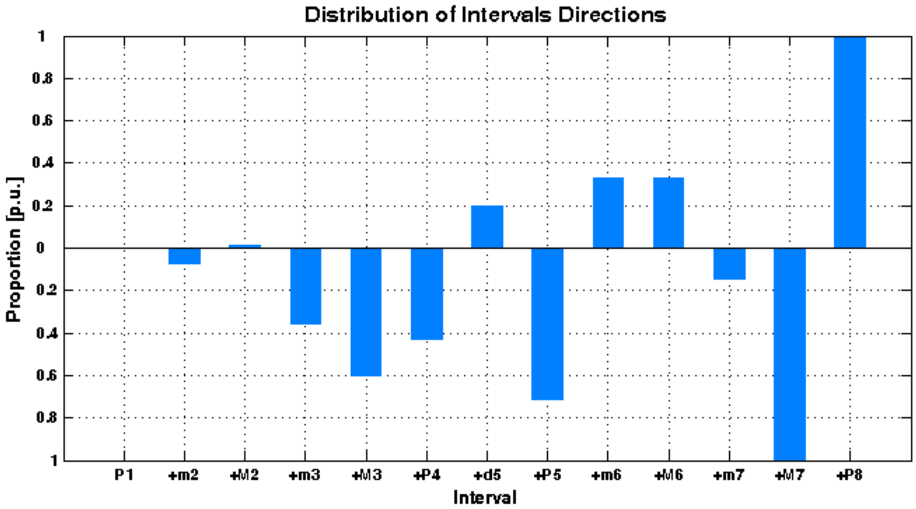


Fig. 5.9 Normalised unweighted distribution of pitch interval directions for the Courante from Bach's Cello Suite in C major.

As discussed in chapter 4, the software application allows for a durational weighing model to be applied in determining the distributions of pitch intervals and pitch classes. Fig. 5.10 compares the distributions of pitch intervals for the *Courante* for unweighted, linear durational weighing and durational accent weighing models. For most pitch intervals, the results correlate closely.

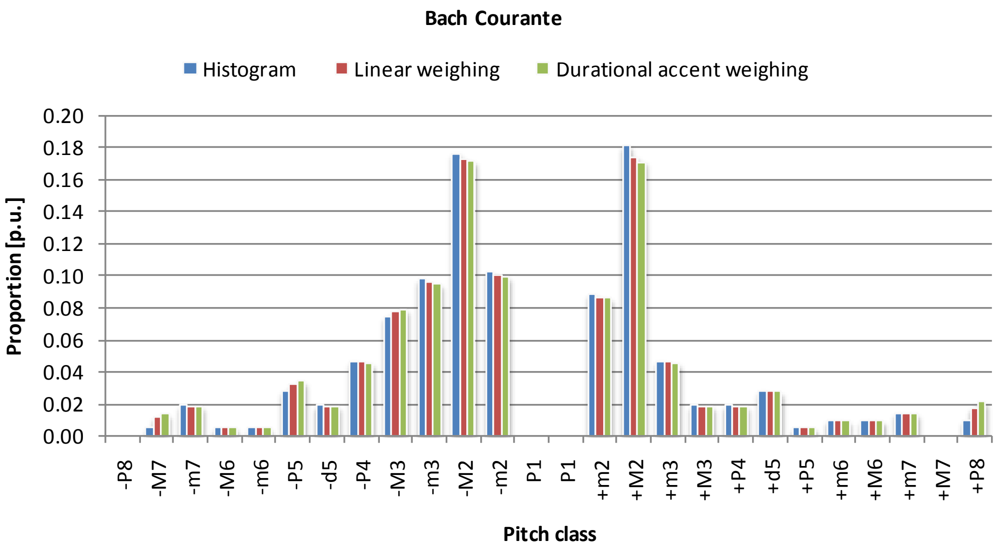


Fig. 5.10 Normalised distributions of pitch intervals for the Courante from Bach's Cello Suite in C major: Comparison of unweighted, linear durational weighing and durational accent weighing ($\tau = 0.5$ and $I_a = 2.0$***).***

5.2.3.3 Distributions of pitch classes

Fig. 5.11 and Fig. 5.12 show the normalised unweighted distributions of pitch classes and dyad pitch classes respectively for the *Courante* from Bach's *Cello Suite in C major*.

From a music theoretical perspective, comparison of the distributions of pitch classes for this score material with the summary of key progressions given in Table 4.12 reflect the following:

- A strong presence of the C major key in the sense that all diatonic pitch classes of the scale are represented.
- A strong presence of the G major key in the sense that all diatonic pitch classes of the scale are represented.
- Indications of modulation to the D minor and E minor keys.

Also note the statistical prominence of successive pitch classes of B-C and G-A in the distribution of dyad pitch classes given in Fig. 5.12. The histograms are given in Appendix I.

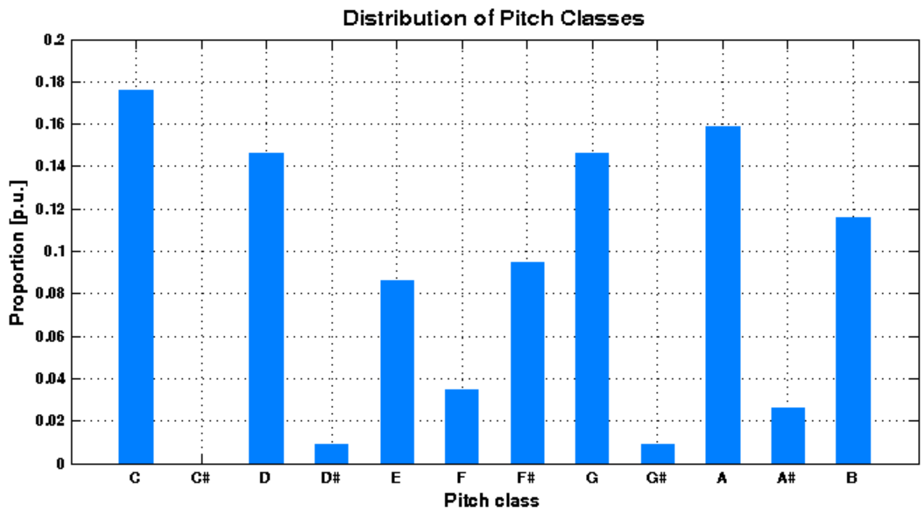


Fig. 5.11 Normalised unweighted distribution of pitch classes for the Courante from Bach's Cello Suite in C major.

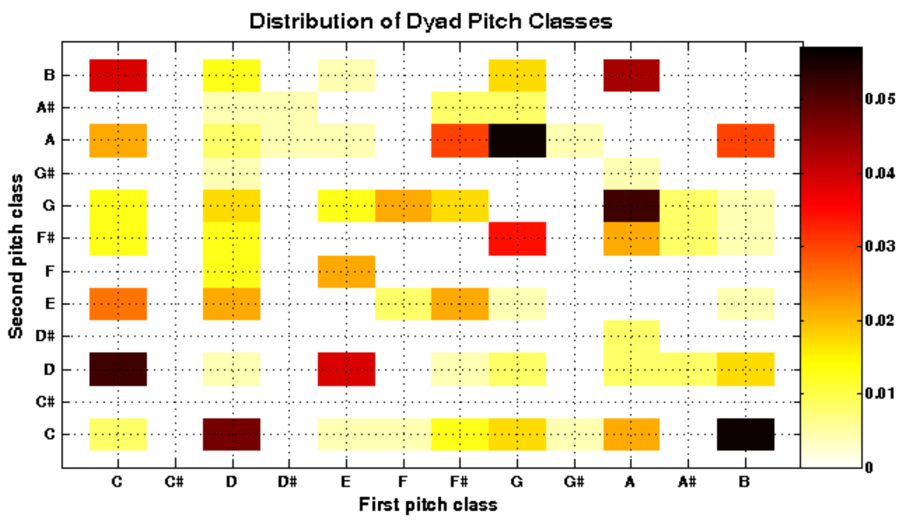


Fig. 5.12 Normalised unweighted distribution of dyad pitch classes for the Courante from Bach's Cello Suite in C major.

Fig. 5.13 compares the distributions of pitch classes for the *Courante* for the unweighted, linear durational weighing and durational accent weighing models. The results for the unweighted case and the linear durational weighing model correlate closely. This is expected in view of the distribution of note durations given in Fig. 5.4, which shows that most of the note events are of the same duration.

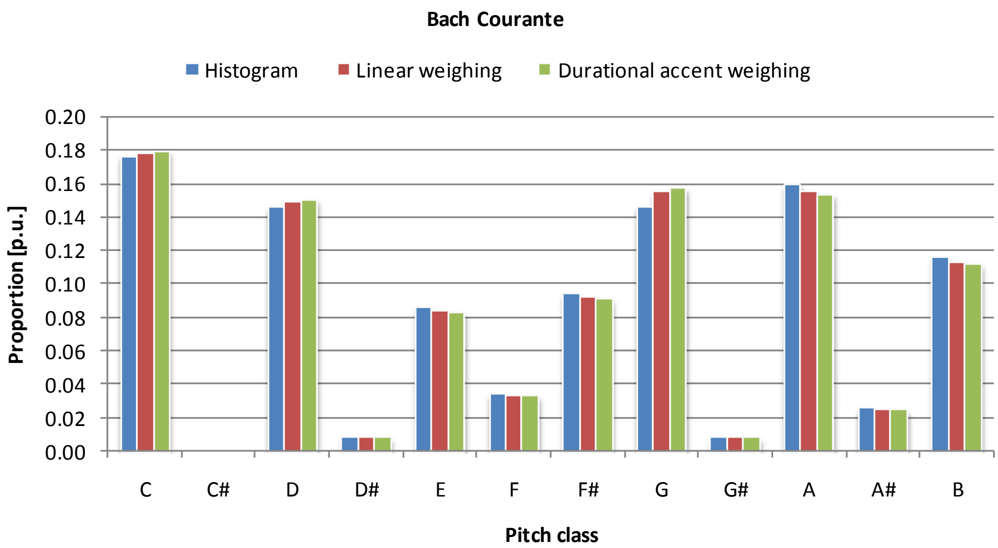


Fig. 5.13 Normalised distributions of pitch classes for the Courante from Bach's Cello Suite in C major: Comparison of unweighted, linear durational weighing and durational accent weighing ($\tau = 0.5$ and $I_a = 2.0$***).***

5.2.4 Gavotte from Bach's French Suite No. 5 in G major (BWV 816)

5.2.4.1 Distributions of note durations

Fig. 5.14 shows the normalised distribution of note durations for the *Gavotte* from Bach's *French Suite No. 5 in G major (BWV 816)*. The results show that the score material contains a limited range of note durations, favouring eighth notes and quarter notes. The histogram is given in Appendix I.

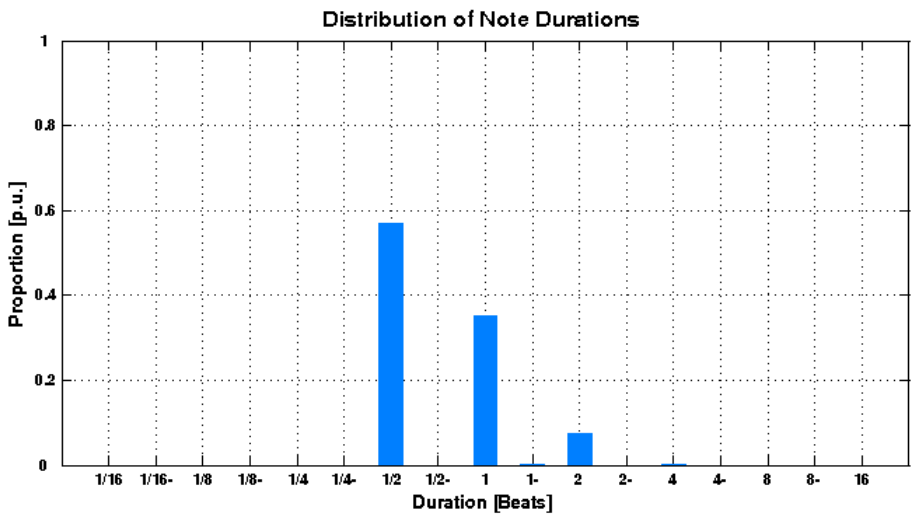


Fig. 5.14 Normalised distribution of note durations for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816).

5.2.4.2 Distributions of pitch classes

Fig. 5.15 shows the normalised distribution of pitch classes for the *Gavotte* from Bach's *French Suite No. 5 in G major (BWV 816)*. The results show clear indications of the home key G major and modulations to D major and E minor. The histogram is given in Appendix I.

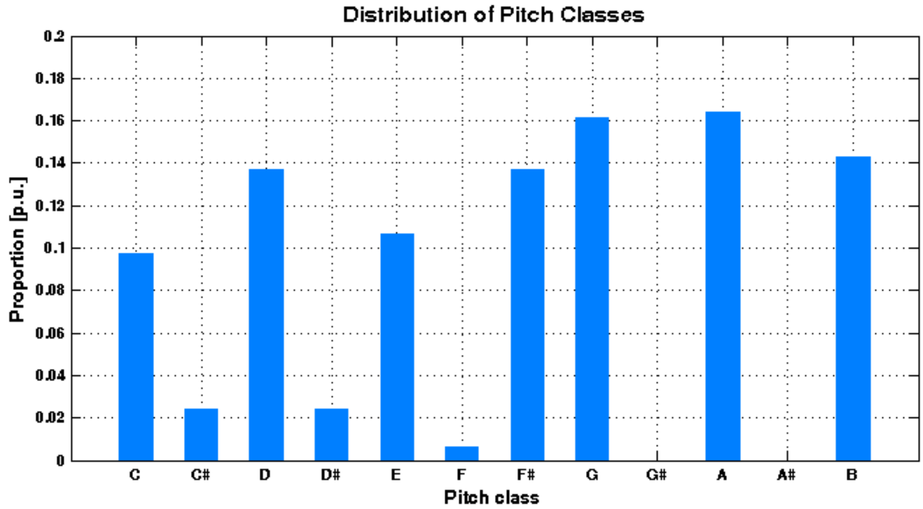


Fig. 5.15 Normalised unweighted distribution of pitch classes for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816).

Fig. 5.16 compares the distributions of pitch classes for the *Gavotte* for the unweighted, linear durational weighing and durational accent weighing models. The results show that the effect of durational weighing can be significant. For instance, in comparison with values given for the unweighted histogram, both durational weighing models reduce the value of pitch class E while increasing the value of pitch class A.

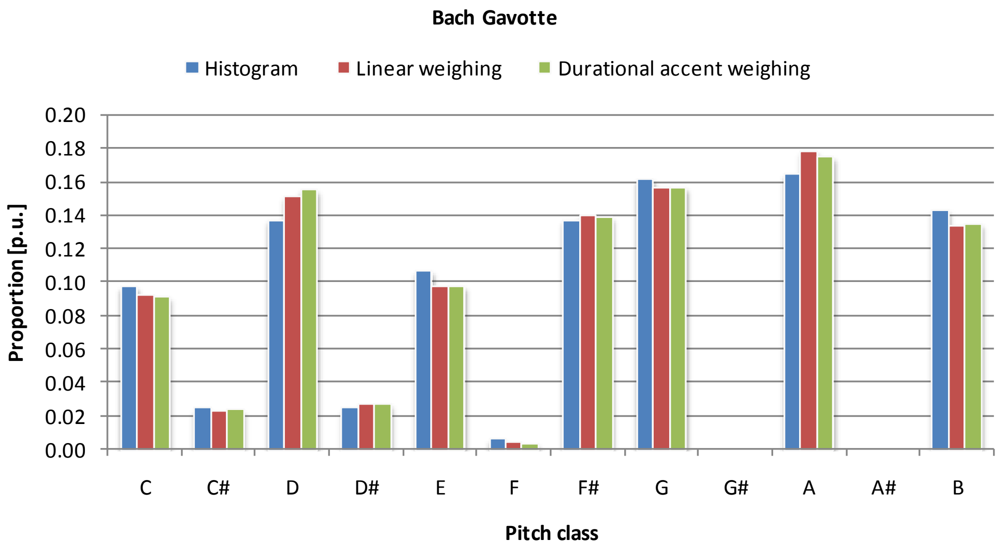


Fig. 5.16 Normalised pitch class distributions for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816): Comparison of unweighted, linear durational weighing and durational accent weighing ($\tau = 0.5$ and $I_a = 2.0$).

5.3 Performance evaluation of the Krumhansl and Schmuckler key-finding algorithm

5.3.1 Overview

The second research objective involves the question of whether a more optimal combination, compared to the combinations evaluated in literature, of the pitch class representation of the sample material and key profile template can be derived for the Krumhansl and Schmuckler key-finding algorithm. The Krumhansl and Schmuckler key-finding algorithm (Krumhansl, 1990), reviewed in section 2.7, have emerged as one of the most successful key-finding algorithms proposed to date. A number of variations of the original algorithm have been proposed and investigated in literature. These can be summarised as follows:

- *Alternative key profile templates:* Two key profile templates have been used, namely the original key profile based on tonality perception experiments proposed by Krumhansl and Kessler (1982) and the modified template proposed by Temperley (1999).
- *Alternative weighing schemes for the distribution of pitch classes:* Three durational weighing models have been applied, namely linear durational weighing as applied by Krumhansl and Schmuckler (Krumhansl, 1990), durational accent weighing as applied by Huron and Parncutt (1993) and the flat pitch class weighing model applied by Temperley (1999). The fourth logical option, namely weighing according to the properties of the histogram of occurrence of the pitch classes, has not been addressed in literature.
- *Alternative formulations of the key correlation formula:* Two key correlation formulas have been applied, namely the standard expression for correlation applied by Krumhansl and Schmuckler (Krumhansl, 1990) and the simplified formula applied by Temperley (1999).

The sample material used in the investigations reported in literature also differs from case to case. The objective of this part of the project is to use the software application to conduct a comprehensive comparative analysis of the performance of the Krumhansl and Schmuckler key-finding algorithm, allowing for all of the above variations for the sample material identified in the introduction of this chapter.

Fig. 5.17 shows a graphical representation of the case studies conducted. The diagram shows that the key-finding performance was evaluated allowing for a range of parameters pertaining to the key-finding algorithm, namely the score material, the weighing model applied for the distribution of pitch classes, the key profile template and the key correlation formula. Fig. 5.17 also shows the alternatives considered for each parameter. A fixed key resolution of 0.001% was used to emulate the maximum correlation principle applied by Krumhansl and Schmuckler.

Table 5.1 summarises the naming conventions adopted for presenting the results of the various case studies.

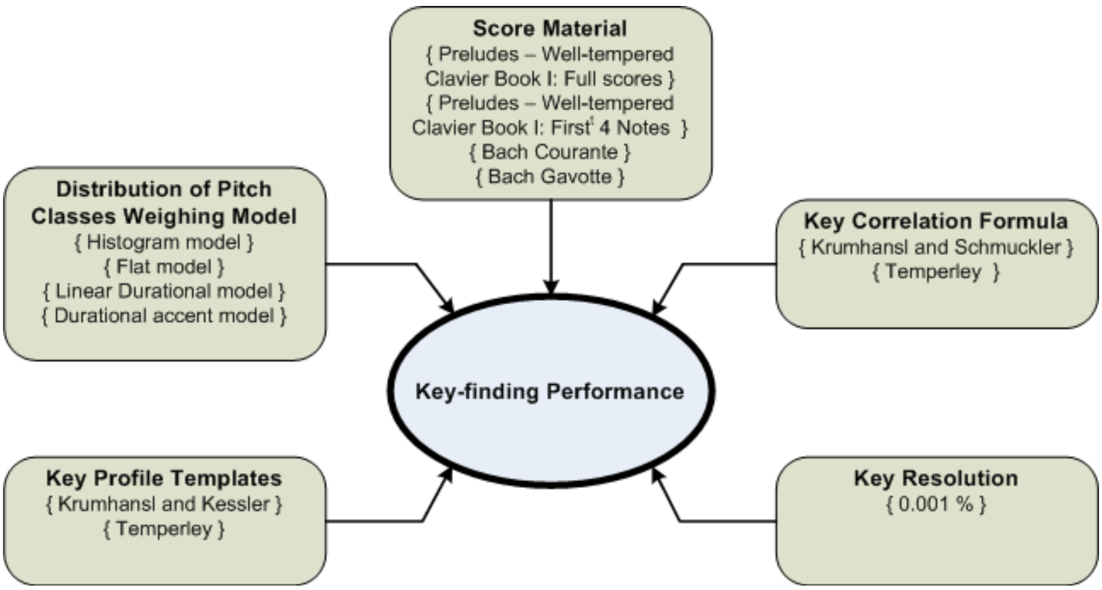


Fig. 5.17 Hierarchy of case studies conducted with the Krumhansl and Schmuckler key-finding algorithm.

Table 5.1 Naming conventions adopted for the case studies conducted for the Krumhansl and Schmuckler key-finding algorithm.

Key-profile	**Pitch classes weighing**			
	Histogram weighing model	**Flat weighing model**	**Linear durational weighing model**	**Durational accent weighing model**
Krumhansl & Kessler	KKI	KKII	KKIII	KKIV
Temperley	TI	TII	TIII	TIV

Only the results obtained with the Krumhansl and Schmuckler key correlation formula will be presented, because the investigations revealed that the choice of key correlation formula does not have a significant effect on the key-finding performance. This is expected, as the difference between the expressions is mostly of a scaling or normalization nature. The choice of key correlation formula does, however, have some computational implications, which will be addressed later.

In order to determine the key-finding performance for cases where the sample material is assigned multiple reference keys and/or where the key-finding algorithm delivers multiple key estimations, the following strategy is adopted:

- Estimated keys that are not represented in the set of reference keys for the sample material receives a score of zero.
- An estimated key that is represented in the set of reference keys for the sample material receives a score of $1/N$, where N denotes the number of keys in the set of estimated keys.

In mathematical terms, this gives rise to the following relationship for the performance figure P in percentage:

$$P = \frac{1}{100} \sum_{i=1}^{N_P} \left(\frac{1}{\left| K_i^E \right|} \left| \left\{ k^E \middle| k^E \in K_i^E , k^E \in K_i^R \right\} \right| \right) \tag{5.1}$$

where K_i^R and K_i^E denote the sets of reference keys and estimated keys respectively for the ith partition of the sample material and N_P denotes the total number of partitions of the sample material. For the test cases, $N_P = 24$ for the preludes of Bach's *Well-tempered Clavier Book I*, $N_P = 40$ for the first 40 measures of the *Courante* from Bach's *Cello Suite in C major* and $N_P = 25$ for the 25 measures of the *Gavotte* from Bach's French *Suite No. 5 in G major (BWV 816)*.

5.3.2 Key-finding case study results

Table 5.2 and Table 5.3 summarise the key-finding performance of the Krumhansl and Schmuckler key-finding algorithm for the full scores and first four notes respectively of the 24 preludes of Bach's *Well-tempered Clavier Book I.* The reference keys are the key signatures designated in the original scores. The reference keys and estimated keys for these test cases are tabulated in Appendix J.

Table 5.4 summarises the key-finding performance of the Krumhansl and Schmuckler key-finding algorithm for the first 40 measures of the *Courante* from Bach's *Cello Suite in C major*. The reference keys are derived from the keys proposed by Temperley (1999). The reference keys and estimated keys for this test case are tabulated in Appendix J.

Table 5.5 summarises the key-finding performance of the Krumhansl and Schmuckler key-finding algorithm for the *Gavotte* from Bach's *French Suite No. 5 in G major (BWV 816)*. The reference keys are the keys proposed by Temperley (1999). The reference keys and estimated keys for this test case are tabulated in Appendix J.

Table 5.2 Performance of the Krumhansl and Schmuckler key-finding algorithm for the full scores of the 24 preludes of Bach's Well-tempered Clavier Book I.

Key-profile	Pitch classes weighing			
	Histogram	**Flat**	**Linear durational**	**Durational accent** $(\tau = 0.5, I_a = 2.0)$
Krumhansl & Kessler	83.33%	7.64%	91.67%	95.83%
Temperley	75.00%	3.47%	91.67%	83.33%

Table 5.3 Performance of the Krumhansl and Schmuckler key-finding algorithm for the first four notes of the 24 preludes of Bach's Well-tempered Clavier Book I.

Key-profile	Pitch classes weighing			
	Histogram	**Flat**	**Linear durational**	**Durational accent** $(\tau = 0.5, I_a = 2.0)$
Krumhansl & Kessler	91.67%	91.67%	91.67%	91.67%
Temperley	93.75%	87.50%	97.92%	97.92%

Table 5.4 Performance of the Krumhansl and Schmuckler key-finding algorithm for the Courante from Bach's Cello Suite in C major.

Key-profile	Pitch classes weighing			
	Histogram	**Flat**	**Linear durational**	**Durational accent** $(\tau = 0.5, I_a = 2.0)$
Krumhansl & Kessler	65.00%	62.50%	65.00%	65.00%
Temperley	85.00%	100.00%	83.75%	83.75%

Table 5.5 Performance of the Krumhansl and Schmuckler key-finding algorithm for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816).

Key-profile	Pitch classes weighing			
	Histogram	**Flat**	**Linear durational**	**Durational accent** $(\tau = 0.5, I_a = 2.0)$
Krumhansl & Kessler	48.00%	64.00%	40.00%	40.00%
Temperley	76.00%	78.00%	74.00%	68.00%

Fig. 5.18 and Fig. 5.19 present comparisons of the key-finding performance of the Krumhansl and Kessler key profile for the different pitch class weighing models for the four test cases.

The results shown in Fig. 5.19 give rise to the following conclusions:

- For the full scores of the *Well-tempered Clavier*, the durational accent weighing model achieves the best performance figure in this group of 95.83%. Good results are also achieved with the histogram and linear durational pitch class weighing models, while a very poor result of 7.64% is achieved with the flat pitch weighing model.
- For the first four notes of the *Well-tempered Clavier,* all weighing models achieve a performance figure of 91.67%.
- For the *Courante*, all weighing models achieve average performance figures, with the flat weighing model performing marginally weaker compared to the rest.
- For the *Gavotte*, all weighing models achieve relatively poor results. The flat weighing model outperforms the rest moderately.

The comparisons presented in Fig. 5.19 indicate that the performance of all weighing models vary widely depending on the nature of the sample material. This is a clear indication that, at least for the Krumhansl and Kessler key profile, the issue of an optimal weighing model remains unresolved. The linear durational and durational accent models perform very similarly, with the durational accent model showing a slight advantage for the full scores of the *Well-tempered Clavier*. The poor performance of the flat weighing model for the full scores of the *Well-tempered Clavier* will be revisited later in this evaluation.

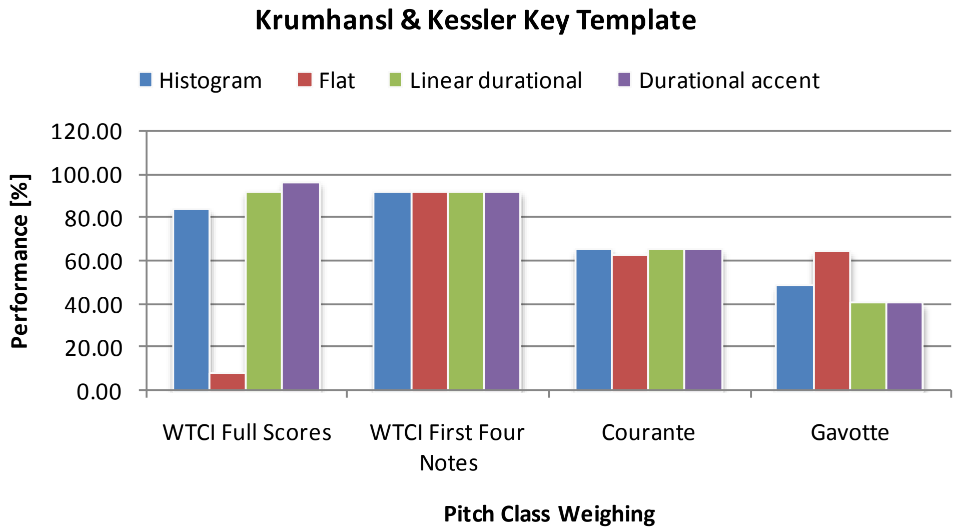


Fig. 5.18 Comparison of the key-finding performance of the Krumhansl and Kessler key profile grouped according to the test score.

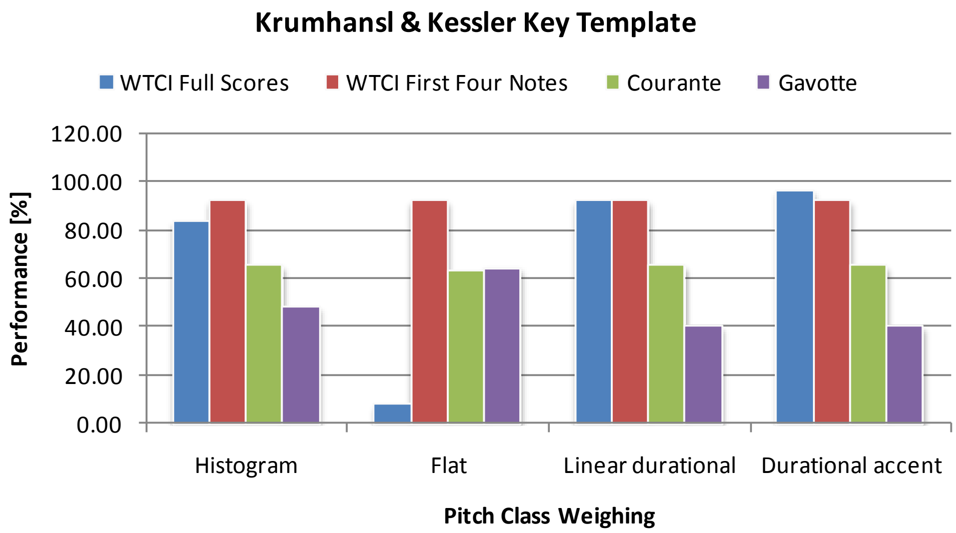


Fig. 5.19 Comparison of the key-finding performance of the Krumhansl and Kessler key profile grouped according to the weighing model.

Fig. 5.20 and Fig. 5.21 present comparisons of the key-finding performance of the Temperley key profile for the different pitch class weighing models for the four test cases. The results shown in Fig. 5.20 give rise to the following conclusions:

- For the full scores of the *Well-tempered Clavier*, the linear durational weighing model achieves a good performance figure of 91.67%, which is the same figure achieved with the Krumhansl and Kessler key profile. As in the case with the Krumhansl and Kessler key profile, a very poor performance figure of 3.47% is achieved with the flat pitch class weighing model.
- For the first four notes of the *Well-tempered Clavier*, all weighing models achieve good results. The linear durational and durational accent weighing models achieve the overall best performance figure of 97.92%.
- For the *Courante*, the flat weighing model outperforms the other models considerably with the overall best performance figure of 100%.

- For the *Gavotte*, all models achieve average results. The flat weighing model achieves the overall best performance figure of 78%.

The comparisons presented in Fig. 5.21 indicate that, with exception of the result for the flat weighing model for the full scores of the *Well-tempered Clavier*, the performance the weighing models is less dependent on the sample material compared to the case for the Krumhansl and Kessler key profile shown in Fig. 5.19. As in the case of the Krumhansl and Kessler key profile, no single weighing model achieves the best performance for all test cases.

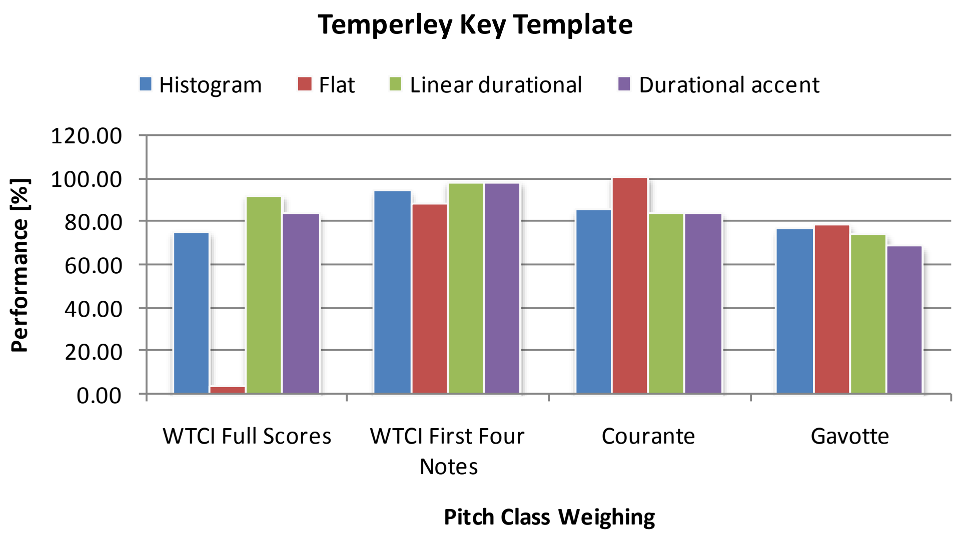


Fig. 5.20 Comparison of the key-finding performance of the Temperley key profile grouped according to the test score.

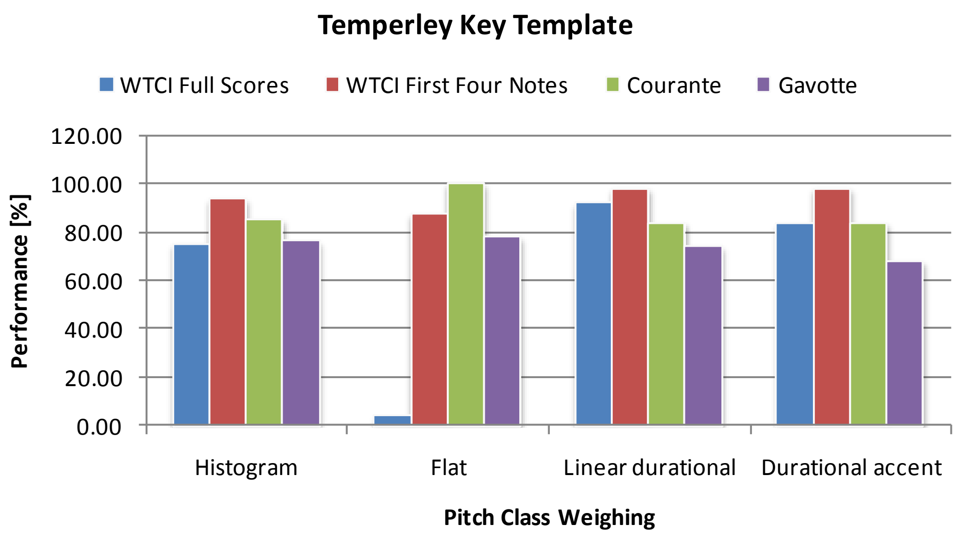


Fig. 5.21 Comparison of the key-finding performance of the Temperley key profile grouped according to the weighing model.

Fig. 5.22, Fig. 5.24, Fig. 5.25 and Fig. 5.26 compare the performance of the various pitch class weighing models for the different key profiles and test cases.

The results shown in Fig. 5.22 confirm that the flat weighing model proposed by Temperley delivers a very poor performance for both key profile templates for the full scores of the *Well-tempered Clavier*. The reason for this is that each prelude contains a large number of note events with a fair degree of tonic diversity. The flat weighing model is entirely unsuitable for this type of sample material because the pitch class distribution invariably tends to reflect equal weights for all pitch classes. This is supported by the detailed key estimation results presented in Appendix J, which shows that a high number of keys are identified as possible candidates by the key-estimation algorithm. Fig. 5.23 shows how the number of pitch classes represented in the partition increases as an expanding window includes an increasing number of measures of the *Gavotte from Bach's French Suite No. 5 in G major (BWV 816)*.

The flat pitch class weighing model does, however, also deliver the best results for the *Courante* and *Gavotte* (with the Temperley key profile) and matches the performance of the durational models for the first four notes of the *Well-tempered Clavier* (with the Krumhansl and Kessler key profile).

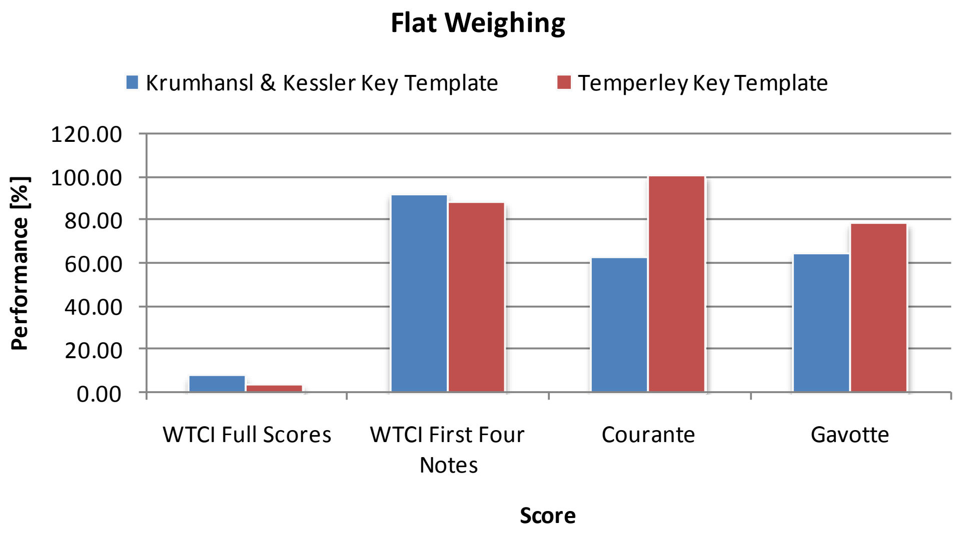


Fig. 5.22 Key-finding performance of the flat pitch class weighing model for the Krumhansl and Kessler and the Temperley key templates for all test scores.

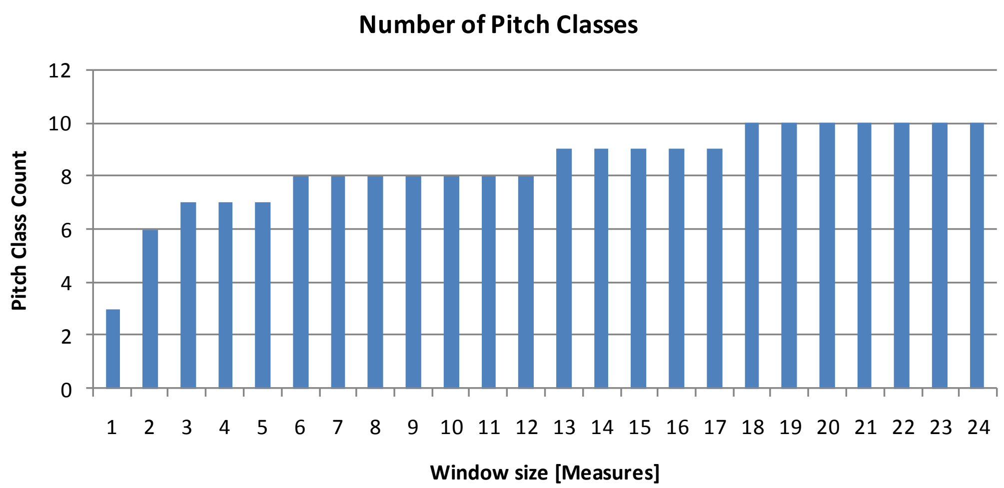


Fig. 5.23 Number of pitch classes as a function of window size in measures for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816).

As expected, the comparisons shown in Fig. 5.24, Fig. 5.25 and Fig. 5.26 indicate that the histogram, linear durational and durational accent models perform similarly, and are particularly suited for sample material with a large number of tonally diverse note events. The linear durational weighing model, on average, outperforms the other two weighing models. The results show no clear evidence that the durational accent model, for the parameters used in the investigation, improves the overall performance figures significantly for the test cases studied. This, however, may change if more appropriate values for the saturation duration and accent index are used. This aspect was not investigated in this project.

The comparative study for the different key profiles and pitch class weighing models does not deliver a clear winning combination. Overall, the results give rise to the following observations:

- The performance of the various weighing models depends on the both the nature of the sample material as well as the choice of key template. The *Courante* and *Gavotte* are not optimal choices of sample material for comparing the relative merits of the flat weighing model versus the durational weighing models. As shown in Fig. 5.4 and Fig. 5.14, the distributions of note durations for these pieces reflect very limited variation. This tends to diminish the effects of durational weighing.

  Given the fact that the *Courante* and *Gavotte* are analysed on a measure by measure basis while only four notes are analysed for each of the preludes, it can be concluded that the flat weighing model may be well suited for this type of analysis, e.g. for tracking key modulations using short excerpts of sample material. In the more general case, depending on the nature of the sample material, the flat key profile always introduces the risk of poor performance if the partition size is too big.

- The performance of the different key profiles similarly depends on the choice of pitch class weighing model as well as the nature of the test material. The fact that there are significant differences between the weights assigned to the individual pitch classes for the two key profiles, suggests that it is necessary to re-evaluate the current key profiles with the view to first derive a more optimal key profile template before revisiting the issue of pitch class weighing.

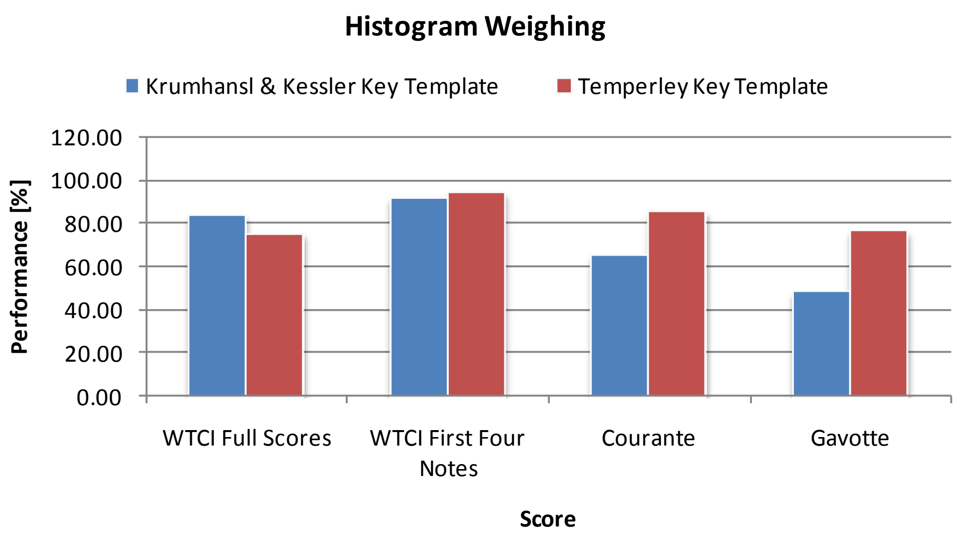


Fig. 5.24 Key-finding performance of the histogram pitch class weighing model for the Krumhansl and Kessler and the Temperley key templates for all test scores.

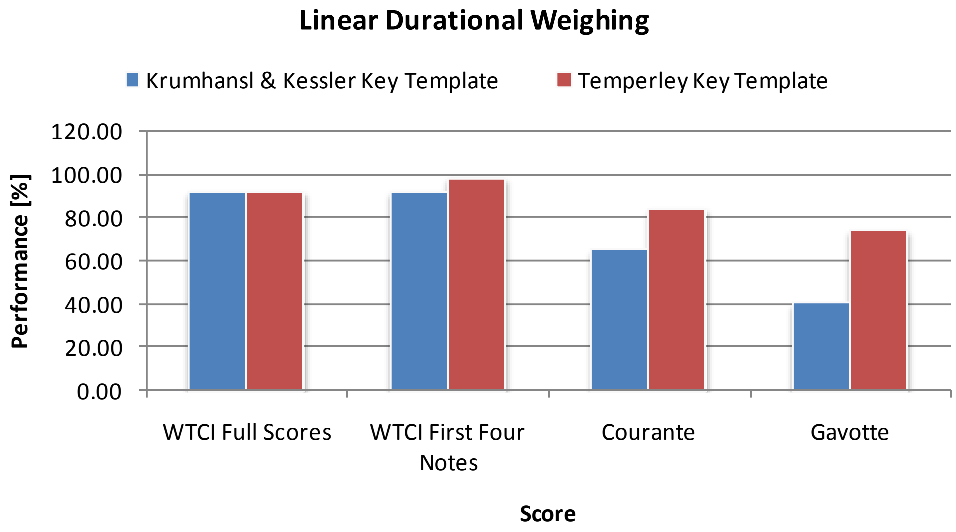


Fig. 5.25 Key-finding performance of the linear durational pitch class weighing model for the Krumhansl and Kessler and the Temperley key templates for all test scores.

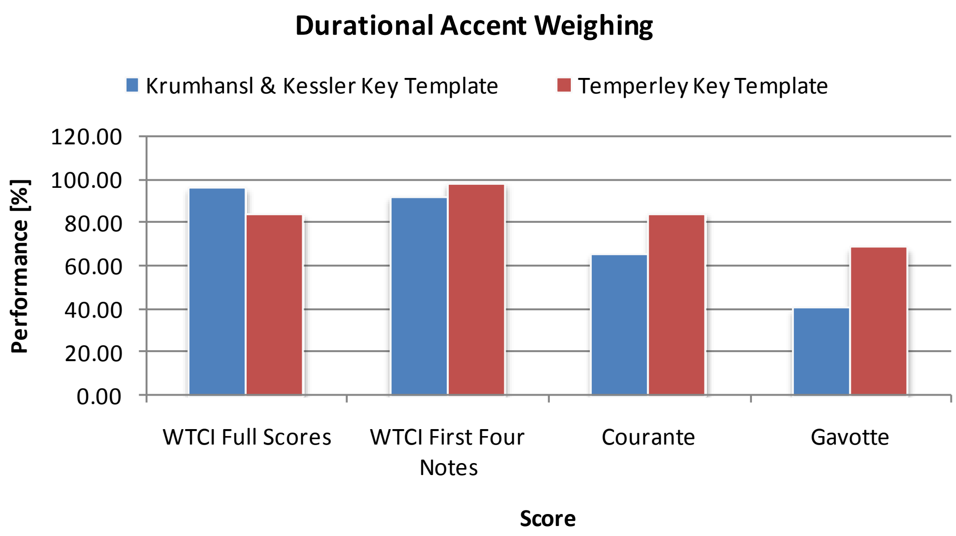


Fig. 5.26 Key-finding performance of the durational accent pitch class weighing model for the Krumhansl and Kessler and the Temperley key templates for all test scores.

5.3.3 Key profile optimization using direct search methods

5.3.3.1 Overview

The third research objective involves the question of whether a more optimal key profile template can be derived for the Krumhansl and Schmuckler key-finding algorithm, using parameter estimation techniques such as direct search methods. Two key profile templates have been proposed in literature, namely the profile derived from tonality perception experiments by Krumhansl and Kessler (1982) and the profile derived from music theoretical considerations by Temperley (1999). As shown in the literature review and in the previous section, these profiles perform well for some cases, but fail to deliver in other cases. In order to address the above research question, an extensive investigation was conducted with the view to derive a key profile that performs more consistently for a diverse body of sample material.

Fig. 5.27 shows a graphical representation of the case studies conducted for this purpose. The diagram shows that the key profile estimations were conducted for a range of parameter values pertaining to the key-finding algorithm, namely the score material, the weighing model applied for the distribution of pitch classes and the key resolution. With regards to the pattern search algorithm, the case studies allowed for permutations of the key profile constraints, cost function and pattern search polling method. Fig. 5.27 also shows the alternatives considered for each parameter. Equal weights of unity are assigned to the tonic of both the major and minor modes for all test cases, which implies that coefficients were estimated for the remaining 22 pitch classes.

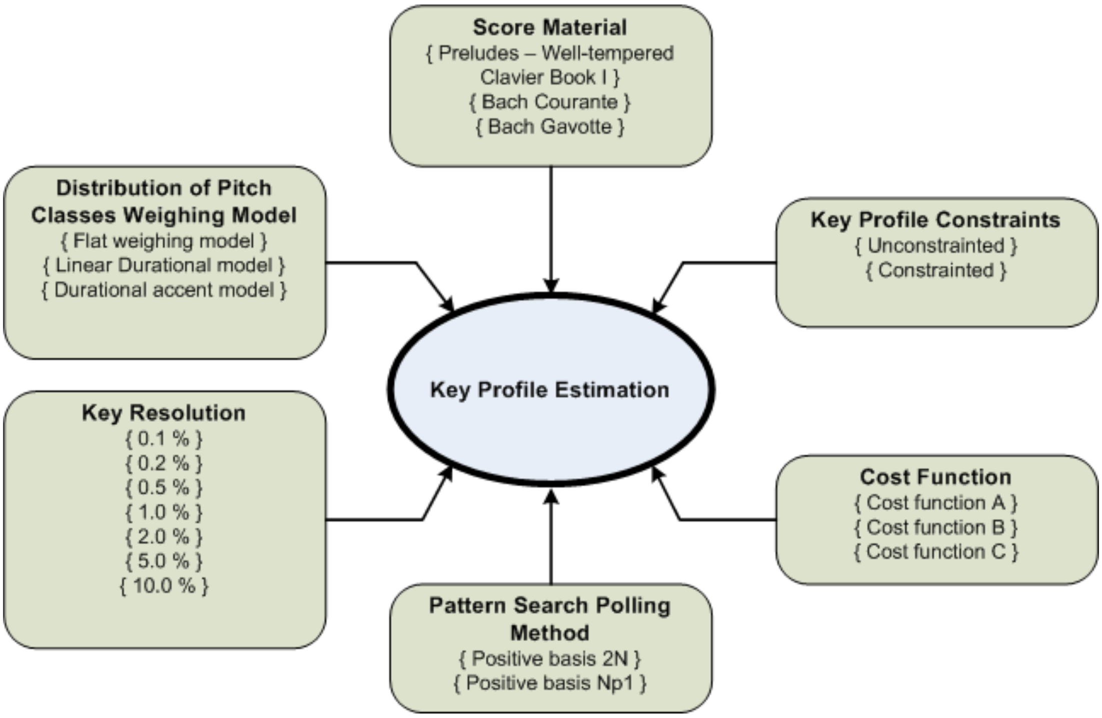


Fig. 5.27 Hierarchy of case studies conducted for the key profile estimation investigation.

The following observations apply for the key profile estimation parameter options used in the study:

- *Score material*: The investigation was conducted for the same test cases used for the key-finding investigation, namely the full scores and the first four notes respectively of the preludes of Bach's *Well-tempered Clavier Book I*, the first 40 measures of the *Courante* from Bach's *Cello Suite in C major* and the *Gavotte* from Bach's *French Suite No. 5 in G major (BWV 816).*
- *Key-finding algorithm parameters*:
  - *Distribution of pitch classes weighing model*: Three options were tested for the weighing model applied for the distribution of pitch classes, namely the flat model proposed by Temperley (1999), the linear durational model proposed by Krumhansl (1990) and the durational accent model proposed by Huron and Parncutt (1993). The durational accent model applied the default parameter values proposed by Parncutt, i.e. $\tau = 0.5$ and $I_a = 2.0$, for all case studies.
  - *Key resolution*: The study revealed that the key resolution used in the key-finding algorithm is an important parameter for the key profile estimation process. This parameter controls the extent to which multiple keys can be assigned for a particular partition and as such has an effect on the behaviour of the cost function. A wide range of options were considered, ranging from 0.1% to 10% as shown in Fig. 5.27.
- *Pattern* search parameters:

- *Key profile constraints*: The investigation considered both unconstrained and constrained optimization. The unconstrained case allows the optimization algorithm the freedom to assign any weight between the lower and upper bounds of zero and unity respective to the individual pitch classes. As such, the results afford the opportunity to determine whether the optimization process confirms the pitch classes hierarchy derived by Krumhansl and Kessler (1982) from tonality perception experiments, or the similar hierarchy proposed by Temperley (1999) on the basis on classical music theoretical principles.

  In the constrained case, the constraints were designed to loosely impose the hierarchical relationships represented in the key profiles proposed by Krumhansl and Kessler (1982) and Temperley (1999). For the major mode, all degrees except the leading tone are constrained to be of an equal or lower rating compared to the mediant and dominant. No relative constraints are imposed on the mediant and dominant. For the minor mode, all degrees except the leading tone are constrained to be of an equal or lower rating compared to the minor second and dominant. No relative constraints are imposed on the minor third and dominant. The relationships between the diatonic and other non-diatonic degrees of the scale were unconstrained for both modes.
- *Cost function*: Three cost functions were considered, namely the functions designated A, B and C in the mathematical formulations presented in section 4.5.3. Each of these has particular strengths and weaknesses, as will be evident from the results presented in the next section.
- *Pattern search polling method*: The two polling methods identified in the section 2.11.2 of the literature review were both tested.

The estimation process is affected by a number of additional parameters pertaining to the pattern search algorithm. As these do not impact directly on the core research objectives of the study, suitable values were derived based on computational considerations and some experimentation. The most important of these are listed in Table 5.6.

Given the range of permutations implied by the input parameter options considered in the investigation, the study gives rise to a substantial body of numerical and graphical results. These include the distributions of pitch classes, key correlation tables, estimated key vectors and major and minor mode key profile coefficients for each test case. Due to space considerations, only a limited set of these results can be reproduced in this document. Therefore, the results presented in the remainder of this section are limited to high-level summaries of the output generated by the software application. Detail will only be presented where required to support a particular research conclusion.

Table 5.6 Pattern search parameter values used in the key profile estimation cases studies.

Parameter	Description	Value
TolMesh	Tolerance on mesh size	1.0e-003
MeshExpansion	Mesh expansion factor	0.2
MeshContraction	Mesh contraction factor	10.0
TolX	Tolerance on variable, i.e. minimum distance between the point found in two consecutive iterations.	1.0e-006
TolFun	Tolerance on objective function value, i.e. minimum change in the objective function in two consecutive iterations.	1.0e-006
TolCon	Tolerance on constraints.	1.0e-006
MaxIter	Maximum number of iterations.	800
MaxFunEvals	Total number of objective function evaluations.	44000

5.3.3.2 Key profile estimation results for the full scores of the 24 preludes of Bach's Well-tempered Clavier Book I

Cost function C is not applicable for the full scores because there are no successive partitions. Table 5.7 and Table 5.8 summarise the unconstrained and constrained key profile optimization performances respectively for the full scores of the 24 preludes of Bach's *Well-tempered Clavier Book I*.

Table 5.7 Unconstrained key profile estimation performance percentage for the full scores of the 24 preludes of Bach's Well-tempered Clavier Book I.

Weighing	Key resolution [%]	Cost function A		Cost function B		Cost function C	
		2N	Np1	2N	Np1	2N	Np1
Flat	0.1	11.81	11.81	11.81	11.81	-	-
	1.0	100.00	11.81	11.81	11.81	-	-
	10.0	11.81	11.81	11.81	11.81	-	-
Linear durational	0.1	100.00	100.00	100.00	100.00	-	-
	1.0	50.00	100.00	50.00	50.00	-	-
	10.0	100.00	100.00	100.00	100.00	-	-
Durational accent	0.1	45.83	45.83	100.00	50.00	-	-
	1.0	45.83	45.83	100.00	50.00	-	-
	10.0	47.92	47.92	50.00	47.92	-	-

Table 5.8 Constrained key profile estimation performance percentage for the full scores of the 24 preludes of Bach's Well-tempered Clavier Book I.

Weighing	Key resolution [%]	Cost function A		Cost function B		Cost function C	
		2N	Np1	2N	Np1	2N	Np1
Flat	0.1	11.81	11.81.81	11.8111.81	11.81 11.81	-	-
	1.0	11.81	11.81.81	11.8111.81	11.81 11.81	-	-
	10.0	11.81	11.81	11.81	11.81	-	-
Linear durational	0.1	91.67	100.00	100.00	100.00	-	-
	1.0	100.00	100.00	100.00	100.00	-	-
	10.0	95.83	95.83	100.00	100.00	-	-
Durational accent	0.1	91.67	100.00	100.00	100.00	-	-
	1.0	95.83	100.00	100.00	100.00	-	-
	10.0	90.97	97.92	97.92	100.00	-	-

Fig. 5.28, Fig. 5.29 and Fig. 5.30 compare the key profile optimization performances for the full scores of the 24 preludes of Bach's *Well-tempered Clavier Book I*, for key resolutions of 0.1%, 1.0% and 10.0% respectively.

The following observations apply for the results:

- *Unconstrained case*: The flat weighing model performs poorly for all key resolutions and for both cost functions for the reasons outlined in section 5.3.2. The linear durational weighing model achieves a 100% score for all key resolutions and cost functions, except for a key resolution of 1.0% with cost function B. The durational accent weighing model achieves a 100% score only for a key resolution of 0.1% and 1.0% with cost function B. The results for the durational weighing models are inconsistent, as similar performances are expected for both models. The poorer performance of the durational accent model is therefore attributed to the effects of local minima on the search function rather than an inherent deficiency of the algorithm and model options.
- *Constrained case*: As for the unconstrained case, the flat weighing model performs poorly for all key resolutions and for both cost functions for the reasons outlined in section 5.3.2. The linear durational weighing model achieves a 100% score for all key resolutions except for a key resolution of 10.0% with cost function A. The durational accent weighing model achieves a similar result, which is much improved from the result achieved for the unconstrained case.

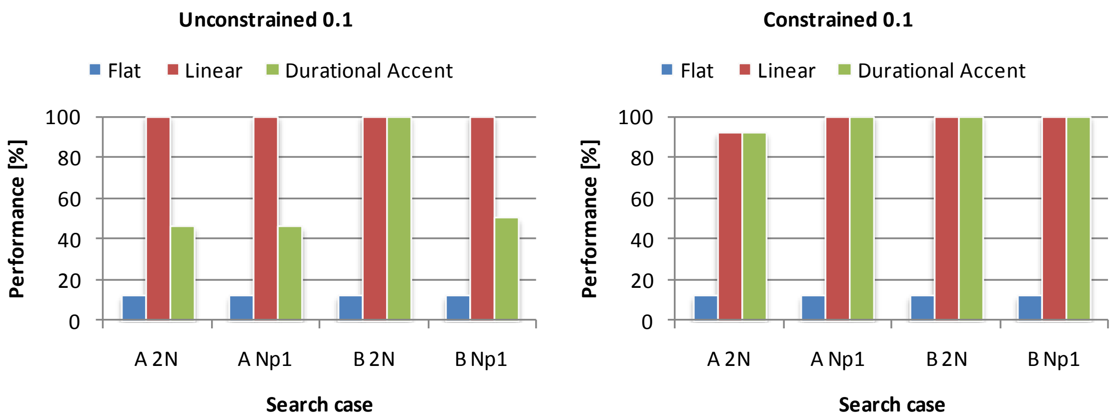


Fig. 5.28 Key profile estimation performances for a key resolution of 0.1% for the full scores of the 24 preludes of Bach's Well-tempered Clavier Book I.

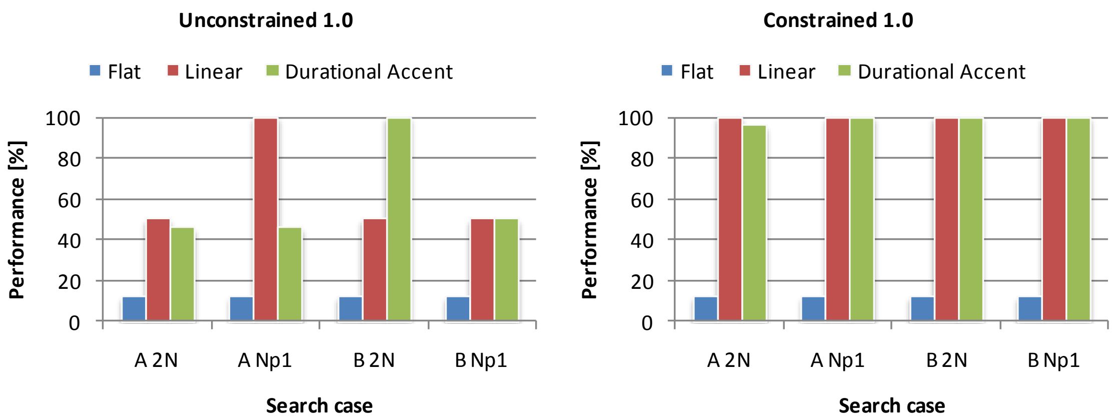


Fig. 5.29 Key profile estimation performances for a key resolution of 1.0% for the full scores of the 24 preludes of Bach's Well-tempered Clavier Book I.

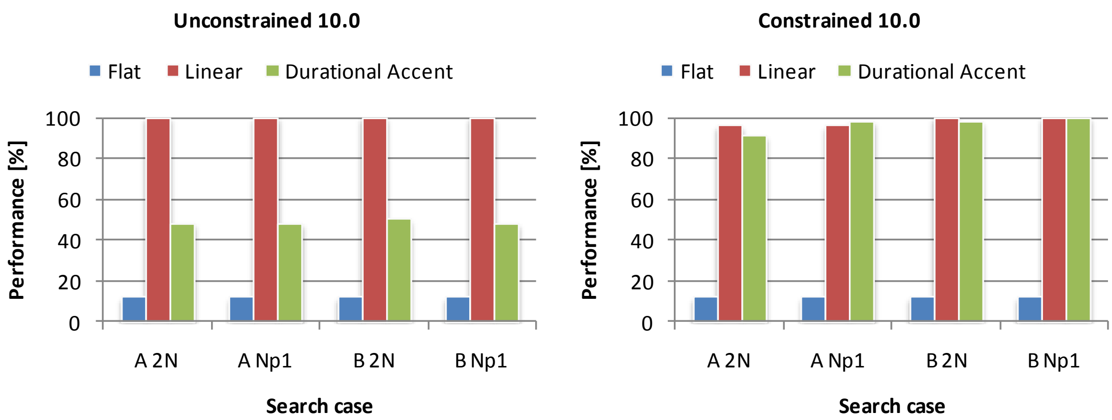


Fig. 5.30 Key profile estimation performances for a key resolution of 10% for the full scores of the 24 preludes of Bach's Well-tempered Clavier Book I.

By comparing the results given in Table 5.7 and Table 5.8 with the results given in Table 5.2, where none of the cases achieved a score of 100%, it is concluded that optimization exercise derived a more optimum key profile for this type of material. It is further evident that the application of constraints assists the pattern search process, yielding much better consistency in finding an optimum solution. The fact that good performances are also obtained for the relatively high key resolution of 10.0% shows that the key-finding algorithm can achieve a high degree of robustness with the right key profile template. The results further confirm that the flat weighing model is unsuitable for key-finding applications involving large samples of tonally diverse material.

Fig. 5.31 and Fig. 5.32 present the unconstrained major and minor mode key profiles respectively, derived with the linear durational weighing model, for the high performance cases are identified by the shaded cells in Table 5.7. Table 5.9 and Table 5.10 show the corresponding correlations between these profiles. It is quite clear that the profiles exhibit a fairly wide degree of variation, slightly more so for the minor mode than for the major mode. This is confirmed by the relatively low correlation coefficients given in Table 5.9 and Table 5.10. As expected, the diatonic pitch classes show less diversity compared to the non-diatonic degrees. The dominant exhibits the least variation. As expected, the orders of the major third and minor third degrees are reversed in the major and minor mode hierarchies,. In general terms, the pitch class hierarchies confirm the hierarchy contained in the Krumhansl and Kessler profiles. The weights for the supertonic and the mediant in the major profile and the supertonic and the minor third in the minor profile are, however, much closer compared to the Krumhansl and Kessler profiles.

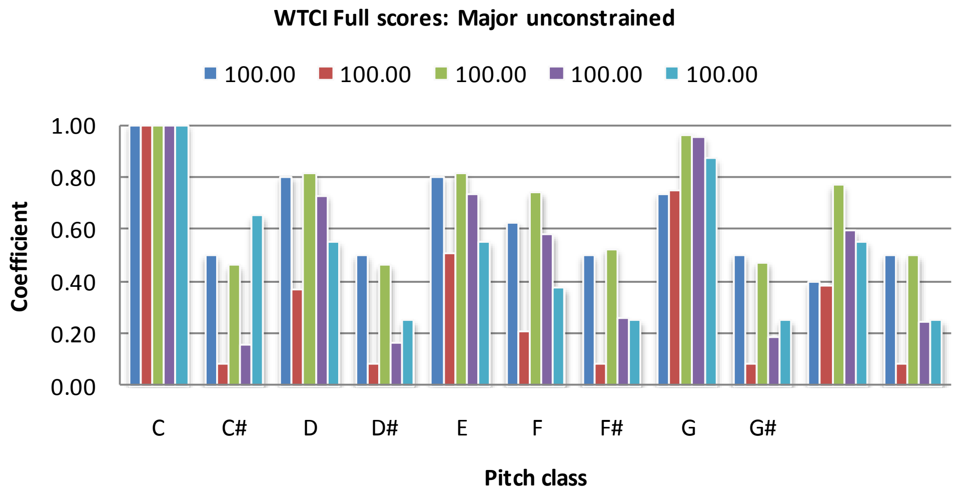


Fig. 5.31 Best scoring unconstrained major mode key profiles with linear durational weighing for the full scores of the 24 preludes of Bach's Well-tempered Clavier Book I.

Table 5.9 Correlations for the major mode key profiles shown in Fig. 5.31.

	10.0_A_2N	10.0_A_Np1	10.0_B_2N	10.0_B_Np1	1.0_A_Np1
10.0_A_2N	1.0000	0.8057	0.7698	0.7962	0.6914
10.0_A_Np1	0.8057	1.0000	0.9323	0.9338	0.8908
10.0_B_2N	0.7698	0.9323	1.0000	0.9979	0.8039
10.0_B_Np1	0.7962	0.9338	0.9979	1.0000	0.8016
1.0_A_Np1	0.6914	0.8908	0.8039	0.8016	1.0000

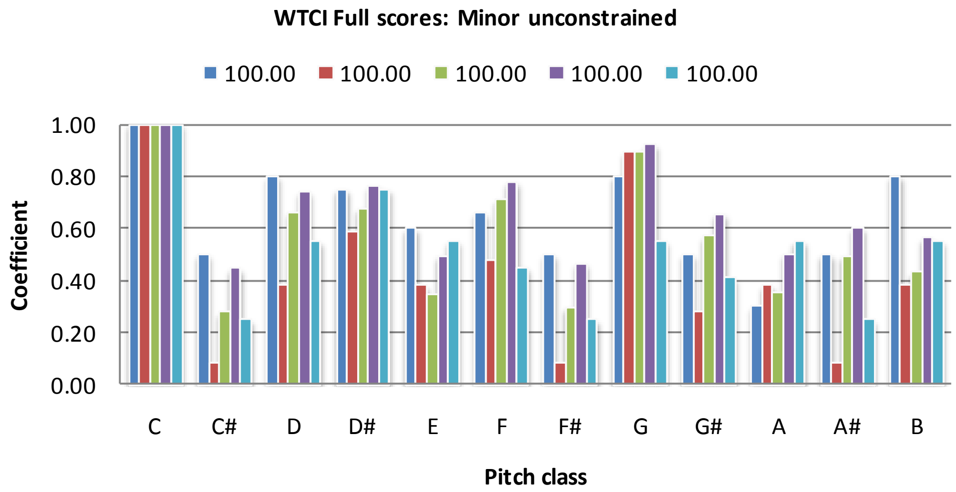


Fig. 5.32 Best scoring unconstrained minor mode key profiles with linear durational weighing for the full scores of the 24 preludes of Bach's Well-tempered Clavier Book I.

Table 5.10 Correlations for the minor mode key profiles shown in Fig. 5.32.

	10.0_A_2N	10.0_A_Np1	10.0_B_2N	10.0_B_Np1	1.0_A_Np1
10.0_A_2N	1.0000	0.7419	0.7683	0.7760	0.7069
10.0_A_Np1	0.7419	1.0000	0.8669	0.8743	0.8632
10.0_B_2N	0.7683	0.8669	1.0000	0.9994	0.6855
10.0_B_Np1	0.7760	0.8743	0.9994	1.0000	0.6935
1.0_A_Np1	0.7069	0.8632	0.6855	0.6935	1.0000

Fig. 5.33 and Fig. 5.34 present the constrained major and minor mode key profiles respectively, derived with the linear durational weighing model, for the high performance cases are identified by the shaded cells in Table 5.8. Table 5.11 and Table 5.12 show the corresponding correlations between these profiles. As for the unconstrained case, the correlation coefficients confirm that the profiles for the major mode exhibit less variation compared to the minor mode. As in the unconstrained case, the orders of the major third and minor third degrees are reversed in the major and minor mode hierarchies. The observations with regard to the relative weights of the supertonic, mediant and minor third given for the unconstrained case, also apply for the constrained case.

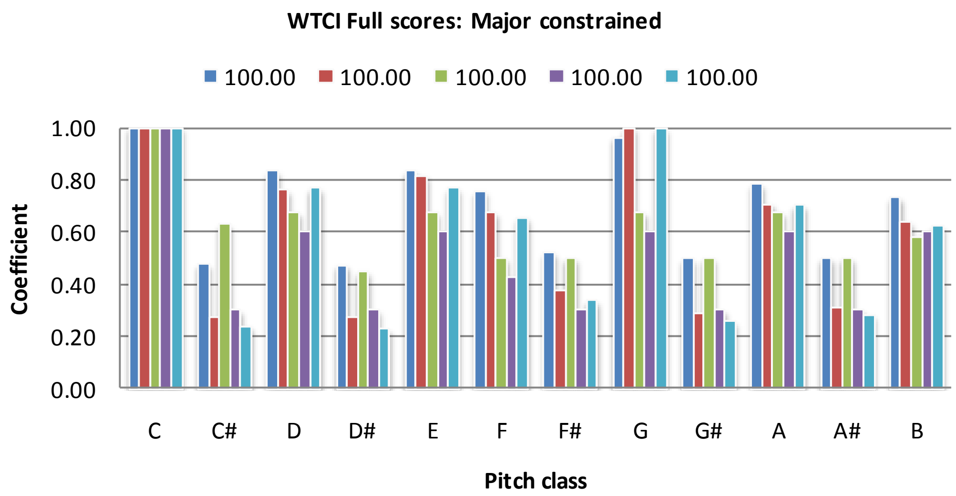


Fig. 5.33 Best scoring constrained major mode key profiles with linear durational weighing for the full scores of the 24 preludes of Bach's Well-tempered Clavier Book I.

Table 5.11 Correlations for the major mode key profiles shown in Fig. 5.33.

	10.0_B_2N	10.0_B_Np1	1.0_A_2N	1.0_A_Np1	1.0_B_2N
10.0_B_2N	1.0000	0.9976	0.7693	0.9045	0.9987
10.0_B_Np1	0.9976	1.0000	0.7496	0.8866	0.9989
1.0_A_2N	0.7693	0.7496	1.0000	0.9141	0.7534
1.0_A_Np1	0.9045	0.8866	0.9141	1.0000	0.8901
1.0_B_2N	0.9987	0.9989	0.7534	0.8901	1.0000

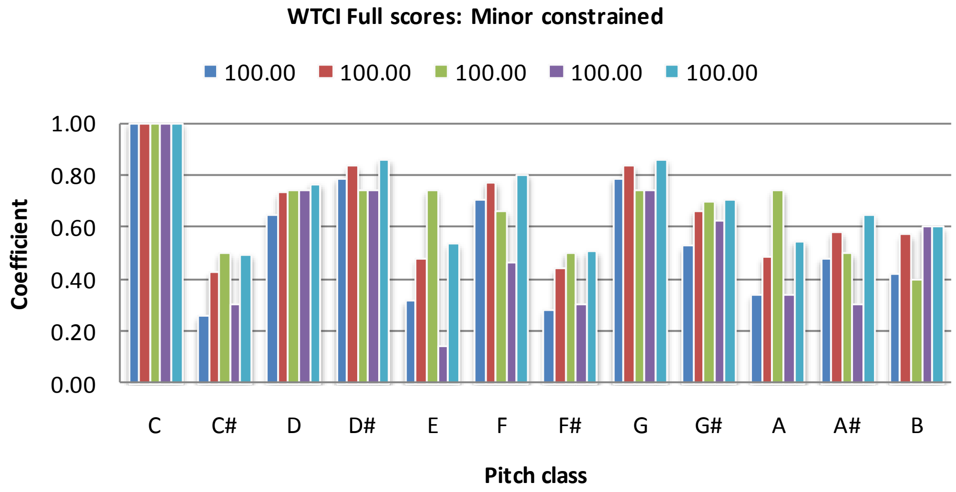


Fig. 5.34 Best scoring constrained minor mode key profiles with linear durational weighing for the full scores of the 24 preludes of Bach's Well-tempered Clavier Book I.

Table 5.12 Correlations for the minor mode key profiles shown in Fig. 5.34.

	10.0_B_2N	10.0_B_Np1	1.0_A_2N	1.0_A_Np1	1.0_B_2N
10.0_B_2N	1.0000	0.9978	0.6955	0.8723	0.9987
10.0_B_Np1	0.9978	1.0000	0.6921	0.8887	0.9986
1.0_A_2N	0.6955	0.6921	1.0000	0.5663	0.6954
1.0_A_Np1	0.8723	0.8887	0.5663	1.0000	0.8746
1.0_B_2N	0.9987	0.9986	0.6954	0.8746	1.0000

Overall, it is important to take cognizance of the fact that very high key-finding performances are achieved for the full scores of the preludes of the *Well-tempered Clavier Book I* with quite diverse key profile templates. This raises the question of whether this material is as well suited for evaluation of key-finding performance as is implied in literature. The general lack of key modulations suggests that it is not.

5.3.4 Key estimation results for the first four notes of the 24 preludes of Bach's Well-tempered Clavier Book I

Table 5.13 and Table 5.14 summarise the unconstrained and constrained key profile optimization performances respectively for the first four notes of Bach's *Well-tempered Clavier Book I*. Fig. 5.35, Fig. 5.36 and Fig. 5.37 compare the key profile optimization performances for the first four notes of the 24 preludes of Bach's *Well-tempered Clavier Book I*, for key resolutions of 0.1%, 1.0% and 10.0% respectively.

The following observations apply for the results presented:

- *Unconstrained case*: The flat weighing model achieves a 100% score for key resolutions of 0.1% and 1.0% with cost function B. A highest score of 97.92% is achieved for a key resolution of 10.0% with cost function B. The linear durational weighing model achieves a 100% score for key resolutions of 0.1% and 1.0% for all cost functions. A highest score of 95.83% is achieved for a key resolution of 10.0% with cost function B. The durational accent weighing model achieves a 100% score for all key resolutions except for a key resolution of 10.0% with cost function B, where a score of 97.92% is achieved. Overall, the durational weighing models outperform flat weighing model.

- *Constrained case*: The results for the constrained case are very similar to the results obtained for the unconstrained case.

By comparing the results given in Table 5.13 and Table 5.14 with the results given in Table 5.3, where none of the cases achieved a score of 100%, it is concluded that optimization exercise derived a more optimum key profile for this type of material. As for the previous test case, the fact that good results are also obtained for the relatively high key resolution of 10.0% shows that the key-finding algorithm can achieve a high degree of robustness with the right key profile template. The results further confirm that the flat weighing model performs well for key-finding applications involving small samples of material.

Table 5.13 Unconstrained key profile estimation performance percentage for the first four notes of the 24 preludes of Bach's Well-tempered Clavier Book I.

Weighing	Key resolution [%]	Cost function A		Cost function B		Cost function C	
		2N	Np1	2N	Np1	2N	Np1
Flat	0.1	45.83	45.83	100.00	100.00	-	-
	1.0	45.83	45.83	100.00	100.00	-	-
	10.0	45.83	45.83	97.92	97.22	-	-
Linear durational	0.1	100.00	100.00	100.00	100.00	-	-
	1.0	100.00	100.00	100.00	100.00	-	-
	10.0	91.67	85.42	95.83	93.75	-	-
Durational accent	0.1	100.00	100.00	100.00	100.00	-	-
	1.0	100.00	100.00	100.00	100.00	-	-
	10.0	100.00	93.75	97.92	97.92	-	-

Table 5.14 Constrained key profile estimation performance percentage for the first four notes of the 24 preludes of Bach's Well-tempered Clavier Book I.

Weighing	Key resolution [%]	Cost function A		Cost function B		Cost function C	
		2N	Np1	2N	Np1	2N	Np1
Flat	0.1	95.83	91.67	100.00	95.83	-	-
	1.0	95.83	91.67	100.00	95.83	-	-
	10.0	97.92	89.58	97.92	97.22	-	-
Linear durational	0.1	100.00	100.00	100.00	100.00	-	-
	1.0	100.00	100.00	100.00	100.00	-	-
	10.0	93.75	95.83	95.83	89.58	-	-
Durational accent	0.1	100.00	100.00	100.00	100.00	-	-
	1.0	100.00	100.00	100.00	100.00	-	-
	10.0	97.92	97.92	100.00	97.92	-	-

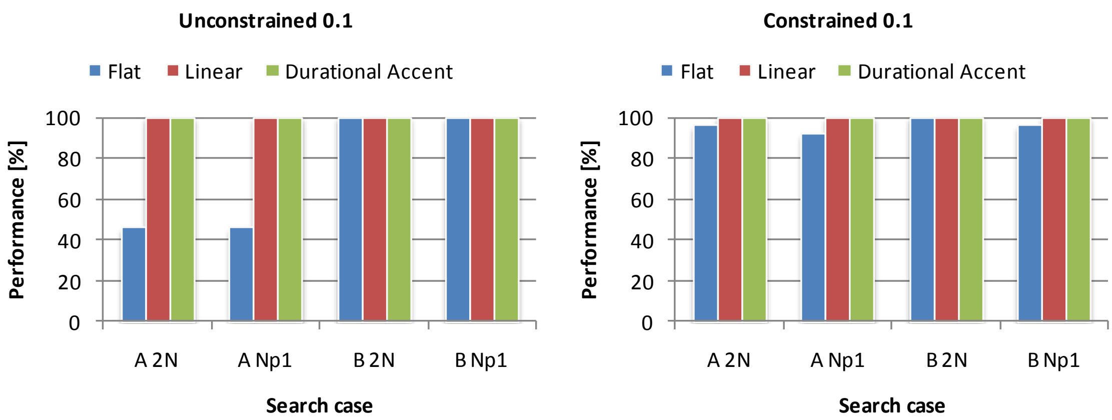


Fig. 5.35 Key profile estimation performances for a key resolution of 0.1% for the first four notes of the 24 preludes of Bach's Well-tempered Clavier Book I.

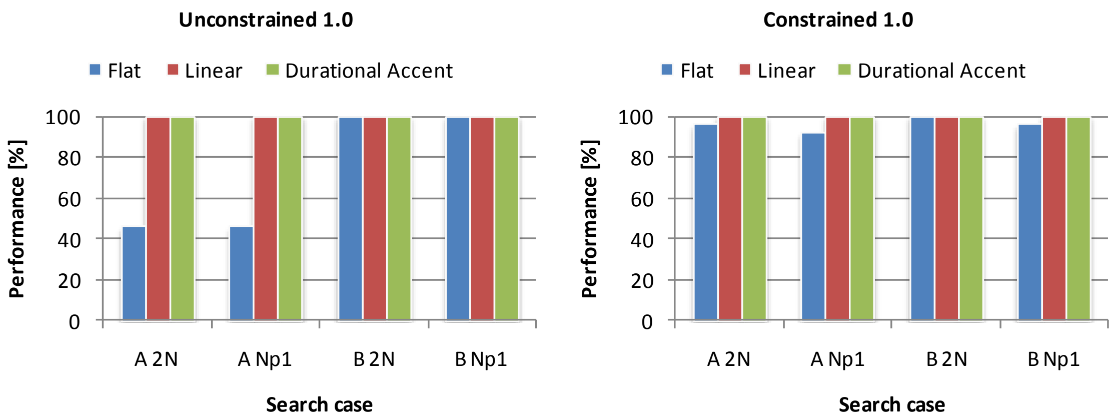


Fig. 5.36 Key profile estimation performances for a key resolution of 1.0% for the first four notes of the 24 preludes of Bach's Well-tempered Clavier Book I.

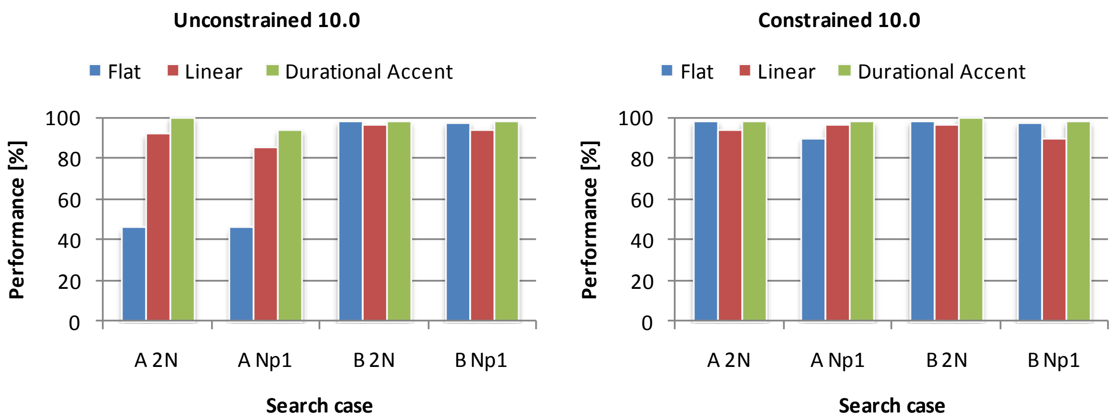


Fig. 5.37 Key profile estimation performances for a key resolution of 10.0% for the first four notes of the 24 preludes of Bach's Well-tempered Clavier Book I.

Fig. 5.38 and Fig. 5.39 present the unconstrained major and minor mode key profiles respectively, derived using the linear durational weighing model, for the high performance cases identified by the shaded cells in Table 5.13. Table 5.15 and Table 5.16 show the corresponding correlations between these profiles.

The results for the major mode show good correlation between profiles 1.0_A_2N and 0.1_A_2N and between profiles 1.0_B_2N 1.0_B_Np1 respectively. The results for the minor mode exhibit good correlation between profiles 1.0_A_2N and 0.1_A_2N and profiles 1.0_B_2N and 1.0_B_Np1 respectively. This gives rise to the important conclusion that, although the top scoring profiles may have different mean values, the underlying pitch class hierarchies are similar.

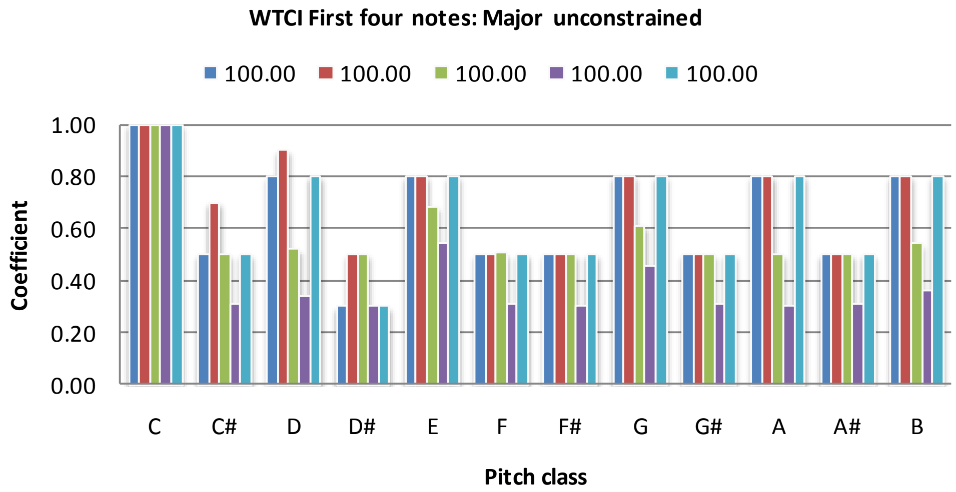


Fig. 5.38 Best scoring unconstrained major mode key profiles with linear durational weighing for the first four notes of the 24 preludes of Bach's Well-tempered Clavier Book I.

Table 5.15 Correlations for the unconstrained major mode key profiles shown in Fig. 5.38.

	1.0_A_2N	1.0_A_Np1	1.0_B_2N	1.0_B_Np1	0.1_A_2N
1.0_A_2N	1.0000	0.9243	0.6856	0.6832	1.0000
1.0_A_Np1	0.9243	1.0000	0.6594	0.6571	0.9243
1.0_B_2N	0.6856	0.6594	1.0000	0.9998	0.6856
1.0_B_Np1	0.6832	0.6571	0.9998	1.0000	0.6832
0.1_A_2N	1.0000	0.9243	0.6856	0.6832	1.0000

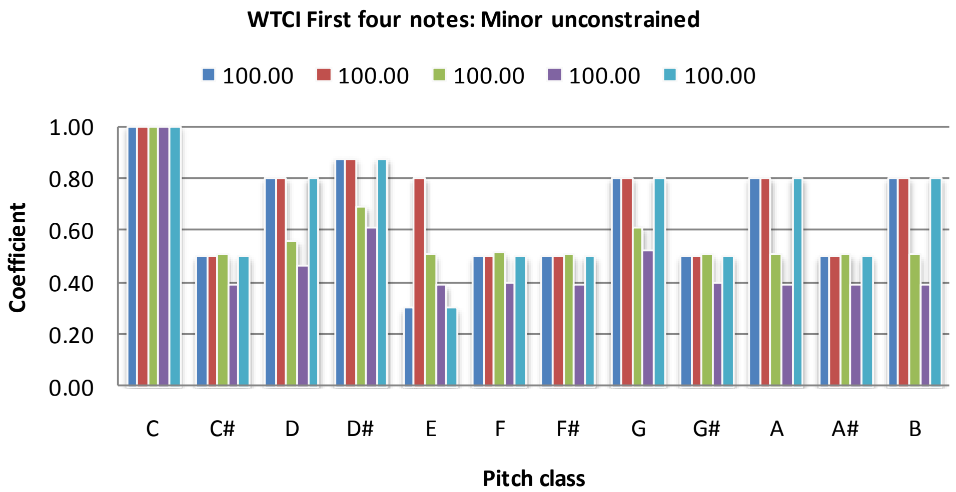


Fig. 5.39 Best scoring unconstrained minor mode key profiles with linear durational weighing for the first four notes of the 24 preludes of Bach's Well-tempered Clavier Book I.

Table 5.16 Correlations for the unconstrained minor mode key profiles shown in Fig. 5.39.

	1.0_A_2N	1.0_A_Np1	1.0_B_2N	1.0_B_Np1	0.1_A_2N
1.0_A_2N	1.0000	0.7441	0.6809	0.6804	1.0000
1.0_A_Np1	0.7441	1.0000	0.6694	0.6639	0.7441
1.0_B_2N	0.6809	0.6694	1.0000	0.9997	0.6809
1.0_B_Np1	0.6804	0.6639	0.9997	1.0000	0.6804
0.1_A_2N	1.0000	0.7441	0.6809	0.6804	1.0000

Fig. 5.40 and Fig. 5.41 present the constrained major and minor mode key profiles respectively, derived using the linear durational weighing model, for the high performance cases are identified by

the shaded cells in Table 5.14. Table 5.17 and Table 5.18 show the corresponding correlations between these profiles.

The results for both modes show good correlation between profiles 1.0_A_2N, 1.0_A_Np1 and 0.1_A_2N and profiles 1.0_B_2N and 1.0_B_Np1 respectively. The same comments as for the unconstrained cases apply with regard to this.

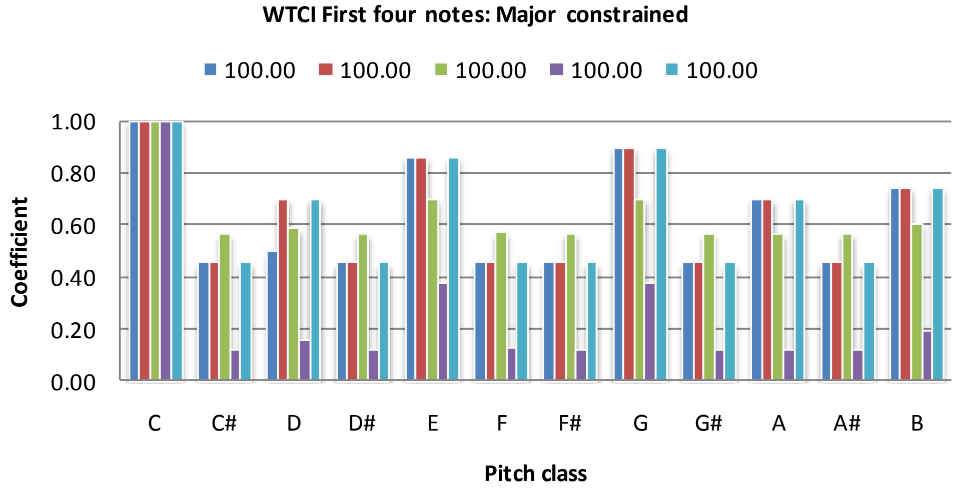


Fig. 5.40 Best scoring constrained major mode key profiles with linear durational weighing for the first four notes of the 24 preludes of Bach's Well-tempered Clavier Book I.

Table 5.17 Correlations for the constrained major mode key profiles shown in Fig. 5.40.

	1.0_A_2N	**1.0_A_Np1**	**1.0_B_2N**	**1.0_B_Np1**	**0.1_A_2N**
1.0_A_2N	1.0000	0.9616	0.8090	0.8017	0.9616
1.0_A_Np1	0.9616	1.0000	0.7882	0.7815	1.0000
1.0_B_2N	0.8090	0.7882	1.0000	0.9997	0.7882
1.0_B_Np1	0.8017	0.7815	0.9997	1.0000	0.7815
0.1_A_2N	0.9616	1.0000	0.7882	0.7815	1.0000

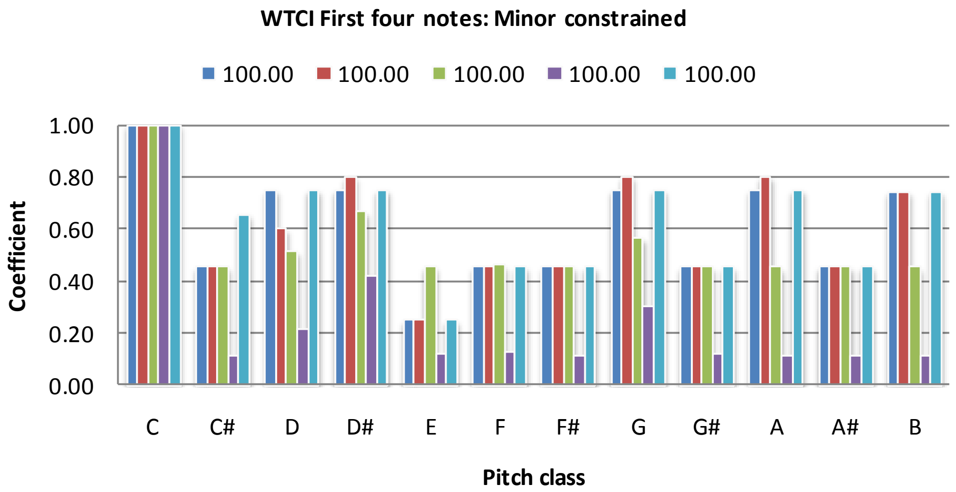


Fig. 5.41 Best scoring constrained minor mode key profiles with linear durational weighing for the first four notes of the 24 preludes of Bach's Well-tempered Clavier Book I.

Table 5.18 Correlations for the constrained minor mode key profiles shown in Fig. 5.41.

	1.0_A_2N	1.0_A_Np1	1.0_B_2N	1.0_B_Np1	0.1_A_2N
1.0_A_2N	1.0000	0.9712	0.7110	0.7116	0.9626
1.0_A_Np1	0.9712	1.0000	0.7079	0.7066	0.9348
1.0_B_2N	0.7110	0.7079	1.0000	0.9993	0.6867
1.0_B_Np1	0.7116	0.7066	0.9993	1.0000	0.6864
0.1_A_2N	0.9626	0.9348	0.6867	0.6864	1.0000

5.3.5 Key estimation results for the Courante from Bach's Cello Suite in C major

Table 5.19 and Table 5.20 summarise the unconstrained and constrained key estimation performances respectively for the *Courante* from Bach's *Cello Suite in C major*. Fig. 5.42, Fig. 5.43 and Fig. 5.44 compare the key profile optimization performances for the *Courante* from Bach's *Cello Suite in C major*, for key resolutions of 0.1%, 1.0% and 10.0% respectively.

The following observations apply for the results:

- *Unconstrained case*: The flat weighing model achieves a 100% score for key resolutions of 0.1% and 1.0% with cost functions B and C. A highest score of 98.75% is achieved for a key resolution of 10.0% with cost function C. The two durational weighing models achieve a 100% score for a key resolution of 0.1% with cost function B. A highest score of 97.50% is achieved for key resolutions of 1.0% and 10.0% with cost function B.
- *Constrained case*: The flat weighing model achieves a 100% score for key resolutions of 0.1% and 1.0% with cost functions B and C. A highest score of 98.75% is achieved for a key resolution of 10.0% with cost function B. The two durational weighing models achieve a highest score of 97.50% for a key resolution of 0.1% with cost function B. A highest score of 97.50% is achieved for a key resolution of 1.0% with cost functions A and B. A highest score of 95.42% is achieved for a key resolution of 10.0% with cost function C.

Overall, the flat weighing model outperforms durational models marginally. The slightly weaker performance figures achieved for a key resolution of 10.0% for all weighing models, although still excellent, suggests that this value may represent an upper limit for practical use. The estimated key profiles outperform the results given in Table 5.4 for the Krumhansl and Kessler and Temperley key profiles significantly. The Krumhansl and Kessler key profile delivered a best score of 65% for all weighing models. The Temperley key profile achieved a score of 100% with the flat weighing model, but only 83.75% with the durational models.

***Table 5.19 Unconstrained key profile estimation performance percentage for the* Courante *from Bach's* Cello Suite in C major.**

Weighing	Key resolution [%]	Cost function A		Cost function B		Cost function C	
		2N	Np1	2N	Np1	2N	Np1
Flat	0.1	62.50	62.50	100.00	100.00	85.00	100.00
	1.0	62.50	62.50	100.00	100.00	97.50	100.00
	10.0	83.75	81.25	98.33	96.25	98.75	97.92
Linear durational	0.1	55.00	55.00	97.50	100.00	51.25	97.5
	1.0	55.00	55.00	97.50	97.50	56.25	75.00
	10.0	91.67	79.17	97.50	96.25	95.42	96.67
Durational accent	0.1	55.00	55.00	97.50	100.00	51.25	97.50
	1.0	55.00	55.00	97.50	97.50	53.75	76.25
	10.0	91.25	77.92	96.25	97.50	91.25	95.83

Table 5.20 Constrained key profile estimation performance percentage for the **Courante** *from Bach's* **Cello Suite in C major.**

Weighing	Key resolution [%]	Cost function A		Cost function B		Cost function C	
		2N	Np1	2N	Np1	2N	Np1
Flat	0.1	62.50	62.50	100.00	97.50	100.00	100.00
	1.0	85.00	62.50	100.00	100.00	100.00	100.00
	10.0	77.50	88.75	98.75	95.00	96.67	94.88
Linear durational	0.1	50.00	87.50	97.50	97.50	55.00	80.00
	1.0	50.00	97.50	97.50	97.50	50.00	92.50
	10.0	89.17	92.92	95.00	95.00	95.42	88.75
Durational accent	0.1	50.00	87.50	97.50	85.00	80.00	60.00
	1.0	50.00	97.50	97.50	85.00	50.00	80.00
	10.0	89.17	92.92	95.00	95.00	95.42	88.75

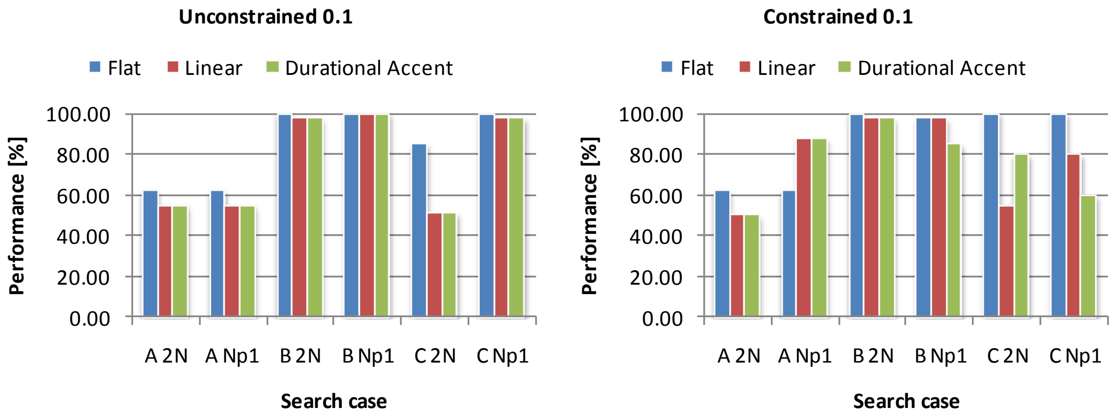


Fig. 5.42 Key profile estimation performances for a key resolution of 0.1% for the Courante from Bach's Cello Suite in C major.

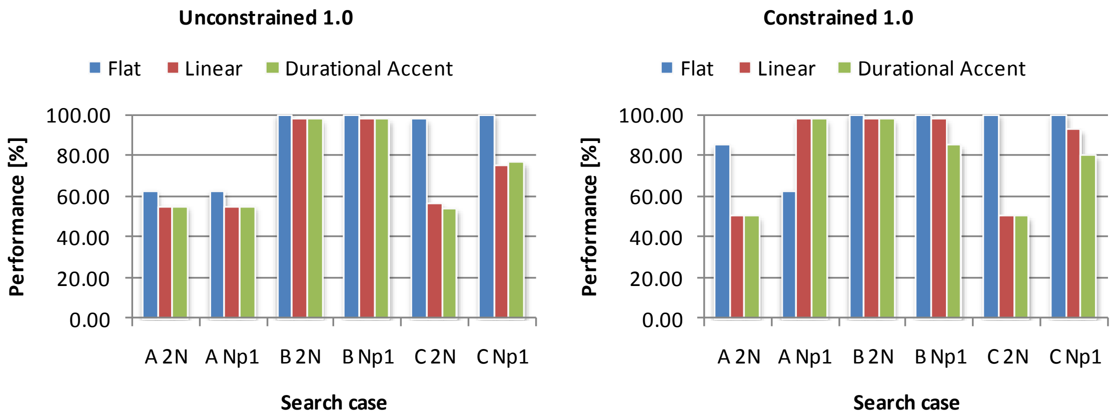


Fig. 5.43 Key profile estimation performances for a key resolution of 1.0% for the Courante from Bach's Cello Suite in C major.

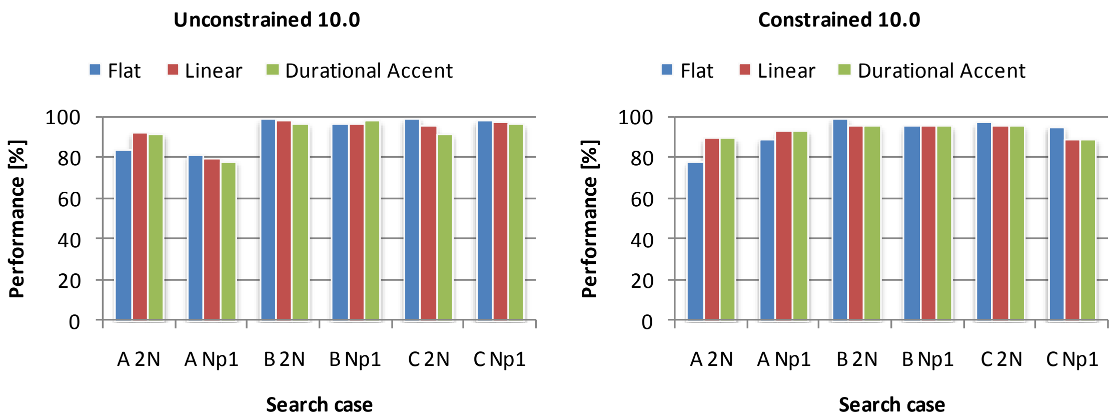


Fig. 5.44 Key profile estimation performances for a key resolution of 10.0% for the Courante from Bach's Cello Suite in C.

Table 4.12 shows multiple reference keys for many measures of the *Courante*. This material therefore represents a much different test case compared to the cases of the *Well-tempered Clavier*, where a single unambiguous reference key is assigned for each partition. The excellent performance figures achieved for the *Courante* confirm that the concept of a key resolution region works well in practice for multiple reference keys.

Fig. 5.45 and Fig. 5.46 present the unconstrained major and minor mode key profiles respectively, derived using the linear durational weighing model, for the high performance cases are identified by the shaded cells in Table 5.19. Table 5.21 and Table 5.22 show the corresponding correlations between these profiles.

The correlation results for both modes show good correlation between the profiles. The high correlation values show that the profiles for the individual modes correspond well and generally exhibit less variation compared to the results obtained for the *Well-tempered Clavier*. The key profiles show that, for both modes, the expected pitch class hierarchies are not adhered to, in the sense that the weights assigned to the supertonic, subdominant and leading tone are similar or even higher compared to the weights assigned to the dominant and mediant. This is an important development, which is most likely due to the fact that multiple reference keys needed to be accommodated.

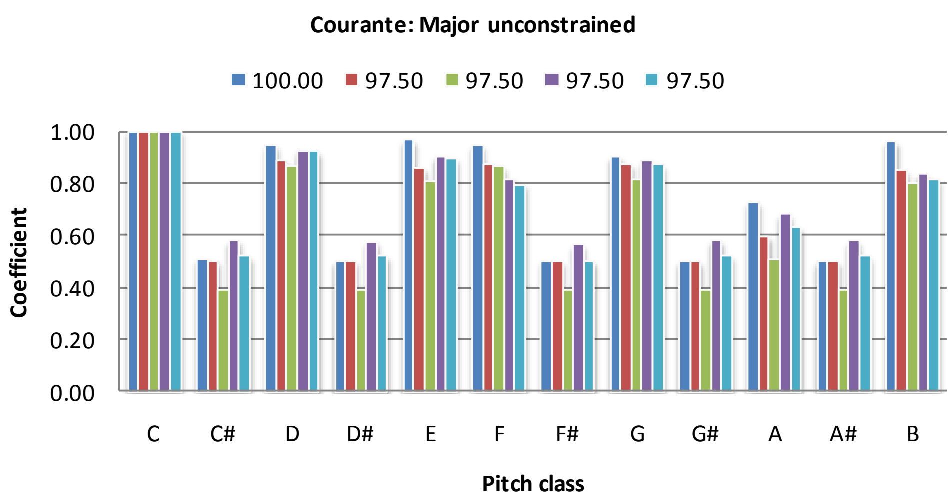


Fig. 5.45 Best scoring unconstrained major mode key profiles with linear durational weighing for the Courante from Bach's Cello Suite in C major.

Table 5.21 Correlations for the unconstrained major mode key profiles shown in Fig. 5.45.

	0.1_B_Np1	10_B_2N	1.0_B_2N	1.0_B_Np1	0.1_B_2N
0.1_B_Np1	1.0000	0.9789	0.9768	0.9673	0.9673
10_B_2N	0.9789	1.0000	0.9988	0.9858	0.9857
1.0_B_2N	0.9768	0.9988	1.0000	0.9822	0.9823
1.0_B_Np1	0.9673	0.9858	0.9822	1.0000	0.9999
0.1_B_2N	0.9673	0.9857	0.9823	0.9999	1.0000

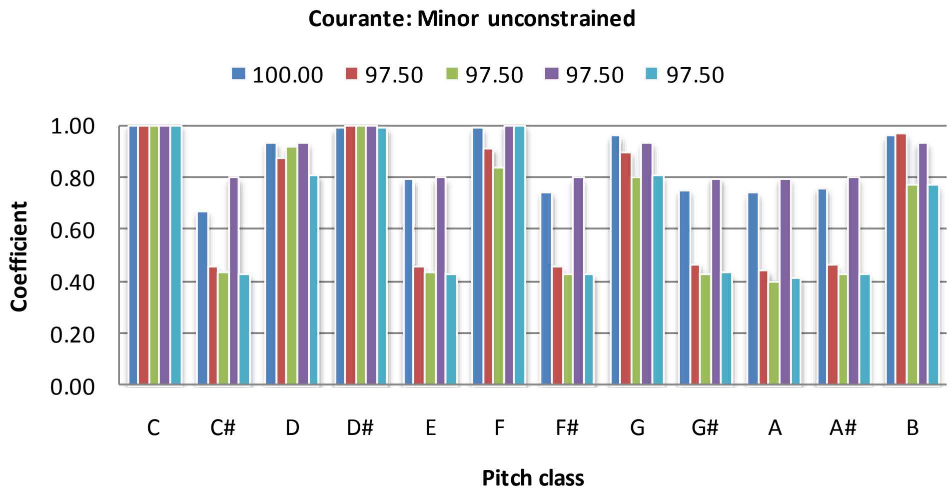


Fig. 5.46 Best scoring unconstrained minor mode key profiles with linear durational weighing for the Courante from Bach's Cello Suite in C major.

Table 5.22 Correlations for the unconstrained minor mode key profiles shown in Fig. 5.46.

	0.1_B_Np1	10_B_2N	1.0_B_2N	1.0_B_Np1	0.1_B_2N
0.1_B_Np1	1.0000	0.9715	0.9473	0.9659	0.9605
10_B_2N	0.9715	1.0000	0.9725	0.9753	0.9675
1.0_B_2N	0.9473	0.9725	1.0000	0.9734	0.9735
1.0_B_Np1	0.9659	0.9753	0.9734	1.0000	0.9991
0.1_B_2N	0.9605	0.9675	0.9735	0.9991	1.0000

Fig. 5.47 and Fig. 5.48 present the constrained major and minor mode key profiles respectively, derived using the linear durational weighing model, for the high performance cases identified by the shaded cells in Table 5.20. Table 5.23 and Table 5.24 show the corresponding correlations between these profiles. Similar observations as for the unconstrained case apply.

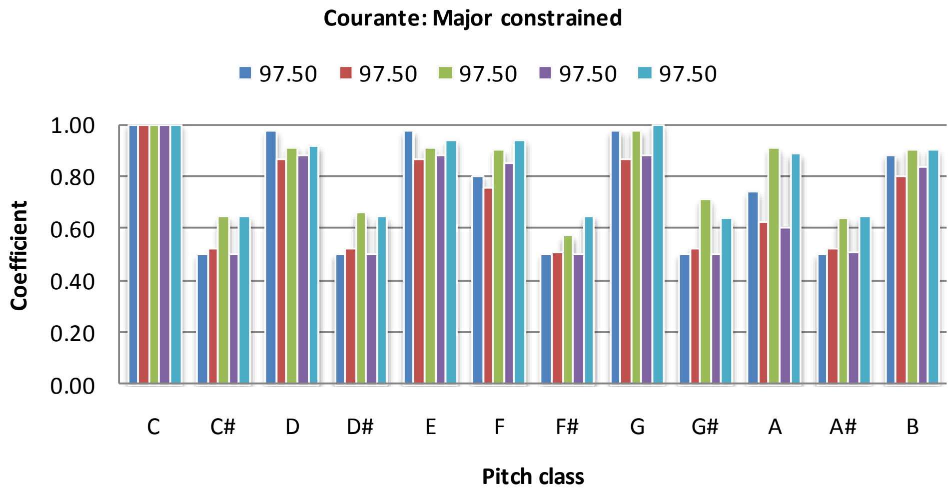


Fig. 5.47 Best scoring constrained majormode key profiles with linear durational weighing for the Courante from Bach's Cello Suite in C major.

Table 5.23 Correlations for the constrained major mode key profiles shown nFig. 5.47.

	1.0_A_Np1	1.0_B_2N	1.0_B_Np1	0.1_B_2N	0.1_B_Np1
1.0_A_Np1	1.0000	0.9772	0.9449	0.9680	0.9675
1.0_B_2N	0.9772	1.0000	0.9154	0.9873	0.9352
1.0_B_Np1	0.9449	0.9154	1.0000	0.9094	0.9733
0.1_B_2N	0.9680	0.9873	0.9094	1.0000	0.9433
0.1_B_Np1	0.9675	0.9352	0.9733	0.9433	1.0000

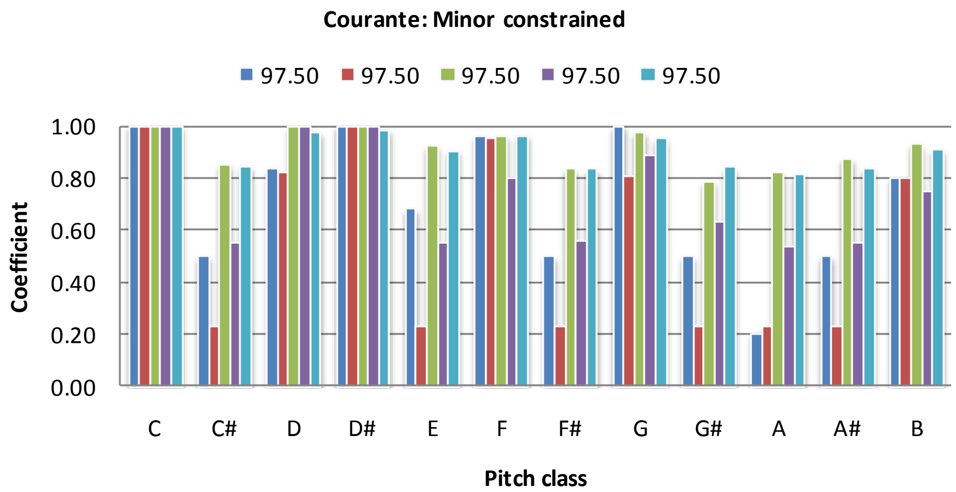


Fig. 5.48 Best scoring constrained minor mode key profiles with linear durational weighing for the Courante from Bach's Cello Suite in C major.

Table 5.24 Correlations for the constrained minor mode key profiles shown in Fig. 5.48.

	1.0_A_Np1	1.0_B_2N	1.0_B_Np1	0.1_B_2N	0.1_B_Np1
1.0_A_Np1	1.0000	0.9039	0.9104	0.8605	0.9462
1.0_B_2N	0.9039	1.0000	0.8835	0.9280	0.9290
1.0_B_Np1	0.9104	0.8835	1.0000	0.8640	0.9591
0.1_B_2N	0.8605	0.9280	0.8640	1.0000	0.9278
0.1_B_Np1	0.9462	0.9290	0.9591	0.9278	1.0000

5.3.6 Key estimation results for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816)

Table 5.25 and Table 5.26 summarise the unconstrained and constrained key estimation performances respectively for the *Gavotte* from Bach's *French Suite No. 5 in G major (BWV 816)*.

Fig. 5.49, Fig. 5.50 and Fig. 5.51 compare the key profile optimization performances for the *Gavotte* from Bach's *French Suite No. 5 in G major (BWV 816)*, for key resolutions of 0.1%, 1.0% and 10.0% respectively.

The following observations apply for the results:

- *Unconstrained case*: The flat weighing model achieves a highest score of 88% for all key resolutions. The linear durational weighing model achieves a highest score of 92% for key resolutions of 0.1% and 1.0% with cost function B. A highest score of 89.33% is achieved for a key resolution of 10.0% with cost function B. The durational accent weighing model achieves a highest score of 96% score for a key resolution of 0.1% with cost function B. A highest score of 92% is achieved for a key resolution of 1.0% with cost functions B and C. A highest score of 89.33% is achieved for a key resolution of 10.0% with cost function B. Overall, the durational models outperform the flat weighing model marginally.
- *Constrained case*: The flat weighing model achieves a highest score of 88% for key resolutions of 0.1%, 1.0% and 10.0% with cost functions B and C. The linear durational weighing model achieves a highest score of 92% for a key resolution of 0.1% with cost functions B and C. A highest score of 92% is also achieved for a key resolution of 1.0% with cost function C. A highest score of 91.33% is achieved for a key resolution of 10.0% with cost function B. The durational accent weighing model achieves a highest score of 96% for a key resolution of 0.1% with cost functions B and C. A highest score of 94% is achieved for a key resolution of 1.0% with cost function C. A highest score of 84.67% is achieved for a key resolution of 10.0% with cost function B. Overall, the durational models outperform the flat weighing model.

The estimated key profiles outperform the results given in Table 5.5 for the Krumhansl and Kessler and Temperley key profiles significantly. The Krumhansl and Kessler key profile delivered a best score of 64% with the flat weighing model, but only 40% with the durational weighing models. The Temperley key profile achieved a score of 78% with the flat weighing model, but only 74% and 68% respectively with the linear durational and durational accent models respectively.

Table 5.25 Unconstrained key profile estimation performance percentage for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816).

Weighing	**Key resolution [%]**	**Cost function A**		**Cost function B**		**Cost function C**	
		2N	**Np1**	**2N**	**Np1**	**2N**	**Np1**
Flat	0.1	80.00	80.00	88.00	88.00	88.00	88.00
	1.0	80.00	80.00	88.00	88.00	88.00	84.00
	10.0	88.00	72.00	88.00	88.00	88.00	88.00
Linear durational	0.1	44.00	44.00	88.00	92.00	72.00	80.00
	1.0	84.00	76.00	92.00	92.00	88.00	88.00
	10.0	80.00	81.33	88.00	89.33	87.33	74.00
Durational accent	0.1	84.00	76.00	88.00	96.00	92.00	92.00
	1.0	84.00	80.00	88.00	92.00	92.00	86.00
	10.0	73.33	73.00	86.00	89.33	72.00	66.00

Table 5.26 Constrained key profile estimation performance percentage for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816).

Weighing	Key resolution [%]	Cost function A		Cost function B		Cost function C	
		2N	Np1	2N	Np1	2N	Np1
Flat	0.1	72.00	76.00	88.00	88.00	88.00	88.00
	1.0	72.00	76.00	88.00	88.00	88.00	88.00
	10.0	80.67	74.67	88.00	88.00	88.00	88.00
Linear durational	0.1	72.00	68.00	88.00	92.00	92.00	92.00
	1.0	76.00	86.00	88.00	88.00	92.00	92.00
	10.0	76.00	74.67	89.33	91.33	87.33	81.33
Durational accent	0.1	48.00	76.00	96.00	88.00	96.00	88.00
	1.0	48.00	76.00	88.00	88.00	94.00	88.00
	10.0	79.33	66.00	84.67	78.67	87.33	82.33

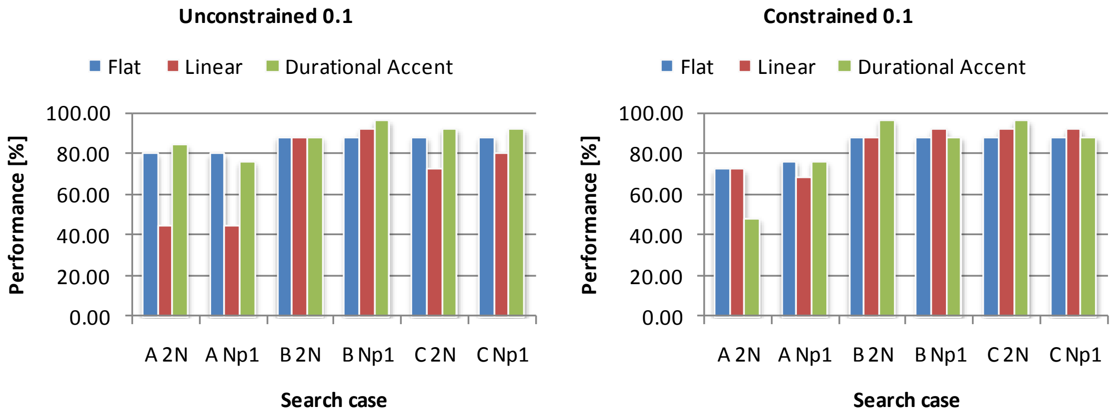


Fig. 5.49 Key profile estimation performances for a key resolution of 0.1% for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816).

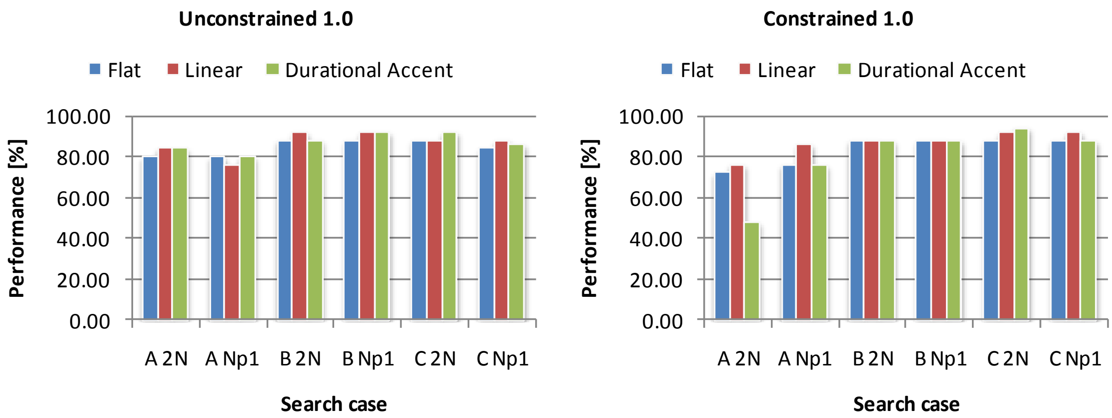
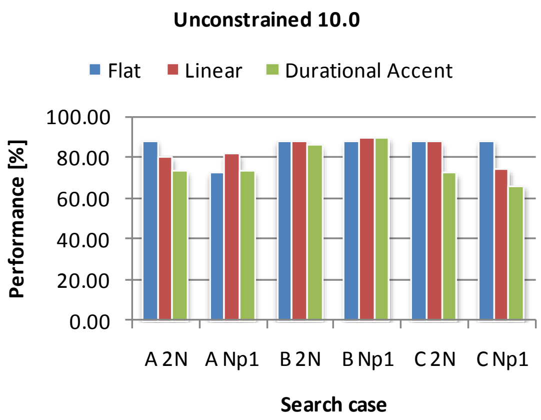


Fig. 5.50 Key profile estimation performances for a key resolution of 1.0% for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816).



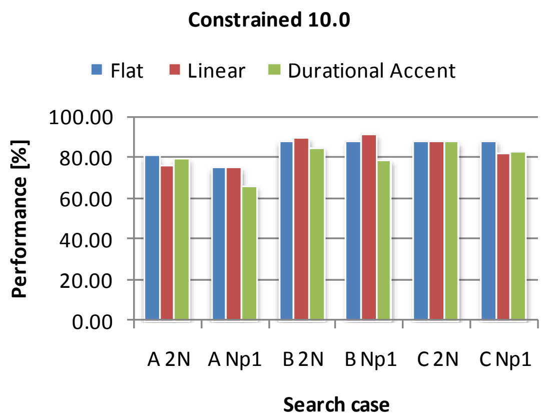


Fig. 5.51 Key profile estimation performances for a key resolution of 10.0% for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816).

Fig. 5.52 and Fig. 5.53 present the constrained major and minor mode key profiles respectively, derived using the linear durational weighing model, for the high performance cases are identified by the shaded cells in Table 5.25. Table 5.27 and Table 5.28 show the corresponding correlations between these profiles.

The correlation results for both modes show good correlation between the profiles. With regard to the pitch class hierarchies reflected by the estimated profiles, the same observations as for the *Courante* apply, in the sense that the weights assigned to the supertonic, subdominant and leading tone are unexpectedly high.

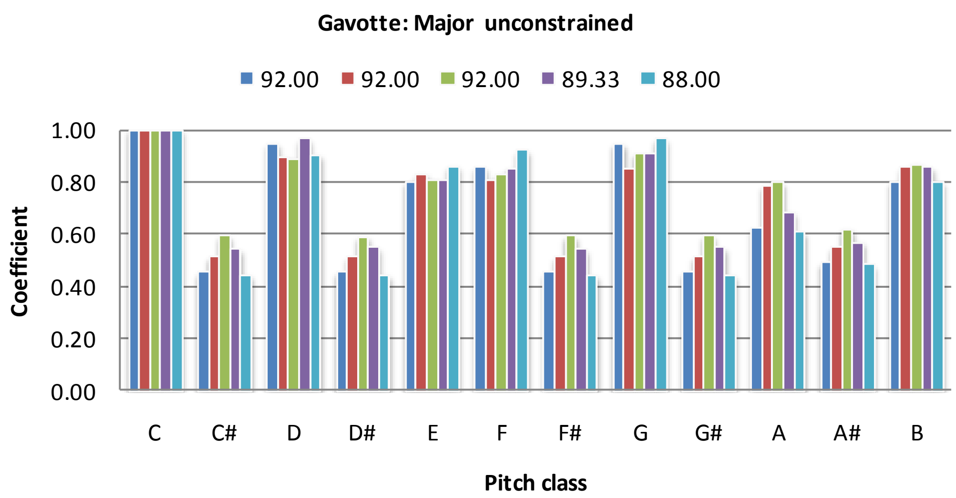


Fig. 5.52 Best scoring unconstrained major mode key profiles with linear durational weighing for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816).

Table 5.27 Correlations for the unconstrained major mode key profiles shown in Fig. 5.52.

	1.0_B_2N	**1.0_B_Np1**	**0.1_B_Np1**	**10.0_B_Np1**	**10.0_B_2N**
1.0_B_2N	1.0000	0.9618	0.9713	0.9953	0.9921
1.0_B_Np1	0.9618	1.0000	0.9916	0.9700	0.9494
0.1_B_Np1	0.9713	0.9916	1.0000	0.9764	0.9559
10.0_B_Np1	0.9953	0.9700	0.9764	1.0000	0.9795
10.0_B_2N	0.9921	0.9494	0.9559	0.9795	1.0000

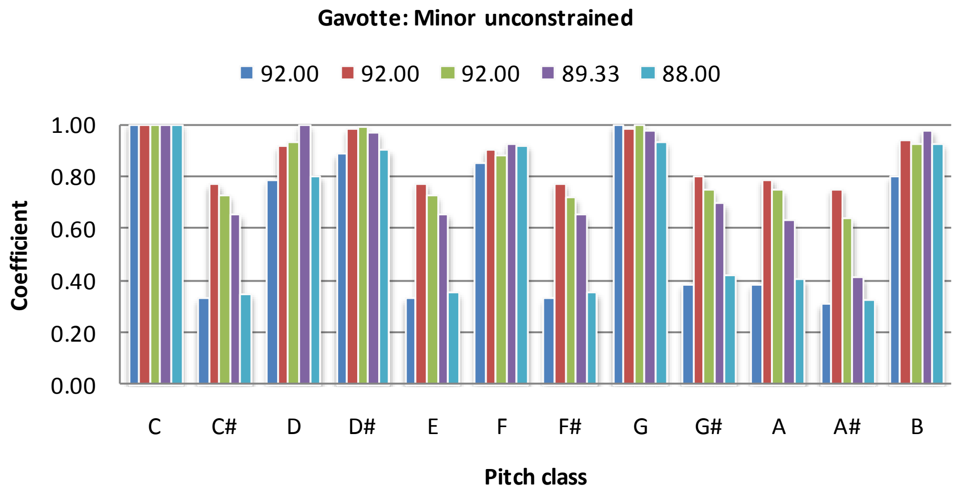


Fig. 5.53 Best scoring unconstrained minor mode key profiles with linear durational weighing for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816).

Table 5.28 Correlations for the unconstrained minor mode key profiles shown in Fig. 5.53.

	1.0_B_2N	1.0_B_Np1	0.1_B_Np1	10.0_B_Np1	10.0_B_2N
1.0_B_2N	1.0000	0.9850	0.9748	0.9282	0.9884
1.0_B_Np1	0.9850	1.0000	0.9890	0.9332	0.9735
0.1_B_Np1	0.9748	0.9890	1.0000	0.9655	0.9605
10.0_B_Np1	0.9282	0.9332	0.9655	1.0000	0.9378
10.0_B_2N	0.9884	0.9735	0.9605	0.9378	1.0000

Fig. 5.54 and Fig. 5.55 present the constrained major and minor mode key profiles respectively, derived using the linear durational weighing model, for the high performance cases are identified by the shaded cells in Table 5.26. Table 5.29 and Table 5.30 show the corresponding correlations between these profiles.

The correlation results for the major mode show excellent correlation between profiles. The correlation results for the minor mode are much poorer, with exception of the correlation between profiles 1.0_C_Np1 and 0.1_C_Np1.

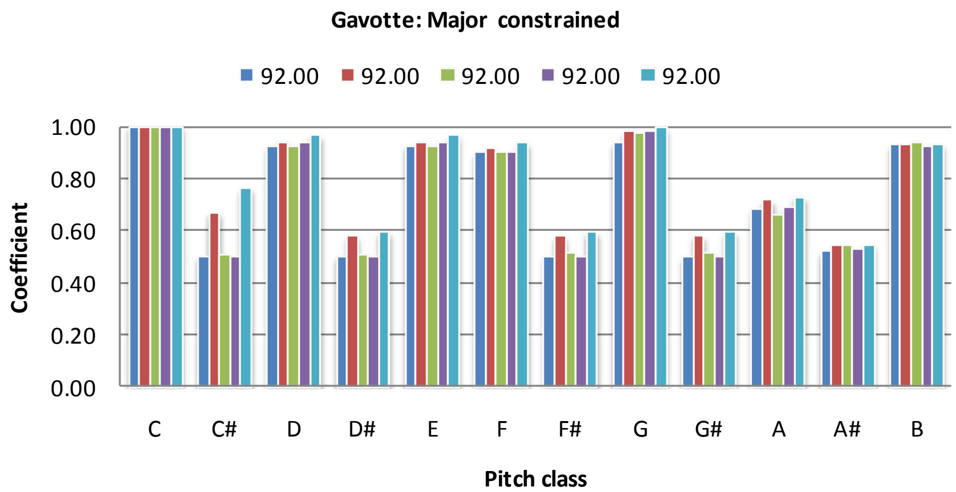


Fig. 5.54 Best scoring constrained major mode key profiles with linear durational weighing for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816).

Table 5.29 Correlations for the constrained major mode key profiles shown in Fig. 5.54.

	1.0_C_2N	1.0_C_Np1	0.1_B_Np1	0.1_C_2N	0.1_C_Np1
1.0_C_2N	1.0000	0.9840	0.9978	0.9986	0.9537
1.0_C_Np1	0.9840	1.0000	0.9839	0.9844	0.9909
0.1_B_Np1	0.9978	0.9839	1.0000	0.9989	0.9531
0.1_C_2N	0.9986	0.9844	0.9989	1.0000	0.9543
0.1_C_Np1	0.9537	0.9909	0.9531	0.9543	1.0000

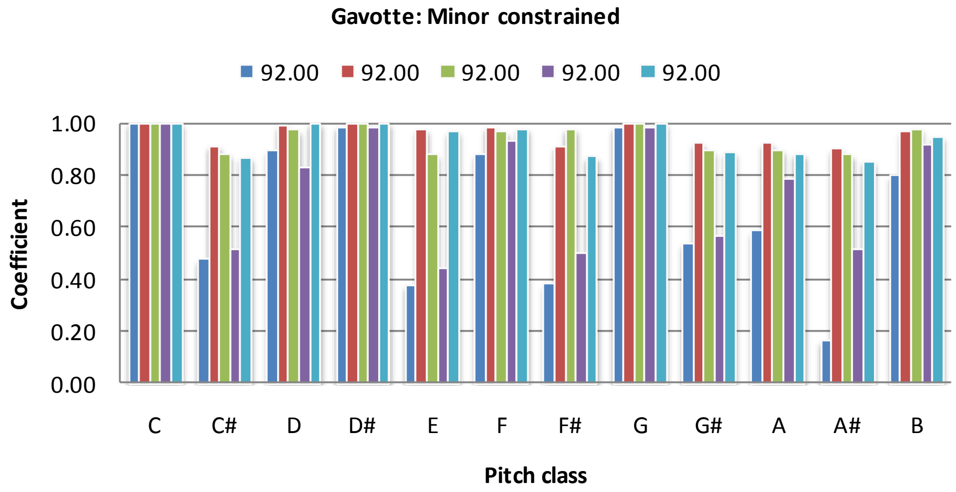


Fig. 5.55 Best scoring constrained minor mode key profiles with linear durational weighing for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816).

Table 5.30 Correlations for the constrained minor mode key profiles shown in Fig. 5.55.

	1.0_C_2N	1.0_C_Np1	0.1_B_Np1	0.1_C_2N	0.1_C_Np1
1.0_C_2N	1.0000	0.8610	0.8143	0.9378	0.8485
1.0_C_Np1	0.8610	1.0000	0.7140	0.7655	0.9977
0.1_B_Np1	0.8143	0.7140	1.0000	0.7936	0.7033
0.1_C_2N	0.9378	0.7655	0.7936	1.0000	0.7445
0.1_C_Np1	0.8485	0.9977	0.7033	0.7445	1.0000

5.3.6.2 Performance evaluation of the optimised key profiles

The key-finding performances of the entire set of optimised key profiles were determined for key resolutions of 0.1%, 1.0% and 10.0% for all weighing models and all test cases. This allowed the average performance of each key profile for the four test cases to be calculated. The optimised profiles with the best average key-finding scores for all test cases are listed in Table 5.31. The profile names indicate the score material for which the profile was derived, whether a constrained or unconstrained estimation was performed and the key resolution, cost function and polling method that was used.

Table 5.31 Optimised key profiles with the best average key-finding scores for all test cases.

Key resolution	**Pitch class weighing model**		
	Flat	**Linear durational**	**Durational accent**
0.1%	Courante_Constrained_0.1_B_2N	Gavotte_Unconstrained_1.0_B_2N	Gavotte_Unconstrained_1.0_B_2N
1.0%	Courante_Unconstrained_0.1_B_Np1	Gavotte_Unconstrained_1.0_B_2N	Gavotte_Unconstrained_1.0_B_2N
10.0%	Courante_Constrained_1.0_A_Np1	Courante_Unconstrained_10.0_B_2N	Courante_Unconstrained_10.0_B_2N

The key-finding scores of the profiles listed in Table 5.31 are now compared with the Krumhansl and Kessler and the Temperley key profiles.

Table 5.32 to Table 5.34 and Fig. 5.56 to Fig. 5.58 summarise the key-finding performances of the various key profiles for a key resolution of 0.1% for the flat, linear durational and durational accent weighing models respectively.

The best average score, i.e. 96.33%, is obtained with linear durational weighing for the optimised key profile derived with the unconstrained algorithm for the *Gavotte* from Bach's *French Suite No. 5 in G major (BWV 816)* with a key resolution of 1.0% and cost function B.

Table 5.32 Key-finding performances of the optimised, Krumhansl and Kessler and Temperley key profiles for a key resolution of 0.1% with flat weighing.

Key profile	WTCI Full scores [%]	WTCI First 4 notes [%]	Courante [%]	Gavotte [%]	Average [%]
Courante_Constrained_0.1_B_2N	5.56	100.00	97.50	88.00	72.76
Krumhansl and Kessler	7.64	100.00	62.50	64.00	58.53
Temperley	3.47	100.00	100.00	78.00	70.37

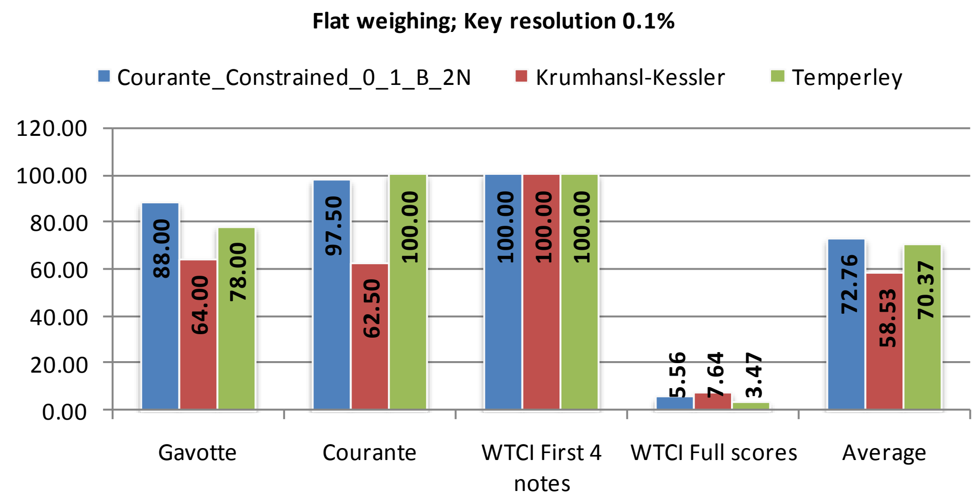


Fig. 5.56 Performances of the optimised, Krumhansl and Kessler and Temperley key profiles for a key resolution of 0.1% with the flat pitch class weighing model.

Table 5.33 Key-finding performances of the optimised, Krumhansl and Kessler and Temperley key profiles for a key resolution of 0.1% with linear durational weighing.

Key profile	WTCI Full scores [%]	WTCI First 4 notes [%]	Courante [%]	Gavotte [%]	Average [%]
Gavotte_Unconstrained_1.0_B_2N	95.83	100.00	97.50	92.00	96.33
Krumhansl and Kessler	93.75	100.00	65.00	40.00	74.69
Temperley	91.67	100.00	83.75	74.00	87.35

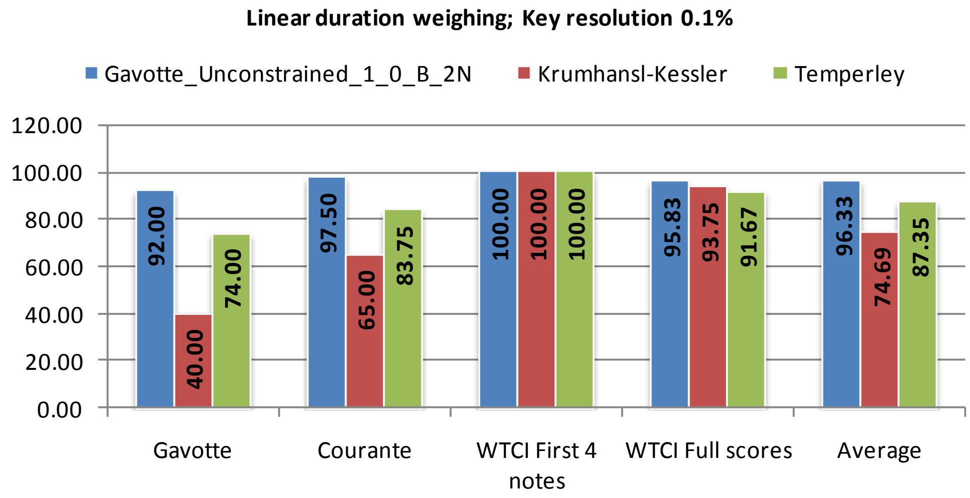


Fig. 5.57 Performances of the optimised, Krumhansl and Kessler and Temperley key profiles for a key resolution of 0.1% with the linear durational pitch class weighing model.

Table 5.34 Key-finding performances of the optimised, Krumhansl and Kessler and Temperley key profiles for a key resolution of 0.1% with durational accent weighing.

Key profile	WTCI Full scores [%]	WTCI First 4 notes [%]	Courante [%]	Gavotte [%]	Average [%]
Gavotte_Unconstrained_1.0_B_2N	91.67	100.00	97.50	92.00	95.29
Krumhansl and Kessler	95.83	100.00	65.00	40.00	75.21
Temperley	83.33	100.00	83.75	68.00	83.77

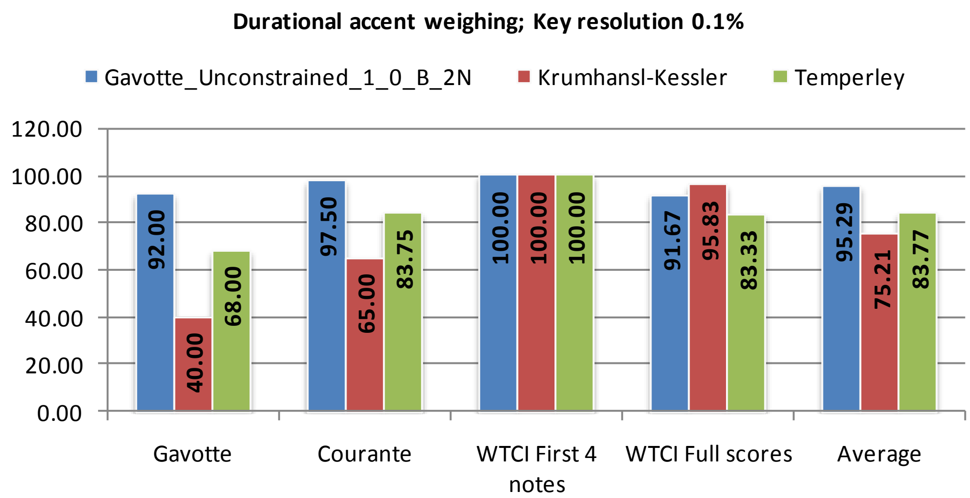


Fig. 5.58 Performances of the optimised, Krumhansl and Kessler and Temperley key profiles for a key resolution of 0.1% with the durational accent pitch class weighing model.

Table 5.35 to Table 5.37 and Fig. 5.59 to Fig. 5.61 summarise the key-finding performances of the various key profiles for a key resolution of 1.0% for the flat, linear durational and durational accent weighing models respectively.

As for a key resolution of 0.1%, the best average score, i.e. 96.85%, is obtained with linear durational weighing for the optimised key profile derived with the unconstrained algorithm for the Gavotte with a key resolution of 1.0% and cost function B.

Table 5.35 Key-finding performances of the optimised, Krumhansl and Kessler and Temperley key profiles for a key resolution of 1.0% with flat weighing.

Key profile	WTCI Full scores [%]	WTCI First 4 notes [%]	Courante [%]	Gavotte [%]	Average [%]
Courante_Unconstrained_0.1_B_Np1	7.64	100.00	97.50	84.00	72.28
Krumhansl and Kessler	7.64	100.00	61.25	64.00	58.22
Temperley	3.47	100.00	100.00	78.00	70.37

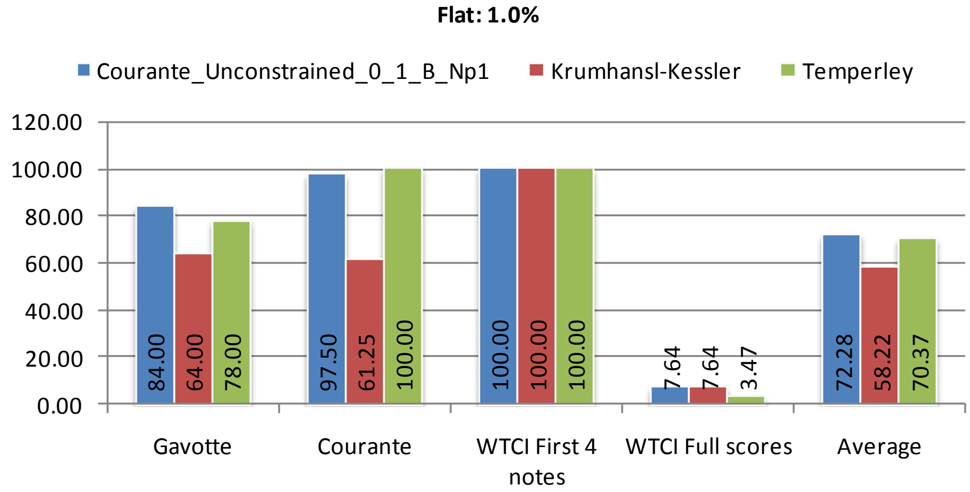


Fig. 5.59 Performances of the optimised, Krumhansl and Kessler and Temperley key profiles for a key resolution of 1.0% with the flat pitch class weighing model.

Table 5.36 Key-finding performances of the optimised, Krumhansl and Kessler and Temperley key profiles for a key resolution of 1.0% with linear durational weighing.

Key profile	WTCI Full scores [%]	WTCI First 4 notes [%]	Courante [%]	Gavotte [%]	Average [%]
Gavotte_Unconstrained_1.0_B_2N	97.92	100.00	97.50	92.00	96.85
Krumhansl and Kessler	95.83	100.00	65.00	40.00	75.21
Temperley	91.67	100.00	83.75	74.00	87.35

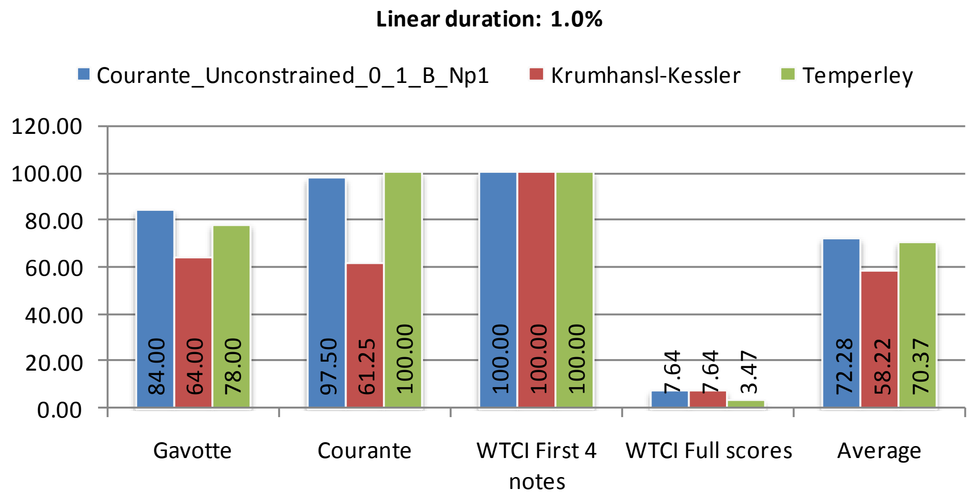


Fig. 5.60 Performances of the optimised, Krumhansl and Kessler and Temperley key profiles for a key resolution of 1.0% with the linear durational pitch class weighing model.

Table 5.37 Key-finding performances of the optimised, Krumhansl and Kessler and Temperley key profiles for a key resolution of 1.0% with durational accent weighing.

Key profile	WTCI Full scores [%]	WTCI First 4 notes [%]	Courante [%]	Gavotte [%]	Average [%]
Gavotte_Unconstrained_1.0_B_2N	91.67	100.00	97.50	92.00	95.29
Krumhansl and Kessler	95.83	100.00	65.00	40.00	75.21
Temperley	83.33	100.00	83.75	70.00	84.27

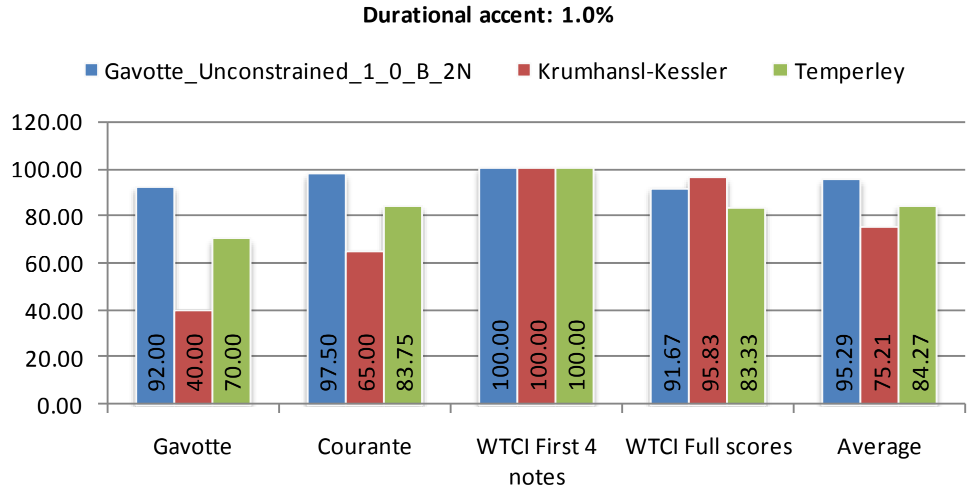


Fig. 5.61 Performances of the optimised, Krumhansl and Kessler and Temperley key profiles for a key resolution of 1.0% with the durational accent pitch class weighing model.

Table 5.38 and Table 5.40 and Fig. 5.62 and Fig. 5.64 summarise the key-finding performances of the various key profiles for a key resolution of 10.0% for the flat, linear durational and durational accent weighing models respectively.

The best average score, i.e. 91.58%, is obtained with linear durational weighing for the optimised key profile derived with the unconstrained algorithm for the *Courante* from Bach's *Cello Suite in C major* with a key resolution of 10.0% and cost function B.

In terms of the average scores for all test cases, the key profiles derived with the pattern search algorithm outperform the Krumhansl and Kessler profile convincingly for all weighing models. The only exception is occurs in Table 5.38, where the Temperley key profile with flat weighing achieves the highest average score for a key resolution of 10.0%. The reason for this is that the optimised key profile is optimised for linear durational weighing.

Table 5.38 Key-finding performances of the optimised, Krumhansl and Kessler and Temperley key profiles for a key resolution of 10.0% with flat weighing.

Key profile	WTCI Full scores [%]	WTCI First 4 notes [%]	Courante [%]	Gavotte [%]	Average [%]
Courante_Constrained_1.0_A_Np1	4.86	100.00	90.00	79.00	68.47
Krumhansl and Kessler	6.53	33.33	57.50	52.67	37.51
Temperley	3.47	100.00	100.00	82.00	71.37

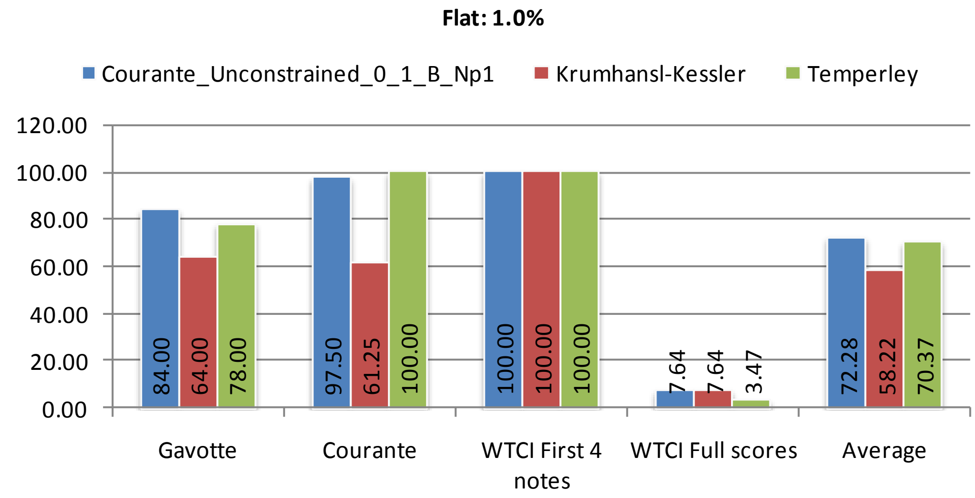


Fig. 5.62 Performances of the optimised, Krumhansl and Kessler and Temperley key profiles for a key resolution of 10.0% with the flat pitch class weighing model.

Table 5.39 Key-finding performances of the optimised, Krumhansl and Kessler and Temperley key profiles for a key resolution of 10.0% with linear durational weighing.

Key profile	WTCI Full scores [%]	WTCI First 4 notes [%]	Courante [%]	Gavotte [%]	Average [%]
Courante_Unconstrained_10.0_B_2N	87.50	100.00	97.50	81.33	91.58
Krumhansl and Kessler	83.33	100.00	52.50	37.33	68.29
Temperley	86.81	100.00	80.42	72.67	84.97

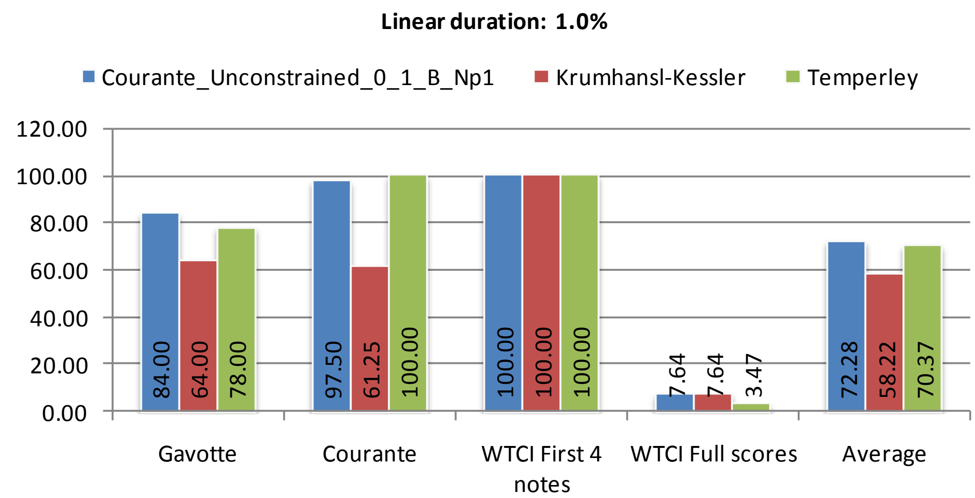


Fig. 5.63 Performances of the optimised, Krumhansl and Kessler and Temperley key profiles for a key resolution of 10.0% with the linear durational pitch class weighing model.

Table 5.40 Key-finding performances of the optimised, Krumhansl and Kessler and Temperley key profiles for a key resolution of 10.0% with durational accent weighing.

Key profile	WTCI Full scores [%]	WTCI First 4 notes [%]	Courante [%]	Gavotte [%]	Average [%]
Courante_Unconstrained_10.0_B_2N	83.33	100.00	97.50	83.33	91.04
Krumhansl and Kessler	81.25	100.00	52.50	38.00	67.94
Temperley	87.50	100.00	80.42	68.67	84.15

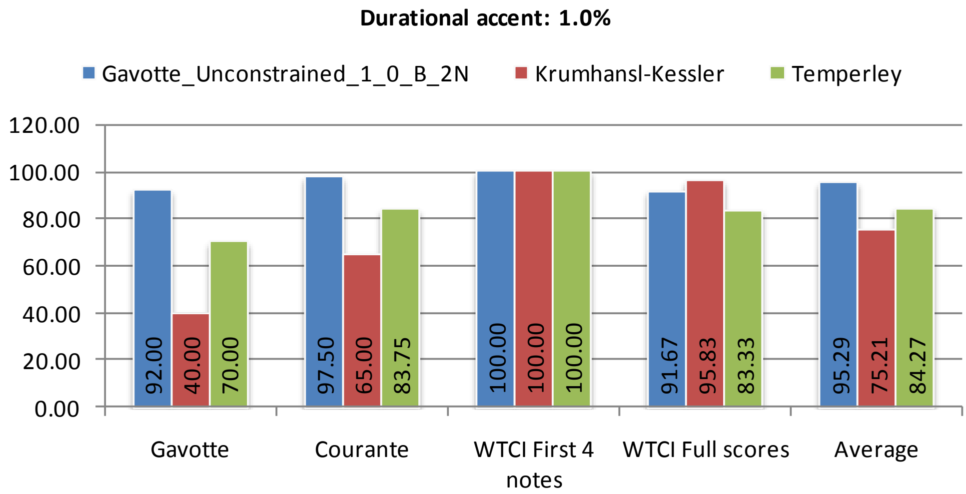


Fig. 5.64 Performances of the optimised, Krumhansl and Kessler and Temperley key profiles for a key resolution of 10.0% with the durational accent pitch class weighing model.

5.3.7 Pitch class hierarchy properties of the optimised key profiles

The pitch class coefficients of the best scoring optimised key profiles identified in section 5.3.6.2 and the profiles proposed by Krumhansl and Kessler (Krumhansl, 1990) and Temperley (1999) are summarised in Appendix K, where all values have been rounded to an accuracy of 0.01.

Fig. 5.65 compares the major mode key profiles proposed by Krumhansl and Kessler (Krumhansl, 1990) and Temperley (1999) with the optimised profiles derived with the unconstrained *Gavotte* from Bach's French *Suite No. 5 in G major (BWV 816)* with a key resolution of 1.0% and unconstrained *Courante* from Bach's *Cello Suite in C major* with a key resolution of 10.0%. The associated pitch class hierarchies, shown in Table 5.41, give rise to the following observations:

- All profiles assign the highest weight, i.e. unity, to the tonic. For the optimised key profiles, this was imposed through the constraints used in the search algorithm.
- All diatonic pitch classes are assigned higher weights compared to the nondiatonic pitch classes in all profiles.
- In the Krumhansl and Kessler hierarchy, the dominant is assigned the second position, followed by the mediant in the third position. The Temperley profile reflects a similar order, except that the dominant and mediant are assigned equal weights for the second position. The optimised hierarchies deviate drastically from this hierarchy. The optimised profile based on the *Gavotte* ranks the supertonic and dominant equally in the second position, with the subdominant in the third position. The optimised profile based on the *Courante* ranks the supertonic and the subdominant together in the second position, with the dominant in the third position.
- In the Krumhansl and Kessler hierarchy, the leading tone is assigned the lowest position of the diatonic pitch classes. The Temperley hierarchy ranks the supertonic and the submediant together in the lowest position of the diatonic pitch classes. Both optimised key profiles rank the submediant in the lowest position of the diatonic pitch classes.
- The rankings assigned to the nondiatonic pitch classes are very similar for all key profiles. The Temperley profile and the optimised profiles all assign equal weights and the penultimate position to degrees ii, iii, v and vi. The optimised hierarchy based on the *Gavotte* ranks degree vii in the sixth position, while both the Temperley hierarchy and the hierarchy based on the *Courante* rank degree vii in the lowest position.

In comparison with the Krumhansl and Kessler and the Temperley profiles, the optimised profiles assign a very high rank, i.e. second position, to the supertonic. Also, the rankings of the mediant and subdominant are reversed in the sense that the optimised profiles assign a higher weight to the subdominant compared to the mediant. The high rank assigned to the supertonic is related to the fact that it represents the dominant of the dominant, which is important in the case of key modulation to the dominant.

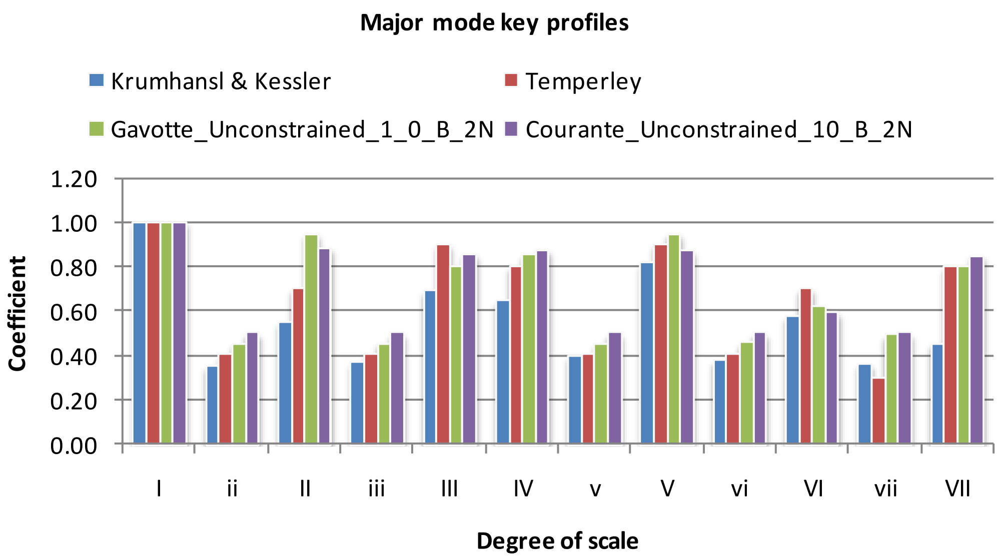


Fig. 5.65 Comparison of major mode key profiles: Krumhansl and Kessler, Temperley, unconstrained Gavotte with a key resolution of 1.0% and unconstrained Courante with a key resolution of 10.0%.

Table 5.41 Comparison of major mode pitch class hierarchies for the Krumhansl and Kessler, Temperley, unconstrained Gavotte with a key resolution of 1.0% and unconstrained Courante with a key resolution of 10.0% profiles.

Krumhansl & Kessler		**Temperley**		**Gavotte_Unconstrained_1.0_B_2N**		**Courante_Unconstrained_10.0_B_2N**	
Position	**Degree**	**Position**	**Degree**	**Position**	**Degree**	**Position**	**Degree**
1	I	1	I	1	I	1	I
2	V	2	III	2	II	2	II
3	III		V		V		IV
4	IV	3	IV	3	IV	3	V
5	VI		VII	4	III	4	III
6	II	4	II		VII	5	VII
7	VII		VI	5	VI	6	VI
8	v	5	ii	6	vii	7	ii
9	vi		iii	7	ii		iii
10	iii		v		iii		v
11	vii		vi		v		vi
12	ii	6	vii		vi		vii

Fig. 5.66 compares the minor mode key profiles proposed by Krumhansl and Kessler (Krumhansl, 1990) and Temperley (1999) with the optimised profiles derived with the unconstrained *Gavotte* with a key resolution of 1.0% and unconstrained *Courante* with a key resolution of 10.0%. The associated pitch class hierarchies, shown in Table 5.42, give rise to the following observations:

- All profiles assign the highest weight, i.e. unity, to the tonic. For the optimised key profiles, this was imposed through the constraints used in the search algorithm.
- Unlike in the case of the major mode, the hierarchies show no clear order with regard to the diatonic and nondiatonic pitch classes.
- The Krumhansl and Kessler hierarchy ranks degree iii in the second position, followed by the dominant. The Temperley hierarchy ranks these pitch classes together in the second position. The optimised profile based on the *Gavotte* ranks the dominant in the second position, followed by degree iii in the third position, i.e. in reversed order compared to the Krumhansl and Kessler hierarchy. The profile based on the *Courante* ranks the tonic and supertonic together in the first position, with the leading tone in the third position.
- In the Krumhansl and Kessler hierarchy, the leading tone is assigned the lowest position of the diatonic pitch classes. The Temperley hierarchy ranks the supertonic and the submediant together in the lowest position of the diatonic pitch classes. Both optimised key profiles rank the submediant in the last position of the diatonic pitch classes.
- The rankings assigned to the nondiatonic pitch classes are very similar for all key profiles. The Temperley profile and the optimised profiles all assign equal weights and the penultimate position to degrees ii, iii and v.

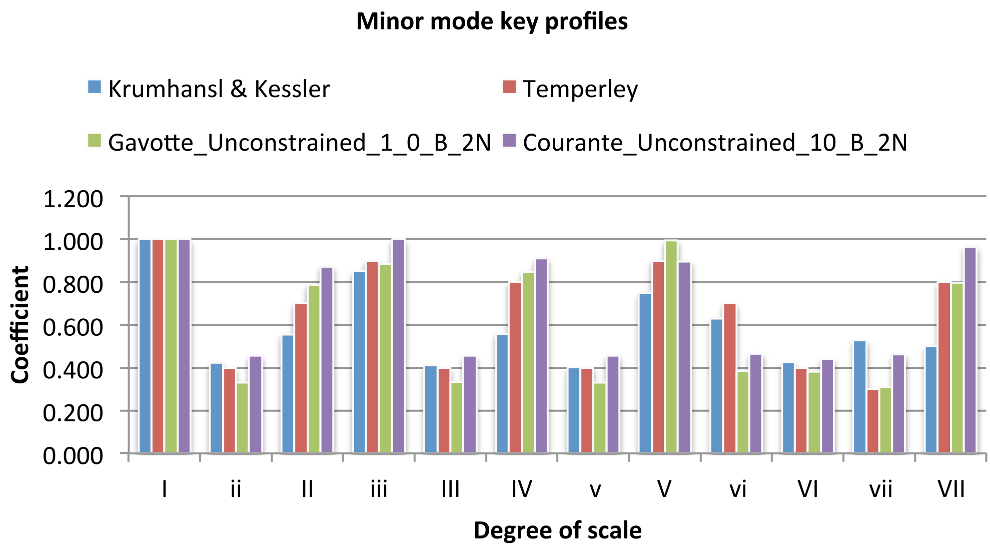


Fig. 5.66 Comparison of minor mode key profiles: Krumhansl and Kessler, Temperley, unconstrained Gavotte with a key resolution of 1.0% and unconstrained Courante with a key resolution of 10.0%.

Table 5.42 Comparison of minor mode pitch class hierarchies for the Krumhansl and Kessler, Temperley, unconstrained Gavotte with a key resolution of 1.0% and unconstrained Courante with a key resolution of 10.0%.

Krumhansl & Kessler		**Temperley**		**Gavotte_Unconstrained_1.0_B_2N**		**Courante_Unconstrained_10.0_B_2N**	
Position	**Degree**	**Position**	**Degree**	**Position**	**Degree**	**Position**	**Degree**
1	I	1	I	1	I	1	I
2	iii	2	iii	2	V		iii
3	V		V	3	iii	2	VII
4	vi	3	IV	4	IV	3	IV
5	II		VII	5	VII	4	V
	IV	4	II	6	II	5	II
6	vii		vi	7	vi	6	vi
7	VII	5	ii	8	VI	7	ii
8	ii		III	9	ii		III
	VI		v		III		v
9	III		VI		v		vii
10	v	6	vii	10	vii	8	VI

The hierarchies for the Temperley profile and the profile based on the *Courante* show close agreement, particularly for the diatonic pitch classes.

6 Conclusions and recommendations

6.1 Conclusions

6.1.1 Overview

The conclusions of the investigation will be presented with reference to the original project objectives. These are as follows:

- The development of a user-friendly Matlab application with a Graphical User Interface (GUI) and versatile score representation and parameter options hierarchy for analysing the statistical and tonal properties of symbolic music.
- Determining whether a more optimal combination, compared to the combinations evaluated to date, of pitch class representation, key profile template and algorithm parameters can be derived for the Krumhansl and Schmuckler key-finding algorithm.
- Determining whether a more optimal key profile template can be derived for the Krumhansl and Schmuckler key-finding algorithm using optimization techniques such as direct search methods.

6.1.1.1 Development of a Matlab application with a Graphical User Interface (GUI) for analysing the symbolic music

A substantial part of the research effort has been devoted to the development of a GUI-driven Matlab application for analysing the statistical and tonality properties of music encoded in symbolic format. The research strategy involved a number of aspects, including the following:

- A literature review was conducted to identify the most prominent methods and algorithms applied to date. This study confirmed that statistical indicators of note durations, pitch intervals and pitch classes feature prominently in modern computational music analysis algorithms. The study also showed that the MIDI format enjoys substantial support as an encoding format for the encoding and exchange of music in symbolic format.
- Where necessary, source code available in the public domain was interpreted with the view to confirm the algorithmic details from a computational perspective. This code includes the MIDI toolbox implementation developed by Eerola and Toiviainen (2004a), which contributed to the various algorithms for the statistical analysis of symbolic music, including the distributions of note durations, distributions of pitch intervals and distributions of pitch classes. The MIDI toolbox source code also assisted in clarifying implementation of the Krumhansl and Schmuckler key-finding algorithm (Krumhansl, 1990), durational accent as proposed by Huron and Parncutt (1993) and pitch salience as introduced by Terhardt (1978). The C source code developed by Temperley (1999) for the preference rule approach to key-finding also served as a valuable resource in implementing the software application.
- Mathematical formulations were proposed for the statistical analysis algorithms. From a computational perspective, the descriptions of many of the algorithms given in literature are inadequate. The general lack of attention to formal formulation of these algorithms in mathematical terms represents a serious challenge to the computational modelling of music analysis. This is particularly true in the sense that it can be difficult for independent researchers to reproduce the published results. Also, ambiguous formulations may threaten the validity of the results as well as the resulting conclusions.
- The algorithms for statistical analysis and both the Krumhansl and Schmuckler (Krumhansl, 1990) and the Temperley (1999) key-finding algorithms were implemented in Matlab and successfully validated using custom-designed score material where applicable.
- Suitable data structures were designed to facilitate the diverse requirements of a versatile computational application for music analysis. These include data structures for the score

material and associated metadata targeted in the analysis, the score selection and partitioning scheme, the parameter options used in the various algorithms, analysis results, etc. This aspect of the project, while representative of implementation rather than research, nevertheless required an extensive research component in order to determine the levels of functionality and versatility that is required for successful implementation. As such, it embodies the insight delivered by the literature review with regard to the computational requirements of a software application of this nature.

- A suitable Graphical User Interface (GUI) was developed. This interface facilitates versatile analysis options with regard to score definition, score partitioning, algorithm parameter options, data exchange options and graphical support. The interface was used for all of the research tasks performed for this investigation.
- A number of case studies were conducted and presented to demonstrate the capabilities of the statistical analysis functionality of the software application, particularly with regard to the graphical presentation of results.

The statistical and key-finding algorithms implemented in the software application, apart from delivering one of the objectives of this project, also played a vital role in realising the remainder of the research objectives. The versatile implementations of the distributions of pitch classes and the Krumhansl and Schmuckler key-finding algorithm represent the core of the computational requirements for the remaining research objectives, i.e. searching for a more optimal key-finding parameter set and determining a more optimal key profile. With regard to the other statistical analysis options offered by the application, it is expected that the distributions of pitch intervals will find serious application in emerging tonality analysis algorithms, such as the methodologies proposed by Madsen and Widmer (2007).

Overall, it is concluded that the implementation of the software application has been successful and that it represents an excellent platform for further development and research. Due to the versatile data structures and modular implementation, further developments are expected to be predominantly of an algorithmic and functional rather than a structural nature.

6.1.1.2 Investigating the optimal combination of pitch class representation and key profile template for key-finding algorithms

A number of key-finding simulations were performed using various combinations of pitch class weighing models and key profile templates with the view to find the most optimal combination for the sample material used in the case studies.

The performance of the various pitch class weighing models depends on the both the choice of key template as well as the nature of the test material. The flat pitch class weighing model performed well for the sample material where the analysis partitions contain a low number of note events, e.g. the first four notes of the 24 preludes of Bach's *Well-tempered Clavier Book I* and the measure by measure analysis of the *Courante* from Bach's *Cello Suite in C* and the *Gavotte* from Bach's *French Suite No. 5 in G major (BWV 816)*. Very poor results were achieved with the flat weighing model, however, for the full scores of the *Well-tempered Clavier Book I*. It is shown in Fig. 5.23 that, for tonally rich material, the number of pitch classes contained in the analysis window increases rapidly with increased window size. As a result, the performance of the flat weighing model decreases with increased window size. Based on the above considerations, it is concluded that the flat pitch class weighing model is only suitable for short excerpts of sample material.

The histogram model, linear durational model and the durational accent model performed similarly for the first four notes of the *Well-tempered Clavier* and the *Courante*. The histogram model outperformed the other two durational models for the *Gavotte*, albeit on a low scoring base. The linear durational model and durational accent model, however, achieved significantly better performance figures for the full scores of the *Well-tempered Clavier*.

The linear durational model performed equally or outperformed the durational accent model in all but one case. It is concluded that the increased computational load imposed by the durational accent model, using the default parameter values proposed by Huron and Parncutt (1993), is not justified by increased performance for the case studies considered.

The linear durational and the flat weighing model delivered similar results for the first four notes of the *Well-tempered Clavier*, *Courante* and *Gavotte,* with one identical score, two scores in favour of the linear durational model and three scores in favour of the flat weighing model. The linear durational model, however, outperformed the flat weighing model completely for the full scores of the *Well-tempered Clavier*.

It is important to note that the distributions of note durations shown in Fig. 5.4 and Fig. 5.14 for the *Courante* and *Gavotte* respectively exhibit very little variation. This type of material reflect similar weights for all pitch classes present, thereby causing the results produced by the durational weighing models and flat weighing model to converge. Temperley's choice of this material for evaluating the flat weighing model versus the durational models is therefore considered inappropriate.

Overall, with regards to the pitch class weighing models, it is concluded that the linear durational model performed best for the sample material considered in this investigation. It is also true, however, that the flat weighing model deliver good results for short excerpts and is computationally less intensive.

The key-finding comparison showed that, irrespective of the weighing model, the Temperley key profile outperformed the Krumhansl and Kessler template in most cases. The difference in performance is not surprising, as the weights assigned to the individual pitch classes for the two key profiles exhibit significant differences. The Krumhansl and Kessler profile is derived from tonality cognition experiments while the Temperley profile is based on music theoretical considerations. Temperley focused on the hierarchy rather than numerical values of the key profile coefficients, and assigned the numerical values in a rather arbitrary fashion.

Overall, the key-finding comparisons indicated that it is necessary to derive a more optimal key template before revisiting the issue of pitch class weighing.

6.1.1.3 Deriving a key profile template using direct search methods

A pattern search algorithm was used to estimate the most optimal key profiles for the four test cases, i.e. the full scores of the 24 preludes of Bach's *Well-tempered Clavier Book I*, the first four notes of the 24 preludes of Bach's *Well-tempered Clavier Book I,* the measure by measure analysis of the *Courante* from Bach's *Cello Suite in C major* and the measure by measure analysis of the *Gavotte* from Bach's *French Suite No. 5 in G major (BWV 816)*. The investigation allowed for an extensive range of combinations of the various algorithmic parameters, including the following:

- Three pitch weighing models, namely the flat model, the linear durational model and the durational accent model.
- Key resolutions of 0.1%, 1.0% and 10.0%.
- Three different cost functions.
- Two pattern search polling options.
- Constrained and unconstrained key profile vectors.

The optimization strategy using the pattern search algorithm worked well in practice. Optimised key profiles were derived for the individual test cases that outperforms the Krumhansl and Kessler profile and the Temperley profile substantially, as shown in Table 6.1 to Table 6.4.

Table 6.1 Performance of the Krumhansl and Schmuckler key-finding algorithm for the full scores of the 24 preludes of Bach's Well-tempered Clavier Book I.

Key-profile	**Pitch classes weighing**		
	Flat	**Linear durational**	**Durational accent** ($\tau = 0.5, I_a = 2.0$)
Krumhansl & Kessler	7.64%	91.67%	95.83%
Temperley	3.47%	91.67%	83.33%
Optimised	11.81%	100.00%	100.00%

Table 6.2 Performance of the Krumhansl and Schmuckler key-finding algorithm for the first four notes of the 24 preludes of Bach's Well-tempered Clavier Book I.

Key-profile	**Pitch classes weighing**		
	Flat	**Linear durational**	**Durational accent** ($\tau = 0.5, I_a = 2.0$)
Krumhansl & Kessler	91.67%	91.67%	91.67%
Temperley	87.50%	97.92%	97.92%
Optimised	100%	100%	100%

Table 6.3 Performance of the Krumhansl and Schmuckler key-finding algorithm for the Courante from Bach's Cello Suite in C major.

Key-profile	**Pitch classes weighing**		
	Flat	**Linear durational**	**Durational accent** ($\tau = 0.5, I_a = 2.0$)
Krumhansl & Kessler	62.50%	65.00%	65.00%
Temperley	100.00%	83.75%	83.75%
	100.00%	100.00%	100.00%

Table 6.4 Performance of the Krumhansl and Schmuckler key-finding algorithm for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816).

Key-profile	**Pitch classes weighing**		
	Flat	**Linear durational**	**Durational accent** ($\tau = 0.5, I_a = 2.0$)
Krumhansl & Kessler	64.00%	40.00%	40.00%
Temperley	78.00%	74.00%	68.00%
	88.00%	92.00%	96.00

The average key-finding performance scores of the optimised key profiles were determined for all test cases. Table 6.5 compares the average performances of the top scoring optimised profiles, the Krumhansl and Kessler key profile and the Temperley key profile, using the linear durational weighing model. The results show that the optimised profiles outperform the other two profiles for all key resolutions.

Table 6.5 Average key-finding performances of the optimised, Krumhansl and Kessler and Temperley key profiles for key resolutions of 0.1%, 1.0% and 10.0% with linear durational weighing.

Key profile	**Key resolution**		
	0.1%	**1.0%**	**10.0%**
Gavotte_Unconstrained_1.0_B_2N Courante_Unconstrained_10.0_B_2N	96.33%	96.85%	91.58%
Krumhansl and Kessler	74.69%	75.21%	68.29%
Temperley	87.35%	87.35%	84.97%

The optimised key profiles introduce new major and minor mode pitch class hierarchies. For the major mode, compared to the Krumhansl and Kessler and the Temperley pitch class hierarchies, the optimised profiles rate the supertonic and the subdominant in higher positions, at the expense of the mediant.

6.1.2 Recommendations

The study gives rise to a number of recommendations for further work, including the following:

- The partitioning scheme should be revised to further increase the versatility of the software application. Specifically, the application should make provision to apply different partitioning window settings to the individual scores selected for simultaneous analysis.
- The key-finding performance of the optimised key profiles should be evaluated for material not used in the optimization process. This represents a stringent test that will shed light on the important question of whether a single key profile can in fact represent all of the diverse characteristics that occur in tonal music.
- The pattern search optimization strategy algorithm performed well, but additional estimations, using more diverse score material, should be conducted with the view to improve the optimised key profiles further.
- The performance of the durational accent weighing model should be evaluated for different values of the *saturation duration* and *accent index*. It is in principle possible to use the pattern search algorithm to estimate optimum values for these parameters.
- The most important shortcoming of the Krumhansl and Schmuckler key-finding algorithm is the fact that it does not consider the temporal properties of the sample material. This can addressed to an extent by introducing a weighing function based on the temporal positions of note events in the analysis partitions. Such a strategy will allow more recent events to be assigned higher weights compared to more distant events. The parameters of the weighing function can in principle be optimised using a direct search routine.

References

[1] Ambler, S.W., 2004. *The Object Primer*. 3rd ed. New York: Cambridge University Press.

[2] American Standards Association, 1960. *S1.1-1960 American Standard Acoustical Terminology*. New York: American Standards Association.

[3] Barnes, J., 1979. Bach's Keyboard Temperament: Internal Evidence from the Well-Tempered Clavier. *Early Music,* 7 (2), pp. 236-249.

[4] Brown, H., 1988. The Interplay of Set Content and Temporal Context in a Functional Theory of Tonality Perception. *Music Perception: An Interdisciplinary Journal*, 5 (3), pp. 219-249.

[5] Brown, J.C., 1993. Determination of meter of musical scores by autocorrelation. *Journal of Acoustical Society of America*, 94(4), 1953-1957.

[6] Butler, D., 1983. The Initial Identification of Tonal Cues in Music. In: J. Sloboda and Rogers, eds. 1983. *Acquisition of Symbolic Skills*. New York: Plenum Press.

[7] Butler, D., 1989. Describing the Perception of Tonality in Music: A Critique of the Tonal Hierarchy Theory and a Proposal for a Theory of Intervallic Rivalry. *Music Perception*, 6 (3), pp. 219-242.

[8] Castan, G., Good, M. and Roland, P., 2001. Extensible Markup Language (XML) for Music Applications: An Introduction. In: W. B. Hewlett and E. Selfridge-Field, eds. 2001. *The Virtual Score: Representation, Retrieval, Restoration*. Cambridge, MA: MIT Press, pp. 95-102.

[9] Chew, E., 2000. *Towards a Mathematical Model of Tonality*. Ph.D. Massachusetts Institute of Technology.

[10] Chew, E., 2001. Modelling Tonality: Applications to Music Cognition. *23rd Annual Conference of the Cognitive Science Society*, Edinburgh, United Kingdom 1-4 August 2001.

[11] Classical Music on Classical Archives, 2011. *Classical Music on Classical Archives: Home*. [online] Available at: < http://www.classicalarchives.com/> [Accessed 27 August 2011].

[12] Cohen, A.J., 1977. Tonality and Perception: Musical Scales Prompted by Excerpts from Das Wohl temperierte Clavier of JS Bach. *Second Workshop on Physical and Neuropsychological Foundations of music*, Ossiach: Austria.

[13] Cohn, R., 1997. Neo-riemannian operations, parsimonious trichords, and their "tonnetz" representations. *Journal of Music Theory*, 41 (1), pp. 1–66.

[14] Cook, N., 1987. The perception of large-scale tonal closure. *Music Perception: An Interdisciplinary Journal*, 5 (2), pp. 197-205.

[15] Dahlhaus, C., 1990. Studies on the origin of Harmonic Tonality. Translated by R.O. Gjerdingen. Princeton, N.J.: Princeton University Press.

[16] Eerola, T. and North, A. C., 2000. Expectancy-based model of melodic complexity. In C. Woods, G.B. Luck, R. Brochard, S.A. O'Neill and J.A. Sloboda, eds. 2000. *Proceedings of the Sixth International Conference on Music Perception and Cognition (ICMPC 2000)*. 5-10 August 2000. Keele, Staffordshire, UK: European Society for the Cognitive Sciences of Music.

[17] Eerola, T. and Toiviainen, P., 2004a. *MIDI Toolbox: MATLAB Tools for Music Research*. Kopijyvä, Jyväskylä, Finland: University of Jyväskylä. Electronic version available from: http://www.jyu.fi/musica/miditoolbox/.

[18] Eerola, T. and Toiviainen, P., 2004b. MIR in Matlab: The Midi Toolbox. In: *Proceedings of 5th International Conference on Music Information Retrieval (ISMIR 2004)*. Barcelona, Spain 10-14 October 2004. Barcelona, Spain: Universitat Pompeu Fabra.

[19] Ferrara, L., 1991. *Philosophy and the Analysis of Music*. New York: Greenwood Press.

[20] Goldman, R., 1961. "Varèse: *Ionisation*; *Density 21.5*; *Intégrales*; *Octandre*; *Hyperprism*; *Poème Electronique*. Instrumentalists, cond. Robert Craft. Columbia MS 6146 (stereeo)" (in Reviews of Records). *Musical Quarterly* 47 (1). January: 133–134.

[21] Gollin, E., 1998. Some Aspects of Three-Dimensional Tonnetze. *Journal of Music Theory*, 42 (2), pp. 195–206.

[22] Good, M., 2001a. MusicXML for Notation and Analysis. In: W. B. Hewlett and E. Selfridge-Field, eds. 2001. *The Virtual Score: Representation, Retrieval, Restoration*. Cambridge, MA: MIT Press, pp. 113-124.

[23] Good, M., 2001b. MusicXML: An Internet-Friendly Format for Sheet Music. In: *Proceedings XML 2001.* Orlando, United States 9-14 December 2001.

[24] Good, M., 2006. Lessons from the Adoption of MusicXML as an Interchange Standard. In: *Proceedings XML 2006*. Boston, MA 5-7 December 2006.

[25] Good, M. and Actor, G., 2003. Using MusicXML for File Interchange. In: Proceedings *Third International Conference on WEB Delivering of Music*. Leeds, UK, 15-17 September 2003. Los Alamitos, CA: IEEE Press.

[26] Grabe, E. and Low, E.L., 2002. Durational variability in speech and the rhythm class hypothesis. In: C. Gussenhoven and N. Warner, eds. Papers in Laboratory Phonology, Volume 7, pp. 377–401. Berlin: Mouton de Gruyter. (Available on-line: http://www.phon.ox.ac.uk/_esther/).

[27] Grande, C., 1997. The Notation Interchange File Format: A Windows-Compliant Approach. In: E. Selfridge-Field, ed. 1997. *Beyond MIDI: The Handbook of Musical Codes*. Cambridge, MA: MIT Press.

[28] Helmholtz, H.L.F., 1954. *On the Sensations of Tone as a Physiological Basis for the Theory of Music*. Translated by A.J. Ellis. New York: Dover.

[29] Holtzman, S.R., 1977. .A Program for Key Determination, *Interface*, 6, pp. 29-56.

[30] Huron, D. and Parncutt, R., 1993. An Improved Model of Tonality Perception Incorporating Pitch Salience and Echoic Memory. *Psychomusicology*, 12 (2), pp. 154-171.

[31] Judd, C., 1998. Introduction: Analyzing Early Music. In: C. Judd, ed. 1998. *Tonal Structures of Early Music*. New York: Garland Publishing.

[32] Kohonen, T., 1995. *Self-Organizing Maps*. Berlin: Springer-Verlag.

[33] Kolda, T.G., Lewis, R. M. and Torczon, V., 2006. A Generating Set Direct Search Augmented Lagrangian Algorithm for Optimization with a Combination of General and Linear Constraints. *Technical Report SAND-5315*, August 2006. Sandia National Laboratories.

[34] Kostka, S., 1995. *Instructor's Manual to accompany Tonal Harmony*. New York: McGraw-Hill.

[35] Kostka, S. and Payne, D., 1995a. *Tonal Harmony*. New York: McGraw-Hill.

[36] Kostka, S. and Payne, D., 1995b. *Workbook for Tonal Harmony*. New York: McGraw-Hill.

[37] Krumhansl, C.L., 1979. The psychological representation of musical pitch in a tonal context. *Cognitive Psychology*, 11, 1979, pp. 346-374.

[38] Krumhansl, C.L., 1986. Tonal and harmonic hierarchies. In: J. Sundberg, ed. 1986. *Harmony and tonality. Music Publication No. 54 1987*. Stockholm: Royal Swedish Academy of Music.

[39] Krumhansl, C.L., 1990. *Cognitive Foundations of Musical Pitch*. New York: Oxford University Press.

[40] Krumhansl, C.L., 2004. The Cognition of Tonality - as We Know it Today. *Journal of New Music Research*, 33 (3), pp. 253-268.

[41] Krumhansl, C.L. and Kessler, E.J., 1982. Tracing the Dynamic Changes in Perceived Tonal Organization in a Spatial Representation of Musical Keys. *Psychological Review*, 89, pp. 334-368.

[42] Krumhansl, C.L. and Shepard, R., 1979. Quantifications of the hierarchy of tonal functions within a diatonic context. *Journal of Experimental Psychology: Human Perception and Performance*, 5, pp. 579-594.

[43] Kuipers, G. and Good, M., 2004. Using XML Technologies in Music. In: *Proceedings 37th Midwest Instruction and Computing Symposium (MICS 2004)*. Morris, MN, 16-17 April 2004.

[44] Lakner, Y., 1960. A new method of representing tonal relations. *Journal of Music Theory*, 4, pp. 194-209.

[45] Lee, C. S., 1991. The perception of metrical structure: Experimental evidence and a model. In: P. Howell, R. West, and I. Cross, eds. 1991. *Representing musical structure*. London: Academic Press. pp. 59-127.

[46] Lerdahl, F., 2001. *Tonal Pitch Space*. Cambridge, MA: MIT Press.

[47] Lerdahl, F., 1996. Calculating tonal tension. *Music Perception, 13* (3), pp. 319-363.

[48] Lerdahl, F. and Jackendoff, R., 1983. *A generative theory of tonal music*. Cambridge MA: MIT Press.

[49] Lewin, D., 1987. *Generalized Musical Intervals and Transformations*. New Haven: Yale University.

[50] Longuet-Higgins, H. C. and Lee, C. S., 1982. The Perception of Musical Rhythms. *Perception*, 11, pp. 115-128.

[51] Longuet-Higgins, H. C. and Steedman, M. J., 1971. On Interpreting Bach. *Machine Intelligence*, 6, pp. 221-241.

[52] Madsen, S.T. and Widmer, G., 2007. Key-finding with Interval Profiles. In: *Proceedings of the International Computer Music Conference (ICMC'07)*, Copenhagen, Denmark, 27-31 August 2007.

[53] MathWorks™, 2009. *The MathWorks Genetic Algorithm and Direct Search Toolbox™ 2 User's Guide*, September 2009 Online only Revised for Version 2.4.2 (Release 2009b).

[54] Massaro, D.W., 1970. Preperceptual auditory images. *Journal of Experimental Psychology*, 85, pp. 411-417.

[55] Midi Toolbox, 2011. *Midi Toolbox*. [online] Available at: <https://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/miditoolbox/> [Accessed 10 October 2009].

[56] Musopen, 2011. *About|Musopen*. [online] Available at: < http://musopen.org/about/> [Accessed 27 August 2011].

[57] Narmour, E., 1990. *The analysis and cognition of basic melodic structures*. Chicago: University of Chicago Press.

[58] Nattiez, J., 1990. *Music and Discourse: Toward a Semiology of Music*. Translated by C. Abbate. Princeton: Princeton University Press

[59] Neisser, U., 1967. *Cognitive psychology*. New York: Appleton-Century-Crofts.

[60] Object Management Group, 2003. *Unified Modeling Language: Superstructure, Version 2.0*. OMG Document Formal/05-07-04. Needham, MA: Object Management Group. 2005.

[61] Parncutt, R., 1994. A perceptual model of pulse salience and metrical accent in musical rhythms. *Music Perception*, 11 (4), pp. 409-464.

[62] Partch, H., 1974. *Genesis of a Music*. New York: Da Capo Press.

[63] Povel, D.J. and Essens, P., 1985. Perception of temporal patterns. *Music Perception*, 2 (4), pp. 411-440.

[64] Povel, D.J. and Okkerman, H., 1981. Accents in equitone sequences. *Perception and Psychophysics*, 30, pp. 565-572.

[65] Purwins, H., Blankertz, B. and Obermayer, K., 2007. Toroidal Models in Tonal Theory and Pitch-Class Analysis. In: *Tonal Theory for the Digital Age: Computing in Musicology 15*. Stanford: CCARH, Stanford University, pp. 73–98.

[66] Rameau, J.P., 1971. *Treatise on Harmony*. Translated by P. Gossett. New York: Dover.

[67] Recordare, 2011. *MusicXML Software | Recordare*. [Online] Available at: <http://www.recordare.com/musicxml/community/software> [Accessed 30 August 2011].

[68] Rees, P., 2001. Synchronisation and SMPTE timecode (time code), [online] Available at: < http://www.philrees.co.uk/articles/timecode.htm > [Accessed 27 August 2011].

[69] Rosenthal, D., 1992, Emulation of human rhythm perception. *Computer Music Journal*, 16 (1), pp. 64--76.

[70] Sadie S., 1980. *The New Grove Dictionary of Music and Musicians*. London: Macmillan Publishers Limited.

[71] Schmuckler, M.A. and Tomovski, R., 2005. Perceptual Tests of an Algorithm for Musical Key-Finding. *Journal of Experimental Psychology: Human Perception and Performance*, 31 (5), pp. 1124-1149.

[72] Schoenberg, A., 1969. *Structural Functions of Harmony*. Translated by L. Stein. New York: Norton.

[73] Sethares, W. A., 2005. *Tuning, timbre, spectrum, scale*. 2nd ed. London: Springer.

[74] Shepard, R., 1964. Circularity in judgments of relative pitch. *Journal of the Acoustical Society of America*, 36, pp. 2346-2353.

[75] Shepard, R., 1982a. Geometrical approximations to the structure of musical pitch. *Psychological Review*, 89 (4), pp. 305-333.

[76] Shepard, R., 1982b. Structural representations of musical pitch. In: D. Deutsch, ed. 1982. *The psychology of music*. London: Academic Press, pp. 343-390.

[77] Simonton, D. K., 1984. Melodic structure and note transition probabilities: A content analysis of 15,618 classical themes. *Psychology of Music*, 12, pp. 3-16.

[78] Sloan, D., 1997. HyTime and Standard Music Description Language: A Document-Description Approach. In: E. Selfridge-Field, ed. 1997. *Beyond MIDI: The Handbook of Musical Codes*. Cambridge, MA: MIT Press.

[79] Temperley, D., 1999. What's Key for Key?: The Krumhansl-Schmuckler Key-Finding Algorithm Reconsidered. *Music Perception*, 17 (1), pp. 65-100.

[80] Temperley, D., 2002. *The Cognition of Basic Musical Structures*. Cambridge: MIT Press.

[81] Terhardt, E., 1977. The two-component theory of musical consonance. In: E. Evans and J. Wilson, eds. 1977. *Psychophysics and physiology of hearing*. London: Academic Press, pp. 381- 390.

[82] Terhardt, E., 1978. Psychoacoustical evaluation of musical sounds. *Perception and Psychophysics*, 23 (6), pp. 483-492.

[83] Terhardt, E., 1979. Calculating virtual pitch. *Hearing Research*, 1, pp. 155-182.

[84] Terhardt, E., Stoll, G. and Seewann, M., 1982a. Pitch of complex signals according to virtual-pitch theory: test, examples, and predictions. *Journal of the Acoustical Society of America,* 71, pp. 671-678.

[85] Terhardt, E., Stoll, G. and Seewann, M., 1982b. Algorithm for extraction of pitch and pitch salience from complex tonal signals. *Journal of the Acoustical Society of America*, 71, pp. 679-688.

[86] The International MIDI Association, 1990. *Standard MIDI*-File Format *Spec*. 1.1. Los Angeles: The International MIDI Association.

[87] The International MIDI Association, 1996. *Standard MIDI-File Format Spec. 1.1.* Los Angeles: The International MIDI Association.

[88] The Just Intonation Network, 2011. *The Just Intonation Network: Home*. [Online] Available at: <http://www.justintonation.net/> [Accessed 14 September 2011].

[89] The MIDI Manufacturers Association, 1996. *The Complete MIDI 1.0 Detailed Specification*. Document version 96.1. Los Angeles: The MIDI Manufacturers Association.

[90] Thomassen, M. T., 1982. Melodic accent: experiments and a tentative model. *Journal of the Acoustical Society of America,* 71, pp. 1596-1605.

[91] Toiviainen, P., 2008. Visualization of tonal content in the symbolic and audio domains. In: W. Hewlett and E. Selfridge-Field, eds. 2008. *Tonal theory for the digital age: Computing in Musicology 15*. Stanford: CCARH, Stanford University, pp. 187-199.

[92] Toiviainen, T. and Krumhansl, C.L., 2003. "Measuring and Modeling Real-Time Responses to Music: The Dynamics of Tonality Induction, *Perception*, 2 (6), pp. 741–66.

[93] Von Hippel, P., 2000. Redefining pitch proximity: Tessitura and mobility as constraints on melodic interval size. *Music Perception*, 17 (3), pp. 315-327.

[94] von Oettingen, Arthur J., 1866. Harmoniesystem in dualer Entwickelung. Dorpat, Livonia.

[95] Wikipedia, the free encyclopedia, 2011a. *Boardwalk Hall Auditorium Organ, Wikipedia, the free encyclopedia*. [online] Available at: < http://en.wikipedia.org/wiki/Boardwalk_Hall_Auditorium_Organ#Diaphone-Dulzian> [Accessed 29 August 2011].

[96] Wikipedia, the free encyclopedia, 2011b. *Equal temperament - Wikipedia, the free encyclopedia*. [online] Available at: <http://en.wikipedia.org/wiki/Equal_temperament> [Accessed 22 October 2011].

[97] Wikipedia, the free encyclopedia, 2011c. *Pythagorean tuning - Wikipedia, the free encyclopedia*. [Online] Available at: <http://en.wikipedia.org/wiki/Pythagorean_tuning> [Accessed 14 September 2011].

Appendix A

Musical excerpts and pitch strings used by Brown

Fig. A1 shows the nine musical excerpts, labelled 1M to 9M, and corresponding pitch strings labelled A, B and C, used by Brown (1988).

Fig. A1   Musical excerpts and pitch strings used by Brown (1988).

Appendix B

Key-finding performance of the Krumhansl and Schmuckler algorithm for the first four notes of the forty-eight preludes of the Well-Tempered Clavier composed by Bach

Table B1 summarises the key-finding results obtained with the Krumhansl and Schmuckler algorithm for the first four notes of forty-eight preludes of the *Well-Tempered Clavier* composed by Bach.

Table B1 Key finding performance of the Krumhansl and Schmuckler algorithm for the first four notes of the forty-eight preludes of the Well-tempered Clavier composed by Bach (Krumhansl, 1990).

Prelude	**Designated key**	**Book I**	**Book II**
		Identified key	**Identified key**
2	C minor	C minor	C major
3	$C^{\#}$ major	$C^{\#}$ major	$C^{\#}$ major
4	$C^{\#}$ minor	$C^{\#}$ minor	$C^{\#}$ minor
5	D major	D major	D major
6	D minor	D minor	D minor
7	E_b major	E_b major	G Minor
8	$D^{\#}$ minor	$D^{\#}/E_b$ minor	$D^{\#}/E_b$ minor
9	E major	E major	E major
10	E minor	E minor	E minor
11	F major	F major	F major
12	F minor	F minor	F minor
13	$F^{\#}$ major		$F^{\#}$ major
14	$F^{\#}$ minor	$F^{\#}$ minor	$F^{\#}$ minor
15	G major	G major	G major
16	G minor	G minor	
17	A_b major	A_b major	A_b major
18	$G^{\#}$ minor	$G^{\#}$ minor	$G^{\#}$ minor
19	A major	A major	A major
20	A minor	A minor	A minor
21	B_b major	B_b major	B_b major
22	B_b minor	B_b minor	B_b minor
23	B major	B major	B major
24	B minor	B minor	B minor
■ Key identification successful. ■ Key identification unsuccessful.			

Appendix C

Key-finding performance of the Krumhansl and Schmuckler algorithm for the first four notes of the twenty-four preludes composed by Shostakovich

Table C1 summarises the key-finding results obtained with the Krumhansl and Schmuckler algorithm for the first four notes of the twenty-four preludes composed by Shostakovich.

Table C1 Key-finding performance of the Krumhansl and Schmuckler algorithm for the first four notes of the twenty-four preludes composed by Shostakovich (Krumhansl, 1990).

Prelude	**Designated key**	**Prelude**	**Designated key**
1	C major	13	$F^{\#}$ major
2	A minor	14	E_b minor
3	G major	15	D_b major
4	E minor	16	B_b minor
5	D major	17	A_b major
6	B minor	18	F minor
7	A major	19	E_b major
8	$F^{\#}$ minor	20	C minor
9	E major	21	B_b major
10	$C^{\#}$ minor	22	G minor
11	B major	23	F major
12	$G^{\#}$ minor	24	D minor

Key identification successful.
Key identification unsuccessful.

Appendix D

Key-finding performance of the Krumhansl and Schmuckler algorithm for the first four notes of the twenty-four preludes composed by Chopin

Table D1 summarises the key-finding results obtained with the Krumhansl and Schmuckler algorithm for the first four notes of the twenty-four preludes composed by Chopin.

Table D1 Key-finding performance of the Krumhansl and Schmuckler algorithm for the first four notes of the twenty-four preludes composed by Chopin (Krumhansl, 1990).

Prelude	Designated key	Prelude	Designated key
1	C major	13	$F^{\#}$ major
2	A minor	14	E_b minor
3	G major	15	D_b major
4	E minor	16	B_b minor
5	D major	17	A_b major
6	B minor	18	F minor
7	A major	19	E_b major
8	$F^{\#}$ minor	20	C minor
9	E major	21	B_b major
10	$C^{\#}$ minor	22	G minor
11	B major	23	F major
12	$G^{\#}$ minor	24	D minor

■ Key identification successful.
■ Key identification unsuccessful.

Appendix E

Comparison of the key-finding performance of the Krumhansl and Schmuckler and the Longuet-Higgins and Steedman algorithms for the forty-eight fugues of the Well-Tempered Clavier composed by Bach

Table E1 compares the key-finding results obtained with the Krumhansl and Schmuckler and the Longuet-Higgins and Steedman algorithms for the forty-eight fugues of the *Well-tempered Clavier* composed by Bach.

Table E1 Comparison of the key-finding performance of the Krumhansl and Schmuckler and the Longuet-Higgins and Steedman algorithms for the forty-eight fugues of the Well-tempered Clavier composed by Bach (Krumhansl, 1990).

Fugue	**Designated key**	**Number of notes**			
		Book I		**Book II**	
		Krumhansl/ Schmuckler	**Longuet-Higgins/ Steedman**	**Krumhansl/ Schmuckler**	**Longuet-Higgins/ Steedman**
1	C major	2	16	4	23
2	C minor	5	5	5	9
3	$C^{\#}$ major	7	16	2	4
4	$C^{\#}$ minor	3	4	12	12
5	D major	2	15	10	9
6	D minor	3	8	3	15
7	E_{b} major	6	11	2	20
8	$D^{\#}$ minor	6	12	15	9
9	E major	12	11	2	6
10	E minor	2	7	15	18
11	F major	10	6	4	17
12	F minor	15	4	5	7
13	$F^{\#}$ major	2	8	5	12
14	$F^{\#}$ minor	18	5	3	18
15	G major	2	15	4	16
16	G minor	3	4	18	18
17	A_{b} major	2	7	12	22
18	$G^{\#}$ minor	5	5	3	25
19	A major	4	7	2	20
20	A minor	5	5	9	5
21	B_{b} major	4	14	2	9
22	B_{b} minor	3	6	3	5
23	B major	11	11	2	12
24	B minor	3	7	3	6

■ Tonic-dominant rule invoked – Key identification successful.
■ Tonic-dominant rule invoked – Key identification unsuccessful.

Appendix F

Key-finding performance of the Krumhansl and Schmuckler algorithm for the twenty-four fugues composed by Shostakovich

Table F1 summarises the key-finding results obtained with the Krumhansl and Schmuckler algorithm for the twenty-four fugues composed by Shostakovich.

Table F1 Key-finding performance of the Krumhansl and Schmuckler algorithm for the twenty-four fugues composed by Shostakovich (Krumhansl, 1990).

Fugue	Designated key	Number of notes	Fugue	Designated key	Number of notes
1	C major	3	13	$F^{\#}$ major	2
2	A minor	2	14	E_b minor	9
3	G major	21	15	D_b major	2
4	E minor	13	16	B_b minor	3
5	D major	7	17	A_b major	3
6	B minor	15	18	F minor	2
7	A major	2	19	E_b major	2
8	$F^{\#}$ minor	22	20	C minor	3
9	E major	2	21	B_b major	2
10	$C^{\#}$ minor	2	22	G minor	2
11	B major	3	23	F major	2
12	$G^{\#}$ minor	2	24	D minor	3

■ Tonic-dominant rule invoked – Key identification successful.
■ Tonic-dominant rule invoked – Key identification unsuccessful.

Appendix G

Key modulation tracking using the Temperley algorithm for the Gavotte from Bach's French Suite No. 5 in G major *(BWV 816)*

Fig. G1 shows the key-finding results obtained by Temperley (1999) for the *Gavotte* from Bach's *French Suite No. 5 in G major (BWV 816)*. In comparing the key modulation results shown in Fig. G1 with an analysis based on music theoretical principles, Temperley concluded that the algorithm tracks the key modulations correctly.

Fig. G1 Key modulation tracking using the Temperley algorithm for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816) (Temperley 1999).

Appendix H

Overview of Unified Modelling Language

Fig. H1 shows a hierarchy of the diagrams specified in the UML standard (Object Management Group, 2003). This hierarchy identifies two main diagram classes, namely structure diagrams and behaviour diagrams. Structure diagrams document the architecture of a software system, focusing on the components that constitute the system as a departure point.

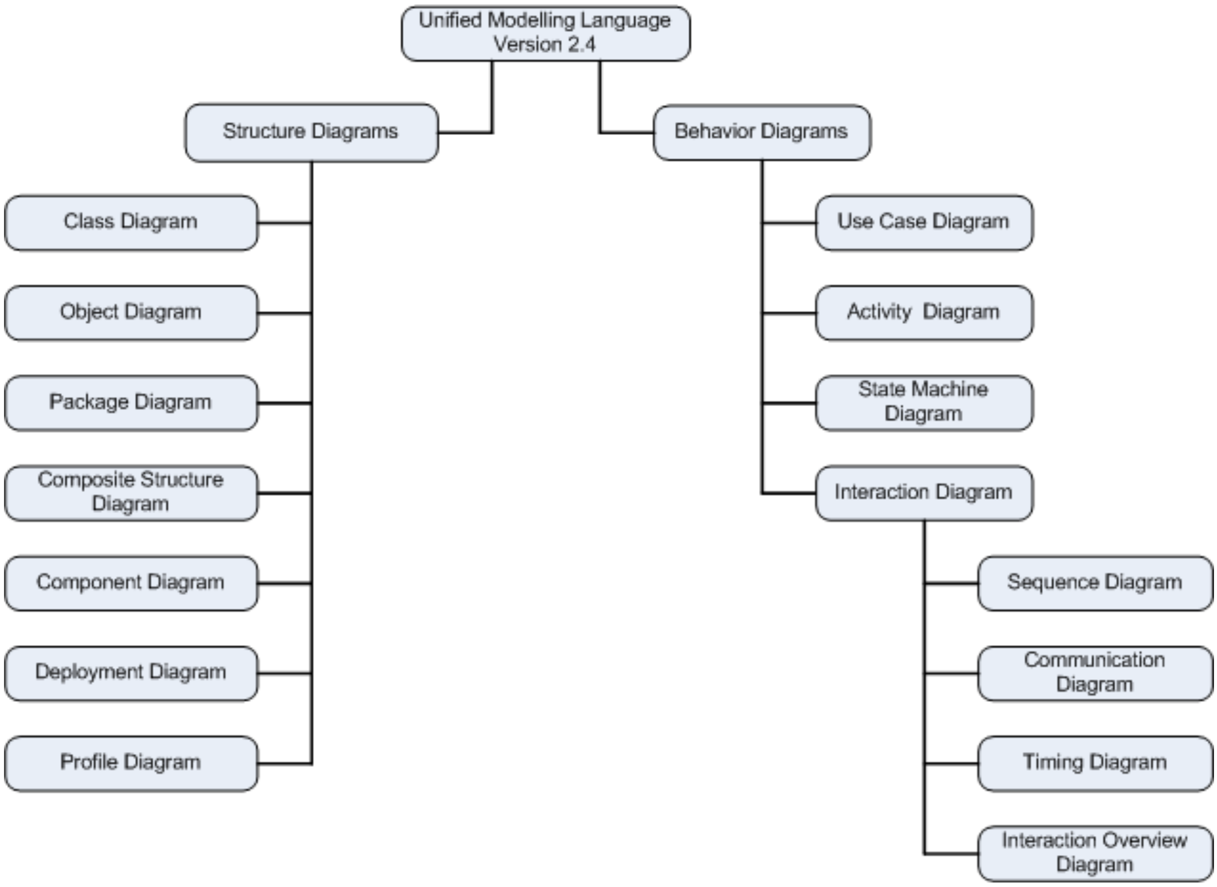


Fig. H1 Hierarchy of UML diagrams

Structure diagrams include the following:

- *Class diagram*: A class diagram describes the structure of a system in terms of classifiers, i.e. classes and interfaces. Attributes, constraints and operations of, and the relationships between the classifiers are shown.
- *Object diagram*: Object diagram shows object instances. The diagram has become obsolete and is not defined in UML 2.4 specification.
- *Package diagram*: A package diagram describes how a system is comprised of logical groupings, i.e. packages, and shows the dependencies among these packages.
- *Composite structure diagram*: A composite structure diagram describes the internal structure of a classifier and the collaborations or behaviours facilitated by the structure.
- *Component diagram*: A component diagram describes how a software system is comprised of components and shows the dependencies among these components.
- *Deployment diagram*: A deployment diagram describes how the software artefacts are deployed on the hardware architecture.

- *Profile diagram*: A profile diagram allows defining custom stereotypes, tagged values and constraints. It thus provides an extension mechanism to the UML standard.

Behaviour diagrams document the processes of a software system, thereby describing the functionality of the system. Behaviour diagrams include the following:

- *Use case diagram*: A use case diagram describes the functional set of actions provided by the system in collaboration with external users, designated as actors in UML terminology.
- *Activity diagram*: An activity diagram describes the sequence and conditions for behaviours or workflows. An activity diagram shows the overall flow of control and objects.
- *State machine diagram:* A state-machine diagram describes the states and state transitions of the system.
- *Interaction diagram*: Interaction diagrams have four subclasses, namely *Sequence* diagrams, *Communication* diagrams, *Timing* diagrams and *Interaction Overview* diagrams.

Appendix I

Supplementary statistical analysis results for the test corpus

This appendix presents supplementary statistical results obtained with the software application for the test corpus, i.e. the full scores and first 4 notes for the 24 preludes of Bach's *Well-tempered Clavier Book I*, the *Courante* from Bach's *Cello Suite in C major* and the *Gavotte* from Bach's *French Suite No. 5 in G major (BWV 816)*.

I1 Preludes of Bach's Well-tempered Clavier Book I

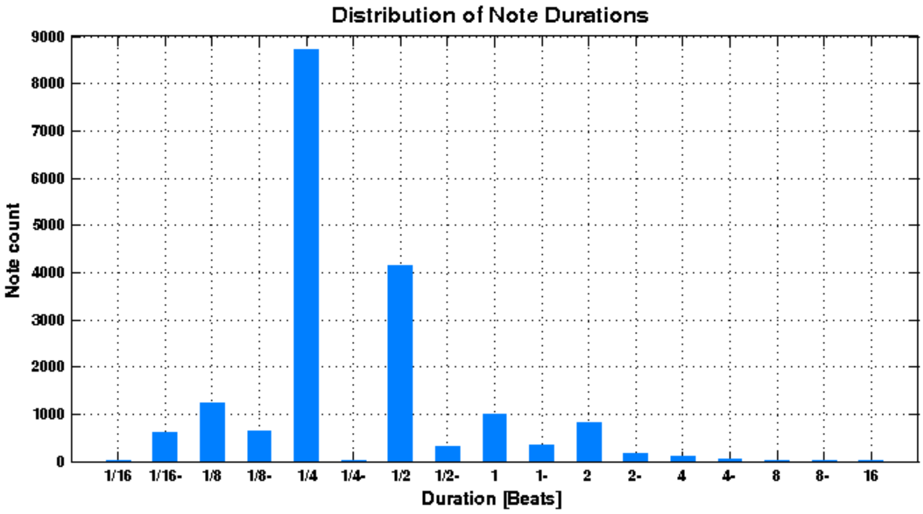


Fig. I1 Distribution of note durations for the 24 preludes of Bach's Well-tempered Clavier Book I.

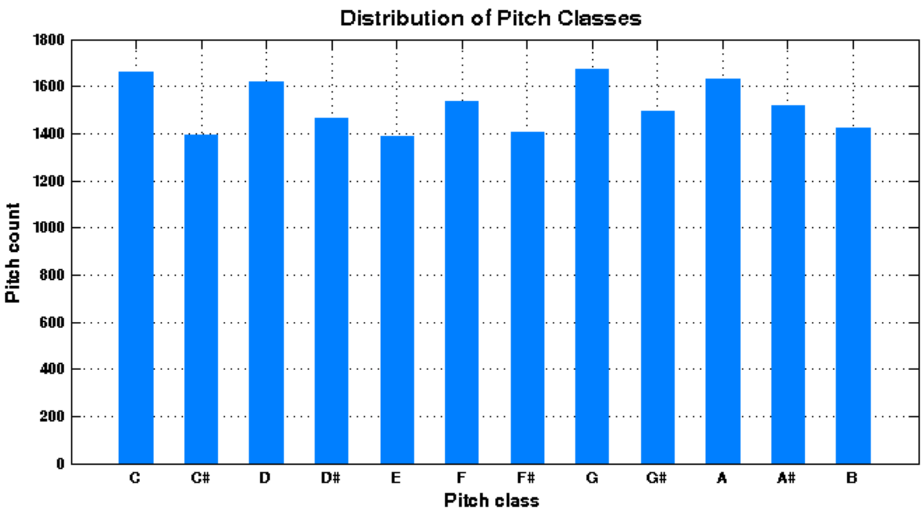


Fig. I2 Distribution of pitch classes for the 24 preludes of Bach's Well-tempered Clavier Book I.

I2 Courante from Bach's Cello Suite in C major:

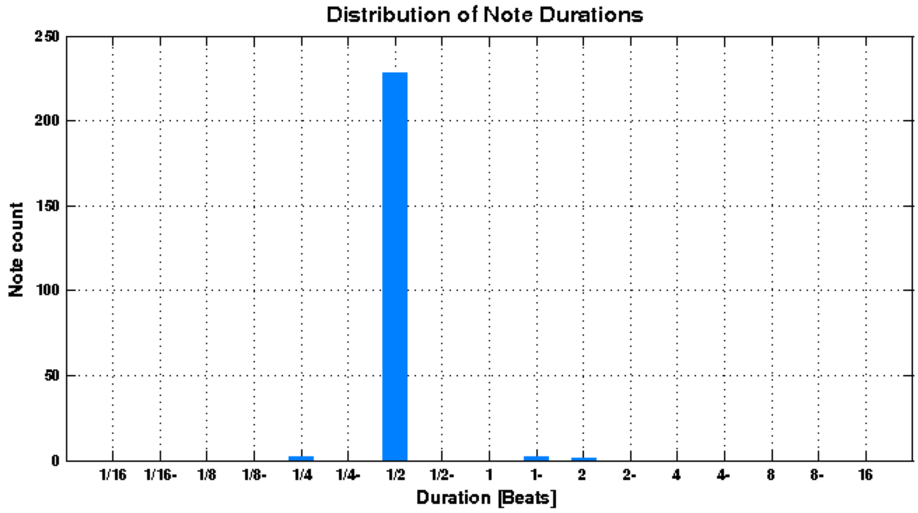


Fig. I3 Distribution of note durations for the Courante from Bach's Cello Suite in C major.

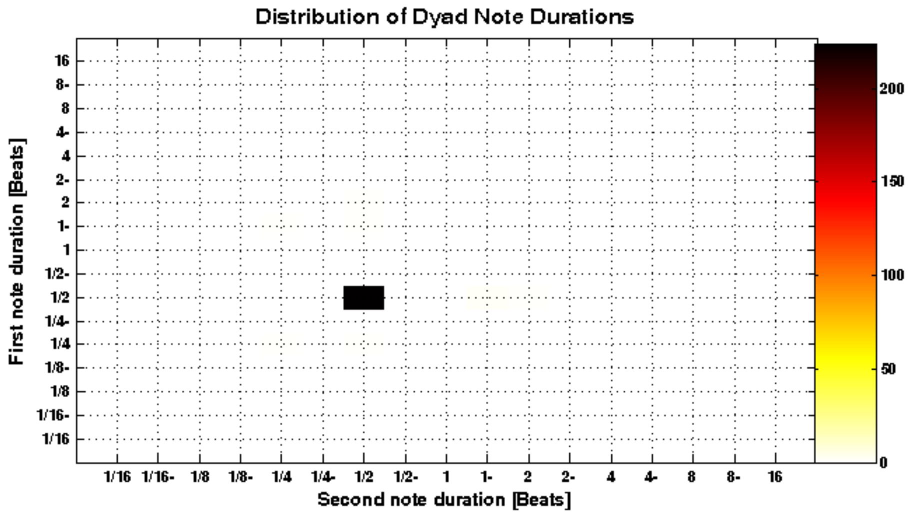


Fig. I4 Distribution of dyad note durations for the Courante from Bach's Cello Suite in C major.

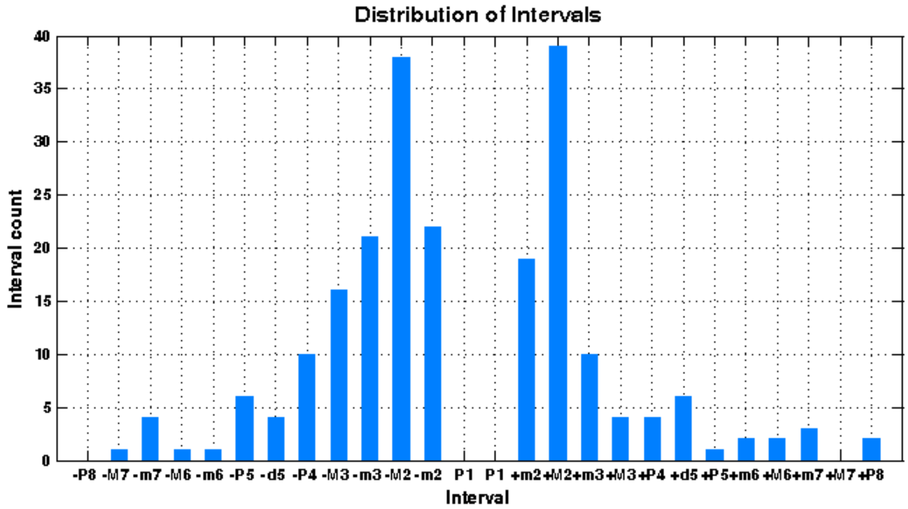


Fig. I5 Distribution of pitch intervals for the Courante from Bach's Cello Suite in C major.

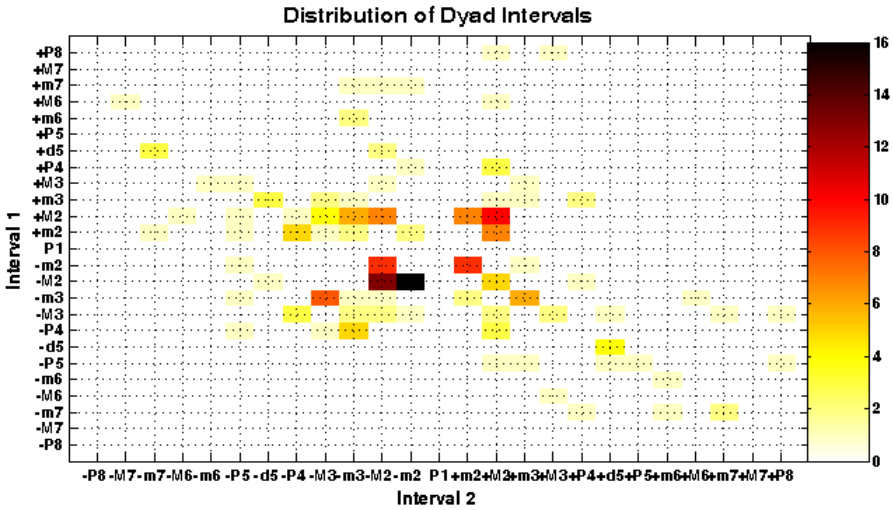


Fig. I6 Distribution of dyad pitch intervals for the Courante from Bach's Cello Suite in C major.

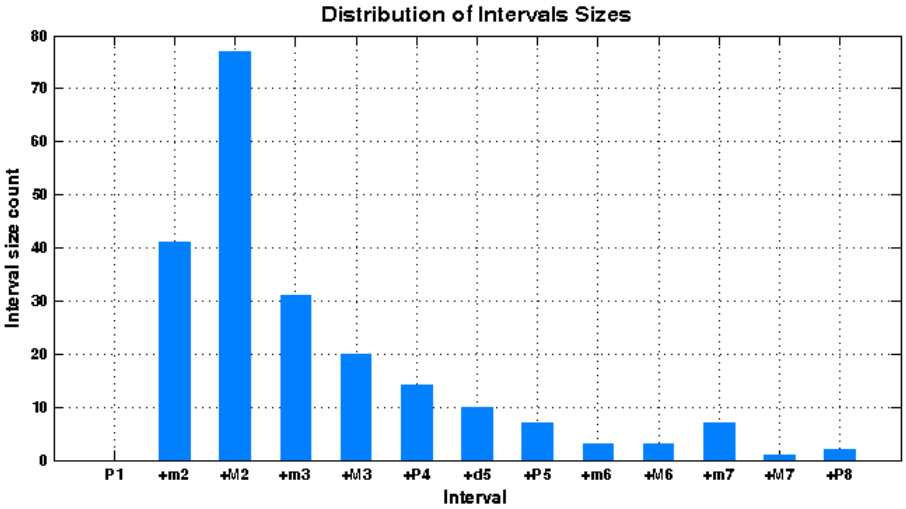


Fig. I7 Distribution of pitch intervals sizes for the Courante from Bach's Cello Suite in C major.

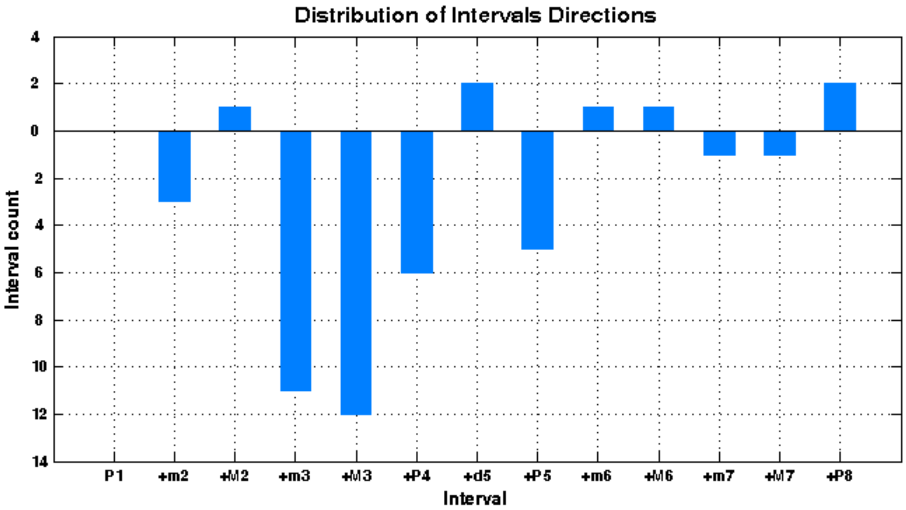


Fig. I8 Distribution of pitch intervals directions for the Courante from Bach's Cello Suite in C major.

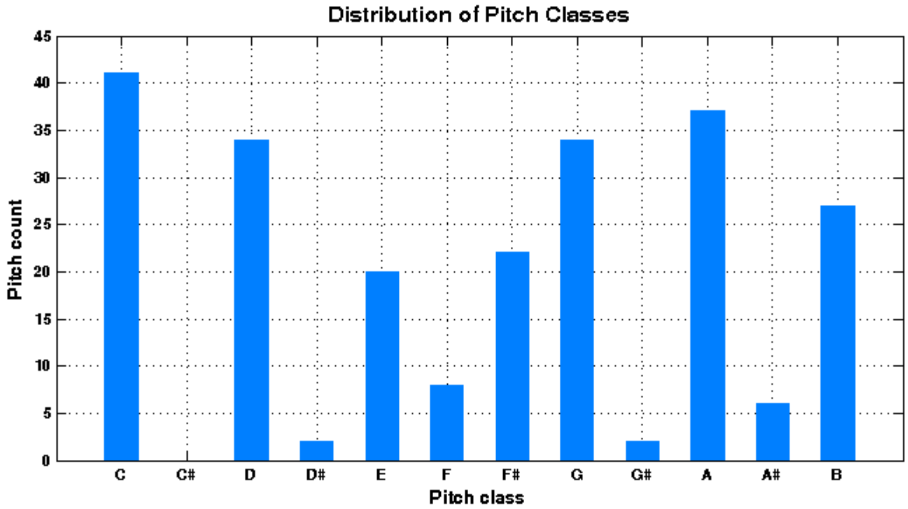


Fig. I9 *Distribution of pitch classes for the Courante from Bach's Cello Suite in C major.*

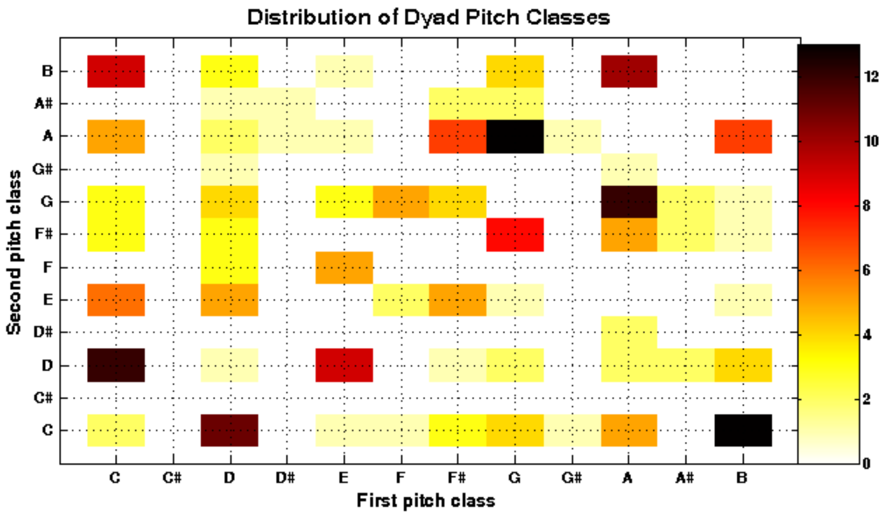


Fig. I10 *Distribution of dyad pitch classes for the Courante from Bach's Cello Suite in C major.*

I3 Gavotte from Bach's French Suite No. 5 in G major (BWV 816)

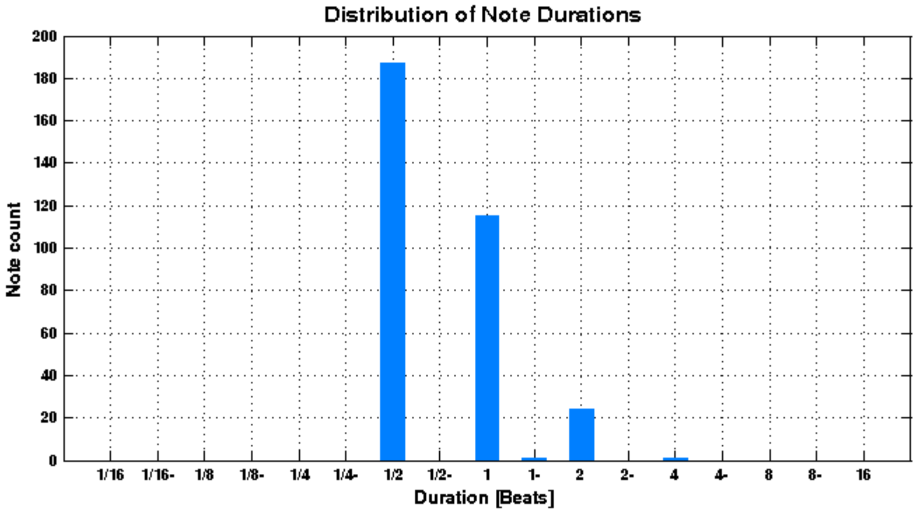


Fig. I11 Distribution of note durations for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816).

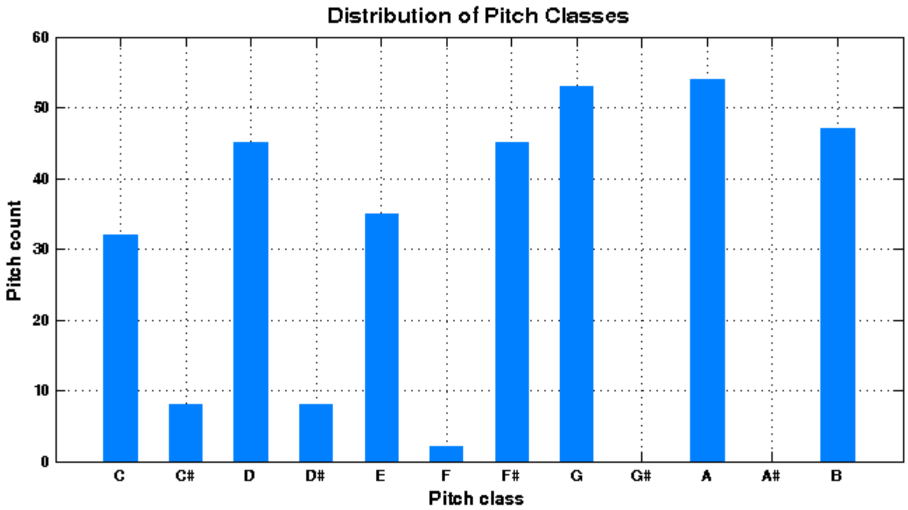


Fig. I12 Distribution of pitch classes for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816).

Appendix J

Key-finding results of the Krumhansl and Schmuckler algorithm

This appendix presents the key-finding results obtained with the software application for the full scores of the 24 preludes of Bach's *Well-tempered Clavier book I*, the first four notes of the 24 preludes of Bach's *Well-tempered Clavier Book I*, the *Courante* from Bach's *Cello Suite in C major* and the *Gavotte* from Bach's *French Suite No. 5 in G major (BWV 816)*.

Table J1 Key-finding results of the Krumhansl and Schmuckler algorithm for the full scores of the 24 preludes of Bach's Well-tempered Clavier Book I.

Prelude	**Reference**	**KKI**	**KKII**	**KKIII**	**KKIV**	**TI**	**TII**	**TIII**	**TIV**
1	CMaj	CMaj	*	CMaj	CMaj	CMaj	*	CMaj	CMaj
2	CMin	CMin	*	CMin	CMin	CMin	*	CMin	CMin
3	C#Maj	C#Maj	*	C#Maj	C#Maj	C#Maj	*	C#Maj	C#Maj
4	C#Min	C#Min	*	C#Min	C#Min	C#Min	*	C#Min	C#Min
5	DMaj	DMaj	*	DMaj	DMaj	DMaj	*	DMaj	DMaj
6	DMin	DMin	*	DMin	DMin	DMin	*	DMin	DMin
7	D#Maj	D#Maj	*	D#Maj	D#Maj	D#Maj	*	D#Maj	D#Maj
8	D#Min	D#Min	*	D#Min	D#Min	D#Min	*	D#Min	D#Min
9	EMaj	EMaj	*	EMaj	EMaj	EMaj	*	EMaj	EMaj
10	EMin	EMin	*	EMin	EMin	GMaj	*	EMin	EMin
11	FMaj	FMaj	*	FMaj	FMaj	FMaj	*	FMaj	FMaj
12	FMin	FMin	*	FMin	FMin	FMin	*	FMin	FMin
13	F#Maj	C#Maj	*	F#Maj	F#Maj	F#Maj	*	F#Maj	F#Maj
14	F#Min	F#Min	C#Min	F#Min	F#Min	AMaj	AMaj AMin	F#Min	AMaj
15	GMaj	DMaj	EMin	GMaj	GMaj	DMaj	CMaj CMin	GMaj	GMaj
16	GMin	GMin	*	GMin	GMin	GMin	*	GMin	GMin
17	G#Maj	G#Maj	G#Maj	D#Maj	D#Maj	G#Maj	*	G#Maj	G#Maj
18	G#Min	G#Min	*	G#Min	G#Min	BMaj	C#Maj	G#Min	Bmaj
19	Amaj	Amaj	*	Amaj	Amaj	Amaj	*	Amaj	Amaj
20	Amin	Amin	*	Amin	Amin	Cmaj	*	Cmaj	Cmaj
21	A#Maj	Fmaj	*	Fmaj	A#Maj	A#Maj	*	A#Maj	A#Maj
22	A#Min	A#Min	*	A#Min	A#Min	A#Min	*	A#Min	A#Min
23	Bmaj	G#Min	F#Min	BMaj	BMaj	BMaj	DMaj DMin	BMaj	BMaj
24	BMin	BMin	*	BMin	BMin	DMaj	*	DMaj	DMaj
Score		20	1+20/24	22	23	18	20/24	22	20
Score out of 24 Shaded cells indicate a score of less than 100% * All keys selected									

Table J2 Key-finding results of the Krumhansl and Schmuckler algorithm for the first four notes of the 24 preludes of Bach's Well-tempered Clavier Book I.

Prelude	**Reference**	**KKI**	**KKII**	**KKIII**	**KKIV**	**TI**	**TII**	**TIII**	**TIV**
1	CMaj	CMaj	CMaj	CMaj	CMaj	CMaj	CMaj	CMaj	CMaj
2	CMin	CMin	CMin	CMin	CMin	CMin	CMin	CMin	CMin
3	C#Maj	C#Maj	C#Maj	C#Maj	C#Maj	C#Maj	C#Maj	C#Maj	C#Maj
4	C#Min	C#Min	C#Min	C#Min	C#Min	C#Min	C#Min	C#Min	C#Min
5	DMaj	DMaj	DMaj	DMaj	DMaj	DMaj	DMaj BMin	DMaj	DMaj
6	DMin	DMin	DMin	DMin	DMin	DMin	DMin	DMin	DMin
7	D#Maj	D#Maj	D#Maj	D#Maj	D#Maj	D#Maj	D#Maj	D#Maj	D#Maj
8	D#Min	D#Min	D#Min	D#Min	D#Min	D#Min	D#Min	D#Min	D#Min
9	EMaj	EMaj	EMaj	EMaj	EMaj	EMaj	EMaj	EMaj	EMaj
10	EMin	EMin	EMin	EMin	EMin	EMin	EMin	EMin	EMin
11	FMaj	FMaj	FMaj	FMaj	FMaj	FMaj	FMaj	FMaj	FMaj
12	FMin	FMin	FMin	FMin	FMin	FMin	FMin	FMin	FMin
13	F#Maj	A#Min	F#Maj	A#Min	A#Min	F#Maj	F#Maj	F#Maj	F#Maj
14	F#Min	F#Min	F#Min	F#Min	F#Min	DMaj	DMaj	F#Min	F#Min
15	GMaj	GMaj	GMaj	GMaj	GMaj	GMaj	GMaj	GMaj	GMaj
16	GMin	GMin	GMin	GMin	GMin	GMin	GMin	GMin	GMin
17	G#Maj	G#Maj	G#Maj	G#Maj	G#Maj	G#Maj	G#Maj	G#Maj	G#Maj
18	G#Min	G#Min	G#Min	G#Min	G#Min	G#Min	G#Min	G#Min	G#Min
19	AMaj	AMaj	AMaj	AMaj	AMaj	AMaj	AMaj F#Min	AMaj	AMaj
20	AMin	AMin	AMin	AMin	AMin	AMin	AMin	AMin	AMin
21	A#Maj	A#Maj	A#Maj	A#Maj	A#Maj	A#Maj	A#Maj	A#Maj	A#Maj
22	A#Min	A#Min	A#Min	A#Min	A#Min	A#Min	A#Min	A#Min	A#Min
23	BMaj	BMaj	D#Min	BMaj	BMaj	BMaj	BMaj	BMaj	BMaj
24	BMin	BMaj	F#Maj	BMaj	BMaj	BMaj BMin	F#Maj F#Min	BMaj BMin	BMaj BMin
Score		22	22	22	22	22.5	21	23.5	23.5
Score out of 24 Shaded cells indicate a score of less than 100%									

Table J3 Key-finding results for the Krumhansl and Schmuckler algorithm for the Courante from Bach's Cello Suite in C major.

Prelude	Reference	KKI	KKII	KKIII	KKIV	TI	TII	TIII	TIV
1	CMaj	CMaj	CMaj	CMaj	CMaj	CMaj	CMaj	CMaj	CMaj
2	GMaj CMaj CMin	CMaj	BMin	CMaj	CMaj	CMaj CMin	GMaj	CMaj CMin	CMaj CMin
3	GMaj	GMaj	GMaj	GMaj	GMaj	GMaj	GMaj	GMaj	GMaj
4	CMaj	GMaj	GMaj	GMaj	GMaj	CMaj	CMaj	CMaj	CMaj
5	CMaj	CMaj	CMaj	CMaj	CMaj	CMaj	CMaj	CMaj	CMaj
6	DMin	DMin	DMin	DMin	DMin	DMin	DMin	DMin	DMin
7	CMaj GMaj	CMaj	GMaj	CMaj	CMaj	CMaj	CMaj GMaj	CMaj	CMaj
8	GMaj	CMaj	CMaj	CMaj	CMaj	CMaj GMaj	GMaj	CMaj	CMaj
9	GMaj	GMaj	GMaj	GMaj	GMaj	GMaj	GMaj	GMaj	GMaj
10	AMin	BMin	EMaj	BMin	BMin	AMin	AMin	AMin	AMin
11	AMin	AMin	AMin	AMin	AMin	AMin	AMin	AMin	AMin
12	CMaj	CMaj	CMaj	CMaj	CMaj	CMaj	CMaj	CMaj	CMaj
13	DMaj	DMaj	DMaj	DMaj	DMaj	DMaj	DMaj	DMaj	DMaj
14	GMaj	DMaj	DMaj	DMaj	DMaj	GMaj AMin	GMaj	GMaj AMin	GMaj AMin
15	GMaj DMaj	GMaj	DMaj	GMaj	GMaj	GMaj	DMaj GMaj	GMaj	GMaj
16	CMaj CMin	GMaj	GMaj	GMaj	GMaj	CMaj CMin	CMaj CMin	CMaj CMin	CMaj CMin
17	EMin GMaj	GMaj	EMin	GMaj	GMaj	GMaj EMin	EMin	GMaj EMin	GMaj EMin
18	GMaj CMaj	CMaj	GMaj	CMaj	CMaj	CMaj GMaj	GMaj	CMaj GMaj	CMaj GMaj
19	GMaj DMaj	AMaj	BMin	AMaj	AMaj	DMaj	GMaj	DMaj	DMaj
20	AMin	AMin	AMin	AMin	AMin	AMin	AMin	AMin	AMin
21	CMaj EMin	EMin	EMin	EMin	EMin	EMin	CMaj	EMin	EMin
22	GMaj GMin	CMaj	DMaj	CMaj	CMaj	AMin	GMaj GMin	AMin	AMin
23	CMaj GMaj	BMin	EMin	BMin	BMin	GMaj	CMaj	GMaj	GMaj
24	DMin	DMin	DMin	DMin	DMin	DMin	DMin	DMin	DMin
25	FMaj CMaj	CMaj	CMaj	CMaj	CMaj	CMaj	CMaj FMaj	CMaj	CMaj
26	GMaj	GMaj	GMaj	GMaj	GMaj	GMaj	GMaj	GMaj	GMaj
27	DMaj	DMaj	DMaj	DMaj	DMaj	DMaj	DMaj	DMaj	DMaj
28	EMin	EMin	EMin	EMin	EMin	EMin	EMin	EMin	EMin
29	GMaj GMin	CMaj	DMaj	CMaj	CMaj	AMin	GMaj GMin	AMin	AMin
30	GMaj GMin	CMaj	DMaj	CMaj	CMaj	AMin	GMaj GMin	AMin	AMin
31	GMin	GMin	GMin	GMin	GMin	GMin	GMin	GMin	GMin
32	GMin	GMin	GMin	GMin	GMin	GMin	GMin	GMin	GMin
33	GMin EMin	AMaj	F#Min	AMaj	AMaj	DMaj EMin	EMin GMin	DMaj EMin	DMaj EMin
34	GMin EMin	AMaj	F#Min	AMaj	AMaj	DMaj EMin	EMin GMin	DMaj EMin	DMaj EMin
35	GMaj GMin	F#Min	DMaj	F#Min	F#Min	DMaj F#Min	GMaj GMin	DMaj F#Min	DMaj F#Min
36	DMaj	DMaj	DMaj	DMaj	DMaj	DMaj	DMaj	DMaj	DMaj
37	GMaj	GMaj	GMaj	GMaj	GMaj	GMaj	GMaj	GMaj	GMaj
38	GMaj AMin	AMin	GMaj	AMin	AMin	GMaj	GMaj	GMaj	GMaj

39	GMaj CMaj	DMaj	DMaj	DMaj	DMaj	GMaj	GMaj	GMaj	GMaj
40	GMaj GMin	GMaj	GMaj	GMaj	GMaj	GMaj GMin	GMaj GMin	GMaj GMin	GMaj GMin
Score		26	25	26	26	34	40	33.5	33.5
Score out of 40 Shaded cells indicate a score of less than 100%									

Table J4 Key-finding results of the Krumhansl and Schmuckler algorithm for the Gavotte from Bach's French Suite No. 5 in G major (BWV 816).

Prelude	**Reference**	**KKI**	**KKII**	**KKIII**	**KKIV**	**TI**	**TII**	**TIII**	**TIV**
1	GMaj	GMaj	GMaj	GMaj	GMaj	GMaj	GMaj	GMaj	GMaj
2	GMaj	EMin	DMaj	EMin	EMin	EMin	DMaj GMaj	EMin	EMin
3	GMaj	CMaj	GMaj	GMaj	GMaj	GMaj	GMaj	GMaj	GMaj
4	GMaj	DMaj	DMaj	DMaj	DMaj	GMaj	GMaj	DMaj	DMaj
5	GMaj	GMaj	GMaj	GMaj	GMaj	GMaj	GMaj	GMaj	GMaj
6	DMaj	EMin	DMaj	EMin	EMin	DMaj	DMaj	DMaj	DMaj
7	DMaj	DMaj	GMaj	AMaj	AMaj	DMaj	GMaj	DMaj	DMaj
8	DMaj	DMaj	DMaj	DMaj	DMaj	DMaj	DMaj	DMaj	DMaj
9	DMaj	DMaj	DMaj	DMaj	DMaj	DMaj	DMaj	DMaj	DMaj
10	GMaj	DMaj	DMaj	DMaj	DMaj	DMaj	GMaj	DMaj	DMaj
11	GMaj	GMaj	GMaj	GMaj	GMaj	GMaj	GMaj	GMaj	GMaj
12	GMaj	AMin	GMaj	AMin	AMin	GMaj	GMaj	GMaj	GMaj
13	EMin	BMaj	BMaj	BMaj	BMaj	BMaj	BMaj	BMaj	BMaj
14	EMin	D#Min	EMin	D#Min	D#Min	EMin	EMin	BMaj EMin	BMaj
15	EMin	BMin	F#Min	BMin	BMaj	EMin	DMaj EMin	EMin	EMin
16	EMin	EMin	EMin	EMin	EMin	EMin	EMin	EMin	EMin
17	EMin	EMin	EMin	EMin	EMin	EMin	EMin	EMin	EMin
18	GMaj	GMaj	CMaj	GMaj	GMaj	CMaj	CMaj	GMaj	GMaj
19	GMaj	GMaj	CMaj	EMin	EMin	GMaj	CMaj	GMaj	CMaj
20	GMaj	AMin	GMaj	AMin	AMin	CMaj	CMaj GMaj	CMaj	CMaj
21	GMaj	DMaj	DMaj	F#Min	F#Min	DMaj	GMaj	DMaj	DMaj
22	GMaj	F#Min	GMaj	AMin	AMin	GMaj	GMaj	GMaj	GMaj
23	GMaj	DMaj	GMaj	CMaj	CMaj	GMaj	GMaj	GMaj	GMaj
24	GMaj	GMaj	GMaj	AMin	AMin	GMaj	GMaj	GMaj	GMaj
25	GMaj	GMaj	GMaj	GMaj	GMaj	GMaj	GMaj	GMaj	GMaj
Score		12	16	10	10	19	19.5	18.5	
Score out of 25 Shaded cells indicate a score of less than 100%									

Appendix K

Pitch class coefficients for the Krumhansl and Kessler, Temperley and optimised key profiles

Table K1 summarises the pitch class coefficients for the key profiles proposed by Krumhansl and Kessler (Krumhansl, 1990) and Temperley (1999). Table K2 lists the pitch class coefficients for the optimised key profiles based on the *Gavotte* from Bach's *French Suite No. 5 in G major (BWV 816)* for a key resolution of 1.0% and the *Courante* from Bach's *Cello Suite in C major* for a key resolution of 10.0%. The coefficients are rounded to a resolution of 0.01 and are ordered from high to low.

Table K1 Pitch class coefficients for the Krumhansl and Kessler (Krumhansl, 1990) and the Temperley (1999) key profiles.

Krumhansl & Kessler				**Temperley**			
Scaled				**Scaled**			
Major profile		**Minor profile**		**Major profile**		**Minor profile**	
Coefficient	**Degree**	**Coefficient**	**Degree**	**Coefficient**	**Degree**	**Coefficient**	**Degree**
1.000	I	1.000	I	1.000	I	1.000	I
0.820	V	0.850	iii	0.900	III	0.900	iii
0.690	III	0.750	V	0.900	V	0.900	V
0.640	IV	0.630	vi	0.800	IV	0.800	IV
0.580	VI	0.560	II	0.800	VII	0.800	VII
0.550	II	0.560	IV	0.700	II	0.700	II
0.450	VII	0.530	vii	0.700	VI	0.700	vi
0.400	v	0.500	VII	0.400	ii	0.400	ii
0.380	vi	0.420	ii	0.400	iii	0.400	III
0.370	iii	0.420	VI	0.400	v	0.400	v
0.360	vii	0.410	III	0.400	vi	0.400	VI
0.350	ii	0.400	v	0.300	vii	0.300	vii

Table K2 Pitch class coefficients for the optimised key profiles based on the Gavotte from Bach's French Suite No. 5 in G major (BWV 816) and the Courante from Bach's Cello Suite in C major.

Gavotte_Unconstrained_1.0_B_2N				**Courante_Unconstrained_10.0_B_2N**			
Estimated				**Estimated**			
Major profile		**Minor profile**		**Major profile**		**Minor profile**	
Coefficient	**Degree**	**Coefficient**	**Degree**	**Coefficient**	**Degree**	**Coefficient**	**Degree**
1.000	I	1.000	I	1.000	I	1.000	I
0.950	II	0.990	V	0.880	II	1.000	iii
0.950	V	0.880	iii	0.880	IV	0.960	VII
0.860	IV	0.850	IV	0.870	V	0.910	IV
0.800	III	0.800	VII	0.860	III	0.900	V
0.800	VII	0.790	II	0.850	VII	0.870	II
0.620	VI	0.380	vi	0.590	VI	0.470	vi
0.490	vii	0.380	VI	0.500	ii	0.460	ii
0.450	ii	0.330	ii	0.500	iii	0.460	III
0.450	iii	0.330	III	0.500	v	0.460	v
0.450	v	0.330	v	0.500	vi	0.460	vii
0.450	vi	0.310	vii	0.500	vii	0.440	VI