Harmonic Visualizations of Tonal Music

Craig Stuart Sapp
Center for Computer Assisted Research in the Humanities
Center for Computer Research in Music and Acoustics
Stanford University
email: craig@ccrma.stanford.edu

Abstract

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

1 Motivation

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

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

One of the best algorithms for determining the key in a region of music is the Krumhansl-Schmuckler key-finding algorithm based on probe-tone ratings generated from experimental results (Krumhansl 1990). The KS algorithm is implemented in the `key` program contained in the Humdrum Toolkit for musical analysis (Huron). Applying the KS algorithm to an entire piece, table 1 lists the analyzed keys for the first half of the first book of J.S. Bach's Well-Tempered Clavier using the `key` program. The Well-Tempered Clavier is an excellent source of test material for testing key-finding algorithms, because each set of prelude and fugue in the collection are in a different key, starting in C major and then progressing chromatically through all 24 major and minor keys.

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

Table 1: KS algorithm results when applied to entire WTC book 1 compositions compared to actual tonic keys of the music. Identification errors are underlined.

Table 1 points out two common errors generated by key-finding algorithms in general. The first error is in fugue no. 3 where the KS algorithm identifies the dominant rather than the correct tonic key of C♯ major. The second error occurs in the eleventh prelude where the relative minor is identified rather than the key of F major.

These two errors are primarily due to more than one key-area being present in the analyzed music, causing slight offsets in the tonic key weightings such that a closely related key becomes more likely for the algorithm than the actual key.

A fifth-relation error can occur if the secondary key areas are predominantly all above or below the tonic key in the circle of fifths. The second error is a modality error. The correct key signature was identified, but the distribution of notes in the music was such that the tonic was incorrectly identified.

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

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

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

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

2 Diagram Types

2.1 Key-To-Color Mappings

To display data from key-finding algorithms in a compact visual manner, each key is mapped to a different color. The principle key-to-color mapping being used is shown in table 2. The colors of the rainbow are mapped onto the circle of fifths collapsed to the seven diatonic pitches. For example, the key C is assigned the color green. Ascending the circle of fifths yields G (blue), D (indigo blue), A(purple), E (red). Going the opposite direction in the circle of fifths takes you through the rainbow colors in the opposite direction: F (yellow), B♭ (orange), E♭ (red).

diatonic pitch class	*color*	*diatonic pitch class*	*color*
E	red	*G*	blue
B	orange	*D*	indigo
F	yellow	*A*	violet
C	green		

root	(*R*, *G*, *B*)	*root*	(*R*, *G*, *B*)
C♭	(36, 255, 0)	G♭	(54, 200, 218)
C	(0, 255, 0)	G	(63, 191, 255)
C♯	(9, 246, 36)	G♯	(63, 177, 255)
D♭	(63, 109, 255)	A♭	(118, 41, 255)
D	(63, 95, 255)	A	(127, 31, 255)
D♯	(73, 86, 255)	A♯	(145, 27, 219)
E♭	(237, 4, 36)	B♭	(255, 109, 0)
E	(255, 0, 0)	B	(255, 127, 0)
E♯	(255, 18, 0)	B♯	(255, 145, 0)
F♭	(255, 237, 0)	F♯	(218, 255, 0)
F	(255, 255, 0)		

Table 2: Sample RGB color mappings for key tonics.

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

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

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

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

2.2 Type 1: discrete time/discrete roots

The first type of diagram for key display divides a piece of music into successively smaller analysis window units as outlined in figure 1. The top level of the diagram analyzes all of the notes in the selection to generate a key identification. The second level of the diagram then splits the music into two equal parts and generates a key for the music in each half, and so on. Each successively lower level of the diagram divides the music into a greater number of equal-sized analysis windows which get continually smaller in duration.

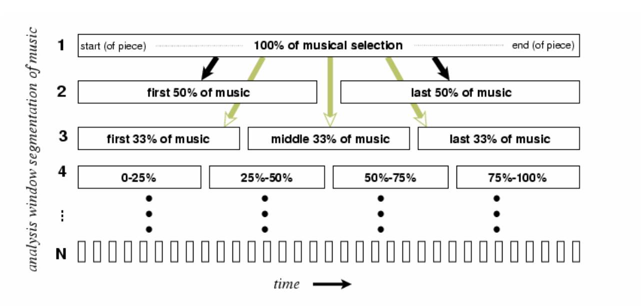

Figure 1: Type 1 analysis window configuration.

The lowest level will then divide the musical selection into N parts. The duration of the analysis windows in this level are *total-time*/N time units wide. Typically N is set to the number of beats in the music, but any smaller or larger value of N can be used as well. When using a value of N equal to the value of beats in the music, the bottom level of the diagram displays chord roots, and the top levels represent the strong key areas present in the piece.

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

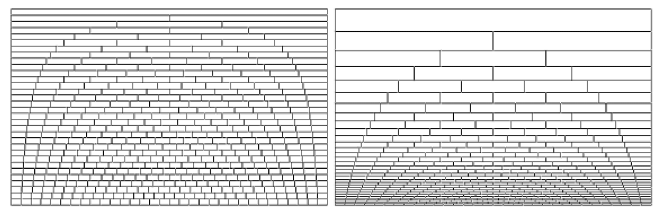

Figure 2: Type 1 plot analysis window layout. Left: linear vertical scale; Right: logarithmic vertical scale.

To correct for the emphasis on larger division levels, the vertical axis can be adjusted with a logarithmic scaling. A strong logarithmic scaling weights the upper levels of the diagrams too much, so good scalings which equally emphasize the top level and bottom levels of the diagrams is generated by this example scaling:

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

Where N is the total number of analysis window levels, n is the current level indexed from 0, M is the number of vertical pixels in the picture, and m is y-axis maximum pixel location of level n with respect to the bottom of the picture. The right side of figure 2 displays this type of mapping where the areas of the window blocks are closer in proportion to the change in window duration between levels.

Figure 3 displays a type 1 key diagram for the first movement of Mozart's Viennese Sonatina No. 1 in C major. The horizontal axis displays the music from the start (left) to the ending (right) of the piece. The vertical axis displays the analysis results for small window durations at the bottom of the picture and larger analysis window durations at the top of the picture. Pitch labels identify important key/chord regions in the piece. Notice that the key of C forms the dominant shaded area of the picture which coincides with the identification of this piece being in the *key* of C major.

The bottom portion of fig. 3 represents individual chords. As the analysis windows get larger towards the top of the picture, the chords merge with adjacent chords. Some chords get absorbed quickly into adjacent chords, particularly if they are remotely related to the surrounding chords. By the middle of the vertical axis, the strongest chords start to represent key

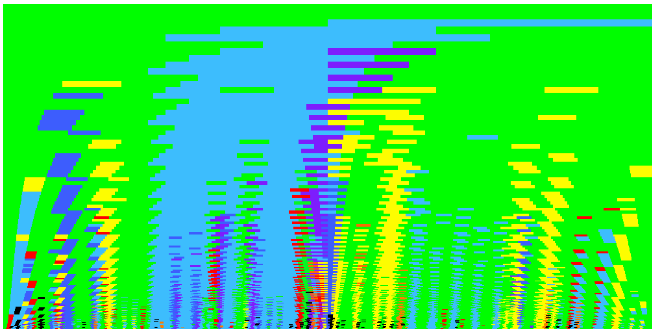

Figure 3: Mozart Viennese Sonatina No.1 in C Major, Mvmt.1: type 1 plot with logarithmic vertical scale.

areas. Some key areas have a short duration and are absorbed into stronger keys areas higher up in the diagram. The top portion of the figure shows the major key regions of the piece which are primarily the keys of C major and G major.

The sonatina movement is clearly in C major, since the upper left and right portions of the diagram are in C. The figure also shows a strong region of G major in the 2nd quarter of the piece. There is a distinction between the F modulation near the middle of the piece and the temporary modulation (tonicization) of F closer to the end of the piece. Tonicizations do not extend towards the top of the piece as high as true modulations into a key area.

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

2.3 Type 2: continuous time/discrete roots

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

Figure 4 gives an overview of the windowing method used to create type 2 key diagrams. The top level of the diagram contains an analysis window that can hold the entire piece of music. For that window, a single pixel is displayed at the top of the diagram. To generate the lower parts of the diagram, the analysis windows continually get smaller. For each window duration, the analysis window is slid continuously over the entire piece of music, placing the result of the key-finding algorithm at the center point of the analysis window. Typically, the analysis windows of the lowest level in the diagram contains one beat of the music. The placement of the analysis output at the center of the window generates a characteristic triangular shape which makes the diagram easy to distinguish from type 1 plots.

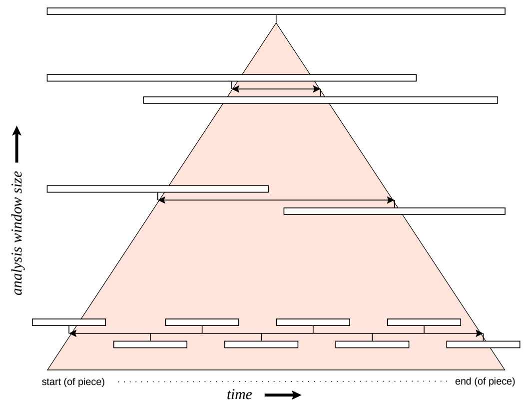

Figure 4: Type 2 analysis window arrangement.

Increased resolution is a big advantage of type 2 plots over type 1 plots. Figure 5 now shows the Mozart sonatina movement with a very nicely formed region in the key of G. Also, the development regions near the center of the movement are easier to distinguish since they are less fragmented.

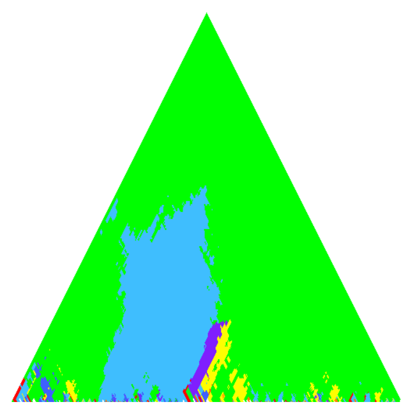

Figure 5: Mozart sonatina: type 2 plot with linear vertical scale (compare to fig. 3).

Just as there was a problem with the vertical scale in type 1 plots which show too much of the lower time-scales, figure 5 displays too much of the higher time-scales. Note that the second level of a type 1 plot is now found midway on the vertical axis in a type 2 plot. The unaltered form of the type 2 plots is equivalent to a strong logarithmic scaling of the type 1

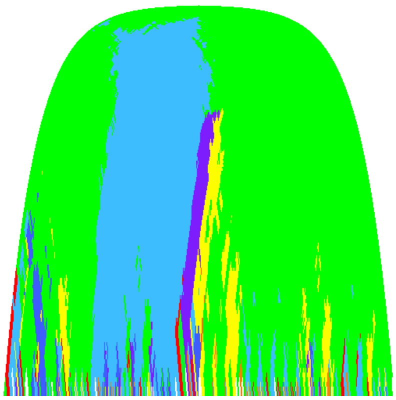

Figure 6: Mozart sonatina: type 2 plot with logarithmic vertical scale (compare with fig. 5.) The vertical scale is approximately the same as in fig. 3.

plots. Therefore, a more balanced view of the top and bottom levels can be achieved by scaling the vertical axes logarithmically, as shown in figure 6.

2.4 Type 3: continuous time and roots

Most key-finding algorithms calculate the likelihood of every possible key and then assign the key with the highest score as the best key to fit the music. For example, Table 3lists the KS algorithm scores of each possible key for the entire Mozart sonatina movement. The third category of key diagram described in this paper takes these secondary key probabilities into account to generate color-interpolated pictures of key based on the general form of type 2 plots.

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

Table 3: KS algorithm r-value scores of each possible key for the Mozart sonatina, sorted from most likely (highest r-value) to least likely key.

Figure 7 plots algorithmic score weightings of the four primary key centers taken from the midsection of figure 6. In this figure the relative strengths of each key score at any

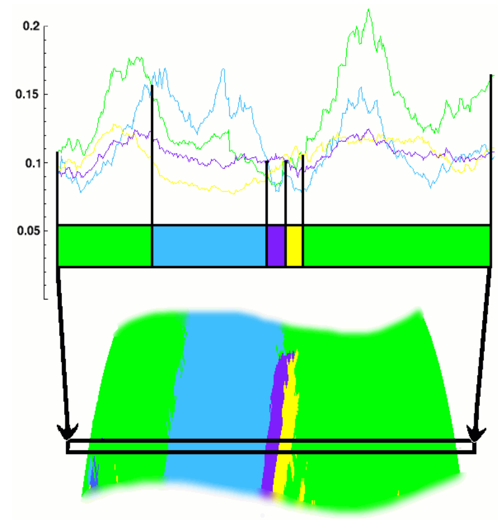

Figure 7: Continuous nature of some key-finding algorithms' key scores. Bottom part of figure is a midrange zoom-in on fig. 6. Top part of figure shows the continuous key-finding scores for important key regions highlighted in the lower figure.

given moment in the music is plotted. Type 1 and 2 key diagrams will only display the most likely key at any given moment, which in this case is the sequence: C, G, A, F, and then a return to C. What is lacking in these visualizations is an indication of how certain the algorithm is in its choice of key.

Two measurements in specific can be extracted from information shown in figure 7 to give a more continuous key diagram. (1) Clarity: The higher the score, the better the hypothesis key fits the music. Regions of musical stability are likely places to find a clear key center. For example, the recapitulation into C major in the first movement of the sonatina coincides with the highest score for C major. (2) Ambiguity: regions of development such as in the middle of the sonatina consist of closely scored keys. The algorithm has difficulty in choosing the best key in this case, because modulations happen so quickly compared to the more stable exposition and recapitulation. Key identification errors are more likely in regions of high ambiguity.

Experiments in displaying just the clarity or ambiguity yield promising diagrams which might prove useful in music analysis; however, key scores are difficult to interpret and relate to each other over different time scales. Figure 8 displays a plot of clarity between the best and second best keys. Notice that the short development region in the middle of the movement is clearly indicated by the dark band rising vertically in the center of the plot. Plots of ambiguity look similar to the regular best-key plots because they usually outline the borders between key areas.

Interpolating the colors of the best and second best keys

Figure 8: Plot of *clarity* – light regions indicate higher certainty of correctness by the key-finding algorithm and darker regions indicate little difference in the scores between best and second best keys.

gives a nice continuous diagram. Figure 9 shows a plot of the second best keys for the sonatina. Notice that in this piece, the second best key is usually tonic in a dominant section, and the dominant in a tonic section, although there is a strong tendency towards the subdominant during the beginning of the piece and close to the final cadence.

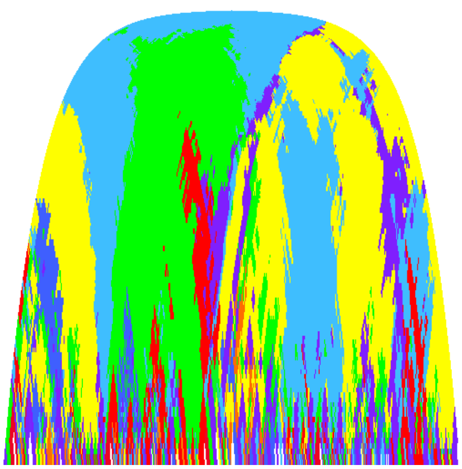

Figure 9: Mozart sonatina: second-best keys. Compare to fig. 6 which shows the best roots.

Figure 10 is an interpolation between the best and second best keys at each point in figures 6 and 9. For each analysis window size, the maximum clarity between the best and second best score is used to normalize the interpolation: at the point of maximum score separation, the best key is displayed fully in its own color. At points during the piece where the best and second best keys trade places (at 100% ambiguity), the color assigned to the diagram is halfway between the colors of the two key centers. Notice that the higher levels of secondary key regions (G and F) in figure 10 blend smoothly into the primary key of C major.

The interpolation for figure 10 was done linearly in the RGB color space. Interpolation in another color space, such as HSI (Hue-Saturation-Intensity) may be possible, but producing visually pleasant diagrams with this type of interpolation has been difficult to control. An HSI interpolation could be used to distinguish between modulations by fifths from other types of more distant key modulations. The more distant the modulation, the higher would be the saturation of the colors in the modulation region.

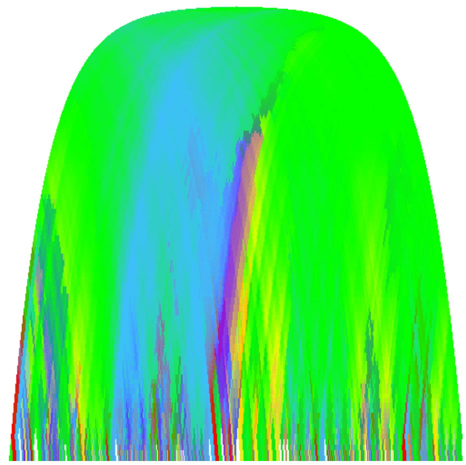

Figure 10: Mozart sonatina: type 3 plot interpolating between best root (fig. 6) and second best root (fig. 9).

3 Applications

3.1 Comparing key-finding algorithms

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

Figure 11 shows a type 2 plot of the Mozart sonatina movement which can be compared directly to figure 6. Good features: (1) definition of C major at start of piece is clearer than in figure 6, (2) dominant key areas in the piece are C and G, (3) incorrect A key region in middle of the piece is less pronounced, (4) more solid F key region in middle of the piece, (5) nicely behaved boundary between G and C key areas. Not-so-good features: (1) E minor key tendencies near end of the movement (modal error from G major), and (2) incorrect identification of key regions just before F major area.

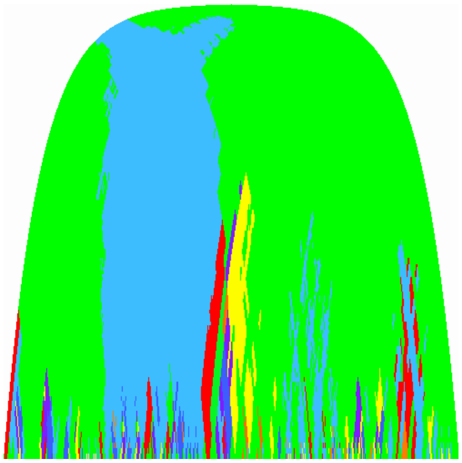

Figure 11: Mozart sonatina: using the KS algorithm. (compare to fig. 6).

A slightly stronger emphasis on the dominant key can also be seen in figure 11 as compared to figure 6. The large G major structure proceeds higher up in the diagram than it does in figure 6 and the fragmented G region around the 2/3 point in figure 11 is much less pronounced in figure 6. Thus, it can be inferred that the KS algorithm will be more prone to identifying the dominant key of the correct key than the algorithm used to generate figure 6.

Figure 11 can also be used to illustrate the difficulty of extracting a good sequence of modulations in a composition as mentioned in the Motivation section. The Mozart sonatina is a very simple and clear tonal piece. I say that the movement contains a sequence of 5 key regions: (C, G, D, F, C), with the following segmentation boundaries:

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

The D minor section is very brief and could be considered part of the F major region. D minor serves as a transition between the keys of G and F by switch roles from a dominant key to a relative minor key. Now notice that this sequence of keys is not present on any horizontal line in figure 11. Therefore, no fixed analysis window size in the KS algorithm will give the correct sequence according to my human-based harmonic analysis. Figure 6 fairs a little better with a wide region in the middle of the diagram containing the sequence (C, G, A, F, C), and the algorithm admits that it is having problems identifying a key in the D minor region (see figure 8).

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

3.2 Music Analysis

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

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

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

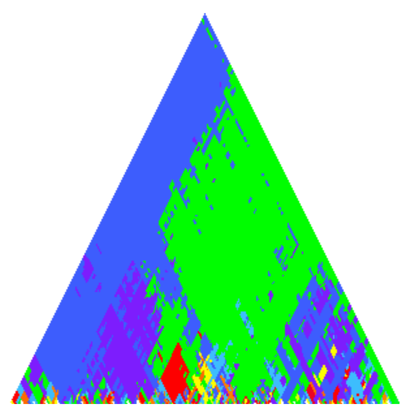

Figure 12: Anton Webern – Op. 27, No. 1. Twelve-tone music (compare to fig. 5 and fig. 13.).

As a contrast to tonal music, figures 12 and 13 display samples of non-tonal music. The Webern piano variation shown in figure 12 uses the modern technique found in twelve-tone music where all twelve pitch classes are used in sequence before being repeated. The twelve-tone technique is therefore used intentionally to destroy any sense of key throughout the composition. Notice that the key center boundaries in figure 12 are fragmented. The diagonal evolution of the key areas towards the top of the diagram is also unlike the vertical structures created in the Mozart sonatina. In particular, the C key region in the middle of the diagram has an inverted foundation. In other words, it is difficult to define the key of C major without playing C major chords. Figure 12 only shows the best keys, when plotting a type 3 diagram of the twelve-tone piece, the keys areas become even less pronounced than in this type 2 plot.

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

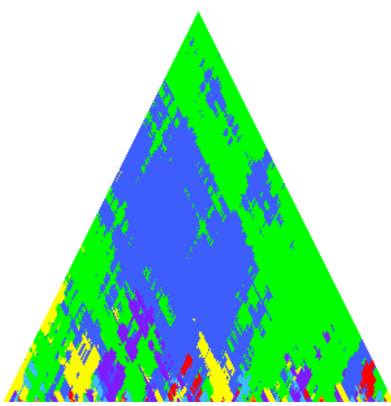

Figure 13: 13th century motet attr. to Petrus de Cruce (compare to fig. 5 and fig. 12.

As a final historical contrast, consider figure 14 which diagrams the key profiles in a prelude written by the late Romantic composer Alexander Scriabin. The key boundaries at the higher levels are fragmented, but the primary key is clearly E minor. The top level key centers in this diagram are closely related (tonic and dominant). The lower level key centers of A and D are arranged in a vertical manner similar to the key regions in the Mozart sonatina.

4 Summary

It is more interesting to know the interaction between key areas in a piece of music than just knowing the key-label for an entire composition or selection of music. With the

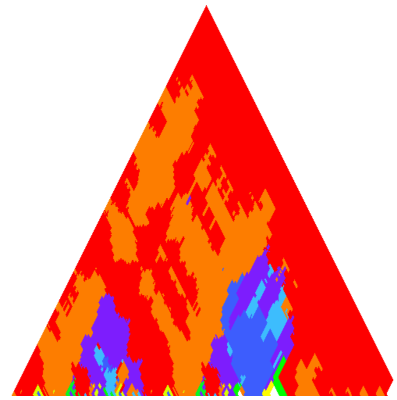

Figure 14: Alexander Scriabin – Prelude in E minor, Op.11, No.4.

harmonic maps generated from these visualizations of key-finding algorithms, the interactions and relations between keys become easier to examine.

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

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

References

Chew, Elaine (2000). *Towards a Mathematical Model of Tonality*. Ph. D. thesis, MIT, Cambridge, Mass.

Huron, David. The Humdrum Toolkit: Software for Music Research. `http://dactyl.som.ohio-state.edu/humdrum`.

Krumhansl, Carol. (1990). *Cognitive Foundations of Musical Pitch*. New York: Oxford University Press.

Sapp, Craig. Tonal Landscape Gallery. `http://www-ccrma.stanford.edu/~craig/keyscape`.

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

Sleator, Daniel and David Temperley. The Melisma Music Analyzer. `http://www.link.cs.cmu.edu/music-analysis`.

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