Music Perception
Fall 1999, Vol. 17, No. 1, 65–100

What's Key for Key? The Krumhansl-Schmuckler Key-Finding Algorithm Reconsidered

DAVID TEMPERLEY
Ohio State University

This study examines the Krumhansl-Schmuckler key-finding model, in which the distribution of pitch classes in a piece is compared with an ideal distribution or "key profile" for each key. Several changes are proposed. First, the formula used for the matching process is somewhat simplified. Second, alternative values are proposed for the key profiles themselves. Third, rather than summing the durations of all events of each pitch class, the revised model divides the piece into short segments and labels each pitch class as present or absent in each segment. Fourth, a mechanism for modulation is proposed; a penalty is imposed for changing key from one segment to the next. An implementation of this model was subjected to two tests. First, the model was tested on the fugue subjects from Bach's *Well-Tempered Clavier;* the model's performance on this corpus is compared with the performances of other models. Second, the model was tested on a corpus of excerpts from the Kostka and Payne harmony textbook (as analyzed by Kostka). Several problems with the modified algorithm are discussed, concerning the rate of modulation, the role of harmony in key finding, and the role of pitch "spellings." The model is also compared with Huron and Parncutt's exponential decay model. The tests presented here suggest that the key-profile model, with the modifications proposed, can provide a highly successful approach to key finding.

Received Nov. 23, 1998; accepted for publication Aug. 11, 1999.

KEY is an essential aspect of Western music. In an important sense, the key of a passage provides the framework whereby notes and harmonies are understood.[1] Knowing the status of a pitch or a chord relative to the current key—for example, knowing that a pitch is the tonic scale degree, or that a chord is a IV chord—is much more important than knowing the identities of pitches and chords in absolute terms. At a larger level, key structure contributes greatly to the expressive effects of tonal music (eigh-

1. Here, of course, I am referring to tonal music: music that conveys a sense of key or tonal center.

Address correspondence to David Temperley, Department of Music, Dodge Hall, Columbia University, New York, NY 10027. (e-mail: dt3@columbia.edu)

teenth- and nineteenth-century music in particular). As Charles Rosen (1971, p. 29) has shown, a modulation can act as a kind of "large-scale dissonance," a conflict demanding a resolution; it is largely this that allows tonal music to convey a sense of long-range motion and drama.

A good deal of work in music cognition has focused on key. Some of this work has explored the way in which the perception of other musical elements is affected by the key context—for example, studies in which the perceived stability or appropriateness of pitches and chords is measured in different tonal contexts (Butler, Brown, & Jones, 1994; Krumhansl, 1990). Other studies have shown that the perception of melody is affected by tonal factors as well. Melodic patterns that project a strong sense of key are more easily remembered than others (Cuddy, Cohen, & Mewhort, 1981); melodies are more easily recognized if they are presented in a suitable tonal context (Cuddy, Cohen, & Miller, 1979). The perception of modulation—change from one key to another—has also been studied. Thompson and Cuddy (1992) found that both trained and untrained listeners were sensitive to changes in key and that the perceived distances of modulations corresponded well to music-theoretical ideas about key distance. In another study, Cook (1987) explored listeners' ability to detect whether a piece began and ended in the same key; listeners were indeed sensitive to tonal closure for short pieces, although this sensitivity declined greatly for longer pieces. Another aspect of key is the distinction between major and minor keys, which appears to be perceptible and important to the emotional connotations of music even for very young children (Kastner & Crowder, 1990).

The studies just mentioned have strongly established the general psychological reality and importance of key. Another crucial question to ask about key structure is *how* it is perceived: by what method do people determine the key of a piece or changes in key within a piece? Considerable attention has been given to this issue. In this study, I focus on one approach to this problem, the key-profile algorithm of Carol Krumhansl and Mark Schmuckler, described most extensively in Krumhansl's book *The Cognitive Foundations of Musical Pitch* (1990). I point to some important problems with the Krumhansl-Schmuckler (hereafter K-S) algorithm and propose solutions. With these modifications, however, I suggest that the key-profile model can provide a highly effective approach to key finding. Along the way, I will examine some alternative solutions to the key-finding problem.

The approach of the current study is not experimental, but rather computational. My reasoning is that seeking a model that performs a task successfully may shed light on how humans perform the task. Of course, having a computer model that performs a task does not prove that humans do it the same way; this must be confirmed by other means. At the same time, if an algorithm does not achieve a reasonable level of success at a task performed by humans, we know that it is *not* the way humans do it, regardless of other psychological evidence. When we have several successful models of key finding, we will have the luxury of choosing between them on the grounds of psychological plausibility. Until then, it seems reasonable to pursue a purely computational approach.

Although I have said that my task is to model a task performed by humans, it is important to specify *which* task by *which* humans. My goal here is to model what might be called expert judgments: judgments of music theorists and other highly trained musicians, made reflectively and analytically (e.g., in the course of doing an analysis). One might also consider composers' scores for this purpose (key signatures and titles of pieces); these are sometimes useful, although also somewhat limited, in that they indicate only the main key of a piece, not secondary keys. Although my model was devised largely by relying on my own judgment as to the correct keys of sections of pieces, I also test the model against other sources.

Modeling expert judgments is not the only goal that might be pursued. One might also try to model the more ephemeral, unconscious judgments taking place during listening—either judgments of experts or of a broader population. In fact, the evidence so far (such as the studies mentioned earlier) suggests that there is considerable agreement between (trained and untrained) listeners' intuitions and expert judgments. In the original experiments on which the key-profile model was based (discussed further later), the subjects were musically trained but lacked extensive training in music theory (Krumhansl, 1990, p. 26). As Krumhansl's own tests have shown, the K-S model succeeds in predicting experts' judgments to a considerable degree, although it is somewhat flawed in this respect. Thus modeling expert analytical judgments and listeners' unconscious judgments may largely be convergent enterprises. But it is the former, rather than the latter, that primarily concerns us here. We should remember also that any failures by the K-S model to predict expert judgments should not necessarily be regarded as *flaws* in the model, because this is not what the model was primarily designed to achieve.

The Krumhansl-Schmuckler Key-Finding Algorithm

The Krumhansl-Schmuckler key-finding algorithm is based on "key profiles." A key profile is a vector of 12 values, representing the stability of the 12 pitch classes relative to a given key. The key profiles were based on data from experiments by Krumhansl and Kessler in which subjects were asked to rate how well each pitch class "fit with" a prior context establishing a key, such as a cadence or scale (Krumhansl & Kessler, 1982). A high value

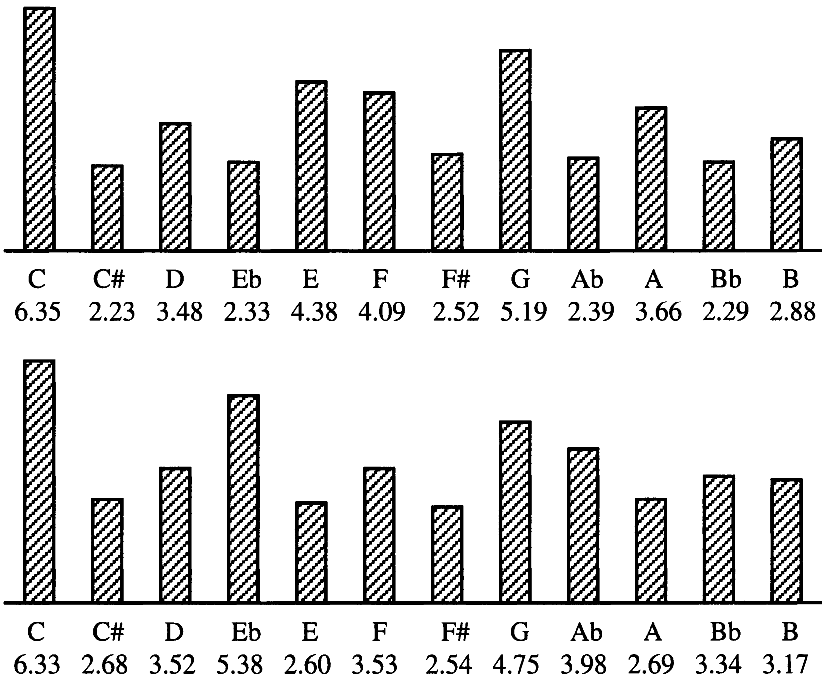

Fig. 1. The Krumhansl-Kessler key profiles, for C major (above) and C minor (below). (Data from Krumhansl, 1990, p. 80.)

in the key profile means that the corresponding pitch class was judged to fit well with that key. Each of the 24 major and minor keys has its own key profile. The key profiles for C major and C minor are shown in Figure 1; other profiles are generated simply by shifting the values around by the appropriate number of steps. For example, whereas the C-major vector has a value of 6.35 for C and a value of 2.23 for C♯, C♯ major would have a value of 6.35 for C♯ and a value of 2.23 for D.[2] As Krumhansl (1990, p. 29) notes, the key profiles reflect well-established musical principles. In both major and minor, the tonic position (C in the case of C major/minor) has the highest value, followed by the other two degrees of the tonic triad (E and G in C major, E♭ and G in C minor); the other four degrees of the diatonic scale are next (D, F, A, and B in C major; D, F, A♭, and B♭ in C minor—assuming the natural minor scale), followed by the five chromatic scale steps.

The algorithm judges the key of a piece by correlating each key profile with the "input vector" of the piece (Krumhansl, 1990, pp. 78–80). The

2. The original data were gathered for a variety of keys, but there was little variation between major keys (after adjusting for transposition), so the data were averaged over all major keys to produce a major-key profile that was then used for all major keys; the same was done for minor keys (Krumhansl, 1990, pp. 25, 27).

input vector is, again, a 12-valued vector, with each value representing the total duration of a pitch class in the piece. Consider Figure 2, the first measure of "Yankee Doodle"; assume a tempo of quarter note = 120. Pitch class G has a total duration of 0.75 (seconds); A has a duration of 0.5; B has a duration of 0.5; D has a duration of 0.25; the other eight pitch classes have durations of 0, because they do not occur at all in the excerpt. The input vector for this excerpt is shown in Figure 2. The correlation value, r, between the input vector and a given key-profile vector is then given by

$$r = \frac{\Sigma(x-\bar{x})(y-\bar{y})}{(\Sigma(x-\bar{x})^2\Sigma(y-\bar{y})^2)^{1/2}}$$

where x = input vector values, $\bar{x}$ = the average of the input vector values, y = the key-profile values for a given key, and $\bar{y}$ = the average key-profile value for that key. To find the key for a given piece, the correlations must be calculated between each key profile and the input vector; the key profile yielding the highest correlation gives the preferred key.

Table 1 shows the results of the algorithm for the first measure of "Yankee Doodle." G major is the preferred key, as it should be. All the pitches in the excerpt are in the G-major scale (as well as several other scales); moreover, the first and third degree of the G-major tonic triad are strongly represented, so it is not surprising that G major receives the highest score.

The first modification I wish to propose is a simplification in the way the key-profile scores are calculated. The formula used in the K-S algorithm is the standard one for finding the correlation between two vectors (Howell, 1997). In essence, this formula takes the product of the corresponding val-

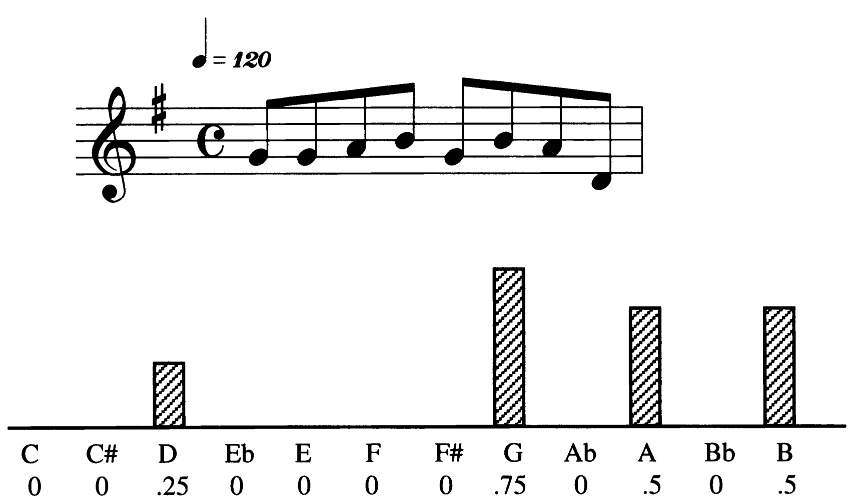

Fig. 2. Measure 1 of "Yankee Doodle," with input vector showing total duration of each pitch class.

TABLE 1
Key-Profile Scores for the First Measure of "Yankee Doodle" (Figure 2)

Key	Score
C major	.245
C♯ major	–.497
D major	.485
E♭ major	–.114
E major	.000
F major	.003
F♯ major	–.339
G major	**.693**
A♭ major	–.432
A major	.159
B♭ major	–.129
B major	–.061
C minor	–.012
C♯ minor	–.296
D minor	.133
E♭ minor	–.354
E minor	.398
F minor	–.384
F♯ minor	.010
G minor	.394
A♭ minor	–.094
A minor	.223
B♭ minor	–.457
B minor	–.436

ues in the two vectors (the input vector and the key-profile vector, in this case) and sums these products. To use Krumhansl's metaphor, this amounts to a kind of "template-matching": if the peaks in the key-profile vector for a given key coincide with the peaks in the input vector, this number will be large. The correlation formula also normalizes both vectors for their mean and variance. However, if our only goal is to find the algorithm's preferred key for a given passage of music, these normalizations are not really necessary. We can obtain essentially the same result by defining the key-profile score simply as the sum of the products of the key-profile vector value and the input-vector value, or Σxy—sometimes known as the "scalar product." The algorithm then becomes to calculate this score for all 24 keys and choose the key with the highest score.[3]

3. Normalizing the input vector values for their mean and variance has no effect on the algorithm's judgments because the input vector is the same for all 24 keys. Normalizing the key-profile vectors does have a slight effect, however. Because the mean value for the minor-key profiles is a bit higher than for major-key profiles, removing this normalization tends to bias the algorithm toward minor keys, relative to the original K-S algorithm. However, this effect could be counteracted simply by adjusting the key-profile values. As we will see, the key-profile values seem to require significant adjustment in any case.

Improving the Algorithm's Performance

Krumhansl (1990, pp. 81–106) reports several tests that were done of the algorithm. First, the algorithm was tested on the first four notes of each of the 48 preludes of Bach's *Well-Tempered Clavier*. (In cases in which the fourth note was simultaneous with one or more other notes, all the notes of the chord were included.) The algorithm chose the correct key on 44 of the 48 preludes, a success rate of 91.7%. Similar tests were done on the first four notes of Shostakovich's and Chopin's preludes, yielding somewhat lower success rates: 70.8% and 45.8%, respectively. In another test, the algorithm was tested on the fugue subjects of the 48 fugues of the *Well-Tempered Clavier* and on the subjects of Shostakovich's 24 fugues. For each fugue, the algorithm was given a series of note sequences starting from the beginning of the piece: first the first note, then the first two notes, then the first three notes, and so on. At the point where the algorithm first chose the correct key, the test for that piece was terminated. On 44 of the Bach fugue subjects and 22 of the Shostakovich fugue subjects, the algorithm eventually found the correct key. As Krumhansl (1990, p. 93) acknowledges, this test is somewhat problematic, because it is unclear how stable the algorithm's choice was; it might choose the correct key after four notes, but it might have shifted to a different key if it was given another note. Thus it is difficult to draw any conclusions from this test as to the algorithm's success rate in finding the correct key for fugue subjects. Finally, the algorithm was tested on each measure of Bach's Prelude no. 2 in C minor, and its judgments were compared with the judgments of two experts as to the key of each measure. In this case, however, the algorithm's results for all keys were combined and represented as a single point on a four-dimensional spatial representation of keys. Moreover, the algorithm's judgment for each measure was based on a weighted sum of the pitch durations in the current measure and also previous and subsequent measures, in order to reflect the effect of context on key judgments.

Although all of these tests are of interest, only the first group of tests provides clear data about the judgments of the basic algorithm for isolated segments of music. The algorithm's performance here was mixed: it performed much better on the Bach preludes than on the Shostakovich and Chopin preludes. However, one could argue that judging the key after four notes is unrealistic in the latter two cases, given their more complex tonal language. In any case, further tests seem warranted.

An easy and informal way of testing the algorithm is by giving it a piece, having it judge the key for many small segments of the piece in isolation—measures, say—and comparing the results with our own judgments. This process was done by using a computer implementation of the algorithm, exactly as it is specified in Krumhansl's book. (For this test, then, the original formula was used, rather than the modified formula proposed here.) In deciding what we think is the correct key for each measure, it is important to stress that each measure is to be regarded *in isolation*, without considering its context, because this is what the algorithm is doing. This is not, of course, how we naturally listen to music, but considering the tonal implications of a small segment of music taken out of context is, I think, not difficult to do. (Note that the current test differs from Krumhansl's test of the Bach Prelude no. 2, where both the experts and the algorithm were taking the context of each measure into account in judging its key.)

Figure 3 shows the first half of the Courante of Bach's Cello Suite in C major (BWV 1009). The algorithm's preferred key is shown above each measure (the top row of symbols, labeled "Krumhansl-Schmuckler"). In a number of cases, the algorithm's choice is clearly correct; measure 1, for example. In some cases, the key is somewhat unclear, and the algorithm chooses one of several plausible choices (in measure 17, it chooses G major, although E minor would certainly be possible). In a number of cases, however, the algorithm's choice seems doubtful; these cases are indicated with an exclamation mark. In measure 4, the algorithm chooses G major, although the measure contains an F—this pitch is not present in a G-major scale and, as a $\flat\hat{7}$ scale degree, is indeed highly destabilizing to the key; however, all the notes of the measure are present in the C-major scale. Similar errors occurs in measures 14 and 16. In a number of other cases (measures 8, 22, 29, 30, 33, 34, and 35), the algorithm chooses a key despite the presence of a pitch outside the scale of that key and despite the existence of another key that contains all the pitches of the segment. This result suggests that the algorithm does not distinguish strongly enough between diatonic and chromatic scale degrees. The algorithm also sometimes chooses minor keys when the lowered seventh or raised sixth degree of the key is present (e.g., measures 10 and 19), although these are much less common than the raised seventh and lowered sixth degree, that is to say, the "harmonic minor" scale. Altogether, the algorithm makes incorrect judgments on 13 of the 40 measures, a correct rate of 67.5%. The discrepancy between this result and the algorithm's performance on the opening four-note segments of the Bach preludes is worth noting. Inspection of the preludes shows that a great number of them begin by outlining or elaborating a tonic triad. The Bach Courante would seem to provide a wider variety of melodic and harmonic situations, although of course it, too, is a highly limited sample.

Inspecting the key-profile values themselves (Figure 1), it becomes clear why some of these errors occur. Although diatonic degrees have higher values than chromatic degrees, the difference is slight; in C major, compare the values for B (2.88) and F♯ (2.52). In minor, we find that the flattened seventh degree (B♭ in the case of C minor) has a higher value than the leading tone (B). This finding seems counterintuitive; as mentioned earlier,

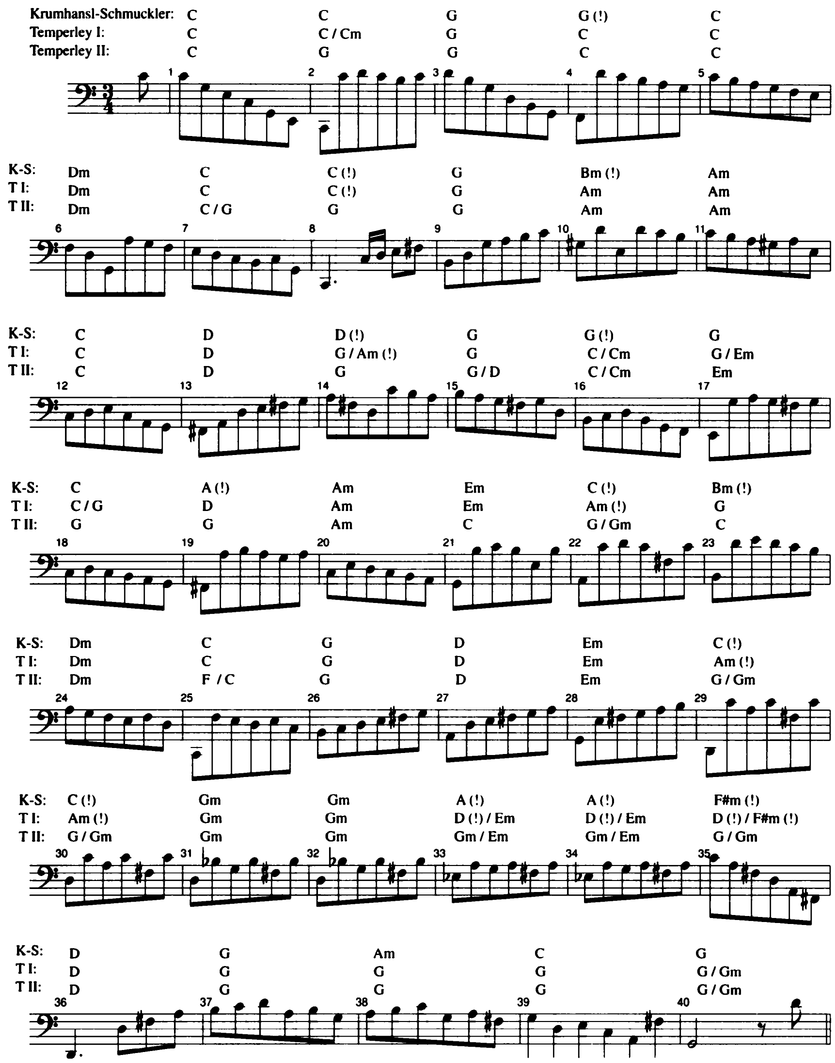

Fig. 3. Bach, Suite for Violoncello No. 3 in C major, Courante, mm. 1–40, showing key judgments for each measure from three different key-finding algorithms. Minor keys are marked with "m"; all other keys are major.

the flat seventh is quite destabilizing to the tonic, whereas the leading tone is often a strong indicator of a new tonic (consider the way G♯ in measure 10 of the Bach Courante points toward A minor). In major, too, it seems odd that the leading tone has the lowest value of the seven diatonic degrees. A related problem should be mentioned: the dominant seventh, which is usually taken as strongly implicative of the corresponding tonic key, is not so judged by the K-S algorithm. Rather, the G dominant seventh (for example) most strongly favors G major, B minor, D minor, D major, G minor, and F major, in descending order of preference; C major is seventh on the list, with C minor even farther down. Finally, it seems likely that some of the minor values are too high; in general, there is a slight tendency for the model to choose minor keys more often than it should. (The total score for the minor triad, 16.46, is slightly higher than that for the major triad, 15.92, which seems odd.)

A revised version of the profile is shown in Figure 4, which attempts to solve these problems. These values were arrived at by a mixture of theoretical reasoning and trial and error, using a variety of different pieces for testing. (An attempt was made to keep all the values in the same range as those in the K-S algorithm, to permit easy comparison.) The basic primacy of the diatonic scale is still reflected; all diatonic steps have higher values than chromatic ones. In the case of minor, I assume the harmonic minor scale, so that b$\hat{6}$ and $\hat{7}$ are within the scale and $\hat{6}$ and b$\hat{7}$ are chromatic. All the chromatic degrees have a value of 2.0, with the exception of b$\hat{7}$, which has a value of 1.5. The unusually low value for b$\hat{7}$ proved necessary, in part, to achieve the right judgment for the dominant seventh, but it appears to lead to good results in general. All the diatonic degrees have a value of at least 3.5; $\hat{2}$ and $\hat{6}$ in major, and $\hat{2}$ and b$\hat{6}$ in minor, have exactly this value.

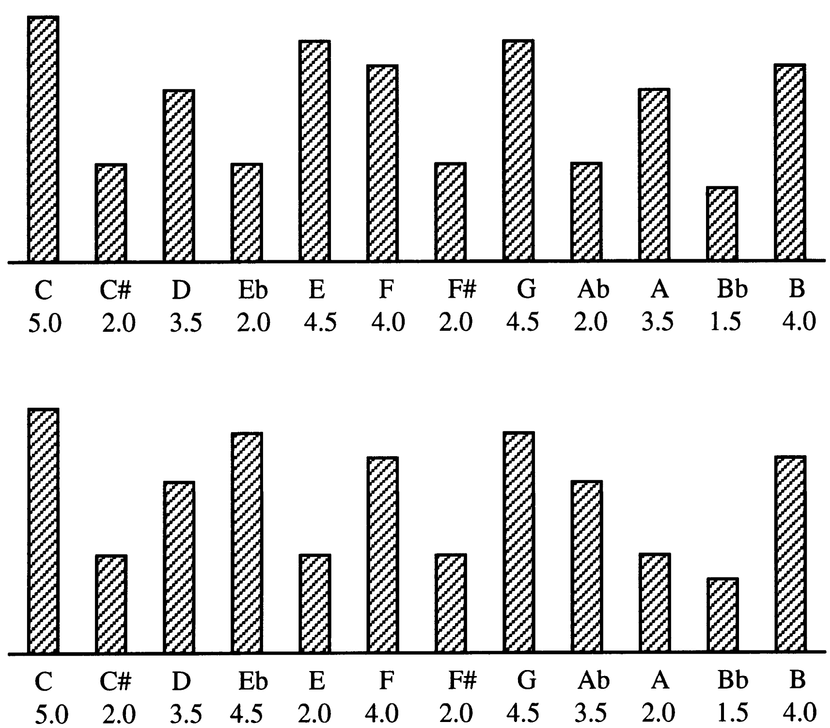

Fig. 4. A revised version of the key profiles, shown for C major (above) and C minor (below).

Degrees $\hat{4}$ and $\hat{7}$ are given slightly higher values (4.0), reflecting their importance (more on this later). The triadic degrees—$\hat{1}$, $\hat{3}$, and $\hat{5}$ in major and $\hat{1}$, $\flat\hat{3}$, and $\hat{5}$ in minor—receive the highest values; the value for $\hat{1}$ is highest of all. The same values are used for both the major and minor tonic triads.

These revised key profiles improve performance considerably on the Bach Courante. (I now use my modified version of the key-profile formula, rather than the original K-S version.) The results of the algorithm are shown as "Temperley I" in Figure 3; questionable choices are again marked with exclamation marks. In a few cases, two keys receive exactly equal scores; in such cases both keys are shown, for example "C/G." The algorithm now makes only 6.5 errors instead of 13. (If the algorithm chooses two keys and one of them seems incorrect, this is counted as half an error). The key-profile approach has another problem, however, that cannot be solved merely by tweaking the key-profile values. Consider measure 29; the pitches here, D-C-A-C-F♯-C, outline a D dominant seventh. Given a simple D dominant seventh, with four notes of equal length played once, my algorithm (unlike the K-S algorithm) chooses G major and G minor equally as the preferred key. In measure 29, however, my algorithm chooses A minor. The reason is clear; there are three C's and one A, all members of the A-minor triad, giving a large score to this key that swamps the effects of the other two pitches. Yet perceptually, the repetitions of the C do not appear to strongly tilt the key implications of the measure toward A minor, or indeed to affect them very much at all.

This finding raises a fundamental question about the key-profile approach. Even if we accept the basic premise of "template matching," it could be done in several ways. A different approach to template matching is found in Longuet-Higgins and Steedman's (1971) earlier key-finding algorithm. This algorithm processes a piece left to right (only monophonic pieces were considered). At each pitch that it encounters, it eliminates all keys whose scales do not contain that pitch. When it is left with only one key, this is the preferred key. (This is a somewhat simplified version of the model, which will be discussed further later.) We could think of the Longuet-Higgins/Steedman model as implying a very simple key-profile model. In this model, each key has a "flat" key profile, where all the pitch classes in the corresponding diatonic scale have a value of one, and all chromatic pitch classes have a value of zero. The input vector is also flat: a pitch class has a value of one if it is present anywhere in the passage, zero if it is not. Choosing the correct key is then a matter of correlating the input vector with the key-profile vectors—which in this case simply amounts to counting the number of pitch classes scoring one in the input vector that also score one in each key-profile vector. We might call this a "flat-input /flat-key" profile model, as opposed to the "weighted-input/weighted-key" profile model proposed by Krumhansl and Schmuckler. Note that the Longuet-Higgins/Steedman model (or rather the simplified version of it that I have presented here) handles a case such as measure 29 of the Bach better than the K-S model. All four pitch classes—D-F♯-A-C—are present in G major (and G minor) and no other keys; thus these keys (and these keys alone) receive a winning score of 4. However, the Longuet-Higgins/Steedman model also encounters problems. In particular, the algorithm has no way of judging passages in which all the pitches present are in more than one scale. Consider measure 1 of the Bach; this C-major arpeggio clearly implies C major, yet all of these pitches are also present in G major and F major (and several minor scales as well); thus the Longuet-Higgins/Steedman algorithm would have no basis for choosing between them. In this case, the K-S model is clearly superior, because it makes important distinctions between diatonic scale degrees.

This result suggests that the best approach to key finding may be a combination of the Krumhansl-Schmuckler and Longuet-Higgins/Steedman approaches: a "flat-input/weighted-key" approach. That is, the input vector simply consists of "1" values for present pitch classes and "0" values for absent ones; the key-profile values, on the other hand, are individually weighted, in the manner of Krumhansl's key profiles (and my revised version). (Judging pitch classes simply as "present" or "not present," without considering their frequency of occurrence at all, might not work so well for longer passages; but my model does not do this for longer passages, as we will see later.) The output of this version for the Bach Courante is shown in Figure 3 as "Temperley II." The algorithm's choice is at least reasonable on all 40 measures. In its judgments for small pitch sets, then, the current algorithm seems to represent an improvement over the original K-S model.

Some connections should be noted between the values I propose and other theoretical work. Fred Lerdahl's (1988, p. 321) theory of tonal pitch space is based on a "basic space" consisting of several levels, corresponding to the chromatic scale, diatonic scale, tonic triad, tonic and fifth, and tonic, as shown in Figure 5. Lerdahl (1988, p. 338) notes the similarity between his space and Krumhansl's key profiles; both reflect peaks for diatonic pitch classes and higher peaks for triadic ones. My own key-profile

C												C
C							G					C
C				E			G					C
C		D		E	F		G		A		B	C
C	C#	D	Eb	E	F	F#	G	Ab	A	Bb	B	C

Fig. 5. Lerdahl's (1988, p. 321) "basic space" (configured for C major).

values correspond more closely to Lerdahl's space than Krumhansl's do, in that pitch classes at each level generally have the same value—the exceptions being the higher values for $\hat{4}$ and $\hat{7}$ and the lower value for $\flat\hat{7}$. (My profiles also assign equal values to the third and fifth diatonic degrees, thus omitting the "fifths" level).

The higher values for $\hat{4}$ and $\hat{7}$ bring to mind another theoretical proposal, the "rare-interval" approach to key finding. David Butler (1989) has argued that certain small pitch class sets have particular relevance for key finding, because of their "rarity": the fact that they are only present in a small number of scales. A major second such as C-D is present in five different major scales; a tritone such as F-B is present in only two. Similarly, F-G-C is present in three major and two harmonic minor scales; F-G-B is present only in C major and minor. Although this is clearly true, it is not obvious that such considerations should be explicitly reflected in key profiles. This point requires some discussion. Let us assume a "flat-key/flat-input" profile model, in which the key of a piece is simply given by the scale that contains the largest number of the pitch classes present. In such a case, both F-G-B and F-G-C receive a score of 3 for C major (let us consider only major keys for the moment). In the case of F-G-B, however, C major is the only key receiving this score and will thus be the clear favorite, whereas F-G-C will also receive scores of 3 from G major and F major and therefore should be ambiguous. In this sense, the importance of certain "rare" pitch sets would be an emergent feature of the system, despite the fact that $\hat{4}$ and $\hat{7}$ are treated no differently from other scale degrees in the key profile itself. Clearly, such emergent effects could arise from weighted key profiles as well, without necessarily giving special weight to $\hat{4}$ and $\hat{7}$. Nonetheless, my trial-and-error tests have shown that it *is* necessary to give special weight to $\hat{4}$ and $\hat{7}$ in the key profile. In particular, it is very difficult to achieve correct results on the dominant seventh (and related pitch sets) unless this is done.

It is beyond the scope of this article to speculate on the reasons (historical, cognitive, or psychoacoustical) for the apparent importance of the $\hat{4}$ and $\hat{7}$ degrees, but I will offer one observation. Imagine a similarity space generated from the diatonic flat-key profile (again considering only the major keys), so that keys with highly correlated profiles were close together. Each key would be closest to the two other keys with which it shared six pitch classes: for example, C major would be closest to F major and G major. It can be seen that $\hat{4}$ and $\hat{7}$ are the scale degrees that maximally distinguish each scale from its nearest neighbors. For example, C major shares all its pitches with G major except F ($\hat{4}$ of C); it shares all its pitches with G except B ($\hat{7}$ of C). Boosting these values in the key profile in effect lowers the similarity between each key's profile and the key profiles of its closest neighbors; the effect of this may be to "sharpen" the model, allowing it to produce clearer and less ambiguous judgments. (The unusually low value for $b\hat{7}$ has a similar effect.) But whether this observation is of any significance requires further study.

Modulation

In one important sense, the K-S algorithm is not so much flawed as incomplete. The algorithm produces a single key judgment for a passage of music it is given. However, a vital part of tonal music is the shifts in key from one section to another. This is a complex matter, because there are different levels of key. Each piece generally has one main key, which begins and ends the piece, and in relation to which (in music theory anyway) intermediate keys are understood. An extended piece will generally contain modulations to several secondary keys—for example, the Bach Courante moves to G major around measure 9, then (in the second half, not shown in Figure 3) to A minor, and then back to C major; there may be even briefer tonal motions as well, so-called tonicizations (e.g., the momentary move to A minor in measures 10–11 of the Courante). One might propose the key-profile system as a way of determining the global level of key. I believe, however, that this is not the most sensible use of the key-profile model. What the key-profile system does well is determine the keys of sections of pieces. It is probably true that most pieces spend more time in their main tonic keys than in other keys, in which case a key-profile model might often work. It seems to me, however, that the global key of a piece really depends on other factors; in particular, the key of the beginning and ending sections. (One global key algorithm that would succeed in the vast majority of cases would simply be to choose the key of the first—or last—section of the piece.)[4]

The key-profile algorithm could easily be run on individual sections of a piece, once these sections were determined. Ideally, however, the division of the piece into key sections would be determined by the algorithm also; presumably, the same information that allows us to infer the correct key also allows us to infer when the key is changing. How might this work? One possibility is that the algorithm could simply examine many small

4. Actual statistical analyses of entire pieces show mixed results. Krumhansl (1990, pp. 66–71) reports that note counts in some pieces show a high key-profile correlation with the appropriate keys, although she does not show that these note counts correlate more highly with the "correct" key than with any other. In other studies, the results have been unpromising. Krumhansl (1990, pp. 71–73) found that the key-profile model predicted G major as the key of Schubert's *Moment Musical* no. 1, whereas the correct key is C major; Butler (1989, p. 225) found that the predicted key of the *Moment Musical* no. 2 was Eb major, rather than the correct Ab major. Of course, one might attribute these failures to the details of the algorithm, such as the fact that the key-profile values are not ideal. As I have said, however, I believe that using the key-profile model to predict global key is misguided.

segments of a piece in isolation; changes of key would then emerge at places where one segment's key differed from that of the previous one. However, this is not very satisfactory. Consider the Bach Courante. The preferred key of measure 3, considered in isolation, is probably G major; heard in context, however, it is clearly outlining a V chord, part of a C-major section. Once we begin to get a series of segments that clearly imply G major, though, we sense a definite shift in key. Intuitively, key has inertia: we prefer to remain in the key we are in, unless there is strong and persistent evidence to the contrary. A simple way of modeling this suggests itself: we apply the key-profile algorithm to each segment in isolation but also impose a penalty for choosing a key for one segment that differs from the key of the previous one. Generating a key analysis for a piece thus involves optimizing the key-profile match for each segment while minimizing the number of key changes. In some cases, this might lead the model to choose a key for a segment that is not the best choice for that segment in isolation (e.g., in measure 3 of the Bach). However, if the scores for a segment or a series of segments favor another key strongly enough, then it will be worth switching.

The algorithm just described was computationally implemented. The implementation of the algorithm searches for the highest-scoring key analysis of the piece it is given. A key analysis is simply a labeling of each segment of the piece with a major or minor key. (The input to the program must contain a segmentation of the piece into low-level segments. The program chooses a key for each segment; it considers changes of key only between segments, not within them. The division of the piece into segments in this way may seem somewhat arbitrary; I will return to this later.) Each segment in an analysis yields a numerical score based on (1) how well its pitches fit the key chosen for that segment (according to the modified key-profile formula) and (2) whether the current segment has the same key as the previous one (if not, a penalty is applied). The score for an analysis is simply the sum of the key-profile scores and change penalties for each segment. It may be noted that the number of possible analyses increases exponentially with the number of segments; however, the program uses dynamic programming to find the highest-scoring possible analysis without actually having to generate them all.[5]

The program was tested on several pieces, including the Bach Courante. A change penalty of 6.0 was used; as before, measures were used as segments. In the case of the Courante, the program's preferred analysis begins with a 12-measure section in C major; it then modulates to G major for the remainder of the first half. The second half moves to A minor at measure

5. For discussion of this computational approach, which has been used elsewhere in musical preference rule systems, see Temperley and Sleator (1999). The computer implementation described here, written in C, is publicly available at the website www.link.cs.cmu.edu/music-analysis.

44 and then to F major at measure 57, returning to C major at measure 65 and remaining there until the end. This analysis seems largely correct. The move to G major should perhaps occur a little earlier (measure 9?); one could also argue that there is a brief move to D minor in measure 61 (alternatively, one could argue that both the moves to F major and D minor are only tonicizations, not real modulations). One might also criticize the fact that the key sections are not overlapping—there are no "pivot chords"; I will return to this point.

Aside from the key profiles themselves, there are two main numerical parameters in the program. One is the change penalty, the penalty for choosing a key for one segment different from that of the previous segment. By choosing a higher value, one can push the program toward less frequent changes and longer key sections; choosing a lower value has the opposite effect. Another, related, parameter is the length of segments. This is a complex matter, requiring some discussion. Consider an extended passage of music and assume for the moment that there is no modulation within the passage. Under the current proposal, the key-profile scores for the passage simply sum together the scores for the individual segments. Thus we can think of a passage as having an aggregate input vector that sums the values of the input vectors in each segment. Note that this aggregate input vector will not be "flat" but will to some extent reflect the frequency of occurrence of pitch classes in the passage. Because a "flat" input profile is being used for the individual segments, however, the combined input vectors for two segments are not the same as the input vector for a single larger segment containing both. If the two smaller segments contain the same pitch classes, the combined input vector value will be two for each pitch class present, whereas the value for the larger segment will be one. For a given passage, then, using shorter segments will produce larger key-profile scores; differences in scores between alternative analyses will be greater, and will be more likely to exceed the change penalty, so that changes in key will be more rapid. Unless this was adjusted, there would be a tendency for shorter segments to produce more rapid key changes. The solution is to adjust the key-profile scores by the length of segments. Then, roughly speaking, two smaller segments combined together will yield the same input vector as a larger segment containing both. Even with this adjustment, however, the length of segments still makes a difference. Consider, say, the first 8 measures of the Bach Courante. If segments of only one note were used, then the overall input vector for the passage would count each note; in effect, the input profile would perfectly reflect the frequency of occurrence of pitch classes just as in the original K-S model. As the segments increase in length, the effect of frequency of occurrence diminishes and the input profile becomes flatter (if the entire passage were considered as a segment, the input profile would be completely flat). Choosing a segment length allows one to strike a balance between a completely flat profile and a completely weighted one. Segments of about 1 measure—typically between 1 and 2 s long—proved to be about optimal in this regard. (This also means that the tempi chosen for pieces may affect the program's analysis.)

The Key-Profile Model as a Preference Rule System

The key-finding algorithm I have proposed could be viewed as a simple preference rule system. A preference rule system is one that considers many possible analyses of a piece or passage, evaluates them by certain criteria, and chooses the highest-scoring one (Lerdahl & Jackendoff, 1983; Temperley, 1997; Temperley & Sleator, 1999). (The original K-S algorithm could be viewed as an even simpler preference rule system, with only one rule.) It is important to note that preference rule systems are not the only approach to modeling musical structure; indeed, some approaches that have been applied to key finding clearly are not preference rule systems. Rather, they are what might be called "procedural" systems: systems that are more easily described in terms of the procedure they follow rather than the output they produce. Longuet-Higgins and Steedman's model, mentioned earlier, provides a simple example. This algorithm begins by proceeding through a (monophonic) passage note by note, eliminating all keys whose scales do not contain the current note. If it reaches the end of the passage with more than one eligible key still remaining, it goes back to the beginning and looks at the first note. If this note is the tonic of one of the remaining eligible keys, that key is chosen; if not, and it is the dominant of one of the remaining keys, that key is chosen. If it ever takes a step that eliminates all remaining scales, it undoes that step, and then performs the tonic-dominant test just described. Clearly, a model such as this is quite different from a preference rule system, in which the preferred analysis is simply the one that best satisfies certain global criteria.

Several other procedural models for key finding deserve mention. Holtzmann's (1977) model searches a melody for certain features, such as the tonic triad and the tonic fifth; it searches for these features first at the beginning and end of the melody and then at points in between. Also worth mentioning are Winograd's (1968) and Maxwell's (1992) systems for harmonic analysis. The intent of these systems is to produce a Roman numeral analysis, showing chord symbols, but this requires inferring key information, because the appropriate Roman numeral symbol for a chord depends on the current key. In both systems, chords are first labeled in a "key-neutral" fashion; key sections are then identified, largely by searching for possible cadences and other conventional harmonic progressions (for a review of these systems, see Temperley, 1997). Yet another procedural model is the parallel processing model of Vos and Van Geenen (1996). This model gathers evidence or "weight" for all keys, considering both pitch evidence (scale degrees of each key) and harmonic evidence (members of the tonic, dominant, and subdominant triads); in this respect, it somewhat resembles a key-profile model. However, the model also involves complex decision procedures—for example, adjusting the weights for keys under certain circumstances—which give it a more procedural flavor (Vos & Van Geenen, 1996, pp. 191–193.)[6]

Although there is no a priori reason to prefer a preference rule system over a procedural one, preference rule systems have several advantages that should be mentioned. The first is their handling of real-time processing. Although my main aim here is to model analytical judgments, having a system that can model real-time listening as well is surely desirable, and it is interesting to consider how a preference rule system could accomplish this. Preference rule systems involve evaluating many analyses of an entire piece or passage and choosing the one that is preferred overall. Because the rules typically involve constraints between one part of the piece and another, segments of the piece cannot be analyzed in isolation; rather, the system must find the analysis of each segment that leads to the most preferred analysis overall. In the case of the key-finding algorithm described earlier, the preferred analysis of one segment may depend on what happened in the previous segment (or the following segment) due to the change penalty. This might at first appear to be a problem in terms of modeling real-time processing, because the best analysis for the piece overall is not known until the piece is over. However, preference rule systems present an elegant solution to this problem. Imagine that the key-finding algorithm proposed here processes a piece from left to right, segment by segment; at each segment S_n, it chooses the best possible analysis for the entire piece up to and including S_n. However, it may be that when it reaches S_{n+1}, the overall best analysis for everything so far may entail a different analysis for segment S_n than the one initially chosen. In this way the algorithm naturally handles the "garden-path" effect: the phenomenon of revising one's hearing of a piece based on what happens afterwards.

An example is shown in Figure 6, the Gavotte from Bach's French Suite No. 5 in G major (BWV 816). The algorithm's key analysis is shown above the staff; each key name indicates a new key section beginning at that segment. (Again, a change penalty of 6.0 was used; measures were used as

6. In discussing alternative models of key finding, we should also mention Bharucha's (1987) connectionist model. This model features three levels of nodes, corresponding to pitches, chords, and keys; sounding pitches activate pitch nodes, which activate chord nodes of the chords containing them, which in turn activate nodes of corresponding keys. Although this model is certainly of interest, it does not appear to have been tested in any significant way, so we will not consider it further here.

Fig. 6. Bach, French Suite No. 5, Gavotte, showing program's key analysis.

segments.) Figure 7 shows a "running" analysis of the same piece. The segments of the piece are listed vertically; for each segment, the program's provisional analysis of the piece up to and including that segment is shown horizontally. The diagonal edge of the chart indicates the program's initial analysis for each segment at the moment it is heard. When the choice of key for a segment in a particular provisional analysis is identical to the choice in the previous analysis, only a hyphen ("-") is shown. In cases where keys of earlier segments are shown, this means that the program revised its initial key choice for a segment and chose something else instead—the "gar-

```
M.no.
     G
 1   - G
 2   - - G
 3   - - - G
 4   - - - - G
 5   - - - - - G
 6   - - - - - D D
 7   - - - - - - - D
 8   - - - - - - - - D
 9   - - - - - - - - - D
10   - - - - - - - - - G G
11   - - - - - - - - - - - G
12   - - - - - - - - - - - - B
13   - - - - - - - - - - - - - B
14   - - - - - - - - - - - - Em Em Em
15   - - - - - - - - - - - - - - - Em
16   - - - - - - - - - - - - - - - - Em
17   - - - - - - - - - - - - - - - - - C
18   - - - - - - - - - - - - - - - - - - C
19   - - - - - - - - - - - - - - - - - - - C
20   - - - - - - - - - - - - - - - - - - - - C
21   - - - - - - - - - - - - - - - - - G G G G G
22   - - - - - - - - - - - - - - - - - - - - - - G
23   - - - - - - - - - - - - - - - - - - - - - - - G
24   - - - - - - - - - - - - - - - - - - - - - - - - G
```

Fig. 7. A "running" analysis of the Bach Gavotte in Figure 6, showing the program's complete analysis at each segment. (The first segment contains only the first half-measure of the piece.) In cases where the program's choice for a segment is the same as its choice for that segment in the previous analysis, only a hyphen is shown.

den-path" effect. For example, consider measures 5 and 6. At measure 5, the program was still considering the first five measures to be in G major. Given measure 6, however, the program decided that measure 5 would be better interpreted as being in D. Other garden-path effects are found at measure 9, measures 12–13, and measures 17–20. Notice that two of the provisional key sections in the running analysis—the move to B major in measures 12–13 and the move to C in measure 17–20—are completely obliterated in the final analysis.

Whether or not the running analysis in Figure 7 exactly captures our moment-to-moment hearing of the Bach Gavotte, it suggests that the current model might provide interesting insight into the kind of tonal analysis that goes on during listening. The program also sheds light on another phenomenon. It is widely agreed that modulations typically involve "pivot chords," chords that are compatible with both the previous key and the following one. In the Bach Gavotte, for example, the D-major chord in the second half of measure 9 could either be considered as I of D or V of G. This would imply that key sections generally overlap by at least one chord. The current approach holds out another possibility, however, which is that pivot regions are essentially a diachronic phenomenon. It is not that a chord is understood as being in two keys at one time, but rather it is first interpreted in one way, based on the previous context, and then in another way, under the influence of the following context. There is nothing in the program that actively searches for, or prefers, such diachronic pivot effects, but they often do seem to emerge at points of modulation.

As well as their handling of real-time listening, preference rule systems have another emergent feature which deserves mention. Preference rule systems operate by considering many possible analyses of a piece, scoring them, and choosing the highest-scoring one. In so doing, they produce not only a preferred analysis of a given piece, but also a numerical score for that analysis (or a series of scores for the segments of the analysis). This can be regarded as a measure of how much the piece itself "satisfies" (or permits a satisfactory analysis from) the preference rules. Daniel Sleator and I have argued elsewhere that this is a musically interesting aspect of preference rule systems with regard to meter and harmony (Temperley & Sleator, 1999). With regard to key, the scores tell us how well the passage in question could be accommodated within a key or series of keys; we might call these "key-fit" scores. When a passage scores low, that means that no suitable key could be found for it (or that many modulations were required to do so). One problem here is that, under the current algorithm, a segment containing more pitch classes will generally have a higher score; this makes it difficult to compare scores across segments. We can cancel this by defining the "key-fit" score as the key-profile scores divided by the number of pitch classes in the segment.

One straightforward use of these measures is as a measure of the tonalness (or the "being-in-a-key") of pitch class sets. Consider the three-note pitch class sets found in Figure 8; if these are run individually through the algorithm, the key-fit scores shown above the staff are produced. The major and minor triads score equally high; both can be accommodated within a diatonic scale (several scales, in fact), and each one matches the peaks of one particular key for the tonic triad ($\hat{1}$, $\hat{3}$ or b$\hat{3}$, and $\hat{5}$). The diminished triad and augmented triad score lower; they can be accommodated within

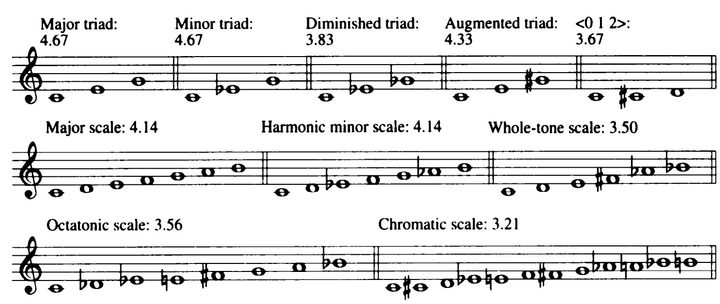

Fig. 8. Some pitch class sets and their "key-fit" scores. The key-fit score is the highest key-profile score obtained by the algorithm, divided by the number of pitch classes in the set.

diatonic scales (the diminished triad fits one major and four minor scales; an augmented triad fits three harmonic minor scales) but are not tonic triads. The pitch class set <012> scores lowest of all, reflecting the fact that it cannot fit into any scale. Turning to larger pitch class sets, the major and harmonic minor scales (which score equally) score higher than the whole-tone and octatonic scales; these in turn score higher than the complete chromatic scale. These ratings seem to correspond well with intuition as to the "tonalness" of these various pitch class sets. (Note that these are the scores yielded for the program's preferred key. In the case of the augmented triad and the whole-tone, octatonic, and chromatic scales, several keys are equally preferred, because these sets are symmetrical. Remember also that the key-profile score for a segment is divided by the number of pitch classes to produce the key-fit score.) The close fit between the key profiles and the major and minor scales and tonic triads has, of course, been observed before (earlier in this paper, and by Krumhansl with respect to her original key profiles). Still, it is an attractive feature of the key-profile model that it yields a measure of the tonalness of any arbitrary pitch set.

These key-fit scores may also capture an important aspect of listening. Figure 9 shows an excerpt from Chopin's Mazurka op. 17, no. 4, a piece famous for its tonally unstable nature. The program analyzes this passage as being in A minor throughout.[7] The scores for each segment reflect how well the pitch classes of the segment fit with A minor. A U-shaped trajectory emerges. Beginning fairly firmly in A minor, the piece flirts with C major in measures 7–8 (introducing G, foreign to the A harmonic minor scale); in measure 9, the piece moves further afield (three of the four pitches in the measure are outside A minor), and then back toward A minor. This reflects one aspect of the rising and falling musical tension of the phrase—only one aspect, to be sure, because there are many other aspects to musical tension (voice leading, harmony, psychoacoustic dissonance, and so on) that the model does not capture. The most interesting aspect of these scores

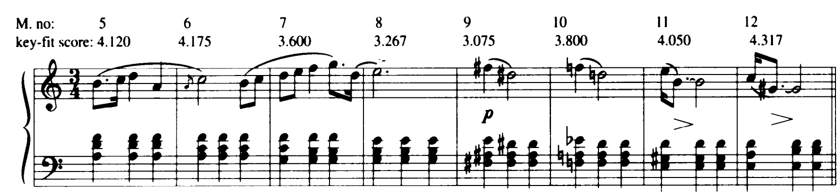

Fig. 9. Chopin, Mazurka op. 17, no. 4, mm. 5–12. The numbers above the staff show the key-fit scores for each measure, relative to A minor.

7. This A-minor analysis is the one that the program produces having analyzed the entire excerpt. Its moment-to-moment analysis might of course be different, as was the case with the Bach Gavotte discussed earlier.

is that nothing special has to be done to produce them; they are a natural by-product of the program's search for the preferred analysis.

Testing the Model

The model described in the preceding section was subjected to two formal tests. First, it was tested on the 48 fugue subjects from Bach's *Well-Tempered Clavier;* then, on a series of excerpts from the Kostka-Payne theory textbook. Some general comments are needed on how the tests were done.

The input required for the program is a list of notes, with a pitch, on-time, and off-time for each note; the program can process this information in the form of a MIDI file. Because the program requires absolute time information, decisions had to be made about the tempi of pieces, as I explain later. In all cases, trills, ornaments, and other notes shown in small noteheads were excluded, owing to the difficulty of deciding objectively on the timing for these. The program also requires a list of segments, with each segment having a start time and end time. (The piece must be exhaustively partitioned into nonoverlapping segments.) It seemed logical to have segments correspond to measures, or some other level of metrical unit (such as half-measure or 2-measure units). It also seemed important to have strict criteria for the length of segments, because this may influence the analysis. The following rule was used: given the tempo chosen, the segments used corresponded to the fastest level of metrical unit above 1 s. For example, if a piece (at the tempo I chose) had 4/4 measures that were 1.6 s long, the measure was used as the unit (because the faster unit, half-measures, would be 0.8 s); however, if the measures were 2.2 s long, then half-measures would be 1.1 s long and could be used. In 45 of the 48 *Well-Tempered-Clavier* cases, and 43 of the 46 Kostka-Payne cases, this resulted in a segment level between 1.0 and 2.0 s; in the remaining cases, the segment length was slightly more than 2.0 s (these were slow triple meter pieces, where there was no level between 1.0 and 2.0 s). In cases where the excerpt began with an incomplete segment, this portion was treated as a separate segment if it was at least half the length of a regular segment; otherwise it was absorbed into the following segment. Partial segments at the end of the excerpt were always treated as complete segments.

The key profiles themselves were not in any way modified to improve the program's score on these tests. (As noted earlier, the key-profile values were set on the basis of theoretical considerations and tests on other pieces.) However, the change penalty value *was* modified; on each test, different values were tried, and the value was used that seemed to yield the best performance. (Both the Bach and Kostka-Payne corpora involve modulations, as I will explain.) For the Bach fugues, a penalty of 6.0 was used; for the Kostka-Payne corpus, the penalty was 12.0. As discussed earlier, key structure is generally thought to be hierarchical; a piece may have one level of large-scale key changes and another level of tonicizations. It seemed fair to adjust the program's change penalty to allow it to maximally match the level of key change in each test corpus.

For the purposes of the current tests, all enharmonically equivalent keys were regarded as the same. For example, if a Bach fugue was notated as being in A♭ major and the program labeled it as G♯ major, this was still considered correct. (The same applies to the Kostka-Payne test.) The program does not distinguish between different "spellings"; I will discuss later how this problem might be addressed.

There were two reasons for choosing the fugue subjects from the *Well-Tempered Clavier* as a test corpus. First, the correct keys of the pieces is obvious, from the key signatures and also from the well-known ordering of pieces within the collection (C major, C minor, C♯ major, C♯ minor, and so on). Another reason for using this corpus was that it has also been used in several other key-finding studies and therefore provides a basis for comparison. Longuet-Higgins and Steedman (1971), Holtzmann (1977), and Vos and Van Geenen (1996) all tested their systems on the corpus (Holtzmann uses only Book I, the first 24 fugues). Krumhansl also tested the K-S model on the fugue subjects. However, as noted earlier, her test was problematic, in that she stopped the algorithm when it had reached the correct key, without giving any measure of how stable that decision was. The other algorithms either self-terminated at some point or ran to the end of the subject and then made a decision.

Because it is not always clear where the fugue subjects end, decisions have to be made about this. Vos and Van Geenen rely on the analyses of Hermann Keller (1976) to determine where the subjects end; I used this source as well. (It is not clear how Longuet-Higgins & Steedman and Holtzmann made these decisions.) Another problem concerns modulation; many of the fugue subjects do not modulate, but several clearly do. Here again, Vos and Van Geenen rely on Keller, who identifies modulations in six of the fugue subjects.[8] Longuet-Higgins and Steedman's and Holtzmann's systems are not capable of modulation; they simply sought to identify the correct main key.

Longuet-Higgins and Steedman's algorithm found the correct main key in all 48 cases. Holtzmann's found the correct key in 23 of the 24 cases in Book I. Vos and Van Geenen's program detected the correct key as one of

8. Vos and Van Geenen's assumption that only six of the fugue subjects modulate is not necessarily correct. Keller mentions modulations in only six cases, but it is not clear that he would have mentioned all cases. It appears that several others may modulate: Book II, number 10, for example. However, we will disregard this for now.

its chosen keys in 47 of 48 cases; it also detected modulations in 2 of the 6 cases noted by Keller. However, it also found modulations (and hence multiple keys) in 10 other cases in which there was no modulation.

The current program was run on all 48 fugue subjects. For the tempi, I used the suggested tempi in Keller (1976). Segments were determined in the manner described earlier; the program chose one key for each segment. The current system, of course, has the option of modulating and was allowed to modulate wherever it chose. Its performance on the corpus is shown in Table 2. On 42 of the 48 fugues, the system chose a single key that was the correct opening key. In two cases, there were ties; in both cases, the correct key was among the two chosen. If we award the program half a point for the two ties, this yields a score of 43 out of 48. Of the six modulating fugues, it modulated on two of them (moving to the dominant in both cases, as is correct). On three of the nonmodulating subjects, the program incorrectly modulated (in all three of these cases it chose the correct opening key). In one modulating theme (Book I, No. 10), the program chose the second key as the main key. There were thus four nonmodulating themes where the program chose a single, incorrect, key.

It is rather difficult to compare the performance of the various programs on this test. Longuet-Higgins and Steedman's system, and then Holtzmann's, perform best in terms of finding the main key of subjects (although their inability to handle modulation is of course a limitation). (Compared with Vos and Van Geenen's system, the current program seems slightly better. The current program produced a perfectly correct analysis in 36 cases, Vos and Van Geenen's in 34 cases.) Although the success of the Longuet-Higgins/Steedman and Holtzmann programs is impressive, we should note that the programs are rather limited, in that they can handle only monophonic passages at the beginning of pieces. When the Longuet-Higgins/Steedman algorithm is unable to choose a key by eliminating those whose scales do not contain all pitches present, it chooses the key whose tonic or dominant pitch is the first note of the theme (this rule is needed on 22 of the 48 Bach fugue subjects). Holtzmann's approach also relies heavily on the first and last notes of the theme. Clearly, this approach is useful only at the beginnings of pieces; it is of no help in determining the keys of internal sections, because there is no obvious "first note." (Relying on the last note of the theme, as Holtzmann does, is even more problematic, because it requires knowing where the theme ends.) Still, it is of course possible that special factors operate in key finding at the beginning of pieces; and it does seem plausible, in some of these cases, that some kind of "primacy factor" is involved. (A "first-note" rule of this kind could possibly be incorporated into the current algorithm as a preference rule, but I will not address this here.) Another way of addressing this would be by considering harmonic information, as I discuss later.

Table 2
The Program's Performance on the 48 Fugue Subjects of Bach's *Well-Tempered Klavier*

Number	Opening Key	Modulating?*	Program's Opening Key†	Modulation Found?*
Book 1				
1	Cma			
2	Cmi			
3	C♯ ma			
4	C♯ mi			
5	Dma		Gma	
6	Dmi			
7	E♭ma	Yes		
8	D♯mi			
9	Ema			
10	Emi	Yes	Bmi	
11	Fma			
12	Fmi			
13	F♯ ma			
14	F♯ mi			Yes (incorrect)
15	Gma			
16	Gmi			
17	A♭ma			
18	G♯mi	Yes		Yes (correct)
19	Ama	Yes		
20	Ami			
21	B♭ma			
22	B♭mi			
23	Bma			
24	Bmi	Yes		Yes (correct)
Book 2				
1	Cma			
2	Cmi			
3	C♯ma			
4	C♯mi			
5	Dma		Gma	
6	Dmi			
7	E♭ma		E♭ma/A♭ma	
8	D♯mi			
9	Ema			
10	Emi			Yes (incorrect)
11	Ema			
12	Fmi			
13	F♯ ma		Bma	
14	F♯ mi			
15	Gma			
16	Gmi		Gmi/B♭ma	
17	A♭ma			
18	G♯mi			
19	Ama			
20	Ami	Yes		
21	B♭ma			
22	B♭mi			Yes (incorrect)
23	Bma			
24	Bmi			

*No unless marked yes.
†If incorrect or tie.

Next, an attempt was made to give the program a more general test. Because part of the purpose was to test the algorithm's success in judging modulations, it was necessary to have pieces where such key changes were explicitly marked. (Again, musical scores usually indicate the main key of the piece in the key signature, or in other ways, but they do not generally indicate changes of key.) A suitable corpus of data was found, namely, the workbook and instructor's manual accompanying Stefan Kostka and Dorothy Payne's textbook *Tonal Harmony* (1995a). This workbook (Kostka & Payne, 1995b) contains a number of excerpts from tonal pieces (almost all of them in the standard tonal repertory) requiring the student to add harmonic and key symbols. The instructor's manual contains answers to the exercises (done by Stefan Kostka, 1995); it also contains a few analyzed excerpts not included in the workbook. This source was chosen because it was the only such source that was (a) well-known and well-respected, and (b) contained a large number of real (as opposed to artificial or especially composed) excerpts with analysis provided. An example of one of Kostka's analyses is shown in Figure 10. All that concerns us here is the key symbols, the letters followed by colons beneath the score (C major and G major, in this case). (Note that the second measure contains a pivot chord, a segment that is in two keys at once.) In some cases the analysis is presented separately from the score; in such cases, the marking of measures in the analyses makes it clear how the analysis corresponds to the score (which is always included either in the workbook or the instructor's manual).

The sample chosen consisted of all the musical examples in the Kostka-Payne workbook and instructor's manual that fit certain criteria. First, they had to have explicit key information provided (in the form of key symbols). Second, they had to be at least 8 measures long. With very short excerpts, it seemed possible that context would be required to determine the key of the excerpt. Third, the excerpts in the last two chapters of the book—entitled "Tonal Harmony in the Late Nineteenth Century" and "An Introduction to Twentieth-Century Practices"—were excluded; the authors themselves note that many of the excerpts included in these chapters "defy tonal analysis" (Kostka & Payne, 1995a, p. 451). This left a corpus of 46

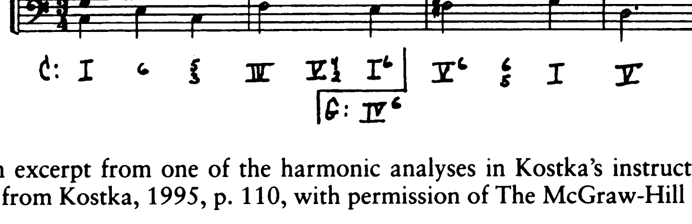

Fig. 10. An excerpt from one of the harmonic analyses in Kostka's instructor's manual. (Reprinted from Kostka, 1995, p. 110, with permission of The McGraw-Hill Companies).

excerpts. Again, decisions had to be made as to the tempo of each excerpt; I simply used my judgment to choose what seemed like reasonable tempi. I did not include any tempo fluctuations or changes in any of the excerpts. (As it happened, there were no excerpts in which internal tempo changes seemed terribly important.)

The program's analyses of the 46 excerpts were then compared with the analyses of the excerpts in Kostka's instructor's manual. For each segment in which the program's key choice was the same as Kostka's, one point was given. One problem was what to do if a segment was notated (by Kostka) as being partly in one key and partly in another. A related problem concerned pivot chords. The Kostka analyses contained many such chords, which were notated as being in two keys simultaneously. (As noted earlier, the program itself does not allow changes of key within a segment; nor does it allow multiple keys for a segment, except in rare cases of exact ties.) The solution adopted was this. When a segment contained either a pivot chord or a change of key—we could call such segments "bitonal"—then 1/2 point was given if the key chosen by the program was one of the keys given by Kostka. Otherwise, zero points were given. In a way, this rule is hard on the algorithm, because it means that in analyses with pivot chords, it is impossible for the program to receive a perfect score. (Because 52 out of a total of 896 segments in the Kostka-Payne corpus are bitonal, and the program cannot score more than 1/2 point on each of these, its total score cannot exceed 870/896 = 97.1%.) On the other hand, one might regard the program's inability to handle pivot chords as an inherent flaw, for which it should be penalized. In cases where the program produced an exact tie, in which one of the keys chosen was the correct one, half a point was given for the segment.

Out of 896 segments, the program attained a score of 751: a rate of 83.8%. The program found 40 modulations, exactly the same number as occurred in Kostka's analyses. It is useful to divide the sample according to where the excerpts occur in the workbook. Like most theory texts, Kostka and Payne's begins with basic chords such as major and minor triads and dominant sevenths and then moves on to chromatic chords—augmented sixths, Neapolitans, and the like. Thus we can divide the examples into two groups: those that occur in the chapters relating to diatonic chords (chapters 1–21), and those occurring in the later chromatic sections (chapters 21–26). Viewed in this way, the program scored a rate of 91.4% on the earlier chapters, 75.6% on the later ones, demonstrating a better ability for more diatonic passages. (A possible reason for this will be discussed later.)

To my knowledge, no earlier key-finding system has been subjected to a general test of this kind, so it is difficult to draw comparisons between systems.[9] Although the program's performance on the *Well-Tempered-Clavier* and Kostka-Payne tests certainly seems promising, it was not perfect, and it is instructive to consider the errors it made. In a number of cases in the Kostka-Payne corpus, the error was simply that the program's rate of modulation was wrong: it either modulated too rarely, missing a move to a secondary key, or it modulated too often, changing key where Kostka does not. An obvious ad hoc solution is to modify the change penalty. In almost all such cases, a virtually perfect analysis was obtained simply by making the change penalty higher or lower. Still, it is clearly a flaw in the program that this parameter has to be adjusted for different pieces. Possibly the program could be made to adjust the change penalty on its own, but it is not clear how this might be done.

In a few other cases, the problem is clearly not with the rate of modulation. In one case, Chopin's Mazurka op. 67 no. 2 (measures 1–16), the program chose a single incorrect key for the entire excerpt. The solution here may lie in harmony; the prevalence of G-minor triads here should perhaps have cued the program that G minor was the correct choice. Possibly a preference rule could be added that would give a bonus to a key for each occurrence of its tonic harmony. (Of course, the pitch classes of each key's tonic harmony already favor that key through the key profile.) It does not appear that such a rule in itself would improve performance much on the Bach fugue subjects. Possibly it could be combined with a primacy rule, so that a bonus was given to the key whose (implied) tonic harmony began the piece. For the program to use harmonic information, of course, it would need to recover the harmonic structure of the input—a complex and difficult problem, particularly in the case of monophonic passages like the Bach fugue subjects (see Temperley, 1997).

Harmony is also important in other excerpts in the Kostka-Payne corpus. Consider Figure 11, part of an excerpt from a Schumann song. Kostka analyzes this passage as being in B♭ major, but the program finds it to be in F major. The problem is the French sixth chords, G♭-B♭-C-E. (A French sixth can be thought of as an inverted dominant seventh—C_7, in this case—with a flattened fifth.) The French sixths are followed by F-major chords,

9. Krumhansl's tests of the algorithm have already been described. Tests on the *Well-Tempered Clavier* fugue subjects by Longuet-Higgins and Steedman, Holtzmann, and Vos and Van Geenen have been mentioned also. Vos and Van Geenen also tested their model on fugue subjects of other composers. Holtzmann tested his model on a corpus of 22 other melodies, but it is not clear how these melodies were chosen. Longuet-Higgins and Steedman's, Holtzmann's, and Vos and Van Geenen's tests were all limited to monophonic excerpts. Winograd presents tests of three short pieces; Maxwell gives results for two pieces.

MIDI files for the excerpts from the Kostka-Payne workbook used in the test described here are publicly available at the website www.link.cs.cmu.edu/music-analysis. It is hoped that this will facilitate testing of other key-finding models on the Kostka-Payne corpus.

Fig. 11. Schumann, "Die beide Grenadiere," mm. 25–28.

as is customary; by convention, this progression would normally be interpreted as Fr_6-V in B♭ major (or minor). However, the progression involves two pitches outside of B♭ major (G♭ and E), so it is not surprising that B♭ major is not chosen by the current model. The Fr_6-V progression is then repeated in other keys, causing further problems for the model. In order to get such a passage right, the model would presumably have to know something about harmony: specifically, the conventional tonal implications of Fr_6-V progressions. Similar problems arise in excerpts involving other chromatic chords, such as Neapolitan chords and other augmented sixth chords.

The current tests suggest, then, that harmonic information is a factor in key finding. This is not surprising, in light of the studies by Brown (1988) and Butler (1989). These authors have shown that the same pitches arranged in different ways can have different tonal implications. In Figure 12a, for example, the pitches F-B-E-C clearly imply C major; arranged differently, in Figure 12b, they are much more ambiguous in their implications. These judgments were obtained experimentally as well (Butler, 1989). A natural explanation for these results is that harmony plays a role in key finding. In Figure 12a, an implied progression of G_7-C clearly suggests C major; the E-F progression in Figure 12b is more ambiguous. In practice, however, it does not appear that such situations arise with great frequency; it is relatively rare that harmonic information is necessary. Most of the cases where it is needed involve chromatic chords whose tonal implications contradict the normal tonal implications of their pitch classes.

Another area where the program might be improved is the key-profile values themselves. These could undoubtedly be refined, although it is not

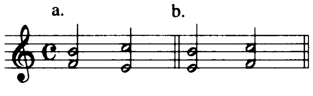

Fig. 12. The same set of pitches can have different tonal implications when arranged in different ways (Butler, 1989).

clear how much gain in performance would result. This could be done computationally, using a "hill-climbing" technique to arrive at optimal values. It could also be done by taking actual tallies of pitch classes in pieces (relative to the key), and using these as the basis for the key-profile values.[10] However, I will not explore this further here.

Spelling Distinctions and the "Line of Fifths"

One final kind of information that deserves mention is spelling. Both the original K-S algorithm and my modifications of it have assumed what might be called a "neutral" model of pitch classes, in which pitch events are simply sorted into 12 categories. In music notation and tonal theory, however, further distinctions are normally made between different spellings of the same pitch, for example A♭ and G♯; we could call these categories "tonal pitch classes" (TPCs), as opposed to the 12 "neutral pitch classes" (NPCs) of pitch class set theory. These distinctions are often applied to chords and keys as well. We can imagine TPCs represented on a "line of fifths," similar to the circle of fifths except extending infinitely in either direction. I have proposed elsewhere that the spelling labels of pitch events are an important part of tonal perception and, furthermore, that these labels can be inferred from context without relying on top-down key information, using preference rules (Temperley, in press). Most importantly, there is a preference to spell events so that they are close together on the line of fifths; for example, given the pitches D♯-E-F♯-G♯/A♭, there is a preference to spell the final event as G♯ rather than A♭ because the first spelling locates this event closer to previous events. (Harmony and voice leading also play a role in spelling.)

If it is possible to infer spelling labels without using key information, this raises the possibility that spelling might be used as input to key determination. Tests of the original K-S algorithm showed a number of cases in which this might be useful. Consider Figure 13—measure 65 from the Bach Courante discussed earlier, containing the pitches G♯-F-E-D-C-B. The K-S algorithm chooses F minor for this measure. If the first pitch event were spelled as A♭, this would be a reasonable choice. If the first pitch is G♯, however, F minor is clearly incorrect; A minor is much more likely. It seems plausible that the spelling of the first event as G♯ could be determined from context—for example, by its voice-leading connection to the A in the next measure; the key-profile model could then distinguish between the different tonal implications of G♯ and A♭. What would such a TPC key profile

10. If the key-profile values were to be based on pitch class distribution in pieces, this information would have to be gathered relative to the *local* key rather than the main key—unlike the pitch class tallies by Krumhansl and Butler mentioned earlier (see footnote 4).

Fig. 13. Excerpt from Bach's Cello Suite No. 3, Courante, mm. 65–66.

look like? A straightforward proposal is shown in Figure 14, for the key of C major. The "line of fifths" is represented on the horizontal axis. TPCs that are diatonic relative to C major have the same values as in the NPC profile (as shown in Figure 4). All other TPCs close to the tonic (within five steps to the left or six steps to the right) are given a value of 2.0, except for $b\hat{7}$ (this is the same as for chromatic NPCs in my original profile); all other TPCs are given a value of 1.5. For example, E is $\hat{3}$ of C major, and thus is given a value of 4.5 (as in my NPC profile); F♭ is chromatic relative to C major, thus its value is 1.5. Profiles could be constructed for minor keys on the same principles.

A version of the key-finding program was devised using TPC profiles of this kind, and it was tested on the Kostka-Payne corpus. (Spelling labels for notes were added to the input files, usually corresponding exactly to the spellings in the notated score.)[11] The TPC version attained a score of 87.4% correct: a modest improvement over the NPC version's score of 83.8%. In some cases, the reasons for the superior performance of the TPC profile are quite subtle. In Chopin's Mazurka op. 67 no. 2 (discussed earlier), whereas the NPC version of the program mistakenly analyzed the excerpt as being in B♭ major, the TPC version correctly identified the key as G minor. This excerpt contains many F♯s; the TPC profile considers F♯ to be less compatible with B♭ major than G♭ would be. That G♭ is more compatible with B♭

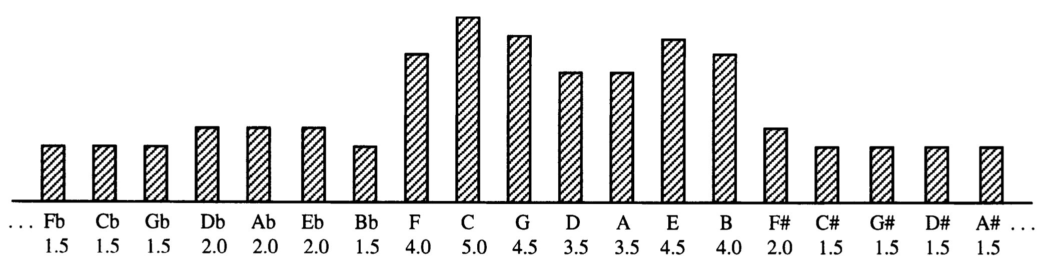

Fig. 14. A "tonal-pitch class" key profile for C major.

11. Occasionally, spellings of pitches may be chosen for reasons of notational convenience. For example, a composer might modulate from D♭ major to A major, avoiding the difficult-to-notate (but musically more logical) B♭♭ major. In such cases, I used the musically logical spelling rather than the notated one.

major than F♯ may seem, at best, a subtle distinction, but taking these distinctions into account allowed the program to modestly improve its performance.

Besides the gain in performance yielded by the TPC version, one might argue that the spelling of key names themselves—D♭ major versus C♯ major—is musically important. The program tested earlier was incapable of making such distinctions and made numerous mistakes (although it was not penalized for these). The TPC-based model performs significantly better in this regard, usually choosing the correct spelling for each key.

An Alternative Approach to Modulation

The preceding algorithm adopts a simple approach to modulation that proves to be highly effective: impose a penalty for changing keys, which is balanced with the key-profile scores. Another possible way of incorporating modulation into the key-profile approach has been proposed by David Huron and Richard Parncutt (1993). Huron and Parncutt suggest that the key at each moment in a piece is determined by an input vector of all the pitch events so far in the piece, weighted according to their recency. An exponential curve is used for this purpose. If the half-life of the curve is 1 s, then events 2 s ago will weigh half as much in the input vector as events 1 s ago.[12] (Huron and Parncutt's algorithm also involves weighting each pitch event according to its psychoacoustical salience; I will not consider this aspect of their model here.) Huron and Parncutt compared the results of this model with experimental data from studies by Krumhansl and Kessler (1982) and Brown (1988) in which subjects judged the key of musical sequences. In Krumhansl and Kessler's study, subjects heard chord progressions and judged the stability of different pitches at different points in the progression; these judgments were compared with key profiles to determine the perceived key. In Brown's study, subjects heard monophonic pitch sequences and indicated the key directly at the end; the same pitches were presented in different orderings, to determine the effect of order on key judgments. The Huron-Parncutt model performed well at predicting the data from Krumhansl's study; it fared less well with Brown's data.

An implementation of the Huron-Parncutt algorithm was devised, in order to test it on the Kostka-Payne corpus. A number of decisions had to be made. It seemed sensible to use the modified version of the key-profile values, because these appear to perform better than Krumhansl's original ones. Exactly the same input format was used as in my test; the same segments were used in this test as well. For each segment, a local input vector

12. The time point of each event was determined by its onset; duration was not considered.

was calculated. These input vectors were flat, as in my algorithm; each value was either 1 if the corresponding pitch class was present in the segment or 0 if it was not. For each segment, a "global input vector" was then generated; this consisted of the sum of all the local input vectors for all the segments up to and including that segment, with each local vector weighted according to the exponential decay function. The key chosen for each segment depended on the best key-profile match for that segment's global input vector. The use of segments here requires some explanation. Unlike my algorithm, the Huron-Parncutt algorithm does not actually require any segmentation of the input; it could conceivably make very fine grained key judgments. For example, the piece could be divided into very narrow time slices, say a tenth of a second, and a new key judgment could be made after each time slice, with each prior segment weighted under the exponential curve. (In this way, the duration of events would also be taken into account.) In the case of my algorithm, however, using segments and calculating flat input vectors for each segment was found to work better than counting each note and duration individually; as noted earlier, when notes are counted individually, a repeated note can have too strong an effect. It seemed likely that the same would be true for the Huron-Parncutt algorithm.

The algorithm was tested with various different half-life values to find the one yielding the best performance. This proved to be a value of 4.0 s. With this value, the algorithm scored correctly on 628.5 out of 896 segments, a rate of 70.1%.[13] Inspection of the results suggest that there are two reasons why this system performs less well than the preference rule system proposed earlier. One reason is its inability to backtrack. Very often, the segment where a key change occurs is not obviously in the new key; it is only, perhaps, a few seconds later that one realizes that a modulation has occurred, and what the new key is. Another problem with the algorithm is that it has no real defense against rapid modulation; with a half-life of 4 s, the algorithm produced 169 modulations (Kostka's analyses contained 40 modulations). Raising the half-life value reduced the number of modulations, but also reduced the level of performance.

While the exponential decay model performs disappointingly on the Kostka-Payne corpus, it may be of interest in other ways. It may well be a better model of actual listening than the preference rule algorithm—for nonexperts at least and perhaps for experts as well. The degree to which tonal backtracking takes place in actual listening is not clear. It might also be possible to improve the decay model to make it more capable of expert judgments. For example, the input vectors could be weighted with subse-

13. The model was also tested using weighted, rather than flat, input vectors (so that the value for a pitch class in a segment's local input vector was given by the total duration of events of that pitch class in the segment). The level of performance was almost identical to the flat-input version; with a half-life of 4.0 s, it yielded a score of 69.8% correct.

quent pitches as well as previous ones, perhaps allowing the system to handle modulations more effectively. However, I will not pursue this further here.

Conclusions

I have argued here that the key-profile model can provide a successful solution to the key-finding problem. Although I have proposed some modifications in Krumhansl and Schmuckler's original model, the basic idea behind their model proves to be a very useful and powerful one. I have cited several other factors—spelling, harmony, and the "primacy" factor proposed by Longuet-Higgins and Steedman—which appear to play a role in key finding; however, these factors appear to be needed rather rarely. Of course, to say that a kind of information is not *necessary* for key finding does not mean that it is not *used* for key finding. This recalls a point made earlier: the fact that one has a model that performs well at a task performed by humans does not prove that humans do it the same way. Although it is true that a key-profile model can perform key finding pretty well without harmony, it might also prove to be the case that a harmony-based model can perform the task well without key-profile information. The same applies to other kinds of information. Several authors have suggested that some kind of pitch salience factors might be involved in key finding, so that not all pitch events carry equal weight in the key-determination process. Krumhansl (1990, pp. 108–109)suggests that accented or metrically strong events might be given greater weight; Huron and Parncutt (1993, pp. 158–160) point to psychoacoustical salience as a factor—for example, the fact that outer-voice pitches tend to be more salient than inner-voice ones. Longuet-Higgins and Steedman's "first-note" rule is in a way a kind of salience consideration as well. Although the current study has found few cases in which such information is required, that does not mean that the information is not used psychologically. To put it another way, it may be that key information is often contained in musical stimuli in more than one way. If this proves to be the case, then other kinds of evidence will have to be considered—experimental psychological data, for example—in deciding which factors truly are "key" in how key finding is actually done.[14]

References

Bharucha, J. J. (1987). Music cognition and perceptual facilitation: A connectionist framework. *Music Perception, 5,* 1–30.

14. Thanks are due to Paul von Hippel, Carol Krumhansl, and two anonymous reviewers for their helpful comments on this paper.

Brown, H. (1988). The interplay of set content and temporal context in a functional theory of tonality perception. *Music Perception, 5,* 219–250.
Brown, H., Butler, D., & Jones, M. R. (1994). Musical and temporal influences on key discovery. *Music Perception, 11,* 371–407.
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,* 219–242.
Cook, N. (1987). The perception of large-scale tonal closure. *Music Perception, 5,* 197–206.
Cuddy, L. L., Cohen, A. J., & Mewhort, D. J. K. (1981). Perception of structure in short melodic sequences. *Journal of Experimental Psychology: Human Perception & Performance, 7,* 869–883.
Cuddy, L. L., Cohen, A. J., & Miller, J. (1979). Melody recognition: The experimental application of rules. *Canadian Journal of Psychology, 33,* 148–157.
Holtzmann, S. R. (1977). A program for key determination. *Interface, 6,* 29–56.
Howell, D. (1997). *Statistical methods for psychology.* Belmont, CA: Wadsworth Publishing Co.
Huron, D., & Parncutt, R. (1993). An improved model of tonality perception incorporating pitch salience and echoic memory. *Psychomusicology, 12,* 154–171.
Kastner, M. P., & Crowder, R. G. (1990). Perception of the major/minor distinction: IV. Emotional connotations in young children. *Music Perception, 8,* 189–202.
Keller, H. (1976). *The Well-Tempered Clavier by Johann Sebastian Bach* (L. Gerdine, Trans.). London: W. W. Norton & Company.
Kostka, S. (1995). *Instructor's manual to accompany tonal harmony.* New York: McGraw-Hill, Inc.
Kostka, S., & Payne, D. (1995a). *Tonal harmony.* New York: McGraw-Hill, Inc.
Kostka, S., & Payne, D. (1995b). *Workbook for tonal harmony.* New York: McGraw-Hill, Inc.
Krumhansl, C. L. (1990). *Cognitive foundations of musical pitch.* New York: Oxford University Press.
Krumhansl, C. L., & Kessler, E. J. (1982). Tracing the dynamic changes in perceived tonal organization in a spatial representation of musical keys. *Psychological Review, 89,* 334–368.
Lerdahl, F. (1988). Tonal pitch space. *Music Perception, 3,* 315–349.
Lerdahl, F., & Jackendoff, R. (1983). *A generative theory of tonal music.* Cambridge, MA: MIT Press.
Longuet-Higgins, H. C., & Steedman, M. J. (1971). On interpreting Bach. *Machine Intelligence, 6,* 221–241.
Maxwell, H. J. (1992). An expert system for harmonic analysis of tonal music. In M. Balaban, K. Ebcioglu, & O. Laske (Eds.), *Understanding Music with AI.* Cambridge, MA: MIT Press.
Rosen, C. (1971). *The classical style: Haydn, Mozart, Beethoven.* New York: The Viking Press.
Temperley, D. (1997). An algorithm for harmonic analysis. *Music Perception, 15,* 31–68.
Temperley, D. (in press). The line of fifths. *Music Analysis.*
Temperley, D., & Sleator, D. (1999). Modeling meter and harmony: A preference rule approach. *Computer Music Journal, 23,* 10–27.
Thompson, W. F., & Cuddy, L. L. (1992). Perceived key movement in four-voice harmony and single voices. *Music Perception, 9,* 427–438.
Vos, P. G., & Van Geenen, E. W. (1996). A parallel-processing key-finding model. *Music Perception, 14,* 185–224.
Winograd, T. (1968). Linguistics and the computer analysis of tonal harmony. *Journal of Music Theory, 12,* 2–49.