Maximos A. Kaliakatsos-Papakostas,*
Andreas Floros,† and Michael N. Vrahatis**
*Department of Mathematics
University of Patras
Office: B/M 248
GR-26110 Patras, Greece
maxk@math.upatras.gr
†Department of Audiovisual Arts
Ionian University
Plateia Tsirigoti 7
GR-49100 Corfu, Greece
floros@ionio.gr
**Department of Mathematics
University of Patras
Office: B/M 241
GR-26110 Patras, Greece
vrahatis@math.upatras.gr

A Clustering Strategy for the Key Segmentation of Musical Audio

Abstract: Key changes are common in Western classical music. The precise segmentation of a music piece at instances where key changes occur allows for further analysis, like self-similarity analysis, chord recognition, and several other techniques that mainly pertain to the characterization of music content. This article examines the automatic segmentation of audio data into parts composed in different keys, using clustering on chroma-related spaces. To this end, the *k*-means algorithm is used and a methodology is proposed so that useful information about key changes can be derived, regardless of the number of clusters or key changes. The proposed methodology is evaluated by experimenting on the segmentation of recordings of existing compositions from the Classic-Romantic repertoire. Additional analysis is performed using artificial data sets. Specifically, the construction of artificial pieces is proposed as a means to investigate the potential of the strategy under discussion in predefined key-change scenarios that encompass different musical characteristics. For the existing compositions, we compare the results of our proposed methodology with others from the music information retrieval literature. Finally, although the proposed methodology is only capable of locating key changes and not the key identities themselves, we discuss results regarding the labeling of a composition's key in the located segments.

The notion of tonality is fundamental in Western music. Most aspects of tonal analysis are based on the relations between pitches, provided a context: the composition's key. The key specifies a set of notes (a seven-note subset of the twelve notes of the chromatic scale) that are perceived as being related, although the utilization of key differs according to musical style and historical period, among other factors. Western classical music typically changes key, or modulates (in the broadest sense), during the course of the piece. Content segmentation and characterization of such music are aided by identifying the composition's main key and the related keys into which it is likely to modulate. Provided an accurate segmentation of a piece at the locations where key changes occur, several other tasks can be performed more accurately, such as self-similarity analysis (Chai 2005) and chord recognition (Lee and Slaney 2007), among others. Additionally, the availability of an enormous number of digital music recordings makes musical content analysis an important tool for automatically categorizing large data sets. Towards this aim, the detection of points where key changes occur can help define the local characteristics of pieces, providing a basis for further semantic analysis.

There are two main branches of the field of music information retrieval: research involving symbolic music representations such as MIDI data, and research involving nonsymbolic data, namely, audio. This article examines the automatic segmentation of audio data, although some concepts can apply equally to segmentation of symbolic data. Specifically, the focus here is segmentation

Computer Music Journal, 37:1, pp. 52–69, Spring 2013
doi:10.1162/COMJ_a_00168

of a piece into parts composed in different keys, through clustering on chroma-related spaces. (The concept of chroma, also known as pitch class, identifies a pitch in the Western equal-tempered tuning, disregarding the pitch's register, i.e., the octave in which the pitch occurs.) Although the proposed methodology is capable only of key segmentation and not labeling (i.e., it locates key changes but not key identities), we also present results of labeling via simple template-matching techniques. The *k*-means algorithm (Hartigan and Wong 1979) is used and a methodology is proposed so that useful information about key changes can be derived, regardless of the number of clusters or the number of key changes. The proposed strategy relies solely on geometric properties of the chroma space and does not need training, avoiding the potential hazard of being ineffective on musical styles different from the ones it has been trained on.

Experimental results are reported on segmentation of (1) recordings of real compositions, together with a comparison between our proposed methodology and previous ones, and (2) artificial music data sets, the construction of which is described subsequently. The construction of these artificial pieces is intended to provide an additional tool for examining the behavior of the proposed approach under "laboratory" conditions, allowing the analysis of the model's capabilities using large data sets of pieces with predefined structure. With these artificial data sets, multiple key-change scenarios were included in order to perform an exhaustive efficiency assessment of the clustering strategy. Finally, key labeling of the segmented parts was applied to the data set of real compositions, with the goal of providing a robust and accurate framework for musical content characterization.

Previous Work, Motivation, and Aims

Several approaches have provided significant insights into the problem of automatic detection of key changes in audio. Some work has extended key detection to the detection of local keys, i.e., areas within a piece that are composed using different keys. These approaches can be divided in two main categories. The first uses a priori information about the expected chroma constitution of keys, either in the form of key templates (Krumhansl 1990; Temperley 2004, 2006), or trained/tuned hidden Markov models (HMM) (Chai and Vercoe 2005; Noland and Sandler 2006; Papadopoulos and Peeters 2009, 2012). The second category includes methods that explore geometric properties of the pitch space to detect key changes without prior information about key templates or expected key changes (Chuan and Chew 2007; Izmirli 2007; Chew 2002). Similar techniques have also been proposed for harmonic segmentation—i.e., dividing a piece of music into a sequence of distinct chords (Harte, Sandler, and Gasser 2006), instead of key segmentation discussed in this work. Finally, a weighted graph approach was also tested for simultaneous chord and key recognition (Rocher et al. 2010). Here a larger data set of 174 pieces was used, but with a small mean number of key changes (1.69) per piece. This last work, however, does not report on segmentation accuracy results.

All works related to localized detection of key changes utilize the chroma information of a piece and look for contiguous chroma segments that could belong to a single key. These chroma segments are expressed as *chroma vectors*. These are vectors that incorporate information about the presence and intensity of the twelve chroma within short segments of a piece. The HMM-related approaches define the tonal constitution of each chroma vector (in terms of HMM, the emission) by associating its probability of belonging to a certain key with a transition probability from the key of the previous vector. Additional information of higher musical structure (i.e., chords) has also been utilized to refine key and key transition probabilities (Papadopoulos and Peeters 2009, 2012). The work described in Izmirli (2007) utilizes nonnegative matrix factorization (NMF) on the chroma matrix V of a piece to produce a set of patterns W and an activation matrix H, so that $V = WH$. The pattern matrix encapsulates information related to the identity of all keys in V, and the activation of each pattern, shown in H, reflects the location of each key. A limitation of this methodology is that the number of keys in the piece needs to be known in advance, so that each pattern of W corresponds to a key. The advantage of our methodology, regarding segmentation, is that there is no necessity for the number of keys to be known in advance. An additional advantage of our approach is that when it uses NMF and principal component analysis (PCA), the number of projecting dimensions does not have a crucial meaning (i.e., does not reflect the number of keys), but serves entirely as a dimension-reduction mechanism to facilitate clustering.

Besides the key segmentation per se, an additional motivation of the work presented here is to emphasize the potential of our methodology from a musical perspective. Such a task should not be performed on individual music pieces, however, because this would restrict the scope of the produced results to the analysis of these music works. To tackle this problem, we propose the construction of data sets of "artificial" music, which incorporate the desired pre-specified musical structure. This approach enables us to produce a large number of different test cases with diverse musical characteristics, thereby allowing a deeper analysis of the effectiveness of a model under different tonal conditions.

The Proposed Strategy

In the proposed approach we examine the detection of key changes in musical audio content using clustering. Clustering is a way to separate a collection of objects into groups, such that objects belonging to the same group are more similar than objects of different groups. This technique has been used in a wide range of applications (Jain, Murty, and Flynn 1999), with the notion of object similarity being defined in dependence on a specific problem. In the proposed application, similarity is measured in the chromatic tonal domain, calculated for short musical segments. This section analyzes the proposed clustering methodology for key segmentation of music recordings. The presentation includes a parallel demonstration of the tasks described, based on Dvorak's *Humoresque* No. 7, which is composed in two keys: G-flat major and F-sharp minor.

Feature Representation

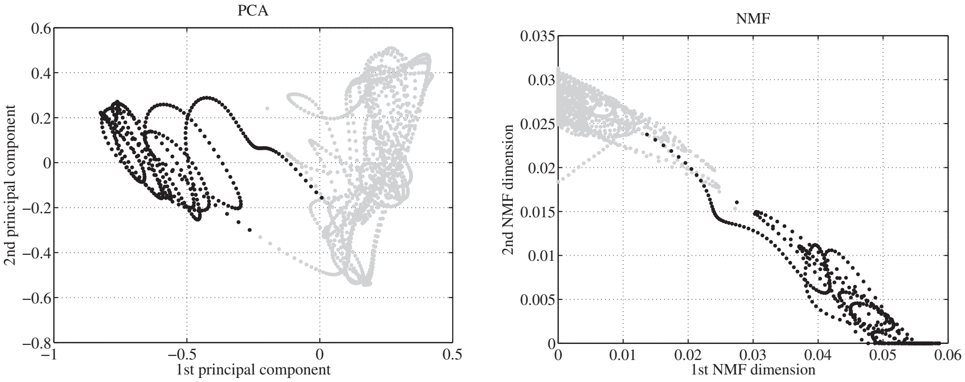
In our approach, we use the chroma energy normalized statistics (CENS) (Müller, Kurth, and Clausen 2005), obtained using the Chroma Toolbox (Müller 2010; Müller and Ewert 2011) for MATLAB. The sampling rate of the pieces that were used for this research was 44,100 Hz. The methodology of the Chroma Toolbox, however, uses down-sampled versions of the signals in order to achieve greater resolution accuracy at lower frequencies (Müller and Ewert 2011). Thus, with the utilization of a constant-Q multirate filter centered at the frequency of each pitch, the chroma profile is evaluated within frames of 0.1 sec. The CENS representation is a statistically smoothed transformation of this chroma profile, achieved through quantization and component-wise convolution of the chroma profile of each frame with its neighbors, using a Hanning window. The window size was selected to be $w = 45$ frames (4.5 sec) in order to have largescale tracking of the chroma activity, avoiding potential misclassification of frames caused by articulations or chromatic passages. It could be argued that the window size should be relative to the tempo of the piece rather than a fixed time unit. This would be worth examining in future research.

In a mathematical sense, the CENS representation transforms the recorded piece into a real matrix $C \in \mathbb{R}^{12\times F}$, with twelve rows—one for each chroma—and F columns, with F being the number of time frames. Our goal is to construct an algorithm that uses C to detect the time position of key changes through clustering all frames, not only in the twelve-dimensional tonal space of C, but also in spaces of reduced dimension derived using PCA and NMF.

Dimension Reduction in the Chroma Space

Before presenting the proposed approach, we briefly but rigorously provide a description of the PCA and NMF dimension reduction techniques and their parts that are associated with clustering. For the PCA, we obtain the covariance matrix S of C_c, which is the centralized per row C matrix (a matrix is centralized per row if we subtract the mean

Figure 1. Two-dimensional projections of each frame of a chroma energy normalized statistics (CENS) matrix using (a) principal component analysis (PCA) and (b) nonnegative matrix factorization (NMF). The frames presented as black dots belong to parts of the piece composed in G-flat major, and the ones presented with gray dots belong to the F-sharp minor part.



(a)

(b)

value of the respective row from each element), by $S = C_c \cdot C_c^T$, $S \in \mathbb{R}^{12\times12}$, where the exponent T denotes matrix transposition. The eigenvectors of S corresponding to the higher absolute eigenvalues are called *principal components*. We denote with Z_m the $\mathbb{R}^{12\times m}$ matrix whose columns are eigenvectors that correspond to the m higher eigenvalues. A projection of each frame of C_c on the subspace of the m principal eigenvectors is obtained by the multiplication $P = Z_m^T \cdot C_c$. Clustering is performed on the projection vectors P. Alternatively, dimension reduction using NMF can be accomplished by factorizing the nonnegative matrix $C \in \mathbb{R}^{12\times F}$ with two nonnegative matrices $W_m \in \mathbb{R}^{12\times m}$ and $H_m \in \mathbb{R}^{m\times F}$, where $m < 12$, so that $C = W_m \cdot H_m$. The matrix W_m concentrates the basic patterns of C, and H_m provides a linear combination of the basic patterns for the reconstruction of C. This means that the H_m matrix has the coordinates of each pattern of C in the projection space created by W_m, which are the tonal patterns. Thus, clustering is performed on H_m.

Although dimension reduction will not necessarily lead to a substantial reduction of complexity of the clustering algorithm, Ding and He (2004) and Lazar and Doncescu (2009) have shown that in some cases the utilization of PCA and NMF improves clustering performance. Additionally, dimension reduction allows visualizations that clarify the motivation and functionality of the proposed method. PCA and NMF have previously been used for finding the overall key of a piece (Izmirli 2006) and for estimating the local key (Izmirli 2007). With PCA projection on the first two principal components and by utilizing two-dimensional NMF, we may obtain representations of each frame of C, such as the ones depicted in Figure 1. Frames that belong in the music parts composed in G-flat major are depicted in black dots, and F-sharp minor parts are presented with gray ones. Clearly, from the visualization provided by both dimensionality-reduction techniques we may observe that frames belonging to a different composition key are "gathered" in a different area. This is accomplished by the "blurring" effect caused by the convolution of the chroma vectors applied on consecutive frames with the CENS algorithm, because every frame captures tonal characteristics of its neighboring frames.

Locating Key Clusters

Figure 1 shows that clustering allows us to separate the two groups of frames that belong to different keys. Indeed, the application of the k-means clustering algorithm on the PCA reduced space of Figure 1a for two clusters is satisfactory, as Figure 2a illustrates. A piece may be composed using an arbitrary number of keys, however, thus a priori assumptions about the number of clusters cannot be made. If we apply the k-means algorithm on the same data assuming more than two clusters, we may not be sure about the meaning of the content of each cluster. Figure 2b illustrates the case where five clusters are considered.

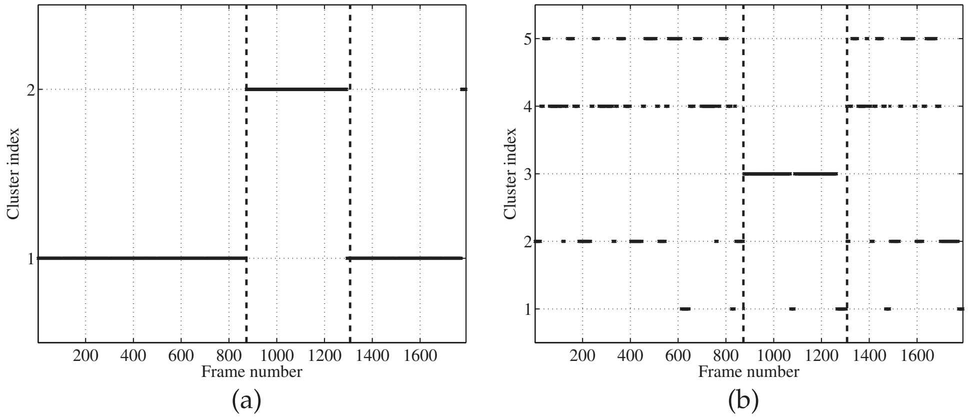


Figure 2. Application of the k-means algorithm expecting two clusters (a) and five clusters (b) for the data depicted in Figure 1a. The vertical dashed lines show the actual key transitions.

Finding Temporal Contiguity within Cluster Combinations

The fundamental idea of the key segmentation strategy presented here is to examine which cluster combinations are more likely to represent contiguous parts of the piece that are composed in the same key. By the term *cluster combination*, we mean the concatenation of several previously identified clusters into a larger one. For this, we can rely on the temporal relations between time frames. In other words, if a proper cluster combination contains frames belonging to a single key, their concatenation would exhibit temporally contiguous behavior, not sporadically distributed segments. Temporal contiguity could also be approached with more-straightforward calculations on chroma values. For example, future work could incorporate a single "change point" detection scheme based on the sequence of chroma vectors (e.g., locating peaks in the derivative of the chroma vectors).

Accumulation and Gradient Difference Curves

To the end of locating cluster combinations that incorporate a single key, we examine the temporal contiguity of all possible cluster combinations. We do this by forming all possible pairs of complementary subsets, $\mathcal{A}$ and $\mathcal{B}$, of the main superset of all clusters, $\mathcal{S}$, so that $\mathcal{A} \cap \mathcal{B} = \emptyset$ and $\mathcal{A} \cup \mathcal{B} = \mathcal{S}$. The selection of every subset should be considered only once; e.g., if $\mathcal{A} = X$ and $\mathcal{B} = Y$ at some selection stage, then at a latter stage we will skip the selection $\mathcal{A} = Y$ and $\mathcal{B} = X$. For example, in Figure 2b we have a superset of five clusters, $\mathcal{S} = \{1, 2, 3, 4, 5\}$, for which we can create 15 different pairs of subsets with the aforementioned properties, because, if we assume that the cardinality of $\mathcal{S}$ is n, $n = |\mathcal{S}|$, then the number of different $\mathcal{A}$ and $\mathcal{B}$ sets is

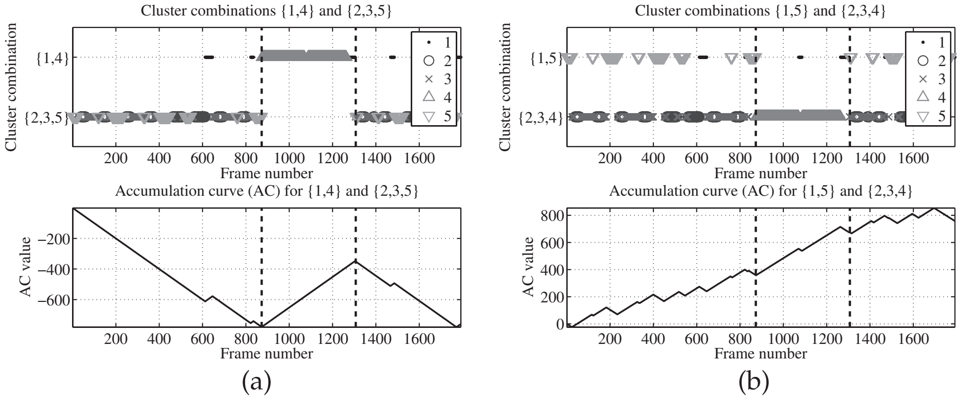
$$\sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} \binom{n}{i},$$

an equation that describes the selection of all x-tuples up to $x = \lfloor \frac{n}{2} \rfloor$. Note that the selection of x-tuples with $x > \lfloor \frac{n}{2} \rfloor$ would result in re-selection of all segmentations, i.e., for $x = \lfloor \frac{n}{2} \rfloor + j$ we would have the same selection as for $x = \lfloor \frac{n}{2} \rfloor - j$, $j \in \{1, 2, \ldots, (\lfloor \frac{n}{2} \rfloor - 1)\}$.

Figure 3 depicts two such pairs of complementary subsets and their temporal representation, together with the respective accumulation curve (AC) of each pair. The AC of a cluster subset reflects the temporal contiguity of the clusters included in this subset. Outlining the way an AC is estimated, we first scan the piece from the beginning, frame by frame. If a frame belongs to cluster subset $\mathcal{A}$, the AC value is increased by one, otherwise it is decreased by one. Before scanning the first frame, the initial AC value is zero. Algorithm 1 in Figure 4 provides a more rigorous presentation of the AC computation procedure. The ACs of the cluster subsets depicted in Figure 3a reveal that when two cluster subsets have captured parts of the piece composed in different

Figure 3. Two sets of cluster combinations and their respective accumulation curves (ACs). Sets (a) $\mathcal{A} = \{1, 4\}$ and $\mathcal{B} = \{2, 3, 5\}$ exhibit temporal contiguity, whereas sets (b) $\mathcal{A} = \{1, 5\}$ and $\mathcal{B} = \{2, 3, 4\}$ are temporally noncontiguous. The vertical dashed lines show the location of the actual key changes.

Figure 4. Algorithm for computing the cluster accumulation curve (AC).



(a)

(b)

Figure 3

Algorithm 1 AC computation

Input: (**i**) the total number of frames F, (**ii**) cluster labeling of each frame in C clusters $K : [1, F] \to \{1, 2, \ldots, C\}$ and (**iii**) subsets of cluster labels $\mathcal{A}$ and $\mathcal{B}$

Output: The AC as a function of frames AC : $[1, F] \to \mathbb{Z}$

```
$\alpha \leftarrow 0$
for $i = 1$ to $F$ do
    if $K(i) \in \mathcal{A}$ then
        $\alpha \leftarrow \alpha + 1$
    else
        $\alpha \leftarrow \alpha - 1$
    end if
    AC$(i) \leftarrow \alpha$
end for
```

Figure 4

keys, the locations where the key changes occur are described by a change of direction between large monotonic segments. On the contrary, this interchange is not present if the cluster subsets have not captured parts in different keys, as illustrated in Figure 3b. Because all cluster combinations are examined, an a priori assumption about the number of clusters is not required, although this may have an influence on the estimation accuracy.

Given the AC of a pair of cluster subsets, we construct the gradient difference curves (GDCs), which indicate the positions where this AC changes monotonicity, (see Figure 5). As mentioned earlier, the change in monotonicity would reveal the position of a key change, if the parts before and after this location are sufficiently long. The term *sufficiently long* refers to an analysis window of sufficient length that information on key, rather than chords, is captured. Previous approaches have addressed this issue by providing results for different constant values describing time analysis windows (Izmirli 2007), or the a priori probability that a key change will not happen (Chai and Vercoe 2005). As in these two earlier studies, this article reports results of an experimental procedure to measure the effectiveness of the proposed approach for different time windows.

The GDC value of a frame f is an estimate of the absolute difference of the gradients (i.e., slopes)

Algorithm 2 GDC computation

Input: (**i**) the total number of frames F, (**ii**) a constant integer T and (**iii**) an AC as a function of frames

Output: The GDC as a function of frames GDC : $[1, F] \rightarrow \mathbb{Z}$

```
for $i = 1$ to $F$ do
    $\text{GDC}(i) \leftarrow 0$
end for
for $i = T + 1$ to $F - T$ do
    $\text{GDC}(i) \leftarrow \left| \left( \frac{\text{AC}(i) - \text{AC}(i-T)}{T} \right) - \left( \frac{\text{AC}(i+T) - \text{AC}(i)}{T} \right) \right|$
end for
```

Figure 5

Figure 5. Algorithm for computing the gradient difference curve (GDC) of the ACs.

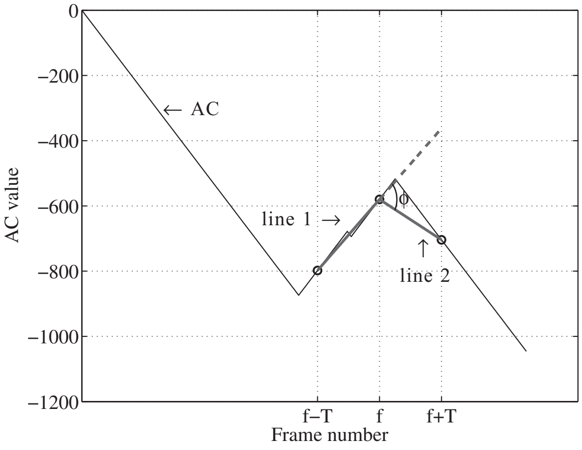


Figure 6

Figure 6. Computation of the GDC using the absolute gradient differences of line 1 and line 2. Lines 1 and 2 join the circled points of the AC. The GDC value on the f point is interpreted geometrically as the tangent of ϕ.

between two lines, as depicted in Figure 6. Line 1 joins the points $[f - T, \text{AC}(f - T)]$ and $[f, \text{AC}(f)]$, and line 2 joins $[f, \text{AC}(f)]$ and $[f + T, \text{AC}(f + T)]$. These three points are marked with circles. The absolute difference of the gradients of these lines provides a means for computing the monotonic behavior of the AC function T frames before and after a specified frame f. The absolute difference can be interpreted geometrically as the tangent of angle ϕ. The GDC value is bounded in $[0, 2]$, because the gradient of the AC is bounded in $[-1, 1]$. The calculations that produce the GDC variation are described in Algorithm 2 in Figure 5.

The locations of large changes of monotonicity in the ACs would result in higher peaks at the GDCs, reflecting greater absolute differences between the gradients. By the term *large monotonicity changes* we refer to the points where the direction of the ACs is changing, i.e., from ascending to descending and vice versa. Segments of large monotonicity indicate persistence in a tonal trend that is expressed by a combination of clusters, a fact that might reveal the existence of a localized key. As we are interested in detecting regions of large changes of monotonicity, we define a threshold for minimum absolute difference in gradients, denoted by m, above which we consider that a key change has occurred. Values below this threshold are ignored, as we assume they contain no important information about key changes. In Figure 7 we demonstrate this for the two resulting cluster subsets of Figure 3. For demonstrational clarity, we have chosen $T = 250$ and $m = 1.5$. The m threshold is illustrated with a horizontal dashed line. The GDC values below m are illustrated with a thin gray line, and the values exceeding the threshold are black and thicker. Figure 7a shows that the respective $\mathcal{A}$ and $\mathcal{B}$ subsets provide strong indications for a key change, approximating the actual locations of the key changes relatively accurately. On the contrary, in Figure 7b, no indication of a key change can be provided.

The content that has surpassed m among all the GDCs is summed to form a unique curve of GDC contributions. The sum of the GDCs of all possible

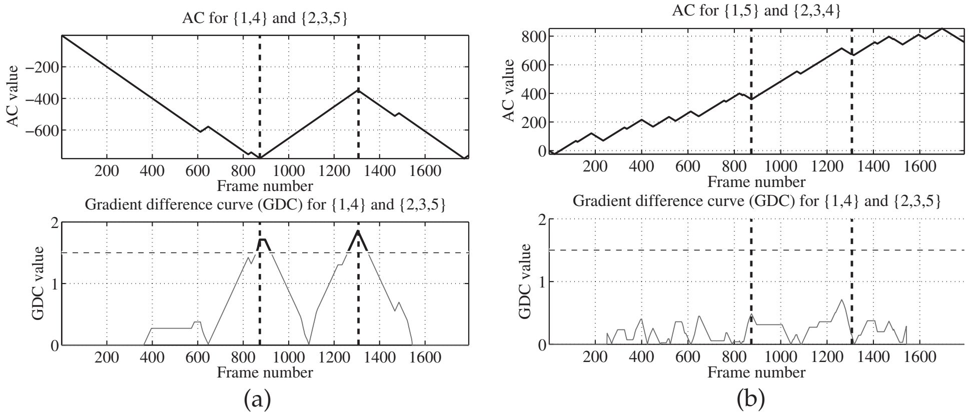


Figure 7

Figure 7. The ACs together with the GDCs of the two sets of cluster combinations depicted in Figure 3. The horizontal dashed line defines the minimum absolute gradient difference threshold (m). The GDC values below and above the threshold are illustrated with thin gray and bold lines, respectively.

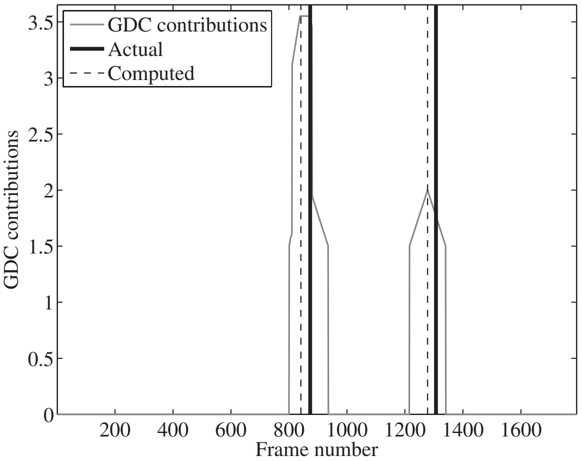


Figure 8

Figure 8. The final contribution of each GDC expressing the change potential. The computed key segmentation positions are exhibited with vertical slim dashed lines, and the positions of actual key changes are shown as thick lines.

cluster subsets provides a curve such as the one depicted in Figure 8, where locations of potential key change are shown as positive valued peaks. The final segmentation is realized at the local maxima of the GDC curve.

Experimental Settings for Performance Assessment

As highlighted in previous studies (Chai and Vercoe 2005; Izmirli 2007; Papadopoulos and Peeters 2009), there is a lack of data sets of musical compositions that are annotated with key change information. Due to this fact, experimental results are reported on small sets of pieces that have been manually analyzed by the respective authors. To address this issue, we have constructed an artificial data set with pieces that include a varying number of key changes between predefined keys. In this way a thorough examination of the proposed methodology can be conducted under any key-change scenario, providing an additional tool towards analyzing the capabilities of the key-change detection model with an abundance of test case paradigms for any given tonal structure. Besides the artificial data set, we have also included results from simulations on recorded compositions that were annotated by the first author. These compositions are shown in Table 1 and were all performed on piano. The CENS representation of these pieces, along with the label annotations, are available at cilab.math.upatras.gr/maximos/keyChangeDatasets, as well as on the DVD-ROM portion of the forthcoming 2013 *Computer Music Journal Sound*

Table 1. The Works Used as the Real Data Set, Denoted as R

Mozart - *Rondo alla turca* (K. 331)
Paderewski - *Minuet* (op. 14, no. 1)
Rubenstein - *Mélodie* in F (op. 3, no. 1)
Dvorak - *Humoresque* No. 7 (op. 101, no. 7)
Mozart - *Piano Sonata* No. 15 (K. 533)
Schubert - *Moments musicaux* No. 2 (op. 94, no. 2)

and Video Anthology. Different recordings of these pieces were also used as piano performances in Chai and Vercoe (2005).

Synthesis of the Artificial Key Change Scenarios

The motivation for creating artificial key-change scenarios is twofold. First, it provides arbitrarily large data sets that are automatically labeled, thereby allowing the extensive examination of a model's potential. Second, the fact that these scenarios have predefined structure allows an analysis of musical texture to be performed on numerous test cases, bypassing the narrow scope of conclusions imposed when analysis is performed on only a limited number of musical pieces.

Our approach for constructing key-change scenarios is quite straightforward: We concatenate excerpts from different pre-existing pieces that are known to be in different keys. The concatenation takes place in the time domain, so that the temporal "blurring" effect of the CENS representation smooths the chroma trace of the transition between the concatenated excerpts. The pieces that we used are piano performances of the 24 fugues from J. S. Bach's *Well-Tempered Clavier*, performed by John Lewis Grant and available at the Piano Society Web site (Stöhlbrand, Helling, and Wallaart 2011). These fugues are composed in the 24 major and minor keys. The choice to utilize these pieces in piano recordings was based on several criteria that have to do with the motivation and aims of this work. First, large parts of these pieces were composed in a single key, and (in the edition we consulted) the key is stated in the title. This facilitated the key labeling process. Second, the fugues of Bach incorporate passages of ambiguous tonality that are not clearly composed in a single key, introducing an effect of "tonal noise" in the artificial data set. This amplifies the realism and increases the difficulty of key separation. Finally, because this work does not focus on audio-to-symbolic music interpretation, we choose piano recordings, which offer a relatively clear high-level transcription. Future work, however, should incorporate a more diverse collection of musical styles.

More specifically, to construct an artificial piece that changes from key A to key B, we concatenate two parts, the one from the fugue composed in key A and the other from the fugue in key B. In order to have an indication that the parts chosen are mostly composed in the keys denoted by the fugue title, we demand that this part matches a key template of the intended key, using the templates proposed in Temperley (2006) and linear correlation (Temperley 2004) as a matching technique. The fact that these parts may elaborate more than one key (template matching is not perfect) increases the difficulty of key separation by imposing the effect of tonal noise mentioned earlier. Thus, the robustness of the subject methodology may be tested on a variety of musical tasks captured in these "noisy" artificial data sets. The concatenation of more than two parts would result in a multiple key-change scenario. The length of the parts under concatenation would also vary, depending on the needs of the experimental procedure.

The approach we decided on was to give each of the concatenated parts an equal duration of 30 seconds. We constructed multiple key-change scenarios: two keys of the same mode (but different tonic), two keys with different mode (and possibly different tonics), and two to five keys with random changes of tonic and mode. In general, the effectiveness of a key-change detection algorithm may vary, depending on the distance of the modulating keys. For the clustering process there is no dependence on the temporal sequence of keys, as long as within-key contiguity remains intact. To provide information about the effectiveness of the proposed strategy in certain modulations, we also focus on the results provided by key changes of varying distance, both with and without change of mode. The notion of distance between keys is discussed in greater detail later in this article.

The Artificial Data Sets

Three types of artificial data sets were constructed. The first contains pieces that each have one modulation between two keys of the same mode; collectively the set contains all possible modulations between keys of the same mode. Because our approach is independent of the order of key change, only six modulations need to be monitored here, from one to six semitone steps. For example, a step of seven semitones from C major to G major would be the same as a step of five semitones from G major to C major. The artificial data set of transitions between keys of the same mode is denoted as A^1_{same}. For each of the six possible modulations starting from each of the 24 keys, we randomly chose three pairs of excerpts from the corresponding fugues, thus the A^1_{same} incorporates $24 \cdot 6 \cdot 3 = 432$ test pieces.

The second artificial data set includes pieces that have a single modulation between keys of different mode. Given the independence of key order, we choose initial segments that belong to a major fugue and concluding segments that belong to a minor one. We construct three pieces for each of the twelve possible semitone step transitions and for all twelve possible initial keys. We denote this data set as $A^1_{\text{different}}$, and it contains $12 \cdot 12 \cdot 3 = 432$ test pieces.

The third artificial data set comprises pieces with one to four key changes, thus with at least two and no more than five keys. The possible combinations of keys in the multiple key-change scenarios for three keys and above are overwhelming. So we create concatenations of parts from random keys with the constraint that two successive parts should not be in the same key. The sets that include these pieces are denoted as $A^n_{\text{random}}, n \in \{1, 2, 3, 4\}$, where the superscript n denotes the number of key changes. For example, the set of pieces with three key changes is denoted as A^3_{random}. Each one of the sets $A^n_{\text{random}}, n \in \{1, 2, 3, 4\}$ includes 100 artificial pieces. Thus, we have 300 artificial pieces with more than one key change and an additional set of 100 pieces with a single random key change. One may ask why we would need the A^1_{random} set, because we have already constructed A^1_{same} and $A^1_{\text{different}}$. The reason is that we first experiment on finding the optimal T (size of the time window) and m (minimum threshold) values for the A^1_{random} and then use these values for analyzing the behavior of our method under the tonal conditions imposed by the A^1_{same} and $A^1_{\text{different}}$ data sets.

The data sets are available online at cilab.math.upatras.gr/maximos/keyChangeDatasets.

Results

In this section we report results on three different types of data. First, we present results on artificial pieces with one to four modulations between random different keys (A^n_{random}, $n \in \{1, 2, 3, 4\}$). Then we report the results on artificial pieces with single key modulations and with simulations for transitions between keys with and without change of mode (A^1_{same} and $A^1_{\text{different}}$). Finally, we report on results obtained for the six real-world pieces (R).

Accuracy Metrics and Parameter Estimation

For measuring the accuracy of key segmentation, two criteria are used that are widely used in music information retrieval (MIR): *precision* and *recall*. Precision describes the number of correctly identified positions of key change as a percentage all the positions identified. Recall describes the number of correctly identified positions, as a percentage of the the annotated "ground truth" positions of key change. Strictly speaking, if L is the set of transitions located by the algorithm and C is the set of the annotated transitions, then precision is computed by $p = |L \cap C| \,/\, |L|$ and recall by $r = |L \cap C| \,/\, |C|$, where $|X|$ denotes the number of elements in set X.

A good result is described by combined high values of both precision and recall. In the presentation of the following results we consider the term *accuracy*, which is the f-measure, denoted by f_m, provided by precision and recall. Accuracy is computed as $f_m = 2pr/(p + r)$. We consider a position of key change to be detected correctly if it is located within 5 sec before or after the actual key change. The margin of 5 sec is comparable to the ones used in previous works (Chai and Vercoe 2005; Papadopoulos and Peeters 2009).

The simulations on the A^n_{random} random and R data sets are conducted for T values equally spaced from 50 to 300 frames at increments of 25 frames, and for m ranging from 0.9 to 1.9 at increments of 0.1. The simulations use the k-means classifier and assume three to five clusters. All presented results are the mean values over ten different clustering simulations with different

Table 2. Accuracy (f-measure) of Results for the A^n_{random} Data Set

Key Changes (n)	*Clusters*	*Dimensions*	*Accuracy*	*Precision*	*Recall*	*T*	*m*
Chroma							
1	5	–	0.9700	0.9700	0.9700	275	1.0
2	3	–	0.9233	0.9367	0.9100	250	1.0
3	4	–	0.8677	0.8825	0.8533	250	1.0
4	4	–	0.8421	0.8952	0.7950	275	1.0
PCA							
1	5	5	0.9800	0.9800	0.9800	275	1.0
2	3	4	0.9357	0.9467	0.9250	250	1.0
3	4	6	0.8716	0.8835	0.8600	250	1.0
4	4	5	0.8459	0.9005	0.7975	275	1.0
NMF							
1	5	3	0.9600	0.9600	0.9600	275	1.0
2	3	4	0.9109	0.9383	0.885	275	1.0
3	4	3	0.8524	0.8955	0.8133	275	1.1
4	5	4	0.8070	0.8303	0.7850	250	1.2

The table shows results when requiring different numbers of clusters, with clustering on the chroma space and on spaces of reduced dimensions with PCA and NMF.

random centroid initialization. It should be also noted that in the present work we do not aim to provide the optimal parameters for the model under discussion. We aim, rather, to explore the quality of the results obtained with the utilization of different parameters. Thus, the presented results provide a coarse description of the potential of our strategy and allow for further analysis of its effectiveness over different tonal conditions.

Artificial Key Change Scenarios

The synthesis of the artificial data sets has been formulated in such a way that a thorough examination of the strategy under investigation can be carried out for almost any key-change scenario. We analyze the responses of our clustering model under two groups of experimental procedures. The first one examines the accuracy when a varying number of key changes occur. The second one examines the accuracy when only a single change occurs.

Random Number of Key Changes

Table 2 presents the accuracy of the proposed strategy for the A^n_{random}, $n \in \{1, 2, 3, 4\}$ data sets when clustering is performed on the chroma space and on spaces with reduced dimensions, using either PCA or NMF. In the latter two cases, the results that we demonstrate are the best ones among all possible numbers of reduced dimensions, from one to six. The column Clusters shows the number of clusters used. In the case of the single key change, all clustering schemes provided better results when five clusters were required. In the case of two key changes, the best clustering performance was achieved when three clusters were assumed. In the cases of three and four key changes, the best performance was achieved when four clusters were assumed, except in the case of four key changes with clustering using NMF. Here, an extremely small improvement in accuracy (0.0004) was achieved by using five clusters. The fact that using three and four clusters produces the best results in the cases of two and three key changes, respectively, appears to be related to the number of keys that take part in the artificial pieces. When the piece has two key changes it includes three keys, which seem to be better separated with the use of three clusters. The same happens in the case of three key changes with the use of four clusters.

On the other hand, in the case of four key changes (i.e., where five keys are included in a composition),

Table 3. Results for the Artificial Data Sets $A^1_{\text{different}}$ and A^1_{same}

	$A^1_{\text{different}}$		A^1_{same}	
Semitone Steps	*Common Tones*	*Accuracy*	*Common Tones*	*Accuracy*
1	3 (3)	1	2	1
2	6 (5)	0.8056	5	0.9684
3	2 (3)	1	4	1
4	6 (5)	0.8889	3	1
5	3 (4)	1	6	0.8991
6	4 (4)	1	2	1
7	5 (4)	0.9444	–	–
8	2 (3)	1	–	–
9	7 (6)	0.7698	–	–
10	2 (3)	1	–	–
11	5 (4)	1	–	–
12 (0)	4 (5)	0.9722	–	–
	Correlation: −0.8365 (−0.8199)		Correlation: −0.8681	

The results displayed here are for a single key transition between keys of different and same modes respectively, with $T = 275$ and $m = 1.0$.
The term *semitone step* refers to the distance between the tonic notes of the two keys, in semitones. The number of common tones between modulating keys is strongly negatively correlated with accuracy. For the major to harmonic minor transition study, the number of common tones and the respective correlation are demonstrated in parentheses. For further analysis on the number of common tones between transitions, the reader is referred to Table 4.

it seems that five clusters cannot separate all five tonal regions as effectively as four clusters can. In fact, these five tonal regions cannot be so clearly separated in any way, because if we try to create every possible combination between five keys, at least two of them will share at least six tones. This means that the chroma traces of at least two keys will be almost identical, because at least six out of their seven tones will be the same, producing an effect that could be described as *tonal saturation*. In this case, it seems that the algorithm works better when these two closely related keys are considered as a single key. It should be noted that there are combinations of four keys where the maximum number of common tones between all key pairs is four, something that does not create the aforementioned tonal saturation effect so intensely. The way that the number of common tones affects the performance of our approach is analyzed in the next paragraph, where the accuracy of single changes between keys with different number of common tones is measured.

Clustering in spaces of reduced dimensions with PCA has a mean accuracy (f-measure) $f_m = 0.8883$, and it slightly outperforms clustering in the chroma space (0.8793) and NMF (0.8726). Concerning the number of dimensions that provided the best results for PCA and NMF, there is no clear pattern that indicates the superiority of a specific number of dimensions. On the other hand, the size of the window (T) that provided the best results was always between 250 and 275. The minimum gradient threshold (m) was mostly at 1.0, except for two cases where the values 1.1 and 1.2, respectively, produced better results. It should be noted that in the case of the artificial pieces the value of T is related to the length of the concatenated parts (300 frames = 30 sec). Similar results were obtained for the real-world pieces.

One Key Change

The results of the data sets A^1_{same} and $A^1_{\text{different}}$ are shown in Table 3. Because these results are

Table 4. The Number of Common Tones Between Two Keys

Circle of Fifths	Semitone Step		Common Tones	
Distance (d)	Major	Minor	Natural	Harmonic
0	0	9	7	6
1	5	2	6	5
2	10	7	5	4
3	3	0	4	5
4	8	5	3	4
5	1	10	2	3
6	6	3	2	3
5	11	8	2	3
4	4	1	3	3
3	9	6	4	4
2	2	11	5	4
1	7	4	6	5

The keys are ordered in clockwise traversal of the circle of fifths. In the first column, the distance (d) is measured as the minimum distance when traversing the circle of fifths clockwise or counter-clockwise. In the second and third columns, distance is measured in semitones from the tonic of the reference major scale (column 2) or its relative natural minor scale (column 3). The fourth column indicates the respective number of common tones if both keys are major or both are natural minor. The number of common tones between two keys is $7 - d$, except when $d = 6$. This formula results in a symmetry of the number of common tones, when keys are ordered by semitone distance. Therefore, measurements for rows 8–12 in column 4 of Table 3 are omitted, because they are mirrored by rows 2–6. The fifth column indicates the number of common tones if the second key is considered to consist of the notes of the relative harmonic minor scale.

intended to provide insights into the effectiveness of the proposed approach in all possible single-key-change scenarios, we provide results only for clustering on the chroma space, using five clusters. A presentation of the results provided with PCA and NMF is omitted. These latter results are similar to the ones presented here. For both sets the values $T = 275$ and $m = 1.0$ have been used, as indicated by the results of A^1_{random} in Table 2, when clustering is performed on chroma space. Table 3 also illustrates the number of common tones between the keys of each transition, and Table 4 displays all common tones between major and minor (natural and harmonic) keys when they are ordered by traversal of the circle of fifths. For major-to-minor transitions we distinguish two cases, one for natural minor and one for harmonic minor. The strong negative correlation (−0.8681 for A^1_{same} and −0.8365 [−0.8199] for $A^1_{\text{different}}$) between the number of common tones and accuracy validates the hypothesis that poorer performance is expected for less distant keys (i.e., keys that have a larger number of common tones).

The lowest accuracy among all single-transition scenarios is the transition between relative natural minor and major, which is denoted as the nine semitone step transition in Table 3. This is an expected result, because all seven tones are common in both keys that form the transition, forming a strong tonal saturation effect, as mentioned earlier. An additional comment should be made about the difference in accuracy between the two-semitone and four-semitone transitions of the $A^1_{\text{different}}$ transitions in Table 3. Because both transitions lead to keys that share the same number of common notes, one might expect similar levels of accuracy. This difference is probably a result of the different roles that the common notes play in these two pairs of keys. The common tones in the two-semitone step transition seem to incorporate musical coherence between the keys under modulation, probably reflecting the ii → I cadence.

Real-World Pieces

The best results provided by the T and m values examined for each clustering space in the case of real pieces, R, are displayed in Table 5. They are grouped according to the space in which clustering is performed with the k-means algorithm. The best segmentation performance was achieved for clustering in the chroma space, using four clusters. The accuracy achieved for all the examined T and m values, together with label accuracy (discussed later), are exhibited in Figure 9 (in a subsequent section). When clustering in the chroma space uses four clusters, for a large time window ($T = 275$) and a low minimum absolute gradient threshold

Table 5. Accuracy (f-measure) for the R Data Set

Clusters	*Dimensions*	*Accuracy*	*Precision*	*Recall*	*T*	*m*
Chroma Space						
3	–	0.7340	0.6409	0.8583	175	1.1
4	–	**0.8601**(0.16)	**0.8619**	**0.8583**	275	1.1
5	–	0.8292	0.8290	0.8167	275	1.1
PCA						
3	2	0.7609	0.8333	0.7000	250	1.4
4	6	**0.8252**(0.16)	**0.7944**	**0.8583**	275	1.1
5	3	0.8108	0.7750	0.8500	225	1.4
NMF						
3	1	0.7764	0.8714	0.7000	175	1.2
4	4	0.7628	0.7589	0.7667	150	1.5
5	5	**0.7896**(0.26)	**0.7643**	**0.8167**	275	1.0

The table shows results when requiring different numbers of clusters, with clustering on the chroma space and on spaces of reduced dimension with PCA and NMF. Best results are highlighted in **boldface**.

($m = 1.1$), an f-measure of 0.8601 was reached, with a precision of 0.8619 and recall at 0.8583. Both dimensionality-reduction techniques failed to yield better results for the R data set than the results achieved in the chroma space, regardless of the number of reduced dimensions. A combination of inversely related T and m values (i.e., large T with small m and vice versa) seems to produce better segmentation results. This relation that is demonstrated graphically with the white diagonal line in Figure 9a.

Comparison with Other Methodologies

To better understand the capabilities of the proposed methodology, we use a comparison with some basic types of algorithms for detecting key changes. These include a typical template matching approach and two HMM models: a basic HMM scheme and the approach presented in Chai and Vercoe (2005). The template matching algorithm assigns to each frame the key with the highest correlation to the respective key template proposed in Temperley (2006). Both HMM models incorporate an initial prior probability for each key (Pr), a transition probability matrix between keys (Tr), and a probability measure that associates each emission (frame) with all keys. For both HMM approaches, the initial prior probability indicates the probability that a piece begins with a certain key. This probability is adjusted as the uniform distribution among all 24 keys.

The transition probability matrix is different in the two HMM approaches. The basic HMM technique incorporates a transition matrix that assumes a higher probability for a key not to change, whereas equal probabilities are assumed for all other possible key transitions. This approach yields a transition matrix with larger values in the diagonal elements and smaller in the off-diagonal elements. Among several value combinations tested, the ones that produced the best results are diagonal values of 0.9886 and off-diagonal values of 0.0004. Furthermore, the probability that relates each frame to a key is obtained by correlation-template matching (using the Temperley templates) with a linear transformation of the $[-1, 1]$ correlation range to $[0, 1]$. The approach taken by Chai and Vercoe also assumes a greater probability that a key does not change, reflecting larger diagonal elements for the transition probability matrix, but does not consider equal probabilities for changes between same and different key modes (i.e., major and minor). This results in a transition probability matrix of the following

Table 6. Comparison of Results Achieved by our Approach and the Other Methodologies Considered in this Article

	Accuracy	*Precision*	*Recall*
Template based	0.1063(0.06)+	0.0571	**1.0000**
Simple HMM	0.7000(0.31)=	0.6319	0.8417
Chai HMM	0.6281(0.27)+	0.5784	0.7417
Clustering	**0.8601**(0.16)	**0.8619**	0.8583

The table displays segmentation accuracy (f-measure), precision, and recall for our approach and the other methodologies considered, while using the R data set. Standard deviation across all pieces is noted in parentheses, with the + and = symbols indicating statistically significant differences with the clustering strategy for $\alpha = 0.15$. The best results are highlighted in boldface.

form:

$$Tr = \begin{array}{c}
\begin{matrix}\overbrace{\hphantom{a\ \ b\ \ \cdots\ \ b}}^{12\text{ major}} & \overbrace{\hphantom{d\ \ c\ \ \cdots\ \ c}}^{12\text{ minor}}\end{matrix}\\
\begin{pmatrix}
a & b & \cdots & b & d & c & \cdots & c \\
b & a & \cdots & b & c & d & \cdots & c \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
b & b & \cdots & a & c & c & \cdots & d \\
d & c & \cdots & c & a & b & \cdots & b \\
c & d & \cdots & c & b & a & \cdots & b \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
c & c & \cdots & d & b & b & \cdots & a
\end{pmatrix}
\end{array}
\begin{array}{l}
\left.\vphantom{\begin{matrix}a\\b\\\vdots\\b\end{matrix}}\right\}12\text{ major}\\
\left.\vphantom{\begin{matrix}a\\b\\\vdots\\b\end{matrix}}\right\}12\text{ minor}
\end{array}$$

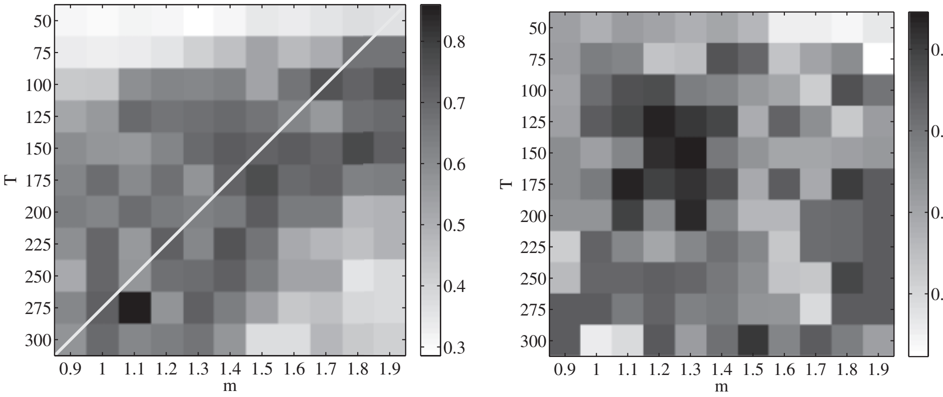
where $a = d = 0.996 \cdot (1 - 10^{-20})$ and $b = c = 0.004 \cdot 10^{-20}$, values that are seen to produce the best results in Chai and Vercoe (2005). The formulation in that study incorporates two transition matrices, one for key prediction (12×12) and one for mode prediction (2×2). Since we consider these transitions to be independent, we have merged these two matrices in a single 24×24 matrix that includes the product of the respective probabilities. The elements on the diagonal 12×12 blocks of the matrix correspond to the probabilities of modulations between keys of the same mode, while the off-diagonal blocks correspond to modulations from major to minor or vice versa. Each emission (frame) is associated with a key using its cosine distance with a binary key template. A binary key template is a vector with twelve elements that incorporates unit values for the tones that are active in the respective key and zeros for the inactive tones. The cosine distance is measured as $d_{cos} =< a, b > /(||a||\ ||b||)$, where $< \cdot, \cdot >$ denotes the inner product. Cosine distance returns values in the range $[0, 1]$.

Table 6 compares our proposed methodology with the the other methodologies studied in this article, in terms of segmentation accuracy on the R data set of real-world pieces. The clustering strategy provides the best accuracy with a considerably higher precision than the other techniques. The template-based approach yields the absolute recall score, but with an extremely low precision. This fact indicates that the technique provides a profusion of segmentation points, most of which are false. The recall values are similar for the HMM techniques and ours. Furthermore, to evaluate whether the differences between the f-measure results of the proposed and the other techniques are statistically significant, we apply a two-sided Wilcoxon signed-rank test (Wilcoxon 1945). The null hypothesis in the test is that the f-measures compared are independent samples from identical continuous distributions with equal medians. The null hypothesis is rejected at the 15% significance level for the Chai/Vercoe HMM approach and at the 1% level for the template-based approach, while it was not rejected for the simple HMM approach.

Key Segmentation versus Key Label Accuracy

Our method has the goal of providing an accurate segmentation of a piece at locations where key changes occur. The correct key labeling of the

Figure 9. Graphical representation of (a) f_m and (b) l for the R data set and for all tested T and m values, when clustering is performed on chroma space requiring four clusters. The diagonal in (a) shows the combinations of T and m for which better results are provided.



(a) *f*–measure

(b) Label accuracy

segmented parts is another major area of research. Table 7a shows two results obtained by clustering in the chroma space requiring four clusters on the *R* data set, and the labeling accuracy achieved by the other techniques examined. For the clustering setup, different combinations of *T* and *m* values have been examined, and the resulting segments have been labeled by matching the templates proposed by Temperley (2006) with linear correlation. The labeling accuracy of a piece is measured as the fraction of frames that are labeled with the correct key divided by the total number of frames. This value is denoted by *l*. The best f_m and *l* were produced for different pairs of *T* and *m* values. The best labeling results yielded by the clustering strategy slightly outperform the ones performed by other techniques and are very similar to the ones produced by the simple HMM approach. The fact that different *T* and *m* values give best f_m and *l* results seems counterintuitive, because better segmentation is expected to lead to a better labeling accuracy. In some cases, however, it seems that labeling accuracy (*l*) is relatively independent of segmentation accuracy (f_m). This is evident in the graphical representations in Figures 9a and 9b, where higher f_m is depicted for *T* and *m* combinations different from the ones that provide higher *l* values.

The last comment is further amplified by the findings in Table 7b, which presents the correlation between *f*-measure and label accuracy for different values of *T* and *m*. This reveals that label and segmentation accuracy are weakly correlated in some cases, a fact that brings up the question of which measure is more important. Although it seems reasonable that bad segmentation should result to bad labeling, the paradigm of the template-based approach provides a counterexample. As seen in Table 7a, the template-based approach gives poor segmentation with moderate labeling accuracy. This is an extreme example of over-segmentation with relatively accurate key matching. This is probably caused by segmenting small parts and mislabeling a small portion of them, an action that yields many false-positive segments. At the same time it results in relatively few falsely labeled parts, a fact that strongly affects segmentation accuracy but affects labeling accuracy to a smaller extent.

Conclusions

This article has introduced a novel method based on clustering for locating positions of key change in recordings of musical audio. This method relies only on geometric properties of musical segments within the chroma space. Its effectiveness is based on the temporal contiguity of successive segments. The temporal contiguity imposed by different cluster combinations within a time window *T* has been measured with the formulation of two curves, namely the accumulation curve and the gradient difference curves. Among all the GDCs, the values

Table 7. (a) Best f_m and l and (b) correlation of f_m and l

	T	*m*	f_m	*l*	*T*	*m*	*Correlation*
Proposed, Best f_m	275	1.1	**0.8601**	0.6457	275	1:0.1:1.9	0.17
Proposed, Best *l*	150	1.3	0.6981	**0.7229**	150	1:0.1:1.9	−0.18
Template based	—	—	0.1063	0.4916	50:25:275	1.1	0.50
Simple HMM	—	—	0.7000	0.7096	50:25:275	1.3	0.69
Chai HMM	—	—	**0.6281**	0.5222			
	(a) Best f_m and *l*				(b) f_m and *l* correlation		

(a) Shows f_m and l among all the examined approaches. (b) Displays correlation of f_m and l for various values of T and m. The notation $\alpha : x : \beta$ represents the set of numbers from α to β with an increment step of x.

that exceeded a predefined threshold *m* provided indications about a key change. Well chosen *T* and *m* values allow the accurate detection of arbitrary numbers of key changes.

For assessing the performance of our method we have not only used real-world compositions, but we also constructed artificial data sets consisting of pieces that include a predefined number of key changes. The advantage offered by artificial data sets is the possibility of constructing arbitrarily large, automatically annotated data sets. Artificially composed pieces in these data sets have predefined structures, and thus constitute test cases that potentially cover a wide range of possible key-change scenarios. The proposed key-segmentation technique has not only been tested on artificial data sets—a procedure that highlighted its strengths and weaknesses in certain musical circumstances—but has also been tested on real-world pieces with promising results. More specifically, with the real-world pieces the accomplished segmentation accuracy (0.8601) is better than the one demonstrated by HMMs (around 0.70). Results have also been calculated when clustering is performed on spaces of reduced dimensions with PCA and NMF for all data sets. Here, no overall improvement could be found.

It is evident that a greater number of real-world pieces is needed to reach stronger conclusions. A large annotated data set, including pieces from different genres, would aid the MIR community in developing a more solid basis for the evaluation of key-change detection algorithms. The artificial pieces that we used in this work should not be considered a substitute for complete compositions. Rather, they are used as an experimental procedure that helps to reveal the behavior of the model under certain tonal conditions. The formulation of proper settings for key-change scenarios with artificial pieces would most likely lead to useful observations about the functionality of a method, even if the results are not exactly the same with real-world pieces.

Among the future steps for improving the presented work is the fine-tuning of estimates for all the parameters incorporated in our methodology (i.e., *T*, *m*, and number of clusters, or number of dimensions in the case of PCA and NMF) for different data sets. The results presented here indicate that the combination of a large time window ($T \sim 275$ frames, 27.5 sec) and a low threshold value ($m \sim 1.1$) produces optimal results for the examined data set of real pieces. The use of different clustering algorithms could also improve the overall performance of the initial clustering stage. Moreover, a clustering algorithm that automatically adjusts the number of clusters could provide better results, since the number of expected clusters seems to have an impact on segmentation accuracy, as shown in Table 2. Furthermore, different algorithms for measuring the temporal contiguity of clusters can be proposed.

Acknowledgments

We would like to thank the anonymous reviewers and the editor for making constructive comments and providing valuable thoughts that improved the content of this article.

References

Chai, W. 2005. "Automated Analysis of Musical Structure." Ph.D. thesis, Massachusetts Institute of Technology, School of Architecture and Planning, Program in Media Arts and Sciences.

Chai, W., and B. Vercoe. 2005. "Detection of Key Change in Classical Piano Music." In *Proceedings of the Sixth International Conference on Music Information Retrieval (ISMIR 2005)*, pp. 468–473.

Chew, E. 2002. "The Spiral Array: An Algorithm for Determining Key Boundaries." In *Proceedings of the Second International Conference on Music and Artificial Intelligence (ICMAI 2002)*, pp. 18–31.

Chuan, C.-H., and E. Chew. 2007. "Audio Key Finding: Considerations in System Design and Case Studies on Chopin's 24 Preludes." *EURASIP Journal of Applied Signal Processing* 2007:156–170.

Ding, C., and X. He. 2004. "K-Means Clustering via Principal Component Analysis." In *Proceedings of the 21st International Conference on Machine Learning*, ICML 2004, pp. 29–38.

Harte, C., M. Sandler, and M. Gasser. 2006. "Detecting Harmonic Change in Musical Audio." In *Proceedings of the First ACM Workshop on Audio and Music Computing Multimedia*, AMCMM 2006, pp. 21–26.

Hartigan, J. A., and M. A. Wong. 1979. "A K-Means Clustering Algorithm." *Applied Statistics* 28(1):100–108.

Izmirli, Ö. 2006. "Audio Key Finding Using Low-Dimensional Spaces." In R. Dannenberg, K. Lemstrom, and A. Tindale, (eds.) *Proceedings of the Seventh International Conference on Music Information Retrieval (ISMIR 2006)*. pp. 127–132.

Izmirli, Ö. 2007. "Localized Key Finding from Audio Using Nonnegative Matrix Factorization for Segmentation." In *Proceedings of the Eighth International Conference on Music Information Retrieval (ISMIR 2007)*, pp. 195–200.

Jain, A. K., M. N. Murty, and P. J. Flynn. 1999. "Data Clustering: A Review." *ACM Computing Surveys* 31:264–323.

Krumhansl, C. L. 1990. *Cognitive Foundations of Musical Pitch*. New York: Oxford University Press.

Lazar, C., and A. Doncescu. 2009. "Non Negative Matrix Factorization Clustering Capabilities: Application on Multivariate Image Segmentation." In *International Conference on Complex, Intelligent and Software Intensive Systems, 2009*, pp. 924–929.

Lee, K., and M. Slaney. 2007. "A Unified System for Chord Transcription and Key Extraction Using Hidden Markov Models." In *Proceedings of the Eighth International Conference on Music Information Retrieval (ISMIR 2007)*. pp. 245–250.

Müller, M. 2010. "Chroma Toolbox: Pitch, Chroma, CENS, CRP." Available online at http://www.mpi-inf.mpg.de/resources/MIR/chromatoolbox/. Accessed November 2012.

Müller, M., and S. Ewert. 2011. "Chroma Toolbox: MATLAB Implementations for Extracting Variants of Chroma-based Audio Features." In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, pp. 215–220.

Müller, M., F. Kurth, and M. Clausen. 2005. "Audio Matching via Chroma-Based Statistical Features." In *Proceedings of the Sixth International Conference on Music Information Retrieval (ISMIR 2005)*, pp. 288–295.

Noland, K., and M. B. Sandler. 2006. "Key Estimation Using a Hidden Markov Model." In *Proceedings of the Seventh International Conference on Music Information Retrieval (ISMIR 2006)*, pp. 121–126.

Papadopoulos, H., and G. Peeters. 2009. "Local Key Estimation Based on Harmonic and Metric Structures." In *Proceedings of the Twelfth International Conference on Digital Audio Effects (DAFx-09)*, pp. 408–415.

Papadopoulos, H., and G. Peeters. 2012. "Local Key Estimation From an Audio Signal Relying on Harmonic and Metrical Structures." *IEEE Transactions on Audio, Speech, and Language Processing* 20(4):1297–1312.

Rocher, T., et al. 2010. "Concurrent Estimation of Chords and Keys from Audio." In *Proceedings of the 11th International Conference on Music Information Retrieval (ISMIR 2010)*, pp. 141–146.

Stöhlbrand, R., E. Helling, and J. Wallaart. 2011. "Piano Society." Available online at http://pianosociety.com/cms/index.php. Accessed November 2012.

Temperley, D. 2004. *The Cognition of Basic Musical Structures*. Cambridge, Massachusetts: MIT Press.

Temperley, D. 2006. *Music and Probability*. Cambridge, Massachusetts: MIT Press, annotated edition.

Wilcoxon, F. 1945. "Individual Comparisons by Ranking Methods." *Biometrics Bulletin* 1(6):80–83.