Compression-Based Clustering of Chromagram Data: New Method and Representations

Teppo E. Ahonen

Department of Computer Science
University of Helsinki
teahonen@cs.helsinki.fi

Abstract. We approach the problem of measuring similarity between chromagrams and present two new quantized representations for the task. The first representation is a sequence of optimal transposition index (OTI) values between the global chroma vector and each frame of the chromagram, whereas the second representation uses in similar fashion the global chroma of the query and frames of the target chromagram, thus emphasizing the mutual information of the chromagrams in the representation. The similarity between quantized representations is measured using normalized compression distance (NCD) as the similarity metric, and we experiment with a variant of k-medians algorithm, where the commonly used Euclidean distance has been replaced with NCD, to cluster the chromagrams. The representations and clustering method are evaluated by experimenting how well different cover versions of a composition can be clustered, and based on the experiments, we analyze various parameter settings for the representations. The results are promising and provide possible directions for future work.

Keywords: chromagram, normalized compression distance, clustering

1 Introduction

The chromagram, extracted from the audio data, is a sequence of 12-dimensional vectors that describe the relative energy of the 12 pitch classes of the western tonal scale. The chromagram is robust towards changes in features such as instrumentation and articulation, making it a very suitable feature for various tonality-based similarity measuring tasks. Because of this, chromagram is one of the most commonly used audio features in music information retrieval (MIR); different applications in tasks such as audio classification (e.g. [1]), audio fingerprinting (e.g. [2]), and chord sequence estimation (e.g. [3]) are based on information contained in the chromagram.

Measuring similarity between chromagram representations is essential in various retrieval and classification tasks. Different methods for chromagram similarity measurement have been presented, mostly in the task of cover song identification, where the goal is to determine whether two pieces of music are different renditions of the same composition. Far from trivial, but highly applicable when

9th International Symposium on Computer Music Modelling and Retrieval (CMMR 2012)
19-22 June 2012, Queen Mary University of London

successful, the task of cover song identification has gained a fair amount of interest from the researchers in the MIR community, yielding a plethora of different representations and similarity metrics. See [4] for an overview of several state-of-the-art methods.

In recent years, a similarity metric called normalized compression distance (NCD) [5] has been succesfully used for parameter-free similarity measuring in various tasks and domains. We apply NCD here, and in order to use the compression-based similarity metric for chromagram data, the continuous chromagram sequences need to be quantized. Several methods of producing a quantized representation from a chromagram exist. The method we use for discretization has been applied in [6], but unlike their work, we are not interested in binary similarity, but instead use the method to produce a sequence of 12 characters that represents the changes in the chromagram during the piece of music.

In this paper we apply the previously discussed discretization method and compression-based similarity metric for the task of clustering chromagram data. In the clustering phase we experiment with a slightly modified variation of the k-medians algorithm, using NCD as the distance metric. We evaluate the performance with real world audio data of cover versions, and examine the effect of smoothing the data with median filtering. The discretization is described in Section 2, and the clustering method is presented in Section 3. The experiments and their results are presented in Section 4, and Section 5 concludes the paper and discusses possible areas of future work.

2 Chroma Contour Representations

In [6], a method for producing a binary similarity matrix between two chromagrams was presented. The method uses optimal transposition index (OTI). OTI calculates the most likely semitone transposition between two chromagrams by first calculating the global chromagrams (i.e. the chromagram frames summed and normalized) and then taking the dot products between one global chromagram and all 12 transpositions of the other. The transposition with the highest dot product value is then used as the most likely semitone transposition between the pieces. Formally, for global chroma vectors G_a and G_b, the OTI is

$$OTI(G_a, G_b) = \underset{0 \le i \le M}{\arg\max} \{G_a \cdot circshift(G_b, i-1)\}, \tag{1}$$

where M is the maximum of possible transpositions; in our work, this is 12.

In [6], the binary similarity matrix between two pieces of music is obtained by calculating the OTI values between each pair of frames of the two chromagrams, and setting the binary value to the similarity matrix according to the obtained dot product value. We apply the idea, but instead of binary values, we consider all 12 possible transpositions between chroma vectors, and instead of comparing two sequences and constructing a similarity matrix, we transform a continuous chromagram into character sequence representation. We start by calculating the OTI values between the global chroma vector of the piece and each chroma frame of the piece. Then, the frames are labeled according to the 12 possible OTI values, thus producing a sequence of character labels from an alphabet of size 12. For the lack of a better term, we call this *chroma contour*, as it describes the relative changes between subsequent chromagram vectors. Formally, for a chromagram g_a of length n, and its global chroma vector G_a, the resulting chroma contour sequence (ccs) is

$$ccs(i) = OTI(g_a(i), G_a), \tag{2}$$

where $1 \leq i \leq n$. Finally, each of the 12 possible OTI values in *ccs* are assigned a related symbol.

The representation has the advantage of being completely key-independent, as the sequences produced by OTI are similar in all transpositions. An illustration of a sequence produced by this method is depicted in Figure 1. The similarity is then measured between two such chroma contour sequences. To calculate the chromagrams, we use a window length of 0.1858 seconds with a hop factor of 0.875.

However, this representation only provides information on how the chromatic features change in a single piece of music. This information is clearly useful, but we also wish to consider the relation between two pieces of music, as this is likely beneficial information considering the similarity measuring. The previously presented method can be straightforwardly applied for two pieces, simply by using the global chroma of one piece (the query) and the chroma vectors of the other (the target), producing a similar chroma contour sequence. Again, for the lack of a better term, we call this *cross-chroma contour*. Formally, for a target chromagram g_a of length n and a global query chroma vector G_b, the cross-chroma contour sequence (cccs) is

$$cccs(i) = OTI(g_a(i), G_b), \tag{3}$$

where $1 \leq i \leq n$. Again, finally each of the 12 possible OTI values in *cccs* are assigned a related symbol.

The reasoning for this representation is the idea that if two chromagrams contain similar features, their contours should be highly similar regardless of the global features, whereas two unrelated chromagrams should provide a target cross-chroma contour that does not bear resemblance to the query chroma contour. An illustration of cross-chroma contours for two pieces of music using the global chroma data of the piece of Figure 1 is depicted in Figure 2.

The cross-chroma contour representation is not key invariant, as for example using a semitone transposed global chroma would produce a highly unsimilar sequence in comparison to a sequence produced with the untransposed version. When comparing chromagrams, this needs to be addressed, as the chromagrams extracted from pieces in different keys would be deemed unsimilar regardless of the similarities they share. To overcome this, we first calculate the OTI between the two global chromagrams, then transpose the query according to the OTI value, and then produce the cross-chroma contour sequence. Formally, using notation from Equation 3, this is

$$cccs(i) = OTI(g(i), G_b^n), \tag{4}$$

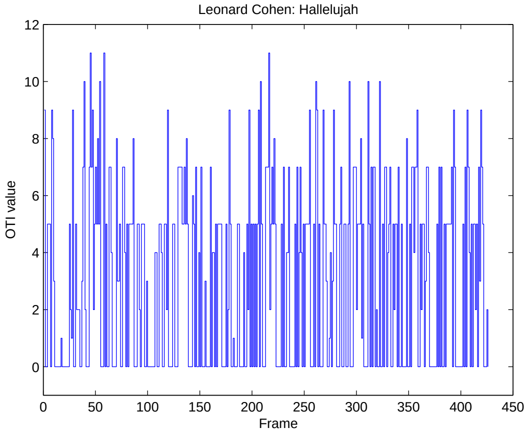

Fig. 1. A chroma contour illustration. For the sake of readability, the sequence depicted here is obtained from a chromagram that was calculated using a quadruple window length.

where G^n is the global chromagram transposed n semitones and $n = OTI(G_a, G_b)$.

3 Clustering Methodology

The clustering method presented in this paper uses normalized compression distance (NCD) [5] as the similarity metric. Using NCD seems appealing for several reasons.

First, NCD is parameter-free, requiring no background information of the data it is applied to. But this, naturally, makes the selection of the data representation crucial for our task: NCD captures the dominant feature similarity between the objects [5] and the representation should therefore express this feature. Second, NCD can be shown to approximate a universal similarity metric, depending on how well the compression algorithm approximates Kolmogorov complexity, the theoretical measure of computational resources needed to produce the object. In addition, NCD has been used succefully for various tasks in music information retrieval (e.g. [7–9]).

To understand what makes NCD a suitable choice as a similarity metric, its background in information theory needs to be explained. Denote $K(x)$ as the Kolmogorov complexity of object x as the length of the smallest program that produces x and $K(x|y)$ as the conditional Kolmogorov complexity, that is, the length of the smallest program that produces x given y as an input. Now, a universal similarity metric called normalized information distance (NID) can be denoted [10]

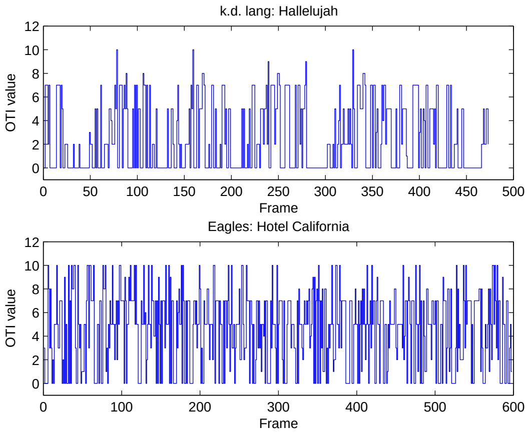

Fig. 2. Two cross-chroma contour illustration, calculated using the global chroma of the piece of music of Figure 1. The upper cross-chroma contour seems to bear more resemblance to the chroma contour of Figure 1. The sequences depicted here are obtained from chromagrams that were calculated using a quadruple window length.

$$NID(x,y) = \frac{\max\{K(x|y), K(y|x)\}}{\max\{K(x), K(y)\}}. \tag{5}$$

NID can be shown to be universal [10], but the incomputability of Kolmogorov complexity makes NID also incomputable. However, the Kolmogorov complexity can be approximated using data compression. This leads to an approximation of NID that is NCD. For two objects x and y, NCD is denoted [5]

$$NCD(x,y) = \frac{C(xy) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}} \tag{6}$$

where $C(x)$ is the length of x when compressed using a fixed lossless compression algorithm C, and xy is the concatenation of x and y. For the experiments conducted here, we use the bzip2 algorithm for data compression.

3.1 k-medians Clustering

The k-medians clustering method is a variant of the well-known k-means clustering method that uses, as the name implies, median instead of mean when selecting the new cluster centroids. Commonly, Euclidian distance is used as the similarity metric in k-medians clustering, but when clustering symbolized data, the Euclidian distance seems unusable. Here, we apply NCD as the distance measure between the strings and based on the pairwise distances between either chroma or cross-chroma contour representations, conduct the standard k-medians clustering. The strings in the clusters are sorted according to a lengthincreasing, lexicographical order, allowing us to select the median from the list of the sorted strings.

4 Evaluation

To evaluate the presented methods and representations, we collected a dataset of cover versions from our personal collections of music. The total dataset consists of 12 ten-song sets of original performances and their cover versions, thus totaling 120 pieces of music. In our experiments, we try to cluster the chromagrams into groups of renditions of the same composition. In practice, this is a cover song identification task expanded with the clustering phase. In order to obtain successful clustering, both the chromagram similarity measuring and the k-medians variant should perform adequately.

To measure the clustering performance we use purity. The purity of a single cluster is calculated as the ratio between the size of the most frequent class (in our case, the cover song set) of the cluster and the size of the whole cluster. The purity for the clustering is then calculated as the average of the single cluster purities. High purity value expresses that the cover versions have been successfully clustered together.

We ran the evaluations for three different-sized datasets. In the *set30* we used three ten-song sets, and in *set60* we doubled the size to six ten-song sets. The *set120* is the complete dataset. For each dataset, the k-medians clustering was run with k of 3, 6, and 12, respectively. The results for the evaluations are presented in Table 1. As the k-medians algorithm selects the initial cluster centroids randomly, we ran the evaluations five times and averaged the results. As a vague baseline comparison, results for a clustering with random distance values is included.

The perfomance of the cross-chroma contour representation is slightly superior to the chroma contour representation. This supports the idea that mutual information between two pieces should be taken into account when producing the representation for a compression-based method. Also, there seems to be robustness in the method, as the performance does not drop significantly as the size of the dataset increases.

4.1 Pre- and Post-Discretization Filtering

The chromagram data extracted from the audio is likely noisy. Often, algorithms that measure chromagram similarity filter the data before applying the similarity measuring. One of the most straightforward ways to smoothen the data is to use median filtering. We experimented with several orders of the median filtering, but here present only the best results, which were obtained with a median filtering of order five.

Also, the sequences produced by our method do oscillate between different OTI values, making the sequences noisy. To make the sequences smoother and thus more compressable, we experimented with median filtering of various orders for the sequences, but as with the chromagram filtering, present only the results for the best choice of the filtering order, which was also five for the sequence filtering.

Based on the results in Table 1, it seems that both the noise on the chroma data and in the produced character sequences are beneficial for the compression-based similarity metric, and combining both smoothings provides the worst results that are only slightly above, or even below, the random baseline. This could possibly be an outcome of over-simplifying the sequences and losing all distinguishing information. However, on occasional runs the results with filtered data were better than with the unfiltered versions.

Table 1. Purity values of the clustering experiments, averaged over five runs.

	set30	set60	set120
Chroma contour	0.367	0.283	0.217
Cross-chroma contour	**0.374**	**0.317**	**0.257**
Chroma contour and chroma filtering	0.310	0.231	0.162
Cross-chroma contour and chroma filtering	0.344	0.312	0.228
Chroma contour and sequence filtering	0.331	0.258	0.189
Cross-chroma contour and sequence filtering	0.337	0.294	0.212
Chroma contour and both filtering methods	0.133	0.104	0.081
Cross-chroma contour and both filtering methods	0.192	0.162	0.132
Random baseline	0.233	0.117	0.067

5 Conclusions and Future Work

We have presented a method for clustering chromagram data based on the contour of the chromagrams. Our method produces a string representation from the chroma data based on the optimal transposition index values between the global chroma vector and the single chroma vectors of the piece. We also presented a variation where the pairwise mutual information can be utilized by using the global chroma of the query when producing a representation from the target chroma vectors.

Also, the effect of smoothing both the chromagram and the sequence data was experimented, and according to our results, both smoothing methods affect the results negatively. We clustered the chromagrams using a variant of k-medians clustering with normalized compression distance as the similarity metric and used purity to measure the performance of the clustering.

At its best, the method does provide a reasonable level of performance, but it is also clear that several issues still demand consideration. A more fine-grained chroma contour could possibly provide higher cluster purity, as currently several false positives occur and pieces of music end up in wrong clusters. This is likely due to the possibly over-simplified representation; although the representations are composed using a rather small alphabet, making them thus presumably suitable for the compression algorithm, the trade-off comes as a loss of distinguishing power. A more fine-grained chroma contour could be obtained by either using a chroma representation of 24 or 36 bins, or producing the character sequences using a more complex method instead of the dot product. We are currently working on this, and in addition, experimenting with our k-medians variation, in order to see if there are other features leading to biased results. We are also aware that this paper has not provided comparison with other represenations, similarity metrics, or clustering methods. A thorough comparison with other methodologies is needed in the future.

Acknowledgments. The work presented here was supported by Helsinki Doctoral Programme in Computer Science – Advanced Computing and Intelligent Systems (Hecse). The author wishes to express gratitude to the anonymous reviewers for their insightful comments on the first version of this paper.

References

1. Casey, M., Slaney, M.: The Importance of Sequences in Musical Similarity. In: Proc. ICASSP'06, pp. 5–8. (2006)
2. Bartsch, M., Wakefield, G.: To Catch a Chorus: Using Chroma-based Representations for Audio Thumbnailing. In: Proc. WASPAA'01, pp. 15–18. (2001)
3. Papadopoulos, H., Peeters, G.: Large-Scale Study of Chord Estimation Algorithms Based on Chroma Representation and HMM In: Proc. CBMI'07, pp. 53–60. (2007)
4. Serrà J., Gómez, E., Herrara P.: Audio Cover Song Identification and Similarity: Background, Approaches, Evaluation, and Beyond. In Advances in Music Information Retrieval, pp. 307–332. Springer Verlab (2010)
5. Cilibrasi, R., Vitányi, P.M.B.: Clustering by Compression. IEEE Trans. Information Theory. 51, 1523–1545 (2005)
6. Serrá, J., Gómez E., Herrera P., Serra X.: Chroma Binary Similarity and Local Alignment Applied to Cover Song Identification. IEEE Trans. Audio, Speech and Language Processing. 16, 1138–1151 (2008)
7. Cilibrasi, R., Vitányi, P., De Wolf, R.: Algorithmic Clustering of Music Based on String Compression. Comput. Music J. 28, 49–67 (2004)
8. Li, M., Sleep, R.: Genre Classification Via an LZ78-based String Kernel. In: Proc. ISMIR'05, pp. 252–259. (2005)
9. Bello, J.P.: Grouping Recorded Music by Structural Similarity. In: Proc. ISMIR'09, pp. 531–536. (2009)
10. Li, M., Chen, X., Li, X, Ma, B., Vitányi, P.M.B.: The Similarity Metric. IEEE Trans. Information Theory. 50. 3250–3264 (2004)