EURASIP Journal on Advances in Signal Processing
a SpringerOpen Journal

RESEARCH Open Access

Music recommendation according to human motion based on kernel CCA-based relationship

Hiroyuki Ohkushi*, Takahiro Ogawa and Miki Haseyama

Abstract

In this article, a method for recommendation of music pieces according to human motions based on their kernel canonical correlation analysis (CCA)-based relationship is proposed. In order to perform the recommendation between different types of multimedia data, i.e., recommendation of music pieces from human motions, the proposed method tries to estimate their relationship. Specifically, the correlation based on kernel CCA is calculated as the relationship in our method. Since human motions and music pieces have various time lengths, it is necessary to calculate the correlation between time series having different lengths. Therefore, new kernel functions for human motions and music pieces, which can provide similarities between data that have different time lengths, are introduced into the calculation of the kernel CCA-based correlation. This approach effectively provides a solution to the conventional problem of not being able to calculate the correlation from multimedia data that have various time lengths. Therefore, the proposed method can perform accurate recommendation of best matched music pieces according to a target human motion from the obtained correlation. Experimental results are shown to verify the performance of the proposed method.

Keywords: content-based multimedia recommendation, kernel canonical correlation analysis, longest common subsequence, *p*-spectrum

1 Introduction

With the popularization of online digital media stores, users can obtain various kinds of multimedia data. Therefore, technologies for retrieving and recommending desired contents are necessary to satisfy the various demands of users. A number of methods for content-based multimedia retrieval and recommendation[a] have been proposed. Image recommendation [1-3], music recommendation [4-6], and video recommendation [7,8] have been intensively studied in several fields. It should be noted that most of these previous works had the constraint of query examples and returned results to be recommended being of the same type. However, due to diversification of users' demands, there is a need for a new type of multimedia recommendation in which the media types of query examples and the returned results can be different. Thus, several recommendation methods [9-12] for realizing these recommendation schemes have been proposed. Generally, they are called cross-media recommendation. In the conventional methods of the cross-media recommendation, the query examples and recommended results need not to be of the same media types. For example, users can search music pieces by submitting either an image example or a music example.

Among the conventional methods of cross-media recommendation, Li et al. proposed a method for recommendation between images and music pieces by comparing their features directly using a dynamic time warping algorithm [9]. Furthermore, Zhang et al. proposed a method for cross-media recommendation between multimedia documents based on a semantic graph [11,12]. A multimedia document (MMD) is a collection of co-existing heterogeneous multimedia objects that have the same semantics. For example, an educational web page with instructive text, images and audio is an MMD. By these conventional methods, users can search for their desired contents more flexibly and effectively.

It should be noted that the above-conventional methods concentrate on recommendation between different types multimedia data. Thus, in this scheme, users are

* Correspondence: ohkushi@lmd.ist.hokudai.ac.jp
Graduate School of Information Science and Technology, Hokkaido University, Sapporo, Japan

forced to provide query multimedia data, although they do not have a limitation of media types. This means that users must make some decisions to provide queries, and this causes difficulties for reflecting their demands. If recommendation of some multimedia data from features directly obtained from users is realized, one feasible solution can be provided to overcome the limitation. Specifically, we show the following two example applications: (i) background music selection from humans' dance motions for non-edited video contents[b] and (ii) presentation of music information from features of target music pieces or dance motions. In the first example, using the relationship obtained between dance motions and music pieces in a database, we can obtain/find matched music pieces from human motions in video contents, and vice versa. This should be useful for creating a new dance program with background music and a music promotional video with dance motions. For example, given human motions of a classic ballet program, we can assign music pieces matched to the target human motions, and this example will be shown in the verification in the experiment section. Next, in the second example, this can present to users information of music that they are listening to, i.e., song title, composer, etc. Users can use sounds of music pieces or the user's own dance motion associated with the music as the query for obtaining information on the music. As described above, the application can also use the relationship between human motions and music pieces, and it can be a more flexible information presentation system than the conventional ones. In this way, information directly obtained from users, i.e., users' motions can retain the potential to get various benefits. These schemes are cross-media recommendation schemes and they remove barriers between users and those multimedia contents.

In this article, we deal with recommendation of music pieces from features obtained from users. Among the features, human motions have high-level semantics, and their use is effective for realizing accurate recommendation. Therefore, we try to estimate suitable music pieces from human motions. This is because we consider that correlation extraction between human motions and music pieces becomes feasible using some specific video contents such as dance and music promotional videos. This benefit is also useful in performance verification. Then, we assume that the meaning of "suitable" is emotionally similar. Specifically, in our purpose, the recommendation of suitable music pieces according to human motions is that the recommended music pieces are emotionally similar to the query human motions.

In this article, we propose a new method for cross-media recommendation of music pieces according to human motions based on kernel canonical correlation analysis (CCA) [13]. We use video contents in which video sequences and audio signals contain human motions and music pieces, respectively, as training data for calculating their correlation. Then, using the obtained correlation, estimation of the best matched music piece from a target human motion becomes feasible. It should be noted that several methods of cross-media recommendation have previously been proposed. However, there have been no methods focused on handling data that have various time lengths, i.e., human motions and music pieces. Thus, we propose a cross-media recommendation method that can effectively use characteristics of time series, and we assume that this can be realized using kernel CCA and our defined kernel functions. From the above discussion, the main contribution of the proposed method is handling data that have various time lengths for cross-media recommendation.

In this approach, we have to consider the differences in time lengths. In the proposed method, new kernel functions of human motions and music pieces are introduced into the CCA-based correlation calculation. Specifically, we newly adopt two types of kernel functions, which can represent similarities by effectively using human motions or music pieces having various time lengths, for the kernel CCA-based correlation calculation. First, we define a longest common subsequence (LCSS) kernel for using data having different time lengths. Since the LCSS [14] is commonly used for motion comparison, the LCSS kernel should be suitable for our purpose. It should be noted that kernel functions must satisfy Mercer's theorem [15], but our newly defined kernel function does not necessarily satisfy this theorem. Therefore, we also adopt another type of kernel function, spectrum intersection kernel, that satisfies Mercer's theorem. This function introduces the p-spectrum [16] and is based on the histogram intersection kernel [17]. Since the histogram intersection kernel is known as a function that satisfies Mercer's theorem, the spectrum intersection kernel also satisfies this theorem.

Actually, there have been kernel functions that do not satisfy Mercer's theorem, and there have also been several proposed methods that use such kernel functions. The effectiveness of the above-described methods has also been verified. Thus, we should also verify the effectiveness of our defined kernel function, which does not satisfy Mercer's theorem, i.e., the LCSS kernel. In addition, we should also compare our two newly defined kernel functions experimentally. Therefore, in this article, we introduce two types of kernel functions. Using these two types of kernel functions, the proposed method can directly compare multimedia data that have various time lengths, and this is the main advantage of our method. Thus, the use of these kernel functions

effectively provides a solution to the problem of not being able to simply apply sequential data such as human motions and music pieces to cross-media recommendation. Consequently, effective modeling of the relationship using music and human motion data that have various time lengths is realized, and successful music recommendation can be expected.

This article is organized as follows. First, in Section 2, we briefly explain the kernel CCA used for calculating the correlation between human motions and music pieces. Next, in Section 3, we describe our two newly defined kernel functions. Kernel CCA-based music recommendation according to human motion is proposed in Section 4. Experimental results that verify the performance of the proposed method are shown in Section 5. Finally conclusions are given in Section 6.

2 Kernel canonical correlation analysis

In this section, we explain kernel CCA. First, two variables $\mathbf{x}$ and $\mathbf{y}$ are transformed into Hilbert space H_x and H_y via non-linear maps $\varphi_{\mathbf{x}}$ and $\varphi_{\mathbf{y}}$. From the mapped results $\varphi_{\mathbf{x}}(\mathbf{x}) \in H_{\mathbf{x}}$ and $\varphi_{\mathbf{y}}(\mathbf{y}) \in H_{\mathbf{y}}$,[c] the kernel CCA seeks to maximize the correlation

$$\rho = \frac{\mathbb{E}[uv]}{\sqrt{\mathbb{E}[u^2]\mathbb{E}[v^2]}} \tag{1}$$

between

$$u = \langle \mathbf{a}, \phi_{\mathbf{x}}(\mathbf{x}) \rangle \tag{2}$$

and

$$v = \langle \mathbf{b}, \phi_{\mathbf{y}}(\mathbf{y}) \rangle \tag{3}$$

over the projection directions $\mathbf{a}$ and $\mathbf{b}$. This means that kernel CCA finds the directions $\mathbf{a}$ and $\mathbf{b}$ that maximize the correlation $\mathbb{E}[uv]$ of corresponding projections subject to $\mathbb{E}[u^2] = 1$ and $\mathbb{E}[v^2] = 1$.

The optimal directions $\mathbf{a}$ and $\mathbf{b}$ can be found by solving the Lagrangian

$$\mathcal{L} = \mathbb{E}[uv] - \frac{\lambda_1}{2}(\mathbb{E}[u^2] - 1) - \frac{\lambda_2}{2}(\mathbb{E}[v^2] - 1) + \frac{\eta}{2}(||\mathbf{a}||^2 + ||\mathbf{b}||^2), \tag{4}$$

where η is a regularization parameter. The above-computation scheme is called regularized kernel CCA [13]. By taking the derivatives of Equation 4 with respect to $\mathbf{a}$ and $\mathbf{b}$, $\lambda_1 = \lambda_2 (= \lambda)$ is derived, and the directions $\mathbf{a}$ and $\mathbf{b}$ maximizing the correlation ρ $(= \lambda)$ can be calculated.

3 Kernel function construction

Construction of new kernel functions is described in this section. The proposed method constructs two types of kernel functions for human motions and music pieces, respectively. First, we introduce an LCSS kernel as a kernel function that does not satisfy Mercer's theorem. This function is based on the LCSS algorithm [18], which is commonly used for motion or temporal music signal comparison since the LCSS algorithm can compare two temporal signals even if they have different time lengths. Therefore, it seems that this kernel function is suitable for our recommendation scheme. On the other hand, we also introduce a spectrum intersection kernel that satisfies Mercer's theorem. This function is based on the p-spectrum [16], which is generally used for text comparison. The p-spectrum uses the continuity of words. This property is also useful for analyzing the structure of temporal sequential data, i.e., human motions. Thus, the spectrum intersection kernel is also suitable for our recommendation scheme.

For the following explanation, we prepare pairs of human motions and music pieces extracted from the same video contents and denote each pair as a segment. The segments are defined as short terms of video contents that have various time lengths. From the obtained segments, we extract human motion features and music features of the jth $(j = 1, 2, \ldots, N)$ segment as $\mathbf{V}_j = [\mathbf{v}_j(1), \mathbf{v}_j(2), \ldots, \mathbf{v}_j(N_{\mathbf{v}_j})]$ and $\mathbf{M}_j = [\mathbf{m}_j(1), \mathbf{m}_j(2), \ldots, \mathbf{m}_j(N_{\mathbf{M}_j})]$, where $N_{\mathbf{v}_j}$ and $N_{\mathbf{M}_j}$ are the numbers of components of $\mathbf{V}_j$ and $\mathbf{M}_j$, respectively, and N is the number of segments. In $\mathbf{V}_j$ and $\mathbf{M}_j$, $\mathbf{v}_j(l_v)$ $(l_v = 1, 2, \ldots, N_{\mathbf{v}_j})$ and $\mathbf{m}_j(l_m)$ $(l_m = 1, 2, \ldots, N_{\mathbf{M}_j})$ correspond to optical flows [19] and chroma vectors [20], respectively. The optical flow is a simple and representative feature that represents motion characteristics between two successive frames in video sequences and is commonly used for motion comparison. Thus, we adopt the optical flow as temporal components of human motion features. Furthermore, the chroma vector represents tone distribution of music signals at each time. The chroma vector can represent the characteristics of a music signal robustly if it is extracted in a short time. In addition, due to the simplicity of the implementation, we adopted these features in our method. More details of these features are given in Appendices A.1 and A.2.

3.1 Kernel function for human motions

3.1.1 LCSS kernel

In order to define kernel functions for human motions having various time lengths, we firstly explain the LCSS kernel for human motions that uses an LCSS-based similarity in [14]. An LCSS is an algorithm that enables calculation of the longest common part and its length (LCSS length) between two sequences.

Figure 1 shows an example of a table produced by LCSS length of two sequences $X = \langle B, D, C, A, B \rangle$ and $Y = \langle A, B, C, B, A, B \rangle$. In this figure, the highlighted

Y \ X		B	D	C	A	B
	0	0	0	0	0	0
A	0	0	0	0	1	1
B	0	1	1	1	1	2
C	0	1	1	2	2	2
B	0	1	1	2	2	3
A	0	1	1	2	3	3
B	0	1	1	2	3	4

Figure 1 An example of a table based on LCSS length of the sequences $X = \langle B, D, C, A, B\rangle$ and $Y = \langle A, B, C, B, A, B\rangle$.

components represent the common components in two different sequences and LCSS length between X and Y becomes four.

Here, we show the definition of similarity between human motion features. For the following explanations, we denote two human motion features as $\mathbf{V}_a = [\mathbf{v}_a(1), \mathbf{v}_a(2), ..., \mathbf{v}_a(N_{\mathbf{v}_a})]$ and $\mathbf{V}b = [\mathbf{v}_b(1), \mathbf{v}_b(2), ..., \mathbf{v}_b(N_{\mathbf{v}_b})]$, where $\mathbf{v}_a(l_a)$ $(l_a = 1, 2, ..., N\mathbf{v}_a)$ and $\mathbf{v}_b(l_b)$ $(l_b = 1, 2, ..., N_{\mathbf{v}_b})$ are components of $\mathbf{V}_a$ and $\mathbf{V}_b$, respectively, and $N_{\mathbf{v}_a}$ and $N_{\mathbf{v}_b}$ are the numbers of components in $\mathbf{V}_a$ and $\mathbf{V}_b$, respectively. In addition, $\mathbf{v}_a(l_a)$ and $\mathbf{v}_b(l_b)$ correspond to optical flows extracted in each frame in each video sequence. Note that $N_{\mathbf{v}_a}$ and $N_{\mathbf{v}_b}$ depend on the time lengths of their segments; that is, they depend on the number of frames of their video sequences. The similarity between $\mathbf{V}_a$ and $\mathbf{V}_b$ is defined as follows:

$$Sim_{\mathbf{v}}(\mathbf{V}_a, \mathbf{V}_b) = \frac{LCSS(\mathbf{V}_a, \mathbf{V}_b)}{\min(N_{\mathbf{v}_a}, N_{\mathbf{v}_b})}, \tag{5}$$

where $LCSS(\mathbf{V}_a,\mathbf{V}_b)$ is the LCSS length of $\mathbf{V}_a$ and $\mathbf{V}_b$, and it is recursively defined as

$$LCSS(\mathbf{V}_a, \mathbf{V}_b) = R_{\mathbf{V}_a\mathbf{V}_b}(l_a, l_b)|_{l_a=N_{\mathbf{v}_a}, l_b=N_{\mathbf{v}_b}}, \tag{6}$$

$$R_{\mathbf{V}_a\mathbf{V}_b}(l_a, l_b) = \begin{cases} 0 & \text{if } N_{\mathbf{V}_a} = 0 \text{ or } N_{\mathbf{V}_b} = 0, \\ 1 + R_{\mathbf{V}_a\mathbf{V}_b}(l_a - 1, l_b - 1) & \text{if } c(\mathbf{v}_a(l_a)) = c(\mathbf{v}_b(l_b)), \\ \max\{R_{\mathbf{V}_a\mathbf{V}_b}(l_a - 1, l_b), R_{\mathbf{V}_a\mathbf{V}_b}(l_a, l_b - 1)\} & \text{otherwise,} \end{cases} \tag{7}$$

where $c(\cdot)$ is a cluster number of optical flow. In the proposed method, we apply a k-means algorithm [21] for all optical flows obtained from all segments, and the obtained cluster numbers assigned to the belonging optical flows $c(\cdot)$ are used for easy comparison of two different optical flows. For this purpose, some kinds of quantization or labeling of the temporal variation of the time series seem to be available. In the proposed method, we adopt k-means clustering for its simplicity.

We then define this similarity measure as the LCSS kernel for human motions $\kappa_{\mathbf{v}}^{LCSS}(\cdot,\cdot)$ as follows:

$$\kappa_{\mathbf{V}}^{LCSS}(\mathbf{V}_a, \mathbf{V}_b) = Sim_{\mathbf{V}}(\mathbf{V}_a, \mathbf{V}_b). \tag{8}$$

The above-kernel function can be used for time series having various time lengths. Not only our LCSS kernel but also other kernel functions are known as non-positive semi-definite. Therefore, these do not strictly satisfy Mercer's theorem [15]. Fortunately, kernel functions that do not satisfy Mercer's theorem have been verified to be effective for classification of sequential data using a kernel function in [18].

Furthermore, several methods using kernel functions that do not satisfy the theorem have been proposed in [22,23]. Also, a sigmoid kernel has been commonly used and is well known as a kernel function which does not satisfy Mercer's theorem. We therefore briefly discuss implications and problems that might emerge using a kernel function that does not satisfy the theorem. In order to satisfy Mercer's theorem, a gram matrix whose elements correspond to values of a kernel function is required to be a positive semi-definite and symmetric matrix. Not only our defined kernel function but also other kernel functions that do not satisfy Mercer's theorem have symmetric and non-positive semi-definite gram matrices. Thus, for the solution based on such kernel functions, several methods have modified eigenvalues of the gram matrices to be greater than or equal to zero. It should be noted that we used our defined kernel functions directly in the proposed method.

3.1.2 Spectrum intersection kernel

Next, we explain the spectrum intersection kernel for human motions. In order to define the spectrum intersection kernel for human motions, we firstly calculate p-spectrum-based features. The p-spectrum [16] is the set of all p-length (contiguous) subsequences that it contains. The p-spectrum-based features on string $\mathcal{X}$ are indexed by all possible subsequences $\mathcal{X}_s$ of length p and defined as follows:

$$\mathbf{r}_p(\mathcal{X}) = (r_{\mathcal{X}_s}(\mathcal{X}))_{\mathcal{X}_s \in \mathcal{A}^p}, \tag{9}$$

where

$$r_{\mathcal{X}_s}(\mathcal{X}) = \text{number of times } \mathcal{X}_s \text{ occurs } in \ \mathcal{X}, \tag{10}$$

and $\mathcal{A}$ is the set of characters in strings. For human motion features, we cannot apply the p-spectrum directly since human motion features are defined as sequences of vectors. Therefore, we apply the p-spectrum to sequences of cluster numbers of optical flows as that done for the LCSS kernel. We use the histogram

intersection kernel [17] for constructing the spectrum intersection kernel. The histogram intersection kernel $\kappa^{HI}(\cdot, \cdot)$ is a useful kernel function for classification of histogram-shaped features and is defined as follows:

$$\kappa^{HI}(\mathbf{h}_a, \mathbf{h}_b) = \sum_{i_h=1}^{N^h} \min\{h_a(i_h), h_b(i_h)\}, \tag{11}$$

where $\mathbf{h}_a$ and $\mathbf{h}_b$ are histogram-shaped features, $h_a(i_h)$ and $h_b(i_h)$ are the i_hth element (bin) values of $\mathbf{h}_a$ and $\mathbf{h}_b$, respectively, and N^h is the numbers of bins of histogram-shaped features. Furthermore, $\sum_{i_h=1}^{N^h} h_a(i_h) = 1$ and $\sum_{i_h=1}^{N^h} h_b(i_h) = 1$ are required to apply the histogram intersection kernel into $\mathbf{h}_a$ and $\mathbf{h}_b$. The p-spectrum-based features also have histogram shapes, and they can be applied to the histogram intersection kernel. Note that the sums of elements have to be normalized in the same way as that done for histogram-shaped features. After that, we define this kernel function as the spectrum intersection kernel for human motions $\kappa_{\mathbf{V}}^{SI}(\cdot, \cdot)$ shown as follows:

$$\kappa_{\mathbf{V}}^{SI}(\mathbf{V}_a, \mathbf{V}_b) = \kappa^{HI}(\mathbf{r}_p(\mathbf{V}_a), \mathbf{r}_p(\mathbf{V}_b)). \tag{12}$$

The above-kernel function can consider statistical characteristics of human motion features. Since the histogram intersection kernel is positive semi-definite [17], the spectrum intersection kernel can satisfy Mercer's theorem [15]. Note that the above-kernel function is equivalent to the spectrum kernel defined in [16] if we use the simple inner product of p-spectrum-based features instead of the histogram intersection in Equation 12.

3.2 Kernel function for music pieces

3.2.1 LCSS kernel

The kernel functions for music pieces are defined in the same way as those of human motions. First, we show the definition of the LCSS kernel for music pieces. For the following explanations, we denote two music features as $\mathbf{M}_a = [\mathbf{m}_a(1), \mathbf{m}_a(2), ..., \mathbf{m}_a(N_{\mathbf{M}_a})]$ and $\mathbf{M}_b = [\mathbf{m}_b(1), \mathbf{m}_b(2), ..., \mathbf{m}_b(N_{\mathbf{M}_b})]$, where $\mathbf{M}_a$ and $\mathbf{M}_b$ are chromagrams [24] and are extracted from segments, $\mathbf{m}_a(l_a)$ $(l_a = 1, 2, ..., N_{\mathbf{M}_a})$ and $\mathbf{m}_b(l_b)$ $(l_b = 1, 2, ..., N_{\mathbf{M}_b})$ are components of $\mathbf{M}_a$ and $\mathbf{M}_b$, and $N_{\mathbf{M}_a}$ and $N_{\mathbf{M}_b}$ are the numbers of components of $\mathbf{M}_a$ and $\mathbf{M}_b$, respectively. In addition, $\mathbf{m}_a(l_a)$ and $\mathbf{m}_b(l_b)$ are chroma vectors [20] that have 12 dimensions. Since $N_{\mathbf{M}_a}$ and $N_{\mathbf{M}_b}$ depend on the time lengths of their segments, the similarity between music features is also defined on the basis of the LCSS algorithm. Note that it is desirable that the similarity between an original music piece and its modulated version becomes high since they have similar melodies, base lines, or harmonics. Therefore, we define similarity considering the modulation of music. In the proposed method, we use temporal sequences of chroma vectors, i.e., chromagrams defined in [24], as music features. One of the advantages of the use of 12-dimensional chroma vectors in the chromagrams is that the transposition amount of modulation can be naturally represented only by the amount ζ by which its 12 elements are shifted (rotated). Therefore, the proposed method effectively uses the above characteristic for measuring similarities between chromagrams. For the following explanation, we define the modulated chromagram $\mathbf{M}_b^{\zeta} = [\mathbf{m}_b^{\zeta}(1), \mathbf{m}_b^{\zeta}(2), .., \mathbf{m}_b^{\zeta}(N_{\mathbf{M}_b})]$. Note that $\mathbf{m}_b^{\zeta}(l_b)$ $(l_b = 1, 2, ..., N_{\mathbf{M}_b})$ represents a modulated chroma vector whose elements are shifted by amount ζ.

The similarity between $\mathbf{M}_a$ and $\mathbf{M}_b$ is defined as follows:

$$Sim_{\mathbf{M}}(\mathbf{M}_a, \mathbf{M}_b) = \max_{\zeta} \left\{ \frac{LCSS(\mathbf{M}_a, \mathbf{M}_b^{\zeta})}{\min(N_{\mathbf{M}_a}, N_{\mathbf{M}_b})} \right\}, \tag{13}$$

where $LCSS(\mathbf{M}_a, \mathbf{M}_b^{\zeta})$ is recursively defined as

$$LCSS(\mathbf{M}_a, \mathbf{M}_b^{\zeta}) = R_{\mathbf{M}_a\mathbf{M}_b^{\zeta}}(l_a, l_b)|_{l_a=N_{\mathbf{M}_a}, l_b=N_{\mathbf{M}_b}}, \tag{14}$$

$$R_{\mathbf{M}_a\mathbf{M}_b^{\zeta}}(l_a, l_b) = \begin{cases} 0 & \text{if } l_a = 0 \text{ or } l_b = 0, \\ 1 + R_{\mathbf{M}_a\mathbf{M}_b^{\zeta}}(l_a - 1, l_b - 1) & \text{if } Sim_{\tau}\{\mathbf{m}_a(l_a), \mathbf{m}_b^{\zeta}(l_b)\} > T_h, \\ \max\{R_{\mathbf{M}_a\mathbf{M}_b^{\zeta}}(l_a - 1, l_b), R_{\mathbf{M}_a\mathbf{M}_b^{\zeta}}(l_a, l_b - 1)\} & \text{otherwise.} \end{cases} \tag{15}$$

$$sim_{\tau}\{\mathbf{m}_a(l_a),\ \mathbf{m}_b^{\zeta}(l_b)\} = 1 - \frac{\left|\tilde{\mathbf{m}}_a(l_a)\tilde{\mathbf{m}}_b^{\zeta}(l_b)\right|}{\sqrt{12}} \tag{16}$$

$$\tilde{\mathbf{m}}_a(l_a) = \frac{\mathbf{m}_a(l_a)}{\max_{\tau} m_{a,\tau}(l_a)}, \tag{17}$$

$$\tilde{\mathbf{m}}_b^{\zeta}(l_b) = \frac{\mathbf{m}_b^{\zeta}(l_b)}{\max_{\tau} m_{b,\tau}^{\zeta}(l_b)}, \tag{18}$$

where $T_h(= 0.8)$ is a positive constant for determining the fitness between two different chroma vectors, $Sim_{\tau}\{\cdot, \cdot\}$ is a similarity between chroma vectors defined in [20], $\tilde{\mathbf{m}}_a(l_a)$ and $\tilde{\mathbf{m}}_b^{\zeta}(l_b)$ are normalized chroma vectors, $m_{a,\tau}(l_a)$ and $m_{b,\tau}^{\zeta}(l_b)$ are elements of the chroma vectors, and τ corresponds to tone, i.e., "C", "D#", "G#", etc. Note that the effectiveness of $Sim_{\tau}\{\cdot, \cdot\}$ is verified in [20]. We then define this similarity as the LCSS kernel for music pieces $\kappa_{\mathbf{M}}^{LCSS}(\cdot, \cdot)$ described as follows:

$$\kappa_{\mathbf{M}}^{LCSS}(\mathbf{M}_a, \mathbf{M}_b) = Sim_{\mathbf{M}}(\mathbf{M}_a, \mathbf{M}_b). \tag{19}$$

3.2.2 Spectrum intersection kernel

Next, we explain the spectrum intersection kernel for music pieces. In order to define the spectrum intersection kernel for music pieces, we firstly calculate p-spectrum-based features in the same way as those of human motions. It should be noted that the proposed method cannot calculate the p-spectrum from music features directly since the music features are defined as sequences of vectors. Therefore, we transform all of the vector components of music features into characters, such as alphabetic letters or numbers, based on hierarchical clustering algorithms, where the characters correspond to cluster numbers. For clustering the vector components, the modulation of music should also be considered in the same way as the LCSS kernel for music pieces. Therefore, clustering considering modulation is necessary. The procedures of this scheme are shown as follows.

Step 1: Calculation of optimal modulation amounts between music features First, the proposed method calculates the optimal modulation amounts ζ^{ab} between two music features $\mathbf{M}_a$ and $\mathbf{M}_b$. This scheme is based on LCSS-based similarity and is defined as follows:

$$\zeta^{ab} = \arg\max_{\zeta} \left\{ \frac{LCSS(\mathbf{M}_a, \mathbf{M}_b^{\zeta})}{\min(N_{\mathbf{M}_a}, N_{\mathbf{M}_b})} \right\}. \tag{20}$$

The optimal modulation amount ζ^{ab} is calculated for all pairs.

Step 2: Similarity measurement between chroma vectors using the obtained optimal modulation amounts Similarity between vector components, which is that between chroma vectors, is calculated using the obtained optimal modulation amounts. For example, the similarity between chroma vectors $\mathbf{m}_a(l_a)$ and $\mathbf{m}_b(l_b)$, which are the l_ath and l_bth components of two arbitrary music features $\mathbf{M}_a$ and $\mathbf{M}_b$, respectively, is calculated using the obtained optimal modulation amount ζ^{ab} and Equation 16 as follows:

$$Sim_c\{\mathbf{m}_a(l_a), \mathbf{m}_b(l_b)\} = 1 - \frac{|\tilde{\mathbf{m}}_a(l_a) - \tilde{\mathbf{m}}_b^{\zeta^{ab}}(l_b)|}{\sqrt{12}}. \tag{21}$$

The above similarity is calculated between two different chroma vectors for all music features.

Step 3: Clustering chroma vectors based on the obtained similarities Using the obtained similarities, the two most similar chroma vectors are assigned to the same cluster for clustering chroma vectors. This scheme is based on the single linkage method [25]. The merging scheme is recursively performed until the number of clusters becomes less than K_M.

Using the clustering results, the proposed method calculates transformed music features $m_j^*(l_M)(l_M = 1, 2, \ldots, N_{\mathbf{M}_j})$, where $m_j^*(l_M)(l_M = 1, 2, \ldots, N_{\mathbf{M}_j})$ is a cluster number assigned to a corresponding chroma vector. Note that vector/matrix transpose is denoted by the superscript ' in this article. The proposed method then calculates p-spectrum-based features from $\mathbf{m}_j^*$. For the following explanations, we denote two transformed music features as $\mathbf{m}_a^* = [m_a^*(1), m_a^*(2), \ldots, m_a^*(N_{\mathbf{M}_a})]'$ and $\mathbf{m}_b^* = [m_b^*(1), m_b^*(2), \ldots, m_b^*(N_{\mathbf{M}_b})]'$, where $\mathbf{m}_a^*$ and $\mathbf{m}_b^*$ are vectors transformed from $\mathbf{M}_a$ and $\mathbf{M}_\mathbf{b}$, respectively, and $m_a^*(l_a)(l_a = 1, 2, \ldots, N_{\mathbf{M}_a})$ and $m_b^*(l_b)(l_b = 1, 2, \ldots, N_{\mathbf{M}_b})$ are the cluster numbers assigned to $\mathbf{m}_a(l_a)$ and $\mathbf{m}_b(l_b)$, respectively. Then, the spectrum intersection kernel for music pieces is calculated in the same way as that for human motions and is defined as follows:

$$\kappa_{\mathbf{M}}^{SI}(\mathbf{m}_a, \mathbf{m}_b) = \kappa^{HI}(\mathbf{r}_p(\mathbf{m}_a^*), \mathbf{r}_p(\mathbf{m}_b^*)). \tag{22}$$

4 Kernel CCA-based music recommendation according to human motion

A method for recommending music pieces suitable for human motions is presented in this section. An overview of the proposed method is shown in Figure 2. In our cross-media recommendation method, pairs of human motions and music pieces that have a close relationship are necessary for effective correlation calculation. Therefore, we prepare these pairs extracted from the same video contents as segments. From the obtained segments, we extract human motion features and music features. More details of these features are given in Appendices A.1 and A.2. By applying kernel CCA to the features of human motions and music pieces, the proposed method calculates their correlation. In this approach, we define new kernel functions that can be

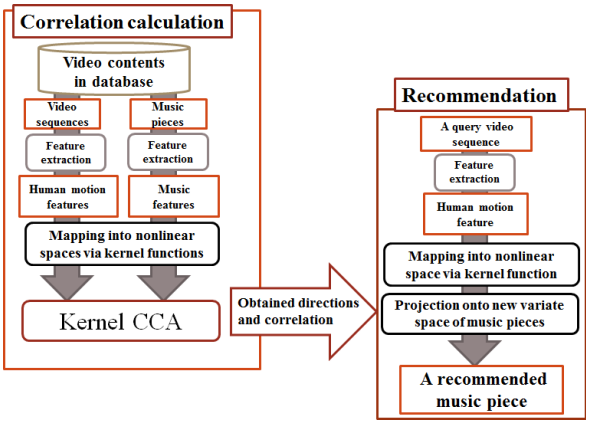

Figure 2 Overview of the proposed method. The *left* and *right* parts in this figure represent the correlation calculation phase and the recommendation phase, respectively, in the proposed method.

used for data having various time lengths and introduce them into the kernel CCA.

Therefore, the proposed method can calculate the correlations by considering their sequential characteristics. Then, effective modeling of the relationship using human motions and music pieces having various time lengths is realized, and successful music recommendation can be expected.

First, we define the features of $\mathbf{V}_j$ and $\mathbf{M}_j (j = 1, 2,..., N)$ in the Hilbert space as $\varphi_{\mathbf{v}}$ (vec[$\mathbf{V}_j$]) and $\varphi_{\mathbf{M}}$ (vec[$\mathbf{M}_j$]), where vec[·] is the vectorization operator that turns a matrix into a vector. Next, we find features

$$\mathbf{s}_j = \mathbf{A}' \left(\phi_{\mathbf{V}}(\mathrm{vec}[\mathbf{V}_j]) - \bar{\phi}_{\mathbf{V}}\right), \tag{23}$$

$$\mathbf{t}_j = \mathbf{B}' \left(\phi_{\mathbf{M}}(\mathrm{vec}[\mathbf{M}_j]) - \bar{\phi}_{\mathbf{M}}\right), \tag{24}$$

$$\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_D], \tag{25}$$

$$\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_D], \tag{26}$$

where $\bar{\phi}_{\mathbf{V}}$ and $\bar{\phi}_{\mathbf{M}}$ are mean vectors of $\varphi_{\mathbf{v}}$ (vec[$\mathbf{V}_j$]) and $\varphi_{\mathbf{M}}$(vec[$\mathbf{M}_j$]) (j = 1, 2,..., N), respectively. The matrices $\mathbf{A}$ and $\mathbf{B}$ are coefficient matrices whose columns $\mathbf{a}_d$ and $\mathbf{b}_d$ (d = 1, 2,..., D), respectively, correspond to the projection directions in Equations 2 and 3, where the value D is the dimension of $\mathbf{A}$ and $\mathbf{B}$. Then, we define a correlation matrix $\mathbf{\Lambda}$ whose diagonal elements are the correlation coefficients $\lambda_d (d = 1,2,..., D)$. The details of the calculation of $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{\Lambda}$ are shown as follows.

In order to obtain $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{\Lambda}$, we use the regularized kernel CCA shown in the previous section. Note that the optimal matrices $\mathbf{A}$ and $\mathbf{B}$ are given by

$$\mathbf{A} = \mathbf{\Xi}_{\mathbf{v}}\mathbf{H}\mathbf{E}_{\mathbf{v}}, \tag{27}$$

$$\mathbf{B} = \mathbf{\Xi}_{\mathbf{M}}\mathbf{H}\mathbf{E}_{\mathbf{M}}, \tag{28}$$

$$\mathbf{\Xi}_{\mathbf{V}} = [\phi_{\mathbf{V}}(\mathrm{vec}[\mathbf{V}_1]), \phi_{\mathbf{V}}(\mathrm{vec}[\mathbf{V}_2]), \ldots, \phi_{\mathbf{V}}(\mathrm{vec}[\mathbf{V}_N])], \tag{29}$$

$$\mathbf{\Xi}_{\mathbf{M}} = [\phi_{\mathbf{M}}(\mathrm{vec}[\mathbf{M}_1]), \phi_{\mathbf{M}}(\mathrm{vec}[\mathbf{M}_2]), \ldots, \phi_{\mathbf{M}}(\mathrm{vec}[\mathbf{M}_N])], \tag{30}$$

where $\mathbf{E}_{\mathbf{V}} = [\mathbf{e}_{\mathbf{V}_1}, \mathbf{e}_{\mathbf{V}_2}, \ldots, \mathbf{e}_{\mathbf{V}_D}]$ and $\mathbf{E}_{\mathbf{M}} = [\mathbf{e}_{\mathbf{M}_1}, \mathbf{e}_{\mathbf{M}_2}, \ldots, \mathbf{e}_{\mathbf{M}_D}]$ are $N \times D$ matrices. Furthermore,

$$\mathbf{H} = \mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}' \tag{31}$$

is a centering matrix, where $\mathbf{I}$ is the $N \times N$ identity matrix, and $\mathbf{1}$ = [1,..., 1]' is an $N \times 1$ vector. From Equations 27 and 28, the following equations are satisfied:

$$\mathbf{a}_d = \mathbf{\Xi}_{\mathbf{V}}\mathbf{H}\mathbf{e}_{\mathbf{V}_d}, \tag{32}$$

$$\mathbf{b}_d = \mathbf{\Xi}_{\mathbf{M}}\mathbf{H}\mathbf{e}_{\mathbf{M}_d}. \tag{33}$$

Then, by calculating the optimal solution $\mathbf{e}_{\mathbf{V}_d}$ and $\mathbf{e}_{\mathbf{M}_d} (d = 1, 2, \ldots, D)$, $\mathbf{A}$ and $\mathbf{B}$ are obtained. In the same way as Equation 4, we calculate the optimal solution $\mathbf{e}_{\mathbf{V}_d}$ and $\mathbf{e}_{\mathbf{M}_d}$ that maximizes

$$\mathcal{L} = \mathbf{e}'_{\mathbf{V}}\mathbf{L}\mathbf{e}_{\mathbf{M}} - \frac{\lambda}{2}(\mathbf{e}'_{\mathbf{V}}\mathbf{M}\mathbf{e}_{\mathbf{V}} - 1) - \frac{\lambda}{2}(\mathbf{e}'_{\mathbf{M}}\mathbf{P}\mathbf{e}_{\mathbf{M}} - 1), \tag{34}$$

where $\mathbf{e}_{\mathbf{V}}$, $\mathbf{e}_{\mathbf{M}}$, and λ correspond to $\mathbf{e}_{\mathbf{V}_d}$, $\mathbf{e}_{\mathbf{M}_d}$, and λ_d, respectively. In the above equation, $\mathbf{L}$, $\mathbf{M}$, and $\mathbf{P}$ are calculated as follows:

$$\mathbf{L} = \frac{1}{N}\mathbf{H}\mathbf{K}_{\mathbf{V}}\mathbf{H}\mathbf{H}\mathbf{K}_{\mathbf{M}}\mathbf{H}, \tag{35}$$

$$\mathbf{M} = \frac{1}{N}\mathbf{H}\mathbf{K}_{\mathbf{V}}\mathbf{H}\mathbf{H}\mathbf{K}_{\mathbf{V}}\mathbf{H} + \eta_1\mathbf{H}\mathbf{K}_{\mathbf{V}}\mathbf{H}, \tag{36}$$

$$\mathbf{P} = \frac{1}{N}\mathbf{H}\mathbf{K}_{\mathbf{M}}\mathbf{H}\mathbf{H}\mathbf{K}_{\mathbf{M}}\mathbf{H} + \eta_2\mathbf{H}\mathbf{K}_{\mathbf{M}}\mathbf{H}. \tag{37}$$

Furthermore, η_1 and η_2 are regularization parameters, and $\mathbf{K}_{\mathbf{V}} (= \mathbf{\Xi}'_{\mathbf{V}}\mathbf{\Xi}_{\mathbf{V}})$ and $\mathbf{K}_{\mathbf{M}} (= \mathbf{\Xi}'_{\mathbf{M}}\mathbf{\Xi}_{\mathbf{M}})$ are matrices whose elements are defined as values of the corresponding kernel functions defined in Section 3. By taking derivatives of Equation 34 with respect to $\mathbf{e}_{\mathbf{V}}$ and $\mathbf{e}_{\mathbf{M}}$, optimal $\mathbf{e}_{\mathbf{V}}$, $\mathbf{e}_{\mathbf{M}}$, and λ can be obtained as solutions of following eigenvalue problems:

$$\mathbf{M}^{-1}\mathbf{L}\mathbf{P}^{-1}\mathbf{L}'\mathbf{e}_{\mathbf{V}} = \lambda^2\mathbf{e}_{\mathbf{V}}, \tag{38}$$

$$\mathbf{P}^{-1}\mathbf{L}'\mathbf{M}^{-1}\mathbf{L}\mathbf{e}_{\mathbf{M}} = \lambda^2\mathbf{e}_{\mathbf{M}}, \tag{39}$$

where λ is obtained as an eigenvalue, and the vectors $\mathbf{e}_{\mathbf{V}}$ and $\mathbf{e}_{\mathbf{M}}$ are, respectively, obtained as eigenvectors. Then, the dth (d = 1, 2,..., D) eigenvalue of λ becomes λ_d, where $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_D$. Note that the dimension D is set to a value for which the cumulative proportion obtained from $\lambda_d (d = 1,2,...,D)$ becomes larger than a threshold. Furthermore, the eigenvectors $\mathbf{e}_{\mathbf{V}}$ and $\mathbf{e}_{\mathbf{M}}$ corresponding to λ_d become $\mathbf{e}_{\mathbf{V}_d}$ and $\mathbf{e}_{\mathbf{M}_d}$, respectively.

From the obtained matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{\Lambda}$, we can estimate the optimal music features from given human motion features, i.e., we can select the best matched music pieces according to human motions. An overview of music recommendation is shown in Figure 3. When a human motion feature $\mathbf{V}_{in}$ is given, we can select the predetermined number of music pieces according to the query human motion that minimize the following distances:

$$d = \| \mathbf{t}_{in} - \hat{\mathbf{t}}_i \|^2 (i = 1, 2, \ldots, M_t), \tag{40}$$

where $\mathbf{t}_{in}$ and $\hat{\mathbf{t}}_i$ are, respectively, the query human motion feature and music features in the database

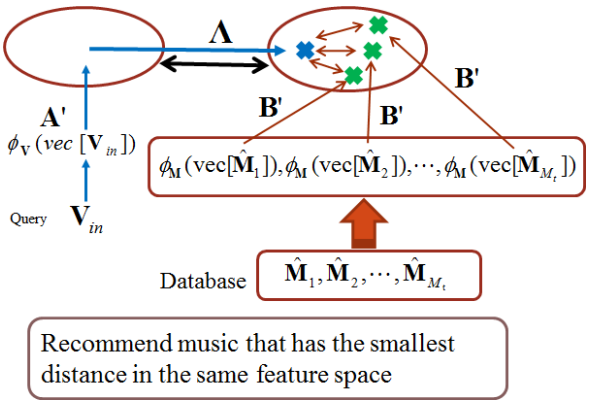

Figure 3 Overview of music recommendation according to human motion.

$\hat{\mathbf{M}}_i (i = 1, 2, \ldots, M_t)$ transformed into the same feature space shown as follows:

$$\begin{aligned}\hat{\mathbf{t}}_i &= \mathbf{B}' \left(\phi_M(\mathrm{vec}[\hat{\mathbf{M}}_i]) - \bar{\phi}_M\right) \\ &= \mathbf{E}'_M \left(\kappa_{\hat{\mathbf{M}}_i} - \frac{1}{N}\mathbf{K}_M \mathbf{1}\right),\end{aligned} \tag{41}$$

$$\begin{aligned}\mathbf{t}_{in} &= \mathbf{\Lambda}\mathbf{A}' \left(\phi_V(\mathrm{vec}[\mathbf{V}_{in}]) - \bar{\phi}_V\right) \\ &= \mathbf{\Lambda}\mathbf{E}'_V \left(\kappa_{\mathbf{V}_{in}} - \frac{1}{N}\mathbf{K}_V \mathbf{1}\right),\end{aligned} \tag{42}$$

and M_t is the number of music pieces in the database. Note that $\kappa_{\mathbf{V}_{in}}$ is an $N \times 1$ vector whose qth elements are $\kappa_V^{LCSS}(\mathbf{V}_{in}, \mathbf{V}_q)$ or $\kappa_V^{SI}(\mathbf{V}_{in}, \mathbf{V}_q)$, and $\kappa_{\hat{\mathbf{M}}_i}$ is an $N \times 1$ vector whose qth elements are $\kappa_M^{LCSS}(\hat{\mathbf{M}}_i, \mathbf{M}_q)$ or $\kappa_M^{SI}(\hat{\mathbf{M}}_i, \mathbf{M}_q)$.

As described above, we can estimate the best matched music pieces according to the human motions. The proposed method calculates the correlation between human motions and music pieces based on the kernel CCA. Then, the proposed method introduces the kernel functions that can be used for time series having various time lengths based on the LCSS or p-spectrum. Therefore, the proposed method enables calculation of the correlation between human motions and music pieces that have various time lengths. Furthermore, effective correlation calculation and successful music recommendation according to human motion based on the obtained correlation are realized.

5 Experimental results

The performance of the proposed method is verified in this section. For the experiments, 170 segments were manually extracted. In the experiments, we used video contents of three classic ballet programs. Of the 170 segments, 44 were from Nutcracker, 54 were from Swan Lake, and 72 were from Sleeping Beauty. Each segment consisted of only one human motion and the background music did not change in the segment. In addition, camera change was not included in the segment. The audio signals in each segment were mono channel, 16 bits per sample and were sampled at 44.1 [kHz]. Human motion features and music features were extracted from the obtained segments.

For evaluation of the performance of our method, we used videos of classic ballet programs. However, there were some differences between motions extracted from classic ballet programs and those extracted in our daily life. In cross-media recommendation, we have to consider whether or not we should recommend contents that have the same meanings as those of queries. For example, when we recommend music pieces from the user's information, recommendation of sad music pieces is not always suitable if the user seems to be sad. Our approach also has to consider the above point. In this article, we focus on extraction of the relationship between human motions and music pieces and perform the recommendation based on the extracted relationship. In addition, we have to prepare some ground truths for evaluation of the proposed method. Therefore, we used videos of classic ballet programs since the human motions and music pieces extracted from the same videos of classic ballet programs had strong and direct relationships.

In order to evaluate the performance of our method, we also prepared five datasets #1 to #5 that were pairs of 100 segments for training (training segments) and 70 segments for testing (testing segments), i.e., a simple cross-validation scheme. It should be noted that we randomly divided the 170 segments into five datasets. The reason for dividing the 170 segments into five datasets was to perform various verifications by changing the combination of test segments and training segments. Then, the number of datasets (five) was simply determined. Furthermore, the training segments and testing segments were obtained from the above prepared 170 segments. For the experiments, 12 kinds of tags representing expression marks in music shown in Table 1 were used. We examined whether each tag could be used for labeling human motions and music pieces. Thus, tags that seemed to be difficult to use for these two media types were removed in this process. Then, we could obtain the above 12 kinds of tags. One suitable tag was manually selected and annotated to each segment for performance verification. In the experiments, one person with musical experience annotated the label that was the best matched to each segment. Generally, annotation should be performed by several people.

Table 1 Description of expression marks

Name	Definition
agitato	Agitated
amabile	Amiable, pleasant
appassionato	Passionately
capriccioso	Unpredictable, volatile
grazioso	Gracefully
lamentoso	Lamenting, mournfully
leggiero	Lightly, delicately
maestoso	Majestically
pesante	Heavy, ponderous
soave	Softly
spiritoso	Spiritedly
tranquillo	Calmly, peacefully

However, since labels, i.e., expression marks in music, were used in the experiment, it was necessary to have the ground truths made by a person who had knowledge of music. Thus, in the experiment, only one person annotated the labels.

First, we show the recommended results (see Additional file 1). In this file, we show original video contents and recommended video contents. The background music pieces of recommended video contents are not original but are music pieces recommended by our method. These results show that our method can recommend a suitable music piece for a human motion.

Next, we quantitatively verify the performance of the proposed method. In this simulation, we verify the effectiveness of our kernel functions. In the proposed method, we define two types of kernel functions, LCSS kernel and spectrum intersection kernel, for human motions and music pieces. Thus, we experimentally compare our two newly defined kernel functions. Using combinations of the kernel functions, we prepared four simulations "Simulation 1"-"Simulation 4", as follows:

- Simulation 1 used the LCSS kernel for both human motions and music pieces.
- Simulation 2 used the spectrum intersection kernel for both human motions and music pieces.
- Simulation 3 used the spectrum intersection kernel for human motions and the LCSS kernel for music pieces.
- Simulation 4 used the LCSS kernel for human motions and the spectrum intersection kernel for music pieces.

These simulations were performed to verify the effectiveness of our two newly defined kernel functions for human motions and music pieces. For the following explanations, we denote the LCSS kernel as "LCSS-K" and the spectrum intersection kernel as "SI-K". In addition, for the experiments, we used the following criterion:

$$\text{Accuracy Score} = \frac{\sum_{i_1=1}^{70} Q_{i_1}^1}{70}, \tag{43}$$

where the denominator corresponds to the number of testing segments. Furthermore, $Q_{i_1}^1 (i_1 = 1, 2, \ldots, 70)$ is one if the tags of three recommended music pieces include the tag of the human motion query.

Otherwise, $Q_{i_1}^1$ is zero. It should be noted that the number of recommended music pieces (three) was simply determined. We next explain how the number of recommended music pieces affects the performance of our method. For the following explanation, we define the terms "over-recommendation" and "mis-recommendation". Over-recommendation means that the recommended results tend to contain music pieces that are not matched to the target human motions as well as matched music pieces, and mis-recommendation means that music pieces that are matched to the target human motions tend not to be correctly selected as the recommendation results. There is a tradeoff relationship between over-recommendation and mis-recommendation. That is, if we increase the number of recommended results, over-recommendation increases and mis-recommendation decreases. On the other hand, if we decrease the number of recommended results, over-recommendation decreases and mis-recommendation increases. Furthermore, we evaluate the recommendation accuracy according to the above criterion. Figure 4 shows that the accuracy score of simulation 1 was higher than accuracy scores of the other simulations. This is because the LCSS kernel can effectively compare human motions and music pieces respectively having different time lengths. Note that in these simulations, we used bi (p = 2)-gram for calculating p-spectrum-based features shown in Equation 9, the number of clusters for chroma vectors is set to K_M = 500 and the parameters in our method are shown in Tables 2, 3, 4 and 5. All of these parameters are empirically determined, and they are set to values that provide the highest accuracy. More details of parameter determination are given in Appendix.

Table 2 Description of parameters used in Simulation 1

Dataset	η_1	η_2	K_c
#1	1.0×10^{-14}	8.0×10^{-3}	1300
#2	6.0×10^{-3}	6.0×10^{-7}	1000
#3	6.0×10^{-13}	8.0×10^{-3}	1200
#4	2.0×10^{-3}	8.0×10^{-13}	1000
#5	6.0×10^{-11}	8.0×10^{-3}	1200

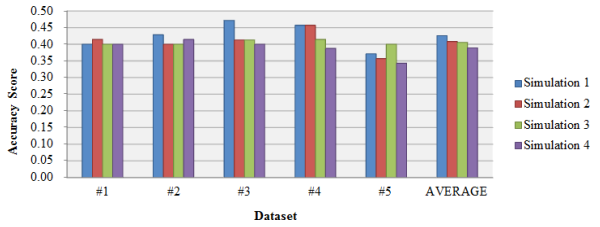

Figure 4 Accuracy scores in each simulation. #1 to #5 are dataset numbers and "AVERAGE" is the average value of the accuracy scores for the datasets.

In the following, we discuss the results obtained. First, we discuss the influence of our human motion features. The features used in our method are based on optical flow and extracted between two regions that contain a human corresponding to two successive frames. This feature can represent movements of arms, legs, hands, etc. However, this feature cannot represent global human movements. This is an important factor for representing motion characteristics of classic ballet. For accurate relationship extraction between human motions and music pieces, it is necessary to improve human motion features into features that can also represent global human movement. This can be complemented using information obtained by much more accurate sensors such as kinect.[d]

Next, we discuss the experimental conditions. In the experiments with the proposed method, we used tags, i.e., expression marks in music, as ground truths. This was annotated to each segment. However, this annotation scheme does not consider the relationship between tags. For example, in Table 1, "agitato" and "appassionato" have similar meanings. Thus, the choice of the 12 kinds of tags might be not suitable. It might be necessary to reconsider the choice tags. Also, we found that it is more important to introduce the relationship between tags into our defined accuracy criteria. However, it is difficult to quantify the relationship between them. Thus, we used only one tag for each segment. This can also be expected by the results of subjective evaluation in next experiment.

We also used comparative methods for verifying performance of the proposed method. For the comparative method, we exchanged the kernel functions into gaussian kernel $\kappa^{\mathrm{G-K}}(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}\right)$ (G - K), sigmoid kernel $\kappa^{\mathrm{S-K}}(\mathbf{x}, \mathbf{y}) = \tanh(\alpha \mathbf{x}'\mathbf{y} + \beta)$ (S-K), and linear kernel $\kappa^{\mathrm{L-K}}(\mathbf{x}, \mathbf{y}) = \mathbf{x}'\mathbf{y}$ (L-K). In this experiment, we set parameters σ(= 5.0), α(= 5.0), and β(= 3.0). It should be noted that these kernel functions cannot be applied to our human motion features and music features directly since the features have various dimensions. Therefore, we simply used the time average of optical flow-based vectors, $\mathbf{v}_j^{\mathrm{avg}}$, for human motion features and the time average of chroma vectors, $\mathbf{m}_j^{\mathrm{avg}}$, for music features. Then, we applied the above three types of kernel functions to the obtained features. Figure 5 shows the results of comparison for each kernel function. These results show that our kernel functions are more effective than other kernel functions. The results also show that it is important to consider the temporal characteristic of data, and our kernel function can successfully consider this characteristic. Note that in this comparison, we used parameters that provide the highest accuracy. The parameters are shown in Tables 6, 7 and 8.

Finally, we show results of subjective evaluation for our recommendation method. We performed subjective evaluation using 15 subjects (User1-User15). Table 9 shows the profiles of the subjects. In the evaluation, we used video contents which consisted of video sequences and music pieces. In the video contents, each video sequence included one human motion, and each music piece was a recommended result by the proposed method according to the human motion. The tasks of the subjective evaluation were as follows:

1. Subjects watched each video content, whose video sequence was a target classic ballet scene and whose music was recommended by the proposed method.

Table 3 Description of parameters used in Simulation 2

Dataset	η_1	η_2	K_c
#1	8.0×10^{-13}	8.0×10^{-3}	1500
#2	4.0×10^{-6}	6.0×10^{-11}	1000
#3	2.0×10^{-11}	8.0×10^{-13}	1000
#4	4.0×10^{-13}	8.0×10^{-13}	1300
#5	1.0×10^{-16}	8.0×10^{-3}	1500

Table 4 Description of parameters used in Simulation 3

Dataset	η_1	η_2	K_c
#1	8.0×10^{-3}	6.0×10^{-11}	1000
#2	4.0×10^{-3}	8.0×10^{-7}	1200
#3	1.0×10^{-14}	8.0×10^{-13}	1000
#4	6.0×10^{-7}	1.0×10^{-2}	1300
#5	1.0×10^{-6}	8.0×10^{-3}	1000

Table 5 Description of parameters used in Simulation 4

Dataset	η_1	η_2	K_c
#1	4.0×10^{-6}	8.0×10^{-13}	1000
#2	2.0×10^{-3}	8.0×10^{-13}	1000
#3	1.0×10^{-13}	8.0×10^{-13}	1200
#4	8.0×10^{-7}	8.0×10^{-3}	1000
#5	1.0×10^{-6}	6.0×10^{-11}	1300

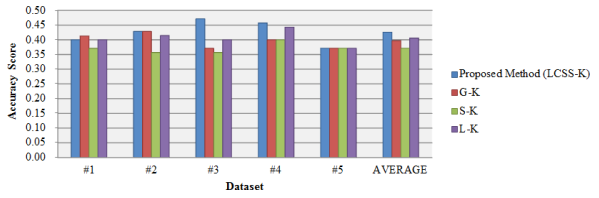

Figure 5 Accuracy comparison in each kernel. #1 to #5 are dataset numbers and "AVERAGE" is the average value of the accuracy scores for the datasets.

2. Subjects determined whether the target classic ballet scene and the recommended music pieces were matched or not. Specifically, they answered yes or no.
3. Procedures 1 and 2 were repeated for 210 video contents.

In the subjective evaluation, we used the recommended results obtained by Simulation 1 in the above-described experiment. We also used datasets #1 and #2 for the subjective evaluation. In the evaluation, we showed the top three recommended results for each query human motion (query segment). Then, 70 query segments were examined and 210 recommended results were obtained for each dataset.

Furthermore, we used two criteria, "Accuracy Score 2" and "Accuracy Score 3", for verifying the performance. Accuracy Score 2 is defined as follows:

$$\text{Accuracy Score 2} = \frac{\sum_{i_2=1}^{70} Q_{i_2}^2}{70}, \tag{44}$$

where the denominator corresponds to the number of query segments. $Q_{i_2}^2 (i_2 = 1, 2 \ldots, 70)$ is one if one or some of the recommended three music pieces at least subjects determined the query human motion and its music piece were matched. Otherwise, $Q_{i_2}^2$ is zero. In addition, Accuracy Score 3 is the ratio of assessment results for 210 music pieces and is defined as follows:

$$\text{Accuracy Score 3} = \frac{\sum_{i_3=1}^{210} Q_{i_3}^3}{210}, \tag{45}$$

where the denominator corresponds to the number of recommended music pieces. Furthermore, $Q_{i_3}^3 (i_3 = 1, 2, \ldots, 210)$ is one if subjects determined the query human motion and its music piece matched. Otherwise, $Q_{i_3}^3$ is zero. Table 10 shows the results of each score in the subjective evaluation. From the results, both scores show higher recommendation accuracy than that of the quantitative evaluation. Therefore, the results of the subjective evaluation confirmed the effectiveness of our method.

6 Conclusions

In this article, we have presented a method for music recommendation according to human motion based on the kernel CCA-based relationship. In the proposed method, we newly defined two types of kernel functions. One is a sequential similarity-based kernel function that uses the LCSS algorithm, and the other is a statistical characteristic-based kernel function that uses the *p*-spectrum. Using these kernel functions, the proposed method enables calculation of the correlation that can consider their sequential characteristics. Furthermore, based on the obtained correlation, the proposed method enables accurate music recommendation according to human motion.

In the experiments, recommendation accuracy was sensitive to the parameters. It is desirable that these parameters be adaptively determined from the datasets. Thus, we need to complement this determination algorithm. Feature selection of the human motions and music pieces is also needed for more accurate extraction of the relationship between human motions and music pieces. These topics will be the subjects of subsequent studies.

Table 6 Description of parameters used in gaussian kernel

Dataset	η_1	η_2
#1	8.0×10^{-13}	8.0×10^{-3}
#2	4.0×10^{-7}	8.0×10^{-13}
#3	8.0×10^{-7}	8.0×10^{-13}
#4	6.0×10^{-13}	2.0×10^{-7}
#5	8.0×10^{-7}	8.0×10^{-13}

Table 7 Description of parameters used in sigmoid kernel

Dataset	η_1	η_2
#1	8.0×10^{-7}	8.0×10^{-3}
#2	6.0×10^{-3}	1.0×10^{-2}
#3	1.0×10^{-6}	2.0×10^{-7}
#4	4.0×10^{-3}	1.0×10^{-14}
#5	1.0×10^{-6}	4.0×10^{-11}

Table 8 Description of parameters used in linear kernel

Dataset	η_1	η_2
#1	4.0×10^{-11}	2.0×10^{-7}
#2	1.0×10^{-16}	1.0×10^{-16}
#3	8.0×10^{-11}	2.0×10^{-3}
#4	1.0×10^{-10}	8.0×10^{-13}
#5	1.0×10^{-14}	8.0×10^{-13}

Table 9 Profiles of the subjects

Number of the subjects (male/female)	15(14/1)
Nationality(number)	Australia(1), Syria(1), China(3), Japan (10)
Ages(years)	22-30

Endnotes

[a]In this article, we simply denote "retrieval and recommendation" as recommendation hereafter. [b]In this article, video sequences are defined as data that contain only visual signals, and video contents are defined as data that contain both visual signals and audio signals. [c]In this section, we assume that $\mathbb{E}[\phi_{\mathbf{x}}(\mathbf{x})] = 0$ and $\mathbb{E}[\phi_{\mathbf{y}}(\mathbf{y})] = 0$ for brief explanation, where $\mathbb{E}[\cdot]$ denotes the sample average of the random variates. [d]http://www.xbox.com/en-US/Kinect.

Appendix A: Feature extraction

In this article, we use human motion features and music features. Here, each feature extraction is explained in detail. Segments are extracted from video contents, i.e., video contents are separated into some segments S_j (j = 1,2,..., N). Then, human motion features and music features are extracted from each segment. In this appendix, we explain methods for extraction of human motion features and music features in **A.1** and **A.2**, respectively.

A.1 Extraction of human motion features

First, the proposed method separates segments S_j into frames $f_j^k(k = 1, 2, \ldots, N_j)$, where N_j is the number of frames in segment S_j. Furthermore, a rectangular region including one human is clipped from each frame, and they are regularized to the same size. In this article, we assume that this rectangular region has previously been obtained. Deciding the rectangular regions including humans might be difficult. However, there are several methods for extracting/deciding human regions from video sequences [26,27]. These methods achieved accurate human region detection by combining visual information and sensor information such as kinect,[d] using a stereo-camera, or using a camera for which position is calibrated. Although we extract the rectangular region manually for simplicity, we consider that a certain precision can be guaranteed using these methods.

Next, we show the calculation of optical flow-based vectors. For calculating optical flows from segments, we firstly divide regions of frame f_j^k into blocks $\mathcal{B}_j^b(b = 1, 2, \ldots, N^{\mathcal{B}_j})$, where $N^{\mathcal{B}_j}(= 1600)$ is the number of blocks in each frame. Then, based on the Lucas-Kanade Algorithm [19], the optical flow in each block $\mathcal{B}_i^b$ is calculated between two successive regions from f_j^{k+1} to f_j^k for all segments S_j. Then, we obtain optical flow-based vectors $\mathbf{v}_j(k)(k = 1, 2 \ldots, N_{\mathbf{v}_j})$ containing vertical and horizontal direction optical flow values for all blocks. Then, $N_{\mathbf{v}_j}$ corresponds to N_j-1.

In this article, the human motion feature $\mathbf{V}_j$ of segment S_j is obtained as the sequence of the optical flow-based vector $\mathbf{v}_j(k)$. The features obtained by the above procedure represent the temporal characteristics of human motions.

A.2 Extraction of music features

The proposed method uses chromagrams [24]. A chromagram represents the temporal sequence of chroma vectors over time and is calculated from each segment. Furthermore, the chroma vector represents magnitude distribution on the chroma that is assigned into 12 pitch classes within an octave, and thus the chroma vector has 12 dimensions. The 12-dimensional chroma vector $\mathbf{m}(t)$ is extracted from the magnitude spectrum $\Psi_\tau(f_{Hz}, t)$, which is calculated using short-time Fourier transform (STFT), where f_{Hz} is frequency and t is time in an audio signal. The $\tau(\tau = 1, 2,\ldots,12)$th element of $\mathbf{m}(t)$ corresponds to a pitch class of equal temperament and is defined as follows:

$$m_\tau(t) = \sum_{h=Oct_L}^{Oct_H} \int_{-\infty}^{\infty} BPF_{\tau,h}(f_{Hz})\Psi(f_{Hz}, t)df_{Hz},$$

where Oct_H and Oct_L represent the highest and lowest octave positions, respectively. Furthermore, $BPF_{\tau,h}(f_{Hz})$ is a bandpass filter that passes the signal at the log-scale frequency $F_{\tau,h}$(in cents) of pitch class τ (chroma) in

Table 10 Accuracy of subjective evaluation of each user in Dataset #1 and Dataset #2

User	Accuracy #1	Score 2 #2	Accuracy #1	Score 3 #2
User1	0.91	0.93	0.53	0.60
User2	0.99	0.97	0.71	0.79
User3	1.00	0.97	0.65	0.47
User4	0.96	0.80	0.40	0.36
User5	0.67	0.51	0.31	0.19
User6	0.93	0.93	0.38	0.33
User7	0.97	0.96	0.55	0.47
User8	0.99	0.99	0.51	0.60
User9	0.56	0.66	0.23	0.29
User10	0.99	0.99	0.46	0.50
User11	0.91	0.91	0.50	0.50
User12	0.93	0.97	0.45	0.43
User13	0.90	0.97	0.45	0.43
User14	0.94	0.99	0.54	0.63
User15	0.90	1.00	0.50	0.50
Average	0.90	0.90	0.48	0.48

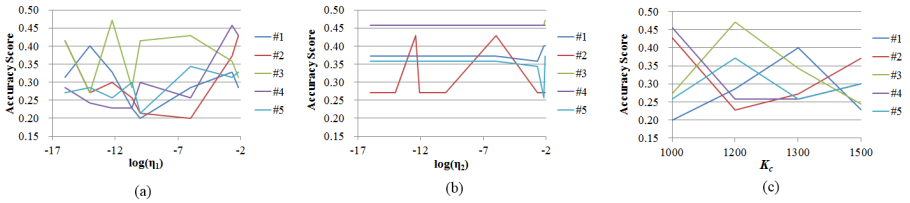

Figure 6 Relationships between η_1, η_2, K_c, and Accuracy Score in Simulation 1. (a) Relationship between η_1 and Accuracy Score, (b) Relationship between η_2 and Accuracy Score, and (c) Relationship between K_c and Accuracy Score. For examining each parameter, the other parameters are fixed. Then, in (b), datasets #3, #4 and #5 have almost the same characteristics.

octave position h (height) as shown in the following equation:

$$F_{\tau,h} = 1200h + 100(\tau - 1).$$

We define a chromagram that represents a temporal sequence of 12-dimensional chroma vectors extracted by the above procedure in segment S_j as the music features $\mathbf{M}_j = [\mathbf{m}_j(1), \mathbf{m}_j(2), \ldots, \mathbf{m}_j(N_{\mathbf{M}_j})]$, where $N_{\mathbf{M}_j}$ is the number of components of $\mathbf{M}_j$. Details of the chroma vector and the chromagram are shown in [20].

Appendix B: Parameter determination

In this section, we explain the parameter determination. First, for the determination of parameters, we performed experiments to show the relationship between the accuracy score and the parameters. Figure 6 shows the relationships between the accuracy score and parameters in Simulation 1. From the obtained results, it can be seen that the kernel CCA-based approach tends to be sensitive for the parameters. It should be noted that in the dataset used for the experiments, there are quite different types of pairs of human motions and music pieces. Then, for similar pairs of human motions and music pieces, we will be able to use fixed parameters and obtain accurate results. Therefore, it can be seemed that stable recommendation accuracy scores are achieved using parameters that are determined from datasets that have similar characteristics. This means that for stable recommendation, some schemes performing clustering and classification of the contents become necessary as pre-procedures. The other simulations and other database are also sensitive the same as the shown results. For the above reasons, we used the parameters that provided the highest accuracy. Thus, the parameters were not determined by cross-validation. However, we recognized that such parameter should be determined by the cross-validation. This is our future work.

Additional material

Additional file 1: Recommended results. Additional file 1.mov; Description of data: This video content shows our recommendation results. In this video content, original video contents and recommended results, whose video contents' background music are music pieces recommended by our method, are shown.

Abbreviations

CCA: canonical correlation analysis; MMD: multimedia documents; LCSS: longest common subsequence; LCSS-K, LCSS: kernel; SI-K: spectrum intersection kernel; G-K: gaussian kernel; S-K: sigmoid kernel; L-K: linear kernel.

Acknowledgements

This study was partly supported by the Grant-in-Aid for Scientific Research (B) 21300030, Japan Society for the Promotion of Science (JSPS).

Competing interests

The authors declare that they have no competing interests.

Received: 15 April 2011 Accepted: 5 December 2011
Published: 5 December 2011

References

1. I Kim, J Lee, Y Kwon, S Par, Content-based image retrieval method using color and shape features, in *Proceedings of the 1997 International Conference on Information, Communication and Signal Processing*, pp. 948–952 (1997)
2. R Zhang, Z Zhang, Effective image retrieval based on hidden concept discovery in image database. IEEE Trans Image Process. **16**(2), 562–572 (2006)
3. X He, W Ma, H Zhang, Learning an image manifold for retrieval, in *Proceedings of the ACM Multimedia Conference* (2004)
4. G Guo, S Li, Content-based audio classification and retrieval by support vector machines. IEEE Trans Neural Networks **14**(1), 209–215 (2003). doi:10.1109/TNN.2002.806626
5. R Typke, F Wiering, R Veltkamp, A survey of music information retrieval systems, in *Proceedings of the ISMIR* (2005)
6. J Shen, J Shepherd, A Ngu, Towards effective content-based music retrieval with multiple acoustic feature combination. IEEE Trans Multimedia **8**, 1179–1189 (2006)
7. H Greenspan, J Goldberger, A Mayer, Probabilistic space-time video modeling via piecewise GMM. IEEE Trans Pattern Anal Mach Intell. **26**(3), 384–396 (2004). doi:10.1109/TPAMI.2004.1262334
8. J Fan, A Elmagarmid, X Zhu, W Aref, L Wu, ClassView: hierarchical video shot classification, indexing, and accessing. IEEE Trans Multimedia **6**(1), 70–86 (2004). doi:10.1109/TMM.2003.819583
9. X Li, T Dacheng, S Maybank, Visual music and musical vision. Neurocomputing **71**, 2023–2028 (2008). doi:10.1016/j.neucom.2008.01.025
10. A Fujii, K Itou, T Akiba, T Ishikawa, A cross-media retrieval system for lecture videos, in *Proceedings of the Eighth European Conference on Speech Communication and Technology (Eurospeech 2003)*, 1149–1152 (2003)
11. Y Zhuang, Y Yang, F Wu, Mining semantic correlation of heterogeneous multimedia data for cross-media retrieval. IEEE Trans Multimedia **10**(2), 221–229 (2008)
12. Y Yang, Y Zhuang, F Wu, Y Pan, Harmonizing hierarchical manifolds for multimedia document semantics under standing and cross-media retrieval. IEEE Trans Multimedia **10**(3), 437–446 (2008)

13. S Akaho, A kernel method for canonical correlation analysis, in *International Meeting of Psychometric Society* **1** (2001)
14. S Jun, B Han, E Hwang, A similar music retrieval scheme based on musical mood variation, in *First Asian Conference on Intelligent Information and Database Systems* **1**, 167–172 (2009)
15. J Mercer, Functions of positive and negative type, and their connection with the theory of integral equations. Trans London Philos Soc (A). **209**, 415–446 (1909). doi:10.1098/rsta.1909.0016
16. C Leslie, E Eskin, W Noble, The spectrum kernel: a string kernel for SVM protein classification, in *Proceedings of the Pacific Biocomputing Symposium*, 566–575 (2002)
17. A Barla, F Odone, A Verri, Histogram intersection kernel for image classification. in *ICIP(3)* 513–516 (2006)
18. C Gruber, T Gruber, B Sick, Online signature verification with new time series kernels for support vector machines. *Advances in Biometrics*. **3832**, 500–508 (2005). doi:10.1007/11608288_67
19. B Lucas, T Kanade, An iterative image registration technique with an application to stereo vision, in *Proceedings of the DARPA IU Workshop*, 121–130 (1984)
20. M Goto, A chorus-section detection method for musical audio signals and its application to a music listening station. IEEE Trans Audio Speech Language Process. **14**(5), 1783–1794 (2006)
21. J MacQueen, Some methods for classification and analysis of multivariate observations, in *Proceedings of the Fifth Berkeley Symposium on Math. Statistics and Probability* **1**, 281–297 (1967)
22. J Mariethoz, S Bengio, A kernel trick for sequences applied to text-independent speaker verification systems. Pattern Recognition **40**(8), 2315–2324 (2007). doi:10.1016/j.patcog.2007.01.011
23. G Camps-Valls, J Martin-Guerrero, J Rojo-Alvarez, E Soria-Olivas, Fuzzy sigmoid kernel for support vector classifier. Neurocomputing **62**, 501–506 (2004)
24. GH Wakefield, Mathematical representation of joint timechroma distributions, in *SPIE* (1999)
25. R Xu, W Dunsch II, Survey of clustering algorithms. IEEE Trans Neural Networks **16**(3), 645–678 (2005). doi:10.1109/TNN.2005.845141
26. D Navneet, T Bill, S Cordelia, Human detection using oriented histograms of flow and appearance. Comput Vision ECCV 2006. **3952**, 428–441 (2006). doi:10.1007/11744047_33
27. K Mikolajczyk, C Schmid, A Zisserman, Human detection based on a probabilistic assembly of robust part detectors, in *Proceedings of the Eighth European Conference on Computer Vision*, vol. 1. Prague, Czech Republic, 69–81 (2004)

doi:10.1186/1687-6180-2011-121
Cite this article as: Ohkushi *et al.*: **Music recommendation according to human motion based on kernel CCA-based relationship.** *EURASIP Journal on Advances in Signal Processing* 2011 **2011**:121.