Journal of Information Processing Vol.22 No.1 67–75 (Jan. 2014)

[DOI: 10.2197/ipsjjip.22.67]

Regular Paper

Adaptive Keypose Extraction from Motion Capture Data

Takeshi Miura[1,a)] Takaaki Kaiga[1,2] Hiroaki Katsura[3] Katsubumi Tajima[1]
Takeshi Shibata[4] Hideo Tamamoto[5]

Received: April 5, 2013, Accepted: September 13, 2013

Abstract: In this paper, we present a novel method to extract keyposes from motion-capture data streams. It adaptively extracts keyposes in response to the motion characteristics of a given data stream. We adopt an approach to detect local minima in the temporal variation of motion speed. In the developed algorithm, the intensity of each local minimum is first evaluated by using a set of signals; it is obtained by applying a set of low-pass filters to a one-dimensional motion-speed data stream. The cut-off frequencies of the filters are distributed over a wide frequency range. By adding up the speed-descent values of each local minimum over all the signals, we exhaustively obtain the information on its intensity provided at all the time-scale levels covered by a given data stream. Then, the obtained intensity values are categorized by a clustering algorithm; the local minima categorized as those of little significance are deleted and the remaining ones are fixed as those giving keyposes. Experimental results showed that the present method provided results comparable to the best of those given by the methods previously proposed. This was achieved without readjusting the values of parameters used in the algorithm. Readjustment was indispensable for the other methods to obtain good results.

Keywords: motion capture, keypose extraction, motion characteristic

1. Introduction

Nowadays, motion-capture (Mocap) data streams are widely used for various applications such as creating computer animations of human-like characters. Nevertheless, they are often marked by their unfavorable characteristics such as high dimensionality, great quantity and the lack of structured information on motion sequences. This makes it difficult to browse, edit and reuse them. Several approaches to overcome this issue have been proposed; a typical one is "summarizing" a Mocap data stream in an organized structure, i.e., a series of keyposes symbolizing motion sequences included in the data stream.

Here, we define a *keypose* as each of the representative poses in a motion sequence. Using keyposes allows us, for example, to extract boundaries of unit gestures or highlight moments in a given motion sequence, or to represent a motion sequence as a train of still images. This definition is almost the same with that of Refs. [1] or [2]. Although this is also similar to that of a *keyframe*, there is a clear distinction between them; we do not necessarily require keyposes to precisely reconstruct the original motion sequence by interpolating themselves. This condition is often required for extracting keyframes from a Mocap data stream [3]. Rather, we regard keyposes as the elements which concisely represent the fundamental structure of a motion sequence, exclusive of subtle motion fluctuations whose information is needed only for the highly accurate reconstruction of the original motion sequence.

In the cases of browsing Mocap data streams in databases, for example, it is desirable that keyposes in each data stream are automatically extracted. As for the cases of subjective tasks such as editing Mocap data streams, on the other hand, users may select keyposes manually by themselves so as to satisfy the conditions they set. However, automatically preparing the candidates of keyposes in advance can help in reducing users' effort even in such cases. Therefore, it is reasonable to think that developing an effective method to automatically extract keyposes from Mocap data streams is an important subject.

Many researchers have proposed the methods to automatically extract keyposes. The details of them will be mentioned in Section 2. We focus on the issue found in common among almost all the methods previously proposed. To utilize these methods, users have to manually adjust the values of several parameters used in keypose-extraction procedures. In most cases, the optimal values of these parameters vary depending on the characteristics of the Mocap data stream analyzed. The readjustment of the parameters can thereby be needed at every Mocap data stream, or at least at every motion category. Readjustment process is often tedious, and sometimes requires the knowledge and experience to judge whether the extracted keyposes are proper or not.

To resolve this issue, we propose a novel method which adaptively extracts keyposes in response to the characteristics of a given Mocap data stream. We adopt an approach in which the moments giving local minima in the temporal variation of motion

[1] Graduate School of Engineering and Resource Science, Akita University, Akita 010–8502, Japan
[2] Digital Art Factory, Warabi-za Co., Ltd., Semboku, Akita 014–1192, Japan
[3] Faculty of Education and Human Studies, Akita University, Akita 010–8502, Japan
[4] Venture Business Laboratory, Akita University, Akita 010–8502, Japan
[5] Akita University, Akita 010–8502, Japan
[a)] miura@ipc.akita-u.ac.jp

© 2014 Information Processing Society of Japan

speed are detected; strong correlation between the appearance of the local minima of motion speed and that of keyposes has been pointed out [4], [5]. However, not all the local minima necessarily correspond to keyposes. Several local minima may be attributed to small speed fluctuations in the middle of motion. Therefore, we try to develop an algorithm to appropriately select the local minima actually giving keyposes.

In the developed algorithm, a one-dimensional motion-speed data stream is first obtained by using the dimensionality-reduction method of Ref. [6]. Then, a set of low-pass filters is applied to the data stream; the cut-off frequencies of the filters are distributed over a wide frequency range. The "intensity" of each motion-speed local minimum is evaluated by adding up its speed-descent values over all the filtered signals. The information provided at all the time-scale levels covered by a given data stream is exhaustively extracted in this process. Finally, the obtained intensity values are categorized by a clustering algorithm; the local minima categorized as those of little significance are deleted and the remaining ones are selected as those giving keyposes.

Since the above categorization is performed by using only the speed data of a given data stream, the obtained result inevitably corresponds to its motion-speed characteristics. Consequently, it becomes possible to extract keyposes in response to the characteristics of a given data stream without manual readjustment.

To evaluate the developed method, we conducted an experiment in which Mocap data streams selected from multiple motion categories were used. In the experiment, we compared the developed method with the methods previously proposed. The developed method provided all the motion categories with results comparable to the best of those given by the other methods. This was achieved without readjusting the values of parameters used in the algorithm newly introduced; readjustment was indispensable for the other methods to obtain good results.

The remainder of this paper is organized as follows. We first review the related work in Section 2. We describe the keypose-extraction algorithm in Section 3. We verify the effectiveness of the developed method in Section 4. Conclusions are finally summarized in Section 5.

2. Related Work

Various keypose-extraction approaches have been proposed up to the present[*1]. We classify them into four categories: curve simplification, clustering, matrix factorization and breakpoint detection.

In curve-simplification algorithms, a Mocap data stream is treated as a curve in a high dimensional space; the curve is simplified into a set of straight lines, and the endpoints of each line are regarded as keyposes. Lim et al. [7] presented a typical recursive procedure; a curve sandwiched between the endpoints of a line is divided into two segments at the point most distant from the line. This procedure is repeated until the maximum distance of any curve point from the line becomes smaller than the error margin prepared. To implement this algorithm, users have to manually set up an appropriate error-margin value.

Clustering-based approaches classify similar poses in a Mocap data stream into a cluster, and select a representative pose in each cluster as a keypose. For example, Liu et al. [8] selected the pose appearing at the first frame of each cluster obtained by a simple clustering algorithm. As for clustering-based approaches, the disadvantage that the temporal relations between poses are not considered has been pointed out [3]. In addition, parameters such as a threshold to judge whether a given pose belongs to existing clusters have to be prepared by users.

Matrix-factorization approaches are applied to a Mocap data stream represented as a matrix; e.g., each frame in a data stream is represented as a vector and all the vectors are placed in the rows of a matrix. Huang et al. [9] proposed a typical matrix-factorization approach called Key Probe, in which the keypose-extraction process is treated as a least-squares optimization problem. Although the algorithm used in Key Probe was well organized, an issue which cannot be ignored has been pointed out [3], [10]; the computing speed of Key Probe is considerably slow due to its quadratic time complexity. Furthermore, this approach also requires users to set up an appropriate error-margin value.

In breakpoint-detection methods, breakpoints in the time series of some sort of quantity representing motion feature are regarded as moments giving keyposes. A typical example is the detection of local minima of motion speed (already mentioned in Section 1); Shiratori et al. [5] presented a trial to extract keyposes from dance performance. On the other hand, So et al. [1] adopted the mutual information measure as the quantity to detect directional change in motion patterns, and Assa et al. [2] used the curvature of a motion curve to detect significant points. Since the quantity peculiar to body motion is used in breakpoint-detection methods, it is easy to understand the correspondence of each computation procedure to the characteristics of actual human motion. However, manually setting up the values of parameters used in keypose-extraction procedures is also needed in this type of approaches.

All the approaches mentioned above require users to manually adjust the values of parameters used in keypose-extraction procedures. Liu et al. [10] presented a method to overcome this issue; the Simplex Hybrid Genetic Algorithm provides the optimal set of keyposes without the readjustment of parameters.

We adopt the style of detecting local minima of motion speed. This is attributed to the fact that an effective method to reduce the dimensionality of motion-speed data was developed [6]; it has thereby become easy to reorganize procedures to detect local minima of motion speed. As mentioned in Section 1, each local minimum is rated by its intensity value in the proposed algorithm. This allows users to easily modify the obtained results as the need arises; the set of intensity values can be used as a reference to select the candidates to be added or deleted. Such a property was not provided in Ref. [10].

As mentioned in Section 1, we compared the method devel-

[*1] Several of the methods mentioned in this section were proposed as those to extract *keyframes*. However, we regard them as methods which also belong to the group of *keypose*-extraction approaches, because adjusting the parameters used in them may provide the possibility of extracting keyposes.

© 2014 Information Processing Society of Japan

oped in this paper with those previously proposed. The latter ones were selected from all the categories of keypose-extraction approaches except the matrix-factorization category, because the matrix-factorization approach takes too much computation time due to its quadratic time complexity. The details of the comparison will be described in Section 4.

3. The Keypose-Extraction Algorithm

3.1 The One-dimensional Motion-speed Data Stream

We derive the keypose-extraction algorithm in this section. A Mocap data stream is given as a high dimensional signal as mentioned in Section 1. This makes it complicated and time consuming to directly analyze raw Mocap data streams. Therefore, it is desirable to reduce the dimensionality of Mocap data streams prior to analyzing them. As for the temporal variation of motion speed, an effective dimensionality-reduction method was presented in Ref. [6] as already mentioned; a one-dimensional motion-speed data stream is provided by a simple calculation. We adopt this method as follows.

Consider the positions of 16 principal joints (including end effectors): shoulders, elbows, wrists, fingers, knees, ankles, toes, neck and head. In this case, a Mocap data stream is described as a time series of $3J$-dimensional vectors ($J = 16$):

$$\boldsymbol{P} = [\ \boldsymbol{p}(1)\quad \boldsymbol{p}(2)\quad \cdots\quad \boldsymbol{p}(N)\ ]^T \tag{1}$$
$$\boldsymbol{p}(n) = [\ p_1(n)\quad p_2(n)\quad \cdots\quad p_{3J}(n)\ ]^T$$

where $p_j(n)$ is the jth coordinate of joint positions at the nth frame and N is the number of frames included in the Mocap data stream analyzed, respectively. Coordinates of joint positions are described in the coordinate system fixed to the pelvis. Bevilacqua et al. [4] pointed out that low-pass filtering position data effectively eliminates the influence of jitter on motion-speed data. Hence, we apply a low-pass filter to $\boldsymbol{P}$:

$$\boldsymbol{P}_{\mathrm{F}} = \boldsymbol{F}_{\mathrm{LP}}[\boldsymbol{P}, f_{\mathrm{c}}] = [\ \boldsymbol{p}_{\mathrm{F}}(1)\quad \boldsymbol{p}_{\mathrm{F}}(2)\quad \cdots\quad \boldsymbol{p}_{\mathrm{F}}(N)\ ]^T \tag{2}$$
$$\boldsymbol{p}_{\mathrm{F}}(n) = [\ p_{\mathrm{F},1}(n)\quad p_{\mathrm{F},2}(n)\quad \cdots\quad p_{\mathrm{F},3J}(n)\ ]^T$$

where $\boldsymbol{P}_{\mathrm{F}}$ is the filtered data stream and $\boldsymbol{F}_{\mathrm{LP}}[\boldsymbol{P}, f_{\mathrm{c}}]$ means the application of a low-pass filter with the cut-off frequency f_{c} to $\boldsymbol{P}$. By using $\boldsymbol{P}_{\mathrm{F}}$, we obtain the one-dimensional motion-speed data stream $\boldsymbol{V}$ as follows [6]:

$$\boldsymbol{V} = [\ v(1)\quad v(2)\quad \cdots\quad v(N)\ ]^T \tag{3}$$
$$v(n) = \frac{\sqrt{\sum_{j=1}^{3J}\{p_{\mathrm{F},j}(n+1) - p_{\mathrm{F},j}(n)\}^2}}{\Delta t}$$

where Δt is the sampling time. We use this data stream to extract keyposes; the detail will be shown in the next subsection.

In this paper, we adopt the Hodrick-Prescott filter (HP filter) [11], [12] as a low-pass filter; the growth component [11] of an inputted time series can be regarded as a low-pass filtered output, and the 50-percent-gain frequency [12] can be used as a cut-off frequency. The computation of the HP filter is fast and its computational complexity does not depend on the value of the cut-off frequency.

3.2 The Keypose-extraction Algorithm

As mentioned in Section 1, we assume that the frames giving keyposes can be extracted from those giving local minima of motion speed; we have to appropriately select the frames actually giving keyposes from the above candidate local minima. The criterion to judge whether a candidate gives a keypose is thought to depend on the motion-speed characteristics of a given Mocap data stream: e.g., the time scale of speed variation or the speed range covered by a given data stream. Here, we present a method to adaptively select appropriate candidates in response to the variation of such motion-speed characteristics.

It seems to be reasonable to think that there is a high possibility a local minimum with a large speed descent gives a keypose. To quantitatively evaluate each candidate, therefore, we define the intensity of a local minimum based on the degree of speed descent.

An example of speed descent around a motion-speed local minimum is shown in **Fig. 1**. In general, the speed descent at a local minimum can be estimated by obtaining the speed difference from the average speed around it. As for the case of the local minimum M in Fig. 1, its depression is slightly deeper than those of the adjacent ones. To appropriately evaluate this tendency, we must consider the speed descent from the average speed obtained at a relatively short time-scale level (blue arrow line in Fig. 1). On the other hand, M is located at a trough of a simplified motion-speed curve (red curve in Fig. 1) which represents the speed characteristics in the long-time-scale range. This means that M plays an important role at a long time-scale level; therefore, we must also consider the speed descent from the long-time-scale average (red arrow line in Fig. 1). These facts suggest that the speed descent at each local minimum should be treated as the composition of various time-scale components.

To take the above property into account, we use a set of low-pass filters; a motion-speed data stream is simplified in accordance with the time-scale level corresponding to the cut-off frequency of each low-pass filter. The process of applying the filters to the motion-speed data stream $\boldsymbol{V}$ is shown in **Fig. 2**. Only the local minima having a certain degree of speed descent at a given time-scale level remain in the simplified data stream $\boldsymbol{F}_{\mathrm{LP}}[\boldsymbol{V}, f]$ ($f = f_{\min}, \cdots, f_i, \cdots, f_j, \cdots, f_{\max}$) (left side of Fig. 2). To evaluate the intensity of each candidate with respect to the entire time-scale range, we add up its speed-descent values over all the time-scale levels, namely over $[f_{\min}, f_{\max}]$ (right side of Fig. 2). At each time-scale level, the speed-descent value is given as follows:

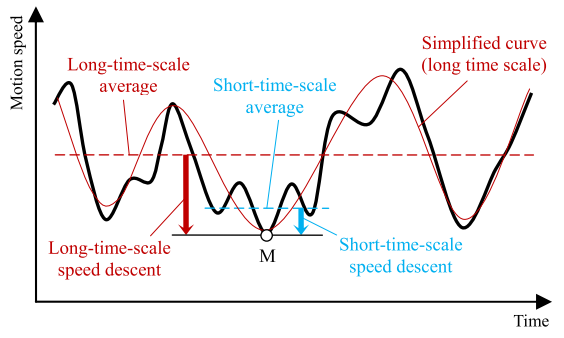


Fig. 1 Motion-speed descent around a local minimum.

© 2014 Information Processing Society of Japan

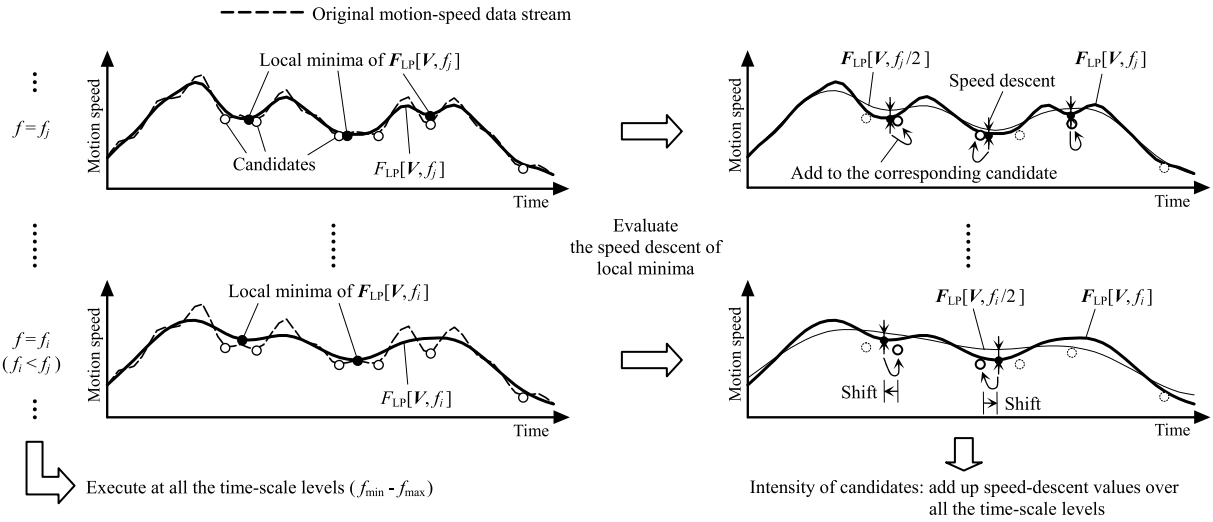


Fig. 2 Evaluation of the intensity of keypose candidates.

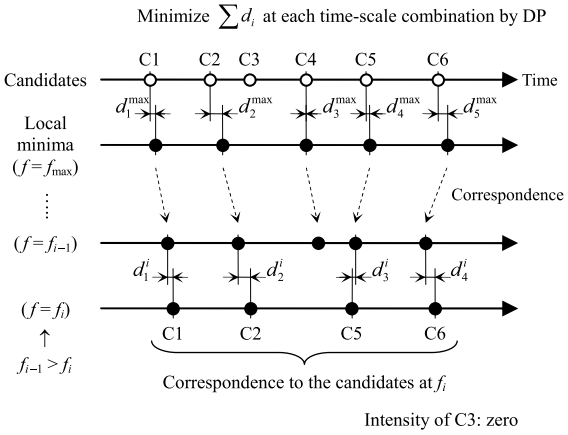


Fig. 3 Correspondence of local minima to keypose candidates.

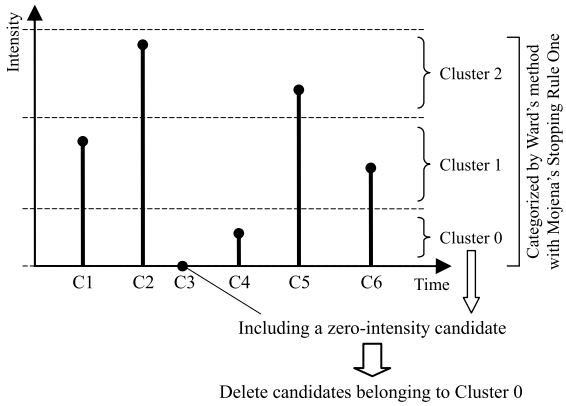


Fig. 4 Categorization of keypose candidates by their intensity values.

$$u(f) = |\boldsymbol{F}_{\mathrm{LP}}[\boldsymbol{V}, f] - \boldsymbol{F}_{\mathrm{LP}}[\boldsymbol{V}, f/2]| \tag{4}$$

where the average speed at each time-scale level is given as $\boldsymbol{F}_{\mathrm{LP}}[\boldsymbol{V}, f/2]$.

The minimum-width time-scale range needed in the above analysis, i.e., the range outside which the $u(f)$ values decrease to almost zero, can change at every data stream, depending on the variation of motion-speed characteristics. On the other hand, the range outside the above minimum range has little influence on intensity values. By making $[f_{\min}, f_{\max}]$ as wide as possible, therefore, we can guarantee the robustness against the variation of motion-speed characteristics without readjusting the time-scale range.

Since time resolution becomes poor in the long-time-scale range, each candidate can shift its position on the time axis as shown in the right side of Fig. 2. The amount of shift often becomes unignorable. Consequently, the correspondence between the original candidates and the local minima extracted at a given time-scale level may become ambiguous. To resolve this issue, we must provide a procedure to properly fix the correspondence of the local minima to the original candidates, guaranteeing consistency throughout the entire time-scale range.

The procedure adopted is shown in **Fig. 3**. First, the correspondence of the local minima extracted at the shortest time-scale level (i.e., $f = f_{\max}$, with the highest time resolution) to the original candidates is fixed by dynamic programming (DP) [13]; the sum of time difference between a local minimum and the corresponding candidate ($d_i^{\max}$'s in Fig. 3) is used as the cost function. Then, DP is also applied to all the combinations of adjacent time-scale levels (e.g., the combination of f_{i-1} and f_i in Fig. 3 where $i = 2, 3, \cdots, i_{\max}$ and f_1 and $f_{i_{\max}}$ correspond to $f_{\max}$ and $f_{\min}$, respectively). As a result, the correspondence of all the local minima to the candidates is fixed with consistency throughout the entire time-scale range. This allows us to appropriately estimate the intensity of each candidate. It is noted that the intensity of the candidates having no correspondence with the local minima extracted at $f_{\max}$ becomes zero (e.g., C3 in Fig. 3).

After calculating the intensity of all the candidates, we categorize them by their intensity values. In this case, a clustering procedure in which the number of clusters is automatically determined must be used. We adopt Ward's method [14] with Mojena's Stopping Rule One [15] as shown in **Fig. 4**. In Ward's method, candidates are grouped using the squared Euclidean distance $(U_i - U_j)^2$ where U_i and U_j are the intensity values of the ith and jth candidates. Ward's method is known as a method giving good approximations of the optimal grouping by minimizing the total within-cluster variance; this is easily realized by a simple Lance-Williams algorithm when the squared Euclidean distance is used [14]. It is also known that the proper number of clusters can be obtained via simple calculations by combining Ward's

© 2014 Information Processing Society of Japan

method with Mojena's Stopping Rule One[*2].

The candidates categorized into the cluster including a zero-intensity candidate are regarded as those of little significance and deleted; in the case of Fig. 4, C3 and C4, which belong to Cluster 0 including the zero-intensity candidate C3, are deleted. The remaining candidates are finally fixed as the frames giving keyposes.

To sum up, the algorithm to implement the above procedures becomes as follows:

(1) The following parameters are set: the time-scale range $[f_{\min}, f_{\max}]$, the ratio between the cut-off frequencies of adjacent time-scale levels r_f ($0 < r_f < 1$) and the constant k used in Mojena's Stopping Rule One [15].
(2) Keypose candidates giving local minima in $\boldsymbol{V}$ are extracted.
(3) At each time scale f, local minima in $\boldsymbol{F}_{LP}[\boldsymbol{V}, f]$ are extracted and the correspondence of them to the original candidates are fixed. The f value at the ith time scale ($1 \le i \le i_{\max}$) is given as $f = f_{\max} r_f^{(i-1)}$ (regular interval on the logarithmic axis), and $i_{\max}$ is given as the maximum integer satisfying $f_{\max} r_f^{(i_{\max}-1)} \ge f_{\min}$.
(4) The intensity of each candidate is calculated.
(5) All the candidates are categorized and the frames giving keyposes are fixed.

As already mentioned, it is better to make $[f_{\min}, f_{\max}]$ as wide as possible. However, excessively widening the frequency range has to be avoided to prevent the excessive increase of computation time. According to the analysis results in Ref. [17], the upper limit of the tempo of human motion is around 250 to 300 BPM (i.e., over 4 Hz). Therefore, we set $f_{\max} = 8.0$ Hz, considering a twofold margin. As for the lower limit, we adopt $f_{\min} = 0.2$ Hz; this was determined under the condition that the $u(f)$ values decrease to almost zero for most Mocap data streams. We set $r_f = 0.95$; this is obtained as the lowest value which does not cause the breakdown of consistency throughout the entire time-scale range. The constant of Mojena's Stopping Rule One was set as $k = 2.75$ which is the value included in the range recommended in Ref. [15]. The cut-off frequency of the low-pass filter used in Eq. (2) was set as $f_c = 10.0$ Hz; this was determined by considering the noise frequency in Mocap data streams.

4. Experimental Results

4.1 Motion-capture Data Streams Used in the Experiment

We report the experimental results in this section. The Mocap data streams used in the experiment are shown in **Table 1**; 20 data streams selected from five motion categories were used. We selected these categories based on whether criteria to determine the ground-truth keyposes can easily be defined or not. The criteria actually used are shown in **Table 2**, and the numbers of the ground-truth keyposes chosen are shown in Table 1. In addition, we also took the diversity of motion characteristics into account. The data streams belonging to Walk, Punch Sequence and Charleston have the motion sequences consisting of the repetition of simple gestures [19], [20], whereas those of *Nishimonai Bon Odori* and *Tsugaru Jinku* show the choreography including a variety of motion patterns [21]. This tendency can be confirmed by obtaining the motion-speed data streams (given by Eq. (3)[*3]) shown in **Fig. 5**. The diversity of both the time scale of speed variation and speed range can also be recognized.

Table 1 Motion-capture data streams used in the experiment.

Motion category	Index	Length [sec]	Frame rate [fps]	Number of frames	Number of keyposes
Walk	07_01	2.63	120	316	4
	07_02	2.74	120	329	5
	07_03	3.46	120	415	5
	07_06	3.48	120	417	6
Punch Sequence	144_13	16.22	120	1,946	44
	144_14	17.13	120	2,055	44
	144_20	18.92	120	2,270	58
	144_21	16.08	120	1,930	50
Charleston	93_03	3.68	120	442	8
	93_04	4.22	120	506	10
	93_05	4.55	120	546	12
	93_08	4.64	120	557	12
Nishimonai Bon Odori, Ondo (1st verse) (Japanese folkdance)	#1	26.07	30	782	14
	#2	24.37	30	731	14
	#3	23.77	30	713	14
	#4	23.10	30	693	14
Tsugaru Jinku (Japanese folkdance)	#1	6.73	30	202	7
	#2	6.53	30	196	7
	#3	6.47	30	194	7
	#4	6.60	30	198	7

Walk, Punch Sequence and Charleston:
Downloaded from Carnegie-Mellon Mocap Database [18].
***Nishimonai Bon Odori* and *Tsugaru Jinku*:**
Acquired by the authors (measured by MotionStar Wireless™ (Ascension Technology Corporation) with LIBERTY™ (Polhemus) ×2).

Table 2 Criteria to determine the ground-truth keyposes.

Motion category	Ground-truth keypose
Walk	Moment of heel strike (boudary of gait cycle [19]).
Punch Sequence	Arm-extension pose and drawing-back pose.
Charleston	Foot-Up pose, Foot-Forward pose (including Crossed pose), Foot-Backward pose and Kick [20].
Nishimonai Bon Odori	Quoted from the illustration of dance choreography in "*Furusato no Min'yō I*" [21].
Tsugaru Jinku	Quoted from the illustration of dance choreography in "*Furusato no Min'yō I*" [21].

4.2 Keypose-extraction Methods Used for Comparison

We compared the present keypose-extraction method with the methods shown in **Table 3**. They were selected from the three categories of keypose-extraction approaches as mentioned in Section 2; both the CM-EE Speed Analysis method [5][*4] and the Mutual-Information-Measure Analysis method [1] belong to the breakpoint-detection category. As for the Clustering-Based Approach, Liu et al. [8] selected the frame appearing first in each cluster as that giving a keypose; however, we chose the frame closest to the centroid of each cluster since slightly better results were obtained. To use these methods, users have to determine the

[*2] Recently, several new clustering methods automatically giving the number of clusters have been proposed (e.g., Ref. [16]). It may be possible to replace the clustering procedure adopted in this paper by one of these methods as the need arises.

[*3] To eliminate the influence of difference in body constitution among data streams, we normalized the joint-coordinate values used in Eq. (3) by the height of the body.

[*4] In the original CM-EE Speed Analysis method, extracted keyposes are refined by using the sound data of musical accompaniment. However, we omitted this procedure to compare with the other methods in which no sound data is used.

© 2014 Information Processing Society of Japan

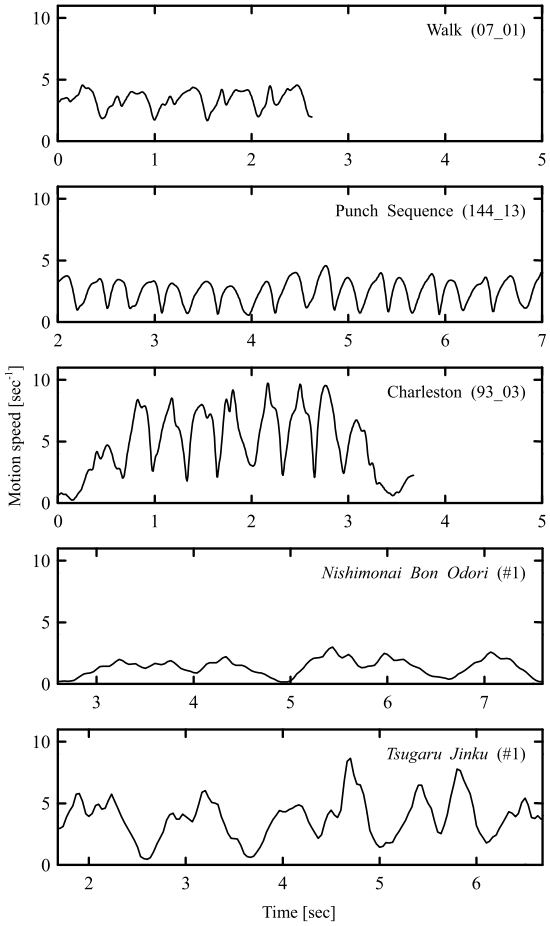


Fig. 5 Examples of motion-speed data streams.

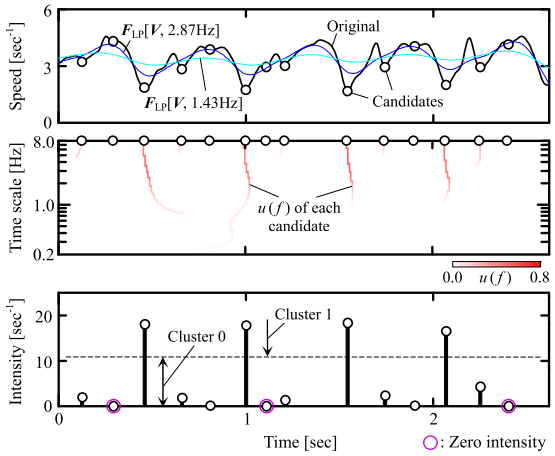
Table 3 Methods used for comparison.

Curve-Simplification Algorithm [7] Evaluation index: · Distance of Mocap curve point from simplified lines. Adjustable parameter: · Error margin (d_e).
Clustering-Based Approach [8] Procedure: · Selecting representative frames from similar-pose clusters. Adjustable parameter: · Threshold of inter-cluster distance (d_c).
CM-EE Speed Analysis [5] Evaluation index: · Local minima of Center-of-Mass- (CM), hands- and feet- (i.e., End Effectors: EE) speeds. Adjustable Parameters: · Minimum- and maximum-speeds of CM (v_{CMmin}, v_{CMmax}). · Minimum- and maximum-speeds of hands (v_{Hmin}, v_{Hmax}). · Minimum motion distance of feet (d_F). · Cut-off frequency of the noise-reduction Gaussian filter (f_c). · Shortest interval between local minima (t_{min}).
Mutual-Information-Measure Analysis [1] Evaluation Index: · Local minima of mutual infomation measure for directional change in motion patterns. Adjustable parameters: · Maximum value of local minima (I_{max}). · Shortest interval between local minima (t_{min}) [22].

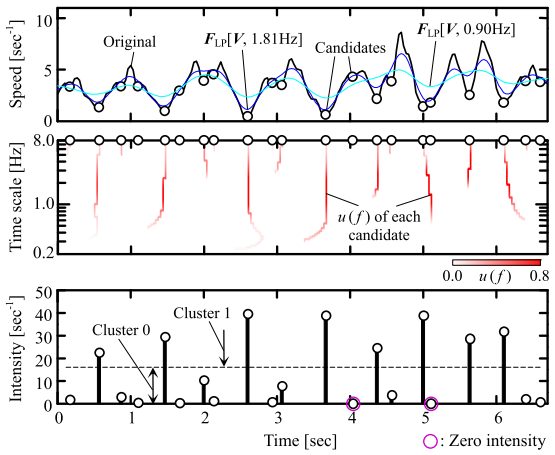
values of the adjustable parameters shown in Table 3. We determined these values at every motion category through the preliminary trial-and-error adjustment process. The obtained values are shown in **Table 4**. It is noted that the adjusted values considerably vary according to motion categories in all the methods.

Table 4 Adjusted parameter values.

Method	Parameter	Motion category				
		Walk	Punch Sequence	Charleston	*Nishimonai Bon Odori*	*Tsugaru Jinku*
Curve Simplification	d_e	2.8	2.0	2.3	3.1	3.4
Clustering	d_c	1.0	1.2	0.9	1.8	2.8
CM-EE Speed	v_{CMmin} [sec⁻¹]	2.0	0.5	0.5	0.3	0.5
	v_{CMmax} [sec⁻¹]	0.0	0.0	0.05	0.05	0.0
	v_{Hmin} [sec⁻¹]	0.5	1.3	1.5	0.4	1.6
	v_{Hmax} [sec⁻¹]	0.0	0.0	0.0	0.2	0.0
	d_F	0.05	0.1	0.1	0.1	0.1
	f_c [Hz]	10	10	10	6	5
	t_{min} [sec]	0.2	0.1	0.08	0.3	0.2
Mutual Information Measure	I_{max}	0.885	0.93	0.89	0.82	0.86
	t_{min} [sec]	0.1	0.1	0.1	0.3	0.3



(a) Walk (07_01).



(b) *Tsugaru Jinku* (#1).

Fig. 6 Examples of keypose-extraction process.

4.3 Keypose-extraction Precess

Prior to presenting the extracted keyposes, we show some examples of the keypose-extraction process in the present method. The cases of Walk (07_01) and *Tsugaru Jinku* (#1) are shown in (a) and (b) of **Fig. 6**, respectively. First, at each time-scale level, a filtered motion-speed curve was obtained and the values of $u(f)$, given by Eq. (4), of each candidate was calculated. For example, the $u(f)$ value on the dark blue curve was given as the speed descent from the light blue curve. Each of the red curves represents the $u(f)$ values and shift of each candidate throughout the entire time-scale range. As for the candidates located at extremely

© 2014 Information Processing Society of Japan

shallow depressions of the original motion-speed data stream, or those each of which is extremely close to another candidate, the red curves showed little extension. On the other hand, all the $u(f)$ values decreased to almost zero at the longest time-scale level (i.e., at 0.2 Hz) in both of (a) and (b). These facts suggest that the characteristics of each candidate were well extracted under the selected time-scale-range condition.

Then, the intensity of each candidate was obtained by adding up its $u(f)$ values, and all the candidates were categorized by their intensity values. In both of (a) and (b) in Fig. 6, candidates were categorized into two clusters. The validity of categorization was confirmed by ANOVA [23]. The null hypothesis was rejected at the 1% significance level in both cases. It was confirmed that the null hypothesis was rejected in the analyses of all the data streams in Table 1. Finally, the candidates categorized into Cluster 0 including zero-intensity candidates were deleted and keypose frames were fixed; four keyposes were extracted from Walk (07_01) and eight from *Tsugaru Jinku* (#1). The extracted keyposes will be presented in the next subsection.

4.4 Results of Keypose Extraction

Examples of extracted keyposes are shown in **Fig. 7** and **Fig. 8**. In Fig. 7 which shows the case of Walk (07_01), all the methods except the Clustering-Based Approach show the results giving relatively good agreement with the ground-truth keyposes. As for the Clustering-Based Approach, decreasing the threshold of inter-cluster distance was needed to obtain keyposes close to the ground-truth ones; this caused the excessive increase of the number of clusters, consequently giving rise to performance degradation. In Fig. 8 which shows the case of *Tsugaru Jinku* (#1), on the other hand, the Mutual-Information-Measure Analysis method shows the best agreement with the ground-truth keyposes (The First Beat which is not shown in Fig. 8 also well agreed). The present method also gave a good result; only a single error keypose caused disagreement with the ground-truth one. The remaining three methods provided worse results; multiple errors occurred in all the cases.

To quantitatively evaluate the results obtained from all the data streams in Table 1, we introduced the evaluation index F_{measure} [24]. In the calculation of F_{measure} for each data stream, only the keyposes extracted from the part of the motion sequences directly related to its motion category were used: e.g., consider only the dance-performance part of Charleston, exclusive of the periods before and after the performance, etc. The parts to which

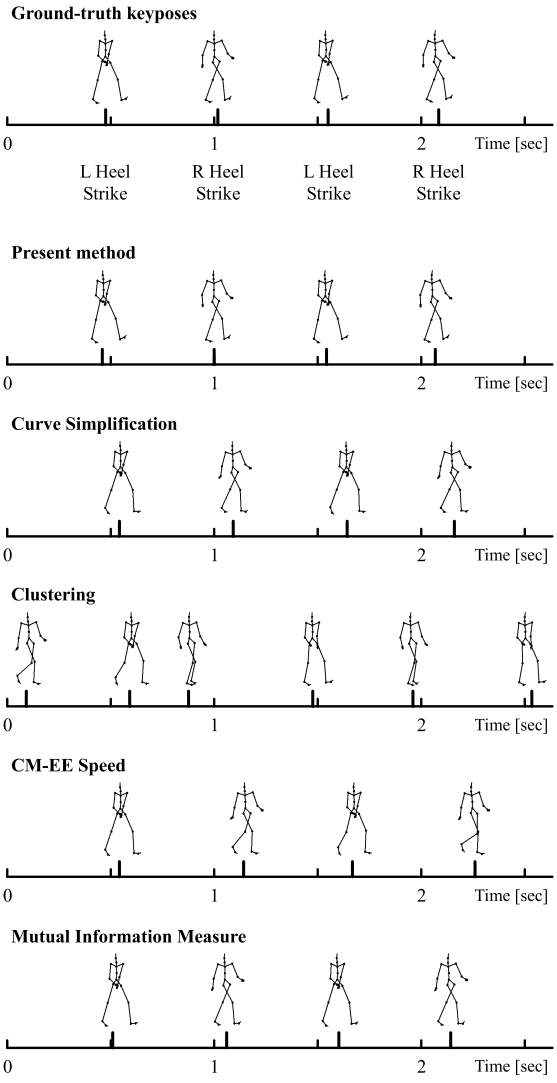


Fig. 7 Keypose extraction from Walk (07_01).

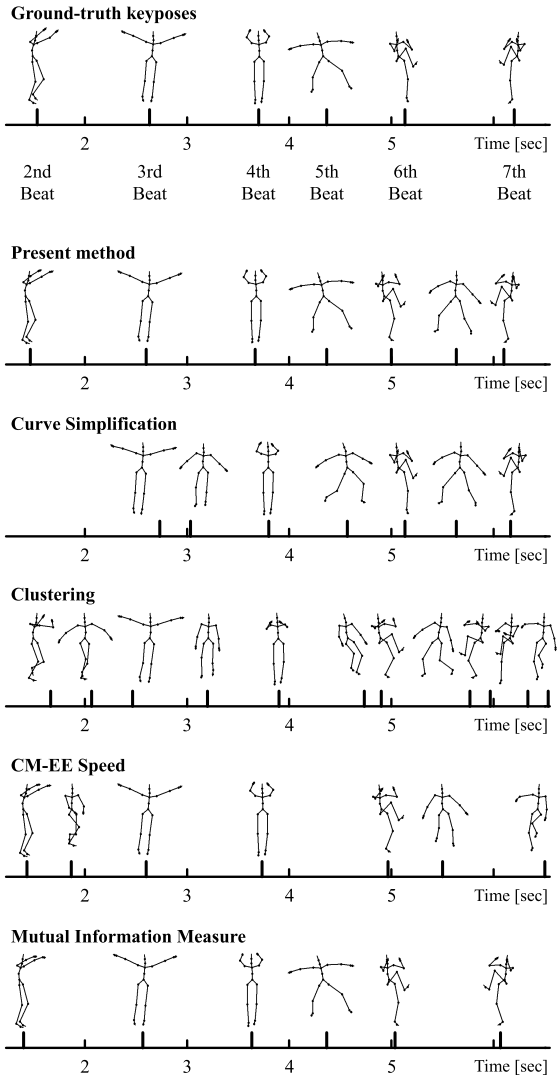


Fig. 8 Keypose extraction from *Tsugaru Jinku* (#1).

© 2014 Information Processing Society of Japan

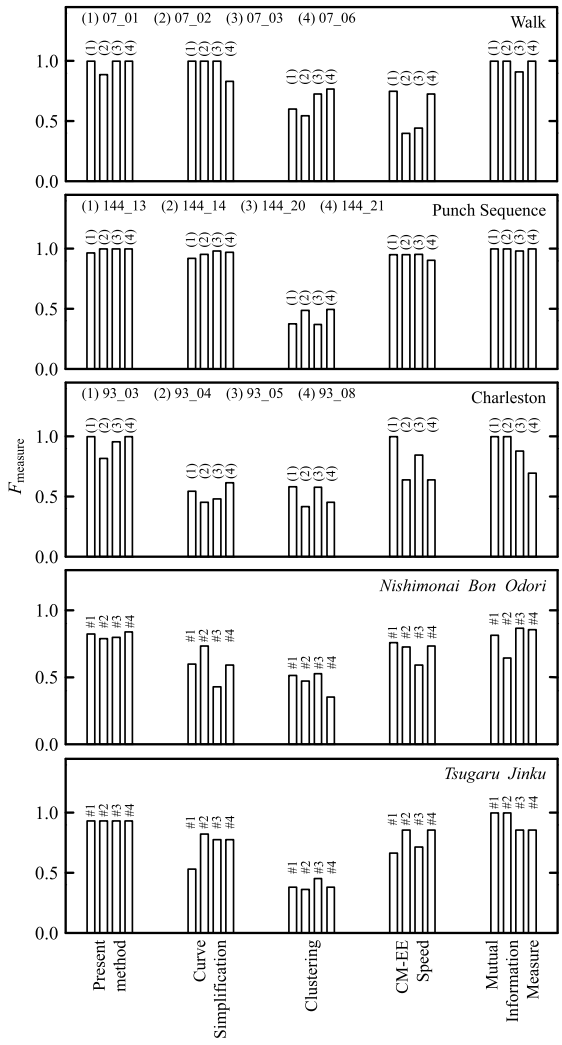


Fig. 9 Evaluation of keypose-extraction results.

the criteria shown in Table 2 are inapplicable were thereby eliminated. We defined a correct keypose as a pose whose time difference from the ground-truth one is shorter than 1/4 of keypose interval.

The resulting data are shown in **Fig. 9**. The present method showed high F_{measure} values in all the motion categories. The Mutual-Information-Measure Analysis method also gave high values as a whole; however, readjusting the parameters at each category was indispensable to achieve this level. On the other hand, both the Curve-Simplification Algorithm and the CM-EE Speed Analysis method provided high values only for specific categories. As for the Clustering-Based Approach, no high F_{measure} value was shown; this is attributed to the fact that the excessive increase of the number of clusters could not have been avoided in all the categories.

4.5 Discussion

As shown in Fig. 9, the present method gave considerably good results. These were given through the keypose-extraction process shown in Fig. 6. This means that the candidates were almost properly rated by the intensity values given by the procedures presented in Section 3. When a user wants to modify an obtained result, therefore, the set of intensity values can be used as a reliable reference in the modification process as mentioned in Section 2.

As for the methods other than the present method, the values of the adjustable parameters shown in Table 3 were prearranged to give good results to every motion category. As already mentioned, the assigned values were different from category to category as shown in Table 4. On the other hand, the parameter values used in the present method, described in Section 3.2, were fixed over all the categories. Nevertheless, the present method gave results comparable to the best of those given by the other methods over all the categories. This suggests the effectiveness of the procedures introduced to adaptively deal with the variation of motion characteristics.

It should also be pointed out, however, that the present method has a limitation; the variation of postures in motion sequences is not considered. The appearance of keyposes sometimes depends on directional change in motion patterns [1]. To detect the moments of directional change, grasping posture variation is indispensable. An example of error caused by this limitation is shown in Fig. 8; the present method extracted an error keypose at the moment between the Sixth and Seventh Beats. Although a relatively large speed descent is observed at this moment, no significant directional change occurs; the Mutual-Information-Measure Analysis method which detects directional change extracted no keypose at this moment. Additional work is needed to address this issue.

5. Conclusions

The main contribution of this paper is the introduction of an adaptive algorithm to the process of extracting keyposes from Mocap data streams; it adaptively extracts keyposes in response to the motion-speed characteristics of a given Mocap data stream. The experimental results showed that the readjustment of the parameters used in the algorithm can be omitted. It is hoped that the present method will help in browsing, editing or reusing Mocap data streams. However, the issue that the posture variation in motion sequences is not considered still remains unresolved. This will be the subject of future work.

References

[1] So, C.K.F. and Baciu, G.: Entropy-Based Motion Extraction for Motion Capture Animation, *Comp. Anim. Virtual Worlds*, Vol.16, pp.225–235 (2005).
[2] Assa, J., Caspi, Y. and Cohen-Or, D.: Action Synopsis: Pose Selection and Illustration, *ACM Trans. Graph.*, Vol.24, No.3, pp.667–676 (2005).
[3] Halit, C. and Capin, T.: Multiscale Motion Saliency for Keyframe Extraction from Motion Capture Sequences, *Comp. Anim. Virtual Worlds*, Vol.22, pp.3–14 (2011).
[4] Bevilacqua, F., Ridenour, J. and Cuccia, D.: 3D Motion Capture Data: Motion Analysis and Mapping to Music, *Proc. Workshop/Symposium on Sensing and Input for Media-Centric Systems* (2002).
[5] Shiratori, T, Nakazawa, A. and Ikeuchi, K.: Detecting Dance Motion Structure through Music Analysis, *Proc. FGR 2004*, pp.857–862 (2004).
[6] Miura, T., Kaiga, T., Matsumoto, N., Katsura, H., Shibata, T., Tajima, K. and Tamamoto, H.: Characterization of Motion Capture Data by Motion Speed Variation, *IEEJ Trans. EIS*, Vol.133, No.4, pp.906–907 (2013).
[7] Lim, I.S. and Thalmann, D.: Key-Posture Extraction Out of Human Motion Data by Curve Simplification, *Proc. 23rd Intl. Conf. IEEE Engineering in Medicine and Biology Society*, Vol.2, pp.1167–1169 (2001).
[8] Liu, F., Zhuang, Y., Wu, F. and Pan, Y.: 3D Motion Retrieval with Mo-

© 2014 Information Processing Society of Japan

tion Index Tree, *Computer Vision and Image Understanding*, Vol.92, pp.265–284 (2003).
[9] Huang, K.S., Chang, C.F., Hsu, Y.Y. and Yang, S.N.: Key Probe: A Technique for Animation Keyframe Extraction, *Visual Comput.*, Vol.21, pp.532–541 (2005).
[10] Liu, X.M., Hao, A.M. and Zhao, D.: Optimization-Based Key Frame Extraction for Motion Capture Animation, *The Visual Computer*, Vol.29, No.1, pp.85–95 (2013).
[11] Hodrick, R.J. and Prescott, E.C.: Postwar U.S. Business Cycles: An Empirical Investigation, *Journal of Money, Credit, and Banking*, Vol.29, No.1, pp.1–16 (1997).
[12] Kaiser, R. and Maravall, A.: Estimation of the Business Cycle: A Modified Hodrick-Prescott Filter, *Span. Econ. Rev.*, Vol.1, pp.175–206 (1999).
[13] Skiena, S.S.: *The Algorithm Design Manual*, 2nd ed., Springer (2008).
[14] Gan, G., Ma, C. and Wu, J.: *Data Clustering: Theory, Algorithms, and Applications*, SIAM (2007).
[15] Mojena, R.: Hierarchical Grouping Methods and Stopping Rules: An Evaluation, *The Computer Journal*, Vol.20, No.4, pp.359–363 (1977).
[16] Ishioka, T.: An Expansion of X-Means for Automatically Determining the Optimal Number of Clusters, *Proc. IASTED Intl. Conf. on Computational Intelligence*, pp.91–96 (2005).
[17] Xu, J., Takagi, K. and Yoneyama, A.: Beat Induction from Motion Capture Data Using Short-Term Principal Component Analysis, *ITE Journal*, Vol.64, No.4, pp.577–583 (2010).
[18] Carnegie-Mellon Mocap Database, available from ⟨http://mocap.cs.cmu.edu/⟩.
[19] The Pathokinesiology Service & The Physical Therapy Department, ed.: *Observational Gait Analysis*, Rancho Los Amigos National Rehabilitation Center (2001).
[20] Walker, B.: *How to Charleston Correctly*, Dance Books Ltd. (2009) (Reprint of the edition published by Great Northern Publishing, Co. (1926)).
[21] National Folk Dance Federation of Japan, ed.: *Furusato no Min'yō I*, National Folk Dance Federation of Japan (2007) (in Japanese).
[22] Tilmanne, J., Hidot, S. and Ravet, T.: MOCKEY: Motion Capture as a Tool for Keyframing Animation, *QPSR of the Numediart Research Program*, Vol.2, No.4, pp.119–124 (2009).
[23] Johnson, R.A. and Wichern, D.W.: *Applied Multivariate Statistical Analysis*, 6th ed., Pearson Education, Inc. (2007).
[24] Tan, P.N., Steinbach, M. and Kumar, V.: *Introduction to Data Mining*, Addison-Wesley (2005).



Takeshi Miura received his D.Eng. degree in electrical engineering from Hokkaido University in 1998. He is currently an associate professor in the Department of Electrical and Electronic Engineering, Graduate School of Engineering and Resource Science, Akita University.



Takaaki Kaiga received his M.E. degree in mechanical engineering from Ibaraki University in 1995. Since 1996, he has been with the computer division of Digital Art Factory, Warabi-za Co., Ltd.



Hiroaki Katsura is currently a professor in the Department of Music Education, Course of School Subject Teaching, Program of School Education, Faculty of Education and Human Studies, Akita University.



Katsubumi Tajima received his D.Eng. degree in electrical engineering from Tohoku University in 1998. He is a professor in the Cooperative Major in Life Cycle Design Engineering, Graduate School of Engineering and Resource Science, Akita University.



Takeshi Shibata received his D.Eng. degree in electrical engineering from Akita University in 2012. He is currently a post-doctoral researcher in the Venture Business Laboratory, Akita University. His research interests include virtual reality technique and archiving and handing-down technique for traditional folk dances.



Hideo Tamamoto received his D.Eng. degree in electrical engineering from the University of Tokyo in 1976. He is currently an executive vice president of Akita University. His research interests include design-for-testability of logic circuits, archiving and handing-down technique for traditional folk dances, and e-learning system.

© 2014 Information Processing Society of Japan