SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ DERGİSİ

Sakarya University Journal of Science
SAUJS

ISSN 1301-4048 e-ISSN 2147-835X Period Bimonthly Founded 1997 Publisher Sakarya University
http://www.saujs.sakarya.edu.tr/

Title: Keyframe Extraction Using Linear Rotation Invariant Coordinates

Authors: Hasan MUTLU, Ufuk ÇELİKCAN

Recieved: 2022-07-25 00:00:00

Accepted: 2022-09-05 00:00:00

Article Type: Research Article

Volume: 26
Issue: 5
Month: October
Year: 2022
Pages: 2040-2051

How to cite
Hasan MUTLU, Ufuk ÇELİKCAN; (2022), Keyframe Extraction Using Linear Rotation Invariant Coordinates. Sakarya University Journal of Science, 26(5), 2040-2051,
DOI: 10.16984/saufenbilder.1148511
Access link
http://www.saujs.sakarya.edu.tr/en/pub/issue/73051/1148511

Sakarya University Journal of Science 26(5), 2040-2051, 2022

Keyframe Extraction Using Linear Rotation Invariant Coordinates

Hasan MUTLU*[1], Ufuk ÇELİKCAN[1]

Abstract

Keyframe extraction is a widely applied remedy for issues faced with 3D motion capture -based computer animation. In this paper, we propose a novel keyframe extraction method, where the motion is represented in linear rotation invariant coordinates and the dimensions covering 95% of the data are automatically selected using principal component analysis. Then, by K-means classification, the summarized data is clustered and a keyframe is extracted from each cluster based on cosine similarity. To validate the method, an online user study was conducted. The results of the user study show that 45% of the participants preferred the keyframes extracted using the proposed method, outperforming the alternative by 6%.

Keywords: Keyframe extraction, linear rotation invariant coordinates, motion data summarization

1. INTRODUCTION

In today's world, motion capture technology is used in many areas, especially in movies and video games. At the same time, editing and transmission of motion capture data are still difficult due to large data sizes. Hence, representing motion capture data compactly continues to be a vital consideration of research.

Skeletal animation is the most effective and commonly used technique of exploiting motion capture data. Skeletal animation consists of two parts, a mesh and a hierarchical set of bones. The mesh part contains surface (skin) information of the character to be rendered, while animation is realized with the spatio-temporal information by the latter. As skeletal animation is performed, the technique fills the gap between two keyframes with interpolation on the timeline. Although skeletal animation provides a solution to represent motions compactly, frame counts remain problematically large for processing, storing and editing. As a remedy, keyframe extraction has been a widely applied solution for issues faced in motion capture -based skeletal animation.

A keyframe extraction method must be capable of sorting out significant keyframes from the others. Also, to improve the success rate of the solution, deriving the characteristics of vertices

* Corresponding author: hasanmutlu9@gmail.com
[1] Hacettepe University
ORCID: https://orcid.org/0000-0001-8686-6988, https://orcid.org/0000-0001-6421-185X
E-mail: ufuk.celikcan@gmail.com

concerning both the vertex itself and its neighbor vertices is important. For these reasons, we argue that representing vertices of joints in alternative coordinate systems and processing the motion data accordingly can provide a better solution.

In this article, we propose a novel keyframe extraction approach. In our approach, we represent the joints in the frames with linear rotation invariant (LRI) coordinates [1], apply principal component analysis (PCA) [2] to reduce the data dimension and extract summary data of each keyframe. Then, we divide them into clusters with the K-means algorithm [3] and select keyframes according to cosine similarity concerning adjacent keyframes. Also, we examine the performance effect of LRI transformation on our method against using regular Cartesian coordinates without LRI.

The structure of this paper is as follows: Section 2 gives a review of prior work on the subject matter. After that, we detail our solution and provide experimental results collected with our online user study in Sections 3 and 4, respectively. Finally, Section 5 concludes the paper.

2. RELATED WORK

There have been a number of different approaches for keyframe extraction. These previous methods either convert the motion data into various spaces, use motion/frame data as trajectory/motion curves, apply clustering algorithms, handle a matrix factorization problem, or solve a kind of machine learning problem with a genetic algorithm.

Representing skeletal animations in different spaces can provide the facility to determine the difference between frames. The method by Kapadia et al. [4] indexes the motion data in a trie-based structure according to structural, geometrically, and dynamic features. This trie-based structure contains most salience keyframes of the animation. Jin et al.'s method [5] focuses on determining the saliency of the frames. The method computes the saliency of each frame and selects groups from these frames. After this step, the solution uses a non-linear dimension reduction algorithm and extracts keyframes. The method proposed by Voulodimos et al. [6] creates physics-based temporal summaries and determines different keyframes, while Sapinski et al.'s method [7] defines a new representation using the spatial location and orientation of the keyframe joints and selects keyframes from this representation. Xia et al. [8] defines a joint kernel sparse representation, and the algorithm determines the sparseness of the frames and then decides keyframes according to the calculated sparseness value. Choensawat et al. [9] introduced an algorithm named GENLABAN by which they calculate a score for each frame by analyzing body motion, body postures, and weight of the body parts. With these scores, the algorithm extracts keyframes.

Solutions based on trajectory or motion curves convert skeletal animation to a curve and then apply their algorithms to this curve. Miura et al. [10] combines curve-simplification and Bayesian information criterion to extract keyframes from given motion capture data. After the algorithm generates the motion curve, the method divides the curve into two segments at the point most distant from the straight line connecting the endpoints. For the calculated error between the curve and simplified line, the method uses the Bayesian information criterion to select keyframes. Bulut and Capin [11] defined a metric named curve saliency. The solution detects salient parts of the curve and uses Gaussian weighted average value distribution to select keyframes. In the method by Togawa and Okuda [12], after the joints in the animation are converted into curves, the algorithm calculates the cost value for all frames and conducts elimination of frames accordingly. These steps repeat until the most important keyframes remain. The algorithm by Yang et al. [13] applies Butterworth filtering and PCA to the input data and then selects keyframes with zero-crossing points of velocity. Zhang et al.'s method [14] creates motion curves from the amplitudes of motion of joints, applies PCA, defines a distance characteristic curve, and eventually uses this curve to extract keyframes. The method proposed by Halit and Capin [15] defined a metric named 'motion saliency'. With this metric, their method analyzes the motion curve of the animation and extracts keyframes.

In contrast, clustering-based approaches convert skeletal animation into a different dimension and handle the task as a type of shortest-path problem. One such method by Roberts et al. [16] simplifies the motion frames by around 10% while retaining most of its detail. The method considers each frame as a node in a weighted graph and calculates the weights of the graph with the perpendicular distance between joint positions in each node (frame). After these calculations, the algorithm selects the nearest N keyframes according to weights. Sun et al.'s method [17] defines the inter-frame similarity metric based on a group of motion joints and uses affine propagation clustering to extract keyframes. Qiang Zhang et al.'s method [18] uses an unsupervised clustering algorithm to divide frames into two classes by similarity distances and, in the last step, uses dynamic clustering ISODATA to centralize similar frames and eliminate them.

Matrix factorization solutions represent given skeletal animation data as matrices. The algorithm by Huang et al. [19] provides a solution handled as a constrained matrix factorization problem with a least-squares optimization technique. This method represents the animation as matrices that contain key weights and non-keyframe weights. The algorithm uses these two matrices to extract keyframes according to user-specified error tolerance iteratively.

Machine learning solutions often use genetic algorithms to determine keyframes from the animation data. For instance, the method by Zhang et al. [20] uses a multiple-population-based genetic algorithm and defines a fitness method to meet minimizing the reconstruction error to select keyframes. Liu et al.'s method [21] uses genetic optimization algorithms and calculates the sparseness of the frames for determining keyframes.

3. METHOD

Although our method uses a clustering approach, unlike other solutions, it applies LRI and PCA methods before the clustering process. Applying LRI and PCA algorithms summarizes the characteristic information of each keyframe. Also, our solution makes use of the cosine similarity measure to estimate similarity between summarized keyframes.

Our proposed method consists of two main steps. The first step comprises representing skeletal motion frames in LRI local frames and dimension reduction by applying PCA. At the end of the first step, we get summarized data for each frame in the motion data. For the second step, we divide obtained data from the first step into clusters with the K-means algorithm. Then we use cosine similarity to determine the selected keyframe for each cluster.

In the following, $A = \left(F_{1,}F_{2,}, F_{3,}, \dots, F_k\right)$ defines a skeletal motion where k is the keyframe count of the motion and F defines a keyframe of a skeletal motion such that $F_i = (j_1, j_2, j_3 \dots, j_n)$, $j_n \in R^3$ where n is the number of joints j in the skeleton model so that F_i defines the set of joint positions for the i^{th} keyframe of the given motion.

As mentioned above, our solution *O(A)* outputs $(C_1^1, C_2^1, C_3^1, C_n^1, C_1^2, C_2^2, C_3^2, \ldots, C_{n2}^2, \ldots, C_{nm}^m)$ where *O* applies LRI, PCA, and K-means algorithms, respectively over *A*. *C* defines a cluster in the result that contains summarized data for each keyframe in the same order after applied LRI conversion and PCA algorithm. *m* is the total number of clusters. n_i is the element count of the i^{th} cluster. Accordingly, n_m is the element count of the related extracted cluster.

After obtaining clusters from the first step, we use cosine similarity *S* as a measure of detecting similarity between two summarized keyframes for the clusters of summarized keyframes as follows.

$$S(X,Y) = \frac{\sum_{n=1}^{3} X_n \times Y_n}{\sqrt{\sum_{n=1}^{3} X_n^2} \times \sqrt{\sum_{n=1}^{3} Y_n^2}} \qquad (1)$$

Our algorithm selects a keyframe from the obtained cluster iteratively. To accomplish that, we define two vectors for each iteration. The first vector is the difference between the candidate summarized frame data and the previous one. The second vector is the difference between the next one and the candidate summarized frame data. With these two vectors, our algorithm gathers information about the motion changes. If these vectors are similar, that means these frames are also similar. For this reason, initially, the algorithm determines the second summarized keyframe C_m^2 as the first candidate keyframe where m is the iterating cluster in the algorithm and calculates two vectors using that. The first one of these is $v_1 = C_m^i - C_m^{i-1}$ where i is the iterating (candidate) summarized keyframe. The equation gives the difference between the candidate summarized keyframe and the previous one. The second one $v_2 = C_m^{i+1} - C_m^i$ is the difference between next one and candidate summarized keyframe. With these two vectors, the first similarity value σ initialized by using the Equation 1 above as

$$\sigma = S(v_1, v_2) \qquad (2)$$

and the selected pose *sp* is initialized as 2.

After this initialization, σ will be updated when the new similarity in the processed iteration is less than the current value. The algorithm tries to find the keyframe that has the least similarity with the rest iteratively, as follows.

$$(sp, \sigma) = \begin{cases} sp = i, \sigma = S(V_{i-1}^c, V_{i+1}^c), & if\ S(V_{i-1}^c, V_{i+1}^c) \leq \sigma \\ resume & , otherwise \end{cases} \qquad (3)$$

In this equation, V_{i-1}^c defines the vector difference between the current summarized keyframe in the iteration and the previous one in the cluster *c*. Similarly, V_{i+1}^c defines the vector difference between the current summarized keyframe in the iteration and the next one in the related cluster *c*.

3.1. Representing Motion as LRI Local Frames

As a representation, LRI defines a separate local frame for each vertex, where the discrete forms encode the relationship and change between adjacent local frames. A local frame contains all characteristic properties of the vertex it belongs to and encodes properties relative to the neighboring vertices.

LRI defines two discrete forms. The first discrete form is for the projections of the neighboring vertices into the tangent plane of the vertex. It also denotes lengths of the projected edges on the tangent plane and signed angles between every two adjacent projected edges. The first discrete form provides invariability for positions of vertices, but it lacks information in the normal direction of neighboring vertices. For this reason, LRI also provides a second discrete form. The second discrete form can be considered as a function that defines height distances from vertex to tangent plane. LRI calculates unit vectors as the differences of these discrete forms of neighbor vertices. In the last step, LRI uses only the coefficients of this calculation to represent meshes.

The critical feature of the LRI representation is that the vertices of a given mesh are represented in relative coordinates using these specified local frames. Because this relative definition contains no global information about the mesh's location or orientation, it also ensures invariance under rigid transformations.

For the discrete equations, vertices are denoted by x^i, their corresponding positions in R^3 are denoted by $\hat{x}^i$. The edge towards the k^{th} neighbor of i is x_k^i. Mesh edges in R^3 are denoted by $\hat{x}_k^i$, and their projection onto the tangent plane by $\tilde{x}_k^i$. Each vertex and their neighboring vertices are parameterized as U_i and triangles are denoted by $\triangle_k^i$ for defined set U_i consisting of vertices x^i, x_k^i and x_{k+1}^i. The first discrete form uses the standard inner product of triangles corresponding in the tangent plane T_iM. Let $\mu = \mu_1 x_k^i + \mu_2 x_{k+1}^i$ be a vector in $\triangle_k^i$. Here, μ_1 and μ_2 are the vector components of the defined triangle. According to this equation, μ becomes the diagonal vector of the triangle.

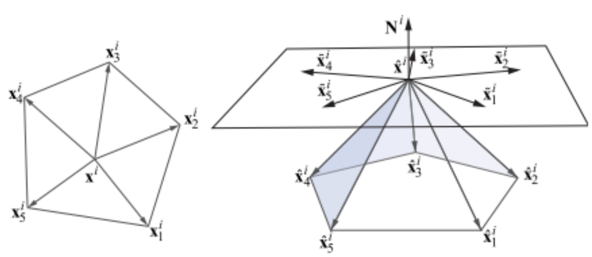

Figure 1 Representing 1-ring neighborhood mesh and tangent plane

The first discrete form equation is given as

$$\tilde{I}(.): \bigcup_{k1}^{d_i-1} \triangle_k^i \to R. \tag{4}$$

where

$$\tilde{I}(\mu) = \langle \mu, \mu \rangle_{R^3} = \langle \mu_1 \widetilde{x_k^i} + \mu_2 \widetilde{x_{k+1}^i}, \mu_1 \widetilde{x_k^i} + \mu_2 \widetilde{x_{k+1}^i} \rangle_{R^3} = \mu_1^2 \widetilde{g_{k,k}^i} + 2\mu_1\mu_2 \widetilde{g_{k,k+1}^i} + \mu_2^2 \widetilde{g_{k+1,k+1}^i} \tag{5}$$

and the second discrete form equation is given as

$$\widetilde{II}(.): \bigcup_{k1}^{d_i-1} \triangle_k^i \to R. \tag{6}$$

where

$$\widetilde{II^i}(\mu) := \mu_1 \langle \widehat{x_k^i}, N^i \rangle_{R^3} + \mu_2 \langle \widehat{x_{k+1}^i}, N^i \rangle_{R^3} = \mu_1 \widetilde{L_k^i} + \mu_2 \widetilde{L_{k+1}^i} \tag{7}$$

in which the coefficients $\tilde{L} = \langle \widetilde{x_k^i}, N^i \rangle_{R^3}$ and N^i is the normal of the vertex i in the tangent plane. LRI defines the local frame with a triplet b_1^i, b_2^i, N^i using these two discrete forms, where $b_1^i \in T_iM$ is a unit vector parallel to $\widetilde{x_1^i}$, b_2^i is a unit vector orthogonal to $\widetilde{x_1^i}$ and δ is the difference operator on the discrete frame vectors:

$$\delta_j(b_1^i) = b_1^j - b_1^i \tag{8}$$
$$\delta_j(b_2^i) = b_2^j - b_2^i \tag{9}$$
$$\delta_j(N^i) = N^j - N^i \tag{10}$$

Finally, the discrete local frame equations

$$\delta_j(b_1^i) = \Gamma_{j,1}^{i,1} b_1^i + \Gamma_{j,1}^{i,2} b_2^i + A_{j,1}^1 N^i \tag{11}$$
$$\delta_j(b_2^i) = \Gamma_{j,2}^{i,1} b_1^i + \Gamma_{j,2}^{i,2} b_2^i + A_{j,2}^1 N^i \tag{12}$$
$$\delta_j(N^i) = \Gamma_{j,3}^{i,1} b_1^i + \Gamma_{j,3}^{i,2} b_2^i + A_{j,3}^1 N^i \tag{13}$$

As previously stated, LRI representation defines local frames that filter out global positions and rotations from the mesh. We use these local frames in our solution. Since the LRI method is defined for meshes, in the first step of our solution, we transform skeleton data for each frame into a 1-ring neighborhood mesh. We assume that each joint position of the skeleton data is a vertex of a 1-ring neighborhood mesh (see Figure 1). After that, we apply LRI to this assumed mesh to extract LRI local frames for each vertex. LRI local frames are a matrix that consists of 9 values and encode characteristic properties of the related vertex and relation between neighborhood vertices. We construct a matrix whose dimension is 9 times the total number of employed joints for each keyframe. Although the extracted local frames are enough to detect similarity between adjacent vertices, we apply dimension reduction by PCA to all keyframe matrices. This way, PCA provides to eliminate the sparse density of the matrices, improving the performance of selecting keyframes computing and obtaining more meaningful data (Figure 2)

In the dimension reduction step, instead of representing LRI data in fixed number of dimensions, our algorithm uses representations of dynamically changing dimensions. This implies that the dimensionality adapts to the given motion. This is carried out according to principles of the PCA method with the condition that the sum of them explains the given LRI data with at least 95% accuracy. Our tests show that between 4 and 12 dimensions are sufficient to explain LRI data with at least 95% accuracy, in general. After these steps, the processed data can be used for extraction and reduction operations.

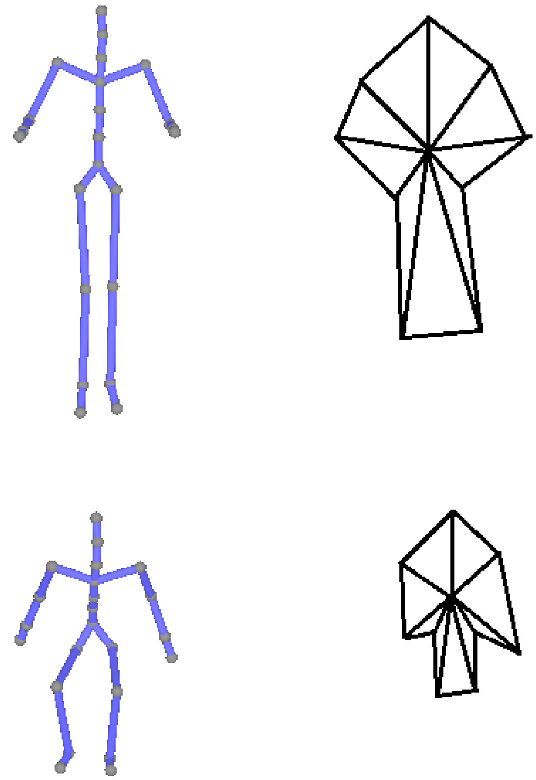

Figure 2 Skeleton data representation as a 1-ring neighborhood mesh

3.2. Clustering and Reducing Keyframes

Our approach uses the K-means classification algorithm for clustering and cosine similarity to measure similarity between keyframes. Firstly, we apply the K-means clustering algorithm to the dimensionally reduced the data (Figures 3 and 4). Our method dynamically clusters up to the desired number of keyframes and calculates the cosine similarity between sequential candidate keyframe changes. Thus, our method selects a keyframe that has the minimum similarity value relative to the rest of the cluster values for each cluster. As a result, the most different keyframes are selected.

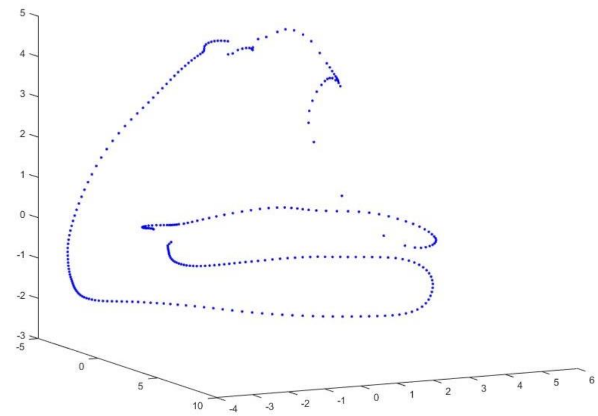

Figure 3 Motion data representation after LRI and PCA steps are applied. The graph shows the distribution of the frames in the motion in 3D space

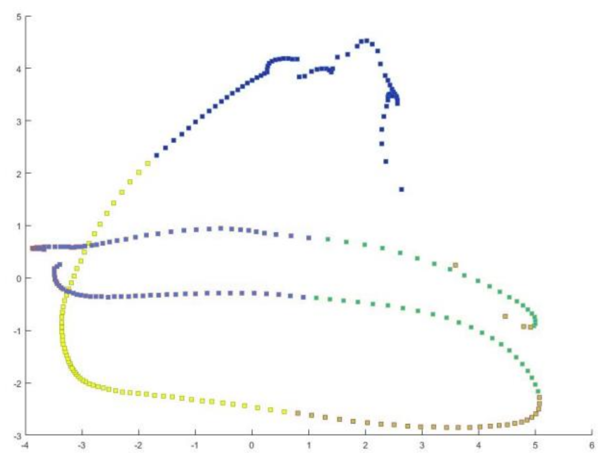

Figure 4 Clustering result of the motion after K-Means applied. This graph contains the distribution of the motion as divided into 5 clusters. Each color (yellow, blue, gold, green, and purple) in the graph represents a distinct cluster. Accordingly, one keyframe for each cluster will be extracted

4. RESULTS

In this study, we used the HDM05 dataset for testing and evaluation of our proposed method. HDM05 is a royalty-free motion capture dataset created by Müller et al. [22]. It features more than 70 motion classes in ten to fifty realizations performed by a variety of performers. The sampling rate of performances in the dataset is 120 Hz. For evaluation, we selected 10 relatively short motions (Table 1) from the dataset and used our method to extract five keyframes from each motion. Figure 5 demonstrates sample sets of keyframes extracted from three of these motions using our proposed method.

Furthermore, as our study results can be subjective, we prepared a website to survey subjective performance evaluations of the participants comparing the results obtained by our method using LRI representations to the ones without. The survey included the ten pre-selected motions under consideration with two sets of five extracted keyframes for each motion, one set including the results of our method and the other including the results obtained using the standard Cartesian coordinate representation. Figures 6 and 7 demonstrate sample results for cartwheel and punch motions in both representations.

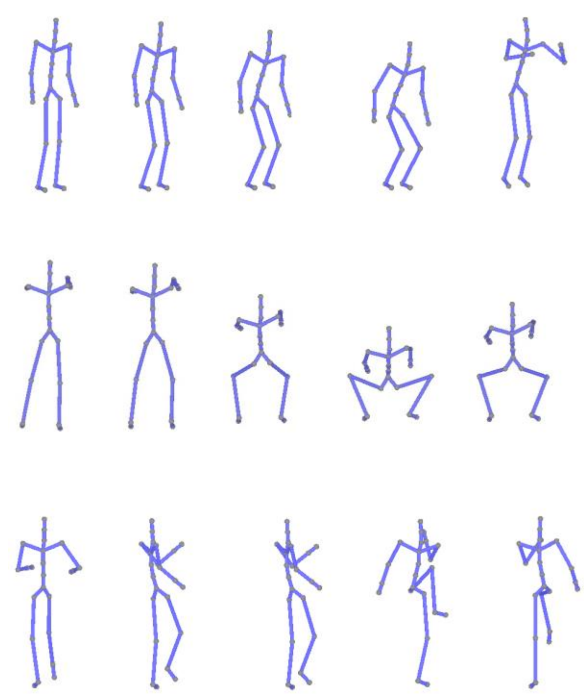

Figure 5 Sample sets of keyframes extracted using our method. Each row illustrates a set of 5 keyframes extracted from jumping, squat and knee to elbows motions, respectively

With the survey, only the gender and age information were collected from the participants, remaining otherwise anonymous. All participants volunteered to take the online survey and none of them have been compensated in any way.

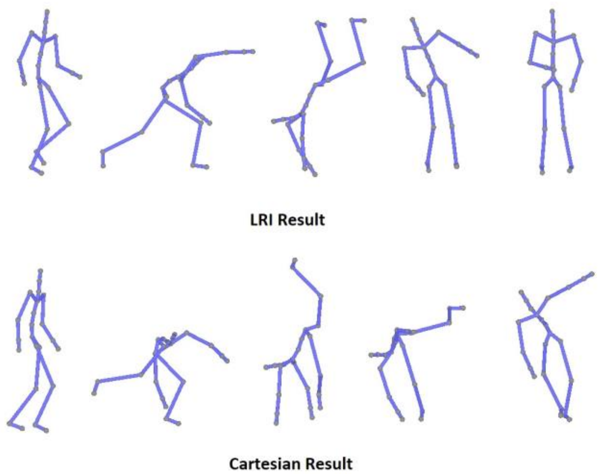

Figure 6 Extracted keyframes from cartwheel motion

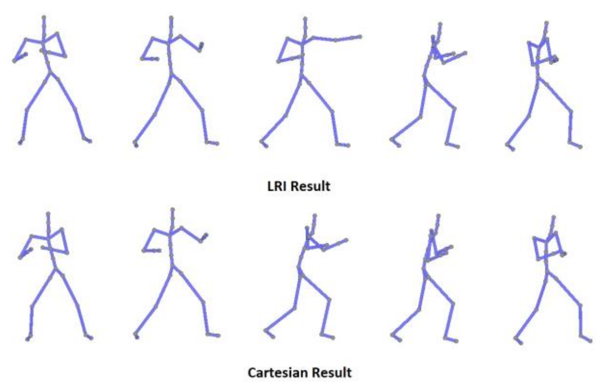

Figure 7 Extracted keyframes from punch motion

Table 1 The motions used in the experiment and their corresponding frame counts

#	Motion	Frame Count
1	Cartwheel	401
2	Elbow to Knee	319
3	Jump Down	272
4	Jumping Jack	142
5	Kick	295
6	Lie Down	621
7	Punch	115
8	Squat	191
9	Throw Basketball	452
10	Throw Ball	427

The survey procedure took place as follows. When a participant visited our online survey website, they were first briefly informed about the study and their gender and age information were collected at this step (Figure 8). Then, the participant started the evaluation of the results. For each motion queried, the participant initially watched the motion in a skeletal animation twice as given in Figure 9. Next, each set of extracted keyframes from the original motion were shown to the participant where the order of the two sets were randomized (Figure 11). Afterwards, the participant watched the original motion with the sets of extracted keyframes shown flanking the original motion on each side as in Figure 12 so that the participant could further assess the differences between the two sets of results. On this page, the participant responded by choosing either of the sets as the best representation of the original motion or 'none' if no significant difference was observed. This is repeated until the participant registered their responses for all 10 motions.

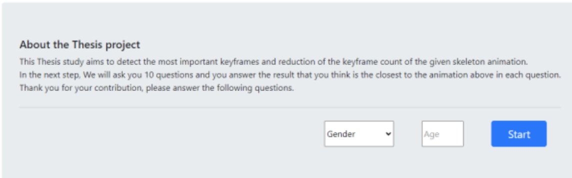

Figure 8 Test step where a participant is informed about the study and reports their gender and age

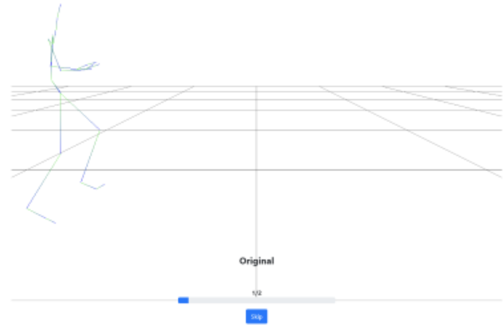

Figure 9 Sample preview of an original motion as shown to the participants with the online survey interface

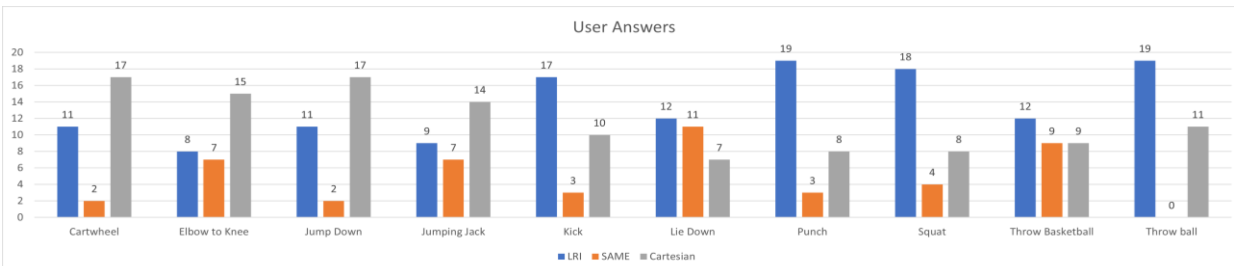

Figure 10 Distribution of participants' preferences per each motion queried. The order of the motions in the figure is the same as in Table 1

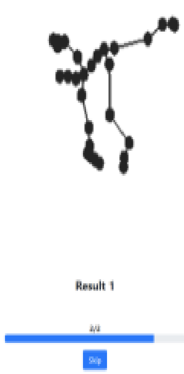

Figure 11 Sample instance where the participant is shown one of the extracted keyframes with the online survey interface

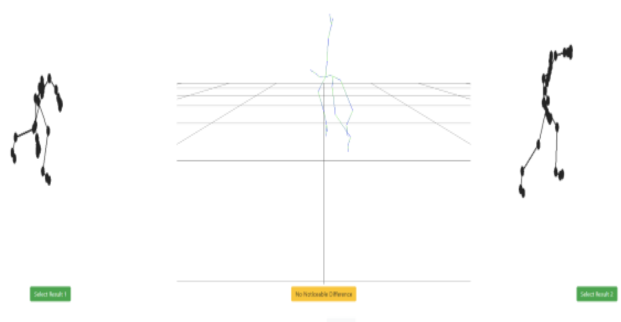

Figure 12 Sample preview of the online survey instance where an original motion is shown to the participant with extracted keyframes of the two alternatives shown consecutively on each side of it

A total of 30 people, 12 female (40%) and 18 male (60%), participated in this study. The average age of the participants was 28 $\pm$ 3.14. Evaluation results are provided in Figures 10 and 13. Figure 10 gives participants' preferences for each motion, while Figure 13 gives the overall distribution of all participants' preferences.

The survey outcomes per motion as shown in Figure 10 demonstrate that the participants preferred mostly the sets of keyframes extracted using the Cartesian coordinate system representation for the first four motions under consideration. However, the keyframes obtained using the LRI representation were preferred by more participants for the remaining six motions. Over the whole set of 10 motion queries, the average preference ratio of the LRI results was 13.6% (±4%) while the average for the alternative was 11.6% (±3.64%).

The aggregated preference results given in Figure 13 show that the participants preferred the keyframes extracted by our proposed approach more than the standard alternative by 6% in general while 16% of the votes indicated no preference.

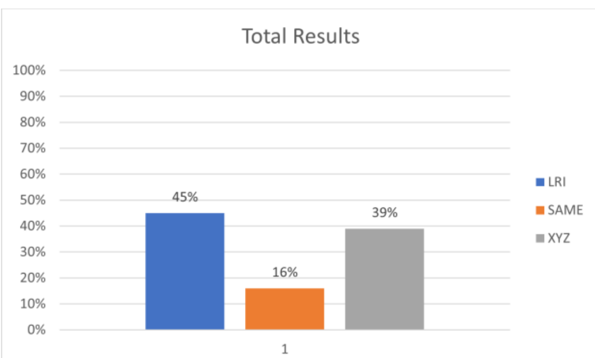

Figure 8 Overall distribution of all participants' preferences collected with the online survey

5. CONCLUSION

We have presented a keyframe extraction method based on LRI coordinates representation and evaluated the method's performance against the ones based on Cartesian coordinates. Our study results underline the potential of LRI for keyframe extraction such that using our solution based on it outperforms the standard representation with a slightly better performance (6%).

This work shows that representing animation keyframes using alternative representations, such as LRI, can provide better extraction performance for skeletal animations. For future work, it is possible to combine our LRI based approach with deep learning methods such as using transformer networks. Our method can use different similarity measures rather than the cosine similarity or it can be combined with other keyframe extraction methods such as curve simplification or matrix factorization approaches towards achieving better performance. Our proposed method can be used for keyframe reduction and compression purposes, as well.

Acknowledgments

The authors would like to thank all volunteers who participated in the online survey anonymously.

Funding

The authors have not received any financial support for the research, authorship, or publication of this study.

Authors' Contribution

The authors contributed equally to the study.

The Declaration of Conflict of Interest/ Common Interest

The authors declare no conflict of interest or common interest.

The Declaration of Ethics Committee Approval

This study does not require ethics committee permission or any special permission.

The Declaration of Research and Publication Ethics

The authors of the paper declare that they comply with the scientific, ethical and quotation rules of SAUJS in all processes of the paper and that they do not make any falsification on the data collected. In addition, they declare that Sakarya University Journal of Science and its editorial board have no responsibility for any ethical violations that may be encountered, and that this study has not been evaluated in any academic publication environment other than Sakarya University Journal of Science.

REFERENCES

[1] Y. Lipman, O. Sorkine, D. Levin, D. Cohen-Or, "Linear rotation-invariant coordinates for meshes" ACM Transactions on Graphics, vol. 24, no. 3, p. 479–487, jul 2005.

[2] A. Mackiewicz, W. Ratajczak, "Principal components analysis (pca)" Computers & Geosciences, vol. 19, no. 3, pp. 303–342, 1993.

[3] S. Lloyd, "Least squares quantization in pcm," IEEE Transactions on Information Theory, vol. 28, no. 2, pp. 129–137, 1982.

[4] M. Kapadia, I.-k. Chiang, T. Thomas, N. I. Badler, J. T. Kider, "Efficient motion retrieval in large motion databases," in Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, ser. I3D '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 19–28.

[5] C. Jin, T. Fevens, S. Mudur, "Optimized keyframe extraction for 3d character animations," Computer Animation and Virtual Worlds, vol. 23, no. 6, pp. 559–568, 2012.

[6] A. Voulodimos, I. Rallis, N. Doulamis, "Physics-based keyframe selection for human motion summarization," Multimedia Tools and Applications, vol. 79, no. 5, pp. 3243–3259, 2020.

[7] T. Sapinski, D. Kaminska, A. Pelikant, G. Anbarjafari, "Emotion recognition from skeletal movements," Entropy, vol. 21, no. 7, 2019.

[8] G. Xia, H. Sun, X. Niu, G. Zhang, L. Feng, "Keyframe extraction for human motion capture data based on joint kernel sparse representation," IEEE Transactions on Industrial Electronics, vol. 64, no. 2, pp. 1589–1599, 2017.

[9] W. Choensawat, M. Nakamura, K. Hachimura, "Genlaban: A tool for generating labanotation from motion capture data," Multimedia Tools and Applications, vol. 74, no. 23, pp. 10823-10846, 2015.

[10] T. Miura, T. Kaiga, N. Matsumoto, H. Katsura, K. Tajima, H. Tamamoto, "Application of the bayesian information criterion to keyframe extraction from motion capture data," in SIGGRAPH Asia 2011 Posters, ser. SA '11. New York, NY, USA: Association for Computing Machinery, 2011.

[11] E. Bulut, T. Capin, "Key frame extraction from motion capture data by curve saliency," CASA, 2007.

[12] H. Togawa, M. Okuda, "Position-based keyframe selection for human motion animation," in 11th International Conference on Parallel and Distributed Systems (ICPADS'05), vol. 2, 2005, pp. 182–185

[13] Y. Yang, L. Zeng, H. Leung, "Keyframe extraction from motion capture data for visualization," in 2016 International Conference on Virtual Reality and

Visualization (ICVRV), 2016, pp. 154–157.

[14] Q. Zhang, X. Xue, D. Zhou, X. Wei, “Motion key-frames extraction based on amplitude of distance characteristic curve,” International Journal of Computational Intelligence Systems, vol. 7, no. 3, pp. 506–514, 2014.

[15] C. Halit, T. Capin, “Multiscale motion saliency for keyframe extraction from motion capture sequences,” Computer Animation and Virtual Worlds, vol. 22, no. 1, pp. 3–14, 2011.

[16] R. Roberts, J. P. Lewis, K. Anjyo, J. Seo, Y. Seol, “Optimal and interactive keyframe selection for motion capture,” in SIGGRAPH Asia 2018 Technical Briefs, ser. SA ’18. New York, NY, USA: Association for Computing Machinery, 2018.

[17] B. Sun, D. Kong, S. Wang, J. Li, “Keyframe extraction for human motion capture data based on affinity propagation,” in 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2018, pp. 107–112.

[18] Q. Zhang, S.-P. Yu, D.-S. Zhou, X.-P. Wei, “An efficient method of key-frame extraction based on a cluster algorithm,” Journal of Human Kinetics, vol. 39, no. 1, pp. 5–14, 2013.

[19] K.-S. Huang, C.-F. Chang, Y.-Y. Hsu, S.-N. Yang, “Key probe: A technique for animation keyframe extraction,” The Visual Computer, vol. 21, pp. 532–541, 2005.

[20] Q. Zhang, S. Zhang, D. Zhou, “Keyframe extraction from human motion capture data based on a multiple population genetic algorithm,” Symmetry, vol. 6, pp. 926–937, 2014.

[21] X.-m. Liu, A.-m. Hao, D. Zhao, “Optimization-based key frame extraction for motion capture animation,” The Visual Computer, vol. 29, 2012.

[22] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, A. Weber, “Documentation mocap database hdm05,” Universität Bonn, Tech. Rep. CG-2007-2, 2007.