



Motion Capture Technique with Enhancement Filters for Humanoid Model Movement Animation

¹Ahmad Yasin Habibillah, ^{2,*} Adhi Prahara, ³ Murinto

^{1,2,3}Department of Informatics, Universitas Ahmad Dahlan, Yogyakarta, Indonesia 55166

¹bilayinata28@gmail.com, ^{2,*}adhi.prahara@tif.uad.ac.id, ³murintokusno@tif.uad.ac.id

*correspondence email

Abstract

The definition of 3D animation is a representation of objects that are made into animation using characters or objects to look more alive and real. Making 3D animation itself requires a long process and a large amount of funding. This is because most 3D animated films still use key-framing technology which causes the process to make an animation to take a lot of steps. In this research, a motion capture technique with an enhancement filter is proposed to make humanoid movement animation using Kinect 2.0. The method consists of several steps such as recording every skeleton joint of human movements using a Kinect sensor, filtering the movements to minimize the shakiness and jitter from Kinect data, mapping skeleton data to the bones of a rigged humanoid model, and recording each movement to make animation. The final result is in the form of a 3D animation of modern dance movements. The method is tested by measuring the similarity between the 3D humanoid model and the user movement. From the 10 animations of modern dance generated by the method and performed by the user, a questionnaire to measure the MRI and MSE value is distributed, and the result achieves 4.27 on a scale of 5 for the averaged MRI score and 0.0539 for the MSE score. The MSE value is less than 5% which means the system is categorized as acceptable.

Keywords: motion capture, enhancement filter, movement animation, humanoid model, Kinect sensor

INTRODUCTION

The definition of 3D animation is a representation of objects that are made into animation using characters or objects to look more alive and real. Making 3D animation itself requires long processing and a large amount of funding. This is because most 3D animated films still use key-framing technology which causes the process to make an animation to take a lot of steps [1]. The cost of making 3D animation in a studio with the full support of animation tools costs around 3000 Euros per month depending on the duration of the tool usage and data collection. An example of technological development in the field of 3D animation is the motion capture technique. Motion capture is a process of recording a movement of an object and then implementing it in digital form or 3D objects. This technique is usually used to make real/animated films or games. The motion capture technique can effectively cut production costs and increase the efficiency of 3D animation production.

The result of 3D animation made using the motion capture method is not always perfect. Usually, there is shakiness in the legs or arms of the object [2]. Therefore, an enhancement method is needed to optimize the final result of 3D animation, namely by using the animation layer and filtering. The animation layer is very useful to enhance the result of the motion capture method because it only edits certain points in the animation that require changes so that the other data/objects are not changed. Filtering can help to prevent movements that penetrate the object's body and overcomes the shakiness in the object's legs or arms.

Generating animation using the motion capture method has been proposed by many researchers [1]–[9]. From the previous related research, the motion capture method of the humanoid model for sports animation movements using markerless sensors did not always produce good results. Therefore extra effort is needed to get a good animation result [8]. Another related research is the use of silhouette-based motion capture without markers. The system component contains silhouette extraction based on the level set which connects the data image with the data model and poses the estimation module. Experiments were carried out in different camera settings and estimated the model components with 21 degrees of freedom at up to two frames per second. To handle the stability problem, a marker-based motion estimation system is performed. The method is applied for the analysis of markerless sports movements [4].

In the reference entitled applicability of a single depth sensor in real-time 3D clothes simulation: augmented reality virtual dressing room using Kinect sensor, Kinect sensor is used to create a virtual dressing room that is useful for assisting users in making online purchases, especially on clothes. The system can simulate clothes in a 3D form in real-time [10]. The last reference discusses the use of motion capture technology with multiple Kinect controllers namely Kinect 360 and Kinect One controllers. Multiple Kinect controllers are used in parallel to produce a good motion capture result. The experiment results also compare the use of single and multiple Kinect to implement the motion capture systems [9].

In this research, the proposed motion capture technique is enhanced with some filters to make animation for the humanoid model. The paper is presented as follows: section 2 explains the proposed motion capture method, section 3 presents the result and discussion and finally, section 4 presents the conclusion of this work.

METHODS

THE PROPOSED MOTION CAPTURE METHOD

The proposed motion capture method consists of several steps namely reading the movement of skeleton joints using a Kinect sensor, performing filtering to enhance the movement, mapping the skeleton joints to humanoid bones, and finally recording the animation. The flowchart of the proposed method is shown in Fig. 1. The details of each step will be explained in the following sub-sections.

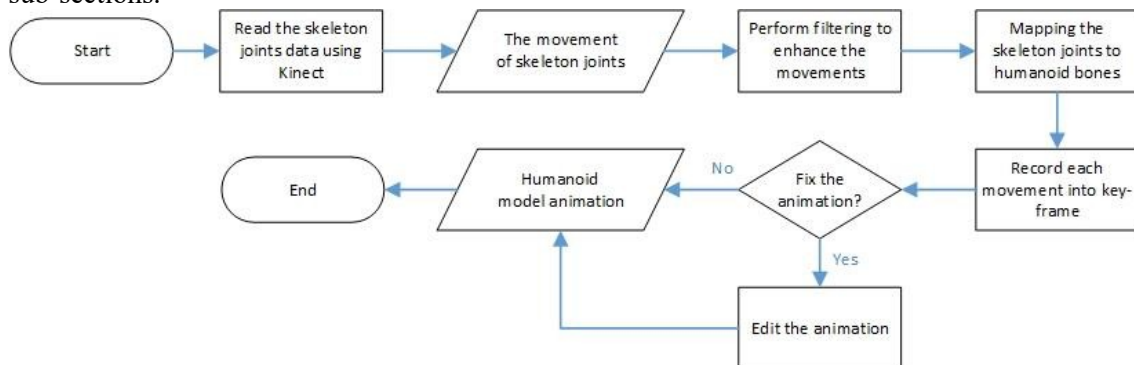


Fig. 1. Flowchart of the proposed motion capture method.

Skeletal Tracking Using Kinect

Skeleton joints are tracked using a Kinect sensor in the form of 3D coordinate positions and orientations from the joint movements of the user which are detected by the sensor. The data obtained from the sensor will be mapped into humanoid body bones data so that it can be implemented into a 3D humanoid model. The configuration of the skeleton joints of the Kinect sensor and humanoid body bones of Unity can be seen in Fig. 2. Based on Fig. 2, (a) is the skeleton joints tracked by the sensor, (b) is the configuration index of the skeleton joints, (c) is the rigged humanoid model, and (d) is the humanoid bones configuration in Unity.

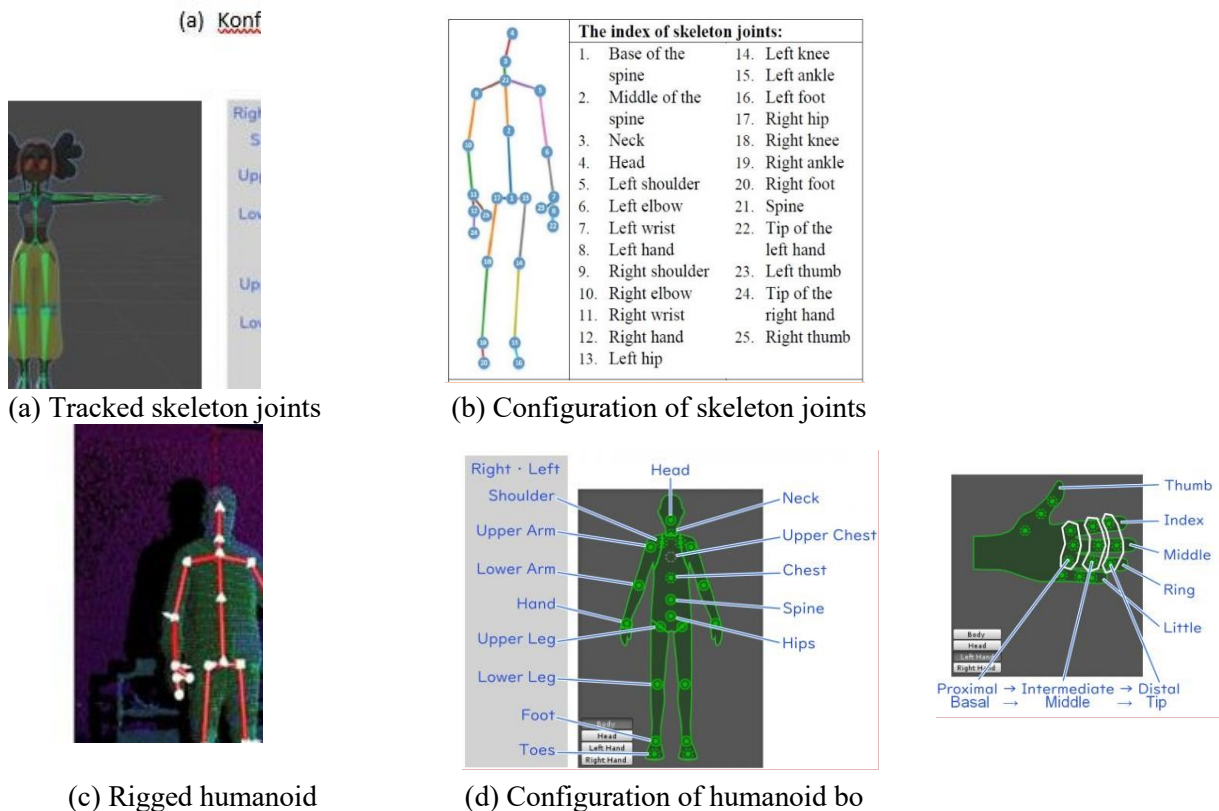


Fig. 2. Configuration of skeleton joints from Kinect and humanoid bones from Unity.

Enhancement Filters

The filtering step is used to optimize the movements of the tracked skeleton joints by the sensor. The filters will limit the joint movements to minimize the shakiness or jitter produced by the sensor or sudden fast movements by the user. The filters are adopted from the MS-SDK library [11] namely:

- Smoothing filter. The smoothing filter is performed by detecting the difference between the coordinates of the current joint and the previous joint. If the difference is too large then it is likely shakiness because the movement of each joint of each frame should be sequential with a small difference in position.
- Bone orientation filter. A bone orientation filter is applied so that the joint does not rotate too far and follows the orientation direction of the humanoid model.
- Movement speed filter. This filter limits the movement and provides motion tolerance so that the animation sequence can be maintained properly.

Mapping Skeleton Joints to Humanoid Bones

The mapping step is performed to map Kinect skeleton joints data into humanoid bones data. This mapping process will produce a motion animation of the humanoid 3D model [1]. There are 25 joints in the Kinect data while there are 52 joints in the humanoid bones model. Therefore, not all joints in the humanoid bones model can be used, such as the finger bones (except for the index and thumb) and the toes. The process of mapping 25 skeleton joints from the Kinect sensor to the bones in the humanoid model (configuration standard from Unity) and the respective parent bones are shown in Table 1. Based on Table 1, the head and chest bones were not used in this research. because the standard Kinect joints did not track these bones' movement and they only applied when face detection is turned on to track head movements.

After mapping, the movement of each humanoid bones model will be updated, that is, translation and rotation are carried out based on the movement of the skeleton joints from the Kinect sensor.

The translation is carried out by changes respective to the root, namely SpineBase by updating the initial position added with the offset of the change in position. Quaternion-based rotation is performed by multiplying the initial rotation of each bone of the humanoid model by the rotation of the Kinect skeleton joint. The rotation of the Kinect skeleton joints is calculated based on the orientation of the current joint with the next connected joint. After getting the rotation value, animation can be performed.

Table 1. Mapping skeleton joints from Kinect to humanoid bones

Skeleton joints	Parent joints	Humanoid bones
SpineBase	SpineBase	Hips
SpineMid	SpineBase	Spine
SpineShoulder	SpineMid	Chest
Neck	SpineShoulder	Neck
Head	Neck	Head
ShoulderLeft	SpineShoulder	LeftUpperArm
ElbowLeft	ShoulderLeft	LeftLowerArm
WristLeft	ElbowLeft	LeftHand
HandLeft	WristLeft	LeftIndex Proximal
HandTipLeft	HandLeft	LeftIndex Intermediate
ThumbLeft	WristLeft	LeftThumbProximal
ShoulderRight	SpineShoulder	Right UpperArm
ElbowRight	ShoulderRight	Right LowerArm
WristRight	ElbowRight	RightHand
HandRight	WristRight	RightIndexProximal
HandTipRight	HandRight	RightIndexIntermediate
ThumbRight	WristRight	RightThumb Proximal
HipLeft	SpineBase	LeftUpperLeg
KneeLeft	HipLeft	LeftLowerLeg
AnkleLeft	KneeLeft	LeftFoot
FootLeft	AnkleLeft	LeftToes
HipRight	SpineBase	Right UpperLeg
KneeRight	HipRight	Right LowerLeg
AnkleRight	KneeRight	RightFoot
FootRight	AnkleRight	RightToes

The results of the mapping skeleton joints to the humanoid bones are shown in Fig. 3 where (a) is the skeleton joints' position and orientation, (b) is the humanoid model movement, and (c) is the user movement. Based on Fig. 3, the humanoid model correctly follows the movement of the user.



Fig. 3. The example of mapping skeleton joints to the humanoid bones.

Animation Recording

The recording step is the final step of the animation-making process using the motion capture technique. At this step, every movement that is generated from the motion capture method will be recorded and then rendered to produce an animation. The animation is stored in .anim file format. The recording step will record each bone or joint that moves each frame into the keyframe. The transformation data such as transform, orientation, and scale will be recorded at this step.

Performance measurements

The performance is measured using MRI (Mean Rating Index) and MSE (Mean Square Error) metrics. MRI is the average rating index (RI) that can be used to measure the success of research involving the opinions of the respondents. MSE is used to measure the performance between the observation (RI_{obs}) and the expectation rating index (RI_{exp}). The lower the MSE score means the better performance of the method. If ω_i is the Likert scale of weight i , S_i is the number of the respondent at each weight, i is the weight where $i = 1, 2, 3, 4, 5$, M is the number of the respondent, N is the number of tests and RI_j is the rating index at test- j then the formulas to compute RI, MRI, and MSE are shown in (1), (2), and (3) respectively [12].

$$RI = \frac{1}{M} \sum_{i=1}^5 \omega_i S_i \quad (1)$$

$$MRI = \frac{1}{N} \sum_{j=1}^N RI_j \quad (2)$$

$$MSE = \frac{1}{N} \sum_{j=1}^N \left(RI_{obs}(j) - RI_{exp}(j) \right)^2 \quad (3)$$

RESULT AND DISCUSSIONS

The proposed motion capture method is implemented using humanoid models from the Mixamo platform [13], C# programming language, Unity [14], Kinect library [15], and MS-SDK library [11]. The test was conducted on a Laptop with an Intel i7 processor, NVidia GTX 1050, and 16GB of RAM. The result of each step to produce an animated motion using the motion capture technique on a humanoid model is described as follows:

Authors and Affiliations

The template is designed so that author affiliations are not repeated each time for multiple authors of the same affiliation. Please keep your affiliations as succinct as possible (for example, do not differentiate among departments of the same organization). This template was designed for two affiliations.

The Result of Mapping Skeleton Joints to Humanoid Bones

Input data from the Kinect sensor such as skeleton joint positions and orientations are mapped into humanoid bones using the standard configuration of the Unity humanoid avatar. The mapping result is shown in Fig. 4 where (a) is the skeleton joints' position in X, Y, and Z coordinates and rotations quaternion and (b) is their respective humanoid bones' positions and orientations. Based on Fig. 4, 25 Kinect skeleton joints data are mapped into 25 bones of a 3D humanoid model to represent a motion respective to the user movement.

Kinect Skeleton Joints Data		Humanoid Body Bones Data	
Spine Base	Pos = [X: -0.77 Y: 0.50 Z: 3.27], Orient = [X: -0.08 Y: -0.08 Z: 0.01 W: 1.00]	Hips	Pos = [X: -0.77 Y: 0.91 Z: 0.77], Orient = [X: -0.01 Y: 0.97 Z: 0.25 W: 0.98]
Spine Mid	Pos = [X: -0.77 Y: 0.80 Z: 3.32], Orient = [X: -0.06 Y: -0.08 Z: 0.01 W: 1.00]	Spine	Pos = [X: -0.77 Y: 1.00 Z: 0.81], Orient = [X: 0.02 Y: 1.00 Z: 0.00 W: 0.98]
Neck	Pos = [X: -0.77 Y: 1.10 Z: 3.36], Orient = [X: 0.03 Y: 0.08 Z: 0.01 W: 1.00]	Chest	Pos = [X: 0.00 Y: 0.00 Z: 0.00], Orient = [X: 0.00 Y: 0.00 Z: 0.00 W: 0.00]
Head	Pos = [X: 0.00 Y: 0.00 Z: 0.00], Orient = [X: 0.00 Y: 0.00 Z: 0.00 W: 0.00]	Neck	Pos = [X: -0.76 Y: 1.47 Z: 0.83], Orient = [X: 0.00 Y: 0.99 Z: -0.15 W: 0.98]
Left Shoulder	Pos = [X: -0.93 Y: 0.98 Z: 3.32], Orient = [X: 0.03 Y: 0.10 Z: 0.41 W: 0.98]	Head	Pos = [X: 0.00 Y: 0.00 Z: 0.00], Orient = [X: 0.00 Y: 0.00 Z: 0.00 W: 0.00]
Left Elbow	Pos = [X: -1.06 Y: 0.83 Z: 3.28], Orient = [X: -0.56 Y: -0.64 Z: 0.09 W: 0.98]	Left Upper Arm	Pos = [X: -0.59 Y: 1.33 Z: 0.84], Orient = [X: 0.94 Y: 0.28 Z: 0.21 W: 0.98]
Left Wrist	Pos = [X: -1.10 Y: 1.00 Z: 3.16], Orient = [X: -0.77 Y: -0.60 Z: 0.16 W: 0.98]	Left Lower Arm	Pos = [X: -0.46 Y: 1.11 Z: 0.86], Orient = [X: 0.40 Y: 0.40 Z: 0.02 W: 0.98]
Left Hand	Pos = [X: 0.00 Y: 0.00 Z: 0.00], Orient = [X: 0.00 Y: 0.00 Z: 0.00 W: 0.00]	Left Hand	Pos = [X: -0.36 Y: 1.30 Z: 0.69], Orient = [X: -0.04 Y: 0.70 Z: 0.00 W: 0.98]
Right Shoulder	Pos = [X: -0.60 Y: 0.98 Z: 3.37], Orient = [X: -0.10 Y: -0.06 Z: 0.47 W: 0.98]	Left Index Prox.	Pos = [X: 0.00 Y: 0.00 Z: 0.00], Orient = [X: 0.00 Y: 0.00 Z: 0.00 W: 0.00]
Right Elbow	Pos = [X: -0.47 Y: 0.80 Z: 3.37], Orient = [X: -0.57 Y: 0.53 Z: -0.13 W: 0.98]	Left Index Inter.	Pos = [X: 0.00 Y: 0.00 Z: 0.00], Orient = [X: 0.00 Y: 0.00 Z: 0.00 W: 0.00]
Right Wrist	Pos = [X: -0.38 Y: 0.96 Z: 3.27], Orient = [X: -0.74 Y: 0.59 Z: -0.22 W: 0.98]	Left Thumb	Pos = [X: 0.00 Y: 0.00 Z: 0.00], Orient = [X: 0.00 Y: 0.00 Z: 0.00 W: 0.00]
Right Hand	Pos = [X: 0.00 Y: 0.00 Z: 0.00], Orient = [X: 0.00 Y: 0.00 Z: 0.00 W: 0.00]	Right Upper Arm	Pos = [X: -0.93 Y: 1.33 Z: 0.78], Orient = [X: 0.93 Y: -0.34 Z: -0.06 W: 0.98]
Left Hip	Pos = [X: -0.84 Y: 0.51 Z: 3.23], Orient = [X: -0.01 Y: -0.08 Z: 0.03 W: 1.00]	Right Lower Arm	Pos = [X: -1.10 Y: 1.14 Z: 0.75], Orient = [X: 0.36 Y: -0.29 Z: 0.10 W: 0.98]
Left Knee	Pos = [X: -0.85 Y: 0.20 Z: 3.22], Orient = [X: -0.04 Y: 0.07 Z: -0.03 W: 1.00]	Right Hand	Pos = [X: -1.11 Y: 1.34 Z: 0.56], Orient = [X: -0.08 Y: -0.61 Z: -0.04 W: 0.98]
Left Ankle	Pos = [X: -0.84 Y: -0.14 Z: 3.25], Orient = [X: 0.05 Y: -0.15 Z: 0.00 W: 0.98]	Right Index Prox.	Pos = [X: 0.00 Y: 0.00 Z: 0.00], Orient = [X: 0.00 Y: 0.00 Z: 0.00 W: 0.00]
Left Foot	Pos = [X: 0.00 Y: 0.00 Z: 0.00], Orient = [X: 0.00 Y: 0.00 Z: 0.00 W: 0.00]	Right Index Inter.	Pos = [X: 0.00 Y: 0.00 Z: 0.00], Orient = [X: 0.00 Y: 0.00 Z: 0.00 W: 0.00]
Right Hip	Pos = [X: -0.68 Y: 0.50 Z: 3.25], Orient = [X: 0.04 Y: 0.08 Z: 0.03 W: 1.00]	Right Thumb	Pos = [X: 0.00 Y: 0.00 Z: 0.00], Orient = [X: 0.00 Y: 0.00 Z: 0.00 W: 0.00]
Right Knee	Pos = [X: -0.66 Y: 0.19 Z: 3.23], Orient = [X: 0.05 Y: -0.08 Z: -0.02 W: 1.00]	Left Upper Leg	Pos = [X: -0.66 Y: 0.90 Z: 0.79], Orient = [X: 0.28 Y: 0.09 Z: -0.95 W: 0.98]
Right Ankle	Pos = [X: -0.67 Y: -0.15 Z: 3.26], Orient = [X: 0.05 Y: -0.15 Z: 0.00 W: 0.98]	Left Lower Leg	Pos = [X: -0.59 Y: 0.50 Z: 0.73], Orient = [X: 0.06 Y: -0.07 Z: -0.99 W: 0.98]
Right Foot	Pos = [X: 0.00 Y: 0.00 Z: 0.00], Orient = [X: 0.00 Y: 0.00 Z: 0.00 W: 0.00]	Left Foot	Pos = [X: -0.53 Y: 0.07 Z: 0.80], Orient = [X: 0.11 Y: 0.47 Z: -0.87 W: 0.98]
Spine Shoulder	Pos = [X: 0.00 Y: 0.00 Z: 0.00], Orient = [X: 0.00 Y: 0.00 Z: 0.00 W: 0.00]	Left Toes	Pos = [X: 0.00 Y: 0.00 Z: 0.00], Orient = [X: 0.00 Y: 0.00 Z: 0.00 W: 0.00]
Left Hand Tip	Pos = [X: -1.13 Y: 1.10 Z: 3.18], Orient = [X: 0.79 Y: 0.60 Z: -0.08 W: 0.98]	Right Upper Leg	Pos = [X: -0.88 Y: 0.91 Z: 0.76], Orient = [X: -0.12 Y: 0.07 Z: -0.99 W: 0.98]
Left Thumb	Pos = [X: -1.05 Y: 1.03 Z: 3.11], Orient = [X: 0.00 Y: 0.00 Z: 0.00 W: 1.00]	Right Lower Leg	Pos = [X: -0.94 Y: 0.50 Z: 0.70], Orient = [X: 0.14 Y: -0.06 Z: -0.99 W: 0.98]
Right Hand Tip	Pos = [X: -0.36 Y: 1.07 Z: 3.27], Orient = [X: -0.79 Y: 0.57 Z: -0.18 W: 0.98]	Right Foot	Pos = [X: -1.00 Y: 0.07 Z: 0.75], Orient = [X: 0.11 Y: 0.49 Z: -0.86 W: 0.98]
Right Thumb	Pos = [X: -0.33 Y: 1.02 Z: 3.27], Orient = [X: 0.00 Y: 0.00 Z: 0.00 W: 1.00]	Right Toes	Pos = [X: 0.00 Y: 0.00 Z: 0.00], Orient = [X: 0.00 Y: 0.00 Z: 0.00 W: 0.00]

Fig. 4. The result of mapping skeleton joints to humanoid bones.

The mapping result in Fig. 4 is already enhanced with enhancement filters such as smoothing filter, bone orientation filter, and movement speed filter. These filters are applied because the skeleton joints data are sensitive to the sensor accuracy so shakiness and jitter effects are often found in the hand or leg joints. The filters can minimize the effects and optimize the movements before they are mapped to humanoid bones. The result of enhancement filters is shown in Fig. 5 where (a) is the movement before applying the enhancement filters and (b) is the movement after applying the enhancement filters. Based on Fig. 5, the filters produce fairly accurate movement results.

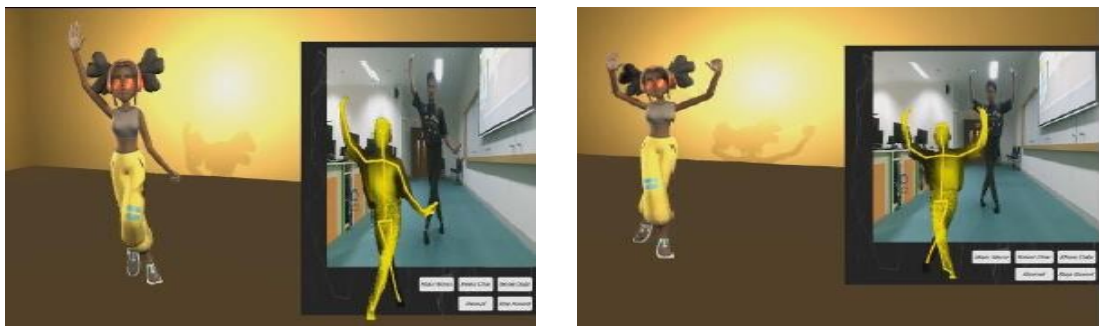


Fig. 5. The result of applying enhancement filters.

An example of an animation result generated by the motion capture technique is shown in Fig. 6.

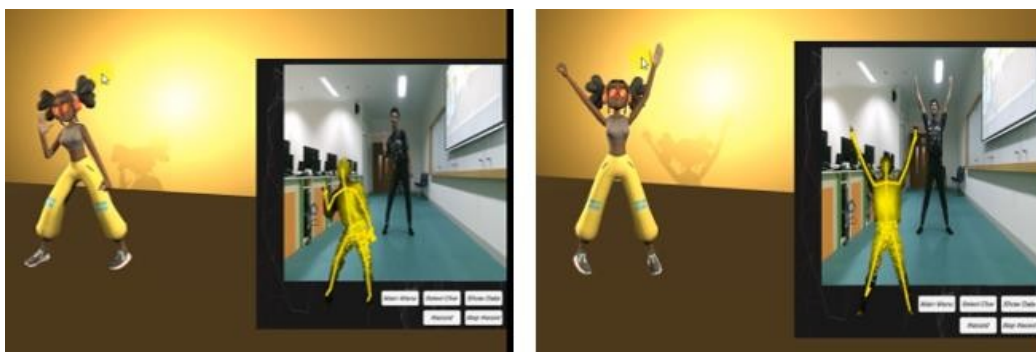


Fig. 6. The result of animation generated by the proposed method.

Based on Fig. 6, the mapping result from skeleton joints into humanoid bones is fairly accurate and the animation sequence in each frame also shows no shakiness and jitter. However, the model is not yet capable to perform rotating motions where the user turns his back from the camera because, in this research, the head position is not tracked.

Performance Evaluation

The performance is evaluated using MRI and MSE scores obtained from the questionnaire that was distributed to 25 respondents. The questionnaire compares 10 modern dance movements performed by the user and the 3D humanoid model and the respondent requires to rate the similarity of the movement. The result achieves an MRI score of 4.276 and an MSE score of 0.053936. Based on the MSE results, the proposed method is considered acceptable because the MSE score is less than 5% [12]. The test results can be seen in Table 2.

Table 2. The MRI and MSE score.

	(Observation)	(Expectation)	Absolute	Square	MSE
Motion 1	4.24	5	0.76	0.5776	0.05776
Motion 2	4.28	5	0.72	0.5184	0.05184
Motion 3	4.20	5	0.80	0.6400	0.06400
Motion 4	4.44	5	0.56	0.3136	0.03136
Motion 5	4.36	5	0.64	0.4096	0.04096
Motion 6	4.40	5	0.60	0.3600	0.03600

CONCLUSIONS

A motion capture technique with enhancement filters for humanoid model movement animation using a Kinect sensor has been proposed and tested. The test was conducted on 10 modern dance animations/movements that have been made in only approximately 3 hours (without editing process and correcting erroneous gestures). This shows the efficiency in production time. The result of applying enhancement filters is shown in the accurate movements generated by the method. The proposed method achieves an MRI score of 4.27 and an MSE score of 0.053936 which means acceptable ($MSE \leq 0.05$).

Acknowledgment: This research is supported by LPPM Universitas Ahmad Dahlan with research grant no. PD-221/SP3/LPPM-UAD/2020.

REFERENCES

- [1] M. Izani, Aishah, A. R. Eshaq, and Norzaiha, "Keyframe animation and motion capture for creating animation: a survey and perception from industry people," in *Proceedings. Student Conference on Research and Development, 2003. SCORED 2003.*, 2003, pp. 154–159, doi: 10.1109/SCORED.2003.1459684.
- [2] T.-Y. Mou, "Keyframe or Motion Capture? Reflections on Education of Character Animation," *EURASIA J. Math. Sci. Technol. Educ.*, vol. 14, no. 12, Oct. 2018, doi: 10.29333/ejmste/99174.
- [3] Mr. Naeemabadi, B. Dinesen, O. K. Andersen, and J. Hansen, "Investigating the impact of a motion capture system on Microsoft Kinect v2 recordings: A caution for using the technologies together," *PLoS One*, vol. 13, no. 9, p. e0204052, Sep. 2018, doi: 10.1371/journal.pone.0204052.
- [4] et al. Rosenhahn, Bodo, "A system for marker-less motion capture," *Künstliche Intelligenz*, vol. 1, pp. 45–51, 2006.

-
- [5] M. Naemabadi, B. Dinesen, O. K. Andersen, and J. Hansen, "Influence of a Marker-Based Motion Capture System on the Performance of Microsoft Kinect v2 Skeleton Algorithm," *IEEE Sens. J.*, vol. 19, no. 1, pp. 171–179, Jan. 2019, doi: 10.1109/JSEN.2018.2876624.
- [6] T. M. Guess, S. Razu, A. Jahandar, M. Skubic, and Z. Huo, "Comparison of 3D Joint Angles Measured With the Kinect 2.0 Skeletal Tracker Versus a Marker-Based Motion Capture System," *J. Appl. Biomech.*, vol. 33, no. 2, pp. 176–181, Apr. 2017, doi: 10.1123/jab.2016-0107.
- [7] A. Napoli, S. Glass, C. Ward, C. Tucker, and I. Obeid, "Performance analysis of a generalized motion capture system using microsoft kinect 2.0," *Biomed. Signal Process. Control*, vol. 38, pp. 265–280, Sep. 2017, doi: 10.1016/j.bspc.2017.06.006.
- [8] A. Shingade and A. Ghotkar, "Animation of 3D Human Model Using Markerless Motion Capture Applied To Sports," *Int. J. Comput. Graph. Animat.*, vol. 4, no. 1, pp. 27–39, Jan. 2014, doi: 10.5121/ijcga.2014.4103.
- [9] P. Kopniak, "Motion capture using multiple Kinect controllers," *Przegląd elektrotechniczny*, vol. 8, pp. 26–29, 2015.
- [10] S. B. Adikari, N. C. Ganegoda, R. G. N. Meegama, and I. L. Wanniarachchi, "Applicability of a Single Depth Sensor in Real-Time 3D Clothes Simulation: Augmented Reality Virtual Dressing Room Using Kinect Sensor," *Adv. Human-Computer Interact.*, vol. 2020, pp. 1–10, May 2020, doi: 10.1155/2020/1314598.
- [11] R. Filkov, "Kinect with MS-SDK," 2013. <https://rfilkov.com/2013/12/16/kinect-with-ms-sdk/> (accessed Feb. 01, 2020).
- [12] et al. Setyati, Endang, "Phoneme-Viseme Mapping for Indonesian Language Based on Blend Shape Animation," *IAENG Int. J. Comput. Sci.*, vol. 42, no. 3, 2015.
- [13] Adobe, "Mixamo Character," 2019. <https://www.mixamo.com/#/?page=1&type=Character> (accessed Feb. 01, 2020).
- [14] Unity, "Unity User Manual," 2020. <https://docs.unity3d.com/Manual/index.html> (accessed Feb. 01, 2020).
- [15] Microsoft, "Microsoft Kinect SDK," 2014. <https://developer.microsoft.com/en-us/windows/kinect/> (accessed Feb. 01, 2020).