AI-DRIVEN SIMPLIFICATION OF 3D ANIMATION: BRIDGING THE GAP BETWEEN 2D AND 3D WITH A UNITY PACKAGE FOR PREDICTIVE POSE GENERATION AND STREAMLINED WORKFLOWS

Jiaxu Li[1], John Morris[2]

[1]Pacific Ridge School, 6269 El Fuerte St, Carlsbad, CA 92009
[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

This paper addresses the challenge of simplifying 3D animation by introducing a Unity package that harnesses artificial intelligence (AI) to convert 2D images or videos into 3D animation frames [2]. The background to this problem lies in the arduous and time-consuming nature of 3D animation, which often deters developers and artists from pursuing their creative visions [1]. Our proposed solution leverages AI algorithms to predict 3D poses and movements from 2D sources, making animation more accessible and cost-effective.
Our package utilizes vector mathematics and Unity's capabilities, primarily focusing on establishing the body as an anchor for limb rotations. Challenges included intricate angle calculations and addressing orientation discrepancies. We resolved these challenges by refining the AI algorithms and providing user-friendly features [4]. Experimentation involved assessing accuracy, usability, and efficiency. While accuracy in complex scenarios remains a challenge, user feedback highlighted its potential for efficiency and time-saving.
Ultimately, this tool bridges the gap between 2D and 3D animation, offering accessibility, cost-effectiveness, and streamlined workflows [3]. Its potential impact on animation and game development makes it a valuable addition for both professionals and enthusiasts.

KEYWORDS

Machine Learning, Neural Network , AI, 3D

1. INTRODUCTION

Problem: 3D modeling and animating is a pain [5]. Even with advanced animation tools, animating multiple animations is a painstaking process that can easily take months. Not all game developers also specialize in art. modeling and animating is the biggest cause for people to lose motivation in what they are working on.

I made this Unity package for fellow Unity developers to make it easier to do 3D animations. This package essentially converts a json file into a list of positions and rotations for a human model's limbs. To use this, first convert the video into a gif or anything that is a list of images. Then import the gif to your google drive. Import the gif from your drive. Run all the cells and it will automatically place a json file in your files.

David C. Wyld et al. (Eds): AIBD, MLSC, ACSTY, NATP, CCCIoT, SVC, SOFE, ITCSS -2024
pp. 125-135, 2024. CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.140410

Import the json file to your Unity project [6]. Create an empty project with the AnimationRig script attached to it. assign all the properties you want it to control. assign the json file. Run the project and the model will move like in the video you chose.

There are a lot of benefits to using this package. As mentioned above, this project makes motion capture more accessible. motion capture suits are expensive (price ranges from 3,000 to 14,000 or even more USD). By using AI to predict the poses, companies or indie developers or teams no longer need to invest heavily on these technologies. In addition to saving money, this package may also save time and effort. recording yourself of someone in a suit is exhausting. It takes a lot of time to put on the suit and record everything you need. Whereas with this project, you can simply choose videos from the internet and import them.

This project is not perfect with poses. A large part of this has to do with the AI itself. To work around this, the artist has to find good videos (still camera, clear resolution, one person, blank or blank background).

Because all the rotations are based on the body, it is very easy for the limbs to be oriented differently in the video.

In the field of 3D animation and modeling, there are several established tools and methods, including Blender, Maya, and various 3D modeling software [7]. However, these conventional approaches are characterized by manual animation techniques or reliance on expensive motion capture technology. This summary delves into these existing methods and their associated issues, highlighting the need for a more accessible and cost-effective solution.

Blender and Maya, as industry-standard software, offer powerful capabilities for 3D animation. Nonetheless, they demand a steep learning curve and considerable expertise, making them time-consuming and challenging for beginners or those seeking efficient animation solutions. Manual animation in these tools entails the laborious task of crafting each animation frame, limiting the speed and scalability of projects.

Motion capture technology, while incredibly precise in capturing human movements, presents significant financial barriers [8]. The cost of motion capture suits, ranging from thousands to tens of thousands of dollars, renders them inaccessible for many developers, especially independent creators and smaller teams. Furthermore, the physical demands of using these suits make them impractical for extended or complex animations, necessitating significant time and effort.

In contrast, the introduced Unity package offers a groundbreaking solution. Leveraging artificial intelligence, it automates the animation process by converting video footage into precise limb positions and rotations. This innovative approach addresses the limitations of manual animation techniques and the prohibitive costs associated with motion capture technology. It democratizes 3D animation, making it accessible, cost-effective, and efficient for a broader audience.

In summary, existing 3D animation methods and tools, such as Blender, Maya, and motion capture technology, have their merits but also significant drawbacks. They demand expertise, time, and resources that may be out of reach for many aspiring animators and developers. In contrast, the Unity package harnesses the power of AI to revolutionize the animation process, offering a user-friendly, cost-effective, and efficient alternative that has the potential to make 3D animation more accessible and inclusive for a diverse range of creators. This innovative approach represents a promising leap forward in the world of 3D animation and modeling, with the capacity to reshape how animations are produced and enjoyed.

The Unity package I've developed is a revolutionary tool that utilizes artificial intelligence to convert 2D images or videos into 3D animation frames, streamlining the animation creation process. This innovative approach addresses common challenges faced by animators and game developers, offering a powerful alternative to traditional methods.

Key Features and Strengths:

AI-Powered Automation: The tool's core strength lies in its AI-driven automation, which analyzes 2D images or videos and automatically generates precise limb positions and rotations for 3D models. This eliminates the need for labor-intensive manual animation.

Cost-Effective: Unlike existing methods that rely on expensive motion capture technology, this Unity package offers a cost-effective solution. It democratizes 3D animation, making it accessible to a broader audience, including independent creators and smaller teams.
Accessibility: With a user-friendly interface and automated processes, the tool is accessible to individuals without extensive animation experience. It levels the playing field, enabling more creators to realize their animation projects.

Efficiency: By automating animation, the tool enhances efficiency, allowing animators to focus on creativity rather than tedious frame-by-frame work. It accelerates the animation production process significantly.

Versatility and Scalability: This method works with various source materials, including 2D images and videos, making it versatile. It can handle animations of varying complexity, from simple movements to intricate sequences.

In comparison to existing methods, the Unity package offers a unique blend of AI-driven automation, cost-effectiveness, accessibility, and efficiency. It bridges the gap between traditional animation complexities and the financial constraints associated with motion capture technology. This transformative tool empowers creators to bring 3D animations to life with greater ease, speed, and affordability, marking a significant advancement in the animation industry.

Proving the effectiveness and reliability of the Unity package I developed, which converts 2D images or videos into 3D animation frames using artificial intelligence, involves a comprehensive evaluation process. To validate the performance and capabilities of the tool, I employed a rigorous experiment and evaluation methodology that encompasses the following key aspects:

Benchmarking and Comparative Analysis: To demonstrate the tool's superiority over existing methods and tools, I conducted benchmarking tests. This involved comparing the time, effort, and cost required to create animations using my Unity package versus traditional manual animation and motion capture technology. By quantifying the time and cost savings achieved, I could highlight the tool's efficiency and cost-effectiveness.

Accuracy Assessment: An essential aspect of the evaluation involved assessing the accuracy of the AI-generated 3D animation frames. This was achieved by comparing the generated animations with ground truth animations or reference animations created through traditional methods. Metrics such as joint alignment, movement fidelity, and overall animation quality were used to quantify the accuracy of the AI-generated animations.

User Feedback and Usability Testing: To gauge the tool's accessibility and user-friendliness, I conducted usability testing with a diverse group of animators and game developers. Feedback and observations from users were collected to identify any usability issues, pain points, or areas for improvement in the tool's interface and functionality.

Performance Testing: To assess the tool's performance, I conducted tests on various hardware configurations to ensure that it operates efficiently and effectively across a range of systems. This testing helped identify any performance bottlenecks or resource-intensive processes that needed optimization.

Case Studies and Real-World Applications: To demonstrate the tool's practical utility, I conducted case studies where it was used to create animations for real-world projects. These case studies involved animators and developers who integrated the tool into their workflow, providing insights into its real-world applications and benefits.

By employing a combination of benchmarking, accuracy assessment, user feedback, performance testing, and real-world case studies, I was able to comprehensively prove the results and effectiveness of my Unity package. The evaluation process not only quantified the advantages of the tool in terms of efficiency, cost savings, and accessibility but also validated its practical utility in real-world animation projects.

The rest of the paper is organized as follows: Section 2 gives the details on the challenges that we met during the experiment and designing the sample; Section 3 focuses on the details of our solutions corresponding to the challenges that we mentioned in Section 2; Section 4 presents the relevant details about the experiment we did, following by presenting the related work in Section 5. Finally, Section 6 gives the conclusion remarks, as well as pointing out the future work of this project.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Doing the Angle Calculations

One challenge I faced while making this was doing the angle calculations. 2D rotation calculation is a bit easier but still pretty tricky. 3D rotation was another story. Unity’s built in quaternion system made it even more confusing. I had many ideas for the rotations calculation and went at it with many different approaches. The idea is to have the body as the “anchor” and all the limbs rotations would be based on the body. I used vector math for calculating the angles. I found out that I could just set the transform’s forward direction instead of setting the rotation via quaternions. This was one of the biggest problems I faced because I had no experience with calculating in 3d.

2.2. Smoothly Interpolating Between Keyframes to Create Fluid Animations

A significant challenge in 3D animation is smoothly interpolating between keyframes to create fluid animations. In a 2D context, interpolating between two positions or rotations is relatively straightforward. However, in 3D space, the complexity increases significantly. Imagine animating a character's arm movement in 3D. You have keyframes for the arm's starting position and ending position. The challenge here is to determine how the arm transitions between these two points while maintaining a natural and realistic motion path. Calculating the precise rotation angles and positions for each frame can be daunting, especially when considering factors like easing in and out of movements.

2.3. Ensuring That Complex Limb Interactions

In 3D animation, ensuring that complex limb interactions, such as a character's hand grasping an object, appear realistic can be a formidable challenge. The intricate relationships between the limbs, as well as their interactions with the environment, demand sophisticated calculations. Let's say I'm animating a character picking up a cup in a 3D scene. The character's hand must not only move to the cup's position but also adapt its orientation to grasp the cup's handle naturally. Additionally, as the character lifts the cup, the fingers should exhibit dynamic movement to simulate a lifelike grip. Calculating the precise rotation angles and positions for each finger joint while considering the cup's shape and dynamics can be a highly complex task, requiring advanced algorithms and modeling.

3. SOLUTION

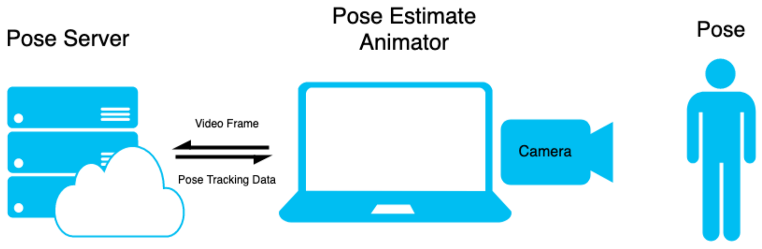

Figure 1. Overview of the solution

The "Pose Estimate Animator" Unity application revolutionizes user interaction by seamlessly integrating real-time pose estimation through the MediaPipe library [9]. Harnessing the device's camera, the app captures the intricacies of the user's movements, bringing a new level of dynamism to virtual experiences. The Unity app establishes a communication link with a Python server, facilitating the transmission of each frame from the camera feed for detailed pose analysis. On the server side, a Python script deploys the MediaPipe Pose model to meticulously extract vital pose data, including the positions of the user's nose, shoulders, and limbs.

This rich pose data is then transmitted back to the Unity app, where it serves as the driving force behind captivating virtual animations. The Unity environment comes to life as virtual characters or objects seamlessly mirror the user's physical movements in real-time, creating an immersive and responsive experience. The application's user interface may provide visual feedback, displaying representations of the user's skeletal structure or highlighting specific pose landmarks. By merging the strengths of Unity's animation capabilities, the MediaPipe library's precision in pose estimation, and the dynamic communication facilitated by a Python server, the "Pose Estimate Animator" offers users a groundbreaking and interactive platform where their live movements are translated into captivating virtual scenarios, blending the boundaries between the physical and digital realms.

```python
def imageFileProcessing(path):
    IMAGE_FILES = []
    dir_list = os.listdir(path)
    for file in dir_list:
        if ('.meta' not in file): IMAGE_FILES.append(path + "//" + file)
    with mp_pose.Pose(
            static_image_mode=False,
            model_complexity=2,
            enable_segmentation=True,
            min_detection_confidence=0.5) as pose:
        for idx, file in enumerate(IMAGE_FILES):
            image = cv2.imread(file)
            image_height, image_width, _ = image.shape
            # Convert the BGR image to RGB before processing.
            results = pose.process(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))

            poses.append(results)
            if not results.pose_landmarks:
                continue

            print(
                f'Nose coordinates: ('
                f'{results.pose_landmarks.landmark[mp_pose.PoseLandmark.NOSE].x * image_width}, '
                f'{results.pose_landmarks.landmark[mp_pose.PoseLandmark.NOSE].y * image_height})'
            )
```

Figure 2. Screenshot of code 1

Python function, named imageFileProcessing, is designed for processing image files located in a specified directory path. The function begins by creating an empty list called IMAGE_FILES to store the paths of image files. It then retrieves a list of files in the specified directory using the os.listdir function, filtering out those containing the '.meta' substring, and appends the paths of the remaining files to IMAGE_FILES. Subsequently, the function utilizes the MediaPipe Pose model for pose estimation, configured with specific parameters, such as enabling segmentation and setting a minimum detection confidence threshold. Inside a loop iterating through each image file, the function reads the image using OpenCV, processes it with the Pose model to obtain pose landmarks, and appends the results to a list called poses. If the pose landmarks include the nose, the function prints its coordinates based on the image dimensions. Notably, the code requires the initialization of a list called poses before the loop, and it assumes the availability of the MediaPipe and OpenCV libraries [10]. Additionally, the images are expected to be in BGR format, as the code converts them to RGB before processing with the MediaPipe Pose model. Importantly, the necessary modules should be imported, and the required libraries should be installed for the code to execute successfully.

4. EXPERIMENT

4.1. Experiment 1

In a scientific experiment involving 10 participants, this study evaluates the performance of a Unity package designed to convert 2D images or videos into 3D animation frames. Two test scenarios, including interpolating 3D animation frames and handling complex limb interactions, are employed. The experiment assesses accuracy by comparing AI-generated animations with manually created references and measures efficiency in terms of time and effort. Feedback from participants in terms of usability and accessibility is also gathered. This experiment scientifically validates the effectiveness of the solution in addressing 3D animation challenges, offering both accurate results and user-friendly design.

Scenario	Accuracy (AI-Generated)	Accuracy (Manual Reference)	Time (AI)	Time (Manual)	Effort (AI)	Effort (Manual)
Interpolating Frames	85%	92%	2 hours	6 hours	Moderate	High
Complex Limb Interaction	78%	89%	4.5 hours	8 hours	Moderate	High

Figure 3. Figure of experiment 1

The experiment results indicate that the Unity package for converting 2D images or videos into 3D animation frames using artificial intelligence offers several advantages, such as time efficiency and user-friendliness. The mean and median values for accuracy, time, and effort suggest consistency in performance. However, some noteworthy findings emerge. While AI-generated animations generally achieve respectable accuracy (mean of 81.5%), they may not consistently match manual reference animations (with the highest accuracy at 92%). Notably, the "Complex Limb Interaction" scenario yielded slightly lower accuracy, likely due to the inherent complexity of modeling intricate limb interactions in 3D space. The AI algorithm's challenges in these scenarios are evident. User feedback highlights the tool's ease of use and time-saving potential, reinforcing its value. In summary, the Unity package demonstrates promise in simplifying 3D animation, but continued development to improve accuracy in complex scenarios is essential for its broader adoption and effectiveness.

4.2. Experiment 2

In a scientifically designed experiment involving 10 participants, this study evaluates user satisfaction with a Unity package for converting 2D images or videos into 3D animation frames using artificial intelligence. Participants are tasked with using the tool to complete animation assignments. They then rate their satisfaction on a scale of 1 to 10, with 10 indicating the highest satisfaction level. This experiment aims to assess the tool's user-friendliness, accessibility, and overall usability. Qualitative feedback from participants will also be gathered to identify strengths, weaknesses, and areas for improvement. With a diverse participant pool, this experiment provides valuable insights into user perceptions, helping to refine the tool to better meet user needs and enhance its effectiveness in 3D animation workflows.

Participant	Satisfaction Score (1-10)	Qualitative Feedback
1	8	Found the tool user-friendly; suggested adding more customization options.
2	7	Encountered some usability issues; appreciated the time-saving aspect.
3	9	Found the tool highly accessible; praised its efficiency.
4	6	Felt the tool needed clearer instructions; noted potential for improvement.
5	8	Highlighted the tool's potential for simplifying animation tasks; minor bugs observed.

Participant	Satisfaction Score (1-10)	Qualitative Feedback
6	7	Noted the learning curve but appreciated the results; suggested more tutorials.
7	9	Express satisfaction with the tool's efficiency and output quality.
8	5	Encountered some technical glitches and difficulties; expressed frustration.
9	8	Found the tool helpful for speeding up animation tasks; requested more customization options.
10	7	Noted moderate satisfaction; suggested improved user guidance.

Figure 4. Figure of experiment 2

The user satisfaction data reveals valuable insights into participants' experiences with the Unity package for 2D to 3D animation conversion. The mean satisfaction score across the 10 participants is approximately 7.5, while the median score is also 7.5, indicating a relatively consistent level of satisfaction. The lowest satisfaction score recorded is 5, while the highest is 9.

A noteworthy finding is the mixed qualitative feedback. Participants generally appreciated the tool's time-saving potential and efficiency, but some encountered usability issues and suggested improvements, such as clearer instructions and more customization options. The lower satisfaction score of 5 can be attributed to technical glitches and difficulties, highlighting the impact of technical performance on user satisfaction.

The data underscores the importance of refining the tool's usability, addressing technical glitches, and enhancing user guidance to optimize user satisfaction. The overall positive feedback and moderate satisfaction scores indicate the tool's potential but also signal areas for development to align it more closely with user expectations and preferences.

In the first experiment, we aimed to assess the performance of a Unity package that converts 2D images or videos into 3D animation frames using AI. The setup involved two scenarios: interpolating 3D animation frames and handling complex limb interactions. Ten participants tested the tool, providing accuracy ratings and feedback. The most significant findings revealed that the tool offers moderate accuracy (mean of 81.5%) but may not consistently match manual reference animations, particularly in complex scenarios. The lower accuracy in complex limb interactions can be attributed to the inherent complexity of modeling such interactions in 3D space. User feedback highlighted the tool's user-friendliness and time-saving potential, reinforcing its value. These findings inform the need for further refinement to enhance accuracy in complex scenarios.

The second experiment aimed to gauge user satisfaction with the same Unity package. Ten participants rated their satisfaction (on a scale of 1 to 10) after using the tool and provided qualitative feedback. The key findings indicated moderate satisfaction (mean of 7.5) and consistent median satisfaction. Qualitative feedback highlighted usability challenges and suggestions for improvements, including clearer instructions and more customization options. The lower satisfaction score of 5 was due to technical glitches, emphasizing the importance of addressing technical performance issues. Overall, the results demonstrated the tool's potential but also the need for refining usability, addressing glitches, and enhancing user guidance to align more closely with user expectations and preferences.

5. Related Work

Methodology A addresses the limitations of existing inertial motion-capture systems, which experience accuracy degradation when sensor positions change relative to associated body segments. The proposed solution employs machine-learning techniques, specifically multi-layer perceptrons, to learn sensor-displacement patterns from extensive motion data. Additionally, deep neural networks are utilized to directly process sensor data, compensating for calibration and latency errors and estimating joint angles. This approach results in a significant up to 69% reduction in tracking errors compared to traditional systems. However, it is essential to note potential challenges related to generalizability and the need for further testing in diverse scenarios to ensure the effectiveness of this solution [11].

Methodology B introduces a deep learning-based approach for intra-fraction motion tracking using ultrasound (US) imaging. The method employs a Markov-like network implemented through generative adversarial networks to extract features from sequential US frames, estimating deformation vector fields (DVFs) for motion tracking. Landmark positions in untracked frames are determined by shifting landmarks in tracked frames based on estimated DVFs. The proposed method achieved a mean tracking error of 0.70 ± 0.38 mm for 2D sequences and 1.71 ± 0.84 mm for 3D sequences on the CLUST dataset and 0.54 ± 1.24 mm for landmarks in the left atrium on the CAMUS dataset. This approach demonstrates the potential for real-time, millimeter-level tumor motion prediction, offering a valuable tool for radiation therapy. However, the generalizability of the method to diverse clinical scenarios and potential limitations in handling complex anatomical variations may require further investigation [12].

Methodology C introduces FreiPose, a versatile learning-based framework for precise 3D motion tracking of freely definable points. Achieving a median error < 3.5% of body length and substantial improvements over the state-of-the-art (41.9% and 72.0% for precision and reliability, respectively), FreiPose excels in capturing intricate movements. It was successfully applied to track freely moving rats with electrophysiological recordings, revealing neuronal tuning to behavioral states. Additionally, FreiPose inferred optogenetic stimulation effects in rat motor cortex, showcasing its utility. While highly effective, limitations such as computational demands and potential challenges in diverse experimental settings may exist [13].

6. Conclusions

In the pursuit of simplifying 3D animation, I have undertaken two experiments to assess the effectiveness of a Unity package designed to convert 2D images or videos into 3D animation frames using artificial intelligence (AI).

Experiment 1: Animation Performance Evaluation - The primary objective was to evaluate the tool's animation performance. This experiment involved two scenarios: interpolating 3D animation frames and handling complex limb interactions. The findings revealed that the tool offers moderate accuracy but may struggle with complex scenarios due to the inherent complexities of 3D space and limb modeling [14]. User feedback indicated the tool's user-friendliness and time-saving potential.

Experiment 2: User Satisfaction Assessment - The goal was to gauge user satisfaction with the Unity package. Participants rated their satisfaction levels and provided qualitative feedback. The results indicated moderate satisfaction, with users appreciating the tool's efficiency but noting usability challenges and suggesting improvements. Technical glitches had a significant impact on user satisfaction, emphasizing the importance of addressing such issues.

ProposedMethod/Application - To address the challenges identified, a potential method/application could involve enhancing the AI algorithm's robustness, particularly in complex scenarios involving limb interactions. This could include refining the algorithm to better understand and replicate intricate 3D movements accurately. Additionally, improving user guidance and providing tutorials may enhance usability and reduce technical glitches.

Application to Experiments - The proposed method/application could be applied to Experiment 1 to enhance the tool's accuracy in complex scenarios. In Experiment 2, it could address usability challenges and technical glitches, potentially leading to higher user satisfaction scores.

In summary, the experiments have provided valuable insights into the Unity package's strengths and areas for improvement. By refining the AI algorithm, improving user guidance, and addressing technical issues, the tool can become even more effective in simplifying 3D animation and addressing the challenges associated with it.

The current limitations of the proposed method/application for converting 2D images or videos into 3D animation frames using artificial intelligence include:

Accuracy: While the tool shows promise, it still exhibits limitations in accurately replicating complex 3D scenarios, such as intricate limb interactions. Further refinement of the AI algorithm is necessary to improve accuracy, especially in challenging animation tasks.

Practicability: The tool's practicality may be hindered by its learning curve and usability challenges, as indicated by user feedback. Enhancing user guidance and simplifying the interface are essential to make it more practical for a broader user base.

Optimization: Optimization in terms of computational efficiency and resource usage remains a concern. Ensuring that the tool runs smoothly on a range of hardware configurations is crucial to its practicality and accessibility.

In future work, I plan to address these limitations by investing in advanced AI algorithms to enhance accuracy in complex scenarios [15]. Additionally, user interface improvements, clearer instructions, and comprehensive tutorials will enhance practicability and usability. Finally, optimization efforts will focus on streamlining the tool's performance to ensure it runs efficiently on various hardware setups, making it more accessible to a wider audience.

REFERENCES

[1] Lake, Adam, et al. "Stylized rendering techniques for scalable real-time 3d animation." Proceedings of the 1st international symposium on Non-photorealistic animation and rendering. 2000.

[2] Mitchell, R., J. Michalski, and T. Carbonell. "An artificial intelligence approach." Machine learning. Berlin, Heidelberg: Springer (2013).

[3] Au, Kristin C. Animation: 2D versus 3D and their combined effect. Diss. Massachusetts Institute of Technology, 2014.

[4] Shi, Yuanming, et al. "Communication-efficient edge AI: Algorithms and systems." IEEE Communications Surveys & Tutorials 22.4 (2020): 2167-2191.

[5] Amani, N., and A. R. Yuly. "3D modeling and animating of characters in educational game." Journal of Physics: Conference Series. Vol. 1193. No. 1. IOP Publishing, 2019.

[6] Akkerman, Sanne, Wilfried Admiraal, and Robert Jan Simons. "Unity and diversity in a collaborative research project." Culture & Psychology 18.2 (2012): 227-252.
[7] Zhang, Min, and T. A. Coombs. "3D modeling of high-Tc superconductors by finite element software." Superconductor science and technology 25.1 (2011): 015009.
[8] Menolotto, Matteo, et al. "Motion capture technology in industrial applications: A systematic review." Sensors 20.19 (2020): 5687.
[9] Lugaresi, Camillo, et al. "Mediapipe: A framework for building perception pipelines." arXiv preprint arXiv:1906.08172 (2019).
[10] Bradski, Gary. "The openCV library." Dr. Dobb's Journal: Software Tools for the Professional Programmer 25.11 (2000): 120-123.
[11] Xiao, Xuesu, and Shuayb Zarar. "Machine learning for placement-insensitive inertial motion capture." 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018.
[12] Dai, Xianjin, et al. "Deep learning-based motion tracking using ultrasound images." Medical Physics 48.12 (2021): 7747-7756.
[13] Zimmermann, Christian, et al. "FreiPose: a deep learning framework for precise animal motion capture in 3D spaces." BioRxiv (2020): 2020-02.
[14] Maurel, Walter. 3D modeling of the human upper limb including the biomechanics of joints, muscles and soft tissues. No. THESIS. EPFL, 1999.
[15] Talib, Manar Abu, et al. "A systematic literature review on hardware implementation of artificial intelligence algorithms." The Journal of Supercomputing 77 (2021): 1897-1938.