Study on 2D Sprite *3.Generation Using the Impersonator Network

Yongjun Choi[1], Beomjoo Seo[1], Shinjin Kang[1] and Jongin Choi[2*]
[1] School of Games (Engineering), Hongik University
2639 Sejong-ro, Jochiwon, Sejong, Korea
[e-mail : cyjun131@naver.com, [bseo,directx]@hongik.ac.kr]
[2] Department of Digital Media, Seoul Women's University
621 Hwarangro, Nowon-Gu, Seoul, Korea
[e-mail : funtech@swu.ac.kr]
[*]Corresponding author : Jongin Choi

Received March 14, 2023; revised June 18, 2023; accepted June 24, 2023; published July 31, 2023

Abstract

This study presents a method for capturing photographs of users as input and converting them into 2D character animation sprites using a generative adversarial network-based artificial intelligence network. Traditionally, 2D character animations have been created by manually creating an entire sequence of sprite images, which incurs high development costs. To address this issue, this study proposes a technique that combines motion videos and sample 2D images. In the 2D sprite generation process that uses the proposed technique, a sequence of images is extracted from real-life images captured by the user, and these are combined with character images from within the game. Our research aims to leverage cutting-edge deep learning-based image manipulation techniques, such as the GAN-based motion transfer network (impersonator) and background noise removal (U^2-Net), to generate a sequence of animation sprites from a single image. The proposed technique enables the creation of diverse animations and motions just one image. By utilizing these advancements, we focus on enhancing productivity in the game and animation industry through improved efficiency and streamlined production processes. By employing state-of-the-art techniques, our research enables the generation of 2D sprite images with various motions, offering significant potential for boosting productivity and creativity in the industry.

Keywords: Generative Adversarial Networks (GANs), 2D Sprite Animation, Deep Learning, Game Development, Impersonator Model

This research was supported by Culture, Sports and Tourism R&D Program Through the Korea Creative Content Agency grant funded by the Ministry of Culture, Sports and Tourism in 2022 (Project Name: Development of artificial intelligence-based game simulation technology to support online game content production, Project Number: 202201590001. This work was also supported by the National Research Foundation of Korea Grant funded by the Korean Government (NRF-2022R1G1A110012513).

http://doi.org/10.3837/tiis.2023.07.003 **ISSN : 1976-7277**

1. Introduction

Recently, active research in Artificial Intelligence (AI) -related field has led to advancements in technologies based on Generative Adversarial Networks (GANs) [1]. Simultaneously, researchers are attempting to use GANs to generate image data for sketches, cartoons, and face images. In the gaming industry, there have been attempts to boost game development productivity by adopting AI technology. Therefore, research is being conducted to introduce GAN-based image-to-image translation techniques to the field of 2D game sprite creation.

In the 2D game development process, creating game sprites is fundamentally labor-intensive. Traditionally, game character animation sprites have been created using pixel art techniques or programs for drawing such as Adobe Photoshop or Illustrator. The mount of 2D animation that is created increases based on the possible motion directions and the number of motions. A sequence of sprite images is created manually for each motion. These sequences of images are placed within a single texture image and uploaded to the game engine that will be used. The traditional 2D sprite creation process involves the manual completion of tasks, resulting in high manual work costs. Quickly creating the sequences of sprites that illustrates the users' desired motions during the process would significantly reduce game development costs.

This study aimed to increase the efficiency of the 2D game character animation sprite creation process using the impersonator network, which is a GAN-based image-to-image technology, in gaming. The proposed technique can significantly reduce game development costs by partially automating 2D game sprite creation.

2. Related Work

A GAN is an essential technique that is widely used to generate images via deep learning. The structure of a GAN comprises a generator (G), which generates images, and a discriminator (D), which discriminates the generated images. G learns to generate the desired image by continuously performing repeated training and correction until D correctly recognizes the image as correct. As research on Convolutional Neural Networks (CNNs) has advanced, the GAN structure has been combined with various networks that have been developed based on CNNs, and these have produced excellent results. Conditional GANs [2] were developed to control the data generation process by applying conditions (conditioning) to make additional information (class labels) known when inputting data into a GAN model. Progressive growing of GANs (PGGANs) [3] is a model developed by Nvidia that develops high-resolution images. Discriminators can be easily circumvented when resolution is low; thus, PGGAN employs an incremental learning approach, which progresses from low-resolution data to high-resolution data, enhancing learning speed. Style GAN [4] is a reconstructed model that applies the concept of Style Transfer [5] to the aforementioned PGGAN.

Recently, multi-modal learning networks have gained prominence, and these are models that receive text as input and generate images [6, 7, 8]. These keyword-based networks can generate the desired image using only a few keywords, and they significantly boost productivity in creating original artwork. Advancements in these image-generating networks are attracting the interest of game developers. In particular, studies on increasing the efficiency of game resource production are receiving considerable attention. Studies continue to investigate game character-related production and level content, such as 2D face sprite generation [9, 10] and 3D face generation [11, 12].

However, the generation of animation images is limited by the fact that generated images are not guaranteed to have spatiotemporal continuity. Of the various game resource generation technologies, the study focused on 2D game character sprite generation. Studies in this field have traditionally belonged to the motion transfer area in computer vision. Most deep learning-based human motion transfer methods train a GAN to generate new poses for a target person. Such methods can be broadly divided into two types based on the generalizability. The first is based on a general model; the model is trained for an unseen target, or the source pose is transferred to a given target within find adjustments [13, 14, 15, 16]. The second is based on an individualized model; the focus is placed on learning the appearance of a specific person, and new poses are generated for the same person [17, 18, 19, 20, 21].

Modeling and transferring human appearance comprise a wide range of topics, from computer graphics pipelines [22] to learning-based pipelines [23, 24]. Graphics-based methods approximate detailed 3D human meshes through clothing and 3D scanners [25] or multi-camera arrays [25]. Subsequently, they can transfer the appearance of a person wearing clothing from one person to another based on the detailed 3D mesh. Amit et al. [23] first trained a pose-guided clothing segmentation synthesis network and then generated the desired images by supplying an encoder-decoder network with clothing parsing results, using the texture function of the source image feed. Mihai et al. [24] used a geometric 3D shape model combined with a learning method to swap the colors of visible vertices in a triangular mesh, and they trained the model to infer the colors of unseen vertices.

Recently, researchers have presented various motion transfer methods that generate single detailed images of human targets in new poses [27, 28, 29]. Ma et al. [30, 31] and Siarohin et al.[32] introduced a new architecture and loss concepts for this purpose. In addition, it has been demonstrated that poses are signals that effectively guide future predictions and video generation [33, 34].

This study presents a method that uses an image learning network to imitate the motions in input images and alter the appearances of people in full-body input images to automatically generate anthropomorphic 2D game character animation sprites.

3. System

In this study, we aimed to generate atlas maps, which are sprite maps used in actual game engines, from a single map. To do this, we propose a novel 5-stage production process. The proposed process comprises the following stages. 1) Create single images with the game style that will be used. 2) Create motion images that are synthesis source for generating sequential animations. 3) Perform synthetic network learning to generate sequential image with a single image style. 4) Apply a background and noise remover to remove GAN-specific noise from the generated images. 5) Generate atlas maps and sprites using a commercial game engine to ensure that the generated image sequences can be used in actual games. **Fig. 1** illustrates the entire production process. Below are descriptions of each stage.

3.1 Single Character Image Production

The proposed process should be able to generate sequential motion images within the 2D art style that is desired by the user. Different 2D art exists, necessitating a redraw of all images upon alteration. Our methodology aims to resolve this problem by enabling users to draw a single image in the desired art style and input it into the network to generate motion images in the input image's style. The user's desired art style should be produced from the same perspective as the final game content that will have the style applied. For instance, if one aims to create sprites for a top-view game, images should be created from the top view, and for a side-view game, images should be created from the side view. The images should be inputted with their background removed to increase the network's learning capacity. As for the size of images, it was possible to generate results with the desired quality by inputting images with the same resolution or 1.5 times the resolution of the final target sprite. The final sprite used in the game depicted only the upper body enabled effective learning when synthesized with the upper body of motion images.

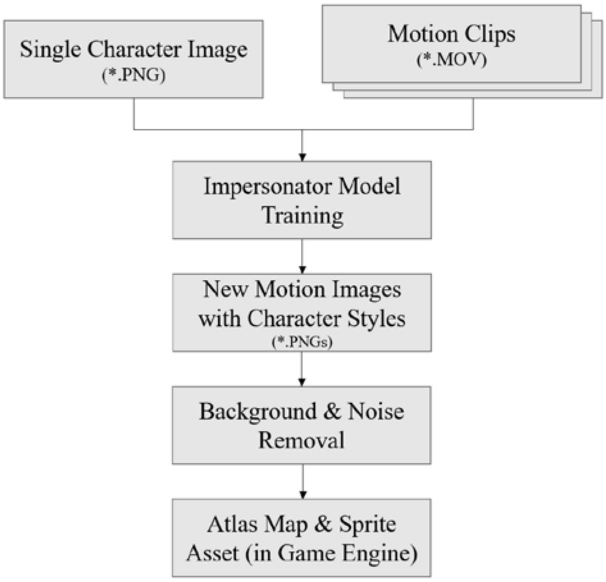
Single Character Image
(*.PNG)
Motion Clips
(*.MOV)
Impersonator Model Training
New Motion Images with Character Styles
(*.PNGs)
Background & Noise Removal
Atlas Map & Sprite Asset (in Game Engine)

Fig. 1. Summary of proposed sprite production process.

3.2 Motion Clip Production

The proposed methodology's key concept is to replace the cost of manual sequential image production with a relatively low-cost image source. To do this, motion clip images (e.g., idle, run, jump, shoot) are needed for the sprite types that will be used in the game. The proposed process does not require a specific space or equipment to produce the motion clips. The motion provider only needs to record the motion that will be used in the game from the game camera's perspective in a space with no background. Because the learning data used for final learning comprise sequential images in units of 3 - 5 frames, the results can be improved by recording high-resolution images at 30 – 60 frames or more if possible. Each motion should be captured separately, and this study confirmed that learning efficiency increased when each color was accurately distinguished or clothing of a single color was worn to be able to distinguish features such as arms, legs, and heads for learning efficiency.

3.3 Impersonator Model Training

In the network learning stage, we used liquid warping GAN (impersonator) [14], specializing in appearance transfer and novel view synthesis. Fig. 2 presents a schematic diagram illustrating the utilization of this network in our process. This network can easily apply actual images to images from new perspectives, and it can use sequential image as learning data to generate images from new perspectives. Such view synthesis techniques have been studied extensively, but there are no cases where they have been applied to sprite creation in the gaming field.

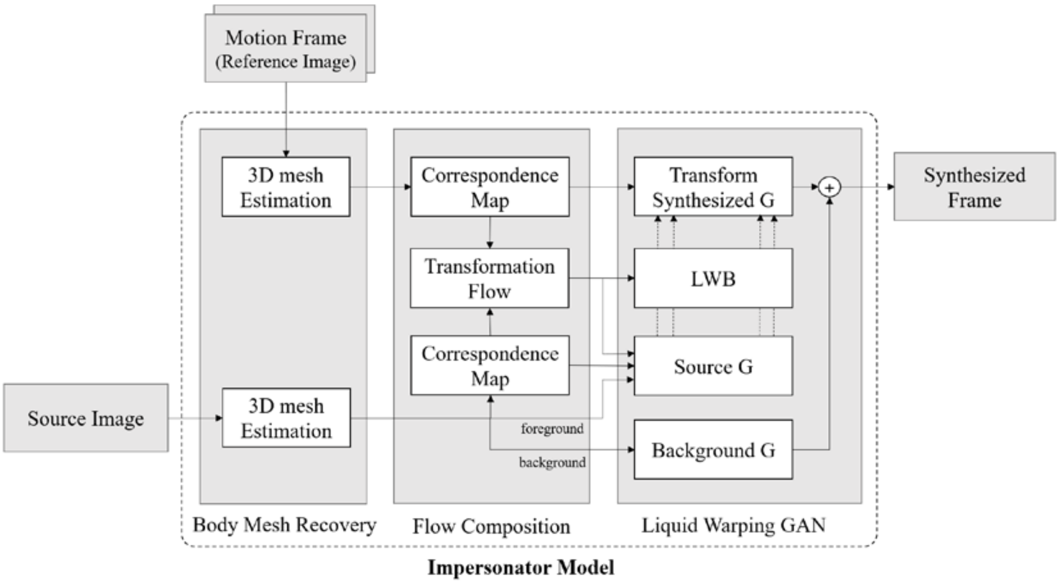

Fig. 2. Schematic diagram of the impersonator model employed in our production process.

The impersonator is characterized by having a 3D body mesh recovery module [35] and a liquid warping block (LWB), and it preserves the target source's feature information effectively during motion transfer. The liquid warping GAN network internally creates meshes from parameters such as the 3D meshes of each source image and reference images. The meshes created in each stage are used to separate the foreground and background of the source image. Finally, the background image, reconstructed source image, and reference image-based image are synthesized based on LWB data in the GAN module to generate a single new image. By repeating this process, a video with sequential images is generated.

Of the three areas where the impersonator network can be used (i.e., (1) motion imitation, (2) appearance transfer, and (3) view synthesis), we used the (1) motion imitation technique to generate sequential images. The (1) motion imitation technique extracts sequential image using transfer on the target image. This can intuitively create the motion that is desired by the actual user and can adjust the animation frames by adjusting the sample level. However, a cost burden for recapturing motion may be incurred if the motion for the desired result cannot be captured successfully or the network's learning results are poor. In (3) view synthesis, 360-degree synthesized images can be extracted from motions observed from a certain point of view. This approach uses a methodology in which the motion provider captures the desired motion in stop motion form during motion capture and then motion-transfers it to 360-degree images. This learning methodology can improve motion clip productivity with intuitive motion capturing; however, there is the difficulty of having to directly select a motion from the desired angle. **Fig. 3** shows the synthesized image results that were generated from the motion that we used. Blurry images were generated, which is typical of a GAN network. Nonetheless, we successfully generated synthesized images that were relatively similar to the desired image.

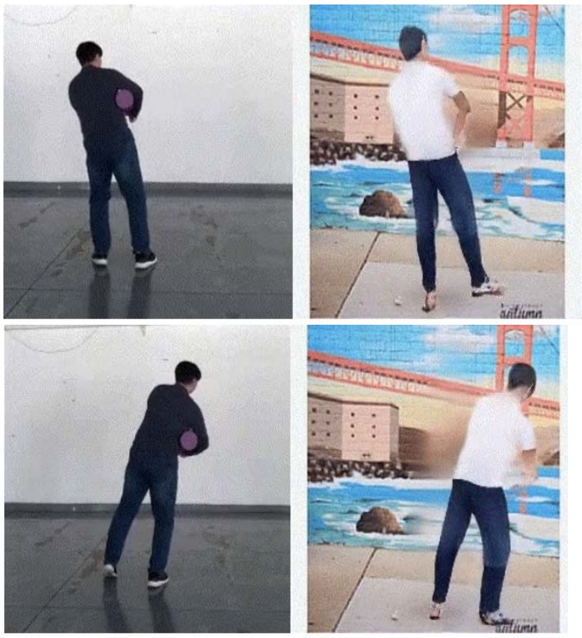

Fig. 3. Results of network learning. (Left) the motion clip captured by the user, (Right) the image generated by the proposed view synthesis technique.

3.4 Background/Noise Removals

The proposed process uses a GAN-based motion transfer network. This network can produce the desired outcomes efficiently; however, it has downsides like unstable learning and production of noise pixels in the testing phase. In actual game sprites, the background should be cleanly removed by assigning alpha values, and the images generated by a GAN network cannot meet this standard. To resolve this problem, we used a dedicated background remover network that we used was U^2-Net [36]. U^2-Net is a network developed based on U-Net [37]. U-Net mainly specializes in tracking Salient Object Detection (SOD) and segmenting objects. A structural feature of this network is that it forms a U shape because there is symmetry in the number of layers when dimension reduction and dimension expansion are performed. U^2-Net increases learning capacity by performing the U-shaped network learning process an additional time. We used this network to remove the image's background while leaving only the target character.

Fig. 4 shows the image results that were generated by the background removal technique that we used. The quality of the results is excellent compared to the results of using Adobe's Photoshop's magic wand function to perform manual removal. While the background remover yielded satisfactory results, 2 – 5 white pixels closer to the character were not completely removed. We directly selected the sprites that would ultimately be used from among the generated images, and we manually removed some of the remaining gray pixels from only the final selected sprites. Normally, classic 2D games use approximately 30 – 100 images to create sprite sets for a single character. The proposed methodology entails a corresponding cost for manual noise removal.

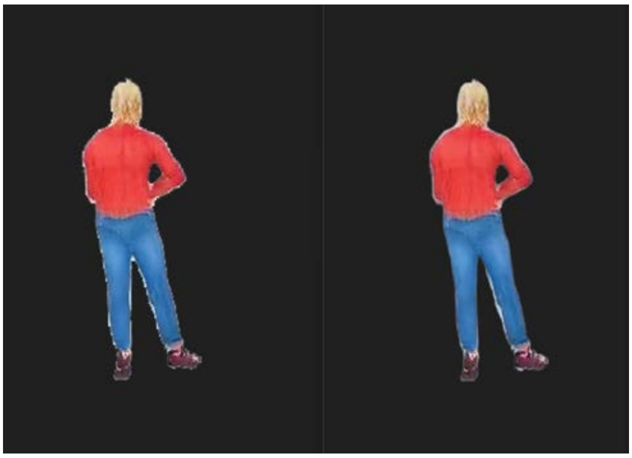

Fig. 4. Comparison of (left) background removal using Photoshop’s magic wand and (right) the image generated by the proposed background and noise remover network.

3.5 Atlas Map and Sprite Asset Generation

The images with background removed following this process were collected in a single folder and selected by the user. For sprites used in games, a single atlas map is generated to minimize memory usage, and this is referenced and rendered in real time in the game. The Unreal and Unity engines, which are commonly used game engines in the current game industry, provide functions that automatically generate atlas maps from sets of single images at a certain size. In the proposed process, the generated maps are ultimately converted into game assets through the game engine’s asset generation process.

4. Experimental Results

To confirm the effectiveness of the proposed methodology, we attempted to generate the sprites required by a game that imitates the classic game Space Harrier [38] by the Japanese company SEGA. Fig. 5 shows a screenshot of the final results that were generated by the proposed method. Google Colaboratory, an artificial environment provided by Google, was used for network learning, and the Unity engine was used to develop game prototyping.

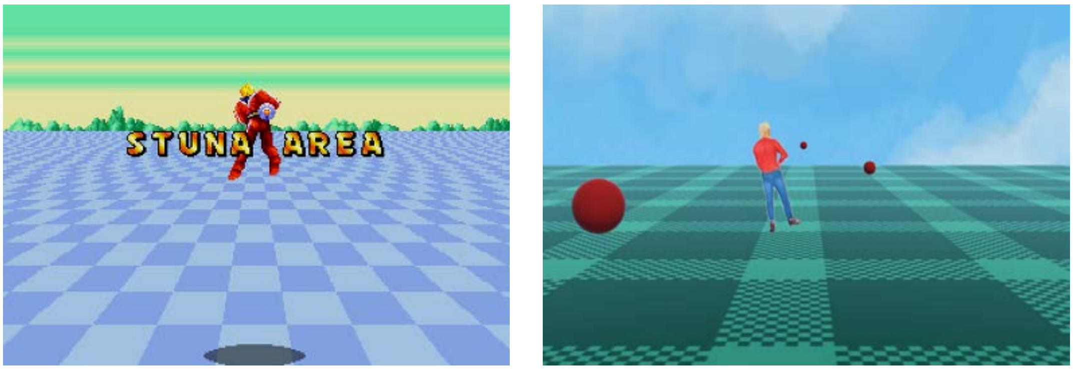

Fig. 5. Screenshots of the target classic game (left) and an imitation game (right) using a sprite character created by the proposed process.

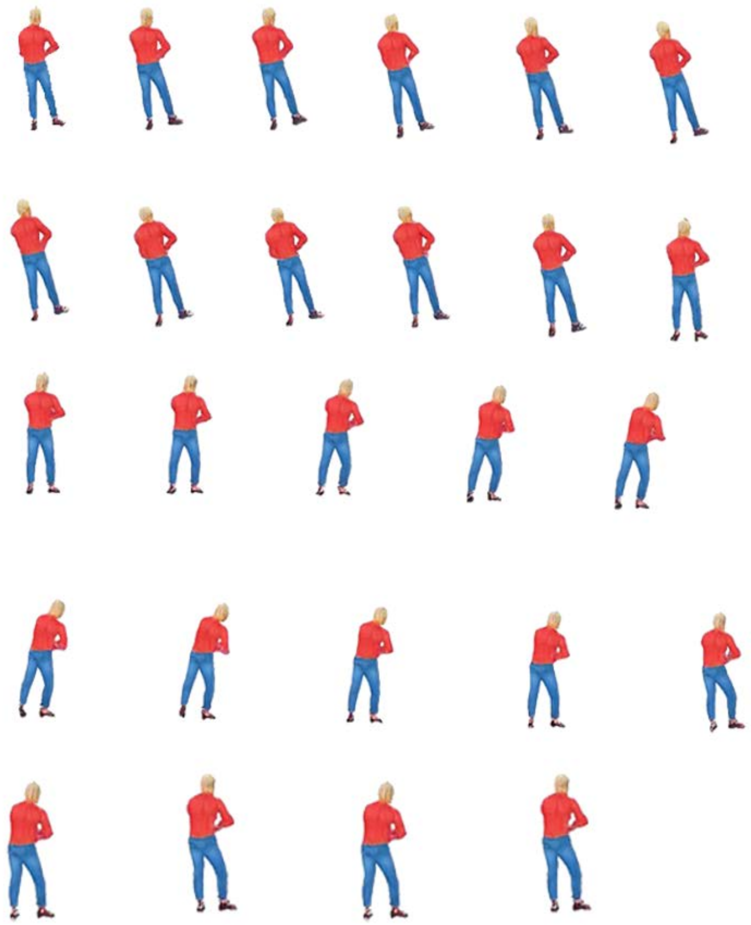

Fig. 6. Sprite image samples generated by the proposed process.

We used the proposed process to generate five motions (short-distance leftward movement, long-distance leftward movement, short-distance rightward movement, long-distance rightward movement, and idling in place). Four to six frames were created for each sprite to generate one atlas map comprising 30 images. Each image was resized to 420 × 420 to make it compatible with the learning environment. The time spent creating all sprites include 1 hour to capture five videos, 1 hour to remove noise, and 1 hour to generate target images, for a total of 3 hours. **Fig. 6** shows the generated sprite results.

The traditional method for producing sprite effects have four stages. The first stage is (a) concept design. In this stage, a concept artist draws an initial effect drawing, making it consistent with the game's visual concept. Subsequently, the lead artist confirms the drawing's visual consistency. This stage requires two to three days of production for a simple effect, and the required time varies based on the original drawing's complexity level. The next stage is (b) the sprite image production phase. In this stage, a 2D sprite creator normally uses a digital painting tool such as Adobe Photoshop to draw a sequence of successive images. This stage often requires one week of working time or more. The next stage is (c) creating the sprite in the game engine. In this stage, the image sequence is loaded into a game engine that is used to produce actual games, such as Unity or Unreal, and the asset that will be used in the actual game is created. The final stage is (d) the in-game evaluation stage, which evaluates the sprite that was produced in the actual game engine. The production of a single sprite passes through these stages in sequence, and a single effect requires approximately two weeks, If the final result does not meet the art director's evaluation standards, it is remade from the beginning or the intermediate results are modified.

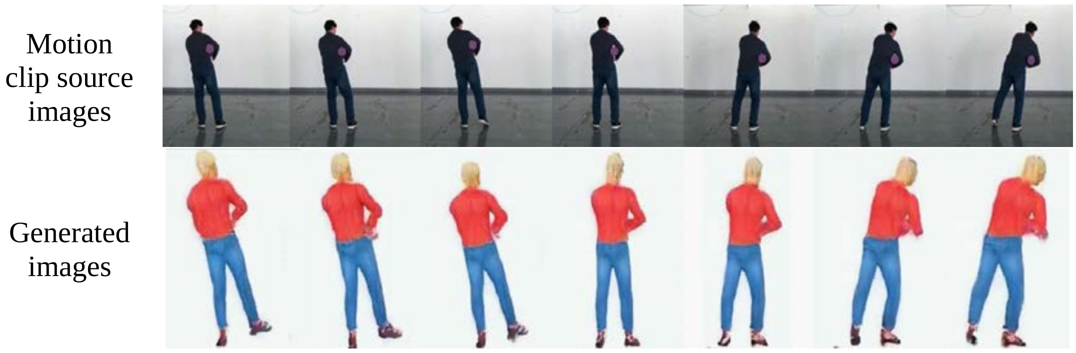

Fig. 7. Original motion clips and their corresponding images generated by the network.

In the proposed process, stages (a) and (b) are replaced. Using our proposed tool, sprites that are directed by the concept artist can be produced quickly and at a low cost. Additional postprocessing tasks may be needed for the GAN-generated images, but the existing sprite creation period of one week or more can be reduced to a maximum of 1 – 2 days, and the proposed process can allow sprites to be quickly replaced when the design change feedback is received. Generally, dozens or hundreds of sprites must be created during the game development process, and the use of the proposed tool can significantly reduce such development costs. The proposed methodology can generate sprites at the level required for prototyping production as intended even when the user does not have knowledge on art production. This shows that proposed method can be used effectively in the prototyping stage.

Because sprites that are generated using this method are based on actual images, they feature relatively smooth transitions between motions. **Fig. 7** shows the 1:1 matching relationship between a recorded motion clip and actual generated sprites. The overall form of the motion is similar to the recorded motion clip. This shows that the motion transfer network that is used in this process reveals the differences in each frame while preserving the overall shape and color of the subject. However, the subject's feet and hands were not successfully generated in the images. Space Harrier requires motions with the main character holding a weapon. To reflect this, we recorded motions while holding a cylinder that is similar to an actual weapon. However, this was not successfully reflected in the obtained images, and the network depicted the corresponding parts in a blurry manner. This shows that the network has limitations regarding learning and generating objects such as character accessory attachments. In a normal game sprite, exaggerated action motions are specified for the key framework to maximize the dynamic feel of the motion. However, because the proposed technique tends to express the motion user's raw motions in the real world, it is difficult to depict the extreme motions that are used in game genres such as action. Moreover, it appears that the technique can be used more effectively in game genres that present characters that are soft and have the same proportions as real-world human bodies.

During the experimental phase we conducted trials and rectified errors. First, we confirmed that the image generated were more accurate when the size and point of view of the applied game character image were consistent with that of the captured image. To ensure this consistency, two to three rounds of preliminary learning were required. The lower the consistency, the greater the inaccuracy of the resultant image. **Fig. 8** shows the results that were generated when the aforementioned size and point of view conditions were not met. Despite these drawbacks, the proposed technique can produce numerous sprites in various styles once a production process that ensures consistency in the image' size and points of view is set up once for a particular game genre. This should increase the efficiency of image production in games with real-world proportions that requires large amount of animation (e.g., Prince of Persia).

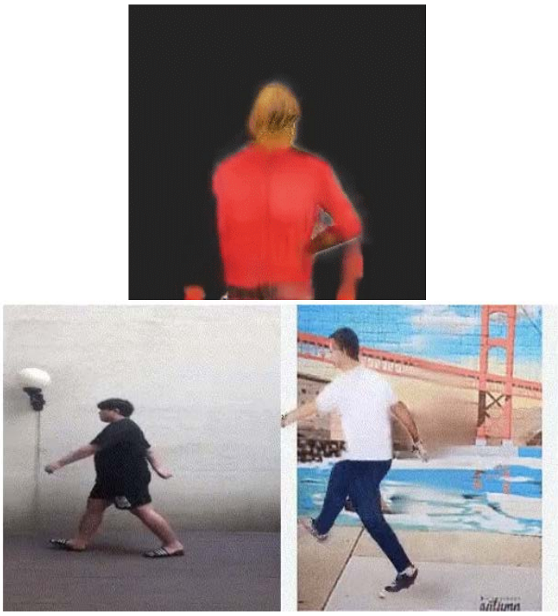

Fig. 8. Images results generated by the network when the image compositions are different (top) and while the image sizes are different (bottom).

5. Conclusion

This study introduces motion transfer technology, which is used in the domain of computer vision, to a production process that can be used to create game sprites. Creating sprites often necessitates a considerable monetary investment. Due to this limitation, classic games and even recent 2D games are still using sprites with limited numbers of frames and perspectives. The proposed methodology facilitates the mass production of various sprites for quick prototyping and live game services. In sprite creation, there are significant differences in working time based on the quality of the sprite; however, even assuming that additional postprocessing is performed, the proposed methodology can drastically reduce working time compared to conventional manual work. This study performed experiments on realistic characters, but in actual 2D games, pixel-style characters are often used. In future studies, we aim to further develop the proposed network to ensure that it can generate pixel-style characters.

References

[1] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," *Communications of the ACM*, vol. 63, no. 11, pp. 139-144, 2020. Article (CrossRef Link)

[2] M. Mirza, and S. Osindero, "Conditional Generative Adversarial Nets," *arXiv preprint arXiv:1411.1784*, 2014. Article (CrossRef Link)

[3] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive Growing of GANs for Improved Quality, Stability, and Variation," in *Proc. of International Conf. on Learning Representation*, 2018. Article (CrossRef Link)

[4] T. Karras, S. Laine, and T. Aila, "A Style-based Generator Architecture for Generative Adversarial Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 12, pp. 4217-4228, 2021. Article (CrossRef Link)

[5] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image Style Transfer Using Convolutional Neural Networks," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 2414-2423, 2016. Article (CrossRef Link)

[6] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-Shot Text-to-Image Generation," in *Proc. of the 37th International Conf. on Machine Learning*, vol. 139, pp. 8821-8831, 2020. Article (CrossRef Link)

[7] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical Text-Conditional Image Generation with Clip Latents," *arXiv preprint arXiv:2204.06125*, 2022. Article (CrossRef Link)

[8] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, pp. 10684-19695, 2022. Article (CrossRef Link)

[9] S. Kang, Y. Ok, H. Kim, and T. Hahn, "Image-to-Image Translation Method for Game-Character Face Generation," in *Proc. of the IEEE Conf. on Games*, pp. 628-631, 2020. Article (CrossRef Link)

[10] S. Hong, S. Kim, and S. Kang, "Game Sprite Generator Using a Multi Discriminator GAN," *KSII Transactions on Internet and Information Systems*, vol. 13, No. 8, pp. 4255-4269, 2019. Article (CrossRef Link)

[11] T. Shi, Y. Yuan, C. Fan, Z. Zou, Z. Shi, and Y. Liu, "Face-to-Parameter Translation for Game Character Auto-Creation," in *Proc. of the IEEE/CVF International Conf. on Computer Vision*, pp. 161-170, 2019. Article (CrossRef Link)

[12] H. Kim, S. Lee, H. Lee, T. Hahn, and S. Kang, "Automatic Generation of Game Content using a Graph-based Ave Function Collapse Algorithm," in *Proc. of the IEEE Conf. on Games*, pp. 1-4, 2019. Article (CrossRef Link)

[13] G. Balakrishnan, A. Zhao, A. V. Dalca, F. Durand, and J. Guttag, "Synthesizing Images of Humans in Unseen Poses," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 8340-8348, 2018. Article (CrossRef Link)

[14] W. Liu, Z. Piao, J. Min, W. Luo, L. Ma, and S. Gao, "Liquid Warping GAN: A Unified Framework for Human Motion Imitation, Appearance Transfer and Novel View Synthesis," in *Proc. of the IEEE International Conf. on Computer Vision*, pp. 5904-5913, 2019. Article (CrossRef Link)

[15] T. Wang, M. Liu, A. Tao, G. Liu, J. Kautz, and B. Catanzaro, "Few-shot Video-to-Video Synthesis," in *Proc. of the 33rd International Conf. on Neural Information Processing Systems*, pp. 5013-5024, 2019. Article (CrossRef Link)

[16] D. Wei, X. Xu, H. Shen, and K. Huang, "GAC-GAN: A General Method for Appearance-Controllable Human Video Motion Transfer," *IEEE Transactions on Multimedia*, vol. 23, pp. 2457-2470, 2021. Article (CrossRef Link)

[17] A. Aberman, M. Shi, J. Liao, D. Lischinski, B. Chen, and D. Cohen-Or, "Deep Video-Based Performance Cloning," *Computer Graphics Forum*, vol. 38, no. 2, pp. 219-233, 2019. Article (CrossRef Link)

[18] C. Chan, S. Ginosar, T. Zhou, and A. Efros, "Everybody Dance Now," in *Proc. of the IEEE/CVF international Conf. on Computer Vision*, pp. 5932-5941, 2019. Article (CrossRef Link)

[19] L. Liu, W. Xu, M. Zollhoefer, H. Kim, F. Bernard, M. Habermann, W. Wang, and C. Theobalt, "Neural Rendering and Reenactment of Human Actor Videos," *ACM Transactions on Graphics*, vol. 38, no. 5, pp. 1-14, 2019. Article (CrossRef Link)

[20] Y. Sun, Y. Jiang, Z. Liu, Y. Lai, H. Fu, and L. Gao, "Human Motion Transfer With 3D Constraints and Detail Enhancement," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 4, pp. 4682-4693, 2023. Article (CrossRef Link)

[21] T. Wang, M. Liu, J. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution Image Synthesis and Semantic Manipulation With Conditional GANs," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 8798-8807, 2018. Article (CrossRef Link)

[22] G. Pons-Moll, S. Pujades, S. Hu, and M. J. Black, "ClothCap: Seamless 4D Clothing Capture and Retargeting," *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 1-15, 2017. Article (CrossRef Link)

[23] A. Raj, P. Sangkloy, H. Chang, J. Hays, D. Ceylan, and J. Lu, "SwapNet: Image Based Garment Transfer," in *Proc. of Computer Vision – ECCV 2018: 15th European Conf.*, pp. 679-695, 2018. Article (CrossRef Link)

[24] M. Zanfir, A. Popa, A. Zanfir, and C. Sminchisescu, "Human Appearance Transfer," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, pp. 5391-5399, 2018. Article (CrossRef Link)

[25] C. Zhang, S. Pujades, M. Black, and G. Pons-Moll, "Detailed, Accurate, Human Shape Estimation from Clothed 3D Scan Sequences," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 5484-5493, 2017. Article (CrossRef Link)

[26] V. Leroy, J. Franco, and E. Boyer, "Multi-view Dynamic Shape Refinement Using Local Temporal Integration," in *Proc. of the IEEE International Conf. on Computer Vision*, pp. 3113-3122, 2017. Article (CrossRef Link)

[27] R. Bern, A. Ghosh, T. Ajanthan, O. Miksik, N. Siddharth, and P. Torr, "A Semisupervised Deep Generative Model for Human Body Analysis," in *Proc. of the European Conf. on Computer Vision*, pp. 500-517, 2018. Article (CrossRef Link)

[28] P. Esser, E. Sutter, and B. Ommer, "A Variational U-Net for Conditional Appearance and Shape Generation," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 8857-8866, 2018. Article (CrossRef Link)

[29] C. Lassner, G. Pons-Moll, and P. V. Gehler, "A Generative Model of People in Clothing," in *Proc. of the IEEE International Conf. on Computer Vision*, pp. 853-862, 2017. Article (CrossRef Link)

[30] L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars, and L. V. Gool, "Pose Guided Person Image Generation," in *Proc. of the 31st International Conf. on Neural Information Processing Systems*, pp. 405-415, 2017. Article (CrossRef Link)

[31] L. Ma, Q. Sun, S. Georgoulis, L. V. Gool, B. Schiele, and M. Fritz, "Disentangled Person Image Generation," in *Proc. of the Conf. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 99-108, 2018. Article (CrossRef Link)

[32] A. Siarohin, E. Sangineto, S. Lathuiliere, and N. Sebe, "Deformable GANs for Pose-based Human Image Generation," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 3408-3416, 2018. Article (CrossRef Link)

[33] R. Villegas, J. Yang, Y. Zou, S. Sohn, X. Lin, and H. Lee, "Learning to Generate Long-term Future via Hierarchical Prediction," in *Proc. of the 34th International Conf. on Machine Learning*, vol. 70, pp. 3560-3569, 2017. Article (CrossRef Link)

[34] J. Walker, K. Marino, A. Gupta, and M. Hebert, "The Pose Knows: Video Forecasting by Generating Pose Futures," in *Proc. of the IEEE International Conf. on Computer Vision*, pp. 3352-3361, 2017. Article (CrossRef Link)

[35] A. Kanazawa, M. Black, D. W. Jaobs, and J. Malik, "End-to-end Recovery of Human Shape and Pose," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 7122-7131, 2018. Article (CrossRef Link)

[36] X. Qin, Z. Zhang, C. Huang, M. Dehghan, O. R. Zaiane, and M. Jagersand, "U^2-Net: Going Deeper with Nested U-Structure for Salient Object Detection," *Pattern Recognition*, vol. 106, 2020. Article (CrossRef Link)

[37] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” in *Proc. of International Conf. on Medical Image Computing and Computer-Assisted Intervention*, pp. 234-241, 2015. Article (CrossRef Link)

[38] Sega, “Space Harrier,” Sega corp., Arcade game, 1985.

Yongjun Choi, He received an BS degree from the School of Games (Game Software Major) at Hongik University in 2022. He is pursuing an MS degree at Hongik University.

Beomjoo Seo, He received the B.S. and M.S. degrees from the Department of Computer Engineering, Seoul National University in 1994 and 1996, respectively, and the Ph.D. degree in Computer Science from the University of Southern California in 2008. He was formerly a Senior Research Fellow at the School of Computing, National University of Singapore. He is currently an assistant professor at the School of Games in Hongik University.

Shinjin Kang, He received an MS degree at Korea University in 2003. After graduation, he joined Sony Computer Entertainment Korea (SCEK) as a video game programmer. From 2006, he has worked at NCsoft Korea as a lead game designer from 2009. He received a PhD degree in Computer Science and Engineering at Korea University in 2011. And he is now a professor at the school of games in Hongik University.

Jongin Choi, He received PhD at Korea University in 2016 from the Department of Computer Science from Korea University. After completion, he joined Nexon Korea as a lead client programmer. He has worked at NCSoft Korea as a lead animation programmer in a new AAA online game. Now he is a professor in the major of digital media in Seoul Women’s University.