2024 IEEE International Conference on Artificial Intelligence and eXtended and Virtual Reality (AIxVR) | 979-8-3503-7202-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/AIxVR59861.2024.00067

Character Animation Pipeline based on Latent Diffusion and Large Language Models

Alessandro Clocchiatti
Computer Science Department
University of Torino
Torino, Italy
alessandro.clocchiatti@unito.it

Nicolò Fumero
Computer Science Department
University of Torino
Torino, Italy
n.fumero@unito.it

Agata Marta Soccini
Computer Science Department
University of Torino
Torino, Italy
agatamarta.soccini@unito.it

***Abstract*—Artificial intelligence and deep learning techniques are revolutionizing the film production pipeline. The majority of the current screenplay-to-animation pipelines focus on understanding the screenplay through natural language processing techniques, and on the generation of the animation through custom engines, missing the possibility to customize the characters. To address these issues, we propose a high-level pipeline for generating 2D characters and animations starting from screenplays, through a combination of Latent Diffusion Models and Large Language Models. Our approach uses ChatGPT to generate character descriptions starting from the screenplay. Then, using that data, it generates images of custom characters with Stable Diffusion and animates them according to their actions in different scenes. The proposed approach avoids well-known problems in generative AI tools such as temporal inconsistency and lack of control on the outcome. The results suggest that the pipeline is consistent and reliable, benefiting industries ranging from film production to virtual, augmented and extended reality content creation.**

***Index Terms*—artificial intelligence, deep learning, generative art, virtual reality, extended reality, computer animation**

I. Introduction

Artificial Intelligence (AI) and deep learning techniques are revolutionizing many aspects of everyday life and industries, including film production and computer animation. Several approaches were proposed to simplify or improve film production, including storyboard generation [1], screenplay writing [2] and AI motion capture [3] among others. Some projects aimed to define an automatic pipeline to create scenes and animation starting from a screenplay. However, these projects mainly focused on screenplay understanding, which involves the use of Natural Language Processing (NLP) techniques to analyze and process the script's content, and on the animation and scene generation phases, where a visual representation of the script is created. Nevertheless, these systems miss the possibility of creating custom characters. Recent improvements of the Latent Diffusion Models (LDMs) such as Stable Diffusion[1], Dall-E[2] and Midjourney[3], allow the generation of images starting from text (known as text-to-image). However, generating videos or animation through these tools [4]–[6] still presents some difficulties, such as temporal inconsistency between consecutive frames and lack of control on the result.

The design of the characters in a screenplay-to-animation pipeline is typically based on pre-existing resources or custom character sheets, hand-drawn by artists or based on large datasets. Furthermore, screenplay understanding typically requires custom NLP algorithms that cannot be easily adapted to different screenplay structures. We questioned whether the integration of currently available generative AI tools and Large Language Models (LLMs, i.e. ChatGPT[4]) could support artists and animators with a reliable 2D animation pipeline starting from screenplay. This constitutes our research question (**RQ**). To address these issues, we propose a high-level pipeline to generate characters and animations from screenplays using LDMs and LLMs as backbones. Our approach uses ChatGPT for the screenplay understanding, to retrieve characters and the scene descriptions. Then, without the need to train custom models, it uses that data to generate custom characters with Stable Diffusion and animates them according to the different scene situations. During an initial feasibility study, we considered several animation styles and techniques and reviewed the high level generative tools available to consumers. While there are a few tools for the generation of 3D geometry from text whose capabilities are promising and evolving [7], [8], the technology is still immature. On the contrary, the generation of 2D images starting from text, still presents several flows but currently generates more accurate results that are closer to the input text. We then focused on 2D animation, in particular with a 2D cutout animation style, a specific type of animation that involves flat characters represented by layered artworks. Among the several existing 2D techniques, we chose this one because 1) it provides an easy way to animate the character, still providing plausible and enjoyable results, and 2) it does not require a large set of movements. The animation process involves both simple continuous deformations (e.g. the

The current work is part of the project "VR4Green: virtual reality solutions to promote environmental awareness and adoption of sustainable behaviors" CUP D11B21005890007. Supported by the National Operative Program (PON) on "Research and Innovation" 2014-2020 of the Italian Ministry of University and Research, and by HST Center (Human Sciences and Technologies) at University of Torino.

[1]https://stability.ai/stable-diffusion
[2]https://openai.com/dall-e-3
[3]https://www.midjourney.com/home
[4]https://openai.com/blog/chatgpt

2771-7453/24/$31.00 ©2024 IEEE
DOI 10.1109/AIxVR59861.2024.00067

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO PARANA. Downloaded on November 05,2025 at 18:58:45 UTC from IEEE Xplore. Restrictions apply.

bending of the elbow) and discrete transitions (e.g. an open hand replaced with a thumb-up pose) [9]. This pipeline avoids the typical temporal inconsistency and lack of control issues of the AI-based approaches in the character generation and animation phases of the pipeline.

II. Related works

Several projects proposed a pipeline to create 2D and 3D animations starting from screenplays leveraging custom NLP techniques [10], [11], which produce structured data in a text form, containing essential information from the script. These data are then fed to a game or animation engine to create a rendered scene. In this phase, these works mainly focused on positioning all the elements and characters in the scene [12], [13] and setting the camera movements [14]. However, these works used character resources that are pre-existing [10], [11], [15] or designed ad-hoc for the specific use cases [9], missing the opportunity to dynamically generate the characters according to the screenplay descriptions. In these terms, we aim to solve the screenplay understanding with pre-existing LLMs and the character generation through a combination of LDMs and LLMs without the need to train any custom base model.

Fable *et. al.* [16] proposed a pipeline (Show-1) to generate full custom episodes of the famous cartoon series *South Park*[5] starting from a text-prompt, defined by the users, that contains a synopsis and major events that should be included in the episode. Furthermore, a user interface allowed to specify the characters that should be included in the episode. These data are then sent to a simulation algorithm, developed by the authors, that generates the desired episode details, such as scenes and dialogues. This approach then used a Stable Diffusion base model pre-trained on the looks of the South Park characters to generate character sets against a keyable background. One limit of the approach is that it only focused on the generation of South Park stories and cannot be used in generic contexts or alternative scenarios. Furthermore, the animation of the character is pretty rough, due to the style of the animated show.

In the cutout animation generation scenario, Pose2Pose [9] is a 2D animation pipeline that, starting from a video source, generates a 2D animated character that mimics a particular human performance. By tracking human performance, this approach defines clusters of representative poses, each associated with a specific character artwork. Finally, the system animates the pre-defined poses to create a video. While the results include both continuous animation and discrete transitions, this tool still requires the artist to hand-draw the character artwork in different poses for the specific context and human performance.

III. Our Approach

In this Section we describe a high-level pipeline that, starting from a screenplay, creates 2D renders of animations using a combination of generative tools and deterministic scripts, lowering the manual intervention of animators. A classical 2D animation pipeline involves several professionals with different skills, mainly artists and technologists, such as, among others, artists to design draw the body parts of the characters and the environments, or animators to bring them to life. While the process takes several months of work for a few minutes long animation, our proposal potentially creates immediate results, leaving animators the possibility to change the aspects they do not like.

In our approach, we choose to use both generative and deterministic phases, to obtain a reliable output and guarantee the maximum level of control over the results for each phase.

Our pipeline consists of 3 main steps:

A. character generation;
B. character compositing and rigging;
C. character animation;

To further constrain the generative tools and to obtain the appropriate output in the non-AI steps, some steps of the pipeline require some hand-crafted elements.

Starting from a screenplay in a textual form, a LLM generates a textual description of the characters appearance, which is used as a prompt in the LDM for generating a character that follows the description. In this phase, the LDM generates an image containing the body parts in different poses. Then, after providing a 2D rig template (made by humans), a script takes as input the generated images and assigns the body parts to the rig as textures. Finally, the LLM selects the correct actions from a provided list for each scene of the screenplay. While the first two steps have already been developed and tested, the third one is a work in progress.

A. Character generation

To define a cutout animation-based character, the design of the body parts for each character pose is needed. We refer to this as a character sheet, which is, following the definition of Marx, a document that helps standardize the appearance, poses, and gestures of a character [17]. The character sheet should have the following requirements:

- it must contain all the body parts and the desired poses;
- the body parts and the poses must be divided from each other and positioned in the same portion of the sheet;
- each body part must have the same position and the same size between different character sheets within the specific portion.

In our approach, the generation of the character sheets is based on both ChatGPT and Stable Diffusion. The process started by uploading the script to ChatGPT and asked it to retrieve the list of characters present in the screenplay. In case of a large script, separate queries are used, splitting the text in several prompts. Then, for each character, ChatGPT is asked to generate a text that describes the character's appearance that will be used in Stable Diffusion. More specifically, here's the query provided to ChatGPT:

> *Giving the following prompt "Character sheet of a X y" change "X" with "male" or "female" and "y"*

[5] https://www.southparkstudios.com/

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO PARANA. Downloaded on November 05,2025 at 18:58:45 UTC from IEEE Xplore. Restrictions apply.

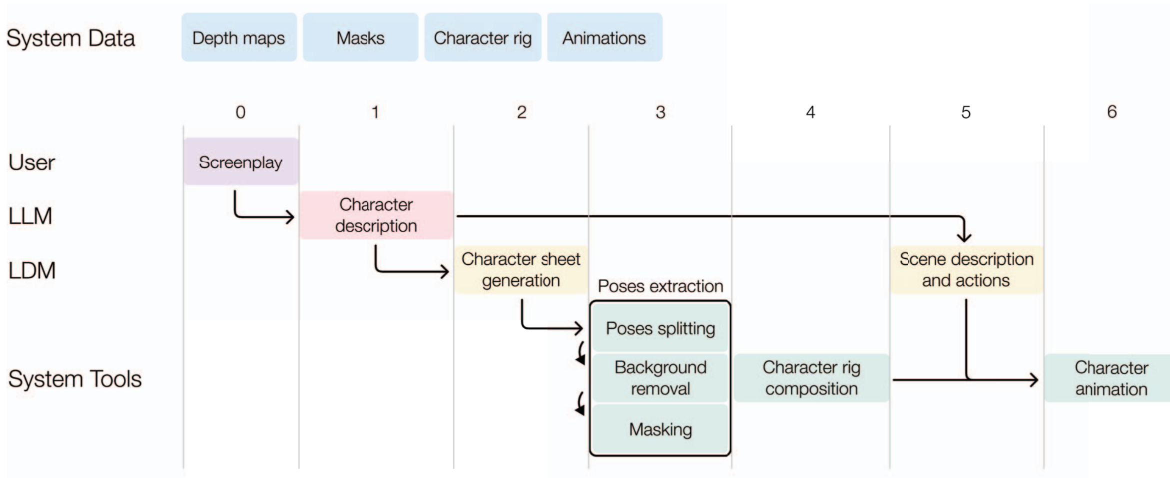


Fig. 1. Visual representation of the process used to create 2D characters and animations from screenplays.

with a very brief description of the physical appearance of the character. Do this for every character in the script. We need to use this prompt in Stable Diffusion, so it must be formatted exactly the way I told you.

This way a set of character descriptions in the desired format is obtained. Then, requirements related to the style and the background of the prompt were added as keywords. In particular, ”flat background” is used to generate the body parts in front of a uniform colour to make the keying of the elements easier in the next phase, while ”2D animation” is used to set the aesthetics of the output. This constituted the positive prompt. To maintain the consistency of the character's appearance, our system generates all body parts in one run. In fact, the generation of body parts in multiple sessions could introduce some variations in character clothing and body aspects. Furthermore, ControlNet[6] is used to force Stable Diffusion to generate the character sheets with a predefined structure: the same body part must always appears in the same sheet spots (creating a grid structure with each element in a different cell), divided by each other and with the same sizes. ControlNet is a neural network structure that controls the generation process by adding extra conditions. To generate the character sheet, two units of ControlNet were used:

- Lineart unit: it forces the model to create a grid-like output and to maintain the body parts separated from each other (Figure 2a);
- Depth unit: it guides Stable Diffusion in generating the body parts poses always in the same position and with the same sizes through the depth map of the specific body part (Figure 2b),

Both grids and the depth maps are handcrafted ad-hoc by the authors for this task and are included in the material of the system (Figure 1). Then, to increase the context of each pose, each depth map includes a partial view of the adjacent body part. At the end of this phase, an image of the character sheet is obtained (Figure 2c). As shown in Fig. 2b and 2c, the sheet contains several poses of the hands, but only two of the legs and the arms (side and front). Two images of arms and legs are enough because their animation does not need to be highly detailed to be believable, therefore their animation can be managed through the continuous interpolation of a shape deformation between the two images. Instead, we cannot deform the hands to achieve different poses since they have a highly detailed shape. We therefore need multiple hand poses to implement different discrete animations.

B. Character compositing and rigging

To produce a more reliable output, this phase is based on non-AI techniques, such as segmentation through predefined masks and replacing textures of a human-made rigged model template. Furthermore, the constraints defined in the generation phase do not require AI algorithms.

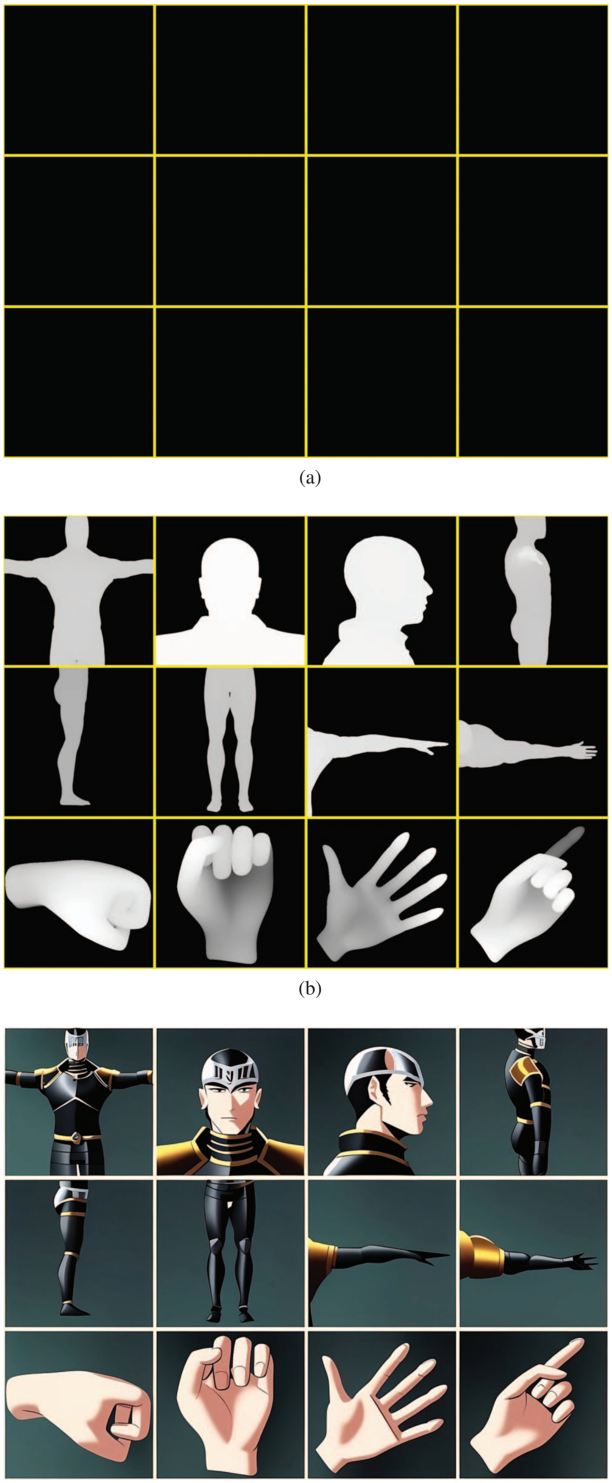
After generating the sheet, the specific parts of the body are automatically extracted. So, a script splits the sheet into cells and saves each one as an image. Then, it removes the background by using chroma-keying. Since a rig with the generated body parts is needed, a script mask each cell by removing the external unnecessary context. For example, in the generation of the torso, the partial view of the arm and head should be removed. To do so, a human-made mask image for each part of the body is created in Adobe Photoshop[7](Figure 3c), starting from the depth maps images. Then, the unwanted context is removed from the mask to have just the specific body part (i.e. for the cell that contains the arm, we remove the shoulder)

The mask image is a grey-scale square image that allows us to exclude any unwanted parts of the original image we want to mask. In particular, the pixel intensity value of the mask

[6]https://github.com/lllyasviel/ControlNet

[7]https://www.adobe.com/products/photoshop.html

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO PARANA. Downloaded on November 05,2025 at 18:58:45 UTC from IEEE Xplore. Restrictions apply.



(a)

(b)

(c)

Fig. 2. Template sheet images for the character sheet generation phase and the generation result (c). A black 4×3 grid is used to generate the body parts separated from each other (a), while a 4×3 grid with depth maps representing the body parts is used to constraint the position and size of the generation of the body parts (b).

image is mapped with the pixel alpha/opacity of the resulting image. So, when the pixel intensity value of the mask image is 1 (white), the pixel in the resulting image is kept, and when is 0 (black), the pixel in the final image is excluded. The in-between values correspond to a partially transparent pixel in the resulting image. The masked process produces a squared transparent image that only contains the specific part of the body, ready for the rigging phase (Figure 3d). To create a movable 2D character, a rig is needed, composed of several joints associated with different body parts. Since the resulting images obtained from the previous phase always have the same structure, the position and the size of the body parts are always the same. So, a hand-crafted rig template (developed in Autodesk Maya 2023[8]) composed of several planes, each associated with a body part of the character, could be provided and reused for each character. Each plane is pre-binded with the joints of its body part. For instance, the plane associated with the right arm is bonded to three joints of the skeleton: right shoulder, right elbow, and right wrist. This way, to rig a specific character, the body parts images should be assigned as textures of the corresponding planes of the rig.

The result of this phase is a fully rigged custom 2D character that can be animated according to specific needs.

C. Character animation

While the previously mentioned phases have been already developed and tested, this phase is under development. Here we describe the main steps of the process.

Once the model is rigged, it is possible to animate it manually or by using a set of custom animations from a library (as the rig structure is the same for each character). However, in the complete screenplay-to-animation pipeline, the animation will be inferred automatically from the screenplay.

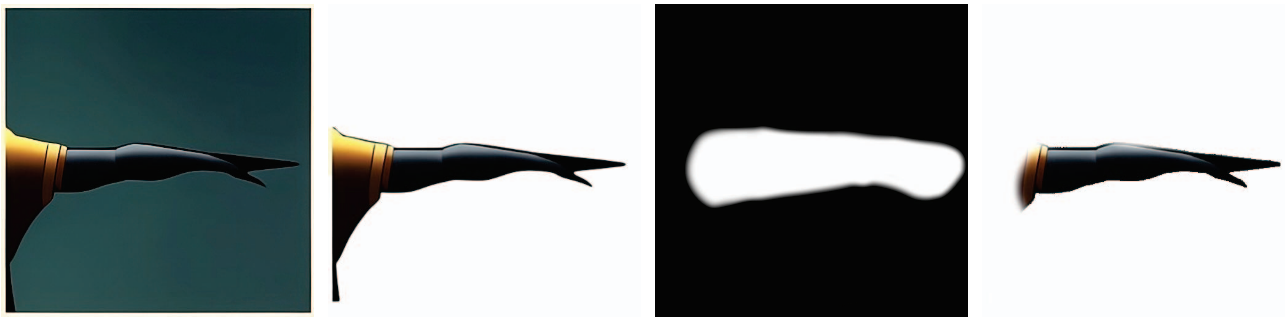
As mentioned earlier, the cutout animation consists of both continuous animation and discrete transitions. While the continuous animations are achieved by moving the joint's rig, the discrete transitions are performed by replacing the texture of the plane of the rig according to the needed pose.

The idea is to automatically animate the characters for each scene in the script. The animation needs to be robust and reliable, so we choose not to train a custom text-to-animation model as it could lead to sub-optimal movements of the rig. To solve this issue, our approach selects the correct animation from a large database of pre-defined animations and smooths the transitions between them. ChatGPT is used to infer the best animations of the characters present in each scene of the screenplay. Given a set of possible actions, ChatGPT is asked to split the scene in time-spans of n seconds and, for each time-span, provide a list of the actions that best fit the character's behavior in the screenplay at that time. For example, given the following section of the script:

> *Mark and Toby are talking while walking at the park. Toby sits on a bench while Mark is getting angry.*

[8]www.autodesk.it/products/maya

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO PARANA. Downloaded on November 05,2025 at 18:58:45 UTC from IEEE Xplore. Restrictions apply.



(a) (b) (c) (d)

Fig. 3. Steps required to produce a body part texture that can be imported into the model's rig. Starting with the generated image (a), the background is keyed (b), and an image mask is applied (c) to obtain the final version (d).

MARK
What in the world are you doing here?

and the following list of predefined actions:

[*walk, talk, greet, dance, sit*]

In a set time-span of 3 seconds, our system will generate the following output:

```
Mark
0-3 : talk - walk
3-6 : talk - walk

Toby
0-3 : talk - walk
0-6 : sit
```

Every action in the list is then assigned to a specific animation in the dataset, which can be imported on the character's rig.

To avoid the same actions to look all the same in the final animation, it should be possible to:

- have multiple variations of the animations in the database associated with the same action;
- introduce some randomness in the movement of the interested joints through python scripting in Autodesk Maya.

It is preferable to have the model reliable for the vast majority of the movement that the characters might do in different scenarios and possibly leave the opportunity to animate them by hand when an action is not present in our database. ChatGPT could select the keyword "Unsure" (instead of an action) in a specific time-span where it thinks that the action is not present in our DB.

IV. Use Case: a medieval knight

Currently, the pipeline consists of the character generation and rigging phases and, starting from a screenplay in a textual form, can provide the rigged models of the characters in the story. The tests confirm that our approach avoids the problems of temporal inconsistency, by directly animating the body parts through a rig, and lack of control, by guiding the results with scripts and tools to control the output of each phase.

To showcase our approach, we generated a character starting from a brief screenplay. We asked ChatGPT to write a brief screenplay of a medieval knight, and we hereby report the result.

EXT. FOREST ROAD - DAY

A dusty road winds through the ancient woods. The imposing figure of SIR GALLANT, a medieval knight in shining armor, emerges from the shadows. His armor glistens in the dappled sunlight as he strides purposefully towards the castle.
SIR GALLANT points his armored finger towards the towering castle in the distance, determination in his eyes.

SIR GALLANT (booming voice)
There lies the Castle of Eldoria! Duty calls!

CUT TO:
FOREST CLEARING - DAY
As SIR GALLANT continues his journey, he stumbles upon a YOUNG WOMAN, gracefully picking wildflowers. She looks up, captivated by the knight's noble presence.

YOUNG WOMAN (innocently)
Good sir knight, what brings you to these woods?
SIR GALLANT is enchanted by her beauty and sincerity.

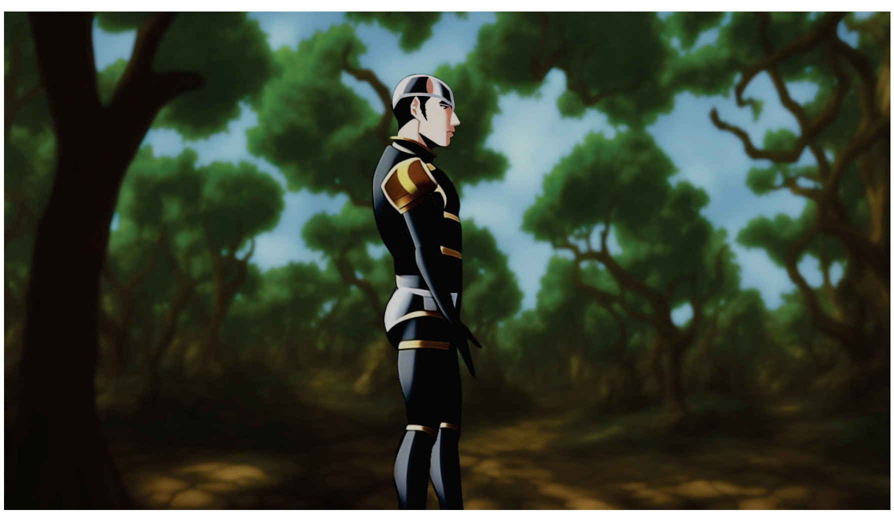
SIR GALLANT (softly)
I am bound for the Castle of Eldoria, but fate has led me to a far more precious encounter.

In a moment of inspiration, SIR GALLANT draws

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO PARANA. Downloaded on November 05,2025 at 18:58:45 UTC from IEEE Xplore. Restrictions apply.

(a)

(b)



(c)

Fig. 4. Resulting rigged character of a knight. The background was generated separately through Stable Diffusion.

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO PARANA. Downloaded on November 05,2025 at 18:58:45 UTC from IEEE Xplore. Restrictions apply.

a lute from his back, and with a tender smile, he begins to sing a ballad that echoes through the trees.

SIR GALLANT (CONT'D) (singing)
"In this realm of ancient lore, A tale unfolds forevermore. A knight, a maiden, hearts entwined, A love that fate itself designed."

The YOUNG WOMAN blushes, moved by the knight's romantic gesture.
YOUNG WOMAN (smiling) Your words are like a melody, noble knight.

CUT TO:
EXT. CASTLE GATES - DAY

SIR GALLANT, still basking in the warmth of newfound love, arrives at the towering gates of the Castle of Eldoria. The drawbridge lowers, and the castle guards stand at attention. SIR GALLANT, with the YOUNG WOMAN in his thoughts, walks through the entrance with purpose, ready to face the challenges that await within.

We then asked ChatGPT to generate a description for each character in the script, and hereby report the description of the main character.

Character sheet of a male beautiful medieval knight with a metal armour and black hair

We then added as keywords "2D animation" and "flat background" to define the aesthetics of the results and to provide a constraint to the background generation, obtaining the character sheet image (Figure 2c). A script then processed it and extracted each part of the body as images (Figure 3a. Each image is then keyed (Figure 3b) and masked to obtain the transparent images of each body part (Figure 3d). Finally, each body part is assigned as texture to the rigged template model, obtaining the final rigged character that can be animated in different poses (Figure 4a, 4b and 4c).

V. Discussion and Conclusions

We proposed and described a high-level pipeline to generate reliable 2D cutout animation starting from screenplays. Our results suggested its validity in automating the creation of simple animation scenes starting from textual screenplays, confirming our research question (RQ).

Our approach can easily adapt to different screenplay contexts, allowing the generation of specific character appearance features. This tool could be beneficial in facilitating the film production pipeline: it can generate rendered 2D cutout animation or could be adapted to make previsualization [18]. Furthermore, it could generate both the character and their animations for 2D games and XR content. As mentioned in Section III, in the character generation process it is possible to introduce some keywords to describe the aesthetic style of the character sheet. Then, it is also possible to further guide the generation of the images toward the chosen aesthetic using a pre-trained Stable Diffusion base model.

By eliminating the need for dedicated base model training, our approach not only optimizes the animation process but also minimizes energy consumption. This environmentally conscious strategy contributes to a greener animation production process, aligning with several initiatives aimed at fostering awareness and sustainable practices [19]–[21].

We conducted multiple tests on the character generation phase with different textual prompts, which produced a good variation in the character sheet results. The obtained results were consistent with the specified prompt, indicating the robustness and reliability of our method.

However, several aspects offer opportunities for improvement. Firstly, the proposed approach does not manage different shots for each scene and the characters initial positioning in the environment. Secondly, in our experiments we used only a limited number of body parts and poses. However, we can easily expand the depth map sheet by including more poses of the body parts in the grid template. Thirdly, our system does not manage the alteration of the mouth and other facial elements. This issue could be addressed by generating these elements using the *img2img* feature in Stable Diffusion. Finally, according to our results, the obtained characters always had a similar body shape. According to the features of the character (e.g. female, male, not-human), a set of different depth grid templates could be provided to emphasize the differences between characters.

References

[1] S. Jo, Z. Yuan, and S.-W. Kim, "Interactive storyboarding for rapid visual story generation," in *2022 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*. IEEE, 2022, pp. 1–4.

[2] P. Mirowski, K. W. Mathewson, J. Pittman, and R. Evans, "Co-writing screenplays and theatre scripts with language models: Evaluation by industry professionals," in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023, pp. 1–34.

[3] S. Laine, T. Karras, T. Aila, A. Herva, S. Saito, R. Yu, H. Li, and J. Lehtinen, "Production-level facial performance capture using deep convolutional neural networks," in *Proceedings of the ACM SIGGRAPH/Eurographics symposium on computer animation*, 2017, pp. 1–10.

[4] H. Huang, Y. Feng, C. Shi, L. Xu, J. Yu, and S. Yang, "Free-bloom: Zero-shot text-to-video generator with llm director and ldm animator," *arXiv preprint arXiv:2309.14494*, 2023.

[5] J. Xing, M. Xia, Y. Liu, Y. Zhang, Y. Zhang, Y. He, H. Liu, H. Chen, X. Cun, X. Wang *et al.*, "Make-your-video: Customized video generation using textual and structural guidance," *arXiv preprint arXiv:2306.00943*, 2023.

[6] Y. Guo, C. Yang, A. Rao, Y. Wang, Y. Qiao, D. Lin, and B. Dai, "Animatediff: Animate your personalized text-to-image diffusion models without specific tuning," *arXiv preprint arXiv:2307.04725*, 2023.

[7] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, "Dreamfusion: Text-to-3d using 2d diffusion," *arXiv*, 2022.

[8] C.-H. Lin, J. Gao, L. Tang, T. Takikawa, X. Zeng, X. Huang, K. Kreis, S. Fidler, M.-Y. Liu, and T.-Y. Lin, "Magic3d: High-resolution text-to-3d content creation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[9] N. S. Willett, H. V. Shin, Z. Jin, W. Li, and A. Finkelstein, "Pose2pose: Pose selection and transfer for 2d character animation," in *Proceedings of the 25th International Conference on Intelligent User Interfaces*, 2020, pp. 88–99.

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO PARANA. Downloaded on November 05,2025 at 18:58:45 UTC from IEEE Xplore. Restrictions apply.

[10] S. Das, S. N. Kumar, S. Yepuri, Y. Ujwal, and R. Srinath, “Storytube-generating 2d animation for a short story,” in *2023 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*. IEEE, 2023, pp. 91–96.

[11] Y. Zhang, E. Tsipidi, S. Schriber, M. Kapadia, M. Gross, and A. Modi, “Generating animations from screenplays,” in *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 292–307. [Online]. Available: https://aclanthology.org/S19-1032

[12] T. Gupta, D. Schwenk, A. Farhadi, D. Hoiem, and A. Kembhavi, “Imagine this! scripts to compositions to videos,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 598–613.

[13] K. Jorgensen, H. Wang, and M. Wang, “From screenplay to screen: A natural language processing approach to animated film making,” in *2023 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2023, pp. 484–490.

[14] Z. Yu, H. Wang, A. K. Katsaggelos, and J. Ren, “A novel automatic content generation and optimization framework,” *IEEE Internet of Things Journal*, 2023.

[15] H. Subramonyam, W. Li, E. Adar, and M. Dontcheva, “Taketoons: Script-driven performance animation,” in *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, 2018, pp. 663–674.

[16] Maas, Carey, Wheeler, Saatchi, Billington, and Shamash, “To infinity and beyond: Show-1 and showrunner agents in multi-agent simulations,” *arXiv preprint*, 2023.

[17] C. Marx, *Writing for animation, comics, and games*. Routledge, 2012.

[18] Y. Jung, S. Wagner, C. Jung, J. Behr, and D. Fellner, “Storyboarding and pre-visualization with x3d,” in *Proceedings of the 15th International Conference on Web 3D Technology*, 2010, pp. 73–82.

[19] V. Masson-Delmotte, P. Zhai, H.-O. Pörtner, D. Roberts, J. Skea, P. R. Shukla *et al.*, *Global Warming of 1.5° C: IPCC Special Report on Impacts of Global Warming of 1.5° C above Pre-industrial Levels in Context of Strengthening Response to Climate Change, Sustainable Development, and Efforts to Eradicate Poverty*. Cambridge University Press, 2022.

[20] A. Clocchiatti, F. Cena, and A. M. Soccini, “Vr4green-walk through the (visual) effects of climate change,” in *Proceedings of the 15th Biannual Conference of the Italian SIGCHI Chapter*, 2023, pp. 1–3.

[21] M. Coeckelbergh, “Ai for climate: freedom, justice, and other ethical and political challenges,” *AI and Ethics*, vol. 1, no. 1, pp. 67–72, 2021.

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO PARANA. Downloaded on November 05,2025 at 18:58:45 UTC from IEEE Xplore. Restrictions apply.