Multimedia Tools and Applications
https://doi.org/10.1007/s11042-024-19904-3

ASAP for multi-outputs: auto-generating storyboard and pre-visualization with virtual actors based on screenplay

Hanseob Kim[1,2] · Ghazanfar Ali[1] · Bin Han[1,3] · Hwang Youn Kim[1] · Jieun Kim[1] · Hyemin Shin[2] · Gerard Jounghyun Kim[2] · Jae-In Hwang[1]

Received: 30 December 2023 / Revised: 9 June 2024 / Accepted: 18 July 2024
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2024

Abstract
One of the pressing desires of content creators is to be able to visualize how their characters will look in a scene as soon as possible. In the early stages of film production, this desire can be partly achieved by the computer graphics-based process known as *Pre-visualization* (*Previz*). However, traditional previz necessitates a high level of expertise and is also time-consuming. This paper introduces the ASAP system, an automated tool that creates pre-visualized animations and storyboards by generating virtual character behavior/animations based on understanding the screenplay. The ASAP system parses the user-written screenplay to extract data, including character names, dialogue, actions, and emotions. This extracted data is then passed to the respective modules, which select virtual characters and automatically generate their speaking gestures, physical movements, and expressive behaviors. We demonstrate the system's fidelity by presenting multiple outputs, including a 2D storyboard, a 3D preview, and a VR-based immersive scenario, along with simulations of potential use cases. The ASAP system can streamline pre-visualization tasks in the pre-production phase and has the potential to be widely adopted by the film industry.

Keywords Pre-visualization · Previz · Character animation · Natural language processing

1 Introduction and motivation

The creation and consumption of films have emerged as prominent cultural and leisure pursuits in contemporary society, leading to a substantial expansion of the film industry. As the demands of audiences for films, encompassing both superior quality and diverse content, have expanded, filmmakers have adopted a range of novel production technologies (e.g., computer graphics, motion tracking [26], avatar digitization [59], virtual/augmented reality [49, 60]) to enhance the overall cinematic experience.

One significant innovation in film production is the introduction of *Pre-visualization*, referred to as *Previz* [17]. This technique emerged from storytellers' natural desire and the necessity to envision how their characters would be visually portrayed on the screen well in advance of actual filming. Previz involves the creation of an animated storyboard

Hanseob Kim and Ghazanfar Ali are both equally contributed to this work.

Extended author information available on the last page of the article

Published online: 03 August 2024

that simulates 3D characters as virtual actors alongside 3D graphical props (or mockups) within graphical environments, providing a visual representation of the envisioned scene. By leveraging previz tools into the production phase, filmmakers can improve communication and proactively identify potential issues that might arise during the actual shooting [15]. Thus, previz is a highly effective and cost-efficient tool in comparison to traditional methods such as hand-drawn annotated storyboards and environment/set sketches, as well as manually enacting acting, camera movements, and props usage. This, in turn, reduces the need for re-shoots during production, resulting in cost savings and reduced crew fatigue.

Nevertheless, it is common for filmmakers to face constraints in terms of both resources and expertise in 3D animation and programming, which are essential for the efficient and timely production of previz [40, 41]. Previz often demands substantial labor, finances, and time investments. Consequently, despite the multitude of advantages previz offers, its implementation and full utilization continue to pose challenges, particularly for low-budget or independent filmmakers.

This paper presents a pre-visualization system that can automatically and efficiently generate pre-visualization scenes from input screenplay text. Our work aims to bridge the gaps in existing pre-viz systems, which often require significant manual labor and expertise in 3D animation and programming, and face limitations in terms of the diversity of characters, actions, props, and environments that can be visualized.

There have been many works, in the domain of the pre-viz, that have addressed and attempted to solve different sub-problems. For example, the ANSWER framework [8, 25] and recent mixed-reality based tools by [23, 24, 52, 54] represent advancements in the GUI-based story-boarding and augmented reality for pre-visualization, respectively. The “One Man Movie” [15, 16] is another significant leap which incorporates the aspects of VR as part of a comprehensive pre-viz scene authoring system, although animating 3D virtual characters remains challenging, which often is an integral part of any compelling 3D/VR content.

In this context, the efforts to fully or partially automate character animation generation directly from storyboards or text, such as CANVAS [27] and CARDINAL [38], show immense progress in reducing the usual manual authoring. However, these systems still require quite amount of domain-specific knowledge and user input, indicating a reliance on design experts. Moreover, automated approaches are still limited in supporting for sufficient diversity in the types of characters, props, and actions, and also still suffers from various domain-specific constraints. [5, 27, 38].

An effective pre-viz system should be able to take a text-based story, understand its content, and control the scene by providing a selection of characters and environments. The system should offer a multi-camera view and be able to simulate the entire story with minimal user interaction. The characters should display a full range of facial emotions and body motions and be capable of interacting with props and the environment. Additionally, the system should be able to save the entire simulation and easily take multi-view screenshots.

A suitable approach would simplify the overall authoring process using a mix of off-the-shelf specialized systems and modules through a streamlined pipeline. For instance, a movie script authoring software could be used for the pre-viz system so as to convert the formatted story and output a structured text as XML for later parsing and semantic/structural understanding. A combination of rule-based and learning-based approaches can be used to understand the content of XML - e.g. a rule set to extract emotion, speech, and action instructions, and also a more advanced LLM-based system for further emotion recognition, co-speech gesture generation, and interaction recognition.

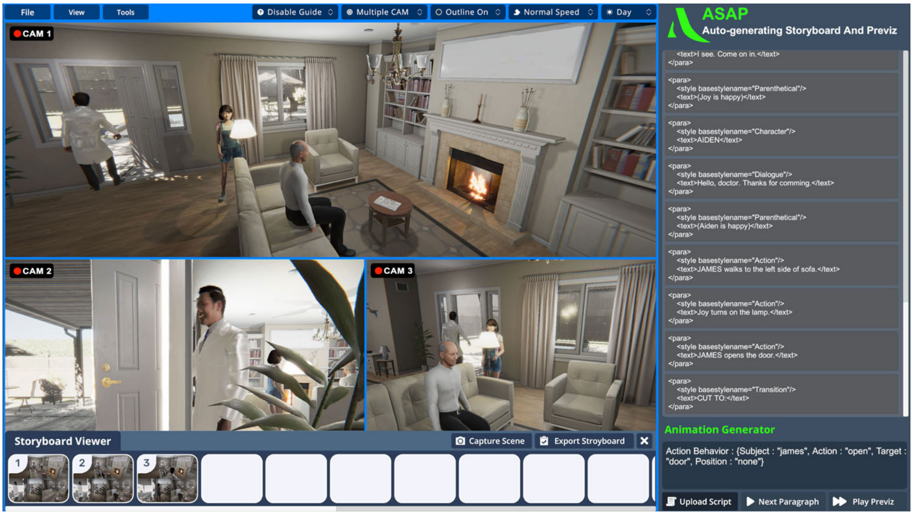
In a screenplay, the animated simulation of the virtual characters is particularly important and, moreover, it must include facial emotions, co-speech gestures, action emphasis, and interaction in the virtual environment (screenplay set). The correct rendition of virtual characters interacting (physically) with virtual objects/environments is tricky, and our approach involves adding particular plausible object/prop specific interactive actions (e.g. for a door - opening, closing, knob turning, for a vase - watering a plant). In addition, using the inverse kinematics, this could be generalized to differently sized characters effectively.

This paper introduces the *Auto-generating Storyboard And Previz* (referred to as *ASAP*) system that can help users produce the previz without any specialized knowledge of 3D animation and programming. The ASAP system analyzes a user-written screenplay based on the latest natural language processing (NLP) technology, automatically generates procedural animations of virtual characters (as virtual actors) with the virtual props (e.g., lamp, door, and sofa), and visualizes the results within the pre-made virtual environment (see Fig. 1). Consequently, users, particularly screenwriters and filmmakers, can easily and concretely envision how the screenplay will unfold and be actually portrayed to the audience.

The following are summarized contributions of this paper:

- We present the design of the ASAP system, which aims to pre-visualized animation scenes by understanding the screenplay.
- We propose efficient automatic 3D animation generation process for virtual characters, especially co-speech gestures, facial expressions, and physical actions.
- We demonstrate the feasibility of the ASAP system by providing immersive use cases and simulating possible scenarios.

The proposed ASAP system, offering the automatic pre-visualization of and animated acting, has a great potential for the widespread adoption in virtual production and filmmaking.The rest of this paper is organized as follows: Section 2 reviews relevant literature. Section 3 presents the overview of the ASAP system and the technical details. Sections 4



```
   <text>I see. Come on in.</text>
</para>
<para>
   <style basestylename="Parenthetical"/>
   <text>(Joy is happy)</text>
</para>
<para>
   <style basestylename="Character"/>
   <text>AIDEN</text>
</para>
<para>
   <style basestylename="Dialogue"/>
   <text>Hello, doctor. Thanks for comming.</text>
</para>
<para>
   <style basestylename="Parenthetical"/>
   <text>(Aiden is happy)</text>
</para>
<para>
   <style basestylename="Action"/>
   <text>JAMES walks to the left side of sofa.</text>
</para>
<para>
   <style basestylename="Action"/>
   <text>Joy turns on the lamp.</text>
</para>
<para>
   <style basestylename="Action"/>
   <text>JAMES opens the door.</text>
</para>
<para>
   <style basestylename="Transition"/>
   <text>CUT TO:</text>
</para>
```

```
Action Behavior : {Subject : "james", Action : "open", Target : "door", Position : "none"}
```

Fig. 1 The graphical user interface (GUI) of the ASAP system. The screenplay is divided into paragraphs and displayed on the right side of the GUI, while the analyzed and extracted words are displayed below it. The multiple cameras show a pre-visualized scene, exhibiting an elderly man seated on a sofa, a young woman turning on a lamp, and a man opening a door. In addition to offering a storyboard viewer, the system allows users to create a storyboard by capturing pre-visualized scenes with virtual actors

and 5 showcase the output of the ASAP system, and discusses the main findings, limitations, and future works. Finally, Section 6 concludes and summarizes the paper.

2 Related works

This section provides an overview of previz tools, primarily in the research domain, that share similar goals and concepts with the ASAP system. Moreover, prior research on virtual character behavior generation, a key component of the ASAP system, is examined.

2.1 3D pre-visualization and storyboard for virtual previz

Pre-visualization and storyboarding play crucial roles in the production of films and visual content, but they can be challenging to implement due to a range of constraints, such as limitations in expertise, financial resources, and time availability [25, 57]. As a result, there is a well-defined research direction on how to more efficiently create and improve their functionality by integrating diverse technologies. Table 1 shows the list of automated/manual previz systems and comparative analysis with the ASAP system.

Jung et al. [8, 25] introduced the *ANSWER* framework based on the X3D to depict 3D characters and scene elements for storyboarding and pre-visualization. The framework provides users with a GUI for writing visual descriptions, the conceptual level of a film's plot. The framework then constructed rule-based models utilizing semantic web technologies in order to visualize and animate virtual agents.

On the basis of maker-based augmented technology, cost-effective systems were developed that allow filmmakers to overlay virtual mockups in real space for storyboarding and previz [52, 54]. Moreover, by seamlessly merging the real and virtual worlds, Ichikari et al. [23, 24] presented a previz tool to improve camera-work authoring, with a focus on fostering camera operations, action rehearsals using motion-captured data, and streamlining on-set workflow.

Galvane et al. [15, 16] introduced *One Man Movie*, a VR-based previz authoring system. While mirroring the processes of film pre-production (e.g., scene layout, character animation, and camera control), they simplified the iterative creative stages from idea conception to physical mockup creation for previz into an all-in-one process, utilizing a VR-based real-

Table 1 Comparative analysis of the key features of the proposed ASAP system and existing automated previz systems. Hybrid means manual and automatic operations

	Virtual Actors				Previz function		
Works	Type	Gesture	Face	Body	Outputs	Timeline	Camera
CARDINAL [38]	2D, 3D	ADAPT [53]	✗	ADAPT [53]	2D, 3D	✓	Manual
TakeToons [56]	2D	Mocap	Mocap	Manual	2D	✓	✗
WordsEye [9]	3D	Rulebased	✗	Rulebased	3D	✗	Hybrid
CANVAS [27]	3D	ADAPT [53]	✗	ADAPT [53]	3D	✗	Manual
One Man Movie [15]	3D	Mocap	✗	Mocap	VR	✓	Hybrid
Cine-AI [13]	3D	Mocap	Mocap	Mocap	3D	✓	Rulebased
ShotPro HQ [60]	3D	Mocap	Mocap	Mocap	3D, VR, AR	✓	Manual
Our ASAP	3D	Automatic	Automatic	Automatic	2D, 3D, VR	✗	Manual

time editing metaphor. Nevertheless, animating 3D/virtual characters still demands expertise, particularly in the realm of motion capture systems [16, 23].

To address these issues, several studies have focused on automating animation generation based on storyboards and/or text scripts such as *CANVAS* [27], *CARDINAL* [38], *WordsEye* [9], and *TakeToons* [56]. The *CANVAS* system [27] utilizes storyboards as a visual input parameter. When users specify key events within the storyboard, this system can rapidly synthesize 3D animations with virtual characters. The system uses the ADAPT framework [53], which incorporates character animation, navigation, and behavior. However, this system is customized for domain-specific knowledge, requires user input, and continues to rely on storyboarding. In other words, design experts are still needed.

The *CARDINAL* system [38] is basically a script-writing tool based on the newly designed timeline view. As one of the functionality, they presented a 3D preview with 3D characters. The system employs a linguistic format to analyze text scripts based on the “subject-verb-object." The analyzed data is utilized to generate affordances, which define possible body actions within a graphic environment (i.e., it may only be possible with a restricted or pre-prepared list of animations and virtual props). The Cardinal system uses the ADAPT framework [53] to generate behavior trees based on affordances.

Consequently, numerous studies still continue to utilize manual authoring approaches [16, 25, 27]. Even in cases where they integrate automated generation, there are still limitations in terms of diversity, such as domain-specific constraints and the absence of props, actions, and actors in the animator’s datasets [5, 27, 38].

The ASAP system introduces an effective approach that requires no professional knowledge to animate virtual agents’ faces and gestures to overcome these challenges. As shown in Table 1, we implemented the complete automation of generating animations so that users can focus more on their storytelling. ASAP system enables creators to generate previsualization or content efficiently by automatically producing animated scenes, such as relationships between characters and the psychology of characters.

2.2 Virtual character behavior generation

Automating character animation can significantly reduce the labor-intensive and expertise-dependent aspects of previz [8, 28]. This subsection reviews generation tasks such as gesticulation and simulating physical actions, which can further improve the performance of previz tools featuring the behavior of virtual characters.

For gesticulation, earlier text-to-gesture systems were rule-based. Notable advancements include Cassell et al.’s conversational agent animation system [6] and the Behavior Expression Animation Toolkit (BEAT) [7], which produces nonverbal behaviors from text. Stone et al. combined motion and speech for character animation [55]. Later research [32, 33, 44] utilized real-time behavior generation, statistical models, and gesture controllers. Advanced methods [4, 37, 51, 61] incorporated audio signals, neural networks, and machine learning for gesture creation. Some, like Ali et al. [2], used video-based pose data to generate high-quality gestures.

For physical behavior generation, many researchers have been interested in finding an efficient way to generate animations with natural language processing(NLP) and animation data [31, 36, 45]. Recently, researchers are exploring the utilization of deep learning techniques for text-to-animation generation [18–21, 29, 34]. Hong et al. [20] presented AvatarCLIP, the text-driven framework to create 3D avatars and animations using the vision-language model CLIP. Ghosh et al. [18] developed a hierarchical two-stream sequential model

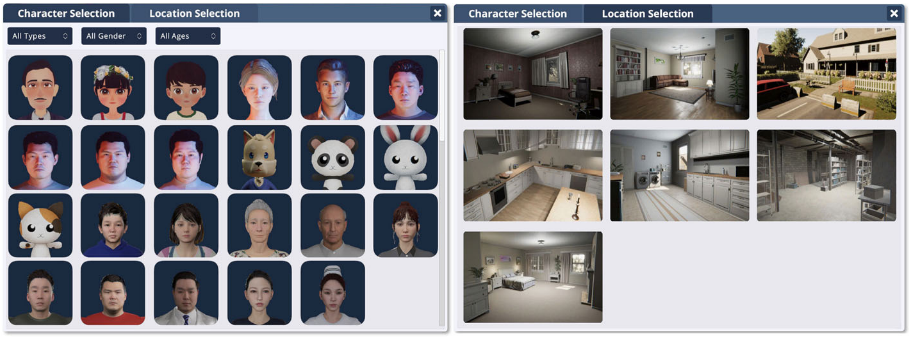


Fig. 2 A total of 23 virtual characters and 7 virtual environments

to generate 3D animations from intricate natural language sentences. Such deep learning-based methods often demand extensive computational resources and may not be suitable for planning-stage applications, i.e., previz, which require iterative editing.

Consequently, prior research has not only emphasized the need to explore the substitution of manual editing with an automated system in previz but also emphasized the importance of this system being efficient for real-time applications. In this paper, we present the ASAP system, which successfully fulfills the specified requirements.

3 Overview of the ASAP system

The graphical user interface (GUI) of the ASAP system, signifying *"Auto-generating Storyboard And Previz,"* is depicted in Fig. 1. The screenplay, segmented into paragraphs, is displayed on the right-hand side of the GUI. Below this, the analyzed outcomes, namely the extracted words intended for behavior generation, are presented. Furthermore, a multi-view screen within the GUI showcases a pre-visualized animation, e.g., featuring three key scenes: an elderly man seated on a sofa, a young woman turning on a lamp, and a man opening a door (see Fig. 1).

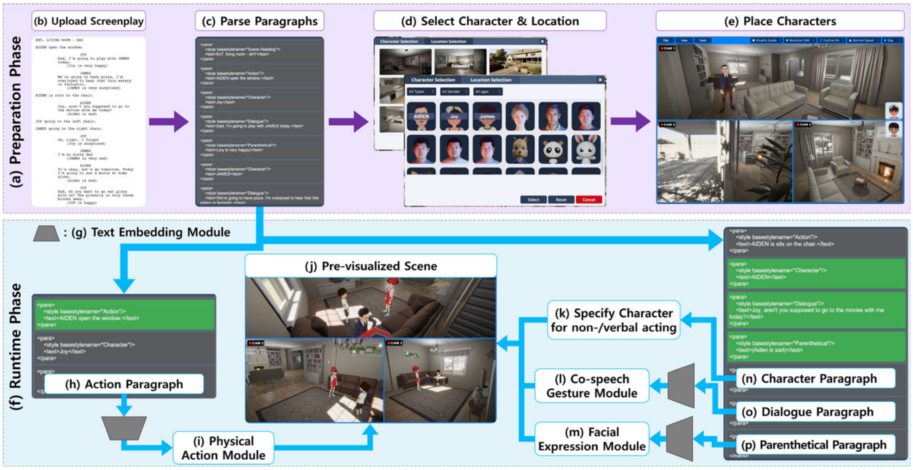
In order to accommodate a wide range of storytelling, a lot of resources for previz generations were prepared. This includes seven distinct virtual environments, including living rooms, children's bedrooms, outdoor landscapes, cellars, and more (see Fig. 2). These settings are encompassed with a wide assortment of virtual props, including beds, chairs, and windows, to further enrich previz generations. Furthermore, the 23 distinct virtual characters of various types/appearances, such as photo-realistic, cartoonish, and animal anthropomorphic designs, can be selected by the user according to the screenplay. It should be noted that the character animation generation technique and module of the ASAP can be applied to any 3D model as long as it includes the commonly used humanoid-type rigging setup. ASAP can import and use characters created with most of the popular 3D modeling tools such as Maya[1], Character Creator[2], and VRoid Studio[3], as well as humanoid-type models available from Mixamo [39] and the Unity Asset Store[4]

[1] https://www.autodesk.co.kr/products/maya/

[2] https://www.reallusion.com/character-creator/

[3] https://vroid.com/en/studio/

[4] https://assetstore.unity.com/



```
EXT. LIVING ROOM - DAY

AIDEN open the window.

                    JOY
          Dad, I'm going to play with JAMES
          today.
               (Joy is very happy)

                    JAMES
          We're going to have pizza. I'm
          overjoyed to hear that this eatery
          is fantastic.
               (JAMES is very surprised)

AIDEN is sits on the chair.

                    AIDEN
          Joy, aren't you supposed to go to
          the movies with me today?
               (Aiden is sad)

JOY going to the left chair.

JAMES going to the right chair.

                    JOY
          Oh, right. I forgot
               (Joy is surprised)

                    JAMES
          I'm so sorry dad
               (JAMES is very sad)

                    AIDEN
          It's okay, Let's go tomorrow. Today
          I'm going to see a movie at home
          alone.
               (Aiden is sad)

                    JOY
          Dad, do you want to go eat pizza
          with us? The pizzeria is only three
          blocks away.
               (JOY is happy)
```

```
<para>
   <style basestylename="Scene Heading"/>
   <text>ExT. living room - dAY</text>
</para>
<para>
   <style basestylename="Action"/>
   <text>AIDEN open the window.</text>
</para>
<para>
   <style basestylename="Character"/>
   <text>Joy</text>
</para>
<para>
   <style basestylename="Dialogue"/>
   <text>Dad, I'm going to play with JAMES today.</text>
</para>
<para>
   <style basestylename="Parenthetical"/>
   <text>(Joy is very happy)</text>
</para>
<para>
   <style basestylename="Character"/>
   <text>JAMES</text>
</para>
<para>
   <style basestylename="Dialogue"/>
   <text>We're going to have pizza. I'm overjoyed to hear that this eatery is fantastic.</text>
```

```
<para>
   <style basestylename="Action"/>
   <text>AIDEN open the window.</text>
</para>
<para>
   <style basestylename="Character"/>
   <text>Joy</text>
</para>
<para>
```

```
<para>
   <style basestylename="Action"/>
   <text>AIDEN is sits on the chair.</text>
</para>
<para>
   <style basestylename="Character"/>
   <text>AIDEN</text>
</para>
<para>
   <style basestylename="Dialogue"/>
   <text>Joy, aren't you supposed to go to the movies with me today?</text>
</para>
<para>
   <style basestylename="Parenthetical"/>
   <text>(Aiden is sad)</text>
</para>
```

Fig. 3 The overall process of the ASAP system is to understand/interpret the screenplay and subsequently generate (or animate) the behavior of virtual characters in the 3D pre-visualization. The first line (purple box) illustrates the (a) Preparation phase of the ASAP system, which consists of the following steps: (b) The written screenplay is uploaded to the ASAP system; (c) The ASAP parses the screenplay into different paragraphs, including *Action, Character, Dialogue, and Parenthetical*; (d) Selecting a location and characters to demonstrate a screenplay; and (e) Placing characters in the selected location. In the second line (blue box), the (f) Runtime phase of the ASAP system is displayed, and (j) previz is sequentially generated by the screenplay. The paragraph of the screenplay is sequentially passed to each pertinent module: (h) *Action* paragraph passed to the (i) physical action module; (n) *Character* paragraph is used to (k) specify the target for applying the animation generated from the following paragraph; (o) *Dialogue* paragraph passed to the (l) co-speech gesture module; (p) *Parenthetical* paragraph pass to the (m) facial expression module; and lastly, (g) the grey trapezoid indicates the module for text embedding

In the subsequent subsections, technical aspects for creating the previz are described as follows: the detailed procedure and pipeline for overall screenplay interpretation, the module for text embedding/interpretation, and modules for the generation of speech and gestures, facial expressions, and physical actions, all tailored for virtual actors (i.e., 3D characters). Following this, an objective evaluation of each module is provided.

3.1 Overall process for generation of previz animation

The ASAP system's process is illustrated in Fig. 3, incorporating two primary phases: the preparation phase, highlighted on the top line of Fig. 3(a), is dedicated to the interpretation of screenplays, whereas the runtime phase, depicted on the bottom line of Fig. 3(f), focuses on the generation of previz animation.

In the preparation phase (see Fig. 3(a)), the ASAP system tasks the screenplay as an input (see Fig. 3(b)), which has been written by screenwriters using the widely-used screenwriting software, Final Draft [14]. The Final Draft software enables the organization of screenplays in a well-structured format and allows for the extraction of the content in XML format. This formalized input makes it easy to extract paragraph types, and the ASAP system employs four distinct types of paragraphs[5], including *"Character," "Dialogue," "Parenthetical," and*

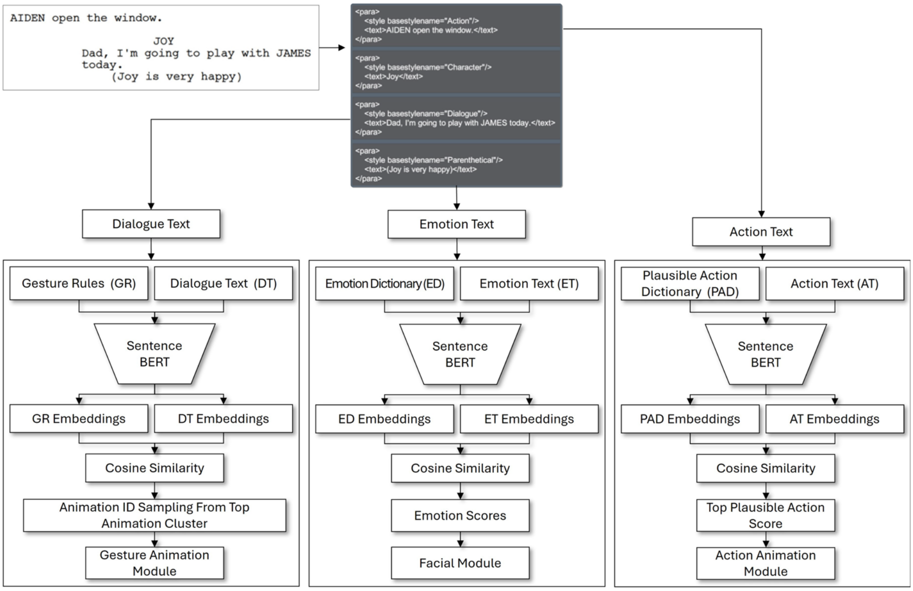
[5] Other paragraph types are *"Scene Heading", "Transition", "Shot", "Cast List", "New Act", "End of Act", "Teaser/Act One", and "Show/Ep. Title"*, etc. These are not suitable sources for automatically generating previz.

"Action" (see Fig. 3(c)). The next step involves the user selecting a specific virtual environment scene (pre-built based on the screenplay) and virtual characters for the screenplay to be played out (see Fig. 3(d)). The final step of the preparation is for the user to place the selected characters into their respective positions within the selected scene (see Fig. 3(e)).

In the run-time phase (see Fig. 3(f)), each paragraph serves as a parameter to aid in generating animations for virtual characters, such as gesticulation and physical action that reflect the screenplay. For this, the text embedding (encoding process indicated in trapezoids in Fig. 3(g)) is employed to analyze and comprehend the semantics of different types of paragraphs. Note that character-type paragraphs do not require text embedding. This is because it is simple and contains only character names in standard screenplay format, e.g., in XML format, it appears as <Text> AIDEN </Text>. These character-type paragraphs are used to identify the names of characters within the visualized scene, and during the preparation phase, they assist in offering user options for selecting specific virtual character models to portray each role in the previz output (see Fig. 3(d)). Figure 4 shows the text processing module taking the input text and generating appropriate outputs to other modules down the overall animation production pipeline (see Fig. 3). The following describes the ASAP system's components and the animation generation process for each paragraph type.

3.1.1 Text embedding module

The text understanding and interpretation module uses NLP techniques to extract the context and meaning of the input screenplay. Text embedding is a method to represent text as numbers. It maps each word or phrase to a high-dimensional vector that captures its meaning and context. Such embeddings serve as input for various task-specific NLP tasks and enable mathematical operations, e.g., finding similarities between texts. In considering of the proper



```
AIDEN open the window.

                    JOY
          Dad, I'm going to play with JAMES
          today.
              (Joy is very happy)
```

```
<para>
  <style basestylename="Action"/>
  <text>AIDEN open the window.</text>
</para>
<para>
  <style basestylename="Character"/>
  <text>Joy</text>
</para>
<para>
  <style basestylename="Dialogue"/>
  <text>Dad, I'm going to play with JAMES today.</text>
</para>
<para>
  <style basestylename="Parenthetical"/>
  <text>(Joy is very happy)</text>
</para>
```

Fig. 4 The input/output and the inner workings of each text processing module at runtime before sending output to each animation module

text representation techniques, we looked into two types of embeddings: word embeddings and sentence embeddings. We began by examining the GloVe [46] word embedding model and BERT [11] for contextual understanding. After conducting internal testing, we empirically found that BERT provided better context sensitivity. We then evaluated various models from the Sentence-BERT [47] library and discovered that they outperformed the base BERT model. To determine the most effective models in terms of performance and speed, we also consulted the Massive Text Embedding Benchmark [42]. The Sentence-BERT library models, specifically designed for sentence-level embeddings, proved to be more suitable for our purposes. Depending on the specific NLP task we are working on, we may select the most appropriate and fine-tuned model from the Sentence-BERT library to achieve optimal results. As for the implementation presented in this paper, we used the "all-mpnet-base-v2" model as the text encoder for Gesture generation and the "multi-qa-mpnet-base-dot-v1" model from the Sentence-BERT library for action generation. However, we opt to refer to these models collectively as the Sentence-BERT [47].

These embeddings are then employed to further map and convert the given screenplay into the parameters for the respective module for generating character behavior (see Figs. 3 and 4). Following are detailed descriptions of how embeddings are utilized in respective modules.

3.1.2 Dialogue paragraph - speech and gesture

Dialogue-type paragraphs correspond to the speech, that is, the lines to be spoken by the given virtual actors. A commercial Text-To-Speech API is used to generate the voice [43]. We also applied phoneme-based lip-sync animation, which is generated by a voice file, to provide natural facial movements of virtual characters [10].

The character's gaze is one important behavior that goes along with the speaking. The ASAP system animates the virtual actor such that it will turn its head and look at another listening character if one's name appears in the dialogue. Otherwise, the speaker will, by default, be made to look at an area where others are gathered.

While mere speech, lip movement, and gaze are not sufficiently persuasive, accompanying body gestures with spoken words is essential. For effective gesture expression, we mapped spoken text-to-3D gestures from diverse topics, such as current events, education, politics, and religion. We employed a blend of motion capture (Mocap) data, 2D poses, and video text data, supplemented by a BERT-driven language model [3] to facilitate this mapping.

We collected a dataset of about 250 hours of video content showcasing speakers' upper body movements (see Fig. 5). The 2D pose data often displayed inconsistencies, necessitating further refinement. We instituted a framework [2] to align the 2D pose data with its 3D counterpart to tackle this. For the essential 3D pose data, we gathered two hours of mocap recordings showcasing interactive dialogues between speakers. Using the mocap data, gesture segments were produced based on expert guidelines to pinpoint the beginning and end of gestures while maintaining a baseline variability.

To enhance the alignment of 2D poses from the videos with the 3D gesture segments, we devised a contrastive learning algorithm named GestureCLR. This model was trained with mocap data by converting 3-second segments into 3D and 2D versions. During the training, intentional distortions were infused into the 2D segments to mimic inconsistent data. After training, the model was adept at estimating the correlation between the 2D poses from the videos and the 3D gesture segments, forming a 3D pose-text pair dataset.

For refining the connection between the text and gestures, we employed text embeddings derived from the BERT-oriented language model. To eliminate redundancy in gesture pat-

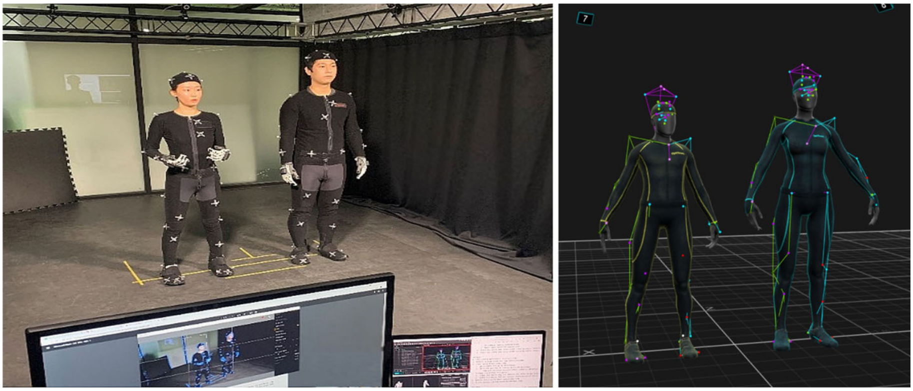


Fig. 5 A photo was taken while recording the upper body movement to generate a dataset for speech gestures

terns for specific text, we categorized the gestures, substituting them with cluster identifiers. Consequently, this produced pairs of cluster IDs and text. In real-time operation, relevant gestures for any given text input are chosen from the cluster tied to that text.

3.1.3 Parenthetical paragraph - facial expression

While the dialogue-type paragraph could be analyzed to produce the corresponding facial expression, here we take advantage of the parenthetical-type paragraph, which describes what kind of emotional expression is necessary as given directly by the scenario writer. To generate facial expressions according to the instructions in the parenthetical-type paragraphs, we developed a zero-shot style approach that utilizes emotion analysis. A dictionary of emotions is created, and each emotion is associated with a set of keywords for which the Sentence-BERT was applied to generate the embeddings. The embedding from the parenthetical paragraph is compared to all the keywords using the cosine similarity measure, and the collective score is computed for each of the seven emotion categories.

We utilized an emotion recognition score to convey the actor's feelings through facial expressions, utilizing a comprehensive set of over 46 blendshapes. These blendshapes enable precise control of various facial parts, with intensities ranging from 0 to 100. For the depiction of character facial animations, we referenced a facial expression dataset [35], basing our designs on Ekman's basic emotions (i.e., neutral, anger, disgust, fear, joy, sadness, and surprise) [12]. We determined specific values for each blendshape component to represent the seven emotions, each modifiable to three distinct intensity levels according to the emotion score (i.e., strong, medium, and weak). We then employed a commercial animation blending tool, SALSA [10], to smoothly transition between these emotional states, producing natural facial expressions. Detailed procedures are illustrated in Fig. 6.

Furthermore, we provide a customizable option for delineating the character's expressions in each paragraph, regardless of emotion analysis (see Fig. 7). This empowers users to convey the character's emotions with greater subtlety as needed (e.g., *"She is smiling outwardly, but inside, she is crying"*). While depicting such subtle emotions can be challenging for automated systems to visualize, it may be considered a limitation; however, these customization features will enhance the user-friendly aspects of the ASAP system.

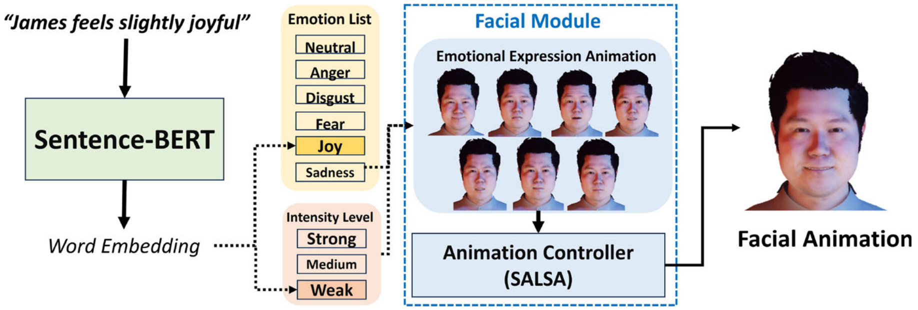


Fig. 6 Facial module generates facial animation with input from parenthetical descriptions

3.1.4 Action paragraph - physical action

The action-type paragraph is interpreted to generate the main physical 3D animations for the given virtual actor - which could be solitary actions (e.g., dancing, greeting, and walking) or simple interactions with objects/props (e.g., door, lamp, and sofa) in the scene. For now, the extracted action is described as a combination of four types of elements: subject, action, object, and position. The identification of the subject is simple, matched by any proper noun in the text.

Similarly to the previous analysis, a dictionary of plausible combinations of a set of actions, objects, and positions is pre-constructed (customized for different scenes and scripts) - e.g., "open–door–right," "turn-on–lamp–none," and "sit–sofa–left". A plausible combination refers to a possible action that can express the actor's various behaviors; e.g., "eat–door–none" is not a plausible combination. As a proof-of-concept, the current implementation can only support a small set of object, action, position descriptors, and likewise plausible combi-

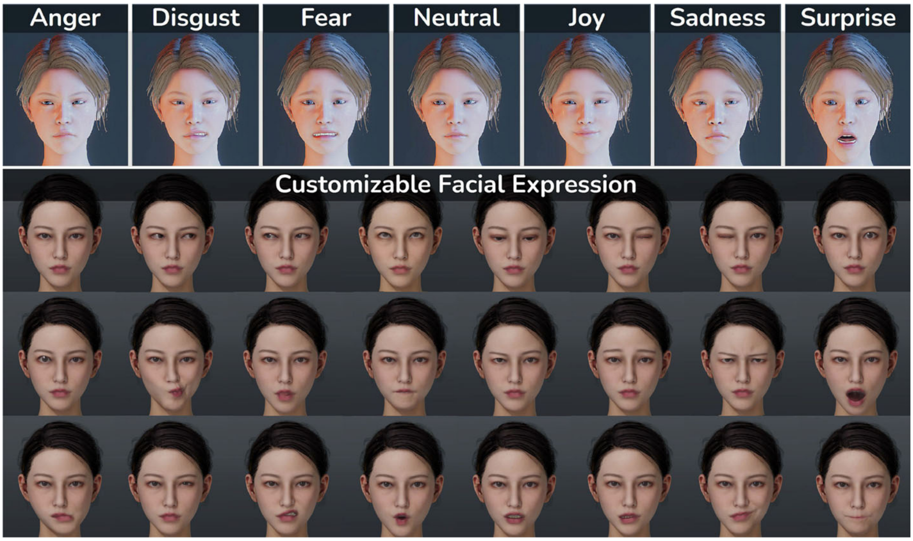


Fig. 7 The first line depicts the seven basic facial expressions: anger, disgust, fear, neutral, joy, sadness, and surprise. The following is a list of facial expressions that users can specify for each paragraph.

nations. Again, the Sentence-BERT [47] embedding is used to find a matching plausible combination to the given text, namely by cosine similarity search.

Finally, the extracted plausible combination is used as an input to generate a sequence of 3D animations of the actor to enact the given interpreted action, following a logic to determine the state of the actor, figure out what specific sub-actions to apply, and check for the physical conditions (e.g., the distance between the target object and the actor). For instance, for a plausible combination of “open-door-right”, following the flowchart in Fig. 8, the system will figure out which door to open, based on the distance to it, generate an animation sequence to first “walk" to it, and apply the final “open" animation once it collides with the target object (actually its box collider). The implementation of this animation sequence generation relies on parameterized animation clips modeled ahead of time given the dictionary of plausible combinations, using a variety of assets such as Mixamo’s animation clips [39], Final IK [48] for character rigging, and Unity’s NavMesh system [58] for finding the movement trajectory within given environment.

3.2 Object evaluation of behavior generation

The backbone of gesture generation is the accurate mapping of the 2D pose with 3D gesture units. For this mapping, we trained the *GestureCLR* model. To evaluate the model’s proficiency, we conducted tests using a subset of 500 gesture units (which were randomly sourced from 3D motion capture data). We generated 2D distorted versions of these units by applying Gaussian noise (with variances set at 0.001 for *Low Noise*, 0.01 for *Medium Noise*, and 0.1 for *High Noise*) to represent different noise levels. Additionally, we produced a 2D version’s *Temporal Alignment* by selecting sequential frame segments and repositioning them into new sequences.

We assessed the *GestureCLR* model’s results against the frame-wise mean cosine similarity measure introduced by Ali et al. [2]. Due to this measure’s inherent limitations of handling noise and temporal misalignment, the *GestureCLR* model consistently outperformed it, especially in managing noise and temporal adjustments. Specifically, both achieved 100% accuracy under low noise conditions. At medium noise levels, the *GestureCLR* model retained its 100% accuracy, whereas the measure dipped to 83%. In high noise situations, the *GestureCLR* model posted a 98% accuracy rate, significantly higher than the cosine similarity’s 16%. When faced with temporal rearrangements, the *GestureCLR* model registered an accuracy of 61%, far surpassing the cosine similarity’s 6%.

Matching the input with the correct action combinations is crucial for action generation. To evaluate the accuracy of our scheme, we created samples of input text and corresponding plausible actions. The input was divided into two categories: simple and complex. Simple sentences were short and direct, e.g., *“open window”*, *“close left drawer”*, while complex sentences were long and complex, e.g., *“He struggled not to fall, but eventually he had no choice but to fall.”*, *“The woman closed the door because she was concerned about the door being open.”*. The evaluation results[6] are presented in Table 2.

For the evaluation of emotion extraction, we employed the emotion recognition dataset [50], which contained 20,000 samples. We selected four primary emotion classes, namely ‘Joy’, ‘Sadness’, ‘Fear’, and ‘Anger’, and adjusted our emotion dictionary accord-

[6] The evaluated sentence dataset for actions is attached as a supplement.

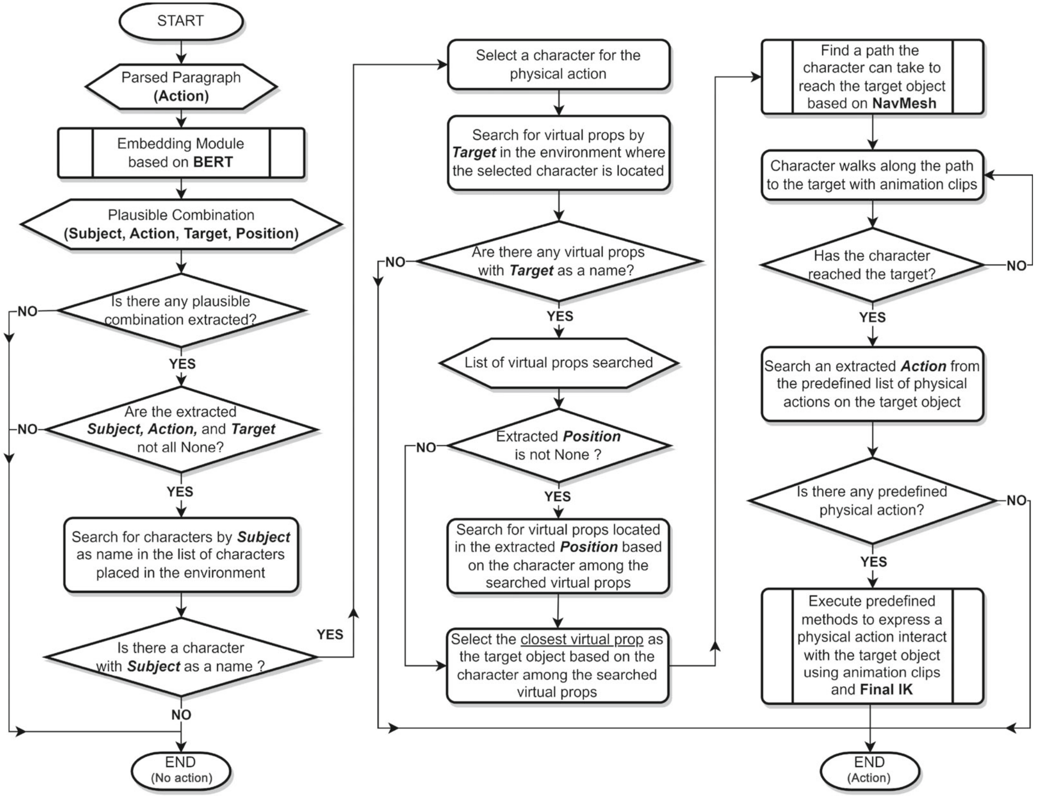

START
Parsed Paragraph
(Action)
Embedding Module
based on BERT
Plausible Combination
(Subject, Action, Target, Position)
Is there any plausible combination extracted?
NO
YES
Are the extracted Subject, Action, and Target not all None?
NO
YES
Search for characters by Subject as name in the list of characters placed in the environment
Is there a character with Subject as a name ?
YES
NO
END
(No action)
Select a character for the physical action
Search for virtual props by Target in the environment where the selected character is located
Are there any virtual props with Target as a name?
NO
YES
List of virtual props searched
Extracted Position is not None ?
NO
YES
Search for virtual props located in the extracted Position based on the character among the searched virtual props
Select the closest virtual prop as the target object based on the character among the searched virtual props
Find a path the character can take to reach the target object based on NavMesh
Character walks along the path to the target with animation clips
Has the character reached the target?
NO
YES
Search an extracted Action from the predefined list of physical actions on the target object
Is there any predefined physical action?
NO
YES
Execute predefined methods to express a physical action interact with the target object using animation clips and Final IK
END
(Action)

Fig. 8 The flowchart illustrates the process of generating procedural animation (i.e., physical action) by using the plausible action combinations extracted from action paragraphs in the screenplay. The left-side endpoint indicates that the action paragraph lacks sources to generate physical actions, whereas the right-side endpoint indicates the successful generation of physical actions for virtual characters

Table 2 Action generation accuracy for simple and complex inputs

Accuracy types	Simple sentences	Complex sentences
Top-1	93%	87%
Top-5	94%	92%

ingly. After filtering the dataset, the final size was 17,640 samples. Like other datasets, this one also suffered from imbalance, with the majority of samples belonging to the 'Joy' and 'Sadness' classes. The extraction algorithm achieved a Top-1 accuracy of 54% and a Top-2 accuracy of 77.3%. Moreover, the confusion matrix shown in Fig. 9 provides a clearer picture and demonstrates its effectiveness.

Our research has revealed several key considerations that need to be addressed. Firstly, natural language cannot be neatly categorized, especially when it comes to emotions, and this becomes even more complicated when dealing with complex narratives such as those found in films. While evaluations have been informative, they are limited by the datasets that we used. To fully understand our model's capabilities, we need to evaluate it using a diverse range of datasets.

From a technical perspective, the gesture generation and action generation modules have inherent limitations. The gesture generation module faces efficiency challenges when interfacing with an extensive rule base, and its performance improves when provided with high-quality animations from a diverse group of actors. The module that maps text to relevant actions relies on the performance of underlying large language models (LLMs), making it crucial to continuously update and assess new LLMs to maintain and enhance performance.

4 Previz creation pipeline and case result

The ASAP system is designed to enhance the pre-visualization process in filmmaking and content creation, offering three distinct outputs that cater to various stages of production planning and execution. The main outputs are as follows: (1) 2D Storyboard (in the form of a series of screen capture of the pre-visualized scenes); (2) 3D Pre-visualized animation video (including 360-degree video format); and (3) VR-based immersive pre-visualization.

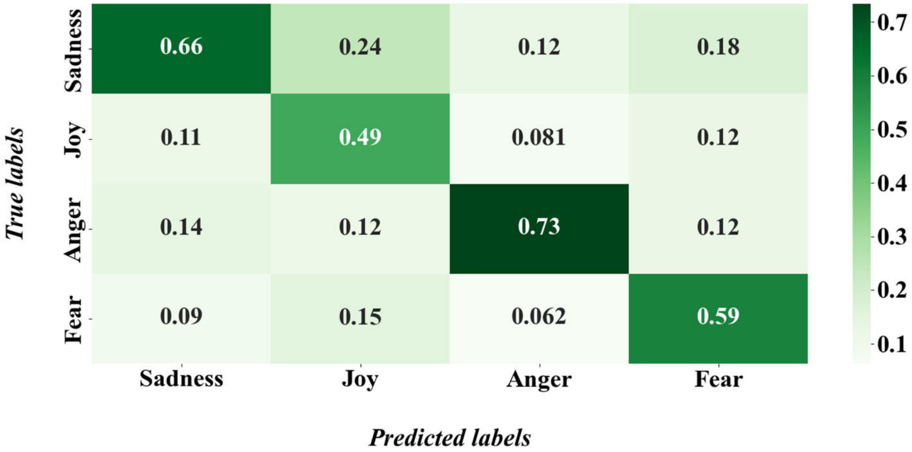


Fig. 9 The confusion matrix of the emotion extraction algorithm results

The following subsections outline the creation process for each output and demonstrate the efficacy of the ASAP system via a detailed case scenario.

4.1 Creation procedure and outputs

4.1.1 2D Storyboard

A storyboard is a collection of rough, quick sketches, typically drawn by hand or computer. These storyboards serve as a visual tool to represent ideas and concepts, outlining the film's plot and aiding in planning narrative flow, camera angles, character positions, backgrounds, and other visual aspects. Thus, the storyboard is treated as if it were a "bible" [57]. Despite its importance, storyboard creation is often a labor-intensive and time-consuming process [57].

The ASAP system enables users to take snapshots of the pre-visualized scene, transforming these snapshots into a storyboard. Users can also add brief descriptions to these snapshots. This feature simplifies the process of converting complex scenes into clear, visual story outlines, offering a lucid guide for subsequent production stages.

4.1.2 3D Pre-visualized animation video

Building on the capabilities for 2D storyboarding, the ASAP system offers the ability to record scenes as 3D pre-visualized animations, which can be rendered into 3D video formats. While the pre-visualized animation is playing, users can adjust perspectives in real-time by controlling multiple cameras set positioned at different angles, which is the common filmmaking practices.

Furthermore, the ASAP system supports the creation of 360-degree video formats, encompassing the user's designated location. These immersive videos serve as valuable tools for filmmakers, offering insights into spatial understanding before commencing the actual stage setup. By immersing themselves in these videos, filmmakers can assess the spatial dynamics, lighting conditions, and camera placement intricacies, enabling them to fine-tune their creative vision and optimize logistical planning.

4.1.3 VR-based immersive pre-visualization

The ASAP system into a VR-based platform marks a significant enhancement, offering filmmakers a fully immersive experience within pre-visualized scenes. This facilitates dynamic interactions with virtual characters and environments, enabling users to traverse scenes using teleportation or ghost-like movements. Such immersion into the pre-visualized world provides filmmakers with an unparalleled depth of spatial understanding, crucial for the development of complex cinematic compositions.

The integration of VR technology into the ASAP system, as depicted in Fig. 10, showcases the system's VR mode through various operational snapshots. This feature not only improves the user experience but also serves as an invaluable tool for filmmakers, aiming to deepen their comprehension of spatial dynamics and thereby enrich the creative quality of their projects.

In conclusion, the ASAP system's innovative approach to filmmaking pre-visualization across its three distinct outputs presents a comprehensive toolset for filmmakers. By leveraging 2D storyboards, 3D pre-visualized animation, and VR-based immersive experiences, the system enhances creative decision-making and streamlines production planning, ultimately contributing to the realization of more intricate and visually compelling cinematic projects.

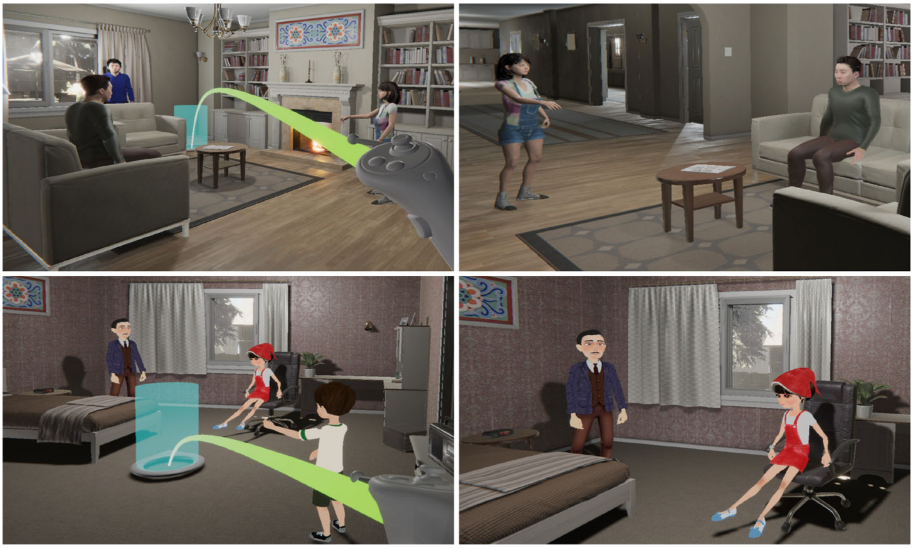
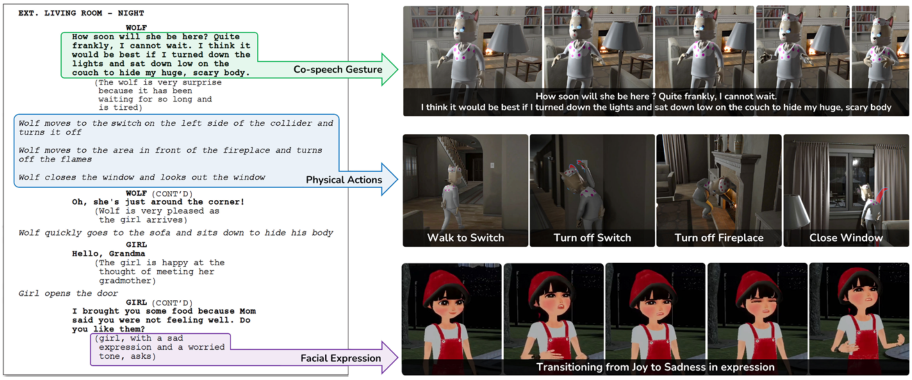


Fig. 10 The snapshot of the VR previz: While the pre-visualized animation is in progress, users have the freedom to observe and immerse themselves in the scene by walking, adjusting their viewpoint with head movements, and utilizing teleportation to different locations

4.2 Case scenario

We used the well-known fairy tale *"Little Red Riding Hood"* as an illustrative scenario to demonstrate the practical applications of the ASAP system. Figure 11 shows both the screenplay and the resultant pre-visualized scene with cartoon-style characters for the girl and the wolf. While the actual screenplay intricately weaves together different types of paragraphs to enrich the story, Fig. 11 serves as an example that organizes character animations sequentially: dialogue, action, and parenthetical cues for straightforward comprehension.



```
EXT. LIVING ROOM - NIGHT
                          WOLF
            How soon will she be here? Quite
            frankly, I cannot wait. I think it
            would be best if I turned down the
            lights and sat down low on the
            couch to hide my huge, scary body.
                 (The wolf is very surprise
                  because it has been
                  waiting for so long and
                  is tired)
Wolf moves to the switch on the left side of the collider and
turns it off

Wolf moves to the area in front of the fireplace and turns
off the flames

Wolf closes the window and looks out the window
                          WOLF (CONT'D)
            Oh, she's just around the corner!
                 (Wolf is very pleased as
                  the girl arrives)
Wolf quickly goes to the sofa and sits down to hide his body
                          GIRL
            Hello, Grandma
                 (The girl is happy at the
                  thought of meeting her
                  gradmother)
Girl opens the door
                          GIRL (CONT'D)
            I brought you some food because Mom
            said you were not feeling well. Do
            you like them?
                 (girl, with a sad
                  expression and a worried
                  tone, asks)
```

Co-speech Gesture
Physical Actions
Facial Expression
How soon will she be here ? Quite frankly, I cannot wait.
I think it would be best if I turned down the lights and sat down low on the couch to hide my huge, scary body
Walk to Switch
Turn off Switch
Turn off Fireplace
Close Window
Transitioning from Joy to Sadness in expression

Fig. 11 Illustrative scenario *"Little Red Riding Hood"*, the ASAP system produces a cartoon-like pre-visualization based on the screenplay. This pre-visualization encompasses key moments such as the wolf demonstrating co-speech gestures while delivering his monologue, the wolf taking a series of physical actions to interact with virtual props, and the girl displaying a range of changing facial expressions

To provide further details, the first animation demonstrates the co-speech gesture derived from the dialogue paragraph for the wolf's monologue (as indicated by the green box in Fig. 11). This animation, representing speech, is further enhanced by synchronized gaze shifts and lip movements. Moreover, the subsequent parenthetical cues are employed to generate emotional facial expressions for the wolf, enhancing the level of expressiveness and character depth.

Following a sequence of action paragraphs (as marked by the blue box in Fig. 11), the animations illustrate the physical actions of the wolf. The wolf initially approaches the light switch to turn it off. As he approaches the switch, he extends his arm to interact with the switch mockup. This interaction involves a visual effect, particularly the act of turning off the light, to signify the occurrence of the event.

Similarly, when the wolf walks to turn off the fireplace, he raises his arm and starts an animation that includes a visual on/off effect to show that the fireplace is off. As another example, *"Wolf closes the window"* (in the last sentence of the blue box, see Fig. 11). As he arrives at the window, he raises his arm, which starts a series of animations. During this sequence, a window-sliding animation is used to show that the window mockup is closed visually.

Lastly, in reference to the parenthetical paragraph, the girl displays a spectrum of emotions, particularly through her facial expressions, aiming to transition from joy to sadness (as highlighted by the purple box in Fig. 11), which aligns harmoniously with the speaking behavior established in the preceding paragraphs.

These examples vividly demonstrate the inherent potential of the ASAP system for previsualizing captivating and expressive animated content. We believe that the ASAP system aids in film pre-production, particularly previz, by generating the virtual character's actings and allowing filmmakers to observe it in an immersive environment to gain a better understanding before the actual filming.

5 Discussion

5.1 The effectiveness of ASAP

We revisit the effectiveness of the ASAP system, as demonstrated by our objective evaluations (as described in Section 3.2). First of all, the ASAP system has shown effectiveness in generating gestures, faces, and actions from natural language inputs. The gesture generation module is capable of producing a wide range of gestures. We demonstrated the system using approximately 3000 gestures, evenly split between female and male body types. Our system is also able to incorporate new gestures and topics from videos [2]. Its effectiveness in generating gestures is due to its training method, which involves using video data and clean motion capture animations to create pairs of text and gestures (as described in Section 3.1.2). The module is based on the *GestureCLR* model, which is capable of accurately mapping 2D poses to 3D gesture units, even in the presence of noise and temporal distortions, highlighting the system's robustness as mentioned in section 3.2. Unlike other models, this model does not require fine-tuning or retraining when dealing with new data, as it is capable of working "out of the box" since it has already learned to map 2D poses to 3D gestures. For the action module, the ASAP system is divided into two parts: extraction and generation. In the extraction part, the ASAP system is proficient in translating input text to relevant/plausible actions with high accuracy for simple and complex sentences (see Table 2). This simplifies the process of adding more actions to the ASAP system.

On the other hand, the generation part of the system requires skilled input, which may be seen as a limitation. To use virtual props, users define the object by name and list possible interactions (e.g., a door can be opened, closed, or leaned against). Additionally, users must indicate where a virtual actor should place their hands for natural interactions. The system includes various pre-imported behavioral animations (from the motion-captured data, Mixamo, and Unity asset store). Thus, users only need to specify interactions and virtual prop's interaction point, as the ASAP system's flexible animation features allow adding new actions without programming skills. Basic animation frames utilize an inverse kinematic function tailored to the specified interaction points for each virtual prop, enabling adjustable animations that offer a range of character behaviors despite a limited number of pre-defined animations. Moreover, movement animation also utilizes AI-based navigation and trajectory-defining algorithms provided by the Unity engine, ensuring that animations can be executed optimally along the best path, regardless of where the virtual props are placed. Therefore, although ASAP's initial step requires defining animation and interaction points for each prop (such tasks are necessary anyway when creating a new environment for a new screenplay), the system allows for diverse actions with minimal pre-defined animations.

Secondly, we evaluated the performance of the ASAP system's emotion extraction module utilizing a widely recognized emotion benchmark dataset [50]. The evaluation revealed a robust performance across four main emotions (i.e., Sadness, Joy, Anger, and Fear), with particularly high accuracy observed in recognizing Anger (73%, see Fig. 9). Our emotion extraction module combines LLM and bag of words techniques. Results are within the acceptable range as long as the intent of the text matches the emotion. However, this method is likely to fail when detecting sarcasm.

Overall, the ASAP system's proficiency highlights its ability to effectively represent emotionally charged behaviors, which is crucial for pre-visualized filmmaking. The combination of facial emotions, body gestures, and interactive capability is the key strength that enables the transformation of emotional scripts into visualization. This strength is particularly significant when creating scripts with complex emotional narratives, where it is important to distinctly convey the nuances of each emotion. Our findings via demonstrations, suggest that the ASAP system can offer substantial benefits, especially in pre-visualized scripts with rich emotional content.

5.2 Demonstrations with user feedback and limitations

The practicality and efficacy of the ASAP system were successfully demonstrated to a substantial audience through various technical demos. These demonstrations took place at prominent events through the peer reviews, including (1) the Real-time Live session of the SIGGRAPH ASIA 2022 conference [22], (2) the Creative Award session of the Korea HCI 2023 conference [30], (3) the poster session of the IEEE ISMAR 2021 conference [28], and (4) the Open-Lab Event at the Korea Institute of Science and Technology. During the demonstrations, we gathered valuable feedback from users. This feedback was organized into three broad categories: (1) common/consistent questions; (2) possible use case; and (3) identified limitations.

5.2.1 User perception

Most user reviews were favorable regarding ASAP's capability to understand screenplays and generate pre-visualized animations through AI-based systems. Many anticipated cost savings and reduced labor and wanted to see movies produced using ASAP. There were various inquiries, ranging from simple questions about how animation rigging was done or which character models were used to questions about commercialization plans. Among them, there were consistent inquiries about how the ASAP system would work if the animation or virtual props extracted from the screenplay were not prepared beforehand or pre-built into the virtual environment. It is quite possible that the AI language model may not properly comprehend the text input that is not well-formed (e.g., not in the form of common ordinary English expressions), in which case, the pertaining text/script would be skipped. Normally, as described in Section 3.1.1, the AI-based text processing model is quite powerful in capturing the deep contextual meanings and nuances to provide the closest possible corresponding expressive actions from the trained data. Therefore, the system's generality, scalability, and full applicability are only bound by the size, variety, and richness of the available virtual objects/props and possible action animation template libraries.

5.2.2 Possible use cases

The main application domain of the ASAP system is the pre-visualization of the movie film's screenplay (prior to the actual shoots) to plan out, confirm, and revise how the actors should play out the scripts. The system would be even more effective and beneficial for scene rehearsals at special and access-limited locations, or for practice sessions with virtual actors when the real actors cannot be present on the actual scene all the time.

The surveyed audiences' most recommended application field was education such as for vehicle driving, cardiopulmonary resuscitation (CPR)s, laboratory safety training, and surgical procedure explanations. A common point they mentioned is that preparing for these educational scenarios requires considering numerous possible contingent cases, leading to a shortage of time and resources. For example, explaining a surgical procedure involves not only the purpose of the surgery but also the patient's current condition and the interactive aspect, which increases the complexity. The ASAP system, which can handle variant training scenarios automatically by just providing the basic screenplay, allows for adjustments to the script to meet specific requirements easily. This capability makes it possible to address scenarios with numerous variables and has the potential for widespread use.

Other possible use cases include those for entertainment (such as in the metaverse and games), service planning, and counseling.

5.2.3 Limitations

Audiences also revealed the several drawbacks of the ASAP system. One audience commented that the use of text-to-speech (TTS) API was not the best and affected the sense of immersion in a negative way, as acting includes not just bodily actions and facial gestures but also the use of voice in subtle ways (e.g. tone, emotion, intonation), which the TTS cannot fully capture. Augmenting the system with emotion-based speech synthesis or voice cloning may be considered in the future, with the additional need to extract the necessary features from the script.

Another limitation is that the ASAP system currently processes the text/scripts/paragraphs sequentially one at a time, for animation generation and visualization (even though the overall context may be extracted from a larger extent of the script). There may be situations where the actions of multiple actors need to be played out simultaneously or in an overlapped fashion (e.g., in heated arguments). A related issue is the current implementation cannot reflect the varied length or subtlety of the animated actions according to the scene situation. For instance, sometimes behavioral actions must be hurried, slowed down, or exaggerated. An elderly character might naturally move and speak slowly.

Such an issue has been addressed in other previz systems through the implementation of a timeline feature [38]. Such a feature allows the manual manipulation of the start (i.e., parallel processing) and end (i.e., playback speed) of characters' dialogues/actions. While the ASAP system could consider incorporating such a timeline feature, our focus is more on automatically handling these situations, e.g., through the use of a more comprehensive data set and refined and complex AI analysis model.

5.3 Future directions

According to Muender et al. [41], a previz software should incorporate nine main components to be a usable tool for domain experts. Of the nine, the authors note that six components are relevant to VR systems that allow for direct manipulation and gestures in three-dimensional space: sketching (modeling), assets/layout, visual effects (VFX), camera control, lighting, and posing/animation [41]. Most features are encompassed within the ASAP system; however, a few are missing components.

With regards to sketching and assets/layout, only pre-built virtual environments are currently available on the ASAP system. Although we have prepared various backgrounds and virtual props, we acknowledge that the system may not cover every story. Moreover, offering a customizable feature, such as allowing users to rearrange the background and props wherever they want, is an important component of usability. Therefore, the future direction for ASAP will be centered on enhancing user-friendly features, particularly the creation and placement of virtual props based on text [1], which is identified as a pivotal aspect in advancing the ASAP system.

Furthermore, the ASAP system did not support a function to create animated VFX in the pre-visualized scene. Many screenwriters include instructions on VFX for the atmosphere of a story, such as *"Bonfire was smoldering in the breeze."* The ASAP system only supports on/off interactions with virtual props, such as lighting a fire in the fireplace, flowing water from a faucet, or turning on the lamp (see Fig. 1), but it does not provide more detailed VFX animation. Integration with performing graphical simulations from text input will be an option to improve the ASAP system.

The ASAP system offers a comprehensive suite of camera tools, including a multiview camera, recorded 360-degree video, and an immersive VR system. These components allow users to observe pre-visualized scenes from a multitude of angles, positions, and viewpoints. However, the ASAP system does not yet support the camera view to be automatically focused on the scene or other advanced camera related functions such as the low/high angle, close-up, and knee shot views. These camera techniques can be supported in a simple way by adding the ability to interpret the screenplay, specifically the *'Shot'* paragraph type. A viable alternative for adjusting camera focus is to analyze the entire narrative of the screenplay and identify the regions of interest (ROI). As examples of ROI, locations where multiple characters congregate or interactions with props take place are considered. Manually setting these regions improves pre-visualization performance but requires additional user effort. Thus, another potential avenue is to estimate the ROI from the screenplay and have camera movements follow it.

6 Conclusion

This paper introduced the ASAP system, which can automatically generate a 3D pre-visualized scene based on the comprehension of the screenplay. As the ASAP system can support to generating behavior of virtual characters from text, it can be used as a streamlined and effective pre-visualization tool. As a proof of concept system, this system still has many limitations. A more flexible text-to-animation technique will be needed to handle more interactive actions, emotional expression, and dynamic constraints. The only viable solution is to prepare a substantial amount of 3D animation in advance, but neural rendering could be considered. Incorporating more filming techniques (e.g., camera work, visual effects) could be another avenue to improve the quality of previz. We believe that people without expertise in animation, programming, or even virtual characters can swiftly and visually convey their creative ideas, in line with the concept of *"As Soon As Possible (ASAP)*."

Supplementary information

In order to demonstrate the fidelity of the ASAP system, we have attached three types of pre-visualized videos and the dataset used for quantitative evaluation. The narratives of the videos are respectively: (1) Pinocchio, (2) Little Red Riding Hood, and (3) Family Drama.

Acknowledgements This work was supported by the Korea Institute of Science and Technology (KIST) Institutional Program (Project No. 2E32303) and the Basic Research Laboratory Program funded by the National Research Foundation of Korea (NRF) (Grant No. 2022R1A4A1018869).

Data Availability Data for action/gesture sentences will be made available on reasonable request. The system code is not available.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Achlioptas P, Huang I, Sung M, et al (2023) Shapetalk: a language dataset and framework for 3d shape edits and deformations. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 12685–12694
2. Ali G, Hwang JI (2022) Improving co-speech gesture rule-map generation via wild pose matching with gesture units. In: SIGGRAPH Asia 2022 posters. ACM, SA '22, https://doi.org/10.1145/3550082.3564185,
3. Ali G, Lee M, Hwang JI (2020) Automatic text-to-gesture rule generation for embodied conversational agents. Comput Animation Virtual Worlds 31(4-5)
4. Bhattacharya U, Rewkowski N, Banerjee A, et al (2021) Text2gestures: a transformer-based network for generating emotive body gestures for virtual agents. In: 2021 IEEE virtual reality and 3D user interfaces (VR), IEEE, pp 1–10
5. Bouali N, Cavalli-Sforza V (2023) A review of text-to-animation systems. IEEE Access
6. Cassell J, Pelachaud C, Badler N, et al (1994) Animated conversation: Rule-based generation of facial expression, gesture, and spoken intonation for multiple conversational agents. Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1994 pp 413–420.https://doi.org/10.1145/192161.192272
7. Cassell J, Vilhjálmsson HH, Bickmore T (2001) Beat: the behavior expression animation toolkit. Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2001 pp 477–486.https://doi.org/10.1145/383259.383315

8. Chakravarthy A, Beales R, Jung Y, et al (2010) A notation based approach to film pre-vis. In: 2010 Conference on visual media production, IEEE, pp 58–63
9. Coyne B, Sproat R (2001) Wordseye: an automatic text-to-scene conversion system. In: Proceedings of the 28th annual conference on Computer graphics and interactive techniques, pp 487–496
10. Crazy Minnow Studio (2014) Salsa lipsync. https://crazyminnowstudio.com/unity-3d/lip-sync-salsa
11. Devlin J, Chang MW, Lee K, et al (2018) Bert: pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805
12. Ekman P et al (1999) Basic emotions. Handbook Cognition Emotion 98(45–60):16
13. Evin I, Hämäläinen P, Guckelsberger C (2022) Cine-ai: Generating video game cutscenes in the style of human directors. Proceedings of the ACM on Human-Computer Interaction 6(CHI PLAY):1–23
14. Final Draft Inc (1990) Final draft. https://www.finaldraft.com/, Accessed 11 Jun 2021 [Online]. https://www.finaldraft.com/
15. Galvane Q, Lin IS, Christie M, et al (2018) Immersive previz: Vr authoring for film previsualisation. In: ACM SIGGRAPH 2018 Studio. p 1–2
16. Galvane Q, Lin IS, Argelaguet F, et al (2019) Vr as a content creation tool for movie previsualisation. In: 2019 IEEE conference on Virtual Reality and 3D user interfaces (VR), IEEE, pp 303–311
17. Gauthier JM (2013) Building interactive worlds in 3D: virtual sets and pre-visualization for games, film & the web. Taylor & Francis
18. Ghosh A, Cheema N, Oguz C, et al (2021a) Synthesis of compositional animations from textual descriptions. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 1396–1406
19. Ghosh A, Cheema N, Oguz C, et al (2021b) Text-based motion synthesis with a hierarchical two-stream rnn. In: ACM SIGGRAPH 2021 posters. p 1–2
20. Hong F, Zhang M, Pan L, et al (2022) Avatarclip: zero-shot text-driven generation and animation of 3d avatars. arXiv:2205.08535
21. Hu L, Qi J, Zhang B, et al (2021) Text-driven 3d avatar animation with emotional and expressive behaviors. In: Proceedings of the 29th ACM international conference on multimedia, pp 2816–2818
22. Hwang JI, Ali G, Kim H, et al (2022) Asap: auto-generating storyboard and previz. In: Proceedings of the SIGGRAPH Asia 2022 real-time live! p 1–1
23. Ichikari R, Kawano K, Kimura A, et al (2006) Mixed reality pre-visualization and camera-work authoring in filmmaking. In: 2006 IEEE/ACM international symposium on mixed and augmented reality, IEEE, pp 239–240
24. Ichikari R, Tenmoku R, Shibata F et al (2008) Mixed reality pre-visualization for filmmaking: on-set camera-work authoring and action rehearsal. Int J Virtual Reality 7(4):25–32
25. Jung Y, Wagner S, Jung C, et al (2010) Storyboarding and pre-visualization with x3d. In: Proceedings of the 15th international conference on web 3D technology, pp 73–82
26. Kammerlander RK, Pereira A, Alexanderson S (2021) Using virtual reality to support acting in motion capture with differently scaled characters. In: 2021 IEEE Virtual Reality and 3D user interfaces (VR), IEEE, pp 402–410
27. Kapadia M, Frey S, Shoulson A, et al (2016) Canvas: computer-assisted narrative animation synthesis. In: Symposium on computer animation, pp 199–209
28. Kim H, Ali G, Hwang JI (2021a) Asap: Auto-generating storyboard and previz with virtual humans. In: 2021 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct), IEEE, pp 316–320
29. Kim H, Ali G, Kim S, et al (2021b) Auto-generating virtual human behavior by understanding user contexts. In: 2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW), pp 591–592. https://doi.org/10.1109/VRW52623.2021.00178
30. Kim H, Kim J, Han B et al (2023) Previz automation system based on movie script using digital humans. Proc HCI Korea 2023:1266–1267
31. Lamberti F, Gatteschi V, Sanna A et al (2019) A multimodal interface for virtual character animation based on live performance and natural language processing. Int J Human-Comput Inter 35(18):1655–1671
32. Lee J, Marsella S (2006) Nonverbal behavior generator for embodied conversational agents. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 4133 LNAI:243–25. https://doi.org/10.1007/11821830_20
33. Levine S, Krähenbühl P, Thrun S et al (2010). Gesture controllers. https://doi.org/10.1145/1778765.1778861
34. Lin AS, Wu L, Corona R et al (2018) Generating animated videos of human activities from natural language descriptions. Learning 1(2018):1
35. Lucey P, Cohn JF, Kanade T, et al (2010) The extended cohn-kanade dataset (ck+): a complete dataset for action unit and emotion-specified expression. In: 2010 ieee computer society conference on computer vision and pattern recognition-workshops, IEEE, pp 94–101

36. Ma M, Mc Kevitt P (2006) Virtual human animation in natural language visualisation. Artif Intell Rev 25:37–53
37. Marsella S, Xu Y, Lhommet M, et al (2013) Virtual character performance from speech. Proceedings - SCA 2013: 12th ACM SIGGRAPH / Eurographics Symposium on Computer Animation pp 25–36https://doi.org/10.1145/2485895.2485900
38. Marti M, Vieli J, Witoń W, et al (2018) Cardinal: computer assisted authoring of movie scripts. In: 23rd International conference on intelligent user interfaces, pp 509–519
39. Mixamo Inc. (2021) Mixamo. https://www.mixamo.com
40. Muender T, Fröhlich T, Malaka R (2018) Empowering creative people: Virtual reality for previsualization. In: Extended abstracts of the 2018 CHI conference on human factors in computing systems. Association for Computing Machinery, New York, NY, USA, CHI EA '18, p 1–6. https://doi.org/10.1145/3170427.3188612
41. Muender T, Volkmar G, Wenig D, et al (2019) Analysis of previsualization tasks for animation, film and theater. In: Extended abstracts of the 2019 CHI conference on human factors in computing systems, pp 1–6
42. Muennighoff N, Tazi N, Magne L, et al (2022) Mteb: massive text embedding benchmark. https://doi.org/10.48550/ARXIV.221007316,
43. NAVER Cloud (2019) Clova premium voice. https://www.ncloud.com/product/aiService/cpv
44. Neff M, Kipp M, Albrecht I et al (2008) Gesture modeling and animation based on a probabilistic re-creation of speaker style. ACM Trans Graph 27(1):1–2. https://doi.org/10.1145/1330511.1330516
45. Oshita M (2010) Generating animation from natural language texts and semantic analysis for motion search and scheduling. Vis Comput 26:339–352
46. Pennington J, Socher R, Manning CD (2014) GloVe: global vectors for word representation. In: EMNLP 2014 - 2014 Conference on empirical methods in natural language processing, proceedings of the conference. https://doi.org/10.3115/v1/d14-1162
47. Reimers N, Gurevych I (2019) Sentence-bert: sentence embeddings using siamese bert-networks. In: Proceedings of the 2019 conference on empirical methods in natural language processing. association for computational linguistics. https://arxiv.org/abs/1908.10084
48. Root Motion (2014) Final ik. http://root-motion.com/
49. Sanna A, Lamberti F, De Pace F et al (2017) Arsset: augmented reality support on set. International Conference on Augmented Reality. Springer, Virtual Reality and Computer Graphics, pp 356–376
50. Saravia E, Liu HCT, Huang YH, et al (2018) CARER: contextualized affect representations for emotion recognition. In: Proceedings of the 2018 conference on empirical methods in natural language processing. association for computational linguistics, Brussels, Belgium, pp 3687–3697. https://doi.org/10.18653/v1/D18-1404
51. Saund C, Bîrlădeanu A, Marsella S (2021) Ccfm : an architecture for realtime gesture generation by clustering gestures by communicative function and motion clustering by communicative function. Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021) 9. www.ifaamas.org
52. Shin M, Kim Bs, Park J (2005) Ar storyboard: an augmented reality based interactive storyboard authoring tool. In: Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'05), IEEE, pp 198–199
53. Shoulson A, Marshak N, Kapadia M, et al (2013) Adapt: the agent development and prototyping testbed. In: Proceedings of the ACM SIGGRAPH symposium on interactive 3D graphics and games, pp 9–18
54. Stamm A, Teall P, Benedicto GB (2016) Augmented virtuality in real time for pre-visualization in film. In: 2016 IEEE Symposium on 3D User Interfaces (3DUI), IEEE, pp 183–186
55. Stone M, DeCarlo D, Oh I et al (2004) Speaking with hands: creating animated conversational characters from recordings of human performance. ACM Trans Graph 23:506–51. https://doi.org/10.1145/1015706.1015753
56. Subramonyam H, Li W, Adar E, et al (2018) Taketoons: script-driven performance animation. In: Proceedings of the 31st annual ACM symposium on user interface software and technology, pp 663–674
57. Tschang FT, Goldstein A (2010) The outsourcing of "creative" work and the limits of capability: the case of the philippines' animation industry. IEEE Trans Eng Manage 57(1):132–143
58. Unity (2020) Unity navigation and pathfinding. https://docs.unity3d.com/2020.3/Documentation/Manual/Navigation.html
59. Venkatasawmy R (2012) The Digitization of Cinematic Visual Effects: Hollywood's Coming of Age. Lexington Books
60. WebGames3D (2014) Shotpro hq. https://www.shotprofessional.com/

61. Yoon Y, Ko WR, Jang M, et al (2019) Robots learn social skills: End-to-end learning of co-speech gesture generation for humanoid robots. In: 2019 International Conference on Robotics and Automation (ICRA), IEEE, pp 4303–4309

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Authors and Affiliations

Hanseob Kim[1,2] · Ghazanfar Ali[1] · Bin Han[1,3] · Hwang Youn Kim[1] · Jieun Kim[1] · Hyemin Shin[2] · Gerard Jounghyun Kim[2] · Jae-In Hwang[1]

✉ Jae-In Hwang
hji@kist.re.kr

Hanseob Kim
khseob0715@kist.re.kr

Ghazanfar Ali
aliust@ust.ac.kr

Bin Han
binhan@usc.edu

Hwang Youn Kim
daqjjang@ust.ac.kr

Jieun Kim
092296@kist.re.kr

Hyemin Shin
mini9974@korea.ac.kr

Gerard Jounghyun Kim
gjkim@korea.ac.kr

[1] Center for Artificial Intelligence, Korea Institute of Science and Technology, 5 Hwarang-ro, Seongbuk-gu, Seoul 02792, Republic of Korea

[2] Department of Computer Science and Engineering, Korea University, 145 Anam-ro, Seongbuk-gu, Seoul 02841, Republic of Korea

[3] Institute for Creative Technologies, University of Southern California, 12015 E Waterfront Dr., Playa Vista 90405, California, USA