Interactive Augmented Reality Storytelling Guided by Scene Semantics

CHANGYANG LI, George Mason University, USA
WANWAN LI, George Mason University and University of South Florida, USA
HAIKUN HUANG, George Mason University, USA
LAP-FAI YU, George Mason University, USA

We present a novel interactive augmented reality (AR) storytelling approach guided by indoor scene semantics. Our approach automatically populates virtual contents in real-world environments to deliver AR stories, which match both the story plots and scene semantics. During the storytelling process, a player can participate as a character in the story. Meanwhile, the behaviors of the virtual characters and the placement of the virtual items adapt to the player's actions. An input raw story is represented as a sequence of events, which contain high-level descriptions of the characters' states, and is converted into a graph representation with automatically supplemented low-level spatial details. Our hierarchical story sampling approach samples realistic character behaviors that fit the story contexts through optimizations; and an animator, which estimates and prioritizes the player's actions, animates the virtual characters to tell the story in AR. Through experiments and a user study, we validated the effectiveness of our approach for AR storytelling in different environments.

CCS Concepts: • **Computing methodologies** → **Graphics systems and interfaces**; **Augmented reality**.

Additional Key Words and Phrases: augmented reality, storytelling, character animation

ACM Reference Format:
Changyang Li, WanWan Li, Haikun Huang, and Lap-Fai Yu. 2022. Interactive Augmented Reality Storytelling Guided by Scene Semantics. *ACM Trans. Graph.* 41, 4, Article 91 (July 2022), 15 pages. https://doi.org/10.1145/3528223.3530061

1 INTRODUCTION

Recent developments of augmented reality (AR) technologies and visual computing techniques lead to the rising popularity of AR applications. The core intuition of AR is to enhance real world experiences with virtual contents. While AR applications nowadays commonly use acquired scene geometries for AR content placement, a growing trend is to leverage additional information such as scene semantics and tracked user's behaviors to deliver immersive and compelling AR experiences such as interactive AR storytelling.

In this paper, we propose a new approach to enable interactive AR storytelling guided by indoor scene semantics. Specifically, we focus on telling AR stories composed of daily activities, which happen in diverse indoor environments all around the world. Think about a simple activity such as a family chitchatting over dinner. This activity is performed by numerous families at numerous homes every evening. While the activity is contextually similar at each occurrence, its instantiation varies with the scene layouts and semantics of different homes, and with the behaviors of the participants. Suppose a story involves such an activity as one of its events. We can realize this story in AR at different homes if we have a computational approach that automatically adapts and animates the AR contents of a story according to the player's real-world surroundings and behaviors. Such an observation motivates our approach.

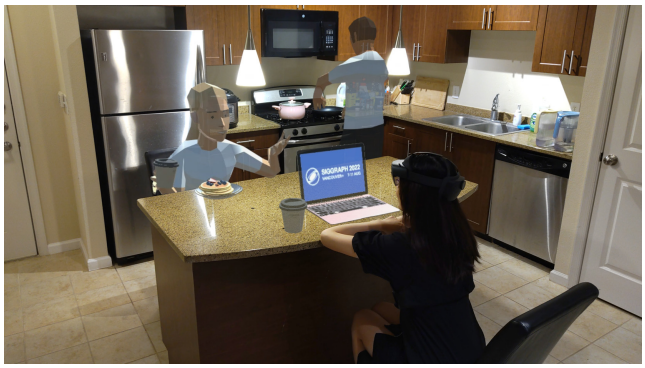


Fig. 1. Our approach automatically poses AR contents based on scene semantics. This example shows an event in a kitchen with a human player participating as a character in the story, interacting with virtual characters.

While there are progresses on positioning virtual agents [Lang et al. 2019; Tahara et al. 2020] and synthesizing a sequence of symbolic behaviors for a virtual pet [Liang et al. 2021b], interactive storytelling is a more challenging task as it involves a sequence of story events and many possibilities of interactions between the characters, including the AR player. Specifically, the challenges include: (1) How to position and pose the virtual contents. A specific difficulty with AR storytelling is that multiple virtual characters or objects might interact with each other, thus their positions and poses should be considered jointly; (2) How to choose an optimal solution to accommodate the story event contexts. For example, between multiple chairs in a scene, which chair should a virtual character choose in an event? (3) How to ensure the temporal coherence between the events of a story. Since a story happens chronologically, how to instantiate a story in a scene should not only consider spatial relations at a certain event, but also temporal relations between events at different time frames; (4) How to adapt the same story to different real-world environments.

In addition to addressing these challenges, our approach integrates user interactions in AR storytelling. The goal is to enable a

Authors' addresses: Changyang Li, George Mason University, USA, cli25@gmu.edu; WanWan Li, George Mason University and University of South Florida, USA, wli17@gmu.edu; Haikun Huang, George Mason University, USA, hhuang25@gmu.edu; Lap-Fai Yu, George Mason University, USA, craigyu@gmu.edu.

0730-0301/2022/7-ART91
https://doi.org/10.1145/3528223.3530061

player to take the role of a character and interact with other virtual contents according to the story plots. The main types of player interactions we consider include where to stay and what items to interact with in different event activities. Figure 1 shows an example. Such a feature requires our approach to dynamically adapt the story to the player's actions.

To this end, we design a hierarchical story sampling approach that takes abstract story descriptions as a sequence of events as the input, and then automatically supplements details to instantiate the story in a real scene. Our approach first samples spatial candidates for each event considering local spatial relations, and then applies a temporal story assembly to link selected spatial candidates together to produce a complete story. During the storytelling process in AR, our approach estimates the player's actions and samples new story assemblies to update the AR contents on the fly. Our approach can be applied in AR gaming and education, such as Fragments [Microsoft 2016], a holographic game in which life-sized characters can share real space and interact with players. The major contributions of our work include the following:

- We propose a novel problem of interactive AR storytelling involving virtual characters, items, and events to take place in different real scenes, considering scene layouts and semantics.
- We propose a hierarchical story sampling approach that adapts story events to the player's dynamic behaviors in a real environment. By sampling spatial candidates followed by story assemblies with temporal considerations, our approach supports interactive update of the story at runtime, enabling the player to participate as a character in AR storytelling.
- We validate the effectiveness of our approach by conducting AR experiments and a user study in real scenes.

Code for our paper is available at https://github.com/Changyangli/ar-storytelling.

2 RELATED WORK

2.1 Scene Understanding for Mixed Reality

Previous works have investigated applying scene understanding for mixed reality experiences. Some AR methods use geometry information to blend virtual items into physical environments. For example, FLARE [Gal et al. 2014] uses planar geometry for generating object layouts for AR applications. SnapToReality [Nuernberger et al. 2016] automatically aligns virtual items to physical constraints such as linear edges and planar surfaces extracted from the real world. Some AR approaches use scene semantics in addition to geometry, for example, exploiting semantic associations between virtual interfaces and physical environments [Chen et al. 2018; Cheng et al. 2021; He et al. 2022; Lindlbauer et al. 2019],authoring context-aware applications [Wang et al. 2020], constraining spatial relations for AR agent positioning [Lang et al. 2019], and synthesizing realistic virtual pet behaviors [Liang et al. 2021b]. Scene understanding also helps enhance virtual reality experiences, such as restructuring virtual scenes to fit with physical environments [Dong et al. 2021] and facilitating navigation [Li et al. 2021]. 3D scene datasets such as Matterport3D [Chang et al. 2017], which we used in our experiments, facilitate scene understanding research.

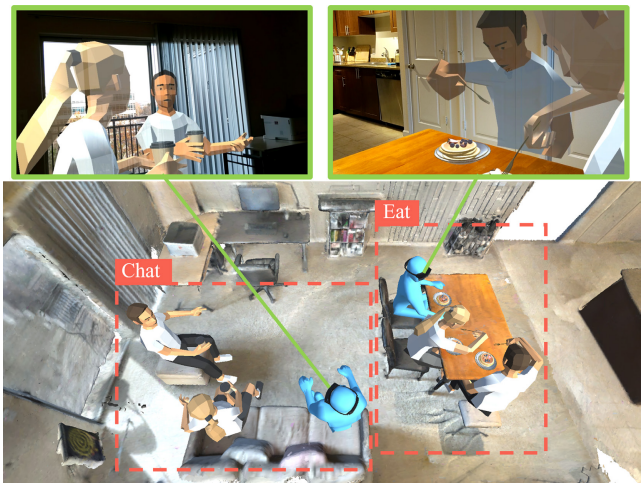


Fig. 2. Two AR story events in an apartment scene. AR views are shown in the green boxes. The blue avatar refers to the AR player.

Our work is inspired by Retargetable AR [Tahara et al. 2020], which uses 3D scene graphs to associate AR contents with physical environments to produce natural spatial arrangements. However, while Retargetable AR considers only spatial relations for a detailed input interaction context, our approach jointly contemplates spatial-temporal relations for an abstract sequence of story events descriptions and covers a large variety of activity possibilities. Moreover, while Retargetable AR places static AR contents in real environments considering scene contexts, our approach enables a human player to participate in the storytelling process, whose activities are adaptive to the player's dynamic actions at runtime.

2.2 Augmented Reality Storytelling

Storytelling is a form of communication where people share understandings and experiences [Cassell and Ryokai 2001]. As for the usage of AR in storytelling, early works has explored applications including education and entertainment [Billinghurst et al. 2001; Grasset et al. 2008; Zhou et al. 2004]. More recently, Rumiński and Walczak [2013] introduced a mobile AR authoring tool called MARAT. Glenn et al. [2020] proposed a system called StoryMakAR, which merges electro-mechanical devices with virtual characters to create stories. SceneAR [Chen et al. 2021] is another mobile application for creating sequential scene-based micro narratives in AR. Other works use mobile phones as direct controllers for animating AR characters [Anderegg et al. 2018; Ye et al. 2020]. Virtual stories can also be created using multi-agent simulators like VirtualHome [Puig et al. 2018]. While most of these prior arts are about efficient story authoring and animating created contents without considering the player's participation, our work focuses on adapting predefined stories to real scenes and player actions to enable interactive AR storytelling.

2.3 Human Activities and Environments

Modeling and generating virtual human behaviors based on scene semantics are critical for augmented reality, and it is a long-standing

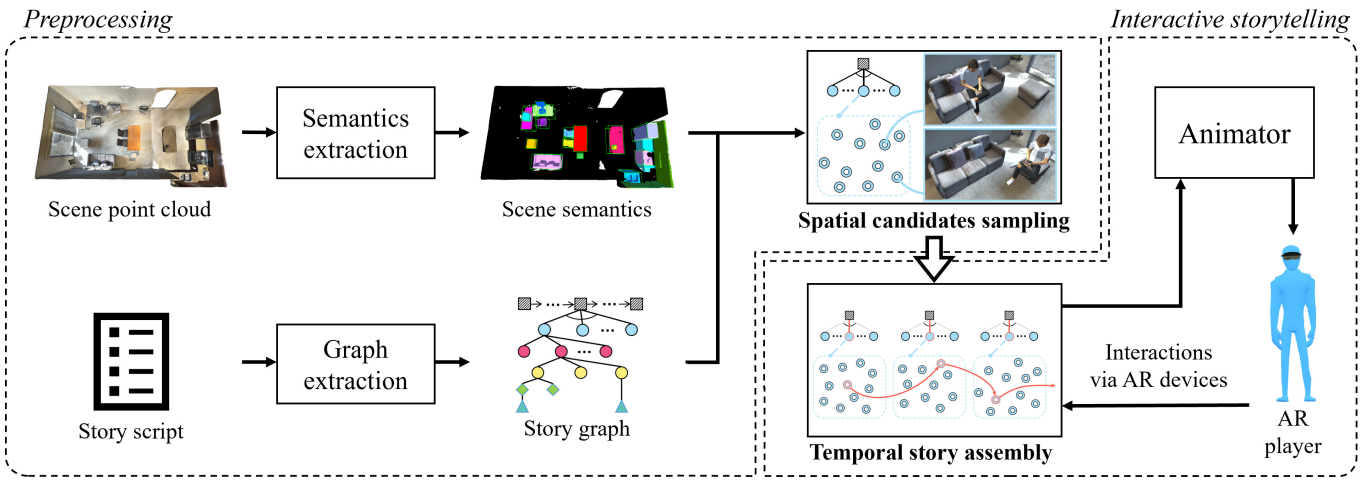


Fig. 3. An overview of our approach. In the preprocessing phase, scene semantics are extracted in a semi-automatic manner, and meanwhile a story graph is extracted from an input story script. A spatial candidates sampling process prepares a set of candidate AR content poses within each event locally. In the interactive storytelling process, the AR player keeps interacting with the AR contents at each event, and such interactions are passed as an input to the story assembly module for picking optimal spatial candidates for the following events adaptive to the player's actions.

problem in graphics and vision. For example, researchers have devised techniques to predict the likelihood of actions and pose humans realistically to interact in scenes [Gupta et al. 2011; Hassan et al. 2021b; Kim et al. 2014; Savva et al. 2014, 2016]. On the contrary, human activities can also guide 3D scene synthesis [Fisher et al. 2015; Ma et al. 2016]. Generating scene-aware human motions is another research problem, which requires understanding contexts for interactions between human and the environments [Agrawal and van de Panne 2016; Bai et al. 2012; Cao et al. 2020; Hassan et al. 2021a; Pirk et al. 2017; Starke et al. 2019; Wang et al. 2021a,b]. Moreover, robots may also participate in activities to facilitate interactions between humans, robots and environments [Choi et al. 2022; Qiu et al. 2020; Zhi et al. 2021].

Synthesizing continuous human motion in 3D space is a challenging task still. Since we focus on adapting AR stories to real scenes by considering spatial-temporal relations between events at discrete time frames, our approach only considers symbolic interaction priors in realizing story events without generating motion transitions. For example, for the question *"where should a human sit to watch TV?"*, the answers could be *"a sofa or a chair"*. We collected a dataset of such priors and used an animator comprising a set of pre-defined animations to complement motion transitions between character states in different events.

2.4 Representing Human Activities Using Graphs

Graphs are expressive and flexible for representing people and environments, and can be generally applied for modeling diverse tasks. For example, abstract graph representations are used for scene comparison and novel scene synthesis [Fisher et al. 2011; Li et al. 2019; Xu et al. 2013]. In computer vision, scene graphs [Chang et al. 2021; Johnson et al. 2015] are used for encoding object relationships in scenes as well as for high-level visual scene understanding and reasoning. Another family of methods use And-Or graphs [Zhu and Mumford 2007] with stochastic grammar models to parse hierarchical scene structures [Qi et al. 2017; Zhao and Zhu 2013] and synthesize 3D scenes [Jiang et al. 2018; Qi et al. 2018]. We use graphs to represent AR stories, which comprise spatial relations between the environments, characters and objects, together with their activities; as well as temporal relations between story events.

3 OVERVIEW

Our approach takes an abstract story description and a target scene as initial inputs, and dynamically assembles and animates a story according to the player's interaction with the scene and virtual characters. Figure 2 shows two example events in an apartment scene. We use a blue avatar wearing AR glasses to represent the AR player, and the first-person AR views are shown for the demonstrated events. Figure 3 shows an overview of our approach, which comprises the following two stages:

Preprocessing. Our approach first extracts scene semantics from the point cloud of the scanned target scene for storytelling (Section 4). The semantics is initialized by an instance segmentation model and oriented bounding boxes (OBBs) estimation heuristics, and then refined by a manual process to guarantee completeness and correctness. Our approach then generates a story graph, which contains a sequence of event graphs, as a detailed instantiation of the input story script (Section 5). Next, the extracted scene semantics and story graph are passed to a spatial candidates sampling module (Section 6.1) to parameterize the spatial graphs, which are sub-graphs of the event graphs and refer to diverse possible spatial relations, via a Markov chain Monte Carlo (MCMC) sampling.

Interactive Storytelling. Based on the generated event graphs, together with the sampled spatial candidates of the spatial graphs under them, a temporal story assembly module (Section 6.2) is triggered. The module selects a spatial graph branch and then a spatial

candidate for each event graph to propose an event instantiation. It then links all event instantiations together to assemble a story considering temporal relations. The module dynamically updates the story by reassembling it based on the players' actions in real time (Section 7.1). The story is animated (Section 7.2) and displayed to the player via an AR headset.

4 LINKING SCENE SEMANTICS TO STORY ACTIVITIES

A prerequisite for our approach, which automatically generates story activities compatible with various input scenes, is understanding the scene semantics including the semantic labels of the objects and their spatial relations. Given an input scene, we first apply a preprocessing step to obtain segmented instances, together with their semantic labels, oriented bounding boxes and room labels.

We devise a semi-automatic scene semantics processing approach. Instead of scanning and processing the 3D mesh only in the field of view at runtime, which is commonly done in AR applications for placing virtual characters or objects onto detected regions, we scan the whole scene and process the spatial scene geometry and semantics. We use SSTNet [Liang et al. 2021a], which is the state-of-the-art 3D scene instance segmentation method, to obtain the initial instances in the scanned 3D scene and predict their semantic labels. Alternatively, this can be achieved using interactive 3D labeling techniques while scanning the environments [Valentin et al. 2015]. Following that, we estimate the OBBs for each segmented instance using geometric heuristics similar to [Tahara et al. 2020].

Following the instance segmentation and OBBs estimation, an annotator performs a manual refinement process for two reasons. First, the number of semantic labels is limited in currently available training dataset for the instance segmentation task, thus the automatic instance segmentation method cannot deduce some instances' categories. Second, to ensure the correctness of story instantiation, the annotator manually inspects the instance segmentation and OBBs estimation results to fix any error. The annotator also assigns room labels to regions inside the input scene, and the labels are inherited by instances within the corresponding regions. The manual refinement is done using a metaphor involving mainly clicking and dragging operations in the Unity3D game engine.

In our framework, a character is associated with one furniture instance at any time event frame. To simplify the problem, we discretize the region of instances that can support a character. For example, a sofa that can support multiple characters is split into multiple slots with the same semantic label, while a chair that can support only one character does not need such a split. Similarly, we discretize the space in a region that a character may stand and access specific furniture or objects (e.g. stand by the stove and cook) to make the solution space consistent. Overall, we define two types of character slots, namely *stand slot* and *sit slot*. Once a character slot is taken by a character, it is marked as *occupied*. The character slot splitting for both types is completed by using geometric heuristics that splits the space by a person's approximate width. Figure 4 shows an example.

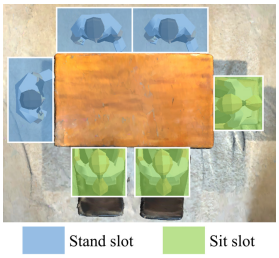


Fig. 4. Discretizing a supporting space into character slots.

5 AR STORY REPRESENTATION

In our approach, an AR story is initially described in a high-level script. The low-level details regarding possible realizations of the story events are automatically completed by our approach in the spatial candidate sampling process. The overall story comprising a sequence of story events together with their possible intantiations is represented as a story graph.

5.1 Raw Story Scripts

We first discuss the representation of the input raw story scripts. We define the atomic element $b = (character, activity, room)$ as a tuple that describes the state of a character in an event. In our work, a raw story only describes the high-level contexts between a fixed number of characters such that *activity* is a symbolic label (e.g., watch TV, read books). Details of the activity will be complemented automatically on the story graph later. For example, for the activity "watch TV", details may contain "sit on a sofa or a chair", "which slot of the sofa or which of the chairs", and "how is the character posed regarding body and head orientation". The parameter *room* refers to the room (e.g., a living room) where the activity happens.

A raw story comprises T event frames. So a story is represented as a set of all event frames $\mathcal{E} = \{e_k\}_{k=1,2,...,T}$. Each event frame $e_k = \{b_c\}$ is itself a set containing the state b_c of each character c at that frame. For example, an event frame may encode "a character watching TV in the living room". Our approach will generate spatial candidates about how a character may watch TV (e.g., sitting on a specific chair with a certain pose) to realize the event.

5.2 Indoor Activity Priors

To cover a variety of possible cases for abstract character activity descriptions, we define a probabilistic model for AR story adaptation. For example, characters' preferences of choosing a sofa or chair to sit for watching TV might be inferred from some daily activity priors. We collected a set of 200 pieces of data for defining our probabilistic model. Instead of retrieving activity priors from observations like in [Savva et al. 2014, 2016], we directly asked about symbolic preferences in situations referring to specific combinations of activities and rooms. For example, we asked if people prefer to sit on a sofa or a chair to "watch TV in a living room". We can then infer the priors for this situation from the users' answers directly. Refer to the supplementary material for more details.

Each character in the story is associated with symbolic attributes $(pose, verb, interactee)$ using the collected priors. *pose* indicates the character's pose in this event (e.g., standing, sitting). *verb* indicates the character's interaction with a real furniture or a virtual item (e.g., watching, reading, using). *interactee* refers to the furniture object or virtual item that the character is interacting with.

5.3 Representing Stories using Graphs

Converting an event from a raw story script into animations introduces uncertainty, which comes from the abstract descriptions of

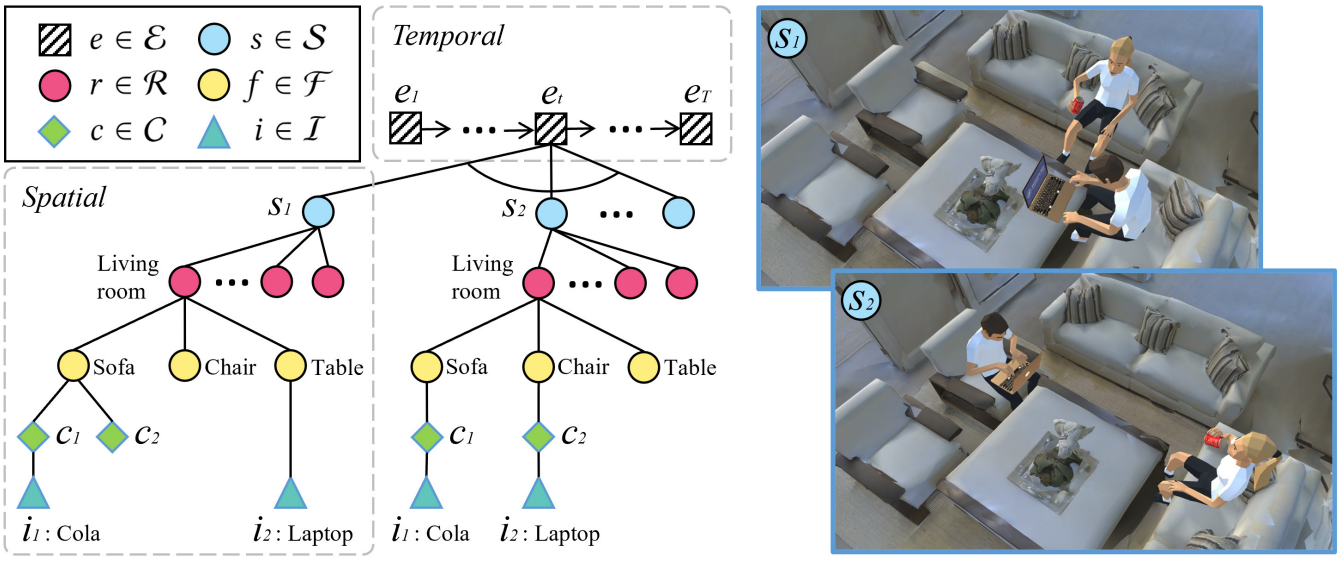


Fig. 5. A story graph representation. A story graph consists of a sequence of event graphs. Each event graph $\mathcal{G}_e$ is rooted at its corresponding event node e. Each event graph comprises multiple spatial graphs. Each spatial graph SG, rooted at a node s, contains the high-level spatial relations between the virtual characters C, virtual items $\mathcal{I}$, and the furniture objects $\mathcal{F}$ in a room $r \in \mathcal{R}$. Instantiating a spatial graph with position and orientation attributes of the characters and items generates a spatial candidate. The two scenes on the right show the virtual characters and objects set using two spatial candidates of the spatial graphs rooted at s_1 and s_2, respectively. Instantiating an event refers to selecting a spatial graph and generating a spatial candidate of this spatial graph.

character activities in each state b. For example, there are multiple ways for a character to sit on a chair and watch TV in a living room. We use the collected indoor activity priors to introduce such varieties by creating probabilistic branches for an event. Especially, since an event e comprises a description of multiple characters' states, various possible cases could fit the context.

Event Graph and Spatial Graph. Figure 5 illustrates our story graph representation, based on which our approach samples event instantiations for driving the animator to show the virtual characters and items in augmented reality to tell the story. At the highest level is the story graph which encodes the sequence of event frames $\{e_k\}$ of the story. Each event frame is associated with an event graph $\mathcal{G}_e$. Each event graph comprises multiple alternative spatial graphs, where each spatial graph SG encodes one possible spatial relations of the virtual characters and items with the layout of a room for realizing event e. Figure 6 illustrates the relations between an event graph and a spatial graph.

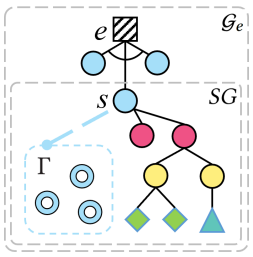


Fig. 6. A spatial graph SG is a sub-graph of an event graph $\mathcal{G}_e$, and is associated with a spatial candidate set Γ. An event instantiation $\hat{e}$ of event e is proposed by selecting a spatial graph branch and selecting a spatial candidate $\gamma \in \Gamma$.

Formally, let an event graph be $\mathcal{G}_e =< e, \mathcal{S}, \mathcal{R}, \mathcal{F}, C, \mathcal{I}, \mathcal{P} >$. Here, event e describes the high-level states of all participating characters. For notation convenience, we also let e refer to the root node of the event graph $\mathcal{G}_e$. $\mathcal{S} = \{s\}$ is the set of root nodes of all the spatial graphs under this event graph. $\mathcal{R}$ is the set of room nodes. $\mathcal{F}$ is the set of furniture nodes. The rooms and furniture objects are extracted from the 3D scan of the indoor scene where the story is played, and these two sets $\mathcal{R}$ and $\mathcal{F}$ jointly depict the static indoor scene structure. $C = \{c\}$ is the set of virtual characters in the event. $\mathcal{I} = \{i\}$ is the set of virtual items needed for character interactions in this event. Again, for notation convenience, let c and i also denote the corresponding virtual character node and virtual item node in the graph. $\mathcal{P}$ denotes a probability model used for selecting which spatial graph SG to use for realizing this event e. Note that room structure nodes (in $\mathcal{R}$ and $\mathcal{F}$) are static in one indoor scene and they are shared between all spatial graphs.

Spatial Candidate. While a spatial graph SG encodes the high-level spatial relations of the virtual characters and items with the layout of a room at an event frame, the low-level details regarding the positions and orientations of the characters and items are not described. For example, given the context "a virtual character sits on the sofa", which part of the sofa the character sits on and what direction it is facing are still unclear. Therefore, each character node $c \in C$ is attributed by $(\rho_c, \theta_{cb}, \theta_{ch})$, where ρ_c is the position, θ_{cb} is the body orientation, and θ_{ch} is the head orientation of the character. As we discretize the space that a character can occupy (Section 4), ρ_c refers to the position of a character slot. Similarly, each virtual item node $i \in \mathcal{I}$ is attributed by (ρ_i, θ_i), where ρ_i and θ_i refer to the item's position and root orientation. In case a character is carrying a virtual item, the item node i is attached to the character node c, and i inherits the position and orientation from c. Given a spatial graph SG, we define an instantiation of the attributes of all the virtual characters and items as a *spatial candidate*

$\gamma = \{\{(\rho_c, \theta_{cb}, \theta_{ch})\}, \{(\rho_i, \theta_i)\}\}$. In other words, a spatial candidate contains the position and orientation information by which the animator puts virtual characters and items in a scene.

Event Instantiation. Given an event e, there are multiple high-level spatial arrangements between the virtual characters, items, and a room layout for realizing the event. For example, to sit in a living room and watch TV, a character may choose to sit on a chair or a sofa. Each spatial arrangement is encoded by a spatial graph SG. So an event graph $\mathcal{G}_e$ typically comprises multiple spatial graphs.

Inspired by [Qi et al. 2017], we define the OR rule on event node e as a switch $e \rightarrow s_1|s_2|\ldots|s_n$, where each s_k refers to the root node of a spatial graph and n refers to the total number of spatial graphs. The probabilities $\mathbb{P}_1|\mathbb{P}_2|\ldots|\mathbb{P}_n$ refer to the probabilities of choosing the spatial graphs, which are defined in the probability model $\mathcal{P}$ of event graph $\mathcal{G}_e$. The probability model is computed using the indoor activity priors described in Section 5.2.

Given an event, to *instantiate* the event means selecting the high-level spatial arrangements between the virtual characters, items, and a room layout, followed by determining the positions and orientations of the virtual characters and items. More formally, we define an *event instantiation* $\hat{e}$ of event e as selecting a spatial graph SG under the event graph $\mathcal{G}_e$ and then generating a spatial candidate γ for this spatial graph SG.

Showing a Dynamic Story. Given a story represented by the set of all event frames $\mathcal{E} = \{e_k\}_{k=1,2,\ldots,T}$, if we generate a corresponding set of event instantiations $\{\hat{e}_k\}_{k=1,2,\ldots,T}$, the animator can then refer to each event instantiation $\hat{e}_k$ to put virtual characters and items in the scene to deliver each event e_k chronologically to tell the story.

As our approach supports the AR player's participation in the story, our approach dynamically updates the set of event instantiations according to the player's interactive behaviors. For example, if the player chooses to sit on a chair, the virtual characters will choose to sit on other chairs. Suppose the current event frame is e_t. We define the set of event instantiations $E_t = \{\hat{e}_k\}_{k=t,\ldots,T}$, which contains the instantiations of the current event frame e_t up till the last event frame e_T of the story. Suppose the player performs an action a_t, our approach resamples E_t through an optimization to generate the "best course" of the current and upcoming events of the story considering both spatial and temporal constraints. The animator then transitions from the previous event frame e_{t-1} to the current event frame e_t by referring to event instantiations $\hat{e}_{t-1}$ to $\hat{e}_t$. We discuss the details of optimization-based story sampling in Section 6 and dynamic story assembly in Section 7.

6 HIERARCHICAL STORY SAMPLING

We devise a story sampling framework for interactive AR storytelling. Since a story is described in the tangled spatial and temporal domains, the overall search space can be large and sparse. It can be difficult to sample a story solution efficiently by directly modeling the whole story as a single configuration, especially when there are multiple events that contain long temporal constraints, and when there are multiple virtual characters participating in a single event with complicated spatial constraints.

Fig. 7. Two cases considering the individual activity cost. For vision only activities, our approach mainly considers the forward directions (red vectors) of the character and item, as well as their relative direction (green vector). For activities involving physical contact, our approach further considers the accessibility to the interactee item.

To cope with these challenges, we divide the story sampling process into two steps: (1) spatial candidates sampling, which considers spatial constraints within each spatial graph SG for generating spatial candidates; and (2) story assembly, which uses the generated spatial candidates to assemble a story solution considering temporal constraints and dynamic player actions.

6.1 Spatial Candidates Sampling

To facilitate story assembly during the AR storytelling runtime, our approach first samples spatial candidates for each spatial graph SG. Formally, let $\Gamma_{t,u} = \{\gamma_{t,u,v}\}$ be the set of spatial candidates for the u-th spatial graph $SG_{t,u}$ under the event graph $\mathcal{G}_{e_t}$ at frame t.

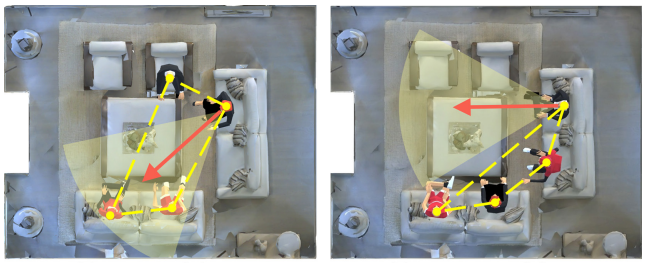
To simplify the problem, we do not consider cross-room activities, which are not very common in real-life scenarios. Therefore we assume that activities happening in different rooms are independent of each other. In the spatial candidates sampling process, solutions in rooms where virtual characters and virtual items stay are separately sampled, and random combinations of them are saved as the spatial candidates $\Gamma_{t,u}$. The following discussion uses an example scenario in a single room which includes multiple virtual characters and items. We use an MCMC sampler to generate spatial candidates.

6.1.1 Cost Function. We define a spatial cost function to evaluate whether virtual characters and items are reasonably placed and posed in an event. To facilitate discussion, we define some variables: (1) For a character c parameterized with $(\rho_c, \theta_{cb}, \theta_{ch})$, we denote the body forward direction as the unit vector $\hat{\mathbf{F}}_{cb}$ and the head forward direction as $\hat{\mathbf{F}}_{ch}$. (2) For an item i parameterized with (ρ_i, θ_i), we denote the forward direction as $\hat{\mathbf{F}}_i$. Overall, the spatial cost function for a spatial candidate γ is a weighted sum of cost terms:

$$C_{\text{S}}(\gamma) = w_{\text{IA}}C_{\text{IA}}(\gamma) + w_{\text{GA}}C_{\text{GA}}(\gamma) + w_{\text{CP}}C_{\text{CP}}(\gamma) + w_{\text{IP}}C_{\text{IP}}(\gamma) \quad (1)$$

We describe the details of each cost term in the following.

Individual Activity. For some activities, characters participate individually and no collaboration is involved, e.g., watching TV and using a computer as shown in Figure 7. We suppose that, ideally, when a virtual character c interacts with an item i, the character's head should be facing the item. We define the individual activity



With group activity cost | Without group activity cost

Fig. 8. Example group activity results in a chat event. In each case, the yellow points and dashed lines form the group convex hull. The yellow region and red vector indicate the field of view and head forward direction of the upper-right character.

Rectangular surface | Round surface

Fig. 9. Placing items on a rectangular or a round surface.

cost considering characters $c \in C$ of the spatial candidate γ as:

$$C_{\text{IA}}(\gamma) = \frac{1}{4|C|} \sum_{c} \left(2 - \langle \hat{\mathbf{v}}_{ci}, -\hat{\mathbf{F}}_i \rangle - \langle \hat{\mathbf{F}}_{ch}, \hat{\mathbf{v}}_{ci} \rangle\right). \tag{2}$$

Here, $\hat{\mathbf{v}}_{ci} = \frac{\rho_i - \rho_c}{|\rho_i - \rho_c|}$ denotes the direction from character c to item i. $\langle \hat{\mathbf{v}}_{ci}, -\hat{\mathbf{F}}_i \rangle$ evaluates whether character c is posed towards the front direction of item i (note that it could also be a real furniture object f in the room, e.g., a TV, but we use just i here for brevity). $\langle \hat{\mathbf{F}}_{ch}, \hat{\mathbf{v}}_{ci} \rangle$ evaluates whether character c's head is posed towards item i.

Note that for an activity that involves a character c physically interacting with an item i, we regard item i as accessible only if its distance from the character $|\rho_i - \rho_c|$ is less than a threshold D_{acc} (we set $D_{\text{acc}} = 0.5$m). If the distance exceeds this threshold, our approach just takes the upper limit of the cost for this character without evaluating the direction relationships.

Group Activity. Some activities require group participation. We constrain the characters in a group activity to keep reasonable poses with respect to each other in mutual gaze scenarios [Admoni and Scassellati 2017; Argyle and Cook 1976]. To achieve this, we approximate the gaze relations from a geometrical perspective.

For a group of two characters, ideally the two characters should face each other, and thus the relation can be constrained akin to the individual activity cost as in (2).

For a group of more than two characters, we define the group activity cost with the following assumptions. Regarding characters as points on the 2D plane, they form a group convex hull. A character c in the group should see as many other group members as possible within its field of view. We use a 60-degree field of view to mimic the near peripheral vision [Grosvenor 2007] of a character c. Denote a character group as G_C. We define the group activity cost of the spatial candidate γ as:

$$C_{\text{GA}}(\gamma) = \frac{1}{|\{G_C\}|} \sum_{G_C} \left(1 - w^{\text{a}}_{\text{GA}} \frac{\varphi_C}{\eta_C} - w^{\text{b}}_{\text{GA}} \frac{\phi_C}{\eta_C}\right), \tag{3}$$

where $\eta_C = \frac{|G_C|^2 - |G_C|}{2}$ is the total number of character pairs. φ_C is the number of pairs of characters who can see each other. ϕ_C is the number of pairs of characters with one character seeing the other character but not the other way round. We set the weights $w^{\text{a}}_{\text{GA}} = \frac{2}{3}$ and $w^{\text{b}}_{\text{GA}} = \frac{1}{3}$ to slightly favor the former case. Note that when iterating through all groups, the upper limit of the cost for a single group G_C is directly taken if there exists any character in the group that does not face the group convex hull.

Figure 8 shows examples of sampled group poses in a chat event with and without this cost consideration. Without the group activity cost, characters' poses are mainly constrained by the character pose cost, which is discussed later. In such a case, they only sit idle and match their body and head orientations with the character slots. As the figure shows, the upper-right character looks outside the group convex hull, and no others are visible to him.

In special cases where attention should be focused on a specific character (e.g. a case with an audience facing a speaker), we constrain the relation using a combination of individual activity (the audience characters that should look at the speaker) and group activity (the speaker should look into the group convex hull).

Character Pose. For a character c in either type of activity, we constrain its pose considering the character slot cs the character takes. We have two assumptions for comfortable poses: (1) The body's forward direction $\hat{\mathbf{F}}_{cb}$ matches the forward direction of the slot $\hat{\mathbf{F}}_{cs}$ (e.g., a chair's seat); (2) the head rotation θ_{ch}, which is relative to the body, is small. We set the maximum magnitude of head rotation as $\frac{\pi}{2}$. We define the character pose cost of a spatial candidate γ as:

$$C_{\text{CP}}(\gamma) = \frac{1}{2|C|} \sum_{c} \left[\frac{1 - \langle \hat{\mathbf{F}}_{cb}, \hat{\mathbf{F}}_{cs} \rangle}{2} + \frac{2|\theta_{ch}|}{\pi}\right]. \tag{4}$$

Item Placement. For a group of virtual items G_I placed together, we consider the following item placement constraints: (1) Each pair of items should keep a minimum distance of D_{id} from each other to avoid overlapping. We may use different values of D_{id} for different cases, depending on how closely we want the items to stay together; (2) The virtual items in the same group G_I should be tidily posed. Our approach considers placing virtual items onto the surfaces of two common types of shapes as shown in Figure 9. For a rectangular surface, we constrain each item to align with others along either axis of the 2D surface. For a round surface, we constrain the items to keep similar distances to the center of the surface. We define the item placement cost as follows:

$$C_{\text{IP}}(\gamma) = \frac{1}{|\{G_I\}|} \sum_{G_I} \left[w^{\text{a}}_{\text{IP}} [1 - \min(1, \frac{\lambda}{D_{\text{id}}})] + w^{\text{b}}_{\text{IP}} F_{\text{IP}}(G_I)\right]. \tag{5}$$

with

$$F_{\text{IP}}(G_{\mathcal{I}}) = \begin{cases} 1 - \frac{\varphi_{\mathcal{I}}}{\eta_{\mathcal{I}}} & \text{for a rectangular surface} \\ \\ \frac{\sigma_{\mathcal{I}}}{r} & \text{for a round surface,} \end{cases}$$

where λ in the first term is the minimum pairwise distance between any two items in $G_{\mathcal{I}}$. In the second term, $\varphi_{\mathcal{I}}$ refers to the number of pairs of items in alignment and $\eta_{\mathcal{I}} = \frac{|G_{\mathcal{I}}|^2 - |G_{\mathcal{I}}|}{2}$ is the total number of item pairs. $\sigma_{\mathcal{I}}$ is the standard deviation of the distances of the virtual items in $G_{\mathcal{I}}$ from the center and r is the radius of the round surface. We set the weights $w_{\text{IP}}^{\text{a}} = \frac{3}{5}$ and $w_{\text{IP}}^{\text{b}} = \frac{2}{5}$ by default.

6.1.2 Optimization. We use simulated annealing [Kirkpatrick et al. 1983] with a Metropolis-Hastings state-search step [Hastings 1970; Metropolis et al. 1953] to efficiently explore the solution space containing various spatial candidate solutions. We define a Boltzmann-like objective function:

$$f(\gamma) = e^{-\frac{1}{\tau} C_{\text{S}}(\gamma)}, \tag{6}$$

where τ is the temperature parameter in simulated annealing. The optimization proceeds iteratively. The temperature drops over iterations by a decay factor until it reaches a low value near zero. Such a setting makes the optimizer greedier in refining the solution towards the end of the optimization. The optimization process is terminated as the change in the total cost converges (changing by less than 3% over the past 100 iterations). By default, we set the weights as $w_{\text{IA}} = 3.0$, $w_{\text{GA}} = 1.5$, $w_{\text{CP}} = 1.0$ and $w_{\text{IP}} = 2.0$.

The main goal of this optimization is to prepare plenty of optimal spatial candidates for all spatial graphs through event frames in the story as Figure 10 shows, where rings denote spatial candidates. With pre-computed spatial candidates, the story assembly process (Section 6.2) does not evaluate the detailed AR contents' poses regarding spatial constraints, thus enhancing the assembly efficiency to support interactive and dynamic needs. For each spatial graph, we repeatedly apply the optimization starting from a random initialization to prepare 50 spatial candidates in our implementation.

6.1.3 Proposed Moves. At each iteration of the optimization, our approach applies a move on the current sample to propose a new sample. There are three types of moves described as follows.

(i) *Character Teleportation.* Characters are discretely positioned in our work. Therefore we define a character teleportation move to assign a character to a different slot. This move randomly chooses a character c and then randomly picks another character slot of the same semantic label as that of c's current slot. If there is no other character occupying that slot, c is instantly teleported there, with c's body orientation θ_{cb} set to be along the forward direction of that slot and head orientation θ_{ch} reset to zero. If the target slot is already occupied by another character, the move swaps the characters' positions, as well as their body and head orientations.

(ii) *Item Translation.* Virtual items are placed on planes which support continuous translations. We define an item translation move to translate a randomly selected virtual item i. It adds a movement amount $\delta\rho$ drawn from a bivariate normal distribution to item i's position.

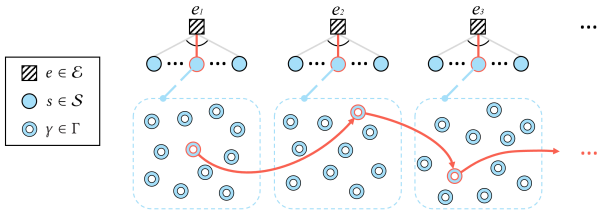


Fig. 10. During story assembly, for each event e, an event instantiation is selected, which comprises a selected spatial graph (rooted at a node s) together with its instantiation (i.e. a spatial candidate γ). Red colors refer to the selected spatial graphs and spatial candidates of an assembled story.

(iii) *Rotation.* Either a character c or an item i is randomly picked. If a character is picked, a rotation amount $\delta\theta$ is drawn from a normal distribution and added to its head orientation θ_{ch} (with a probability of 0.6) or body orientation θ_{cb} (with a probability of 0.4). Similarly, if an item is picked, a random rotation amount is added to its root orientation θ_i.

By default, move (i), (ii), and (iii) are selected with probabilities of 0.1, 0.15, and 0.75, respectively.

6.2 Story Assembly

Using the sampled spatial candidates for the spatial graphs, the story assembly stage dynamically assembles a story during the AR runtime to fit with the player's behaviors and the environment. Note that this temporal story assembly process does not evaluate the detailed spatial relations, which have already been considered in spatial candidates sampling stage. Essentially, our approach transfers the search for story solutions from the tangled spatial and temporal domains into the domain of spatial candidates γ, and only evaluates the temporal relations in story assembly, thus enhancing its efficiency for supporting dynamic animation updates in AR.

Formally, the whole story is represented as a set of event instantiations $\Psi = \{\hat{e}_t\}_{t=1,2,\ldots,T}$. Based on our definition, each $\hat{e}_t$ is an instantiation of event e associated with an event graph $\mathcal{G}_{e_t}$ by selecting (1) a branch of spatial graph $SG_{t,u}$; and (2) a spatial candidate $\gamma_{t,u,v}$ from the spatial candidate set $\Gamma_{t,u}$ of spatial graph $SG_{t,u}$. Figure 10 shows the process. Akin to the spatial candidates sampling stage, we apply another MCMC sampler for the story assembly stage with a different cost function to model the energy landscape considering the temporal relations.

6.2.1 Cost Function. We use a cost function to encode the temporal constraints. Overall, the temporal cost function for a sampled story Ψ is a weighted sum of normalized cost terms:

$$C_{\text{T}}(\Psi) = w_{\text{TL}} C_{\text{TL}}(\Psi) + w_{\text{PC}} C_{\text{PC}}(\Psi) + w_{\text{OSC}} C_{\text{OSC}}(\Psi). \tag{7}$$

We describe the details of the cost terms in the following.

Trajectory Length. As a character's position changes when transitioning from event frame e_{t-1} to e_t, it will need to navigate and change its pose from $(\rho_c, \theta_{cb}, \theta_{ch})_{t-1}$ to $(\rho_c, \theta_{cb}, \theta_{ch})_t$. Denote a single trajectory as χ, which is estimated as the shortest path in the given environment for such a transition. We aim to reduce the navigation distances of characters throughout the story, in order

to further reduce the overall characters' navigation time in interactive storytelling (Section 7). We consider two aspects regarding the trajectory lengths: (1) Reducing the overall trajectory lengths of characters in the same event; (2) Balancing the trajectory lengths of different characters to avoid the situation that one takes a short path while another takes a long path. The cost considering the trajectory set $\{\chi\}$ on an event instantiation $\hat{e}$ in Ψ is:

$$C_{\mathrm{TL}}(\Psi) = \frac{1}{|\Psi|}\sum_{\hat{e}}\left[\frac{w^{\mathrm{a}}_{\mathrm{TL}}}{|\{\chi\}|}\sum_{\chi}(1-\frac{|\chi|_*}{|\chi|}) + w^{\mathrm{b}}_{\mathrm{TL}}(1-e^{-\sigma_\chi})\right], \quad (8)$$

where $|\chi|_*$ is the estimated lower bound (i.e. navigating to the nearest character slot) for a character on the event instantiation $\hat{e}$. σ_χ is the standard deviation of trajectory lengths in $\{\chi\}$. We set $w^{\mathrm{a}}_{\mathrm{TL}} = \frac{3}{4}$ and $w^{\mathrm{b}}_{\mathrm{TL}} = \frac{1}{4}$ as the weights of the two terms.

Position Consistency. Under certain conditions, a character's position likely remains consistent even if the activity changes. For example, if a character's activity is *"sit on the sofa and watch tv"* at event frame e_{t-1}, and it becomes *"sit on the sofa and chat"* at event frame e_t, we assume that the character should stay at the same slot. We define a cost to consider the positions of characters $c \in C$ through all adjacent events e_{t-1} and e_t in Ψ:

$$C_{\mathrm{PC}}(\Psi) = \frac{1}{|\Psi|-1}\sum_{t}\left[\frac{1}{|C|}\sum_{c}F_{\mathrm{PC}}(c,t)\right], \quad (9)$$

where F_{PC} is a function that checks if such a consistency should exist for c between event instantiations $\hat{e}_{t-1}$ and $\hat{e}_t$. It returns 1 if such a consistency should exist but c's position changes, and it returns 0 otherwise.

Overall Spatial Cost. Each selected event instantiation $\hat{e}$ in Ψ has its own spatial cost, which is an optimization target itself in the temporal domain. Essentially, our approach aims to assemble a story using event instantiations associated with spatial candidates with low spatial costs. We define the overall spatial cost considering event instantiations $\hat{e} \in \Psi$ as:

$$C_{\mathrm{OSC}}(\Psi) = \frac{1}{|\Psi|}\sum_{\hat{e}}C_{\mathrm{S}}(\gamma), \quad (10)$$

where the spatial candidate γ is associated with the current event instantiation $\hat{e}$.

6.2.2 Optimization. Akin to spatial candidates sampling, we use simulated annealing with the Metropolis-Hastings criterion to optimize a story assembly. The initial temperature in this stage is set lower to make the optimization greedier, given pre-computed optimal spatial candidates. Starting from a random combination of event instantiations, our approach searches for new story assemblies iteratively and outputs an optimized solution in the end. By default, we set the cost weights as $w_{\mathrm{TL}} = 1.0$, $w_{\mathrm{PC}} = 2.0$ and $w_{\mathrm{OSC}} = 0.5$.

6.2.3 Proposed Moves. When operating in the domain of spatial candidates γ, the story assembly moves do not directly change detailed attributes (ρ, θ) for characters and items, but propose to switch the selection of spatial candidate γ for one event instantiation $\hat{e}$ at a time. There are three steps in proposing a move:

(1) Randomly select one event frame e_t;

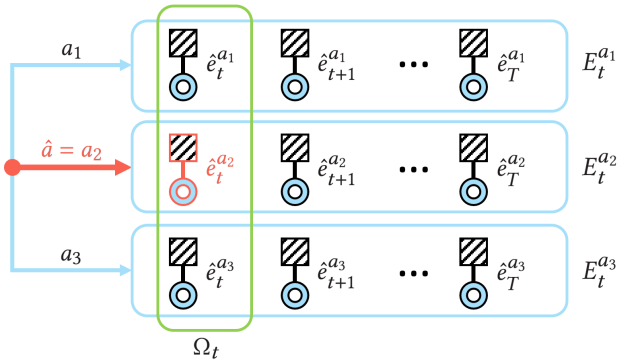


Fig. 11. An illustration of dynamic story assembly at event frame e_t. Event instantiations $(\hat{e}_t^a)$ are precomputed for all possible player actions $a \in A$ in this event and saved in the buffer Ω_t, and one is picked for animation based on the estimated action $\hat{a}$ at runtime. Red color denotes the estimated action and selected event instantiation.

(2) Decide whether to switch to another spatial graph branch according to the probabilistic model $\mathcal{P}$ of event graph $\mathcal{G}_{e_t}$;
(3) Within the spatial candidates set Γ, randomly select a target candidate γ' from the previous candidate γ's k nearest neighbors (we set $k = 5$) to allow a moderate update. The distance between two candidates is measured in Manhattan distance.

7 INTERACTIVE AR STORYTELLING

Synthesizing fixed stories for a group of virtual characters adapted to a real scene is feasible by running the hierarchical story sampling pipeline described in Section 6. In this work, we further describe how our approach can support interactive AR storytelling, enabling a player to participate as a character in the story and interact with virtual contents in augmented reality.

Our approach is human-centric in that the player has the priority to make decisions in the story, and the virtual characters and events will adapt to the decisions. For example, if all characters should take a seat in a living room in an event, the AR player can be the first one to choose where to sit, and all other characters will make their choices accordingly. More specifically, the action a character should take in an event includes: (1) where to stay, which means picking a specific character slot; (2) what item to interact with, which means deciding if the character should interact with a virtual item and which instance to pick. Such considerations make the AR storytelling experience dynamic and interactive. The player's actions determine how the story evolves.

7.1 Dynamic Story Assembly

In an interactive AR storytelling application, a player continuously participates in the story and takes actions. Our approach should update the assembled story accordingly. A main challenge is that updating the story at runtime based on the player's actions is computationally expensive. Our approach overcomes this challenge by precomputing spatial candidates in the preprocessing stage before starting the AR application. The spatial candidates are optimized with respect to different spatial considerations (e.g., group activities,

ALGORITHM 1: Dynamic Story Assembly

Input: Event sequence $\mathcal{E}$, spatial candidate sets $\{\Gamma_{t,u}\}$ for all spatial graphs SGs, and possible player actions $\{A_t\}$ for all events
Output: Animation of assembled story $\Psi = \{\hat{e}_t\}_{t=1,2,...,T}$
for $t = 1$ **to** T **do**
  Buffer event instantiation set $\Omega_t = \{\}$
  /* Precompute event instantiations for all possible actions */
  for $a \in A_t$ **do**
    Update $\mathcal{G}_{e_t}$ given a (i.e., disabling the selection of spatial graphs SGs and spatial candidates in $\Gamma_{t,u}$ that violate the context when a is taken)
    Sample E_t^a with the player character fixed given a (Sec. 6.2)
    Add $\hat{e}_t^a \in E_t^a \rightarrow \Omega_t$
  end
  /* Dynamically estimate player actions and update animation */
  do
    Estimate the player's action $\hat{a}$ (Sec. 7.2)
    Switch animation based on $\hat{e}_t^{\hat{a}} \in \Omega$
    $\hat{e}_t = \hat{e}_t^{\hat{a}}$
  while *event* $\hat{e}_t$ *is animating*;
end

individual activities, item placement). During the runtime, using the precomputed spatial candidates, our approach efficiently updates (reassembles) the story according to the player's behaviors in the real environment. The update is performed repeatedly for each event until the end of the story.

Concretely, before the start of each event e_t, we sample the remaining story represented using $E_t^a = \{\hat{e}_k^a\}_{k=t,...,T}$ with respect to each possible player action a in that event. While we only care about the selected event instantiation $\hat{e}_t^a$ in the sampled story for a given action a at event frame e_t, our approach samples the whole event instantiation set E_t^a to minimize the temporal costs that could be introduced in the following events. To prepare for animating the event e_t dynamically, a buffer event instantiation set Ω_t is used to save precomputed event instantiations corresponding to all possible player actions $A = \{a\}$ such that $\Omega_t = \{\hat{e}_t^a | \forall a \in A\}$. An animator (Section 7.2) estimates the player action $\hat{a}$ at runtime and determines the action eventually taken, and animates the AR content until event e_t ends. We describe the dynamic story assembly process in Algorithm 1 and illustrate the idea in Figure 11.

7.2 Animating the Assembled Story

Recall that we define a story as a sequence of events that describe character states at discrete event frames. The animator of our approach visualizes the characters' states at each frame and interpolates the intermediate states between any two adjacent frames. Specifically, it animates a sampled story by the following:

- At each event frame, place each character with the proper pose according to its high-level labels $(pose, verb, interactee)$ and low-level position and orientation attributes $(\rho_c, \theta_{cb}, \theta_{ch})$.
- During the transition from one event to the next event, if the state of a character changes, interpolate intermediate animation states accordingly. If a character's position changes, navigate the character to the new destination.

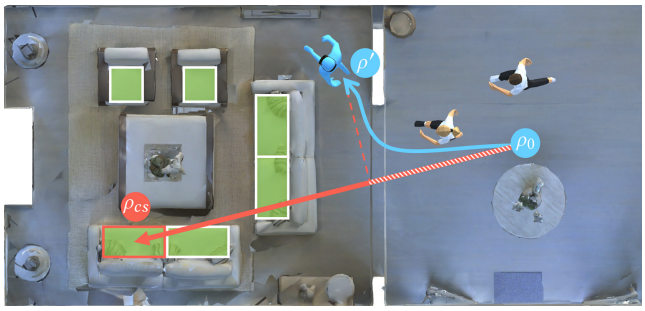


Fig. 12. Action progress estimation in an event. The predicted player action $\hat{a}$ is associated with a destination character slot at ρ_{cs}. The blue curve shows the player's trajectory in this event, which started from ρ_0 and has progressed to ρ' currently. The estimated action progress for action $\hat{a}$ is indicated by the striped red-white line segment, which refers to the projection of the player's displacement onto $\overrightarrow{\rho_0\rho_{cs}}$. Depending on the player's action progress, the other virtual characters adjust their walking speeds.

When transitioning to the next frame, to give priority to the player's decisions, the animator adjusts the speed of the virtual characters according to the player's behaviors in real time. Two main strategies are applied: (1) We estimate the player's speed μ_P and set the speeds of the virtual characters as $\mu_c = h_\mu \mu_P$, where $h_\mu \in (0, 1)$ is a speed control hyper-parameter; (2) We also estimate the player's action $\hat{a}$, the destination character slot of this estimated action, and the action progress (i.e. completion progress to the destination). Depending on the action progress, our approach adjusts the speed of the virtual characters.

Figure 12 shows an example. Suppose the player starts at ρ_0 and navigates to a destination character slot at ρ_{cs}, and the player's current position is ρ'. The action progress is estimated as the projected vector on $\overrightarrow{\rho_0\rho_{cs}}$, which is $\frac{\overrightarrow{\rho_0\rho_{cs}} \cdot \overrightarrow{\rho_0\rho'}}{|\overrightarrow{\rho_0\rho_{cs}}|^2}$. Similarly, we estimate other virtual characters' action progresses to their destinations.

Until the player arrives at a particular character slot, the player's action a is still not finalized, and hence the destinations for the player and the other virtual characters remain uncertain. Therefore, to play safe, we take the action $a \in A$ with the lowest progress between all possible actions as the estimated player's action $\hat{a}$. If a virtual character's action progress is more than h_p of the progress of the estimated player's action $\hat{a}$, where $h_p \in (0, 1)$ is a progress control hyper-parameter, the virtual character's speed is set to a predefined minimum speed $\mu_{\min} = 0.1\text{ms}^{-1}$ in our implementation. If the player has already arrived at the destination, the virtual character's speed is set to a predefined normal speed $\mu_{\text{norm}} = 1.0\text{ms}^{-1}$ in our implementation. We set $h_\mu = 0.7$ and $h_p = 0.8$ by default.

Note that at each event, as long as the player has arrived at a character slot, our approach immediately starts to sample story assemblies for the next event. By doing so we aim to minimize the wait time of the sampling process before the new event starts as navigating virtual characters to their destinations also takes time.

8 EXPERIMENTS AND RESULTS

We show experiment results and discussions in this section. To show third-person view that includes both the AR player and the virtual

Table 1. Descriptions of the selected events of the four stories in different indoor scenes. [P] denotes the AR player; [m] denotes the manager; [prof] denotes the professor; and others are general characters without specific roles.

Apartment	**Office**	**Teaching building**	**Makerspace**
Event 1: [P] uses a cell phone in the dining area, [a] cooks in the kitchen, [b] uses a computer in the living room	Event 1: [m] hosts a meeting in the meeting area, [P,a,b] listen to [m]	Event 1: [P] reads a book, [a] texts [prof] to make an appointment, in the meeting area	Event 5: [P,a,b] edit digital designs on computers in the media room
Event 7: [a] makes coffee, [P,b] carry coffee, in the kitchen	Event 3: [P,a,b] use computers in the workspace	Event 4: [P,prof,a] discuss in the discussion area	Event 7: [P,a] watch the 3D printing in the makerspace
Event 8: [P,a,b] chat in the living room	Event 8: [P,m,a,b] chat in the meeting area	Event 8: [P] presents the project to [prof,a,b,c] in the meeting area	Event 8: [P,a,b] chat in the makerspace

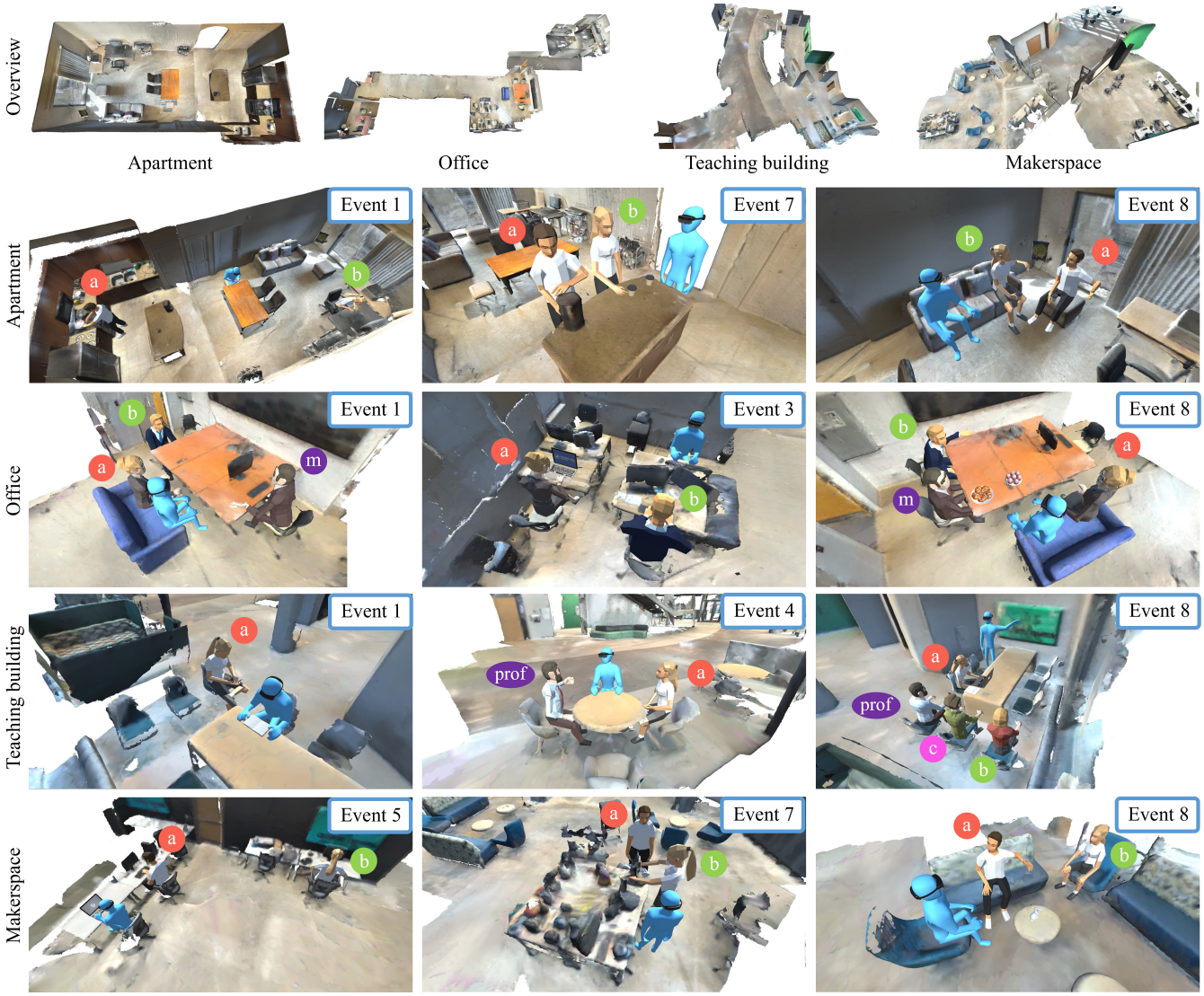


Fig. 13. Results of selected events described in Table 1 in four different scenes. The input scenes are shown at the top. The blue avatar denotes the AR player, whose pose is approximated from the recorded AR headset trajectory considering specific events.

characters, we recorded AR player trajectories during the experiments and used the blue avatar to recover interactive AR story. Note that since we were only able to track the position and orientation of the AR headset, we used an inverse kinematics algorithm [Aristidou et al. 2018] to approximate the human pose and assigned proper animations to the stickman according to the states of the character that the AR player took in certain story events. For example, in an event that the player should sit down and use a computer, we set a *sitting* pose and retrieved the hand gestures of hovering over the keyboard from the associated animation. While our main paper shows static screenshots, we replay the recorded dynamic storytelling trajectories in our supplementary video for all experiments in this section. We include additional experiments of augmenting physical environments for AR stories and retargeting AR stories to different scenes in our supplementary material.

8.1 Implementation

We conducted our experiments on a machine equipped with 32 GB of RAM, an Intel Core i7-9700 CPU and an NVIDIA GeForce RTX 2070 GPU. The overall framework was built using the Unity3D game engine. We used a Microsoft HoloLens 2 to provide AR experiences. All of the four different stories in our experiments comprised eight events, and in our scenes with moderate numbers of character slots, spatial candidates sampling for each took about 3−5 minutes. During the dynamic storytelling process, the preparation time for the buffer event instantiation set at an event frame, which depends on the number of possible player actions and the number of remaining events to instantiate, ranged from about 0.5 seconds to 5 seconds.

8.2 AR Storytelling in Real Environments

We evaluate the overall framework in four different real world environments: an *apartment*, an *office*, a *teaching building* and a *makerspace*. Different stories, each consisting of eight events, are adapted to the four scenes. Table 1 shows three selected events for each story, which are visualized in Figure 13. Our supplementary material contains full descriptions of the stories.

In general, the assembled stories are shaped well by the cost function. First, AR contents are reasonably placed into the real environments such that they do not violate the scene geometries (e.g. overlapping with the environments) and semantics (e.g. standing on a chair or sitting on a stove). Second, virtual characters are naturally posed to achieve activities described in the events: For individual activities, characters keep probable gestures and interact with either real furniture objects or virtual items, like touching either virtual laptops or real PCs in the *office*. For group activities, they are able to keep mutual gazes with others. Third, temporal relations are embodied during the AR storytelling. For example, in Event 7 of the *apartment* story, characters [P] and [b] go to slots where coffee cups are accessible, which are placed in the previous event. Consistency constrained by Equation 9 is also reserved between event frames, which is better visualized in our supplementary video. Fourth, the coexistence of players and virtual characters helps validate the interactiveness. These results suggest that our approach is capable of assembling different stories and showing them in different types of environments given specific scene geometries and semantics.

8.3 Adaptive AR Activities

In Figure 14, for each event we show the virtual character's behaviors with respect to a certain action of the AR player. As described in Section 7.1), our approach should dynamically adapt the virtual character's behaviors according to the AR player's action. To validate that event instantiations produced by our approach are adaptive to the player's actions, we pick one event from each story in the previous experiment and show such variations. Figure 14 shows the results. All these events include the AR player and multiple virtual characters, who headed to sittable character slots in the same room. Since we prioritize the AR player's actions, the virtual characters' behaviors adapted to the player's actions in such events. According to the results, when the player took different character slots, virtual characters always picked other slots accordingly, while keeping reasonable gestures that fit the contexts of the events.

9 USER STUDY

We performed a user study to validate the quality of AR stories assembled by our approach. As our formulation is centered around the realistic placement of virtual characters and objects in the scene with respect to the story events, we aim to evaluate the plausibility of the synthesized AR contents, taken as "how realistic, natural, and believable the AR contents are considering their poses". We explained this term to designers who manually arranged AR contents to create stories for comparison purposes, and to user study participants who experienced and rated the stories.

9.1 Settings

We compared the outputs of our approach with three manually created stories. The *office* story described in Table 1 was selected in this study. The manual stories were created by three designers, who have more than 2 years of VR game and simulation development experiences and are familiar with level design using Unity3D. The manual creation mainly included clicking and dragging operations using the Unity3D game engine to translate and rotate AR contents. Since fulfilling all possible branches of events in a story manually is extremely time-consuming, we only asked each designer to choose one branch for each event, thus a single path was taken on the story graph for each story. The goal was to place virtual characters and items as plausibly as possible, while preserving the temporal consistency, and trying to minimize the overall walking distance of characters. We denote the manual stories trials as I^m, II^m and III^m. To allow reasonable comparisons, we ran our approach to sample along the same selected branches for each manual solution, and thus produced the corresponding stories I^o, II^o and III^o. Note that for such a reason, stories were not dynamically assembled at runtime but precomputed. In total, we formed three comparison conditions I, II and III, each including two story trials.

We recruited 30 participants to experience those stories. All participants were undergraduate or graduate students, including 8 females and 22 males aged 18 through 31. All of them had no experience with AR before. Each participant was asked to experience one of the three conditions. Through a pseudorandom procedure, we ensured that (1) when experiencing one condition, whether the manual story or the story created by our approach came first was counterbalanced;

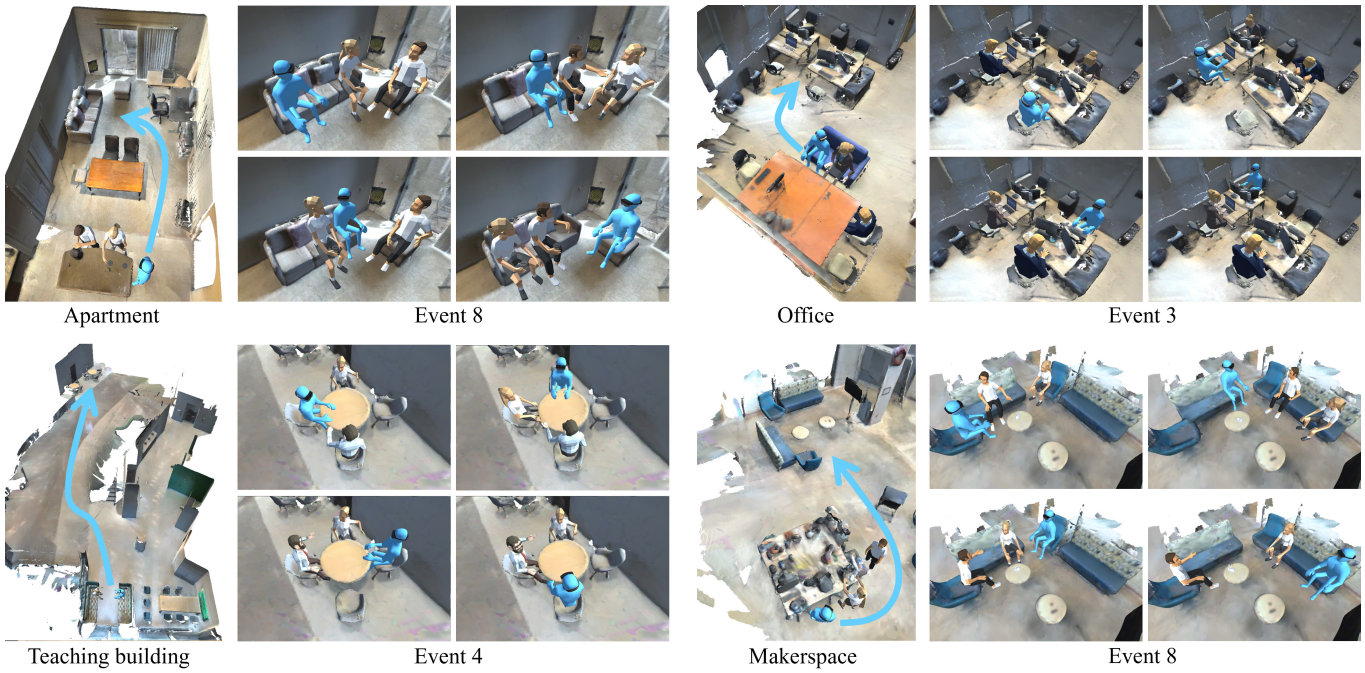


Fig. 14. Some events where the virtual characters' behaviors adapt to the player's actions. For each scene, virtual characters' selections of character slots and poses change according to the player's actions as shown. The blue avatar denotes the AR player, whose pose is approximated from the recorded AR headset trajectory considering specific events.

(2) the occurrences of all conditions were counterbalanced such that there were 10 records for each condition.

9.2 Procedure

Our user study procedure was approved by the Institutional Review Boards. One round of study comprised two stages as follows:

Warm up. Before experiencing the prepared AR stories, we showed demo virtual characters and items near the meeting area in the *office* scene, and guided participants to observe the AR content. They could stay in the warm up session as long as they wanted. We assumed that all participants got familiar with how virtual events were visualized in AR after this session.

Storytelling. Participants were first informed about their role as an employee in the *office* story. In each event, they were told about what the event was and what they should do (including which character slot they should choose, since branches in each condition were pre-selected by designers). Participants were asked to pay attention to the poses of the virtual characters (especially when they were in individual activities or group activities) and the items.

After each trial, we asked three questions about how plausible the AR contents were. The questions related to our costs in the spatial candidates sampling included: (1) How plausible the characters were when they carried out activities individually; (2) How plausible the characters were when they carried out group activities; (3) How plausible the virtual items were. We used a 5-point Likert scale, with 1 meaning "the least plausible" and 5 meaning "the most plausible".

Table 2. Quantitative results of participants' ratings on the plausibility of individual activity (individual ACT.), group activity (group ACT.) and item placement (item), including the average (avg.), standard deviation (std.) and p value. I^o, II^o and III^o were synthesized by our approach. I^m, II^m and III^m were created by designers.

	Individual ACT.			**Group ACT.**			**Item**		
	avg.	std.	p	avg.	std.	p	avg.	std.	p
I^m	3.6	0.66	0.51	3.8	0.87	0.80	4.0	0.77	0.61
I^o	3.8	0.60		3.9	0.83		4.2	0.87	
II^m	3.7	0.49	0.70	4.1	0.30	0.56	4.1	0.83	0.60
II^o	3.8	0.60		4.2	0.40		4.3	0.78	
III^m	3.5	0.67	0.53	4.0	0.63	0.75	4.3	0.78	0.77
III^o	3.7	0.64		4.1	0.70		4.4	0.66	

9.3 Outcome and Analysis

Table 2 summarizes the quantitative results of user ratings on all trials. We performed a t-test to examine the differences in user ratings. The p-values indicate no significant differences in all conditions for the three questions. An interesting observation during the study was that ratings for group activity plausibility were generally higher than for individual activity plausibility. According to some post-study feedback, many participants felt more immersed into the stories when they participated in group activities as such scenarios appeared more interactive. The results suggest that our approach is capable of constraining the poses of both virtual characters and items, and that it can place virtual contents in real environments as plausibly as human designers do. We also collected user ratings on

additional metrics regarding their feelings about the participation experience. Similarly, no significant differences were found. Refer to our supplementary material for additional results and analysis.

However, our approach significantly reduces time expenses compared to manual creation by designers. While I^m, II^m and III^m took the three designers 27, 19, and 22 minutes to create, it took about 0.6 seconds to automatically assemble each of the I^o, II^o and III^o. We do not count the manual time expenses of using our approach here because manual efforts, including scene preprocessing (Section 4) and optimization parameter setting (Section 6) (if needed), are only done once for a scene to generate various story assemblies for that scene. Note that it took less time to sample the stories for this study compared to dynamic storytelling scenarios, as the sampler only explored alongside the selected branches by the designers. This further suggests that while our approach can easily handle dynamic and interactive AR storytelling, having designers manually create all possible story event scenarios is impractical.

10 DISCUSSION AND FUTURE WORK

We present a novel approach to deliver AR stories with multiple virtual characters and items in diverse indoor scenes considering scene semantics. Our hierarchical story sampling formulation supports efficient interactive storytelling, enabling an AR player to participate as one of the characters in a story. Through an user study, we validated that our approach can produce stories that are similarly plausible as human-designed AR stories while considerably reducing the labor work of designers.

In our work, we only use an AR headset for visualizing AR contents. We are interested in incorporating more hardware for enhancing AR storytelling experiences. Using only an AR headset for localizing the AR player is unstable especially when the player is moving around; and introduces misalignment between the AR contents and the physical environments, which is noticeable in some of our screenshots and videos. Using additional sensors and trackers to track body joints could not only improve the localization precision, but also provide higher-quality trajectories replay. Besides, accessories like haptic gloves could enhance the players' interaction with the virtual items, allowing them to sense the force feedback of grasping, pushing, pressing, etc., hence making the interactive AR experiences more interactive, realistic, and immersive.

Due to the limited 3D scan quality and 3D instance segmentation performance, our semi-automatic scene semantics extraction step still requires some manual refinement. In the future, if more powerful instance segmentation, 3D reconstruction and scene understanding methods become available, we can reduce the labor work or even make the whole story authoring process fully automatic.

The item alignment is evaluated by two cases referring to the shape of the supporting surface. In future work, it will be interesting to introduce scene geometry analysis to facilitate virtual item placement, akin to how 3D edges and planar surfaces are utilized for object placements in SnapToReality [Nuernberger et al. 2016].

Currently, our approach only supports fixed story plots. We are interested in integrating automatic story authoring into our method. For example, higher-level story descriptions in natural languages might be automatically converted into alternative story narratives, so as to grant the players more freedom to take desired actions rather than strictly following the fixed plots, and to further drive the whole story with dynamic events.

Our approach supports player interactions including choosing character slots and interacting with virtual items. In future work, we may include additional types of interactions, especially direct interactions between players and virtual characters, to make the AR storytelling experience more realistic and immersive. For example, it will be compelling to enable players to pass virtual items directly to virtual characters, and vice versa.

We employed an optimization method, which could flexibly incorporate additional considerations as extensions. However, some results synthesized by the optimization might slightly deviate from ideal arrangements. For example, in a group activity where a player talks to a character, the character may not directly face the player though it keeps the player within its peripheral vision as constrained by our cost function. We believe that learning-based approaches could capture some subtle characteristics not modeled by handcrafted heuristics. A possible extension is to leverage some 3D human motion synthesis techniques [Cao et al. 2020; Hassan et al. 2021a; Wang et al. 2021a,b] to adapt characters' behaviours to specific scene geometries and semantics.

Another limitation of our animator is that it simply waits if some characters have finished their animations but some have not. Future work may include asynchronous animations for different characters to make the whole storytelling process smoother and more natural. It will also be interesting to accommodate more characters. A possible solution is to apply our approach to sub-groups of characters. Finally, our approach may inspire exciting future efforts to develop multiplayer, interactive AR storytelling experiences.

In our user study, we asked participants to rate a condition after experiencing the whole trial. While this ensured the coherence of storytelling regarding temporal relations, an alternative setting is to let participants rate immediately after each event, thus they have the best memory of that event when rating considering spatial relations.

ACKNOWLEDGMENTS

We thank Linghe Zheng for her assistance with the experiments and the demonstrations. This project was supported by an NSF CAREER Award (award number: 1942531).

REFERENCES

Henny Admoni and Brian Scassellati. 2017. Social eye gaze in human-robot interaction: a review. *Journal of Human-Robot Interaction* 6, 1 (2017), 25–63.

Shailen Agrawal and Michiel van de Panne. 2016. Task-based locomotion. *ACM Transactions on Graphics* 35, 4 (2016), 1–11.

Raphael Anderegg, Loïc Ciccone, and Robert W Sumner. 2018. PuppetPhone: puppeteering virtual characters using a smartphone. In *Proceedings of the 11th Annual International Conference on Motion, Interaction, and Games*. 1–6.

Michael Argyle and Mark Cook. 1976. Gaze and mutual gaze. (1976).

Andreas Aristidou, Joan Lasenby, Yiorgos Chrysanthou, and Ariel Shamir. 2018. Inverse kinematics techniques in computer graphics: A survey. In *Computer graphics forum*, Vol. 37. Wiley Online Library, 35–58.

Yunfei Bai, Kristin Siu, and C Karen Liu. 2012. Synthesis of concurrent object manipulation tasks. *ACM Transactions on Graphics* 31, 6 (2012), 1–9.

Mark Billinghurst, Hirokazu Kato, and Ivan Poupyrev. 2001. The magicbook-moving seamlessly between reality and virtuality. *IEEE Computer Graphics and Applications* 21, 3 (2001), 6–8.

Zhe Cao, Hang Gao, Karttikeya Mangalam, Qi-Zhi Cai, Minh Vo, and Jitendra Malik. 2020. Long-term human motion prediction with scene context. In *European Conference on Computer Vision*. Springer, 387–404.

Justine Cassell and Kimiko Ryokai. 2001. Making space for voice: Technologies to support children's fantasy and storytelling. *Personal and ubiquitous computing* 5, 3 (2001), 169–190.

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. 2017. Matterport3D: Learning from RGB-D Data in Indoor Environments. *International Conference on 3D Vision* (2017).

Xiaojun Chang, Pengzhen Ren, Pengfei Xu, Zhihui Li, Xiaojiang Chen, and Alex Hauptmann. 2021. Scene Graphs: A Survey of Generations and Applications. *arXiv preprint arXiv:2104.01111* (2021).

Long Chen, Wen Tang, Nigel John, Tao Ruan Wan, and Jian Jun Zhang. 2018. Context-aware mixed reality: A framework for ubiquitous interaction. *arXiv preprint arXiv:1803.05541* (2018).

Mengyu Chen, Andrés Monroy-Hernández, and Misha Sra. 2021. SceneAR: Scene-based Micro Narratives for Sharing and Remixing in Augmented Reality. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 294–303.

Yifei Cheng, Yukang Yan, Xin Yi, Yuanchun Shi, and David Lindlbauer. 2021. SemanticAdapt: Optimization-based Adaptation of Mixed Reality Layouts Leveraging Virtual-Physical Semantic Connections. In *UIST*. 282–297.

Sung Ho Choi, Kyeong-Beom Park, Dong Hyeon Roh, Jae Yeol Lee, Mustafa Mohammed, Yalda Ghasemi, and Heejin Jeong. 2022. An integrated mixed reality system for safety-aware human-robot collaboration using deep learning and digital twin generation. *Robotics and Computer-Integrated Manufacturing* 73 (2022), 102258.

Zhi-Chao Dong, Wenming Wu, Zenghao Xu, Qi Sun, Guanjie Yuan, Ligang Liu, and Xiao-Ming Fu. 2021. Tailored Reality: Perception-aware Scene Restructuring for Adaptive VR Navigation. *ACM Transactions on Graphics* 40, 5 (2021), 1–15.

Matthew Fisher, Manolis Savva, and Pat Hanrahan. 2011. Characterizing structural relationships in scenes using graph kernels. In *ACM SIGGRAPH 2011 papers*. 1–12.

Matthew Fisher, Manolis Savva, Yangyan Li, Pat Hanrahan, and Matthias Nießner. 2015. Activity-centric scene synthesis for functional 3D scene modeling. *ACM Transactions on Graphics* 34, 6 (2015), 1–13.

Ran Gal, Lior Shapira, Eyal Ofek, and Pushmeet Kohli. 2014. FLARE: Fast layout for augmented reality applications. In *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 207–212.

Terrell Glenn, Ananya Ipsita, Caleb Carithers, Kylie Peppler, and Karthik Ramani. 2020. StoryMakAR: Bringing stories to life with an augmented reality & physical prototyping toolkit for youth. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–14.

Raphaël Grasset, Andreas Dünser, and Mark Billinghurst. 2008. Edutainment with a mixed reality book: a visually augmented illustrative childrens' book. In *Proceedings of the international conference on advances in computer entertainment technology*. 292–295.

Theodore P Grosvenor. 2007. *Primary care optometry*. Elsevier Health Sciences.

Abhinav Gupta, Scott Satkin, Alexei A Efros, and Martial Hebert. 2011. From 3d scene geometry to human workspace. In *CVPR 2011*. IEEE, 1961–1968.

Mohamed Hassan, Duygu Ceylan, Ruben Villegas, Jun Saito, Jimei Yang, Yi Zhou, and Michael J Black. 2021a. Stochastic scene-aware motion prediction. In *ICCV*. 11374–11384.

Mohamed Hassan, Partha Ghosh, Joachim Tesch, Dimitrios Tzionas, and Michael J Black. 2021b. Populating 3D Scenes by Learning Human-Scene Interaction. In *CVPR*. 14708–14718.

W Keith Hastings. 1970. Monte Carlo sampling methods using Markov chains and their applications. (1970).

Fengming He, Xiyun Hu, Tianyi Wang, Ananya Ipsita, and Karthik Ramani. 2022. ScalAR: Authoring Semantically Adaptive Augmented Reality Experiences in Virtual Reality. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*.

Chenfanfu Jiang, Siyuan Qi, Yixin Zhu, Siyuan Huang, Jenny Lin, Lap-Fai Yu, Demetri Terzopoulos, and Song-Chun Zhu. 2018. Configurable 3d scene synthesis and 2d image rendering with per-pixel ground truth using stochastic grammars. *International Journal of Computer Vision* 126, 9 (2018), 920–941.

Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. 2015. Image retrieval using scene graphs. In *CVPR*. 3668–3678.

Vladimir G Kim, Siddhartha Chaudhuri, Leonidas Guibas, and Thomas Funkhouser. 2014. Shape2pose: Human-centric shape analysis. *ACM Transactions on Graphics* 33, 4 (2014), 1–12.

Scott Kirkpatrick, C Daniel Gelatt, and Mario P Vecchi. 1983. Optimization by simulated annealing. *Science* 220, 4598 (1983), 671–680.

Yining Lang, Wei Liang, and Lap-Fai Yu. 2019. Virtual agent positioning driven by scene semantics in mixed reality. In *2019 IEEE VR*. IEEE, 767–775.

Changyang Li, Haikun Huang, Jyh-Ming Lien, and Lap-Fai Yu. 2021. Synthesizing scene-aware virtual reality teleport graphs. *ACM Transactions on Graphics* 40, 6 (2021), 1–15.

Manyi Li, Akshay Gadi Patil, Kai Xu, Siddhartha Chaudhuri, Owais Khan, Ariel Shamir, Changhe Tu, Baoquan Chen, Daniel Cohen-Or, and Hao Zhang. 2019. Grains: Generative recursive autoencoders for indoor scenes. *ACM Transactions on Graphics* 38, 2 (2019), 1–16.

Wei Liang, Xinzhe Yu, Rawan Alghofaili, Yining Lang, and Lap-Fai Yu. 2021b. Scene-Aware Behavior Synthesis for Virtual Pets in Mixed Reality. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–12.

Zhihao Liang, Zhihao Li, Songcen Xu, Mingkui Tan, and Kui Jia. 2021a. Instance Segmentation in 3D Scenes using Semantic Superpoint Tree Networks. In *ICCV*. 2783–2792.

David Lindlbauer, Anna Maria Feit, and Otmar Hilliges. 2019. Context-aware online adaptation of mixed reality interfaces. In *UIST*. 147–160.

Rui Ma, Honghua Li, Changqing Zou, Zicheng Liao, Xin Tong, and Hao Zhang. 2016. Action-driven 3D indoor scene evolution. *ACM Trans. Graph.* 35, 6 (2016), 173–1.

Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. 1953. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics* 21, 6 (1953), 1087–1092.

Microsoft. 2016. Fragments. www.microsoft.com/en-us/p/fragments/9nblggh5ggm8

Benjamin Nuernberger, Eyal Ofek, Hrvoje Benko, and Andrew D Wilson. 2016. Snaptoreality: Aligning augmented reality to the real world. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 1233–1244.

Sören Pirk, Vojtech Krs, Kaimo Hu, Suren Deepak Rajasekaran, Hao Kang, Yusuke Yoshiyasu, Bedrich Benes, and Leonidas J Guibas. 2017. Understanding and exploiting object interaction landscapes. *ACM Trans. Graph.* 36, 3 (2017), 1–14.

Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. 2018. Virtualhome: Simulating household activities via programs. In *CVPR*. 8494–8502.

Siyuan Qi, Siyuan Huang, Ping Wei, and Song-Chun Zhu. 2017. Predicting human activities using stochastic grammar. In *ICCV*. 1164–1172.

Siyuan Qi, Yixin Zhu, Siyuan Huang, Chenfanfu Jiang, and Song-Chun Zhu. 2018. Human-centric indoor scene synthesis using stochastic grammar. In *CVPR*. 5899–5908.

Shuwen Qiu, Hangxin Liu, Zeyu Zhang, Yixin Zhu, and Song-Chun Zhu. 2020. Human-Robot Interaction in a Shared Augmented Reality Workspace. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 11413–11418.

Dariusz Rumiński and Krzysztof Walczak. 2013. Creation of interactive AR content on mobile devices. In *International Conference on Business Information Systems*. Springer, 258–269.

Manolis Savva, Angel X Chang, Pat Hanrahan, Matthew Fisher, and Matthias Nießner. 2014. SceneGrok: Inferring action maps in 3D environments. *ACM Transactions on Graphics* 33, 6 (2014), 1–10.

Manolis Savva, Angel X Chang, Pat Hanrahan, Matthew Fisher, and Matthias Nießner. 2016. Pigraphs: learning interaction snapshots from observations. *ACM Transactions on Graphics* 35, 4 (2016), 1–12.

Sebastian Starke, He Zhang, Taku Komura, and Jun Saito. 2019. Neural state machine for character-scene interactions. *ACM Transactions on Graphics* 38, 6 (2019), 209–1.

Tomu Tahara, Takashi Seno, Gaku Narita, and Tomoya Ishikawa. 2020. Retargetable AR: Context-aware Augmented Reality in Indoor Scenes based on 3D Scene Graph. In *2020 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*. IEEE, 249–255.

Julien Valentin, Vibhav Vineet, Ming-Ming Cheng, David Kim, Jamie Shotton, Pushmeet Kohli, Matthias Nießner, Antonio Criminisi, Shahram Izadi, and Philip Torr. 2015. Semanticpaint: Interactive 3d labeling and learning at your fingertips. *ACM Transactions on Graphics* 34, 5 (2015), 1–17.

Jiashun Wang, Huazhe Xu, Jingwei Xu, Sifei Liu, and Xiaolong Wang. 2021a. Synthesizing long-term 3d human motion and interaction in 3d scenes. In *CVPR*. 9401–9411.

Jingbo Wang, Sijie Yan, Bo Dai, and Dahua Lin. 2021b. Scene-aware generative network for human motion synthesis. In *CVPR*. 12206–12215.

Tianyi Wang, Xun Qian, Fengming He, Xiyun Hu, Ke Huo, Yuanzhi Cao, and Karthik Ramani. 2020. CAPturAR: An augmented reality tool for authoring human-involved context-aware applications. In *UIST*. 328–341.

Kun Xu, Kang Chen, Hongbo Fu, Wei-Lun Sun, and Shi-Min Hu. 2013. Sketch2Scene: Sketch-based co-retrieval and co-placement of 3D models. *ACM Transactions on Graphics* 32, 4 (2013), 1–15.

Hui Ye, Kin Chung Kwan, Wanchao Su, and Hongbo Fu. 2020. ARAnimator: in-situ character animation in mobile AR with user-defined motion gestures. *ACM Transactions on Graphics* 39, 4 (2020), 83–1.

Yibiao Zhao and Song-Chun Zhu. 2013. Scene parsing by integrating function, geometry and appearance models. In *CVPR*. 3119–3126.

Jixuan Zhi, Lap-Fai Yu, and Jyh-Ming Lien. 2021. Designing Human-Robot Coexistence Space. *IEEE Robotics and Automation Letters* 6, 4 (2021), 7161–7168.

Zhiying Zhou, Adrian David Cheok, JiunHorng Pan, and Yu Li. 2004. Magic Story Cube: an interactive tangible interface for storytelling. In *Proceedings of the International Conference on Advances in Computer Entertainment Technology*. 364–365.

Song-Chun Zhu and David Mumford. 2007. *A stochastic grammar of images*. Now Publishers Inc.