Location-Aware Adaptation of Augmented Reality Narratives

Wanwan Li[★]
wli17@gmu.edu
George Mason University and
University of South Florida
Fairfax, Virginia, USA

Changyang Li[★]
cli25@gmu.edu
George Mason University
Fairfax, Virginia, USA

Minyoung Kim
mkim229@gmu.edu
George Mason University
Fairfax, Virginia, USA

Haikun Huang
hhuang25@gmu.edu
George Mason University
Fairfax, Virginia, USA

Lap-Fai Yu
craigyu@gmu.edu
George Mason University
Fairfax, Virginia, USA

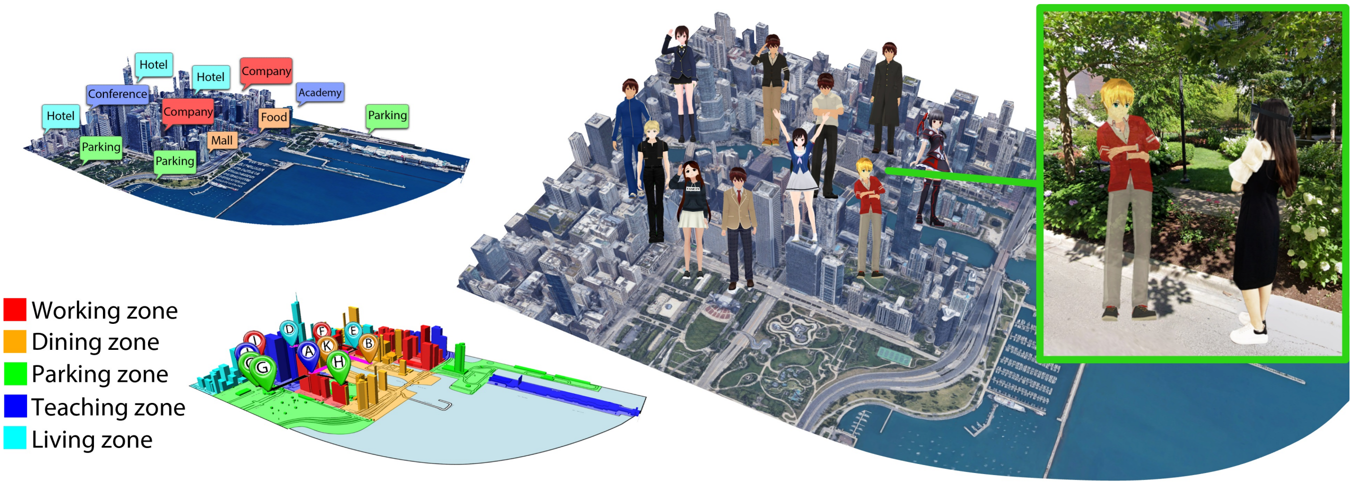


Figure 1: Given a map (of Chicago) with different zones and the narrative of a story as input, our approach automatically distributes the story events to compatible locations and synthesizes a navigation graph to guide the players to go through different story branches in augmented reality.

ABSTRACT

The recent popularity of augmented reality (AR) devices has enabled players to participate in interactive narratives through virtual events and characters populated in a real-world environment, where different actions may lead to different story branches. In this paper, we propose a novel approach to adapt narratives to real spaces for AR experiences. Our optimization-based approach automatically assigns contextually compatible locations to story events, synthesizing a navigation graph to guide players through different story branches while considering their walking experiences. We validated the effectiveness of our approach for adapting AR narratives to different scenes through experiments and user studies.

CCS CONCEPTS

• **Computing methodologies → Mixed / augmented reality**.

KEYWORDS

interactive narratives, augmented reality, storytelling, path generation

ACM Reference Format:
Wanwan Li[★], Changyang Li[★], Minyoung Kim, Haikun Huang, and Lap-Fai Yu. 2023. Location-Aware Adaptation of Augmented Reality Narratives. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23), April 23–28, 2023, Hamburg, Germany.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3544548.3580978

1 INTRODUCTION

Advances in augmented reality (AR) promise to change the way people experience stories in the future. Instead of watching TV or

[★] W. Li and C. Li contributed equally to this work.

CHI '23, April 23–28, 2023, Hamburg, Germany

ACM ISBN 978-1-4503-9421-5/23/04.
https://doi.org/10.1145/3544548.3580978

movies, people may participate in narratives to experience stories delivered via AR in a real space. Such AR-based stories involves a series of events where a player interacts with virtual characters and objects at real-world locations. Depending on the player's choices, the player may go through different story branches, encountering different events at different places.

Prior arts have investigated how to integrate virtual contents in real environments considering low-level scene details, including geometries [18, 41] and semantics [8, 10, 25, 33, 34, 36, 51]. As for the problem of adapting narratives to real-world maps, which needs to be addressed from a high-level perspective, a key challenge is that story events, which describe the characters involved and the interaction with the characters, are often related to certain contexts provided by the surroundings, and thus should happen at contextually compatible real-world locations to deliver a meaningful story. In recent years, there are rising research interests in location-aware applications [3, 4], for which outdoor scenes are especially important because they can accommodate larger-scale narratives with varied environmental features in comparison to indoor scenes. From a storytelling perspective, suppose a story contains an event where the player comes across a virtual character eating a sandwich. It would be natural for this event to happen in a dining area rather than a parking lot. Conventional location-aware AR creation requires developers to manually distribute story events to real-world maps with the help of development tools (e.g., ARKit [1], ARCore [2]), and thus needs intensive labor work. Furthermore, additional rounds of manual creations are needed to retarget the same story to different real environments. As players may be situated all over the world and may have different storytelling preferences, it is difficult for designers to adapt the same story to accommodate the different real environments and preferences of diverse players in a scalable manner.

To resolve this challenge, we propose an automatic approach to adapt interactive narratives to real-world maps by considering location compatibility (i.e. how compatible it is to assign specific story events to the location considering scene context), enabling the deployment of story plots for AR experiences at scale. Figure 1 illustrates our approach, which takes a map as an input and automatically distributes story events in the narrative to compatible locations considering zone types. Given self-defined location compatibilities of events that only need to be specified once, this automatic approach can retarget the same narrative efficiently to different real maps. Additionally, it synthesizes a navigation graph that guides players to go through different story branches in the real world while considering the walking experiences such as ensuring short walking distances.

By adapting narratives to real-world locations via an optimization, our approach help virtual content creators disseminate their stories in a fast, automatic, and scalable manner for AR experiences. The major contributions of our work include:

- Investigating a novel problem of automatically adapting interactive narratives to real-world places for AR experiences.
- Devising an optimization-based approach to automatically assign compatible locations to story events and synthesizing a navigation graph to guide players through story branches in the real world.
- Validating the efficacy of our approach by adapting several narratives to different real-world maps, examining the effectiveness of our optimization considerations via user studies, and testing the extensibility by additional design constraints.

2 RELATED WORK

2.1 Computer-Assisted Narrative Creation

Graphics and game researchers have created handy tools and user interfaces for authoring narratives. Zhang et al. [57] proposed an accelerated partial order planner for generating open world narratives. Braunschweiler et al. [5] created an event-based interactive storytelling system that offered free-form player experiences in virtual environments while following the designer's intent. McCoy et al. [40] created a system called Comme il Faut that provided a playable model of reusable and reconfigurable social interactions for authoring stories. Mason et al. [39] introduced the Lume system for procedural narrative generation. Regarding user interfaces, Kapadia et al. [27] introduced CANVAS, an accessible interface for synthesizing and visualizing narrative animations. Ha et al. [24] proposed physics storyboards comprising space-time snapshots that highlighted critical events and outcomes to facilitate parameter tuning of physics simulation games. Refer to Poulakos et al. [43] for a comprehensive review of narrative authoring tools. Although we study the adaptation of narratives rather than narrative creation, the prior arts inspire our representation of stories. Specifically, our work uses a story tree to incorporate story events and branches.

2.2 Interactive Narratives

Interactive narratives refer to a form of digital interactive experiences in which the story branches are given based on players' interactions with the story world [46]. Such narratives have been employed in various application scenarios. For example, Stock et al. [50] developed a system involving animated agents and adaptive video documentaries to guide museum visitors. Lim et al. [35] used improvisational storytelling to create a mobile tour guide. Gustafsson et al. [22] devised a game prototype that enabled players to experience narratives while traveling in a car.

As for interactive narrative generation, HEFTI [42] is an early system that uses genetic algorithms to recombine story components. El-Nasr et al. [14] presented an interactive narrative architecture designed using dramatic techniques based on training in film and theatre. Riedl et al. [45] developed a multiplayer storytelling engine to manage a story world at individual and group levels. Scherazade-IF [23] is an interactive narrative system that learns a domain model from crowdsourced example stories. Dominguez et al. [13] investigated the relationship between the sense of narrative roles and the options to choose during gameplay. More recently, Wang et al. [52, 53] used deep learinng methods for interactive narrative planning and personalization. Chung et al. [12] proposed TaleBrush, which used a line sketching interaction for intuitive control and sensemaking of narrative generation. Those methods could possibly be combined with collaborative creation methods like skWiki [58] for collaborative narrative generation.

[1] Apple ARKit. https://developer.apple.com/augmented-reality/arkit/
[2] Google ARCore. https://developers.google.com/ar

A related problem is behavior planning for narratives. For example, Cavazza et al. [7] proposed a method for planning the behaviors of artificial actors. Magerko and Laird [38] investigated finding a balance between the players’ degree of interaction and a story-based experience. Shoulson et al. [49] proposed an event-centric planning framework for directing interactive narratives in 3D environments with virtual humans. Ramirez and Bulitko [44] proposed an experience management system that combined generative experience management and player modelling.

As we focus on interactive narratives for AR experiences, placing virtual content in real environments coherently with both the physical space and story plots is critical. Therefore we address the problem of adapting narratives onto real maps. A similar problem was studied by Macvean et al. [37] who proposed WeQuest, an interactive tool for authoring alternate reality games. After a user distributes the events of a story on a map, WeQuest can retarget them to a different map considering the similarities of locations using people’s reviews on websites such as Yelp. In contrast, our approach reduces labor efforts by automatically assigning the events of a story plot to different compatible locations on a map, where compatibility can be defined by zones or inferred using street view images. As we demonstrate, our approach can sample different narrative adaptations on a single map and also retarget the same story to different maps. It can also incorporate additional design constraints such as event centers and landmark visibility, which are useful for AR experiences.

2.3 Storytelling in Augmented Reality

Prior works have studied AR storytelling for education and entertainment [21, 59]. Recently, with the increasing popularity of AR functionalities integrated into mobile devices, researchers devised mobile authoring tools for creating and telling AR stories. For example, Rumiński et al. [48] created a mobile AR authoring tool named MARAT. Kapadia et al. [26] created a computer-assisted authoring tool for interactive narratives. A system called StoryMakAR [20] combined electro-mechanical devices with virtual characters to create stories. Chen et al. proposed SceneAR [9], which was a mobile application for creating sequential scene-based AR narratives. Some other techniques used mobile devices to direct the behaviors of AR characters, like manipulating virtual puppets using smartphones [1], and controlling virtual characters’ animations in AR scenes using motion gestures of mobile phone [55].

With the increasing need of location-based or scene-aware AR applications, some recent works focus on creating adaptive AR experiences for specific environments or scenarios considering scene geometries and semantics. Gal et al. [18] created a rule-based framework named FLARE that used planar geometries to generate object layouts for AR applications. Nuernberger et al. [41] proposed SnapToReality, an alignment technique that helped users to pose virtual content to real environments. Chen et al. [8] presented a semantic-based framework for generating high-level context-aware interactions. A series of recent methods demonstrate how to automatically adapt virtual interfaces [10, 25] or layouts [36, 54] to real scenes with respect to the scene context. Another important topic is adapting virtual characters to real environments, like posing virtual humans in real scenes [31, 51] and synthesizing and animating behaviors of virtual humans or animals [33, 34]. Our work uses AR devices to deliver interactive narratives that are automatically adapted onto real-world maps via optimization.

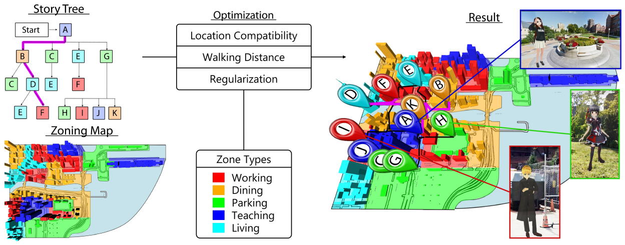


Figure 2: An overview of our approach, which distributes events to real-world locations and synthesizes a navigation graph to guide players to experience story branches at real-world places via augmented reality.

3 OVERVIEW

Figure 2 shows an overview of our approach. The input of our approach consists of a narrative represented as a story tree, and a map of the real-world environment where the AR experience will take place. The story tree comprises different story branches, each of which is composed of a series of story events. Each story event consists of a character interacting with the player in a certain way (e.g., saying a dialogue, performing an action) according to the story. A story event also carries an event location description which indicates its expected compatibility with different zone types (e.g., a “coming across a classmate” event may happen in the parking or teaching zone) of the real-world environment.

The map of the real environment consists of locations and roads that players can walk through to experience the story. The locations carry zone type information (e.g., working, dining, parking, teaching, living), which can be set by publicly-available zoning information of the city, by the designer’s manual specification, or by a data-driven zoning inference method.

Given the input, our approach automatically synthesizes a navigation graph that guides players to experience different story branches in augmented reality. A navigation graph is synthesized by solving an optimization problem comprising three major cost terms: (1) a location compatibility cost used to ensure that the events and their assigned locations are compatible regarding the zone types; (2) a walking distance cost used to constrain the lengths of walking paths in the adapted narrative; and (3) a regularization cost used to prompt the walking distances between the events to be evenly distributed. We also demonstrated in our experiments how to extend the framework to incorporate other design constraints.

4 PROBLEM FORMULATION

4.1 Representation

4.1.1 Story Tree. Interactive narratives with different story branches are created according to players’ interactions with the story during the gameplay [28, 47, 56]. A player’s actions will decide which story branches the player will enter into. We use a *story tree* to represent all story events and branches that the player can experience. Figure 3 shows an example story tree. The root of the tree corresponds to the first event (i.e. Event A). A traversal from the root down to a leaf node corresponds to a story branch containing a sequence of events that the player goes through in chronological order. At an intermediate event, the player can decide to take a certain next event down the tree leading to possible sub-branches. Formally, let the story event set $S = \{s_i\}_{i=1,\ldots,m}$ contain all possible events, and the story branch set $B = \{(\ldots, s_i, s_j, \ldots)\}$ contain all possible branches in the story.

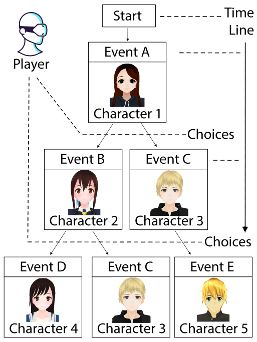


Figure 3: An example story tree.

In the example of Figure 3, there are five events in the story event set $S = \{A, B, C, D, E\}$, and six story tree nodes. The story branch set is $B = \{(A, B, D), (A, B, C), (A, C, E)\}$. Note that as multiple story branches may involve common events, multiple story tree nodes can be associated with the same event. For example, the story branches (A, B, D) and (A, B, C) share the same sub-sequence (A, B), so they share the nodes associated with events A and B in the story tree. On the other hand, as event C appears in both (A, B, C) and (A, C, E), two story tree nodes are associated with the event.

A story event s_i is attributed as $s_i = (e_i, l_i)$, where e_i is an event location description indicating what zone types would fit the event. For example, the event of "coming across a friend who is jogging" is more likely to take place in a living zone rather than a dining zone. l_i is the location (expressed in terms of the index of a vertex of the road network graph of the map) where the event happens. We explain these parameters in the following sections.

4.1.2 *Navigation Graph.*

Given a story tree that encodes the story branches and events, our approach synthesizes a *navigation graph* to distribute the events on a map. Graph representations have been used for effective scene-aware virtual reality experiences [32], and we propose navigation graphs to facilitate AR storytelling in this work. As shown in Figure 4, the streets and roads on the map form a road network through which players can navigate the environment. The map also contains information of the buildings, zone types, etc. which will be used for evaluating the environmental features for assigning semantically compatible locations to events. The distribution of events is also constrained by considering the navigation experiences (e.g., walking distances) of players according to the story branches.

Formally, a road network is a graph $G = (V, E)$, of which the vertices $V = \{v_i\}_{i=1,\ldots,n}$ denote the road endpoints and the edges $E = \{(v_i, v_j)\}$ are the roads. As Figure 4 shows, given story events S and branches B encoded by a story tree, a navigation graph $G' = (V', E')$, which is a subgraph of the road network G, can be sampled via a mapping reduction function $G' = \mathcal{F}(G, S, B)$ defined as follows:

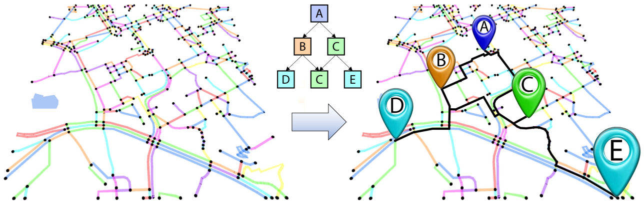


Figure 4: Navigation graph generation. Left: An input map showing a road network graph G with road crossings (depicted as black dots) and roads. Right: a navigation graph G' sampled from the road network based on the story tree.

(1) For each story event $s_i = (e_i, l_i)$, assign a unique location $l_i \in [1, n]$ corresponding to a vertex index in G;
(2) Based on the assigned locations of the story events, extract vertices V' for the navigation graph G' by mapping story events $S = \{s_i\}_{i=1,\ldots,m}$ to the vertices of the road network G such that $V' = \{v'_i\}_{i=1,\ldots,m} = \{v_{l_i}\}_{i=1,\ldots,m}$;
(3) Form edges E' of the navigation graph G' by mapping branches in B into edges such that $E' = \{(v_{l_i}, v_{l_j}) | \exists(\ldots, s_i, s_j, \ldots) \in B\}$. Here, $\exists(\ldots, s_i, s_j, \ldots) \in B$ means that there exists two consecutive events s_i and s_j in any of the story branches.

As shown in Figure 4, a navigation graph is sampled from a road network. Note that event C, which appears in two story branches on the tree, is assigned to only one unique location on the map. We enforce this unique location assignment for several reasons: as a player replays an experience using the same navigation graph, the player will experience the same event at the same location; similarly, if different or multiple players experience a story using the same navigation graph, the players will experience the same event at the same location; finally, such assumption would make it more intuitive to devise the navigation graph sampling formulation. For a special case involving looping missions, our approach can support it by duplicating the events of the loop with the same locations and appending the duplicated events to the story branch. Note that, in case a designer wants an event with the same characters and behaviors to happen at two different locations, the designer could simply define it as two separate events. Besides, additional vertices could be sampled (e.g., along a long straight road) for the road network graph if desired.

4.2 Event Location Compatibility

Assigning story events to compatible locations would enhance the immersiveness of AR experiences. We define a compatibility score function to evaluate how appropriate it is to assign an event to a specific location.

Consider a zoning map with k different zone types, corresponding to labels $Z_1, Z_2, \ldots, Z_k$. Given a location x on the map, we formulate $Z_i(x)$ as a random variable that obeys Bernoulli distribution, such that $\Pr[Z_i(x) = 1]$ denotes the probability that x belongs to the i^{th} zone type, and $\Pr[Z_i(x) = 0] = 1 - \Pr[Z_i(x) = 1]$ otherwise. For notation convenience, we use $\Pr[Z_i(x)]$ and $\Pr[\overline{Z_i(x)}]$ to denote these two probabilities for the variable $Z_i(x)$ and its complement $\overline{Z_i(x)}$. In our main results, such probabilities are directly derived from public zoning information released by governments, thus we set $\Pr[Z_i(x)] = 1$ deterministically if x is of type Z_i, or

$\Pr[Z_i(x)] = 0$ otherwise. Note that in certain scenarios where no zoning map is available or additional specific zone types are needed, the probabilities can also be manually set or inferred using data-driven methods. In Section 5.2, we show results generated by replacing zoning maps with inferences produced by an autoencoder. Note that a single location x may belong to multiple zone types, thus the random variables $Z_1(x), \ldots, Z_k(x)$ are independent of each other.

Recall that for each story event $s_i = (e_i, l_i)$, e_i is the event location description. We represent e_i as a Boolean formula that encodes how compatible the story event s_i is with locations of different zone types. An illustrative example is that a "meeting with a professor" event may have a high compatibility with a teaching zone Z_{teaching} or a working zone Z_{working}, but not a living zone Z_{living}, thus can be described as $e_i = (Z_{\text{teaching}} \vee Z_{\text{working}}) \wedge \overline{Z_{\text{living}}}$. Note that the event location description e_i of the story event s_i is predefined by the designer as a constant attribute associated with that event. In this example, the probability of satisfying the Boolean formula e_i at location x is $\Pr[e_i(x) = 1] = (1 - \Pr[\overline{Z_{\text{teaching}}(x)}]\Pr[\overline{Z_{\text{working}}(x)}])\Pr[\overline{Z_{\text{living}}(x)}]$, which is calculated based on the zone type distributions at x.

Given the event location description e_i of event s_i, we define the compatibility score function for s_i and a location x:

$$F_{\text{LC}}(s_i, x) = \lambda \Pr[e_i(x) = 1] + (1 - \lambda) \frac{\sum_{\gamma \in \Gamma} \Pr[\gamma(x) = 1]}{|\Gamma|}, \quad (1)$$

where $\lambda \in [0, 1]$ is a weight for balancing the two terms; $\Gamma = \{\gamma\}$ is the set of literals in the Boolean formula e_i, and each literal is either a variable or the complement of a variable. Figure 5 visualizes the location compatibility score maps for three different events considering their location descriptions.

The motivation for introducing the second term is to compensate for the strict calculation of probability in the first term. Suppose an event needs to be assigned somewhere of both zone types Z_1 and Z_2 in an ideal case, i.e. its event location description is $\mathbf{e} = Z_1 \wedge Z_2$. For two candidate locations x_a and x_b for this event, with $\Pr[Z_1(x_a)] = 1.0$ and $\Pr[Z_2(x_b)] = 0.0$; and $\Pr[Z_1(x_b)] = 0.0$ and $\Pr[Z_2(x_a)] = 0.0$, while both locations receive the same score 0.0 from the first term, location x_a is intuitively a closer fit. We set $\lambda = 0.8$ by default. Users who want stricter compatibility evaluation based on zone type probabilities may set higher λ values.

4.3 Cost Functions

Our approach aims to optimize the location assignment for all events in a narrative to distribute them reasonably in a real-world environment for AR experiences. We define a cost function $C_{\text{total}}(G')$ to evaluate the navigation graph $G' = (V', E')$ sampled from the road network graph $G = (V, E)$ on a real-world map. The cost function consists of three cost terms for evaluating three properties of the navigation graph G': the (1) location compatibility term $C_{\text{LC}}(G')$ evaluates how appropriate the events are correlated with their assigned locations in terms of zone types; the (2) walking distance cost term $C_{\text{WD}}(G')$ evaluates the walking distances following the navigation graph in the real world; and the (3) regularization cost term $C_{\text{REG}}(G')$ evaluates the distribution of walking distances between adjacent event locations. Mathematically, the total cost term $C_{\text{total}}(G')$ is defined as:

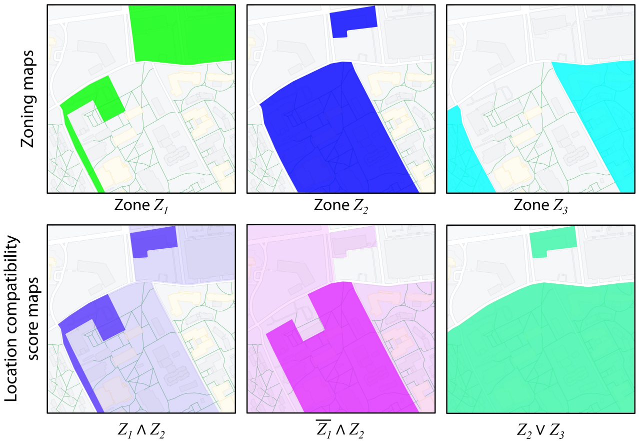


Figure 5: Example location compatibility score maps corresponding to two event location descriptions $Z_1 \wedge Z_2$, $\overline{Z_1} \wedge Z_2$ and $Z_2 \vee Z_3$. Colored masks of higher transparency denote areas of lower scores.

$$C_{\text{total}}(G') = w_{\text{LC}} C_{\text{LC}}(G') + w_{\text{WD}} C_{\text{WD}}(G') + w_{\text{REG}} C_{\text{REG}}(G'), \quad (2)$$

where w_{LC}, w_{WD}, and w_{REG} are the weights for the three cost terms, respectively. Our approach synthesizes a navigation graph G' by minimizing $C_{\text{total}}(G')$ through an optimization.

4.3.1 *Location Compatibility Cost.*

Given a navigation graph $G' = (V', E')$ sampled from the road network graph G, we evaluate how well the vertices of the navigation graph fit with their associated story events to deliver a realistic AR experience. Given a story event $s_i = (e_i, l_i)$ and its associated location $v'_i = v_{l_i}$, the evaluation is done by computing the compatibility score depicted in Equation 1. To evaluate the overall location assignments compatibility, we define a location compatibility cost for the navigation graph G':

$$C_{\text{LC}}(G') = 1 - \exp\left(\frac{1}{|V'|} \sum_{v'_i \in V'} [F_{\text{LC}}(s_i, v'_i) - 1]\right). \quad (3)$$

Essentially, the more compatible the locations of the vertices are with respect to the associated events, the lower the cost is.

4.3.2 *Walking Distance Cost.*

As the player walks physically in the real world to experience the interactive narratives via AR, it is important to consider the walking distance which is related to the player's navigation experience. For example, players may prefer not to walk too much unless necessary. We define a walking distance cost to incorporate such a consideration.

At this stage, we assume that the player may prefer a shorter walking distance as possible. Given the navigation graph $G' = (V', E')$ sampled from the road network graph G, we penalize the walking distance between every two consecutive events' vertices v_{l_i} and v_{l_j} considering the edges set $E' = \{(v_{l_i}, v_{l_j})\}$. Note that a designer may set a lower weight for the walking distance cost if needed, for example, in case a certain amount of walking is desired for a narrative that expects some walking.

To facilitate the computation of the walking distance cost, our approach first precomputes the minimum walking distance between every pair of vertices in the road network graph G. We denote the minimum walking distance between vertices v_i and v_j as $D(i, j)$, which is calculated using the Floyd–Warshall algorithm [16]. In practice, we precompute and store the shortest path for every pair of vertices. We define the walking distance cost as:

$$C_{\text{WD}}(G') = 1 - \exp(-\frac{\overline{D}}{D_{\max}}), \tag{4}$$

where $\overline{D}$ is the average walking distance of all pairwise adjacent event locations in navigation graph G', and $D_{\max}$ is a normalization term set as the maximum value among the minimum walking distances between any pair of vertices in road network graph G. Essentially, this cost term penalizes long walking distances in the sampled navigation graph G'.

4.3.3 Regularization Cost. Our approach can also incorporate other considerations in forming the navigation graph G'. For example, we define a regularization cost to balance the walking distances between consecutive events' locations. In other words, we penalize the biased distribution of walking distances between consecutive events' locations. Given a navigation graph $G' = (V', E')$, we measure the standard deviation of walking distances between consecutive events and penalize it through a regularization cost $C_{\text{REG}}(G')$ defined as:

$$C_{\text{REG}}(G') = 1 - \exp(\frac{\sigma_D}{\overline{D} - D_{\max}}), \tag{5}$$

where σ_D is the standard deviation of walking distances of all pairwise consecutive locations in navigation graph G'.

4.4 Graph Optimization

Recall that our optimization goal is to assign locations to the story events so that they are reasonably distributed in the real world for AR experiences. Hereby, we formulate the optimization problem as a graph sampling problem. We employ the Markov chain Monte Carlo method [19] to search for a solution that minimizes the total cost function $C_{\text{total}}(G')$, where $G' = \mathcal{F}(G, S, B)$ is sampled from the road network graph G.

At each iteration of the optimization, given the current story events S set with certain event location assignments, an updated set of story events S' with updated event location assignments is proposed through the following steps:

(1) Randomly select an event $s_i = (e_i, l_i)$ from story event set S;
(2) Create a new story event $s'_i = (e_i, l'_i)$ with a new random location vertex index l'_i that is not associated with any existing story event. That is, $l'_i \in [1, |V|]$, $l'_i \neq l_i$, and $\forall s_j \in S \rightarrow l'_i \neq l_j$.
(3) Create an updated set of story events S' by replacing story event s_i with s'_i such that $S' = S - \{s_i\} \cup \{s'_i\}$.

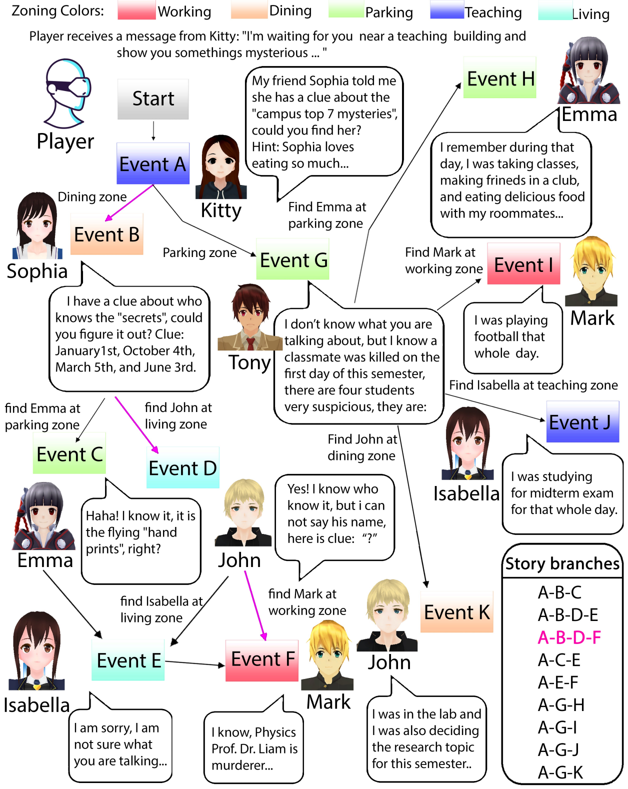

Zoning Colors: 🟥 Working 🟧 Dining 🟩 Parking 🟦 Teaching 🟩 Living
Player receives a message from Kitty: "I'm waiting for you near a teaching building and show you somethings mysterious ... "
Player
Start
Event A
Kitty
My friend Sophia told me she has a clue about the "campus top 7 mysteries", could you find her? Hint: Sophia loves eating so much...
Dining zone
Event B
Sophia
Parking zone
Event G
Find Emma at parking zone
Event H
Emma
I remember during that day, I was taking classes, making frineds in a club, and eating delicious food with my roommates...
I have a clue about who knows the "secrets", could you figure it out? Clue: January1st, October 4th, March 5th, and June 3rd.
Tony
I don't know what you are talking about, but I know a classmate was killed on the first day of this semester, there are four students very suspicious, they are:
Find Mark at working zone
Event I
Mark
I was playing football that whole day.
Find Isabella at teaching zone
Event J
Isabella
I was studying for midterm exam for that whole day.
find Emma at parking zone
find John at living zone
Event C
Event D
Find John at dining zone
Emma
Haha! I know it, it is the flying "hand prints", right?
John
Yes! I know who know it, but i can not say his name, here is clue: "?"
find Isabella at living zone
find Mark at working zone
Event K
John
Event E
Event F
Isabella
Mark
I am sorry, I am not sure what you are talking...
I know, Physics Prof. Dr. Liam is murderer...
I was in the lab and I was also deciding the research topic for this semester..
Story branches
A-B-C
A-B-D-E
A-B-D-F
A-C-E
A-E-F
A-G-H
A-G-I
A-G-J
A-G-K

Figure 6: Our illustrative story *Detective AR*. One story branch (A, B, D, F) is highlighted in purple to help with illustration.

Our approach then determines whether to accept the proposed story events S' by computing an acceptance probability $\Pr(S'|S)$ using the Metropolis criterion [11, 30], as follows:

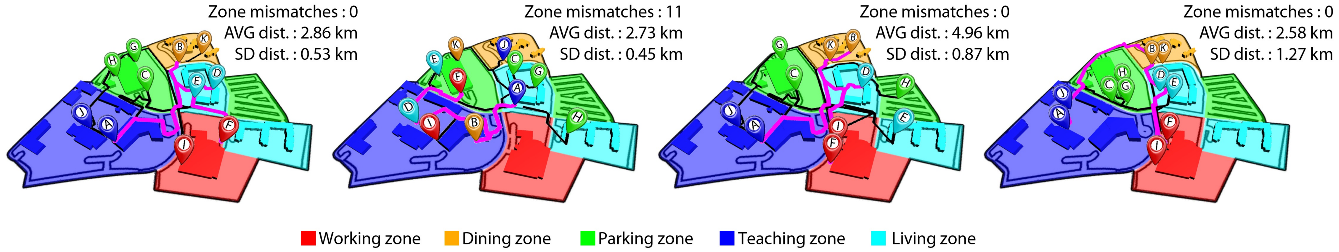
$$\Pr(S'|S) = \min(1, \frac{f(S')}{f(S)}), \tag{6}$$

where $f(S)$ is a Boltzmann-like objective function that follows a Metropolis-Hastings state searching step:

$$f(S) = \exp\left(-\frac{1}{t} C_{\text{total}}(\mathcal{F}(G, S, B))\right), \tag{7}$$

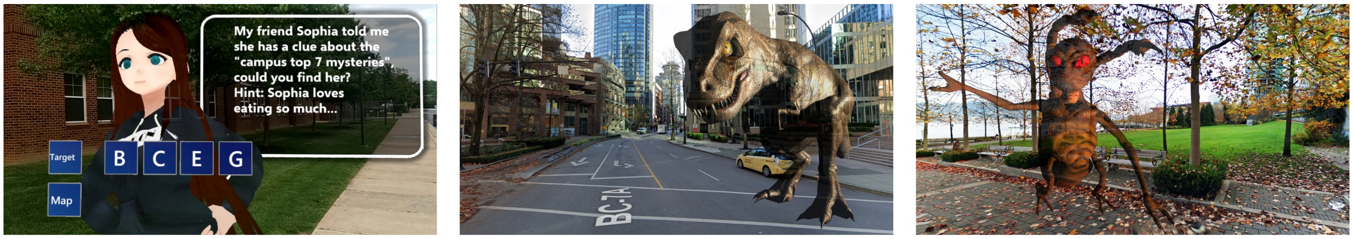
where t is the temperature of simulated annealing. As the temperature decreases gradually over iterations, the optimizer becomes more greedy. By the end, the temperature drops to a low value near zero so the optimizer accepts better solutions only. We empirically set temperature $t = 1.0$ at the beginning of the optimization and decrease it by 0.2 every 100 iterations until it reaches zero. The optimization is terminated if the total cost change is smaller than 3% over the past 50 iterations. Unless otherwise specified, we set the cost weights as $w_{\text{LC}} = 0.4$, $w_{\text{WD}} = 0.3$, and $w_{\text{REG}} = 0.3$ in our experiments. We apply a self-adaptation to perform local refinement near the end of the optimization when sampling the navigation graph. Please refer to the supplementary material for more details.

Figure 7(b-d) shows the effects of omitting costs when synthesizing navigation graphs, in contrast to the result in Figure 7(a) with all costs in place. Specifically, omitting any cost will decrease the location compatibility or walking experience. Note that we used Figure 7(a) in the main paper as the initialization to synthesize (b) to (d) for comparison and illustration. Figure 6 depicts the story



(a) All costs (b) No location compatibility cost (c) No walking distance cost (d) No regularization cost

Figure 7: Effects of omitting costs. (a) A navigation graph considering all costs; (b) Without the location compatibility cost, events are assigned to locations with mismatched zone types; (c) Without the walking distance cost, the average walking distance vastly increases; (d) Without the regularization cost, the standard deviation of walking distances is high.


My friend Sophia told me she has a clue about the "campus top 7 mysteries", could you find her? Hint: Sophia loves eating so much...

(a) *Detective AR* on a campus (b) *Dinosaur World AR* in a downtown (c) *Exterior Star War AR* in a park

Figure 8: Screenshots of AR contents populated in real-world places.

details. One story branch (A, B, D, F) is highlighted in purple to help comparison and illustration. Figure 7(b) shows a result synthesized without the location compatibility cost, where the events are not distributed to places that fit with the story context. Figure 7(c) shows a result synthesized without the walking distance cost, where the events are distributed sparsely with respect to each other, resulting in a longer overall walking distance. Figure 7(d) shows a result synthesized without the regularization cost, where the events are not evenly distributed spatially: some consecutive events (e.g., A & B) are very far away from each other while some (e.g., B & D) are close to each other.

5 EXPERIMENTS

In this section, we provide details of our experimental designs and results. Section 5.1 shows our main results of adapting AR narratives to different outdoor environments. Section 5.2 presents a learning-based zoning inference module, which can be substituted for predefined zoning maps. We also demonstrate the extensibility of our approach by incorporating a few additional design constraints in Section 5.3. Lastly, we test our approach's scalability to cope with larger stories with more events and branches in Section 5.4.

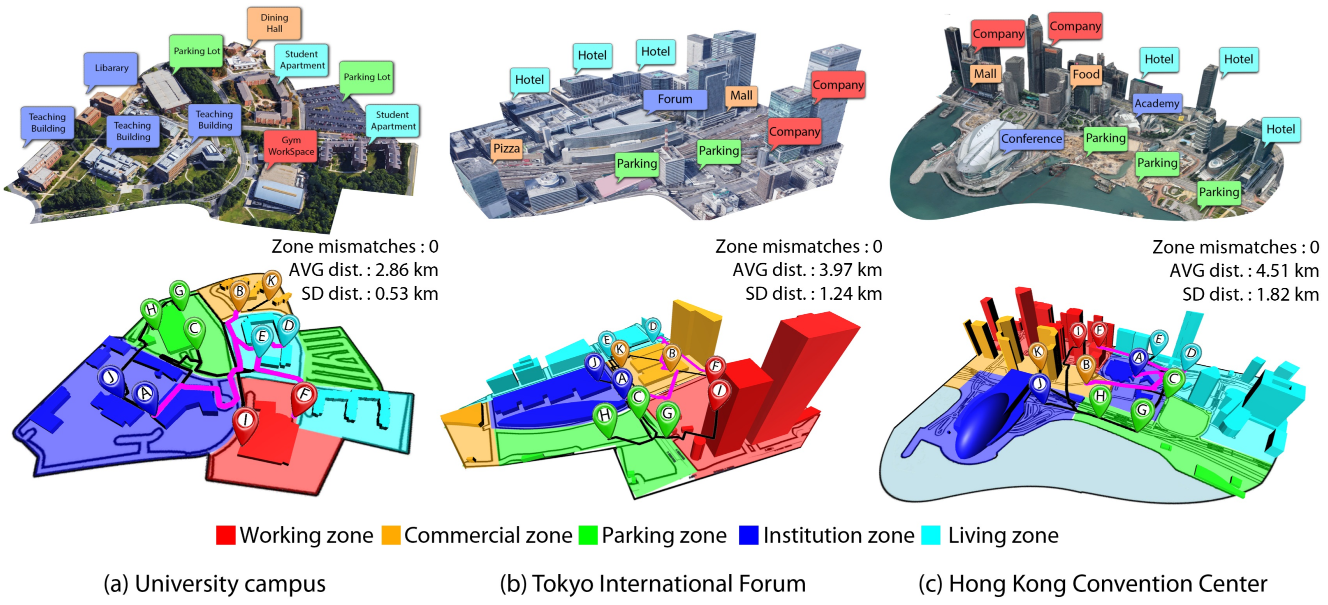
We implemented our approach on a machine with an Intel Core i7-9700 CPU, an NVIDIA GeForce RTX 2070 graphics card, and 32GB of RAM. The implementation was done in C# on the Unity3D game engine. It typically takes 1 to 2 minutes to generate all-pairs shortest paths depending on the density of the road network. A navigation graph is generally synthesized in about 500 to 1, 000 iterations. Optimizing a solution generally takes about 2 to 3 minutes in total based on our implementation.

We also developed an application that runs on the HoloLens 2 headset to allow players to experiment with the generated experiences in AR. Figure 8 shows some screenshots.

5.1 Adapting Augmented Reality Narratives

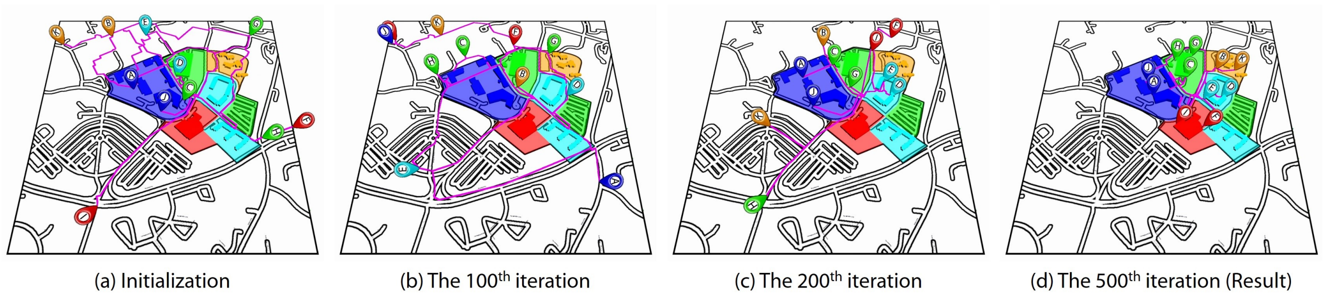
In this experiment, we focus on a *Detective AR* story whose structure is shown in Figure 6. The story consists of 11 events and 9 story branches, and is supposed to be experienced in a real-world environment comprising different types of zones. Our approach adapts this story by assigning real-world locations for the events to take place. The adaptation result in a campus scene is shown in Figure 9(a).

Our approach can automatically retarget the same story to different real-world places. For the same *Detective AR* story, we apply our approach to re-distribute the story to different city scenes. We show examples of story adaptation at the Tokyo International Forum (Figure 9(b)) and the Hong Kong Convention Center (Figure 9(c)). For each scene, the zone types are defined based on the functionalities of the buildings and regions. For example, a convention center neighborhood is designated as an institution (teaching) zone. Our approach distributes the story events compatibly with the zone types. For example, in the Tokyo International Forum, Event F (finding a character in a working zone) occurs near Gran Tokyo South Tower (in red) which is an office building. Events H, C, and G, which should happen in a parking zone, take place in the Kajibashi Parking Lot (in green). In the Hong Kong Convention Center result,



(a) University campus (b) Tokyo International Forum (c) Hong Kong Convention Center

Figure 9: Navigation graphs synthesized on different real-world maps.



(a) Initialization (b) The 100th iteration (c) The 200th iteration (d) The 500th iteration (Result)

Figure 10: Optimization process. The event locations are depicted by the pins whose colors refer to different zone types that the events are compatible with. (a) The sampled navigation graph is initialized randomly; (b-d) the sampled graphs over the iterations; (d) the optimized graph with the events distributed to locations of compatible zone types.

Event J (finding a character in an institution zone) happens near the convention center (in blue), while Event D (finding a character in a living zone) happens near the hotels (in cyan).

Figure 10 shows the optimization process on a road network graph $G = (V, E)$ which is converted from the satellite data of a university campus downloaded from the Google Map. At the beginning, random vertex indices on the road network graph G are sampled and assigned to the story events $S = \{s_i\}$. Then a navigation graph $G' = (V', E')$ is sampled from the road network graph G through the mapping reduction function $G' = \mathcal{F}(G, S, B)$. The optimization iteratively updates the location assignments and the navigation graph until convergence.

5.2 Autoencoder Zoning

We provide an alternative, learning-based approach to infer the probabilities that a location belongs to different zone types for the location compatibility score function in Equation 1. Prior arts have used computational methods to generate urban zones by using contextual information [17], self-training-based cluster ensemble strategy [2], or higher-order Markov random fields [15] on urban images. In this experiment, we employed an autoencoder [29] for inferring the probabilities that a location belongs to certain zone types. To illustrate this approach, we prepared a set of exemplar street-view images of three representative zone types including parks, roads and buildings. The autoencoder learned the environmental features of such zone types and produced the mean codes for each zone type by averaging the codes extracted from the exemplar images. The similarity between the code of input street-view

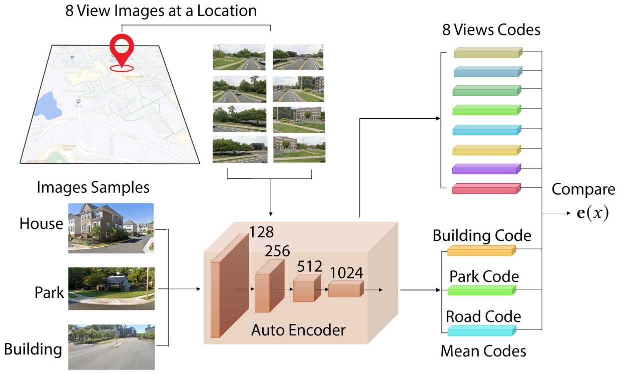


Figure 11: Environmental feature analysis using an autoencoder.

images at a certain location, and a zone type mean code, is used as the probability that the input location belongs to that zone type.

Figure 11 illustrates how to use the trained autoencoder/decoder framework to infer the zone types for a location. Given a location x on the map, our approach first obtains the street view images around that location using the Google Maps API. Then, it calculates the similarity $\psi(x, Z_i)$ between the visual feature of location x and the mean code of zone type Z_i by counting how many view images $I_k(x)$ at location x out of K different view directions (we set $K = 8$) are closest to the mean code c_i of zone type Z_i through this equation:

$$\psi(x, Z_i) = \frac{1}{K}\sum_{k=1}^{K}\delta\left(i == \operatorname*{argmin}_{j}||E(I_k(x)) - c_j||\right), \qquad (8)$$

where $E(I_k(x))$ is the code of image $I_k(x)$ computed by the autoencoder. $\delta()$ returns 1 if the input is true and returns 0 otherwise. The inferred similarity $\psi(x, Z_i)$ is taken as the probability that x belongs to the zone type Z_i, and is further used to compute the location compatibility cost for a given navigation graph.

In our experiments, we used 7, 440 Google street view images for training. We trained a 4-layered convolutional neural network as an encoder to convert 256×512 resolution RGB images to $2,048 \times 3$ -dimensional latent vectors corresponding to the codes of the images. We used 10, 000 batch iterations on the training images. We used a batch size of 64, an Adam optimizer, the mean square error as the loss function, and a learning rate of 0.001. After using these exemplar images to train the autoencoder, our approach computes the mean code c_i for a zone type Z_i by averaging all codes of exemplar images that belong to that zone type Z_i, extracted using the autoencoder.

We test the accuracy of our autoencoder using a cross-validation approach. As analyzed from the statistical tests, the accuracies of our autoencoder in predicting the building, park, and road zones are 76%, 87%, and 78%, respectively. More specifically, the true positive (TP), true negative (TN), false positive (FP), and false negative (FN) for the three zone types are: building (TP=0.763; TN=0.0057; FP=0.189; FN=0.041), park (TP=0.859; TN=0.012; FP=0.11; FN=0.017), and road (TP=0.785; TN=0.004; FP=0.172; FN=0.037). During the accuracy test, the threshold for our autoencoder classifier is set as 0.1, which means that if the probability of the region belonging to any zone is larger than 10%, then the region is classified as that zone. We iterate over all the sample points in the scene and statistically accumulate the errors that our autoencoder classifier made to get the cross-validation results.

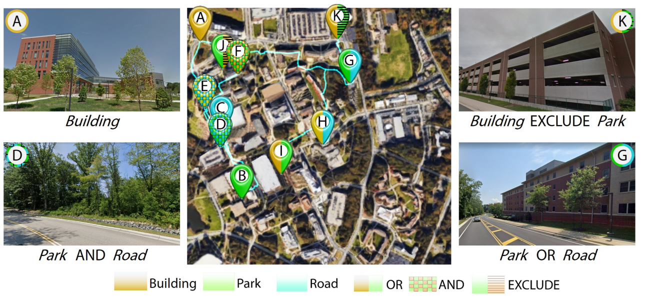


Figure 12: A navigation graph synthesized on a campus map using zoning maps inferred by an autoencoder. Three illustrative zone types (building, park, and road) are considered. *OR* (denotes $Z_1 \vee Z_2$), *AND* (denotes $Z_1 \wedge Z_2$) and *EXCLUDE* (denotes $Z_1 \wedge \overline{Z_2}$) in the legend are used for illustration convenience.

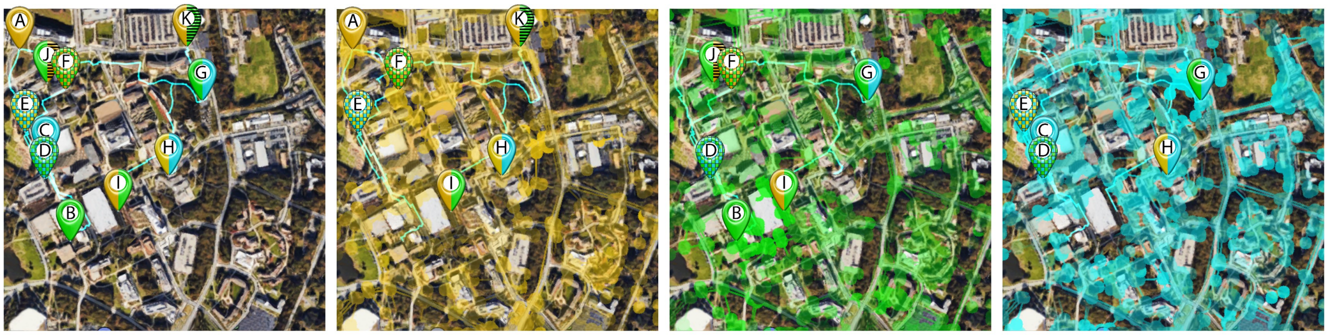
Figure 12 shows a navigation graph synthesized using the zoning maps inferred by the autoencoder. Example street view images at the assigned event locations suggest that the autoencoder captures the environmental features of different zone types. Our synthesized navigation graph using the inferred zoning maps is based on a modified version of the *Detective AR* story as depicted in Figure 3(b) in our supplementary material, in which we change the zone types and event location descriptions accordingly. Figure 13(a) shows the synthesis results including the distributed events and the navigation graph. Figure 13(b–d) shows the inferred zones with strong building (yellow), park (green), and road (blue) features, as well as the events in the navigation graph that are compatible with these zones.

5.3 Additional Constraints

We also extend our approach with additional constraints to handle the design requirements of some special scenarios. While we provide results of applying three additional constraints in this experiment to examine the extendability and flexibility of our approach, other self-defined constraints could be applied similarly for other purposes. For example, some constraints may enhance accessibility by picking only wheelchair-friendly roads for wheelchair users; and some other constraints may exclude regions with traffic (e.g., intersections, highways) for pedestrian safety.

Event Center. Designers using our approach may specify an event center around which the story events will be distributed. This consideration can be encoded as an additional cost:

$$C_{\text{center}}(G') = 1 - \exp\left(-\frac{1}{|V'|D_{\max}}\sum_{v'_i \in V'}||v'_i - L_c||\right), \qquad (9)$$



(a) All events (b) Events compatible with buildings (c) Events compatible with parks (d) Events compatible with roads

Figure 13: Navigation graphs synthesized on a campus using zones inferred by the autoencoder. (a) All events and their locations. (b-d) Events compatible with buildings, parks, and roads are shown, respectively. The color intensities correspond to how likely the locations belong to the respective zone types.

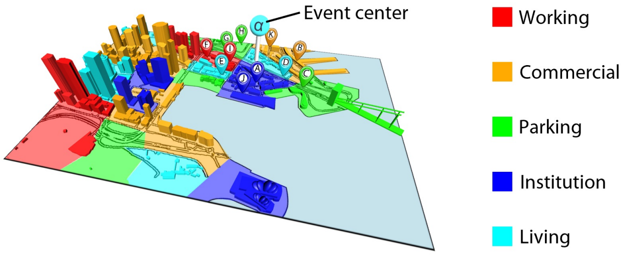


Figure 14: An event center specified at α. The story events are distributed around the event center by our optimizer.

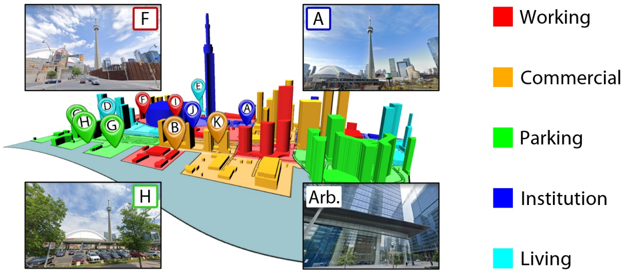


Figure 15: Considering the visibility of a landmark (a tower). The tower is visible from the locations of events A, F, and H, while it is not visible from an arbitrary location.

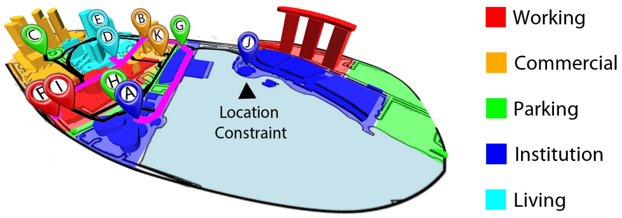


Figure 16: A location constraint is specified for event J. The optimizer distributes the remaining events while maintaining this constraint.

where L_c refers to the location of the event center on the map. This cost term penalizes the distance between the event center's location L_c and each story event s_i's assigned location v'_i in the navigation graph. $D_{\max}$ is a normalization term as defined in the walking distance cost. Figure 14 shows an example of setting an event center at α around which our optimizer synthesizes a navigation graph and distributes events.

Landmark Visibility. Our approach can be extended to consider a landmark's visibility which may facilitate wayfinding. This consideration can be encoded as an additional cost term:

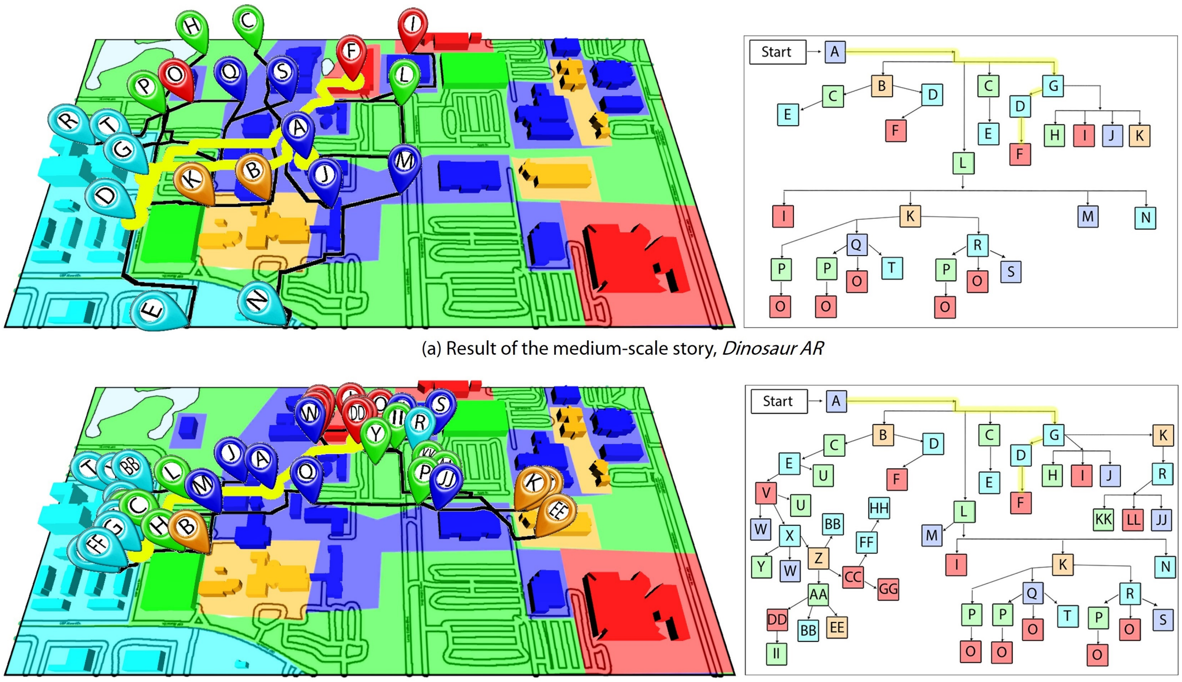
$$C_{\text{visibility}}(G') = 1 - \exp\left(-\frac{1}{|V'|}\sum_{\forall v'_i \in V'} \xi(v'_i)\right), \qquad (10)$$

where the function $\xi(v'_i)$ returns 1 if the landmark is not visible from location v'_i as determined by a ray casting in 3D, otherwise it returns 0. The visibility evaluation is done from the location v'_i of each story event s_i. Our approach favors locations from which the landmark can be seen when distributing events. Figure 15 depicts a synthesized navigation graph that considers the CN Tower's visibility from the event locations in Toronto.

Location Constraint. Our approach also enables designers to incorporate hard location constraints to fix the assignment of events to certain locations. It then further distributes the remaining events of the story with respect to those location constraints. Figure 16 shows an example based on a region near Singapore's Marina Bay Sands. To encourage players to walk around the scene, the designer manually assigns event J (finding Isabella in a teaching zone) to happen near the institution buildings (in blue) on the right. Our optimizer distributes the remaining story events with respect to this hard location constraint.

5.4 Scalability Test

To validate the scalability of our approach, we tested our approach with two additional stories of larger scales than the *Detective AR*



(a) Result of the medium-scale story, *Dinosaur AR*

(b) Result of the large-scale story, *Exterior Star War AR*

Figure 17: Results synthesized by our approach on two larger-scale stories.

story. Based on the statistics that our user study participants took about 45 minutes on average to complete one branch of the *Detective AR* story (comprising 11 events and 9 story branches), we designed the following larger-scale stories for feasible AR experiences:

(1) *Dinosaur World AR* (refer to Figure 7 in our supplementary material). In this story, there are 20 events and 30 story branches in total. The story is about a player who has found a strange huge egg and realizes that it is a dinosaur egg, leading to the beginning of a fascinating journey.
(2) *Exterior Star War AR* (refer to Figure 8 in our supplementary material). In this story, there are 38 events and 45 story branches in total. The story is about a player who received a message from a stranger who claims to be a time traveler from 2048. Then a fascinating journey begins near a teaching building.

Figure 17 shows results synthesized with the default parameter settings. In order to demonstrate the story branches more clearly, we plot the abstract story tree using boxes tagged with the event symbols. A navigation route ((A, B, D, E) in our case) is highlighted as yellow both in the synthesized result on the map and in the abstract story tree to facilitate comparison. It took about $4-5$ minutes to synthesize the medium-scale result and $5-6$ minutes to synthesize the large-scale result. Their final costs were 0.037 and 0.063, respectively. The results show that our approach can produce optimized navigation graphs of different scales efficaciously.

6 USER STUDY

We conducted two user studies to evaluate the effectiveness of our approach from different perspectives. In the first study, we evaluated players' experiences regarding event-location compatibility. In the second study, we evaluated our approach's efficiency in distributing events in comparison to manual designs. All participants were recruited through advertising the user studies in our university. Our user study experiments were approved by the university's institutional review boards.

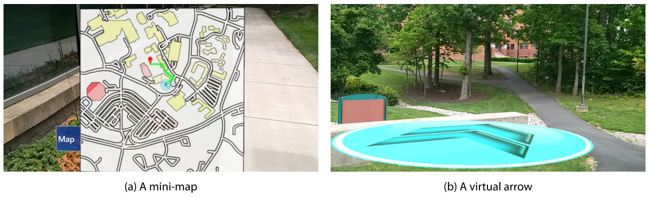
6.1 Experiencing Augmented Reality Narratives

6.1.1 Evaluation Goal. A key objective of our approach is to assign story events to compatible locations that fit the contexts of the story plots. Therefore we conducted an AR user study to validate the effectiveness of the location compatibility cost, which is our core consideration.

6.1.2 Evaluation Details. We recruited 20 university students, including 4 females and 16 males aged 19 through 28, and examined if they could notice such location compatibility brought about by this cost term. The study was based on the same *Detective AR* story shown in Figure 6 and was conducted on a university campus as depicted in Figure 7. Most of the participants were familiar with the campus and knew about the functionalities of the buildings. Each participant wore a HoloLens 2 headset to go through two conditions of the story:

(1) *All Costs:* the navigation graph was synthesized and the events were distributed with all costs considered.
(2) *No Location Compatibility Cost:* the navigation graph was synthesized and the events were distributed with the location compatibility cost omitted. In other words, the events' location compatibility was not considered.

In both conditions, the players were guided to navigate our sampled story event locations based on the story plot. During the



(a) A mini-map

(b) A virtual arrow

Figure 18: Wayfinding aids for facilitating navigation. The users navigated to our sampled event locations guided by two types of hints including (a) a mini-map; and (b) a virtual arrow. The hints were shown in the AR interface during the user's navigation along the optimized route.

storytelling, when any virtual character appeared, a dialogue box popped up to show the character's script, which was also played in audio. If needed, the players could trigger two wayfinding aids: a mini-map and a virtual arrow. So the players could follow the plot of the story closely. The players were asked to rate the location compatibility of events after completing a condition of the story.

We implemented two wayfinding aids on the HoloLens2 AR headset to help players navigate: (1) a mini-map, which draws a navigation path on a 2D flat map; (2) a virtual arrow: which visualizes an arrow showing the direction to the next event in 3D space. Refer to Figure 18 for visualizations of the wayfinding aids seen in AR. The players can trigger both wayfinding aids as desired.

Before going through the story, each player was allowed to interact with the AR setup and user interface in a warm-up session. If a player was unfamiliar with the campus, the helper would briefly explain the campus layout and building functionalities. Then the player would go through the two conditions of the story. For counterbalancing, half of the players went through the *All Costs* condition followed by the *No Location Compatibility Cost* condition, and vice versa. Besides using the wayfinding aids, the players could also ask the helper for help in case they had troubles finding the destinations or got lost during the navigation.

After going through each condition, the player would recall what story events they encountered and where those events happened by talking to the helper. Then the player would be asked to rate whether the locations are compatible with the contexts of the story events using a 5-point Likert scale (1: the lowest compatibility; 5: the highest compatibility).

6.1.3 Results. Figure 19 shows the user ratings. The *All Costs* condition has a mean score of 4.55 (median=5, SD=0.75), while the *No Location Compatibility Cost* condition has a mean score of 1.85 (median=2, SD=0.87). We conducted a paired t-test on the scores of the two conditions. A significant difference ($p < 0.001$) was found between the scores of the two conditions under the 99% confidence interval, suggesting that the participants thought that the story events were more compatible with their locations under the *All Costs* condition in comparison. In addition, we also collected users' ratings on the question "which location-assignment condition leads to a more reasonable storytelling?". All of the users rated that the events with locations assigned by our approach are more reasonable in comparison.

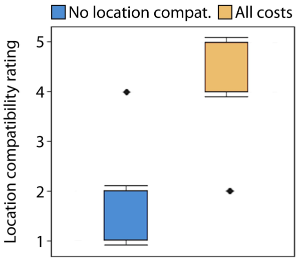


Figure 19: User ratings.

We collected general feedback from the participants about the storytelling experience in AR. Most participants found our AR storytelling interesting. Some participants who had never used AR devices before believed that our application was more exciting than traditional desktop adventure games. Some participants suggested that our application could be realized as a tour guide for new students coming to campus or as an exercise application. Most participants said that the walking distance of the story was not too long. Some participants had unpleasant experiences during the storytelling. For example, some found the user interface in AR a bit unresponsive and hard to control. Some participants missed the virtual characters that had already appeared in front of them so they needed to walk back to find the characters again.

6.2 Comparison with Manual Design

6.2.1 Evaluation Goal. The goal was to investigate whether our walking distance and regularization considerations can help designers assign events' locations on a real map more efficiently, assuming the same design goals of ensuring location compatibility, short walking distances, and even walking distance distribution.

6.2.2 Evaluation Details. We recruited 15 designers, including 5 females and 15 males aged 21 through 30, for another user study. For comparison, we asked them to manually assign event locations for the medium-scale story, *Dinosaur World AR* (details described in our supplementary material), with 20 events and 30 story branches, on a university campus map. We collected metrics and feedback from their manual design processes.

All the designers were experienced in using the Unity game engine for creating computer games or simulation applications so they were familiar with the operations of Unity. We developed an interactive user interface as an Unity plugin (shown in Figure 20) that enabled designers to put event pins on a map via simple drag-and-drop operations. The designer can see the story tree containing all the story branches. The designer can also press a hotkey to trigger an automatic analysis of their current design, which would show the average and standard deviation of the distances of the story branches. The designer can also see if there is any mismatch between the story events and their locations in terms of the zone types as well through pin and zone colors.

Before starting the design task, each designer got familiar with our plugin interface in a warm-up session, where the helper explained its functionality. Then the designer was tasked with assigning event locations for the story, considering the same major design objectives as our approach: (a) assigning events to their compatible zones (i.e. with a matching color); (b) minimizing the walking distances of all story branches; and (c) minimizing the standard deviation of the walking distances of the story branches. A designer finished the design when he/she thought that the design could not be improved further with respect to these objectives.

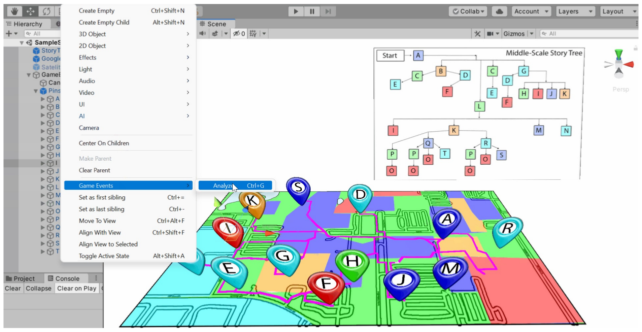
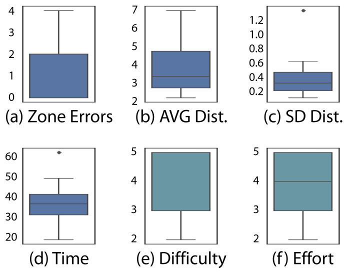


Figure 20: Manual design user interface.

6.2.3 Results. Figure 21 shows the metrics collected from the manual design tasks. The following summarizes the metrics: (a) Number of zone mismatches (mean=0.93; SD=1.33; median=0); (b) Average walking distance (in km) (mean=3.81; SD=1.60; median=3.33); (c) SD of walking distance (in km) (mean=0.39; SD=0.29; median=0.30); (d) Completion time (in min) (mean=37.86; SD=10.37; median=37); and (e) Number of distance analysis triggered (mean=48.73; SD=17.92; median=42). Comparing the manual designs with our *Dinosaur World AR* synthesis result, our synthesis result has a similar number of zone mismatches, but a generally shorter average walking distance of 2.98km and a similar SD in walking distance of 0.40km. Our result was synthesized automatically in about 5 minutes, while on average it took a designer 37.86 minutes to create a design manually with 48.73 distance analyses triggered.



(a) Zone Errors (b) AVG Dist. (c) SD Dist.

(d) Time (e) Difficulty (f) Effort

Figure 21: Data collected from the designers. The distances are in km and the time is in minutes.

The designers also rated the manual design task difficulty using a 5-point Likert scale (1: very easy; 5: very hard) and the task effort (1: not effort-demanding; 5: very effort-demanding). Figure 21(e-f) shows the ratings of difficulty (mean=3.60; SD=1.12; median=3) and effort (mean=3.80; SD=1.14; median=4). Overall, the designers rated the manual design task as slightly difficult and demanding.

We also asked the designers for their general feedback. Many designers thought that while it was easy to accommodate individual events at compatible locations, it was difficult to find overall optimal short paths for the story branches while ensuring the location compatibility of all events. Please refer to the supplementary material for detailed records of their feedback.

7 LIMITATIONS AND FUTURE WORK

We propose an automatic approach for adapting interactive narratives to real-world places for AR experiences. Our optimization-based approach considers the compatibility of the story events with their assigned locations, and the players' walking experiences in going through the story. By sampling a navigation graph from a real-world map, our approach can be integrated into an AR storytelling application to guide a player to experience different story branches in the real world. In our experiments, we adapted AR stories to different real-world places. We also conducted user studies to validate the effectiveness of our cost terms and tested the extensibility of our approach by incorporating additional constraints. We will release the code of our approach to ease future extensions.

Our approach assumes that the input map contains all the zone types needed for storytelling. This assumption may not hold in reality. For instance, a rural area may not have locations matching the parking zone type. Similarly, we assume fixed zone types, which may vary in practice (e.g., a bustling eating area might only be active in the evening). It would be powerful to dynamically analyze the compatibility of real-world places for experiencing an AR story.

We define a walking distance consideration for constraining the lengths of walking trajectories on navigation graphs. While users may personalize the target lengths by adjusting the cost term's parameters, it is worth investigating the relationships between walking and user experiences in future work. For example, future work may investigate how to avoid the potential feeling of discontinuity during the travel from one position to another, and how the ability to get to a location may affect the storytelling experience.

Our approach could be extended to consider low-level scene features by using a learning-based method. In subsection 5.2, we used an autoencoder to produce zoning maps with customized zoning type labels (e.g., building, park). However, due to the limitations of current computer vision techniques (e.g., segmentation, reconstruction, registration) for outdoor AR, we cannot dynamically adapt virtual characters to interact with scene objects on the fly (e.g., playing a swing). Future extensions may analyze and utilize lower-level scene semantics in real-time with computer vision advancements.

Our current implementation does not consider potential concerns regarding safety, privacy and responsibility. A possible solution to mitigate such concerns using our approach is to filter out locations (e.g., roads with traffic) where such issues may exist when sampling on the map. In future extensions, we want to adopt recommendations provided by Cardenas et al. [6] to avoid risks of depicting sensitive narratives through immersive technologies.

Our current approach only supports stories with fixed events and branches even though players may experience different story branches based on their choices. Future work may introduce wearable sensors to track players' metrics (e.g., gazes, body poses) for triggering narratives. We are also interested in integrating multiple random initializations with multi-thread acceleration to expedite navigation graph syntheses.

ACKNOWLEDGMENTS

We are grateful to the anonymous reviewers for their constructive comments. This project was supported by NSF grants (award numbers: 1942531 and 2128867).

REFERENCES

[1] 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.

[2] Biplab Banerjee, Francesca Bovolo, Avik Bhattacharya, Lorenzo Bruzzone, Subhasis Chaudhuri, and B Krishna Mohan. 2014. A new self-training-based unsupervised satellite image classification technique using cluster ensemble strategy.

IEEE Geoscience and Remote Sensing Letters 12, 4 (2014), 741–745.

[3] Arpita Bhattacharya, Travis W Windleharth, Rio Anthony Ishii, Ivy M Acevedo, Cecilia R Aragon, Julie A Kientz, Jason C Yip, and Jin Ha Lee. 2019. Group interactions in location-based gaming: A case study of raiding in pokémon go. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.

[4] Anne E Bowser, Derek L Hansen, Jocelyn Raphael, Matthew Reid, Ryan J Gamett, Yurong R He, Dana Rotman, and Jenny J Preece. 2013. Prototyping in PLACE: a scalable approach to developing location-based apps and games. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1519–1528.

[5] Manuel Braunschweiler, Steven Poulakos, Mubbasir Kapadia, and Robert W Sumner. 2018. A Two-Level Planning Framework for Mixed Reality Interactive Narratives with User Engagement. In *2018 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*. IEEE, 100–107.

[6] Ana María Cárdenas Gasca, Jennifer Mary Jacobs, Andrés Monroy-Hernández, and Michael Nebeling. 2022. AR Exhibitions for Sensitive Narratives: Designing an Immersive Exhibition for the Museum of Memory in Colombia. In *Designing Interactive Systems Conference*. 1698–1714.

[7] Marc Cavazza, Fred Charles, and Steven J Mead. 2002. Planning characters' behaviour in interactive storytelling. *The Journal of Visualization and Computer Animation* 13, 2 (2002), 121–131.

[8] 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).

[9] 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.

[10] 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.

[11] Siddhartha Chib and Edward Greenberg. 1995. Understanding the metropolis-hastings algorithm. *The american statistician* 49, 4 (1995), 327–335.

[12] John Joon Young Chung, Wooseok Kim, Kang Min Yoo, Hwaran Lee, Eytan Adar, and Minsuk Chang. 2022. TaleBrush: Sketching Stories with Generative Pretrained Language Models. In *CHI Conference on Human Factors in Computing Systems*. 1–19.

[13] Ignacio X Domínguez, Rogelio E Cardona-Rivera, James K Vance, and David L Roberts. 2016. The mimesis effect: The effect of roles on player choice in interactive narrative role-playing games. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 3438–3449.

[14] Magy Seif El-Nasr. 2007. Interaction, narrative, and drama: Creating an adaptive interactive narrative using performance arts theories. *Interaction Studies* 8, 2 (2007), 209–240.

[15] Tian Feng, Quang-Trung Truong, Duc Thanh Nguyen, Jing Yu Koh, Lap-Fai Yu, Alexander Binder, and Sai-Kit Yeung. 2018. Urban zoning using higher-order markov random fields on multi-view imagery data. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 614–630.

[16] Robert W Floyd. 1962. Algorithm 97: shortest path. *Commun. ACM* 5, 6 (1962), 345.

[17] Björn Fröhlich, Eric Bach, Irene Walde, Sören Hese, Christiane Schmullius, and Joachim Denzler. 2013. Land cover classification of satellite images using contextual information. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 3, W1 (2013), 1–6.

[18] 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.

[19] Charles J Geyer. 1992. Practical markov chain monte carlo. *Statistical science* 7, 4 (1992), 473–483.

[20] 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.

[21] 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.

[22] Anton Gustafsson, John Bichard, Liselott Brunnberg, Oskar Juhlin, and Marco Combetto. 2006. Believable environments: generating interactive storytelling in vast location-based pervasive games. In *Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology*. 24–es.

[23] Matthew Guzdial, Brent Harrison, Boyang Li, and Mark Riedl. 2015. Crowdsourcing Open Interactive Narrative.. In *FDG*.

[24] Sehoon Ha, Jim McCann, C Karen Liu, and Jovan Popović. 2013. Physics storyboards. In *Computer Graphics Forum*, Vol. 32. Wiley Online Library, 133–142.

[25] 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*.

[26] Mubbasir Kapadia, Jessica Falk, Fabio Zünd, Marcel Marti, Robert W Sumner, and Markus Gross. 2015. Computer-assisted authoring of interactive narratives. In *Proceedings of the 19th Symposium on Interactive 3D Graphics and Games*. 85–92.

[27] Mubbasir Kapadia, Seth Frey, Alexander Shoulson, Robert W Sumner, and Markus H Gross. 2016. CANVAS: computer-assisted narrative animation synthesis.. In *Symposium on Computer Animation*. 199–209.

[28] Bilal Kartal, John Koenig, and Stephen J Guy. 2014. User-driven narrative variation in large story domains using monte carlo tree search. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*. Citeseer, 69–76.

[29] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1312.6114

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

[31] Yining Lang, Wei Liang, and Lap-Fai Yu. 2019. Virtual Agent Positioning Driven by Scene Semantics in Mixed Reality. In *IEEE Virtual Reality*.

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

[33] Changyang Li, Wanwan Li, Haikun Huang, and Lap-Fai Yu. 2022. Interactive augmented reality storytelling guided by scene semantics. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–15.

[34] Wei Liang, Xinzhe Yu, Rawan Alghofaili, Yining Lang, and Lap-Fai Yu. 2021. 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.

[35] Mei Yii Lim and Ruth Aylett. 2007. Narrative construction in a mobile tour guide. In *International Conference on Virtual Storytelling*. Springer, 51–62.

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

[37] Andrew Macvean, Sanjeet Hajarnis, Brandon Headrick, Aziel Ferguson, Chinmay Barve, Devika Karnik, and Mark O Riedl. 2011. WeQuest: scalable alternate reality games through end-user content authoring. In *Proceedings of the 8th international conference on advances in computer entertainment technology*. 1–8.

[38] Brian Magerko and John E Laird. 2004. Mediating the tension between plot and interaction. In *AAAI Workshop Series: Challenges in Game Artificial Intelligence*, Vol. 1. 4.

[39] Stacey Mason, Ceri Stagg, and Noah Wardrip-Fruin. 2019. Lume: a system for procedural story generation. In *Proceedings of the 14th International Conference on the Foundations of Digital Games*. 1–9.

[40] Joshua McCoy, Mike Treanor, Ben Samuel, Noah Wardrip-Fruin, and Michael Mateas. 2011. Comme il faut: A system for authoring playable social models. In *Seventh Artificial Intelligence and Interactive Digital Entertainment Conference*.

[41] 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.

[42] TeongJoo Ong and John J Leggett. 2004. A genetic algorithm approach to interactive narrative generation. In *Proceedings of the fifteenth ACM conference on Hypertext and hypermedia*. 181–182.

[43] Steven Poulakos, Mubbasir Kapadia, Guido M Maiga, Fabio Zünd, Markus Gross, and Robert W Sumner. 2016. Evaluating accessible graphical interfaces for building story worlds. In *International Conference on Interactive Digital Storytelling*. Springer, 184–196.

[44] Alejandro Ramirez and Vadim Bulitko. 2014. Automated planning and player modeling for interactive storytelling. *IEEE Transactions on Computational Intelligence and AI in Games* 7, 4 (2014), 375–386.

[45] Mark Riedl, Boyang Li, Hua Ai, and Ashwin Ram. 2011. Robust and authorable multiplayer storytelling experiences. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, Vol. 6.

[46] Mark Owen Riedl and Vadim Bulitko. 2013. Interactive narrative: An intelligent systems approach. *Ai Magazine* 34, 1 (2013), 67–67.

[47] Mark O Riedl and Robert Michael Young. 2006. From linear story generation to branching story graphs. *IEEE Computer Graphics and Applications* 26, 3 (2006), 23–31.

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

[49] Alexander Shoulson, Max L Gilbert, Mubbasir Kapadia, and Norman I Badler. 2013. An event-centric planning approach for dynamic real-time narrative. In *Proceedings of Motion on Games*. 121–130.

[50] Oliviero Stock, Massimo Zancanaro, Paolo Busetta, Charles Callaway, Antonio Krüger, Michael Kruppa, Tsvi Kuflik, Elena Not, and Cesare Rocchi. 2007. Adaptive, intelligent presentation of information for the museum visitor in PEACH. *User Modeling and User-Adapted Interaction* 17, 3 (2007), 257–304.

[51] 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.

[52] Pengcheng Wang, Jonathan P Rowe, Wookhee Min, Bradford W Mott, and James C Lester. 2017. Interactive Narrative Personalization with Deep Reinforcement Learning.. In *IJCAI*. 3852–3858.

[53] Pengcheng Wang, Jonathan P Rowe, Wookhee Min, Bradford W Mott, and James C Lester. 2018. High-Fidelity Simulated Players for Interactive Narrative Planning.. In *IJCAI*. 3884–3890.

[54] 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 *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 328–341.

[55] 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.

[56] Hong Yu and Mark O Riedl. 2012. A sequential recommendation approach for interactive personalized story generation.. In *AAMAS*, Vol. 12. 71–78.

[57] Xun Zhang, Bhuvana C Inampudi, Norman I Badler, and Mubbasir Kapadia. 2017. Dynamic and Accelerated Partial Order Planning for Interactive Narratives. In *Thirteenth Artificial Intelligence and Interactive Digital Entertainment Conference*.

[58] Zhenpeng Zhao, Sriram Karthik Badam, Senthil Chandrasegaran, Deok Gun Park, Niklas LE Elmqvist, Lorraine Kisselburgh, and Karthik Ramani. 2014. skWiki: a multimedia sketching system for collaborative creativity. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1235–1244.

[59] 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.