Scene-Aware Behavior Synthesis for Virtual Pets in Mixed Reality

Wei Liang
Beijing Institute of Technology
Beijing, China

Xinzhe Yu
Beijing Institute of Technology
Beijing, China

Rawan Alghofaili
George Mason University
Fairfax, Virginia, USA

Yining Lang
Alibaba Group
Beijing, China

Lap-Fai Yu
George Mason University
Fairfax, Virginia, USA

Figure 1: Left: Through a Hololens helmet, a user is observing a virtual kitten whose behavior is synthesized by our approach. Right: We demonstrate a synthesized behavior sequence for the virtual kitten considering the geometry and semantics of the real scene. The virtual kitten rests on the couch for a while ①. Then it moves to the food bowl and starts to eat ②. Afterward, it jumps up to the coffee table to idle ③.

ABSTRACT

Virtual pets are an alternative to real pets, providing a substitute for people with allergies or preparing people for adopting a real pet. Recent advancements in mixed reality pave the way for virtual pets to provide a more natural and seamless experience for users. However, one key challenge is embedding environmental awareness into the virtual pet (*e.g.*, identifying the food bowl's location) so that they can behave naturally in the real world.

We propose a novel approach to synthesize virtual pet behaviors by considering scene semantics, enabling a virtual pet to behave naturally in mixed reality. Given a scene captured from the real world, our approach synthesizes a sequence of pet behaviors (*e.g.*, resting after eating). Then, we assign each behavior in the sequence to a location in the real scene. We conducted user studies to evaluate our approach, which showed the efficacy of our approach in synthesizing natural virtual pet behaviors.

CHI '21, May 8–13, 2021, Yokohama, Japan

ACM ISBN 978-1-4503-8096-6/21/05…$15.00
https://doi.org/10.1145/3411764.3445532

CCS CONCEPTS

• **Human-centered computing** → **Human computer interaction (HCI)**; *Mixed reality*.

KEYWORDS

Virtual Pets, Behavior Synthesis, Scene Semantics

ACM Reference Format:
Wei Liang, Xinzhe Yu, Rawan Alghofaili, Yining Lang, and Lap-Fai Yu. 2021. Scene-Aware Behavior Synthesis for Virtual Pets in Mixed Reality. In *CHI Conference on Human Factors in Computing Systems (CHI '21), May 8–13, 2021, Yokohama, Japan.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3411764.3445532

1 INTRODUCTION

Pets enrich our lives in many ways. Pets help people live healthier lives [30] [11]; alleviate loneliness [21]; and assist with therapy [2]. However, not everyone's situation allows for adopting a pet. For example, people may live in apartments which may not be pet-friendly. A person may not be able to keep a pet due to allergies. In such situations, virtual pets can be a good alternative to a real pet.

Virtual pet applications date back to 1995 when Dogz was released. It allowed users to adopt, raise, and breed virtual dogs. After that, many virtual pet applications were developed, *e.g.*, Tamagotchi, Digimon, Giga Pets, Nintendogs. The experience of keeping virtual pets resembles a real pet-keeping experience. People took care of

a virtual pet as if it were real, *e.g.*, by feeding it and bathing it. Compared to real pets, virtual pets take up no physical space and are convenient to care for. On the other hand, owners of virtual pets do not need to worry about allergies and costs.

Researchers of many fields have shown increasing interest in virtual pets. They explore along different directions. Some researchers in robotics domain study the implementation of an autonomous pet robot, *e.g.*, perception and kinematics [31, 56]. Some psychologists investigate the relations between human and virtual animals, *e.g.*, the therapeutic effects in assisted therapy [30, 33]. Recently, the arrival of mixed reality devices, *e.g.*, Microsoft Hololens, provides opportunities to introduce new forms of virtual pets that support high-quality immersive experiences and natural interactions in mixed reality. One of the core problems for those researches is creating autonomous and realistic behaviors for virtual pets to enable natural interaction. There are two critical challenges to solve such a behavior synthesis problem:

(1) How can we synthesize virtual pet behaviors akin to real pet behaviors? Some traditional works defined pet behaviors using fixed rules. For example, a virtual kitten always sleeps after eating. However, those hard-coded or randomly-generated behaviors may appear unrealistic, which hardly resemble real pet behaviors, resulting in unnatural user experiences.

(2) How can we enable virtual pets to behave rationally in a real scene? The key feature of mixed reality—fusing the virtual world with the real world—hints that it is foundational for the virtual pets to understand scene information. In existing applications, *e.g.*, HoloPet [41], a virtual pet is placed in front of the user regardless of the real scene context, which may lead the virtual pets floating in the air. Another common way is to place the virtual pet on a surface specified by the user. Consequently, the virtual pet is restricted to act within the specified zone, hindering its flexibility and variation.

To address the above two challenges, in this paper, we propose a scene-aware behavior synthesis approach for virtual pets, aiming at generating autonomous and realistic virtual pet behaviors to provide highly immersive user experiences. To synthesize natural behaviors, our approach trains a pet behavior generator based on real pets data [54] using a Long Short-Term Memory (LSTM) network. Applying this behavior generator, we can generate high-level pet behavior sequences automatically, *e.g.*, eating after idling. Then we want a virtual pet to perform the generated behaviors in a real scene rationally. We leverage computer vision techniques to enable the virtual pet to understand the semantics of the scene, *e.g.*, identifying the location of a couch. Then the generated behaviors are instantiated at the corresponding locations, *e.g.*, performing the idling behavior on the couch. In addition, our approach optimizes a feasible path for the virtual pet to travel between two adjacent locations using an adjusted A* algorithm, *e.g.*, a path to travel from the food bowl (where a pet is performing eating behavior) to the couch (where a pet is going to perform idling behavior).

The major contributions of our paper include:

- Propose to synthesize virtual pet behaviors based on the geometry and semantics of a real scene.
- Devise a high-level pet behavior generator via training with real pet data, and instantiate the synthesized pet behaviors in a real scene.
- Validate the effectiveness of our approach by user studies.

2 RELATED WORK

Emerging technologies such as mixed reality have reshaped users' expectations for their experiences with virtual pets. Highly-immersive mixed reality environments bring new opportunities and technical challenges to virtual pet applications. One important challenge is to enable virtual pets to behave in the real world more naturally. In this section, we briefly review virtual pet research and applications, as well as behavior synthesis and scene semantics understanding.

2.1 Virtual Pet Research

Researchers in human-computer interaction, virtual reality, psychology, and robotics have shown substantial interest in techniques for creating virtual pets [51] motivated by a variety of potential applications such as entertainment [1, 14], education [7, 36] and therapeutic domain [30, 33]. We also drew inspiration from some previous works [35, 47] while defining our evaluation metrics.

Lawson and Chesney [37] hypothesized that younger children might benefit more from companionship with virtual pets than adults. There are some discussions in the literature about the different experiences of owning virtual pets and real pets [8]. Beetz et al. [5] and Wang et al. [62] discussed the influence of virtual pets on humans. Norouzi et al. [44] investigated the effects of virtual dogs in AR environment on participants' perception and behaviors, including locomotion related to proxemics, with respect to their AR dog and other real people in the environment.

An important prerequisite for the above applications is that virtual pets can behave realistically and reasonably. But the virtual pet behavior in most applications is generally designed manually based on rules or is generated randomly, lacking varieties or reasonability. The observations motivate us to design a framework to synthesize virtual pet behaviors by learning from a real pet dataset. In addition, the synthesized virtual pets perform behaviors at appropriate scene locations based on the understanding of scene semantics, which facilitates the virtual pets applications in mixed reality.

2.2 Behavior Synthesis

Behavior synthesis aims at creating models automatically synthesizing behaviors. Many works have been proposed to tackle behavior and motion synthesis for virtual characters [28, 34, 48] and robots [12].

There are two main categories of behavior synthesis approaches: data-driven approaches and model-based approaches. The data-driven approaches use machine learning techniques to learn behavior models from collected behavior datasets; in other words, using supervised approaches to synthesize behaviors. The machine learning techniques applied vary from simple naive Bayes classifiers [4], hidden Markov models [60], and dynamic Bayesian networks [45], to support vector machines [19] and incremental classifiers [46].

With the development of deep learning techniques, Recurrent Neural Networks (RNN) received substantial attention from researchers because of its ability to model sequential data. The variants including Long Short Term Memory (LSTM) [27] and Gated Recurrent Units [9] have proven to be very successful for sequence

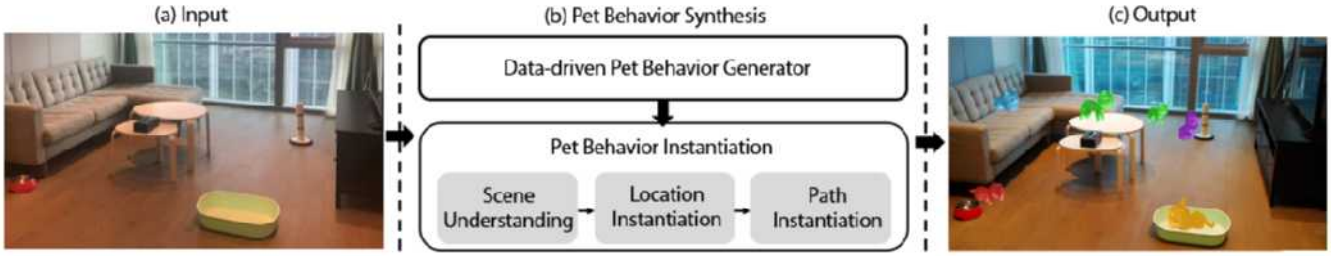


Figure 2: The overview of our approach. (a) The input is a real scene. (b) The pet behavior synthesis includes two components: data-driven pet behavior generator and pet behavior instantiation. First, the pet behavior generator synthesizes a sequence of behaviors. Then in the instantiation component, a location sequence is generated for the virtual pet to perform the behaviors in the real scene. (c) The output shows some examples of the generated pet behaviors, with the color of red, blue, green, purple, and yellow, depicting the behavior of eating, resting, idling, scratching and soiling, respectively.

generation tasks, such as text generation [58] , handwriting prediction [23], and image caption generation [32].

One application direction of behavior synthesis techniques is robot behavior synthesis, aiming at enhancing and enriching interactive experiences with users. Jung et al. [31] and Song et al. [56] use emotional models to create behaviors for robots. Desai et al. [12] propose a simulation-driven robot motion design system that enables the design of expressive behaviors using high-level and semantic descriptions of behavior properties.

Another application direction of behavior synthesis is video games. An agent-based system is usually designed to guide behaviors synthesis in video games. In some works, Finite State Machine [15, 61] is used to synthesize the characters' behaviors, where the bahaviors are regarded as states to construct the state machines. The Belief-Desire-Intention (BDI) model [20] is also an option to deal with behavior synthesis in some games, *e.g.*, the work of [13]). Concepts such as attentional and emotional involvement [64] and immersion [24] also play vital roles in the behavior synthesis in the game designs.

Compared with the behavior synthesis approaches in the applications of video game and robot pets, our approach is mainly different from previous works in terms of two aspects: (1) The behaviors in previous works are generally designed manually based on rules. Our approach synthesizes behaviors based on learning from a real pet dataset. (2) Previous virtual pets are unaware of scene semantics. Our virtual pet perceives the semantics to perform behaviors at appropriate scene locations. These two features entail more realistic and reasonable virtual pets behaviors.

2.3 Scene Semantics Understanding

Another focus of our approach is on using scene understanding knowledge to facilitate behavior synthesis in a real scene. To this end, we detect objects which are typically associated with specific pet behaviors. Object detection aims to determine whether there are any instances of objects from given categories, such as a table, chair, couch in the real-world environment. If any such object is present, the spatial location is returned via a bounding box (an axis-aligned rectangle tightly bounding the object) [18, 53], a precise pixel-wise segmentation mask [65], or a closed boundary [38]. Please refer to the recent survey [39, 65] for a comprehensive review.

Recently, deep learning techniques flourished given their powerful capability of learning representations directly from raw images. Deeper CNNs have led to record-breaking improvements in the detection of general object categories. A region-based framework is commonly used in deep detection approaches, such as DetectorNet [59], OverFeat [55], MultiBox [17], and fast RCNN [22]. Region proposals are first generated from the input. CNN features are extracted from these regions and classifiers are used to determine the category labels of the proposals.

In our approach, a virtual pet is visualized in a 3D scene via mixed reality. Understanding the scene context is a prerequisite for placing the virtual pet at reasonable spatial locations in the real world. To achieve this, we apply Mask RCNN [26], a state-of-the-art segmentation mask approach to obtain the pixel-wise location of furniture objects. Mask RCNN is an extension of Faster RCNN [52], adding a branch to output a binary mask for each region of interest. It detects objects in an image while simultaneously generating a high-quality segmentation mask for each object instance.

3 OVERVIEW

Our goal is to synthesize virtual pets with behavior at appropriate locations naturally and reasonably in a real scene. Fig. 2 shows the overview of our approach for accomplishing this goal.

The input of our approach is a real scene, as which is shown in Fig. 2 (a). It can be captured by a 3D sensor, *e.g.*, the depth camera of a Hololens. The approach consists of two components: data-driven pet behavior generator and pet behavior instantiation. To test our approach, we define five categories of common pet behaviors [6], namely, resting, idling, eating, soiling, and scratching, which our approach can synthesize.

Our approach proceeds as follows. Firstly, it generates a sequence of high-level pet behaviors through a data-driven pet behavior generator. The generator is designed based on a two-layer LSTM network and is trained on an annotated pet behavior dataset. Secondly, each behavior is assigned to take place at a location in the real scene by the pet behavior instantiation, which consists of three phases: scene understanding, location instantiation, and path instantiation. In the scene understanding phase, we generate several object location proposals in the real scene using computer vision

techniques. Each object may be associated with one or more pet behaviors, *e.g.*, a couch is associated with resting and idling behaviors. In the location instantiation phase, based on the object proposals, we assign the generated sequence of high-level pet behaviors to a series of physical locations, where the virtual pet should perform the behaviors. Finally, in the path instantiation phase, we employ a pathfinding algorithm to generate natural and smooth paths for the virtual pet. After completing a behavior at a location, the pet follows the generated path to travel to the next location to perform the next behavior.

By wearing a mixed reality helmet, *e.g.*, a Hololens, a user may observe the virtual pet synthesized by our approach. It is worth noting that we use a virtual kitten as an example to demonstrate our approach. It is possible to extend and apply our approach to synthesize behaviors for other virtual pets as well.

4 DATA-DRIVEN PET BEHAVIOR GENERATOR

The real pet behaviors encode patterns, reflecting the relationships among behaviors, *e.g.*, idling behavior most likely following eating behavior. Applying such patterns to the generation may improve the realism of the virtual pets, resulting in better human experiences. To this end, we apply a data-driven approach to learn a behavior generator to model the behavior patterns based on a real pet dataset. By such a generator, a synthesized virtual pet is able to mimic real pets' behaviors. Suppose $B = \{eating, resting, idling, soiling, scratching\}$ is the behavior set. We automatically generate a behavior sequence $S = (s_1, s_2, \cdots, s_N)$, where $s_n \in B$.

4.1 Pet Behavior Dataset

To train the behavior generator, we annotated a cat dataset [54] with the corresponding behaviors. The original dataset recorded two cats' positions in one apartment over time, which were captured with bluetooth tracking devices. Then a k-nearest neighbors algorithm was used to cluster positions. Each cluster is assigned an object label, indicating that this cluster's positions are nearer to the object. The objects include a table, couch, window, and so forth. In other words, the sequences of positions the cats traversed are used to create an object sequence according to the object present near each position. Totally, the sequence consists of 1,440 minutes data.

In order to annotate the locations with the pet behaviors, we invited 34 participants, who had experienced raising cats for 1 to 10 years, to complete a questionnaire about where pet behaviors typically take place. The participants were given 24 common objects, one by one. They were asked to choose one of the five behaviors from B that they think is most closely associated with each type of object. For example, one may think that *resting* is the most closely associated behavior with a bed.

From the questionnaire, we estimate the frequency of a behavior associated with each type of object. To mitigate the influence of outliers, for each object type, we only consider the behaviors whose votes are higher than 10% as associated with that object type. Then, we assign the behavior labels to the object by sampling from the frequency. The obtained behavior sequences are later used to learn the pet behavior patterns. Please refer to the supplementary for participants' choices for the behavior most closely associated with each object type.

4.2 Sequential Pet Behavior Generation

We utilize a LSTM network [63] to learn pet behavior patterns. A LSTM network is capable of modelling long sequence data patterns well by encoding history information with low computation cost. Although a higher-order Markov chain can also consider history information, learning the Markov chain is usually time expensive and convergence can be problematic.

To train the model, we input the annotated behavior sequences. A $5d$ one-hot vector represents one behavior in the sequence. Each entry of the vector has a binary value, with 1 indicating correspondence to a behavior type and 0 otherwise. For example, if the ith entry is 1, it indicates the ith behavior type. The behavior sequences are divided into fragments, and each fragment contains 100 sequential behaviors. The fragments are then fed to the LSTM network for the training.

We use a two-layer LSTM structure for the training process, each of which consists of 512 hidden units. A dropout of 0.2 is added after each layer. The weights of the network are updated iteratively.

After the training, the LSTM network can predict a behavior based on previous behavior. To enrich the variety of the generated results, we adjust the process slightly in the generation. We do not output the behavior with the highest probability as we do in the training process. Instead, we sample from the probability distribution of all behaviors from the softmax layer. A random behavior initializes the generation. Iteratively, we obtain a behavior sequence $S = (s_1, s_2, \cdots, s_N)$. Please refer to the supplementary materials for the structure in the training and generation process.

5 PET BEHAVIOR INSTANTIATION

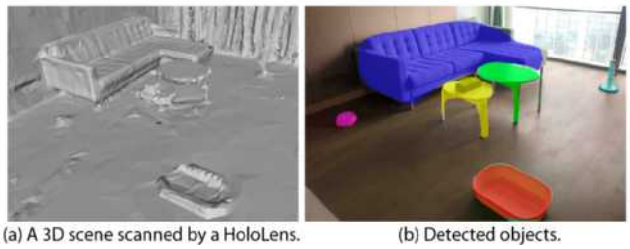
Using the generated behavior sequence, next we instantiate the abstract and high-level behaviors in a real scene, *i.e.* generating a physical location for each behavior to take place and generating a feasible transition path between two locations. As mixed reality aims to fuse the real and virtual worlds, understanding the real scene is the first step for the instantiation, such as knowing the object's location and the scenic terrain. Then the instantiation process is performed according to the understanding results.

Formally, we generate a location sequence $L = (l_1, l_2, \cdots, l_N)$, where the location l_n corresponds to the behavior s_n. Each location l_n is a 3D location at which the behavior s_n takes place.

Besides, we generate a path sequence $P = (p_1, p_2, \cdots, p_{N-1})$. Following the path sequence P, the virtual pet walks around in the scene to perform the synthesized behaviors. If the location changes, *i.e.* $l_n \neq l_{n+1}$, following the path $p_n \in P$, the virtual pet walks from location l_n to the next location l_{n+1}. Each path p_n is a sequence of 3D locations: $p_n = (t_1, t_2, \cdots, t_E)$, where $t_1 = l_n$ is the starting point of the path and $t_E = l_{n+1}$ is the ending point.

5.1 Scene Understanding

We obtain two types of information of the scene in the understanding process: geometry and semantics. The geometry reflects the scene's terrain information, whereas the semantics reflects where and what the objects are.



(a) A 3D scene scanned by a HoloLens. (b) Detected objects.

Figure 3: A scene understanding result. (a) The 3D reconstruction of a scene scanned by a Hololens. (b) The detected objects, with each mask color depicting a distinct object.

Scene Geometry. We use the spatial mapping technique of the Hololens to obtain the 3D model of a scene. For a room of 200 square feet, it takes about 2 minutes to scan and reconstruct its 3D model. The time varies according to the scene's size. The geometry is represented by triangular meshes, as shown in Fig. 3 (a).

Scene Semantics. A behavior may take place around an object, *e.g.*, resting on a bed. Thus, we detect all objects related to pet behaviors. Since detection on 2D images is more robust and accurate, we utilize a 2D object detection approach to detect objects and then project the detection results to 3D space. We take two steps to do the detection.

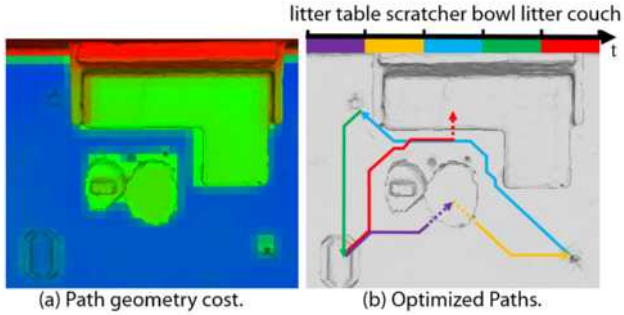
Firstly, we apply the Mask R-CNN approach [26] to detect the objects (where behaviors may take place) on 2D images, which are captured by the Hololens camera. Fig. 3 (b) shows an example of object detection. The objects (*e.g.*, couch) are detected and assigned different colors according to the masks generated by Mask R-CNN.

Secondly, we roughly estimate the camera's parameters from the 2D image using the method of Horry et al. [29]. Specifically, we extract the vanishing point based on the perspective projection principle to estimate the camera parameters. With the estimated parameters, we set up a virtual camera in the 3D scene to render images. Among the rendered images, the image that is most similar to the 2D image in the detection, is used to calculate the corresponding locations of the objects in the 3D scene. Then we project the 2D detection results onto the 3D scene to obtain the categories and locations of the objects in the 3D scene.

5.2 Behavior Location Instantiation

For the generated abstract and high-level behaviors, virtual pets need physical locations to perform them. Because behaviors are related to objects, we use the captured relationship between the behavior and object from the questionnaire in Section 4.1 to obtain the related object. Then the location is represented by the 3D coordinate of a related object's center. For each behavior s_n, we generate a corresponding location l_n so that the virtual pet can perform the behavior s_n at location l_n in the real world.

For each object, we define a prior probability distribution to model the possibility that each behavior may take place at the object. Two constraints are considered: a pet' preference to perform the behavior at that object and the distance from the current location of the virtual pet to the object's location.



(a) Path geometry cost. (b) Optimized Paths.

Figure 4: (a) A visualization of the path geometry cost for each cell. The redder a cell is, the higher its cost value is. (b) An illustration of the optimized paths. The state bar at the top shows the sequence of objects that the pet travels to. Each color refers to a path going from one object's location to another object's location (*e.g.*, purple refers to going from the litter to the table).

Suppose I objects are in the scene. We define the prior probability $\theta_n(i)$ for the pet to perform the nth behavior at the ith object as:

$$\theta_n(i) = \frac{1}{E} f_n(i) \frac{e^{-\frac{1}{d_{\max}}|D(o_0,o_i)|}}{\sum_{i=1}^{I} e^{-\frac{1}{d_{\max}}|D(o_0,o_i)|}}, \tag{1}$$

where $f_n(i)$ is the frequency of the nth behavior taking place at the ith object, $D(\cdot)$ is the Euclidean distance between two locations, where o_0 is the current pet location and o_i is the location of the ith object. The normalization term $d_{\max}$ is the maximum distance between two objects in the scene. $\frac{1}{E}$ is a parameter to normalize $\theta_n(i)$ to $[0, 1]$.

Given a behavior sequence generated from the LSTM network (Section 4.2) and an initial location, we sample a corresponding location sequence $L = (l_1, l_2, \cdots, l_N)$ for the virtual pet according to the probability defined in Eq. (1).

5.3 Path Instantiation

After the location instantiation step, we instantiate a path for every two adjacent locations, *i.e.* l_n and l_{n+1}, to enable the virtual pet to walk around in the real scene. Based on the observation of real pet behavior, we assume empirically that pets prefer to walk on paths with few obstacles during the transition (walking with fewer efforts) and walk in an open area instead of a crowded area (walking in open areas). Thus we model these factors by a cost function defined in Eq. (2) and an adjusted A^* algorithm [25] to optimize the cost.

To calculate the path efficiently, we discretize the 3D space by gridding the surfaces of the 3D scene with $10cm \times 10cm$ cells. Then the path is represented by a sequence of adjacent cells.

Path Cost. Starting from the current location l_n, we apply the A^* algorithm to find a path to the destination l_{n+1} with the smallest path cost. At each iteration, A^* chooses a cell to extend the path. It does so based on three costs: the geometry cost of the next cell, the current path cost, and an estimate of the cost required to extend the path all the way to the destination. We define the path cost as:

$$C_{\text{total}}(m) = \lambda_{\text{g}} C_{\text{g}}(m) + \lambda_{\text{s}} C_{\text{s}}(m) + \lambda_{\text{h}} C_{\text{h}}(m), \tag{2}$$

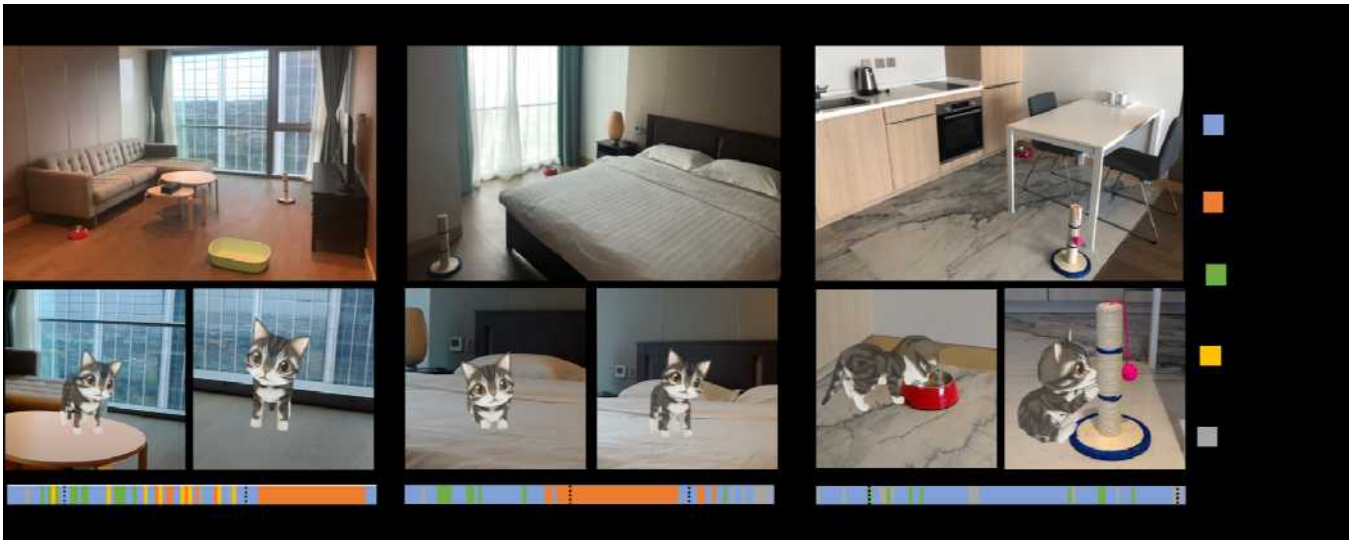


Figure 5: The living room, bedroom, and kitchen scenes used in our experiments. The top row shows the input scene. The middle row demonstrates some generated behaviors. The bottom row shows the generated behavior sequence visualized by a state bar. Each sequence consists of 100 behaviors and each behavior is shown by one color in the bar.

where m is the next cell on the path. $C_{\mathrm{g}}(m)$ is the geometry cost; $C_{\mathrm{s}}(m)$ is the path cost from the starting location to cell m; $C_{\mathrm{h}}(m)$ is a heuristic term which estimates the path cost from m to the destination. λ_{g}, λ_{s}, and λ_{h} are the weights of the costs and are set as 0.2, 0.6, and 0.2, respectively.

We use the geometry cost $C_{\mathrm{g}}(m)$ to let the virtual pet mimic a real pet's movement. We consider two constraints: (1) A real pet usually prefers to spend less energy travelling from one location to another. For example, compared to a path which requires the pet to jump over a high obstacle (*e.g.*, a cupboard) to reach its destination, the pet may prefer another path without high obstacles. (2) A real pet also prefers to move in spacious rather than in crowded areas to reach the destination. To favor such two considerations, we define the geometry cost for the cell m as:

$$C_{\mathrm{g}}(m) = \lambda_{\mathrm{t}} H_{\mathrm{t}}(m) + \lambda_{\mathrm{c}} H_{\mathrm{c}}(m), \tag{3}$$

where $H_{\mathrm{t}}(m)$ is the height of the cell m, which allows us to model the "roughness" of the terrain. This term penalizes paths which require the virtual pet to cross over high objects. $H_{\mathrm{c}}(m)$ is the average height difference between the m cell and its 8 neighbors. $H_{\mathrm{c}}(m)$ penalizes crowded paths. λ_{t} and λ_{c} are the corresponding weights, which are both set as 0.5 by default in our experiments. Fig. 4 (a) visualizes the path geometry cost for each cell of the scene.

$C_{\mathrm{s}}(m)$ is the cost of the path from the start location l_n to m. It is defined as:

$$C_{\mathrm{s}}(m) = \sum_{i=l_n}^{m} C_{\mathrm{g}}(i). \tag{4}$$

$C_{\mathrm{h}}(m)$ is the cost of the path from the extended cell m to the destination l_{n+1}. It is defined as:

$$C_{\mathrm{h}}(m) = \sum_{i=m}^{l_{n+1}} C_{\mathrm{g}}(i). \tag{5}$$

Path Finding. The A^* algorithm selects the path that minimizes the cost defined in Eq. (2). The implementation of A* algorithm uses a priority queue to perform the repeated selection of minimum-cost cells to expand. At each step of the algorithm, the cell with the lowest $C_{\mathrm{total}}(m)$ value is extracted from the queue; and the $C_{\mathrm{s}}(m)$, $C_{\mathrm{g}}(m)$, and $C_{\mathrm{h}}(m)$ values of its neighbor cells are updated accordingly. These neighbor cells are added to the queue.

The algorithm proceeds until the destination cell have a lower $C_{\mathrm{total}}(l_{n+1})$ value than any cell in the queue (or until the queue is empty). The cells along the path constitute the solution, following which the virtual pet can travel from location l_n to location l_{n+1}. Fig. 4 (b) shows some examples of the optimized paths.

6 EXPERIMENTS

Our approach is implemented using C# and Unity 5.6 and is run on a Hololens. Due to the limited computing resource of the Hololens, some components ran on a PC, including object detection, behavior learning and generation. The PC is equipped with 16GB RAM, an Nvidia Titan X graphics card, and a 2.60GHz Intel i7 processor.

We conducted experiments on three common scenes: *living room*, *bedroom*, and *kitchen*. The scenes are shown on the top row in Fig. 5. The initial behavior of the pet in each scene was randomly generated and then propagated to the pet behavior generator. Since some behaviors may not have corresponding objects in one scene, *e.g.*, no soiling behavior related objects in the bedroom and in the kitchen, our approach first examined the objects. If there were not any objects for one behavior, we would avoid sampling the corresponding behavior in the generation. After that, each behavior was instantiated to a physical location in the scene. Finally, the adjusted A^* algorithm optimizes a path for the virtual pet to walk along between two adjacent locations. Please refer to the supplementary materials for the detected object lists in each experiment scene.

There are 5, 4, and 3 behaviors generated in the living room, bedroom, and kitchen scene, respectively. In the bedroom scene, the soiling behavior is excluded because there is no corresponding object, *i.e.* litter, detected in the scene. Similarly, The kitchen scene does not have the resting and soiling behavior.

It is worth noting that some behaviors and objects exist many-to-many relationships. One behavior may happen around different objects. Take the idling behavior in the living room as an example. Its corresponding objects in the scene comprise table, window, and couch. During the location instantiation process, the idling behavior was initiated on the table, on the couch, or near the window, based on the prior probability defined in Eq. 1. As shown in the middle row of Fig. 5 (a), one idling behavior was instantiated on the table, and the other was near the window. Of course, one object may relate with more than one behavior. For example, the bed in the bedroom scene is related with both resting and idling. The middle row of Fig. 5 (b) demonstrates the two behaviors on the bed. In addition, there was a consistent one-to-one match between the behavior of eating and food bowl, the behavior of scratching and the cat scratcher. Fig. 5 (c) shows two examples of the eating and scratching behavior, which were instantiated to the location of the food bowl and of the scratcher, respectively.

Please see the supplementary video for the visualization of the behavior results used in our experiments.

7 USER STUDY

We conducted user studies to validate the effectiveness of our approach and investigated whether the synthesized behaviors were realistic and reasonable. We carried out experiments to evaluate the component of behavior generation, location instantiation, and path instantiation, respectively.

Participants. We recruited 20 participants to take part in the user studies. The participants were aged 18 to 50, consisting of 10 males and 10 females. All subjects reported normal or corrected-to-normal vision with no color-blindness.

10 participants reported that they had a pet cat for more than one year. 8 participants reported that they had the experiences of playing with cats more than once a month. The other 2 participants reported that they had the experiences of playing with cats less than once a month. Note that due to the limited group size, we did not analyze prior ownership influences.

Procedure. Through wearing a HoloLens helmet, the participants may observe the synthesized virtual pet. During the experiments, the participants were allowed to walk around freely in the scene.

Each participant observed the virtual pet in three scenes, i.e. living room, bedroom, and kitchen. The three scenes were shown to the participants with a random order. After experiencing in each scene, the participants were required to answer a questionnaire about their observations. We opted to use a 5-point Likert scale, with 1 meaning "strongly disagree" and 5 meaning "strongly agree".

Please refer to the supplementary materials for the original ratings and the detailed numbers of statistics test results.

	Idle	Rest	Eat	Soil	Scratch
Idle	0.75	0.11	0.04	0.04	0.06
Rest	0.27	0.62	0.01	0.05	0.05
Eat	0.43	0.07	0.38	0.05	0.07
Soil	0.55	0.14	0.05	0.17	0.09
Scratch	0.52	0.13	0.06	0.09	0.20

Table 1: Prior probability matrix for behavior transitions.

7.1 Behavior Generation Evaluation

In this experiment, we investigated the efficacy of the behavior generation component. We compared the behavior sequence generated by our approach to the ones generated by two other approaches. The compared approaches were:

(1) Ours. The behavior sequence was generated based on the LSTM network discussed in Section 4.

(2) Prior sampling. The prior sampling approach resembles a first-order Markov chain approach [60], which is akin to some digital pets synthesis, e.g., Neopets [42]. We created a 5×5 prior probability matrix for the behavior transition, which is shown in Table 1. The element in the ith row and jth column was the frequency of transferring from the ith behavior to the jth behavior, which was estimated from the pet behavior dataset and ranged from 0 to 1. Given the previous behavior, the next behavior was generated by sampling according to the prior probability.

(3) Random sampling. Given a previous behavior, the next behavior was randomly generated by selecting a behavior following a uniform distribution. This approach is similar to some game logic, e.g., random monsters in Dragon Quest [16].

We applied each approach to generate one sequence with 100 behavior states using the same initial behavior (idling) as the input. We control other components such as location and path instantiation that could affect the ratings. In other words, we fixed the location for each behavior in one scene (*e.g.*, resting always took place on the couch in the living room). We also fixed the paths across different locations, which were generated by the path instantiation in our approach.

After observing the generated behaviors, the participants were asked to rate whether "*the virtual pet switches its behavior naturally*". During the experiment, the participants were not explicitly informed which approach we used to generate the pet's behavior. The behavior sequences generated by the three approaches were shown to the participants in a random order.

Fig. 6 shows the visualization of the participants' ratings using box plots. We conducted a Friedman test on the participants' ratings in all three scenes overall. The results showed a significant difference among the three approaches ($\chi^2 = 37.86, p < .05, df = 2$) at the $\alpha = 0.05$ significance level. A post-hoc test using Wilcoxon Signed-Ranks Test with Bonferroni correction (at the correlated significance level of $\alpha = 0.017$) showed that the median rating of our approach (Md = 4, SD = 0.93) was significantly higher than those of the prior sampling approach (Md = 3, SD = 0.93)($W = 320.5, p < 0.017, r = 0.43$) and the random sampling approach (Md = 2, SD = 1.18)($W = 180.0, p < 0.017, r = 0.65$). The results indicated that the behaviors synthesized by our approach were

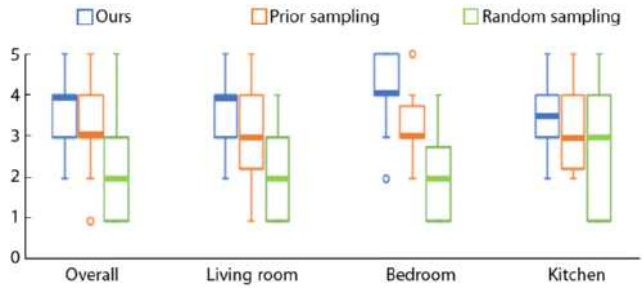


Figure 6: The box plots of the participants' ratings on the behavior sequences generated by our approach, the prior sampling approach, and the random sampling approach. The bottom and top edges of the boxes depict the 25th and 75th percentiles, respectively. The horizontal lines depict the median ratings. The whiskers extend to the most extreme data points. The circles depict the outlier ratings.

perceived as more natural than the behaviors synthesized by the prior and random sampling approaches.

Furthermore, we conducted a Friedman test on the participants' ratings in each scene individually to investigate whether our approach is efficient in each scene. The results also showed significant differences among three approaches in the living room ($\chi^2 = 14.19, p < .05, df = 2$) and bedroom ($\chi^2 = 23.76, p < .05, df = 2$). The Wilcoxon Signed-Ranks Test with Bonferrroni correction (at the correlated significance level of $\alpha = 0.017$) supported that the behavior sequence generated by our approach in the living room (Md = 4, SD = 0.89) was perceived as more natural than those generated by the prior sampling approach (Md = 3, SD = 1.02)($W = 19.5, p < 0.017, r = 0.63$) and the random sampling approach (Md = 2, SD = 1.12)($W = 18.0, p < 0.017, r = 0.66$). In the bedroom scene, the results were similar. Please refer to the supplementary materials for more details.

The Friedman test did not show any significant difference among approaches for the kitchen scene. We believe the reason is that there were only three behaviors in the kitchen (i.e. idling, eating, and scratching) which were all short-term behaviors. One advantage of our generator is modeling behavior patterns, especially long-term behaviors. For "long-term", the t-th behavior could be affected by the 1st to t-1th behaviors; for "short-term", a common setting is to consider information in a time window, *e.g.*, several frames.

For example, when the resting behavior was generated by our approach, the virtual pet stayed at one place for a while to perform "resting". In constrast, the compared approaches generated a behavior for the virtual pet to prompt it to frequently and quickly switch from the resting state to another state, which the participants might find unnatural for the virtual pet. This quick switching pattern occurred more frequently in the living room and bedroom scenes when using the compared approaches, which might have prompted the participants to rate our approach higher for those scenes. Since the kitchen contains only short-term behaviors, it was difficult for the participants to perceive apparent differences among the behaviors generated by different approaches.

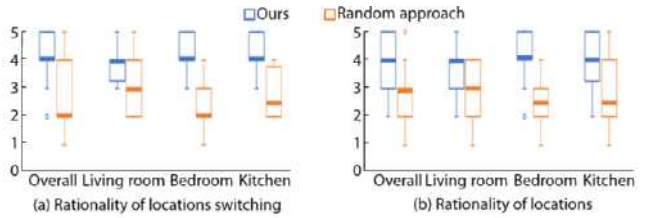


Figure 7: The box plots of the participants' ratings on the locations instantiated by our approach and by the random approach. The bottom and top edges of the boxes depict the 25th and 75th percentiles, respectively. The horizontal lines depict the median ratings. The whiskers extend to the most extreme data points. The circles depict the outlier ratings.

7.2 Location Instantiation Evaluation

In this experiment, we want to evaluate the design of location instantiation. We compared two approaches:

(1) Ours. The behavior location was instantiated by our approach as discussed in Section 5.2.

(2) Random approach. Performing a given behavior at the location of an object which is randomly selected using a uniform distribution among the possible objects of that behavior. For example, in the living room, the "idling" behavior could be performed at 3 possible objects (table, window, and couch), and the "idling" behavior was instantiated at the location of one of these objects selected with a probability of 0.33.

We also controlled the behavior generation and path instantiation in this study. Both approaches used the same behavior sequence generated by our approach in Section 7.1 as input to instantiate the corresponding location sequence for each behavior. We fixed the paths between every two objects, which were generated by the path instantiation component of our approach (Section 5.3).

After the participants observed the virtual pets in the scene, we asked them two questions to investigate the efficacy of the location instantiation component: (1) whether *"the switch between two locations are reasonable"* and (2) whether *"the locations are reasonable for the behaviors"*. The participants' answers to these two questions are visualized by box plots in Fig. 7 (a) and (b) respectively. We carried out a Wilcoxon Signed-Ranks Test to analyze the ratings.

For the first question about the rationality of location switching, the Wilcoxon Signed-Ranks Test indicated that the median rating on the results of our approach in all three scenes overall (Md = 4, SD = 0.77) was significantly higher than that of the random approach (Md = 2, SD = 0.97)($z = 5.478, p < .05$). For the individual scene, our approach performed similarly. The median ratings were all 4 (the standard deviations were 0.77, 0.80 and 0.70 respectively) and were significantly higher than the ones of the random approach in the living room (Md = 3, SD = 0.97)($z = 2.715, p < .05$), the bedroom (Md = 2, SD = 1.02)($z = 3.186, p < .05$), and the kitchen (Md = 2.5, SD = 0.83)($z = 3.690, p < .05$). The results supported that the location switches generated by our approach were more rational than the ones generated by the random approach.

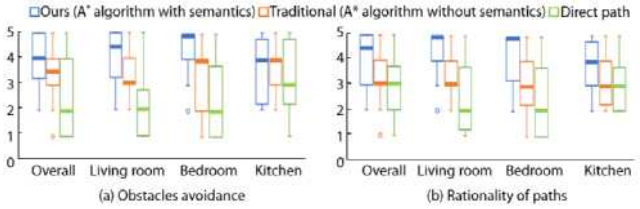


Figure 8: The box plots of the participants' ratings on the paths instantiated by our approach, by the traditional approach, and the direct path approach. The bottom and top edges of the boxes depict the 25th and 75th percentiles, respectively. The horizontal lines depict the median ratings. The whiskers extend to the most extreme data points. The circles depict the outlier ratings.

For the second question about location rationality, the Wilcoxon Signed-Ranks Test indicated that the median rating of our approach in all three scenes overall (Md = 4, SD = 0.84) was also significantly higher than that of the random approach (Md = 3, SD = 1.03)($z = 5.103, p < .05$). For the three individual scenes, the Wilcoxon Signed-Ranks Test results supported that the median ratings of our approach were all significantly higher than the ones of the random approach. For example, for the kitchen scene, the median rating of our approach (Md = 4, SD = 0.83) was significantly higher than that of the random approach (Md = 2.5, SD = 1.19)($z = 2.775, p < .05$).

7.3 Path Instantiation Evaluation

We compared the path instantiation component with other two approaches:

(1) Ours. The path was optimized by the adjusted A^* algorithm by considering the geometry cost (Equation (3)) in the optimization.

(2) Traditional approach. The traditional approach is commonly adopted by robots (*e.g.*, [10]). The path was optimized by a traditional A^* algorithm without considering the scene geometry. To make the compared approach more reasonable, we assume that the virtual pet prefers to walk on the ground when it travels from one location to another. Before the optimization, we preprocessed the scene by excluding cells whose height was higher than 10 cm.

(3) Direct path approach. The direct approach is commonly adopted by most AR applications (*e.g.*, [43, 57]). The path was a straight line between two locations. This approach considered neither scene geometry nor physical constraints. If there were some obstacles on the straight-line path, the virtual pet would cross it directly.

The participants were asked to answer two questions about the obstacles avoidance and the rationality of the paths: (1) whether *"the virtual pet takes rational paths to avoid the obstacles"*. and (2) whether *"the virtual pet moves around with reasonable paths"*. The ratings are visualized by box plots in Fig. 8 (a) and (b), respectively.

We conducted Friedman tests on the participants' ratings on obstacles avoidance. The results showed a significant difference among the three approaches in all the scenes ($\chi^2 = 29.84, p < .05, df = 2$) at the $\alpha = 0.05$ significance level. The results of post-hoc tests using Wilcoxon Signed-Ranks Test with Bonferroni correction (at the correlated significance level of $\alpha = 0.017$) indicated that the median rating of our approach was significantly higher (Md = 4, SD = 1.09) than the median ratings of both the traditional approach (Md = 3.5, SD = 0.99)($W = 222.5, p < 0.017, r = 0.44$) and the direct path approach (Md = 2, SD = 1.43)($W = 234, p < 0.017, r = 0.59$) in all of the scenes. When the tests were done on the scenes individually, the results supported similar conclusions in the living room scene and in the bedroom scene. Take the bedroom scene as an example. The Friedman tests showed a significant difference among the three approaches ($\chi^2 = 11.39, p < .05, df = 2$). The results of the post-hoc tests using Wilcoxon Signed-Rank Test with Bonferroni correction (at the correlated significance level of $\alpha = 0.017$) indicated that the median rating of our approach was significantly higher (Md = 5, SD = 0.95) than the median ratings of the traditional approach (Md = 4, SD = 1.23)($W = 17, p < 0.017, r = 0.60$) and the direct path approach (Md = 2, SD = 1.50) ($W = 29, p < 0.017, r = 0.64$).

Interestingly, we found no statistically significant differences between our approach and other approaches for the kitchen scene. We believe this is due to the characteristics of the kitchen scene's layout. It has few obstacles between any two locations. So the participants could not perceive obvious differences among the results of the three approaches.

To investigate the rationality of paths, we conducted the Friedman test on the participants' ratings. The test showed statistically significant differences among the three approaches at the $\alpha = 0.05$ significance level for all three scenes overall ($\chi^2 = 34.60, p < .05, df = 2$), the living room scene ($\chi^2 = 16.54, p < .05, df = 2$), the bedroom scene ($\chi^2 = 12.48, p < .05, df = 2$), and the kitchen scene ($\chi^2 = 7.05, p < .05, df = 2$).

We conducted post-hoc tests using Wilcoxon Signed-Ranks Test with Bonferroni correction (at the correlated significance level of $\alpha = 0.017$) to determine the differences among the three approaches. The results indicated that the median rating of our approach (Md = 4.5, SD = 1.09) was significantly higher than the median ratings of the traditional approach (Md = 3, SD = 1.00)($W = 171.5, p < 0.017, r = 0.57$) and the direct path approach (Md = 3, SD = 1.43)($W = 174.0, p < 0.017, r = 0.60$) for all three scenes overall. For the individual scene tests, the results supported similar conclusions for the living room and the bedroom. For example, for the living room, the median rating of our approach (Md = 5, SD = 1.09) is significantly higher than the median ratings of the traditional approach (Md = 3, SD = 0.77) ($W = 24, p < 0.017, r = 0.58$) and the direct path approach (Md = 2, SD = 1.07)($W = 8.0, p < 0.017, r = 0.76$). For the kitchen, the Wilcoxon Signed-Ranks Test did not show a significant difference in the ratings of our approach and the traditional approach ($W = 25.0, p = 0.067, r = 0.41$). It did not find a significant difference in the ratings of our approach and direct approach neither ($W = 26.5, p = 0.097, r = 0.20$).

7.4 User Feedback

In the experiments, most participants thought that the virtual kitten driven by our approach was interesting, appealing, and vivid. They stated that if there was such a virtual pet application, they would like to try it. To further analyze users' attitudes, we conducted sentiment analysis of the user comments via the Stanford CoreNlp natural language processing toolkit [40]. We input the

comments to the model and got the attitudes as "positive" or "negative" respectively. The results show that most participants (18 of 20) commented positively on our approach. There were some adjectives they used to describe their positive user experiences: realistic, vivid, attractive, etc. Several users (2 of 20) left negative comments. They thought they "disliked virtual pets" and "had no interests" in using such an application.

Some participants stated that the virtual kitten was smarter than those they had seen in other pet applications. For example, one participant said "*It is amazing that the pet can choose different places to sleep. It seems that the pet understands the real environment*". Another participant commented that "*It is surprising to see that the pet crossed the obstacle in the real scene*" after experiencing the compared direct path approach, though it was a common strategy in most current applications. Overall, most participants were aware of the effect of each component in our pipeline, e.g., generating behaviors akin to real pets, considering the surroundings. Some participants commented on the relationship between the pet and real-world objects in the scene. For example, some participants commented that the sheet should be wrinkled as the kitten walked on it for the bedroom scene. This is an issue with regards to physical simulation. While in our work, we focused on how pets understand the real world and behave naturally.

On the other hand, some participants asked whether they could select another type of animal as their virtual pet, *e.g.*, a dog. We used a kitten as an example to demonstrate our approach. We could extend our pipeline to synthesize the behaviors for other pets. Some participants expressed interests in raising some imaginary pets, *e.g.*, dragon, unicorn. Such feedbacks inspire us to create behavior generators for imaginary pets in our future work, *e.g.*, we may augment a real pet behavior dataset with some fictional pet behaviors to train a behavior generator for imaginary pets.

8 CONCLUSION

We proposed to synthesize natural and reasonable virtual pet behaviors according to the semantics of a real scene. Our approach learned behavior patterns from real pet data to synthesize high-level behavior sequences. We leveraged computer vision techniques to associate the high-level behaviors with a real scene by generating the corresponding location sequences and paths across locations.

We focused on exploring the pipeline of generating virtual pet behaviors by considering scene context. The behavior generator was trained on a real cat's location dataset. Using this dataset, there were two limitations : (1) The locations were captured in an indoor scene through Bluetooth devices. Trained on this dataset, our approach was only applied for synthesizing indoor behaviors. It is also interesting to synthesize outdoor behaviors by learning from outdoor data captured by sensors, *e.g.*, GPS trackers. (2) The original dataset only specified the pet's locations. We used questionnaire responses to estimate the prior probability of a behavior being associated with a location to annotate locations with possible behaviors indirectly. In the field of Animal-Computer Interaction (ACI), recent efforts have been made to gather pet behaviors automatically [3]. Pons et al. [49, 50] have extensively looked into the data collection methods. Using such pet behavior data may increase the variety and performance of the behavior generator.

Our approach can run in scenes besides the tested ones. However, scene complexity could affect the results. For example, object detection could be inaccurate in a cluttered scene, which affects the instantiation phase. Due to the performance limitations of the Hololens, some objects might not be captured or detected, which might affect the results of our approach. For example, the Hololens helmet could not capture and reconstruct black objects. Although object detection approaches achieve good performance, some failure cases are caused by occlusion or non-uniform illumination. In our framework, the path planning is performed once in path instantiation due to HoloLens's limited computing power, so we cannot handle dynamic scenes currently. If the computing power allows, in behavior instantiation, the scene model and path planning can be updated in real-time considering dynamic obstacles, *e.g.*, a moving person.

Our experiments aimed to validate each component of our approach, so the questionnaire was mainly about each designed component. Integrating our synthesis framework with specific applications to evaluate the overall interaction experiences is a promising future direction. For example, the therapeutic effect can be investigated with more realistic and vivid pets behaviors. We believe the findings may shed light on the importance of rational behavior synthesis, *e.g.*, user engagement.

As an early attempt to apply scene semantics for animating virtual pets, we synthesized abstract and high-level behaviors by our approach. The low-level actions and poses of the pet were pre-scripted animations in our experiments. One possible future direction is to explore low-level pose and motion synthesis, which can complement the high-level behaviors for obtaining fine-grained behaviors. For example, different realistic poses for the pet can be generated when instantiating a behavior by considering the 3D geometry proximal to the pet. Such low-level pose and motion variations would make the virtual pet appear more vivid.

Our proposed approach mainly focused on autonomous behavior synthesis. The interactions with users also play a vital role in virtual pet applications such as education and therapeutic studies. A possible extension of our work is to trigger virtual pet's behaviors by a user's voice, gesture, and gaze, which could be captured by a HoloLens. The user's commands may take priority over the pet's autonomous behaviors. Designing more interactive modes will be another exciting direction in virtual pets research.

Another future direction is to extend our approach to drive robot pets. Currently, robot pets can partially mimic a real pet, showing a realistic appearance, making sounds like real pets, and performing some characteristic actions. However, robot pets cannot generally behave realistically according to a real scene's semantics. Our approach could complement robot pet techniques by synthesizing scene-aware behaviors. Equipped with high-quality sensors, a robot pet can capture a map of its environment in real-time and obtain the 3D model. Thus our approach could be applied for generating and instantiating behaviors for robot pets.

In the future, we may also extend our approach to the pets synthesis in video games. In a video game, the scene's arrangement is known beforehand as it is created manually. Our approach can use the categories and locations of the objects in the game scene to synthesize behaviors and paths for a virtual pet automatically.

ACKNOWLEDGMENTS

We thank Jingwen Gao for all her assistance with the experiments and the demonstrations. Wei Liang was supported by the National Natural Science Foundation of China (NSFC) under Grant No. 61972038. Lap-Fai Yu was supported by an NSF CAREER Award (award number: 1942531).

REFERENCES

[1] Sun Joo Ahn, Kyle Johnsen, Tom Robertson, James Moore, Scott Brown, Amanda Marable, and Aryabrata Basu. 2015. Using virtual pets to promote physical activity in children: An application of the youth physical activity promotion model. *Journal of health communication* 20, 7 (2015), 807–815.

[2] Eric Lewin Altschuler. 1999. Pet-facilitated therapy for posttraumatic stress disorder. *Annals of Clinical Psychiatry* 11, 1 (1999), 29–30.

[3] Shir Amir, Anna Zamansky, and Dirk van der Linden. 2017. K9-Blyzer: Towards video-based automatic analysis of canine behavior. In *Proceedings of the Fourth International Conference on Animal-Computer Interaction*. 1–5.

[4] Ling Bao and Stephen S Intille. 2004. Activity recognition from user-annotated acceleration data. In *International conference on pervasive computing*. Springer, 1–17.

[5] Andrea Beetz, Kerstin Uvnäs-Moberg, Henri Julius, and Kurt Kotrschal. 2012. Psychosocial and psychophysiological effects of human-animal interactions: the possible role of oxytocin. *Frontiers in psychology* 3 (2012), 234.

[6] Linda P Case et al. 2003. *The cat: its behavior, nutrition & health*. Iowa State Press.

[7] Zhi-Hong Chen, Calvin Liao, Tzu-Chao Chien, and Tak-Wai Chan. 2011. Animal companions: Fostering children's effort-making by nurturing virtual pets. *British Journal of Educational Technology* 42, 1 (2011), 166–180.

[8] Thomas Chesney and Shaun Lawson. 2007. The illusion of love: Does a virtual pet provide the same companionship as a real one? *Interaction Studies* 8, 2 (2007), 337–342.

[9] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

[10] Shi-Gang Cui, Hui Wang, and Li Yang. 2012. A simulation study of A-star algorithm for robot path planning. In *16th international conference on mechatronics technology*. 506–510.

[11] Hayley Cutt, Billie Giles-Corti, Matthew Knuiman, Anna Timperio, and Fiona Bull. 2008. Understanding dog owners' increased levels of physical activity: results from RESIDE. *American journal of public health* 98, 1 (2008), 66–69.

[12] Ruta Desai, Fraser Anderson, Justin Matejka, Stelian Coros, James McCann, George Fitzmaurice, and Tovi Grossman. 2019. Geppetto: Enabling Semantic Design of Expressive Robot Behaviors. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–14.

[13] Rodrigo Dias and Carlos Martinho. 2011. Adapting content presentation and control to player personality in videogames. In *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology*. 1–8.

[14] Tawanna Dillahunt, Geof Becker, Jennifer Mankoff, and Robert Kraut. 2008. Motivating environmentally sustainable behavior changes with a virtual polar bear. In *Pervasive 2008 Workshop Proceedings*, Vol. 8. 58–62.

[15] Eric Dybsand. 2001. A Generic Fuzzy State. *Game Programming Gems 2* (2001), 337.

[16] ENIX. 2017. DragonQuest XI. https://https://dragonquest.square-enix-games.com/tact/en-us/..

[17] Dumitru Erhan, Christian Szegedy, Alexander Toshev, and Dragomir Anguelov. 2014. Scalable object detection using deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2147–2154.

[18] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. 2010. The pascal visual object classes (voc) challenge. *International journal of computer vision* 88, 2 (2010), 303–338.

[19] Iram Fatima, Muhammad Fahim, Young-Koo Lee, and Sungyoung Lee. 2013. A unified framework for activity recognition-based behavior analysis and action prediction in smart homes. *Sensors* 13, 2 (2013), 2682–2699.

[20] Michael Georgeff, Barney Pell, Martha Pollack, Milind Tambe, and Michael Wooldridge. 1998. The belief-desire-intention model of agency. In *International workshop on agent theories, architectures, and languages*. Springer, 1–10.

[21] Andrew Gilbey, June McNicholas, and Glyn M Collis. 2007. A longitudinal test of the belief that companion animal ownership can help reduce loneliness. *Anthrozoös* 20, 4 (2007), 345–353.

[22] Ross Girshick. 2015. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*. 1440–1448.

[23] Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850* (2013).

[24] Juho Hamari, David J Shernoff, Elizabeth Rowe, Brianno Coller, Jodi Asbell-Clarke, and Teon Edwards. 2016. Challenging games help students learn: An empirical study on engagement, flow and immersion in game-based learning. *Computers in human behavior* 54 (2016), 170–179.

[25] Peter E Hart, Nils J Nilsson, and Bertram Raphael. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics* 4, 2 (1968), 100–107.

[26] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*. 2961–2969.

[27] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[28] Daniel Holden, Jun Saito, and Taku Komura. 2016. A deep learning framework for character motion synthesis and editing. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 138.

[29] Youichi Horry, Ken-Ichi Anjyo, and Kiyoshi Arai. 1997. Tour into the picture: using a spidery mesh interface to make animation from a single image. (1997).

[30] Kyle Johnsen, Sun Joo Ahn, James Moore, Scott Brown, Thomas P Robertson, Amanda Marable, and Aryabrata Basu. 2014. Mixed reality virtual pets to reduce childhood obesity. *IEEE transactions on visualization and computer graphics* 20, 4 (2014), 523–530.

[31] Malte F Jung. 2017. Affective grounding in human-robot interaction. In *2017 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI*. IEEE, 263–273.

[32] Andrej Karpathy, Armand Joulin, and Li F Fei-Fei. 2014. Deep fragment embeddings for bidirectional image sentence mapping. In *Advances in neural information processing systems*. 1889–1897.

[33] David W Kritt. 2000. Loving a virtual pet: Steps toward the technological erosion of emotion. *The Journal of American Culture* 23, 4 (2000), 81.

[34] Vining Lang, Wei Liang, and Lap-Fai Yu. 2019. Virtual agent positioning driven by scene semantics in mixed reality. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 767–775.

[35] Bettina Laugwitz, Theo Held, and Martin Schrepp. 2008. Construction and evaluation of a user experience questionnaire. In *Symposium of the Austrian HCI and usability engineering group*. Springer, 63–76.

[36] Ana Lilia Laureano-Cruces and Arturo Rodriguez-Garcia. 2012. Design and implementation of an educational virtual pet using the OCC theory. *Journal of Ambient Intelligence and Humanized Computing* 3, 1 (2012), 61–71.

[37] S Lawson and Thomas Chesney. 2007. The impact of owner age on companionship with virtual pets. *Eighth International Conference on Information Visualisation* (2007).

[38] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*. Springer, 740–755.

[39] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. 2018. Deep learning for generic object detection: A survey. *arXiv preprint arXiv:1809.02165* (2018).

[40] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*. 55–60.

[41] Microsoft. 2015. HoloPet. https://www.windowscentral.com/microsoft-registers-holopet-trademark-augmented-reality-kittens-all.

[42] Neopets. 2003. http://neopets.com..

[43] NeuroHive. 2019. Pets ARound - Virtual Friend. https://appadvice.com/app/pets-around-virtual-friend/1441956434.

[44] Nahal Norouzi, Kangsoo Kim, Myungho Lee, Ryan Schubert, Austin Erickson, Jeremy Bailenson, Gerd Bruder, and Greg Welch. 2019. Walking your virtual dog: Analysis of awareness and proxemics with simulated support animals in augmented reality. In *2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 157–168.

[45] Nuria Oliver, Ashutosh Garg, and Eric Horvitz. 2004. Layered representations for learning and inferring office activity from multiple sensory channels. *Computer Vision and Image Understanding* 96, 2 (2004), 163–180.

[46] Fco Javier Ordóñez, José Antonio Iglesias, Paula De Toledo, Agapito Ledezma, and Araceli Sanchis. 2013. Online activity recognition using evolving classifiers. *Expert Systems with Applications* 40, 4 (2013), 1248–1255.

[47] Heather L O'Brien, Paul Cairns, and Mark Hall. 2018. A practical approach to measuring user engagement with the refined user engagement scale (UES) and new UES short form. *International Journal of Human-Computer Studies* 112 (2018), 28–39.

[48] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. 2018. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 143.

[49] Patricia Pons and Javier Jaen. 2016. Towards the Creation of Interspecies Digital Games: An Observational Study on Cats' Interest in Interactive Technologies. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. 1737–1743.

[50] Patricia Pons, Javier Jaen, and Alejandro Catala. 2017. Assessing machine learning classifiers for the detection of animals' behavior using depth-based tracking. *Expert Systems with Applications* 86 (2017), 235–246.

[51] Thomas S Ray. 2001. Aesthetically evolved virtual pets. *Leonardo* 34, 4 (2001), 313–316.
[52] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*. 91–99.
[53] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision* 115, 3 (2015), 211–252.
[54] Sana-malik. 2014. Cat Location Dataset. Github.
[55] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. 2013. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229* (2013).
[56] Sichao Song and Seiji Yamada. 2017. Expressing emotions through color, sound, and vibration with an appearance-constrained social robot. In *2017 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI*. IEEE, 2–11.
[57] Playrock Studios. 2018. Wildlife AR. https://play.google.com/store/apps/dev?id=8981397870093114957.
[58] Ilya Sutskever, James Martens, and Geoffrey E Hinton. 2011. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*. 1017–1024.
[59] Christian Szegedy, Alexander Toshev, and Dumitru Erhan. 2013. Deep neural networks for object detection. In *Advances in neural information processing systems*. 2553–2561.
[60] Tim Van Kasteren, Athanasios Noulas, Gwenn Englebienne, and Ben Kröse. 2008. Accurate activity recognition in a home setting. In *Proceedings of the 10th international conference on Ubiquitous computing*. ACM, 1–9.
[61] Ferdinand Wagner, Ruedi Schmuki, Thomas Wagner, and Peter Wolstenholme. 2006. *Modeling software with finite state machines: a practical approach*. CRC Press.
[62] Hao Wang and Jing Liu. 2009. Mobile phone based health care technology. *Recent Patents on Biomedical Engineering* 2, 1 (2009), 15–21.
[63] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*. 802–810.
[64] Chloe Shu-Hua Yeh. 2015. Exploring the effects of videogame play on creativity performance and emotional responses. *Computers in Human Behavior* 53 (2015), 396–407.
[65] Xin Zhang, Yee-Hong Yang, Zhiguang Han, Hui Wang, and Chao Gao. 2013. Object class detection: A survey. *ACM Computing Surveys (CSUR)* 46, 1 (2013), 10.