applied sciences

MDPI

Article

Silhouettes from Real Objects Enable Realistic Interactions with a Virtual Human in Mobile Augmented Reality

Hanseob Kim [1,2], Ghazanfar Ali [1,3], Andréas Pastor [4], Myungho Lee [5], Gerard J. Kim [2] and Jae-In Hwang [1,*]

1 Center for Artificial Intelligence, Korea Institute of Science and Technology (KIST), Seoul 02792, Korea; khseob0715@kist.re.kr (H.K.); aliust@ust.ac.kr (G.A.)
2 Department of Computer Science and Engineering, Korea University, Seoul 02841, Korea; gjkim@korea.ac.kr
3 Division of NT-IT, University of Science and Technology, Seoul 02792, Korea
4 LS2N, University of Nantes, 44200 Nantes, France; andreas.pastor@univ-nantes.fr
5 School of Computer Science and Engineering, Pusan National University, Busan 46241, Korea; myungho.lee@pnu.edu
* Correspondence: hji@kist.re.kr

Abstract: Realistic interactions with real objects (e.g., animals, toys, robots) in an augmented reality (AR) environment enhances the user experience. The common AR apps on the market achieve realistic interactions by superimposing pre-modeled virtual proxies on the real objects in the AR environment. This way user perceives the interaction with virtual proxies as interaction with real objects. However, catering to environment change, shape deformation, and view update is not a trivial task. Our proposed method uses the dynamic silhouette of a real object to enable realistic interactions. Our approach is practical, lightweight, and requires no additional hardware besides the device camera. For a case study, we designed a mobile AR application to interact with real animal dolls. Our scenario included a virtual human performing four types of realistic interactions. Results demonstrated our method's stability that does not require pre-modeled virtual proxies in case of shape deformation and view update. We also conducted a pilot study using our approach and reported significant improvements in user perception of spatial awareness and presence for realistic interactions with a virtual human.

Keywords: augmented reality; mobile AR; deep learning; segmentation; realistic interaction; virtual human; perceptual issue; occlusion, multi-modal system; user experience

Citation: Kim, H.; Ali, G.; Pastor, A.; Lee, M.; Kim, G.J.; Hwang, J.-I. Silhouettes from Real Objects Enable Realistic Interactions with a Virtual Human in Mobile Augmented Reality. *Appl. Sci.* **2021**, *11*, 2763. https://doi.org/10.3390/app11062763

Academic Editor: Luis M. Camarinha-Matos

Received: 2 March 2021
Accepted: 17 March 2021
Published: 19 March 2021

1. Introduction

Unlike virtual reality (VR) which immerses users into a virtual environment, augmented reality (AR) allows users to see a mixed environment where virtual objects are superimposed on the real views [1]. Users wear AR glasses or use handheld devices such as a smartphone or tablet to see the mixed environment and interact with virtual objects in real-time. Since users can see the real environment (see Figure 1a), AR systems often require an accurate registration of virtual objects to provide seamless interactions in various situations [2,3]. Incorrect registration of a virtual object in the real space can cause unrealistic occlusions [4,5] or physically implausible situations [6,7], leading to perceptual quality degradation and breaks in presence [8].

The interaction between virtual humans and real objects such as animals, toys, and robots can improve user experience in information dissemination [9]. Since people, especially kids, expect virtual humans to behave as real humans [10], it is necessary to provide realistic interaction with real objects. Commercial apps in the entertainment industry usually use a pre-modeled real object as a virtual proxy [11–13]. The apps then exploit a virtual proxy to provide an illusion that the virtual object seems to interact with a physical object (a.k.a physical–virtual interaction [7]). However, modeling real objects in advance

and importing them into the system are labor-intensive and not trivial tasks. We discuss in detail the benefits of interaction with real objects in cognitive terms and the difficulty of supporting interactions with pre-modeled objects in Section 2.

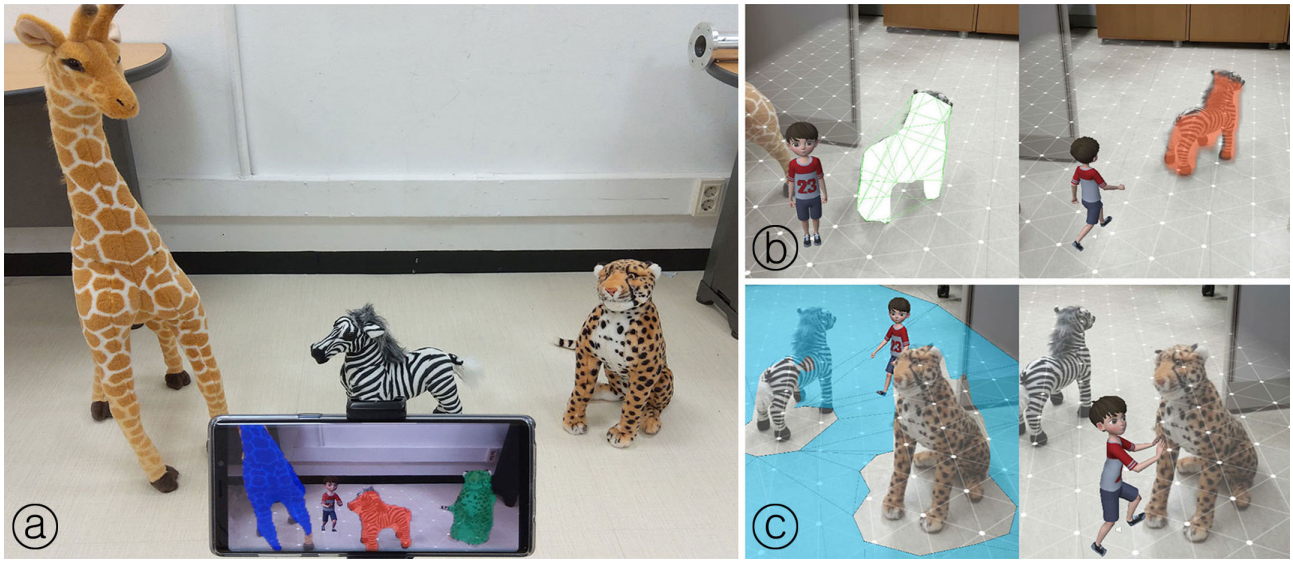


Figure 1. Overview of the proposed system. (**a**) Segmenting the multiple deformable real objects on the smartphone. (**b**) Generating the virtual proxy that replaces a zebra doll in the augmented reality (AR) space to enable virtual human interacts with real objects. We call this type of virtual proxy a Silhouette Mesh. (**c**) Using the silhouette meshes, a virtual human walks among the dolls in the walkable areas and interacts with a jaguar doll.

In this paper, we present a practical method that enables augmented virtual humans to interact with physical objects in real space without the need for pre-modeled virtual proxies on mobile devices. We use the real object's silhouettes instead of using pre-modeled proxies to support interactions. To extract the silhouette from the real objects, we trained the segmentation module on content-related objects (animal dolls, in our case). We then pass the camera feed to the segmentation module, which recognizes the interactable objects and creates a segmentation mask (see Figure 1a). This mask is passed to the mesh creation module. The creation module builds a silhouette mesh (i.e., virtual proxy) and places it in the augmented space corresponding to the real object's location (see Figure 1b).

For the case study, we applied our proposed method to the multimodal-based mobile AR system. Our system has a scenario in which a virtual human interacts with real animal dolls. This system aims to allow users to experience interactive AR when playing with dolls. Our virtual human acts as a playmate who answers children's inquiries or interacts with animal dolls in response to commands. For example, the virtual human provides users with realistic interactions by leveraging the silhouette mesh (see Figure 1c walking among dolls, pushing a doll). We demonstrate the stability of our silhouette meshes that support realistic/seamless interactions with real objects that are even deformable in mobile AR. We also conducted a pilot study to evaluate the impacts of our method on user perception of AR interactions. Our participants reported significantly improved perception with a virtual human (e.g., naturalness, physical ability) and spatial sense of AR environment (e.g., awareness, presence) when they experienced interaction using silhouette mesh. In the rest of the paper, we discuss the benefits of our proposed method in detail.

This paper is organized as follows: Section 2 briefs conventional methods in supporting interactions with real objects in AR. We detail our proposed method, silhouette mesh, in Section 3. Then we provide the performance evaluation of our method on mobile devices in Section 4. Section 5 demonstrates silhouette mesh-based realistic interactions between a virtual human and real animal dolls in a multimodal AR system, and also reports our pilot study's details and results. Finally, we discuss limitations and possible solutions in Section 6 and conclude the paper in Section 7.

2. Background

This section describes previous works on interactions with the real object in AR and segmentation methods that extract silhouettes from real objects to create virtual proxies.

2.1. Interacting with Real Objects in AR

AR systems need to focus on providing realistic interactions with real objects such as toys, dolls, and robots for providing a successful user experience [14–18]. To enable virtual objects to interact with real objects, AR systems often use a virtual proxy corresponding to the real object. Since virtual proxies should have similar characteristics to real objects (e.g., appearance, location in the real world), commercial apps usually model real objects in advance [11–13,18–20]. Then, to accurately place the virtual proxy in the 3D space, AR apps place a marker for reference [17,18]. These markers help the AR Apps deploying proxies and coordinate (i.e., overlay) them with real objects [21]. After deployment, these proxies are rendered transparently in AR views to prevent users from being aware of proxies. Thus, virtual humans interact with these invisible virtual proxies, but users perceive that the virtual human is interacting with real objects [6].

Such an illusion by physical-virtual interactions can affect users positively [6,7], but modeling real objects and importing them on the system are not trivial tasks [16,18]. This is because commercial systems often add new objects or new interactions to keep users interested. The addition of new real objects requires their modeling, but each time it is costly and labor-intensive. Furthermore, if a real object is deformed: shape and/or size (e.g., a moving animal, fallen doll, user's hand) at runtime, the system may either lost track or provide an incorrect interaction [22–24]. Incorrect interactions may reflect poorly on the user experience [5,25,26]. In the case of research on virtual human, Kim et al. [6] reported that the incorrect interactions negatively affects the social/co-presence with an augmented virtual human in the shared space.

To solve these limitations, several AR studies [27–30] reconstruct real objects and space at runtime by using advanced depth cameras with other spatial sensors. However, these systems are still bulky and these tasks should be completed using only a single device. Single-camera-based approaches can be widely used in our daily lives because they can ensure the usability of the system. Nevertheless, these approaches often require multi-view scanning [27] or user interventions (e.g., look around, select) for the segmentation of target objects from an image [31,32], making it challenging to support interactions in real-time. Therefore, creating the virtual proxies at runtime is still challenging for usability in the AR system. Our proposed method does not need any spatial sensors, multi-view scanning, or user intervention while creating a virtual proxy.

2.2. Deep Learning-Based Segmentation

To create virtual proxies that replace real objects, we exploit the silhouettes of real objects. We plan to use deep learning-based methods to extract the segmentation mask, which is considered as the silhouette of real objects. Multiple research has been focused on semantic and instance segmentation. Mask R-CNN [33], DeepLabv3 [34], and PSPNet [35] are state of the art on this task. The performances of these models are compared on datasets [36,37]. These models give astonishing results but need lots of computing power and are not running in real-time. PSPNet [35] is running at 1 FPS on a laptop equipped with a GPU. Other researchers have focused on real-time segmentation. ESPNet [38] and ENet [39] are targeting the inference speed and the size of the network to run inference in real-time and reduce memory consumption. All these improvements are targeting the architecture of deep networks to make them more efficient, run in real-time and conserve the highest accuracy possible. However, the above research used a high-performance platform, and it is still not possible to achieve fast segmentation on commercial smartphones or tablets. Thus, we developed a practical and lightweight network that can segment real objects from commercial smartphones. The following sections describe our methods.

3. The Proposed Method

We present a practical method to create a silhouette mesh as a virtual proxy to provide realistic interactions. To extract the real object's silhouette, we developed the segmentation network that works on the mobile device equipped with a single RGB camera. When we pass a single camera feed through the network, our system can build a silhouette mesh corresponding to the real object and place it on the 3D space with the projection method. Our system then uses the silhouette mesh for the physical–virtual interactions instead of pre-modeled virtual proxies. To demonstrate our interactions, we applied our proposed method to the scenario in which a virtual human interacts with deformable animal dolls. The following subsections describe each step in detail.

3.1. Extract the Silhouettes from Interactable Real Objects

Our target AR platform is mobile devices equipped with a single RGB camera. To have seamless interactions, an AR system needs to understand the surrounding space (e.g., to recognize objects and their positions). Thus, we implemented a fast object segmentation network that uses images from the camera of mobile devices. Our network can recognize and segment multiple real objects at high speed with only the smartphone computational resources. Below we describe the dataset we created to train our network and detailed network configurations to improve the performances.

3.1.1. Animal DOLLS Dataset

We aim to implement an AR scenario where a user plays with toys. A virtual human will appear in the AR space and interact with the real toys as a playmate. For convenience's sake, here we used a small set of animal dolls as interactable real objects. However, we believe the method described here can be applied to other types of objects, such as exhibits in a museum for an interactable AR curation.

Dataset creation: We prepared seven animal dolls with different appearances. The animals we chose were a jaguar, zebra, giraffe, wallaby, panda, deer, and a fox (see Figure 2). From this set of dolls, we shoot an average of 25 videos lasting 45 seconds per animal. For each video, we varied the position of the doll, the background selection, the lighting condition, and the appearance of the animal in the video by walking around, dynamically zooming in/out, and rotating the camera. These choices improve the diversity in terms of positions and angles within the dataset and ensure the seamless interaction with deformable real objects. For annotation and labeling, we used the Interactive Video Object Segmentation model from [40] to extract the ground truth masks. Then, we extracted the frames to train the segmentation network. We sampled images at a rate of 5 frames per second to avoid duplicated images of the same position or appearance. Our dataset contains 37,990 images and each category of animals has 4856–6109 images. Our dataset is openly available at https://github.com/khseob0715/DollDataset (accessed on 18 March 2021).

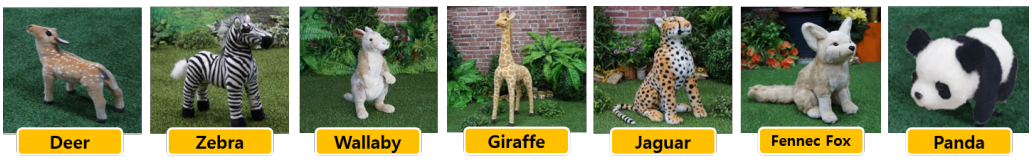


Figure 2. Photo illustration of our dataset with seven animal dolls.

Data augmentation: We applied data augmentation techniques to our dataset before training our segmentation network. We added 10^4 synthesized images into the dataset by (1) replacing the background using the Describable Textures Dataset [41]; (2) rotating images in the range of 360 degrees, to improve the predictability when the mobile device rotates; (3) randomly changing the ratio of the doll area in the image, to mimic different zoom in/out conditions; (4) randomly flipping images on the horizontal and vertical axis. As thus, our network started training with the dataset that was robust for variation.

3.1.2. Recognition and Segmentation Network

In consideration of the RAM and CPU available on commercial mobile devices, we developed a deep neural network compressed from *U*-net [42]. We first reduced the number of filters per channel from 64 to 16 filters. In addition, we considered the input size of the network which directly influences the computation cost of the network. To find an optimal input size, we compared the performances of the network for various input sizes (see Table 1). We calculated pixel accuracy (PA) and mean intersection over union (mIOU) score as the model accuracy measures in addition to the inference speed. As shown in Table 1, the larger the input size, the higher the accuracy the network achieved. However, we chose the input size of 192 pixels which achieved a comparable level of accuracy to the original *U*-net while keeping the inference time below 500 ms. As a result, the number of parameters was reduced to 2.4 million from 7.76 million (cf. [42]). Finally, we trained the network for 5 epochs using *Adam* [43] optimizer with a learning rate started from 10^{-4}, and was divided by 10 after every two epochs. Figure 3 shows the recognition accuracy of our dataset. We trained the model with TensorFlow considering portability to mobile platforms.

Table 1. Performance of our network implanted on Note 9 with different input sizes.

Input Size	Inference Time	IOU	PA
128 pixels	250 ms	0.935	0.973
192 pixels	465 ms	0.971	0.988
224 pixels	820 ms	0.973	0.990
256 pixels	1310 ms	0.977	0.991
384 pixels	2840 ms	0.980	0.994

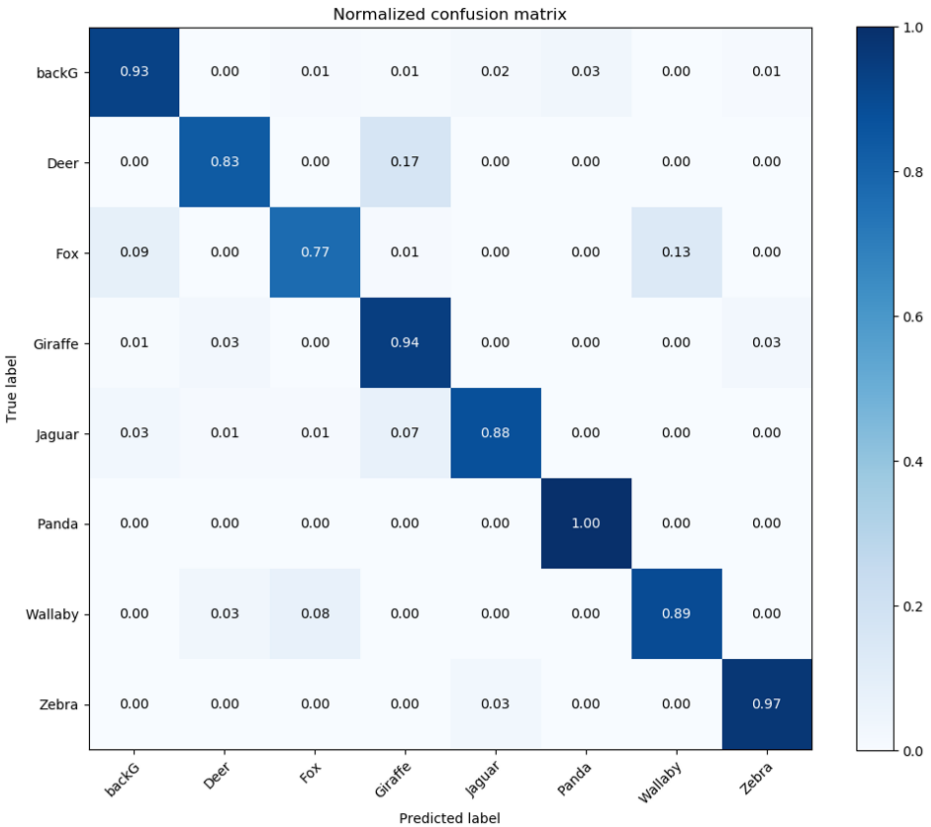


Figure 3. Recognition accuracy with seven animal dolls.

3.2. Generating the Silhouette Mesh

We describe how we create a virtual proxy to replace a recognized real object and how to register the virtual proxies into the 3D space for realistic interactions. Our deep learning network generates a segmentation mask that is considered as the real object's silhouette. We build a polygonal surface from the segmentation mask and deploy it in real space. This polygonal surface—we call this type of mesh the silhouette mesh—plays a role as an interactable virtual proxy. We place the silhouette mesh in real space in a way that the mesh overlaps its associated object from the user's perspective. Instead of reconstructing volume, we use the slope of the device to tilt the silhouette mesh when placing the mesh in the 3D space so that the silhouette mesh takes the corresponding area on the floor (see Figure 1c). Our generation process is in the following order:

Step 1–2D segmentation: Our process starts with the 2D projection of the segmentation results on the screen plane (see Figure 4 Segmentation) On the segmentation result, we retrieve the connected components of pixels, then each component is associated with an object instance. If the number of pixels for an instance is over 500, we create a bounding box around the instance (see Figure 4 Recognition). Concurrently, we also create the virtual surface mesh from the detected floor plane using ARCore [44].

Step 2–2D vertices connection: If there are associated instances for creating silhouette meshes, we used a perspective camera to cast a ray from the bounding box's bottom-midpoint to the virtual surface. If the ray collides with the virtual surface, we generate the 2D vertices along the borderline of the instance using the contour tracing algorithm [45] as well as the sequence of connections between the vertices using Delaunay's triangular algorithm [46] (see Figure 4 Vertex connection). We then created a 2D polygon mesh on the image plane using the 2D vertices and connection sequence. We also calculated the distance from the camera to the collision point, which we will use in the next step as a collision distance.

Step 3–Convert 2D vertices to 3D: From the previous steps, we have the 2D polygon mesh of the recognized object on the image plane. We also have the camera position and the virtual surface in the 3D coordinate system. To convert vertices on the 2D silhouette to 3D vertices in the 3D coordinate system, we cast rays to each vertex on the 2D silhouette mesh. Then using the distance between the camera to the collision point, calculated in the previous step, we determine the positions of each 3D vertice. As a result of this step, we have a set of 3D vertices that have an equal distance from the camera position in the 3D coordinate system (see Figure 4 Convert vertex to 3D).

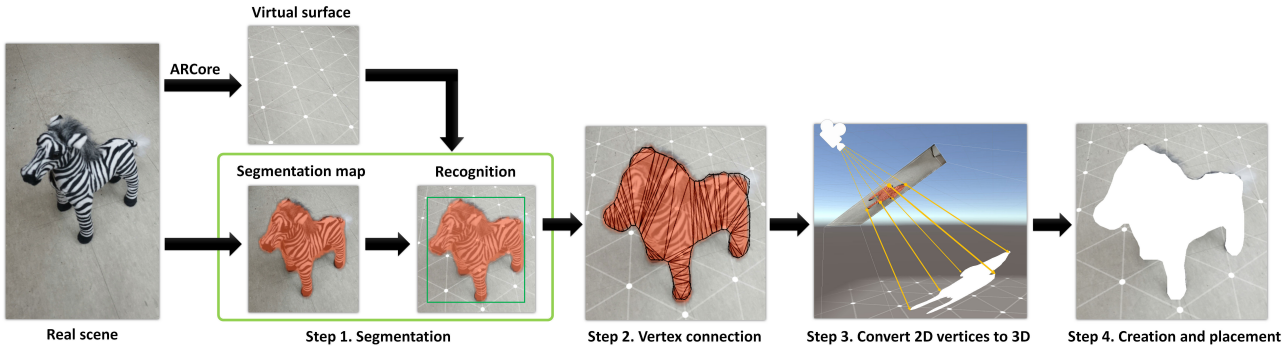


Figure 4. Overview of our silhouette mesh creation process. First, we detect the floor from the phone camera and place the virtual surface on it. At the same time, images from the camera are fed into our network to segment and recognize the interactable object. Based on the segmented region, bounding boxes are generated. For each object, we create vertices along the boundary of the segmented area on the image plane. Then we cast a ray from the camera toward the midpoint of the bottom line of the bounding box and we calculate the distance between the camera and the collision position on the virtual surface. Similarly, we cast rays toward all vertices on the image plane and set the 3D vertices that have equal distance with regard to the camera. Finally, the silhouette mesh is formed (see Section 3.2).

Step 4–Creation and placement: We create the silhouette mesh object using the 3D vertices along with the order of connection obtained in the previous steps. Since the 3D vertices have an equal distance from the camera, the created silhouette mesh and the device that embeds the camera have the same inclination angles. This incline ensures the volume that the real object occupies in the real space, and is used to define the walkable areas for the virtual human. Lastly, the collision point on the virtual surface detected in step 2 approximates the real object's position in the 3D space. Thus we place the silhouette mesh accordingly with the parameters (see Figure 4 Placement).

We underline that during each procedure the user was not required to intervene (e.g., look around, select real objects [31,32]). All procedures take around 25 ms in a mobile device, more results in evaluation Section 4.

3.3. Updating the Silhouette Mesh

For continuous and seamless interactions with a recognized real object, e.g., following a moving zebra doll (see Figure 1b), our AR system needs to distinguish the interactable objects from others and track the target object's location in real-time. In this part, we describe how our system handles it.

Figure 5 shows the detailed process of object updating in our system. Our system takes images of real scenes from the camera on the device and recognizes real objects through the segmentation network. As described in Step 2 of the previous section, we then check whether to create the silhouette mesh for each recognized object. Once the system proceeds to the silhouette mesh creation, we first search for the existing silhouette meshes with the animal class of the newly recognized object. If there exists a silhouette mesh with the same animal class, we further examine the distance between the existing mesh to the new one. If the distance is small, we consider the user is looking at the same real object, and update the parameters of the previously created silhouette mesh. On the other hand, if the user sees another object, we simply place the newly created silhouette mesh in the 3D space. Finally, the current pipeline ends, and the same pipeline is executed repeatedly. The proposed pipeline ran on under 500 ms for all mobiles devices we tested (see Table 2). It should also be noted that the pipeline runs independently from the rendering process, thus it does not decrease rendering speed, keeping the frame rate at around 30 fps.

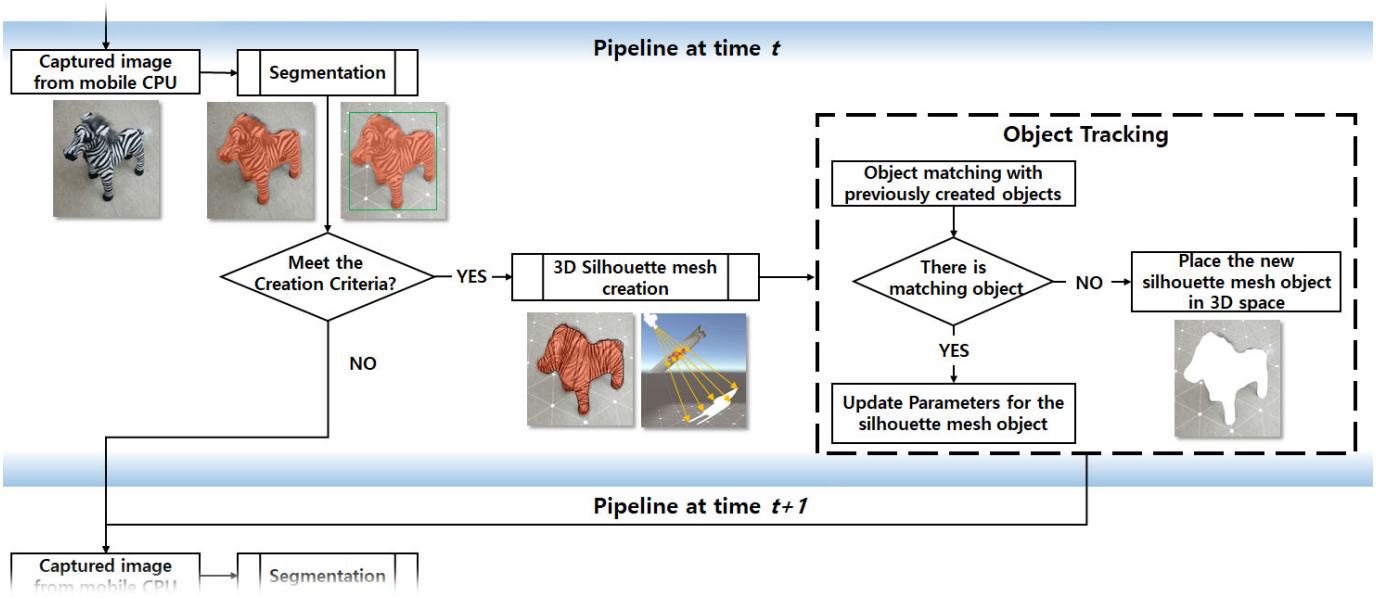


Figure 5. Overview of our system's main pipeline and detailed silhouette mesh update process.

Table 2. Processing time comparison on different mobile device performed with median average on 100 trials.

Device Name	Image Processing	Segmentation	Silhouette Mesh
Galaxy Note 10	35 ms	347 ms	21 ms
Galaxy Tab s5e	39 ms	352 ms	23 ms
Galaxy Note 9	42 ms	465 ms	21 ms
Galaxy S 10	44 ms	572 ms	26 ms

4. Performance Evaluation on Mobile Devices

We present the performance evaluation of the silhouette mesh creation process conducted on various mobile devices. The devices used in the evaluation were Galaxy Note 10, Galaxy Tab s5e, Galaxy Note 9, and Galaxy S10. Our method was implemented using Unity *2017.04.10f1* and ARCore SDK for Unity *1.5.0*.

For the evaluation, we set a room with a zebra doll placed on the floor. A tripod was used to keep the position of the device during the evaluation. For each device, we mounted the device on the tripod and executed our AR application for three minutes.

We measured image processing, segmentation, and silhouette mesh creation time respectively. The Image processing includes image acquisition and the conversion to the input format of the segmentation network. In detail, our process accesses the CPU of the device and obtains the raw image data using ARCore API. The raw image data is in the YUV-420–888 format with 640 × 480 pixels. This raw data is converted to the input format of the segmentation network model—RGB with 192 × 192 pixels. Up to here is the pre-processing in our evaluation. Once the converted data is fed into the segmentation module, the segmentation network runs on the mobile CPU to generate a segmentation map and classes of the recognized objects in the scene. Next, our system executes the 3D silhouette mesh creation procedure using the results from the segmentation module.

The results of our performance evaluations are summarized in Table 2. On all tested devices, our entire process ran below 600 ms (roughly equivalent to about 20 frames). In the meantime, the positions of the created silhouette meshes are maintained and updated, as the camera position updated by ARCore. Thus, the camera rendering speed is maintained on average 30 fps.

Pre-processing and segmentation times are not affected by the number of objects in the scene and the effect of the increased number of objects in silhouette creation time is insignificant. Moreover, our segmentation network is robust to the deformation of the object shape (Figure 6a) and works even with a complex background (Figure 6b). Note that we only used mobile CPU to achieve the reported performance, which means that GPU is still available to render a complex virtual environment. Thus, our silhouette mesh-based method has the potential to be integrated into various AR applications that need realistic interactions between real and virtual objects.

(**a**) (**b**)

Figure 6. Photo illustrations of our silhouette meshes generated from a single camera feed. Our proposed method is robust even in various orientation and deformable object (**a**), as well as in the complex background (**b**).

5. Case Study: Users Play with Animal Dolls in Mobile AR Application

We demonstrate realistic interactions between a virtual human and real animal dolls using the silhouette mesh. Our virtual human is equipped with voice chat [3], gesture animation [47] to provide interactive scenarios (see Figure 7). While playing with the dolls, the virtual human will point or approach a doll to draw the user's attention, and occasionally exhibit the behavior of touching the doll plausibly to make the experience more appealing. Moreover, when near the dolls, the virtual human should also be realistically occluded, and avoid collisions as if he is actually in the real space. In the following subsections, we present how we realized such realistic interactions using our silhouette mesh and a pilot study to evaluate user perception for realistic interactions.

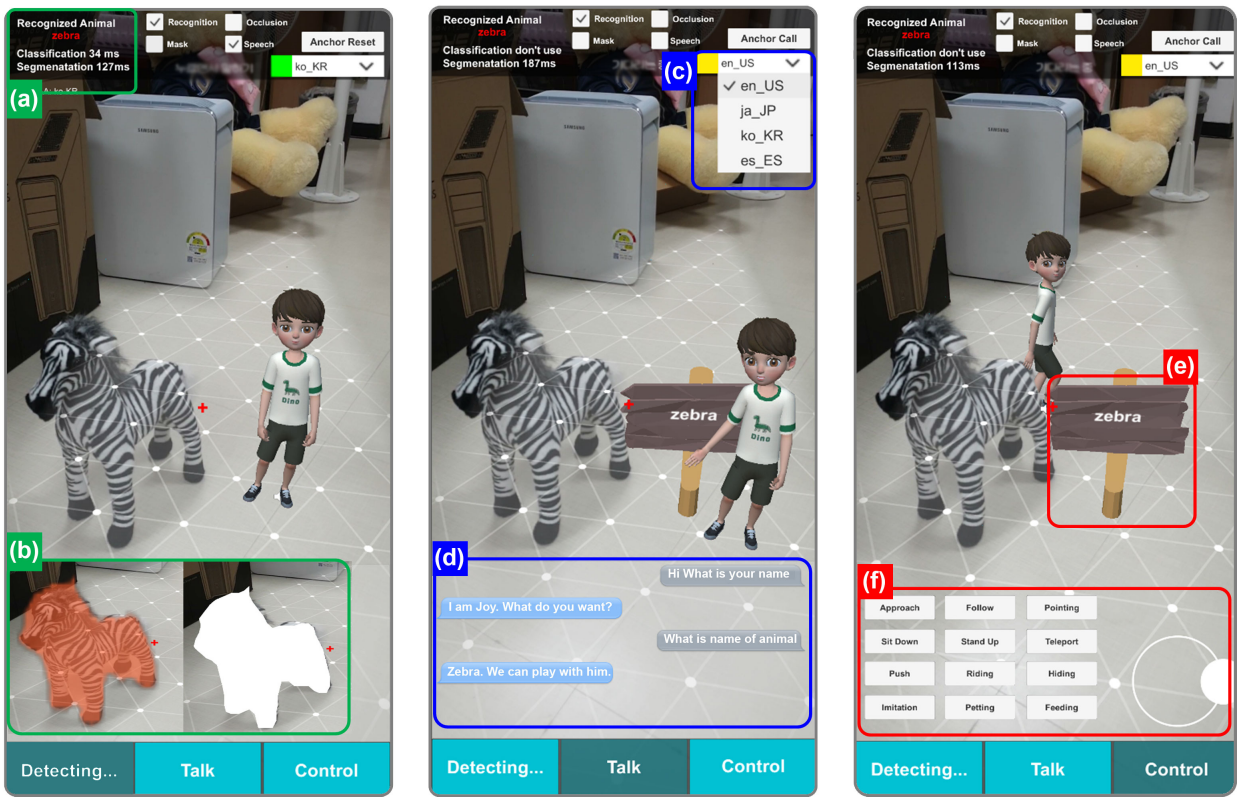


Figure 7. Overview of our multimodal mobile AR system. (**a**) Recognizing real objects; (**b**) displaying the segmentation mask and placing the silhouette mesh; (**c**) supporting multi-language; (**d**) Voice chat between a user and a virtual human based on knowledge database [3]; (**e**) Placement of persistent anchor based on spatial mapping [3]; and (**f**) realistic interactions with real animal dolls using the silhouette mesh.

5.1. Finding, Pointing, and Approaching

When a user provides information about a specific animal in a set of dolls, the virtual human points at the target object with his finger for clarity. Or the virtual human needs to find an animal doll the user is asking for and approach the doll accordingly. For such interactions, we need to know the position of the target animal doll, the type of the animal, as well as unique identification if the target doll is in a group of the same type. Our silhouette mesh includes all of the information required. We first identified the target doll by the gaze position of the user or speech recognition [3]. We used ray-casting from the camera, toward the forward-direction of the user's gaze position, and a custom speech recognition module. When the ray hit a silhouette mesh, we retrieved the unique identification, position, and type of animal doll to carry out appropriate actions, e.g., pointing and approaching (see Figure 8). In subsequent scenes, we used the identification of the target doll to retrieve other information. Thus, the virtual human could follow the

target animal doll. In the case of speech recognition, we used the animal type as keywords (e.g., name) to search the target doll in the scene.

(**a**)

(**b**)

Figure 8. Realistic interactions between a virtual human and real animal dolls by using the silhouette meshes. (**a**) Approaching and following interaction: the virtual human follows the moving zebra doll using the object's location updated in real-time; (**b**) pointing, pushing, and riding interaction: the virtual human uses the gaze point to select the target object and perform interactions.

5.2. Walking among Animal Dolls

When approaching or following a target animal doll, the virtual human should not pass through the other dolls (i.e., collision avoidance). We made the virtual human walk around the other dolls to avoid such an unnatural interaction. As mentioned in Section 3.2, we define the walkable areas for the virtual human on the virtual surface (made with the Unity game engine). By projecting the silhouette meshes onto the virtual surface, we defined the areas that the virtual human should not walk over because it is an area occupied by real objects. Thus, the walkable area integrates holes, at the positions of dolls on the virtual surface (see Figure 1c). We set the locomotion position of virtual humans within the walkable areas near the target doll for interaction. Then the virtual human finds the path to the target position and walks along the path, avoiding collisions with other dolls.

5.3. Touching Animal Dolls

The silhouette mesh allows us to implement plausible interactions between the virtual human and the dolls, such as petting, pushing, and riding (see Figure 8b). We grouped them as touch interaction as a part of the virtual human's body seemed to contact the target doll. For petting and pushing, the virtual human approached near the target silhouette mesh and stood next to it. We chose the standing position of the virtual human slightly closer toward the camera from the center position of the un-walkable areas defined by the silhouette mesh. Then, we played pushing and petting animations, respectively. For riding, the virtual human jumped to the top position of the silhouette mesh. Although this method is relatively simple, from the user's perspective, the virtual human seemed to interact with the real animal doll.

5.4. Mutual Occlusion

Most AR applications simply overlay the virtual objects on top of the real scene. Such a method might be useful when overlaying information in a scenario where the visibility of the information is crucial. However, for the illusion of a virtual object existing in the real world, mutual occlusion between the real and virtual objects should be provided [48]. In the case of a virtual human walking among real animal dolls, occlusion should be applied to the virtual human when behind dolls. Conversely on dolls, when the virtual human is in front of them. Since we created our silhouette mesh from the real object rendered on the image plane, the shape of the silhouette is matched with the shape of the real object on the image plane. We also placed all silhouette meshes at the corresponding positions of real objects. Therefore, by applying a texture that excludes the virtual object's layer to silhouette mesh, we could achieve a realistic mutual occlusion.

Figure 9 shows results of three different occlusion strategies. As seen in the first image, a simple overlay of the virtual object on the real scene, for example, used in Pokémon Go, may result in spatially incorrect placement of the virtual object. Figure 9b simulates the occlusion based on object detection without segmentation. Most object detection algorithms return such bounding boxes around recognized objects, which might be faster compared to generating a segmentation map as our method does. However, in this case, the virtual human can be incorrectly cropped in midair even if the spatial relation between virtual and real might be kept. On the other hand, since our silhouette mesh has the shape of the real object, it can provide a delicate mutual occlusion for sophisticated objects.

(**a**) no occlusion (**b**) bounding box (**c**) segmented mask

Figure 9. Different types of strategies for applying occlusion.

5.5. Pilot Study

To evaluate the silhouette mesh's benefits and realistic interaction offered by our system, we conducted a video-based pilot study using an online survey due to the COVID 19 situation. We compared silhouette mesh (SM) against two conditions, i.e., No occlusion (NO) and object bounding box (BB) to explore the user's perception of the virtual human (see Figure 9). Each condition was evaluated on five aspects: Spatial Awareness, Spatial Presence, Perceived Naturalness, Object occlusion, and Physical Ability. Spatial awareness evaluates the relative size and position of virtual humans and dolls. Spatial presence evaluates how well the model fits in the real world. Perceived naturalness evaluate the naturalness of the movement of virtual human between dolls. Object Occlusion evaluates the usefulness of silhouette mesh as a proxy. Physical ability evaluate the user perception of virtual human's awareness of objects. For each condition, we prepared interaction videos in three situations, i.e., there are a total of nine video clips (see Figure 10). The videos were captured from a fixed view and the virtual human walked around the dolls to interact with them.

5.5.1. Participants

We recruited 30 participants from a local community. However, we omitted data from 6 participants who did not pass our screening criteria; 1 by trick questions and 5 by short survey time. Thus, we had 24 valid participants (7 females, 17 males, age 20–31, M = 26.04, SD = 3.12) for the analysis. Participants had a low familiarity with AR (M = 2.62,

SD = 0.87), measured on a five-point Likert scale (1: Not at all, 2: Slightly, 3: Moderately, 4: Knowledgeable, and 5: Expert). Lastly, 13 of them had a previous experience of participating in an AR experiment.

(**a**) one small doll (**b**) two small dolls (**c**) one large doll

Figure 10. Photo illustrations of interaction scenarios with the silhouette meshes applied. (**a**) One doll that is similar in size to a virtual human, (**b**) two dolls that are similar in size to a virtual human, and (**c**) one doll that is larger than a virtual human.

5.5.2. Procedures and Measurements

We prepared an online survey consisting of five pages. The first page was a brief instruction for our study, followed by a pre-questionnaire (e.g., demographics, AR familiarity). For the next three pages, each page contained three video clips for each condition, followed by a questionnaire that was only visible after the participant finished watching the video. The order of conditions was predefined in a counter-balanced manner, and the order of questions was randomized. The questionnaire comprises 15 questions, three for each of the five aforementioned aspects, which we extracted and modified from the previous work [7,49]. The order of the questions was provided randomly, and all questions were answered on a seven-point Likert scale. On the last page, we asked the participants to leave a comment. Below is our questionnaire.

- **Spatial Awareness**
  - Q1 *I was able to imagine the arrangement of virtual agents with regard to the physical space very well.*
  - Q2 *I was able to make a reasonable estimate of the virtual agent's size regarding the presented space.*
  - Q3 *I was able to estimate how far apart the virtual agent and the doll were from each other.*
- **Spatial Presence**
  - Q4 *I felt like virtual agent was a part of the environment in the presentation.*
  - Q5 *I felt like virtual agent was actually there in the environment with the doll.*
  - Q6 *I felt like virtual agent was physically present in the environment of the presentation.*
- **Perceived Naturalness**
  - Q7 *I thought virtual agent movement/behavior in real space seemed natural.*
  - Q8 *I thought the virtual agent was naturally moving around to avoid a doll.*
  - Q9 *I thought virtual agent looked natural when it overlapped with a doll.*
- **Object Occlusion**
  - Q10 *I felt that virtual agent was properly occluded behind the doll.*
  - Q11 *I thought part of a virtual human body is occluded the same as the silhouette of a doll.*
  - Q12 *I thought virtual agent's appearance was out of harmony with the background space/things.*
- **Physical Ability**
  - Q13 *I felt that virtual agent could not pass through a doll.*
  - Q14 *I felt that virtual agent could touch a doll.*
  - Q15 *I felt virtual agents were well aware of the space around the doll's front and back sides.*

5.5.3. Results

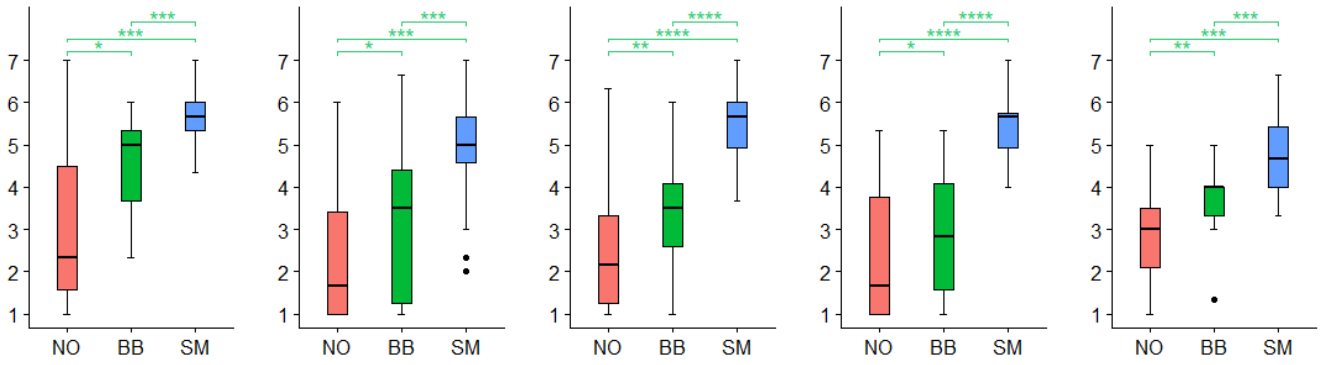
We report statistical analysis results (see Figure 11, Table 3). Considering the ordinal aspect of our measures, we decided to perform non-parametric Friedman tests on the measures at the 5% significance level. For the pairwise comparisons, we performed the Wilcoxon signed-rank tests with Bonferroni adjustment.

Our results show significant differences in all aspects and all conditions (see Figure 11, Table 3). As in the previous study [50], participants reported the lowest score for the NO condition in which there is no occlusion by real objects. Although all aspects have been slightly improved under the BB's condition, participants reported that they felt awkward due to the poor occlusion in the following comments:

> *P11: "BB looks like a person passing behind a square because the doll is cut less, but this can be seen as an error."*
>
> *P17: "BB is unnatural, but it can still help me in deciding whether the virtual human is behind or in front of the doll."*

Under our proposed SM condition, we supported occlusion in the same shape as real objects (see Figure 10), so participants felt high satisfaction with occlusion and perceived that the virtual human's behavior was natural. In addition, by well-providing depth perception between a virtual human and dolls [51], the spatial awareness and presence of the virtual human have significantly increased. The virtual human's physical ability increased because they followed physical rules that physical objects cannot pass through real dolls [7,52]. Thus our findings reveal that the method we propose has positive impacts on user perception of AR content using the mobile platform.

However, due to video observation procedures, our results are limited, and it remains to be seen whether such results will still hold in real use cases. Thus we plan to conduct a face-to-face human subject study involving dynamic views with user control.



(**a**) Spatial Awareness (**b**) Spatial Presence (**c**) Perceived Naturalness (**d**) Object Occlusion (**e**) Physical ability

Figure 11. Results of subjective measures on three conditions: No occlusion (NO), Bounding box (BB), and Silhouette mesh (SM). Statistical significance levels: * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$, and **** $p < 0.0001$.

Table 3. Summary of statistical analysis results.

Measure	Tests		
	Cronbach	**Friedman**	**Post-Hoc**
Spatial Awareness	$\alpha = 0.75$	$\chi^2 = 26.58$, $p < 0.0001$	NO < BB * NO < SM *** BB < SM ***
Spatial Presence	$\alpha = 0.87$	$\chi^2 = 29.30$, $p < 0.0001$	NO < BB * NO < SM *** BB < SM ***
Perceived Naturalness	$\alpha = 0.73$	$\chi^2 = 34.69$, $p < 0.0001$	NO < BB ** NO < SM **** BB < SM ****
Object Occlusion	$\alpha = 0.73$	$\chi^2 = 42.75$, $p < 0.0001$	NO < BB * NO < SM **** BB < SM ****
Physical Ability	$\alpha = 0.67$	$\chi^2 = 30.78$, $p < 0.0001$	NO < BB ** NO < SM *** BB < SM ***

Statistical significance levels: * $p < 0.05$; ** $p < 0.01$; *** $p < 0.001$; **** $p < 0.0001$.

6. Discussion

Once *Pokémon GO* (Niantic, 2016) was introduced and hit, people in the entertainment industry, especially the game community, expected rapid and remarkable growth of the AR market. However, due to the lack of AR content that can capture public attention, AR users significantly decreased. Among factors that possibly caused the public to turn away from the AR application, we addressed the followings in this paper; namely, incorrect registration, unnatural interaction with real objects, and distorted spatial sense. Providing seemingly realistic interactions require labor-intensive 3D modeling [13] as well as special hardware devices [27], such as depth camera, IR tracker. Such requirements might also be the reason that hinders AR contents from being widely used in our daily lives [6,7].

The presented AR system consisted of several modules, including recognition, segmentation, silhouette mesh, chatbot, and virtual human/objects, all running on a single mobile device (see Figure 7). We adopted the modular design so that each part can be independently improved without affecting the rest of the system. Through the user studies, we demonstrated the potential entertainment use of our system for both children and adults. Due to the modular design and the simplicity, we believe that the proposed method can easily adapt to different scenarios and real objects in the entertainment industry, e.g., botanical garden scenario, living room, playground, museum, etc.

Despite such promising results of our system, there are two main issues in this work: (1) the limitations of the online-based pilot study, (2) the need for a pre-trained segmentation model. In the following subsections, we discuss each issue in detail, present potential solutions, and future research directions.

6.1. Limitation 1: Limited Pilot Study

Due to the unprecedented pandemic situation, in-person and lab-based typical human-subjects experiments were out of option until now. Thus, we collected our data through the online video-based survey for this study. One might raise the question of the validity of the online video-based data collection for user studies conducted in this paper. In addition, compared to the existing researches that collected data via online surveys, the number of participants in our survey seemed a bit small [53–55].

Thus, we plan to extend our present study to a face-to-face experiment, where participants actually experience the AR contents in person instead of observing pre-recorded video clips. We will compare the results of the online-based survey presented in this paper with the data collected through the face-to-face experiment to confirm the validity of the video-based survey and/or to see how user's perception differs. This further allows us to objectively measure the engagement levels of participants during the AR experience by, for example, collecting (1) usage time, (2) number of interactions (voice or touch), (3) number of errors, and (4) task load, etc.

6.2. Limitation 2: Pre-Trained Segmentation Model

The main limitation of our method stems from the fact that the segmentation model was trained in advance. The role of the segmentation model was to extract the silhouette of the real object to prepare a virtual proxy. Thanks to the modular design, the segmentation model can be easily replaced with a newer one with more objects trained; however, the system will fail to give an appropriate AR experience if a user tries to interact with untrained real objects.

In such a case, we can consider a minimum intervention of the user for unsupervised learning [56]. For example, the system could ask users to mark the rough boundaries of the untrained objects in the scene from time to time [31]. This additional operation may negatively affect the user experience; however, it also could prevent the user experience degradation from the failure to generate appropriate silhouettes of the untrained objects.

Another solution would be the use of an on-demand cloud API for the segmentation model. This solution requires users to provide pictures of objects they want to interact with beforehand. It may seem similar to replacing the pre-trained segmentation model;

however, this direction would not require any programming knowledge so that layman could perform without any issues.

7. Conclusions

In this paper, we proposed the silhouette mesh, a practical method that allows virtual objects to perform realistic interactions with real objects without pre-modeled virtual proxies. Our silhouette mesh creation requires no user intervention, even on deformable real objects, and it uses only a single mobile device. In our case study, we prepared the animal dolls dataset to train the segmentation network. We then applied it to the multimodal-based AR system to demonstrate seamless and realistic interaction between the virtual human and real-world objects. The result of our pilot study indicated that our approach, silhouette mesh, significantly improves the user experience and perception with the virtual human in the physical–virtual interaction in the shared space. Although our work has limitations and has realized only four types of interactions to pursue the case study, our approach can be used in touch-driven AR contents to interact with real-world objects and extend the interaction type depending on the AR scenarios. We envision that our work has given a promising glimpse into one of the ways to ensure usability and robust interactions with natural objects in the single-based mobile AR platforms.

Author Contributions: Conceptualization, H.K., M.L., and J.-I.H.; formal analysis, H.K., A.P.; funding acquisition, J.-I.H.; investigation, H.K., G.A., A.P., and M.L.; methodology, H.K., A.P.; software, H.K., G.A.; data curation, H.K., A.P.; supervision, G.J.K., J.-I.H.; writing—original draft, H.K.; writing—review and editing, all author; All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Council of Science and Technology (NST) Grant by the Korean Government through the Ministry of Science and ICT (MSIT) (CRC-20-02-KIST).

Institutional Review Board Statement: This work was involved human-subjects. However, our study is an anonymous web survey and not a face-to-face interview. Thus, ethical review and approval was not required for the study on human participants in accordance with the local legislation and institutional requirements.

Informed Consent Statement: Consent form was not provided because this experiment was conducted through an anonymous web survey.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: We express sincere gratitude to the participants in the experiments. We also thank the reviewers for their valuable contributions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Azuma, R.T. A Survey of Augmented Reality. *Presence Teleoperators Virtual Environ.* **1997**, *6*, 355–385. [CrossRef]
2. Keil, J.; Edler, D.; Dickmann, F. Preparing the HoloLens for user studies: An augmented reality interface for the spatial adjustment of holographic objects in 3D indoor environments. *KN J. Cartogr. Geogr. Inf.* **2019**, *69*, 205–215. [CrossRef]
3. Ali, G.; Le, H.Q.; Kim, J.; Hwang, S.W.; Hwang, J.I. Design of Seamless Multi-modal Interaction Framework for Intelligent Virtual Agents in Wearable Mixed Reality Environment. In Proceedings of the 32nd International Conference on Computer Animation and Social Agents, Paris, France, 1–3 July 2019; pp. 47–52. [CrossRef]
4. Sanches, S.R.R.; Tokunaga, D.M.; Silva, V.F.; Sementille, A.C.; Tori, R. Mutual occlusion between real and virtual elements in Augmented Reality based on fiducial markers. In Proceedings of the 2012 IEEE Workshop on the Applications of Computer Vision (WACV), Breckenridge, CO, USA, 9–11 January 2012; pp. 49–54. [CrossRef]
5. Kruijff, E.; Swan, J.E.; Feiner, S. Perceptual issues in augmented reality revisited. In Proceedings of the 2010 IEEE International Symposium on Mixed and Augmented Reality, Seoul, Korea, 13–16 October 2010; pp. 3–12.
6. Kim, H.; Lee, M.; Kim, G.J.; Hwang, J.I. The Impacts of Visual Effects on User Perception With a Virtual Human in Augmented Reality Conflict Situations. *IEEE Access* **2021**, *9*, 35300–35312. [CrossRef]

7. Kim, H.; Kim, T.; Lee, M.; Kim, G.J.; Hwang, J.I. Don't Bother Me: How to Handle Content-Irrelevant Objects in Handheld Augmented Reality. In Proceedings of the 26th ACM Symposium on Virtual Reality Software and Technology, Ottawa, ON, Canada, 2–4 November 2020; pp. 1–5. [CrossRef]
8. Kim, K.; Maloney, D.; Bruder, G.; Bailenson, J.N.; Welch, G.F. The effects of virtual human's spatial and behavioral coherence with physical objects on social presence in AR. *Comput. Animat. Virtual Worlds* **2017**, *28*, e1771. [CrossRef]
9. Milne, M.; Luerssen, M.H.; Lewis, T.W.; Leibbrandt, R.E.; Powers, D.M.W. Development of a virtual agent based social tutor for children with autism spectrum disorders. In Proceedings of the 2010 International Joint Conference on Neural Networks (IJCNN), Barcelona, Spain, 18–23 July 2010; pp. 1–9. [CrossRef]
10. Gratch, J.; Rickel, J.; Andre, E.; Cassell, J.; Petajan, E.; Badler, N. Creating interactive virtual humans: Some assembly required. *IEEE Intell. Syst.* **2002**, *17*, 54–63. [CrossRef]
11. Matsumura, K.; Sumi, Y. Poster: Puppetooner: A puppet-based system to interconnect real and virtual spaces for 3D animations. In Proceedings of the 2013 IEEE Symposium on 3D User Interfaces (3DUI), Orlando, FL, USA, 16–17 March 2013; pp. 159–160.
12. Held, R.; Gupta, A.; Curless, B.; Agrawala, M. 3D Puppetry: A Kinect-Based Interface for 3D Animation. In Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology, Cambridge, MA, USA, 7–10 October 2012; pp. 423–434. [CrossRef]
13. Lee, Y.; Choi, J. Tideland animal AR: Superimposing 3D animal models to user defined targets for augmented reality game. *Int. J. Multimed. Ubiquitous Eng.* **2014**, *9*, 343–348. [CrossRef]
14. Igarashi, T.; Matsuoka, S.; Tanaka, H. Teddy. In Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques—SIGGRAPH '99, Los Angeles, CA, USA, 8–13 August 1999; ACM Press: New York, NY, USA, 1999; pp. 409–416. [CrossRef]
15. Barakonyi, I.; Psik, T.; Schmalstieg, D. Agents That Talk And Hit Back: Animated Agents in Augmented Reality. In Proceedings of the Third IEEE and ACM International Symposium on Mixed and Augmented Reality, Arlington, VA, USA, 5 November 2004; pp. 141–150. [CrossRef]
16. Taheri, A.; Shahab, M.; Meghdari, A.; Alemi, M.; Amoozandeh Nobaveh, A.; Rokhi, Z.; Ghorbandaei Pour, A. Virtual Social Toys: A Novel Concept to Bring Inanimate Dolls to Life. In *International Conference on Social Robotics*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 286–296. [CrossRef]
17. Sesame Street; Vuforia. Sesame Workshop Prototype Playset Augmented Reality Vuforia. 2016. Available online: https://youtu.be/iR8uGxiX5ak (accessed on 18 March 2021).
18. Desierto, A.J.R. GoonAR: A Bilingual Children Storybook through Augmented Reality Technology Using Unity with Vuforia Framework. *Int. J. Adv. Trends Comput. Sci. Eng.* **2020**, *9*, 3681–3686. [CrossRef]
19. Tsvetkova, I.; Kinaneva, D.; Hristov, G.; Raychev, J.; Zahariev, P. A complex workflow for development of interactive and impressive educational content using capabilities of animated augmented reality trends. In Proceedings of the 2018 17th International Conference on Information Technology Based Higher Education and Training (ITHET), Olhao, Portugal, 26–28 April 2018; pp. 1–7.
20. PTC; Vuforia. Vuforia Engine: How to Create Model Targets. 2020. Available online: https://youtu.be/jbaUDMvv2Zw (accessed on 18 March 2021).
21. Kato, H.; Billinghurst, M. Marker tracking and HMD calibration for a video-based augmented reality conferencing system. Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99). *IEEE Comput. Soc.* **1999**, 85–94. [CrossRef]
22. Fujimoto, Y.; Smith, R.T.; Taketomi, T.; Yamamoto, G.; Miyazaki, J.; Kato, H.; Thomas, B.H. Geometrically-Correct Projection-Based Texture Mapping onto a Deformable Object. *IEEE Trans. Vis. Comput. Graph.* **2014**, *20*, 540–549. [CrossRef]
23. Mueller, F.; Mehta, D.; Sotnychenko, O.; Sridhar, S.; Casas, D.; Theobalt, C. Real-Time Hand Tracking Under Occlusion From an Egocentric RGB-D Sensor. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
24. Tang, X.; Hu, X.; Fu, C.W.; Cohen-Or, D. GrabAR: Occlusion-aware Grabbing Virtual Objects in AR. In Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology, Minneapolis, MN, USA, 20–23 October 2020; pp. 697–708.
25. Azuma, R.T. Making Augmented Reality a Reality. In *Imaging and Applied Optics 2017 (3D, AIO, COSI, IS, MATH, pcAOP)*; Optical Society of America, OSA: Washington, DC, USA, 2017; p. JTu1F.1. [CrossRef]
26. Kim, M.J. A framework for context immersion in mobile augmented reality. *Autom. Constr.* **2013**, *33*, 79–85. doi:10.1016/j.autcon.2012.10.020. [CrossRef]
27. Prisacariu, V.A.; Kahler, O.; Murray, D.W.; Reid, I.D. Real-Time 3D Tracking and Reconstruction on Mobile Phones. *IEEE Trans. Vis. Comput. Graph.* **2015**, *21*, 557–570. [CrossRef]
28. Runz, M.; Buffier, M.; Agapito, L. MaskFusion: Real-Time Recognition, Tracking and Reconstruction of Multiple Moving Objects. In Proceedings of the 2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Munich, Germany, 16–20 October 2018; pp. 10–20. [CrossRef]
29. Ozawa, T.; Nakajima, Y.; Saito, H. Simultaneous 3D Tracking and Reconstruction of Multiple Moving Rigid Objects. In Proceedings of the 2019 12th Asia Pacific Workshop on Mixed and Augmented Reality (APMAR), Ikoma, Japan, 28–29 March 2019; pp. 1–5. [CrossRef]

30. Newcombe, R.A.; Izadi, S.; Hilliges, O.; Molyneaux, D.; Kim, D.; Davison, A.J.; Kohli, P.; Shotton, J.; Hodges, S.; Fitzgibbon, A. KinectFusion: Real-time dense surface mapping and tracking. In Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2011, Basel, Switzerland, 26–29 October 2011; pp. 127–136. [CrossRef]
31. Bastian, J.; Ward, B.; Hill, R.; van den Hengel, A.; Dick, A. Interactive modelling for AR applications. In Proceedings of the 2010 IEEE International Symposium on Mixed and Augmented Reality, Seoul, Korea, 13–16 October 2010; pp. 199–205. [CrossRef]
32. Lepetit, V.; Berger, M.O. A semi-automatic method for resolving occlusion in augmented reality. Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662). *IEEE Comput. Soc.* **2000**, *2*, 225–230. [CrossRef]
33. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 386–397. [CrossRef] [PubMed]
34. Chen, L.C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking Atrous Convolution for Semantic Image Segmentation. *Comput. Vis. Pattern Recognit.* **2017**. Available online: http://arxiv.org/abs/1706.05587 (accessed on 18 Mach 2021).
35. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid Scene Parsing Network. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6230–6239. [CrossRef]
36. Everingham, M.; Eslami, S.M.A.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes Challenge: A Retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136. [CrossRef]
37. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The Cityscapes Dataset for Semantic Urban Scene Understanding. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 3213–3223. [CrossRef]
38. Mehta, S.; Rastegari, M.; Caspi, A.; Shapiro, L.; Hajishirzi, H. ESPNet: Efficient Spatial Pyramid of Dilated Convolutions for Semantic Segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 November 2018.
39. Paszke, A.; Chaurasia, A.; Kim, S.; Culurciello, E. ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation. *arXiv* **2016**, arXiv:1606.02147.
40. Oh, S.W.; Lee, J.Y.; Xu, N.; Kim, S.J. Fast User-Guided Video Object Segmentation by Interaction-and-Propagation Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 5242–5251. [CrossRef]
41. Cimpoi, M.; Maji, S.; Kokkinos, I.; Mohamed, S.; Vedaldi, A. Describing Textures in the Wild. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 3606–3613. [CrossRef]
42. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241. [CrossRef]
43. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
44. Google. ARCore. 2020. Available online: https://developers.google.com/ar/discover/supported-devices (accessed on 18 March 2021).
45. Reddy, P.R.; Amarnadh, V.; Bhaskar, M. Evaluation of Stopping Criterion in Contour Tracing Algorithms. *Int. J. Comput. Sci. Inf. Technol.* **2012**, *3*, 3888–3894.
46. Ruppert, J. A Delaunay Refinement Algorithm for Quality 2-Dimensional Mesh Generation. *J. Algorithms* **1995**, *18*, 548–585. [CrossRef]
47. Ali, G.; Lee, M.; Hwang, J.I. Automatic text-to-gesture rule generation for embodied conversational agents. *Comput. Animat. Virtual Worlds* **2020**, *31*, e1944. [CrossRef]
48. Kiyokawa, K.; Kurata, Y.; Ohno, H. An optical see-through display for mutual occlusion with a real-time stereovision system. *Comput. Graph.* **2001**, *25*, 765–779. [CrossRef]
49. Vorderer, P.; Wirth, W.; Gouveia, F.R.; Biocca, F.; Saari, T.; Jäncke, F.; Böcking, S.; Schramm, H.; Gysbers, A.; Hartmann, T.; et al. MEC spatial presence questionnaire (MEC-SPQ): Short documentation and instructions for application. In *Report to the European Community, Project Presence: MEC (IST-2001-37661)*; Online, 2004; Volume 3. Available online: https://www.researchgate.net/publication/318531435_MEC_spatial_presence_questionnaire_MEC-SPQ_Short_documentation_and_instructions_for_application (accessed on 18 March 2021).
50. Kim, K.; Bruder, G.; Welch, G. Exploring the effects of observed physicality conflicts on real-virtual human interaction in augmented reality. In *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology, Gothenburg, Sweden, 8–10 November 2017*; ACM: New York, NY, USA, 2017; pp. 1–7. [CrossRef]
51. Cakmakci, O.; Ha, Y.; Rolland, J. A Compact Optical See-Through Head-Worn Display with Occlusion Support. In Proceedings of the Third IEEE and ACM International Symposium on Mixed and Augmented Reality, Arlington, VA, USA, 5 November 2004; pp. 16–25. [CrossRef]
52. Norouzi, N.; Kim, K.; Lee, M.; Schubert, R.; Erickson, A.; Bailenson, J.; Bruder, G.; Welch, G. Walking your virtual dog: Analysis of awareness and proxemics with simulated support animals in augmented reality. In Proceedings of the 2019 IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2019, Beijing, China, 14–18 October 2019; pp. 157–168.
53. Popovici, I.; Vatavu, R.D. Understanding Users' Preferences for Augmented Reality Television. In Proceedings of the 2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Beijing, China, 14–18 October 2019; pp. 269–278.

54. Haugstvedt, A.C.; Krogstie, J. Mobile augmented reality for cultural heritage: A technology acceptance study. In Proceedings of the 2012 IEEE international symposium on mixed and augmented reality (ISMAR), Atlanta, GA, USA, 5–8 November 2012; pp. 247–255.
55. Knierim, P.; Woźniak, P.W.; Abdelrahman, Y.; Schmidt, A. Exploring the potential of augmented reality in domestic environments. In Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services, Taipei, Taiwan, 1–4 November, 2019; pp. 1–12.
56. Li, Y.; Paluri, M.; Rehg, J.M.; Dollár, P. Unsupervised learning of edges. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1619–1627.