Ubiquitous Virtual Humans: A Multi-Platform Framework for Embodied AI Agents in XR

Arno Hartholt Ed Fast Adam Reilly Wendy Whitcup Matt Liewer Sharon Mozgai

USC Institute for Creative Technologies
Playa Vista, USA
{hartholt, fast, reilly, whitcup, liewer, mozgai}@ict.usc.edu

***Abstract*—We present an architecture and framework for the development of virtual humans for a range of computing platforms, including mobile, web, Virtual Reality (VR) and Augmented Reality (AR). The framework uses a mix of in-house and commodity technologies to support audio-visual sensing, speech recognition, natural language processing, nonverbal behavior generation and realization, text-to-speech generation, and rendering. This work builds on the Virtual Human Toolkit, which has been extended to support computing platforms beyond Windows. The resulting framework maintains the modularity of the underlying architecture, allows re-use of both logic and content through cloud services, and is extensible by porting lightweight clients. We present the current state of the framework, discuss how we model and animate our characters, and offer lessons learned through several use cases, including expressive character animation in seated VR, shared space and navigation in room-scale VR, autonomous AI in mobile AR, and real-time user performance feedback based on mobile sensors in headset AR.**

***Index Terms*—virtual humans, AI, VR, AR, XR, character animation, character modeling, embodied conversational agents, interactive virtual agents**

I. Introduction

Virtual humans – interactive digital representations of humans who can perceive real humans and respond appropriately, both verbally and nonverbally – have seen many beneficial uses in education, medicine, the military, and entertainment [3] [23]. Typically, these systems are deployed on traditional desktop computing platforms. Recently, mobile, web, Augmented Reality (AR) and Virtual Reality (VR)[1] are becoming increasingly viable platforms as both hardware and software continue to advance [32].

To fully explore the possibilities of virtual humans, it is imperative that researchers and developers are able to rapidly create and iterate on characters of sufficient fidelity, with appropriate functionality, and on any computing platform of interest. This is true in particular for AR and VR, given their novel status and potential. To address this challenge, we present an extension of the Virtual Human Toolkit [11] that supports numerous computing platforms, enabling the exploration of their strengths and weaknesses.

II. Background

Virtual humans have been applied to a range of domains, including job interviewing [1] [6], assessment [22] and education [3]. Early signs indicate that virtual humans can provide benefits over human to human interaction related to impression management and reduced perceived bias [21] [4].

Virtual humans have traditionally been bound to the desktop [11], with mobile applications following [3]. Recent VR advances have resulted in much attention [8], although many applications focus on avatars, i.e. digital characters that are embodied by their real human users [19] [29], rather than autonomous characters. Virtual humans in AR are rare. Early R&D prototypes exist, e.g. a health related coach in 2016 [9], but have seen little in-depth attention, with the exception of broad investigations of social interaction [20] and commercial work [16]. However, AR is an important platform with early studies in education indicating increased learning gains and motivation, partly due to increased interactivity [2].

There are numerous relevant technologies available in support of developing virtual humans, from individual capabilities, including speech recognition [25], natural language processing [17], text-to-speech generation [28], and game engines[2] [3], to larger frameworks that cover an integrated subset of required functionality [18].

One of these frameworks is the publicly available Virtual Human Toolkit [11], which is "a collection of modules, tools, and libraries designed to aid and support researchers and developers with the creation of virtual human conversational characters."[4] The Toolkit supports audio-visual sensing [24], speech recognition [12], natural language processing [15], nonverbal behavior generation [14], nonverbal behavior realization [26], text-to-speech generation [28], and rendering[2]. While rich in its integrated capabilities, the Toolkit only supports Windows, which allows support for Desktop VR (e.g., HTC Vive, Oculus Rift, Samsung Odyssey, etc.), but not for mobile VR or AR (e.g., Oculus Quest, HoloLens, ARKit, ARCore, etc.). This paper focuses on expanding the Virtual Human Toolkit to add support for these platforms.

III. Architecture

A. Overview

This work builds on the Virtual Human Toolkit (aka Toolkit). The Toolkit is based on a modular architecture where

[1]Increasingly captured under the umbrella term eXtended Reality (XR).

[2]https://unity.com
[3]https://www.unrealengine.com
[4]https://vhtoolkit.ict.usc.edu

978-1-7281-5604-0/19/$31.00 ©2019 IEEE
DOI 10.1109/AIVR46125.2019.00072

individual modules (e.g., speech recognition, natural language processing, etc.) communicate to each other primarily through message passing using ActiveMQ [30]. The main exception is the communication between the nonverbal behavior realizer and the renderer, given the need to send rotations for each bone in the character skeleton (or characters' skeletons) every frame. Rather than flooding the message bus with this data, the realizer sits within the renderer process as a native DLL.

See Figure 1 for a visual overview of how the modules relate to each other. See [11] for more details.

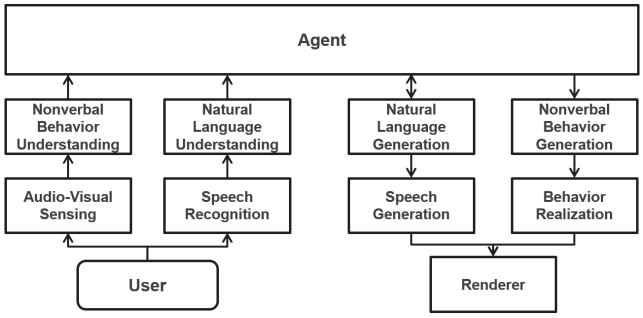

Fig. 1. The Virtual Human Toolkit architecture, adapted from [11].

B. Adding Multi-Platform Support

In adding multi-platform support to the Toolkit (e.g., web, mobile, AR, VR), we had the following requirements:

1) Keep initial development costs low.
2) Avoid creating one-offs for individual platforms.
3) Allow for re-use of both logic and content.
4) Future-proof for upcoming platforms, in particular XR.
5) Keep required maintenance effort manageable.

The modules within the Toolkit are developed using several programming languages, including C#, C++, and Java. Therefore, individual modules have the *potential* to run on multiple platforms, but as a whole the Toolkit only supports Windows.

One possible path to support additional platforms is to ensure that each module can run natively on the desired target platform, by re-configuring and/or re-compiling each module. This has two main drawbacks, however. One, while the number of main modules is manageable, they collectively depend on dozens of libraries, each of which needs to be ported to each target platform. Second, this approach only works for desktop platforms (e.g., MacOS, Linux), because the modularity cannot be maintained on mobile platforms (e.g., iOS, Android), as they require more tightly integrated solutions.

Another possible path is to pursue a client-server architecture, where most modules run on a server, communicating to a Unity client (e.g., on iPhone or HoloLens). We opted for this approach, because of its many advantages:

1) Once a module runs on the server, it can be re-used with little to no effort for any newly supported platform.
2) Adding support for a new platform only requires porting the Unity client.
3) The modularity of the architecture is retained.
4) New platforms supported by Unity are automatically supported by our framework.
5) Maintenance effort is only needed for the original modules, plus one client for each newly supported platform.

The obvious downsides are that applications now require an always-on Internet connection, which in turn introduces latency. However, given that these systems typically already require a connection in support of speech recognition (which typically needs to phone home), this was deemed acceptable.

We follow several strategies to minimize data requirements sent back and forth between the server and the client. First, where possible, known data is cached in the client, in particular the audio data of the character speech. Second, rather than send the character skeleton bone rotation data each frame from the server to the client, we opted to include the nonverbal behavior realization functionality within the Unity client. To keep the client lightweight and portable, we ported the core functionality of the existing module that was written in C++ to C# so it could use Unity's native animation system. The port maintains the original interface between the modules, so that the architecture as a whole remains unchanged. Third, we only use two (sequential) data streams between client and server: 1) client sending user speech to server, and 2) server sending character behavior instructions back to client. This also allowed us to set up the required server capabilities without an exorbitant amount of effort. The existing modules run on a Windows server and are wrapped collectively in an ASP.NET application that receives input from the client. The server modules communicate among themselves through ActiveMQ and do not need to be aware of the fact that they are hosted on a server, thus requiring no additional development effort. The final result is caught by the ASP.NET application and send back to the client. While certainly advantageous as a first step, the runtime performance is not optimized and we therefore aim to develop proper web services for these modules in the future.

C. Character Modeling and Animation

Our in-house developed art pipeline aims to abstract from specific art asset sources and character setups in order to support many applications concurrently. Character meshes can originate from high-fidelity scans (e.g., the LightStage [7]), commodity scans [27], 3rd party sources[5], or traditional methods (e.g., artist sculpt in Autodesk Mudbox). Characters are rigged and skinned in Maya[6] and exported to Unity. Where desired, we use 3rd party or custom functionality. For instance, we generate textures with Substance[7] (e.g., to allow for dynamic wear and tear on clothing) and developed custom shaders when utilizing high-fidelity facial scans (e.g., blending texture maps to simulate subsurface scattering).

While the skeleton setup can be arbitrary (e.g., in order to support non-humanoid characters, like [31]), there is a

[5]https://www.mixamo.com/
[6]https://www.autodesk.com/products/maya/overview
[7]https://www.substance3d.com/

standard skeleton in order to reduce complexity. The facial rig supports both blend-shapes and joints. Exported characters use the Unity Humanoid Avatar system, with a custom setup of compartmentalized animation controllers and layers, see Figure 2. This allows for increased flexibility in blending animations and controlling parts of the body separately, which is particularly important for natural conversational nonverbal behaviors. Behaviors are adapted from [26] and include procedural gazing, head nods and shakes, lipsync, facial expressions, gesturing, and locomotion. Animation data can be obtained by either traditional key-frame animation or motion capture. However, we typically do not animate entire performances. Rather, we provide each character archetype (typically male and female) with a suite of behaviors, including dozens of individual conversational gestures, and then create behavior schedules from these basic building blocks. These behavior schedules are in the form of the Behavior Markup Language (BML) [13], following the SAIBA framework [5]. A behavior schedule can be either authored offline by an artist or generated (in real-time or offline) by the appropriate module in the Toolkit [14]. This allows us to:

1) Re-use animation data.
2) Mix pre-authored animations (e.g., conversational gesture) with procedural animations (e.g., head gaze).
3) Create nonverbal behavior for any kind of verbal content without having to create new animations.
4) Choose between editorial control of the behavior when needed and generated behavior otherwise.
5) Develop rule-based systems that adapt a character's behavior to a user's (verbal or nonverbal) input.

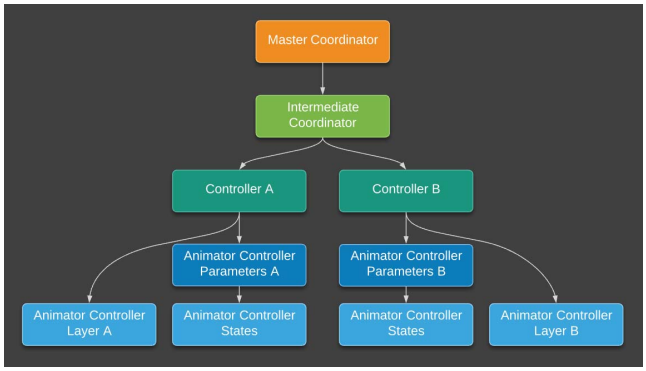

Fig. 2. Overview of animation controllers within Unity.

IV. Use Cases

A. Seated VR

Our first virtual human VR application was a proof of concept on what a future virtual therapist session could look like, using the HTC Vive. This application was not meant to assess or treat actual patients, but served as an interactive experience to think through technical, organizational and ethical considerations [8]. The application offered a seated experience, where the user interacted with the character verbally. The system contained a User Perceived State (UPS) component, which analyzed the user's input and mapped it to a two-dimensional emotional state consisting of Valence (positive or negative affect) and Arousal (intensity).[8] For instance, Low Valence / Low Arousal maps to Sadness, while Low Valence / High Arousal maps to Anger. This enabled a feedback loop between character and user over the course of a conversation. In particular, the character could:

1) Change posture (e.g., lean forward to try to convey interest and bring someone back into the conversation).
2) Blend between six different facial expressions (e.g., to convey concern or being content).
3) Increase or decrease the intensity of conversational gestures (e.g., to match the energy of the user).

Informal user testing taught us that users often felt an immediate connection with the character, primarily due to effectively maintaining eye contact, even when the user would get up and walk around the character (Figure 3). This was due to the existing procedural gaze controller, which uses eyes, neck and spine, to realistically look around. By making the virtual camera the gaze target, the character would automatically look at the user in a natural fashion. However, the character would not respond when users invaded her personal space, which came across as too passive and unnatural. In addition, having only purely procedural animations (e.g. joints driven by math) and pre-authored animations (e.g. conversational gestures) was limiting, as behaviors were either too mechanic or too specific. Also, these were separate systems that did not always play well with each other. We therefore started experimenting with example-based animation controllers, addressing both issues.

Fig. 3. Seated VR, with virtual human maintaining eye contact with user.

B. Room-scale VR

Our next VR application was an internal, technical prototype with the main goal of exploring room-scale VR with a virtual human, using an HTC Vive. We developed a virtual Thomas Jefferson – third president of the United States – who was able able to walk around, talk about objects of interest in his environment, and answer user's questions, see Figure 4. Users could interact with the virtual objects, interact verbally with Thomas Jefferson, and walk around. Based on lessons learned, Thomas Jefferson greeted the user when being approached, and backed away and put up both his hands in a "whoa there" manner when the user would invade his personal space.

[8]In this particular application, the UPS only analyzed the user's words, but additional analysis (e.g. audio-visual sensing) can be added.

Informal user testing informed us that there was a bigger learning curve than in our seated VR experience, due to the increased level of interactivity. This points to the need for more formal user testing, which fell outside the scope of this technical exercise.[9]

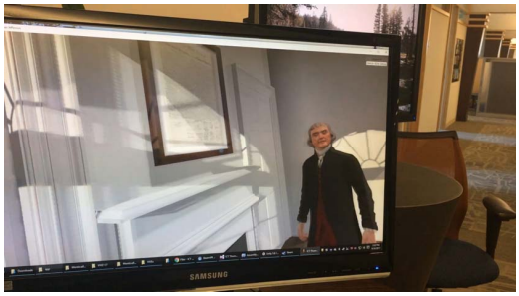

Fig. 4. User and virtual human walk in same space with the Vive.

C. Mobile AR

We ported the Thomas Jefferson prototype to iOS ARKit. Users could walk around with their phone or tablet, interact with the same objects as before, ask Thomas Jefferson questions, and invade his personal space (Figure 5). This version was of particular interest in groups of people, because the user was not isolated from the others (as is typically the case in VR) and because of the novelty of seeing a responsive AI character embedded in the real world, often next to other people.[10]

Porting the prototype from the Vive to iOS was relatively easy, given that the main functionality and content had already been created. All these could be re-used, including art assets, actor voiced character utterances, and natural language processing logic and data. This allowed us to focus on porting the Unity application from Windows Desktop VR to iOS ARKit, getting everything else basically for free. Taking advantage of all the features this new platform offers, though (e.g., real-world lighting and navigation) takes additional effort.

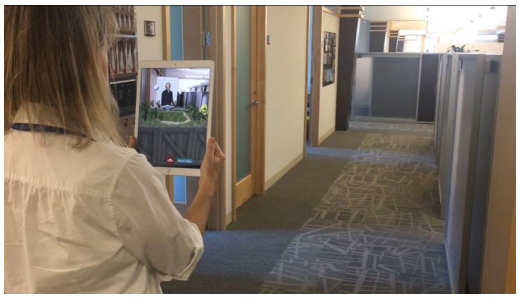

Fig. 5. iOS ARKit uses the same server modules and content as the Vive.

D. Headset AR

We have developed virtual humans for both the HoloLens and the Magic Leap, with the latter focusing on a virtual human job interview practice prototype. The user can choose between a male and female interviewer. The character is automatically placed on the nearest detected flat surface (e.g., a chair), after which they will ask the user a series of common job interview questions, see Figure 6. At the end of the interview, the user gets immediate feedback on a range of metrics, including eye contact, blink rate, and response delay, based on Magic Leap sensors. For more details, see [10].

Initial feedback has been encouraging. Early tests show that users are drawn to this novel technology and use of virtual humans, praise the immersion of characters, and appreciate the ability to practice valuable skills within a real-world setting. Next steps will focus on performing usability testing with our target audience, validating sensing data (e.g., blink rate, eye gaze), and adding more content.

Fig. 6. Embedding a virtual human within the real world on Magic Leap.

V. Conclusion

We presented an extension to the Virtual Human Toolkit, that supports a range of computing platforms, including most modern AR and VR solutions. Our solution is based on a client-server architecture, which maintains its modularity and flexibility, and which allows for server logic and content re-use, as well as new platform support by porting a lightweight Unity client. Our art pipeline can rig and skin arbitrary humanoid and non-humanoid characters and allows for the re-use of animation data by automatically generating nonverbal behavior performances that use a mix of procedural animation and pre-authored animation building blocks.

Future work will focus on improving 1) run-time performance by taking full advantage of web services, 2) expressive character animation by advancing example based controllers, 3) AR navigation by taking into account real world objects, and 4) automated user performance feedback by gathering and analyzing more sensor data.

Acknowledgment

We'd like to thank all of our collaborators at USC ICT. This work was supported in part by the U.S. Army. Any opinion, content or information presented does not necessarily reect the position or the policy of the United States Government, and no official endorsement should be inferred.

[9] Also, a virtual Hamilton may have been more lucrative.

[10] Taking "selfies" with Thomas Jefferson proved to be popular.

REFERENCES

[1] ANDERSON, K., ANDRÉ, E., BAUR, T., BERNARDINI, S., CHOLLET, M., CHRYSSAFIDOU, E., DAMIAN, I., ENNIS, C., EGGES, A., GEBHARD, P., JONES, H., OCHS, M., PELACHAUD, C., PORAYSKA-POMSTA, K., RIZZO, P., AND SABOURET, N. The tardis framework: Intelligent virtual agents for social coaching in job interviews. In *Advances in Computer Entertainment* (Cham, 2013), D. Reidsma, H. Katayose, and A. Nijholt, Eds., Springer International Publishing, pp. 476–491.

[2] BACCA, J., BALDIRIS, S., FABREGAT, R., GRAF, S., AND KINSHUK, D. Augmented Reality Trends in Education: A Systematic Review of Research and Applications. *Educational Technology and Society 17* (2014), 133–149.

[3] BICKMORE, T., SCHULMAN, D., AND SIDNER, C. Automated interventions for multiple health behaviors using conversational agents. *Patient education and counseling 92* (06 2013).

[4] BICKMORE, T. W., UTAMI, D., MATSUYAMA, R., AND PAASCHE-ORLOW, M. K. Improving access to online health information with conversational agents: A randomized controlled experiment. *J Med Internet Res*.

[5] CAFARO, A., BRUIJNES, M., VAN WATERSCHOOT, J., PELACHAUD, C., THEUNE, M., AND HEYLEN, D. Selecting and expressing communicative functions in a saiba-compliant agent framework. In *Intelligent Virtual Agents* (Cham, 2017), J. Beskow, C. Peters, G. Castellano, C. O'Sullivan, I. Leite, and S. Kopp, Eds., Springer International Publishing, pp. 73–82.

[6] CALLEJAS, Z., RAVENET, B., OCHS, M., AND PELACHAUD, C. A computational model of social attitudes for a virtual recruiter. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems* (Richland, SC, 2014), AAMAS '14, International Foundation for Autonomous Agents and Multiagent Systems, pp. 93–100.

[7] DEBEVEC, P. The light stages and their applications to photoreal digital actors. *SIGGRAPH Asia 2*, 4 (2012).

[8] GORDON, C., LEUSKI, A., BENN, G., KLASSEN, E., FAST, E., LIEWER, M., HARTHOLT, A., AND TRAUM, D. R. Primer: An emotionally aware virtual agent. In *IUI Workshops* (2019).

[9] HARTHOLT. HoloLens Virtual Human - YouTube - https://www.youtube.com/watch?v=vifHh4WjEFE.

[10] HARTHOLT, A., MOZGAI, S., FAST, E., LIEWER, M., REILLY, A., WHITCUP, W., AND RIZZO, A. S. Virtual humans in augmented reality: A first step towards real-world embedded virtual roleplayers. In *Proceedings of the 7th International Conference on Human-Agent Interaction* (2019), ACM, pp. 205–207.

[11] HARTHOLT, A., TRAUM, D., MARSELLA, S. C., SHAPIRO, A., STRATOU, G., LEUSKI, A., MORENCY, L.-P., AND GRATCH, J. All Together Now: Introducing the Virtual Human Toolkit. In *13th International Conference on Intelligent Virtual Agents* (Edinburgh, UK, Aug. 2013).

[12] HUGGINS-DAINES, D., KUMAR, M., CHAN, A., BLACK, A. W., RAVISHANKAR, M., AND RUDNICKY, A. I. Pocketsphinx: A free, real-time continuous speech recognition system for hand-held devices. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings* (2006), vol. 1, IEEE, pp. I–I.

[13] KOPP, S., KRENN, B., MARSELLA, S. C., MARSHALL, A., PELACHAUD, C., PIRKER, H., THÃ³RISSON, K. R., AND VILHJÃ¡LMSSON, H. Towards a Common Framework for Multimodal Generation: The Behavior Markup Language. In *Proceedings of the Intelligent Virtual Humans Conference* (Marina del Rey, CA, Aug. 2006).

[14] LEE, J., AND MARSELLA, S. Nonverbal behavior generator for embodied conversational agents. In *International Workshop on Intelligent Virtual Agents* (2006), Springer, pp. 243–255.

[15] LEUSKI, A., AND TRAUM, D. Npceditor: Creating virtual human dialogue using information retrieval techniques. *Ai Magazine 32*, 2 (2011), 42–56.

[16] MAGIC LEAP. I Am Mica — Magic Leap - https://www.magicleap.com/news/op-ed/i-am-mica.

[17] MANNING, C., SURDEANU, M., BAUER, J., FINKEL, J., BETHARD, S., AND MCCLOSKY, D. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations* (2014), pp. 55–60.

[18] MCKEOWN, G., VALSTAR, M. F., COWIE, R., AND PANTIC, M. The semaine corpus of emotionally coloured character interactions. In *2010 IEEE International Conference on Multimedia and Expo* (2010), IEEE, pp. 1079–1084.

[19] MCVEIGH-SCHULTZ, J., MÁRQUEZ SEGURA, E., MERRILL, N., AND ISBISTER, K. What's it mean to be social in vr?: Mapping the social vr design ecology. In *Proceedings of the 2018 ACM Conference Companion Publication on Designing Interactive Systems* (2018), ACM, pp. 289–294.

[20] MILLER, M. R., JUN, H., HERRERA, F., YU VILLA, J., WELCH, G., AND BAILENSON, J. N. Social interaction in augmented reality. *PLOS ONE 14*, 5 (may 2019), e0216290.

[21] MOZGAI, S., LUCAS, G., AND GRATCH, J. To tell the truth: Virtual agents and morning morality. In *Intelligent Virtual Agents* (Cham, 2017), J. Beskow, C. Peters, G. Castellano, C. O'Sullivan, I. Leite, and S. Kopp, Eds., Springer International Publishing, pp. 283–286.

[22] RIZZO, A., CUKOR, J., GERARDI, M., ALLEY, S., REIST, C., ROY, M., ROTHBAUM, B. O., AND DIFEDE, J. Virtual reality exposure for ptsd due to military combat and terrorist attacks. *Journal of Contemporary Psychotherapy 45*, 4 (2015), 255–264.

[23] RIZZO, A., SCHERER, S., DEVAULT, D., GRATCH, J., ARTSTEIN, R., HARTHOLT, A., LUCAS, G., MARSELLA, S., MORBINI, F., NAZARIAN, A., STRATOU, G., TRAUM, D., WOOD, R., BOBERG, J., AND MORENCY, L. P. Detection and computational analysis of psychological signals using a virtual human interviewing agent. *Journal of Pain Management* (Nov. 2016), 311–321.

[24] SCHERER, S., MARSELLA, S., STRATOU, G., XU, Y., MORBINI, F., EGAN, A., MORENCY, L.-P., ET AL. Perception markup language: Towards a standardized representation of perceived nonverbal behaviors. In *International Conference on Intelligent Virtual Agents* (2012), Springer, pp. 455–463.

[25] SCHUSTER, M. Speech recognition for mobile devices at google. In *Pacific Rim International Conference on Artificial Intelligence* (2010), Springer, pp. 8–10.

[26] SHAPIRO, A. Building a character animation system. In *International conference on motion in games* (2011), Springer, pp. 98–109.

[27] SHAPIRO, A., FENG, A., WANG, R., LI, H., BOLAS, M., MEDIONI, G., AND SUMA, E. Rapid avatar capture and simulation using commodity depth sensors. *Computer Animation and Virtual Worlds 25*, 3-4 (2014), 201–211.

[28] SHI, H., AND MAIER, A. *Speech enabled shopping application using Microsoft SAPI*. PhD thesis, 1996.

[29] SLATER, M. Immersion and the illusion of presence in virtual reality. *British Journal of Psychology 109*, 3 (2018), 431–433.

[30] SNYDER, B., BOSNANAC, D., AND DAVIES, R. *ActiveMQ in action*, vol. 47. Manning Greenwich Conn., 2011.

[31] SWARTOUT, W., NYE, B. D., HARTHOLT, A., REILLY, A., GRAESSER, A. C., VANLEHN, K., WETZEL, J., LIEWER, M., MORBINI, F., MORGAN, B., WANG, L., BENN, G., AND ROSENBERG, M. Designing a Personal Assistant for Life-Long Learning (PAL3). In *Proceedings of The Twenty-Ninth International Flairs Conference* (Key Largo, FL, May 2016), AAAI Press, pp. 491–496.

[32] VIAR360. How did the VR market do in 2018 and what's the forecast for 2019? - https://www.viar360.com/virtual-reality-market-size-2018/.