SYSTEM ARCHITECTURE FOR SUPPORTING MULTIPLE LIVE ACTORS IN WEB3D VIRTUAL CONFERENCE

Extended Abstract

Tin Hok
Department of Computer Science, Chungbuk National University, Seoul, South Korea
hoktin@chungbuk.ac.kr

Chan Park
KIS Corporation, South Korea
szell@gmail.com

Kwan-Hee Yoo
Department of Computer Science, Chungbuk National University, Seoul, South Korea
khyoo@chungbuk.ac.kr

ABSTRACT

For live actors to perform as 3D models with seamless graphics computing, we need to overcome the huge challenges of developing a methodology to construct a cleaned and smooth model in real time and a system architecture design that supports the streaming of multi-model live actors. By applying the technique of semantic image segmentation to generate the entire human body, a 3D model needs to be built up with a segmented image as a texture. Moreover, to enable a live actor to virtually interact with others, multi-actor models must be produced for telecommuting systems. Therefore, by combining the concepts above, a system with multiple live actors can be created to conduct live group conferences in virtual reality that provide an immersive experience.

CCS CONCEPTS

• **Computer graphics**; • **Multiple live actors in Web3D virtual conference**;

KEYWORDS

Multiple live actors, live actor and entity (LAE), DeepLab, WebRTC, ThreeJS, Web3D, mixed augmented reality (MAR), head-mounted display (HMD)

ACM Reference Format:
Tin Hok, Chan Park, and Kwan-Hee Yoo. 2020. SYSTEM ARCHITECTURE FOR SUPPORTING MULTIPLE LIVE ACTORS IN WEB3D VIRTUAL CONFERENCE: Extended Abstract. In *The 25th International Conference on 3D Web Technology (Web3D '20), November 09–13, 2020, Virtual Event, Republic of Korea.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3424616.3424696

1 INTRODUCTION

Nowadays, researchers in the field of computer graphics are devoting much effort to discover new ways to explore, visualize, and experience real and imaginary nonphysical worlds. Thus far, applications of video conferencing for group meetings have mostly been implemented in 2D graphics. On the other hand, 3D conference rooms hosted in an application, which allow users to interact in a virtual environment, have not been sufficiently researched owing to the difficulty of reforming real-world objects into virtual objects with sufficient speed. To realize the design of a complete system, Chen et al. [Chen et al. 2018] developed a semantic image segmentation system called DeepLab to track the target object, i.e., the human body, from the real world and convert it into sequences of 2D images for 3D model construction in ThreeJS [Cabello 2010], which is a JavaScript library and application programming interface (API) for creating and displaying animated 3D computer graphics in a web browser. In addition, multi-actor modeling in real-time communication requires WebRTC [Johnston and Burnett 2012], which is a powerful voice- and video-communication solution available on all modern browsers.

2 PROPOSED SYSTEM PROCESS ARCHITECTURE

Inspired by Yoo [Yoo 2016], we proposed an approach to virtualize chromakeying images as multiple live actors in a virtual scene, called a mixed augmented reality (MAR) scene. Here, the system serves the function of allowing a connection between computers to share captured images such that multiple live actors, i.e., the models formed by segmented images, are displayed in one place in the virtual world. Through this concept, we can implement a system that has the potential to handle a virtual conference in which the captured and segmented images can be presented in 3D space to provide a real-world-like experience. The actor can perform actions or movements spatially based on the mapped information in the system.

The system is composed of several technological components, and DeepLab is mainly used to settle the segmentation of image sequences. In addition, ThreeJS can create and display animated 3D computer graphics in a web browser. In particular, for distributing information from different machines, peer-to-peer connections are needed and can be implemented with WebRTC. As briefly mentioned above, an overview of the architecture is shown in Figure 1

Web3D '20, November 09–13, 2020, Virtual Event, Republic of Korea

ACM ISBN 978-1-4503-8169-7/20/11…$15.00
https://doi.org/10.1145/3424616.3424696

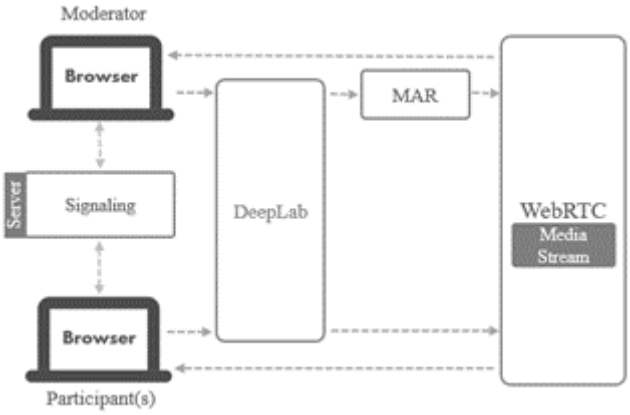

Figure 1: Overview of the architecture for supporting multiple live actors in a virtual conference.

3 MODELING MULTIPLE LIVE ACTORS

As shown in Figure 2, an essential implementation of [Hok et al. 2019] to virtualize real images in a 3D world is the modeling of a human body, i.e., an actor, in a virtual scene. Technically, a web application can directly access a connected general camera to acquire images. Subsequently, captured images for segmentation are analyzed at the pixel level by DeepLab [Chen et al. 2018], a deep learning algorithm, for extracting the target object, i.e., the entire human body. In the phase of placing actors in a virtual environment, a square-box geometry is applied with the segmented image as a texture to represent a live actor in the system.

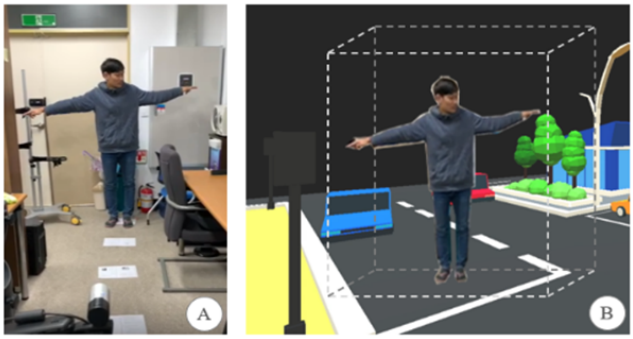

Figure 2: (A) By using a general camera, the ground-truth image is captured in sequences. (B) The segmented image is converted as a texture mapped on a 3D model in a virtual scene.

4 HANDLING WEB3D VIRTUAL CONFERENCE

By applying the proposed method, the system can achieve a complete Web3D virtual conference offering online meetings via browsers. As Figure 1 shows, the moderator can organize and manage a conference by designing a virtual environment with available materials defined for decoration. Further, the position of each live-actor model can be varied using a spatial mapper in the MAR scene, for which the moderator takes full responsibility. On the other hand, a participant can only connect to a hosted conference and share sequences of captured images and information. In return, directly in the browser, other participants can view the designed virtual environment to know when live actor(s) exit the scene.

4.1 MAR SCENE

As per Chheang and Yoo [Chheang and Yoo 2017], MAR represents a continuum that encompasses all domains or systems that use a combination of real and virtual representations. Hence, MAR is defined as a mixture of real and virtual representations but excludes purely real environments and purely virtual environments.

By referring to their paper, an MAR scene can be virtually constructed using many 3D models that are freely designed for any particular purpose with a list of modeled elements. For instance, in an MAR scene, it is possible to decorate classrooms, houses, news broadcast studios, office rooms, etc. As stated earlier, only the moderator can control aspects of the MAR scene such as the environment design and live-actor model positioning.

4.2 SPATIAL MAPPING

As mentioned in Chheang and Yoo [Chheang and Yoo 2017], a live actor in the physical world is embedded into a 3D virtual world in an MAR application. The role of the spatial mapper is to support the natural movement of the live actor within the 3D virtual world. The spatial mapper of live actor and entity (LAE) provides spatial information, such as position, orientation, and scale, between the physical-world space and MAR scene space by applying transformations for calibration.

Chheang and Yoo [Chheang and Yoo 2017] clearly describes the functionalities of spatial mapping, which intentionally translates real-world objects into 3D virtual words in a relative manner while supplying explicit mapping information. Within this mapping, each object model can be defined in the MAR scene with the position, orientation, and scale as variables. The moderator manages the models in space such that the environment is dynamically sketched based on the positioning of the provided models. Importantly, there are two types of pre-defined models. First, 3D object models are used to build a complete 3D-model-based structure for decorating an empty space. The second type is the live-actor model, which is the principal feature to discuss and it plays a critical role in forming the segmented 2D image as a texture of the 3D model. With this, the spatial mapper can technically locate a live-actor model with coordinates in accordance with the defined variables.

4.3 WebRTC

For multiplying modeled actors into many forms for a virtual conference, a web application can manage the communication between computers through the browser. Basically, a server is used for signaling with the connected clients' signals and allowing browsers to access peer-to-peer data as media streams via WebRTC [Johnston and Burnett 2012]. Moreover, video, voice, and generic data can be sent between peers in real time. This is demonstrated in Figure 3, which shows the WebRTC architecture.

Figure 3: WebRTC architecture serving a virtual conference.

Figure 3 conceptually illustrates that there is a web server in charge of exchanging signals among computers for a peer-to-peer connection. Remarkably, this does not mean that every computer must establish a peer-to-peer connection with every other connected computer; rather, a computer connects directly to the moderator computer, which hosts the Web3D virtual conference. For media streaming, each participant prepares and sends sequences of segmented images, which constitute a live video of a live actor, and the participant receives the stream of the MAR scene in return, which is the final display of the virtual conference in its entirety.

5 IMPLEMENTATION AND RESULTS

By integrating the process of constructing a segmented image as the model with the streaming of data between peers, the system can finally form a complete structure that aims to create a system of modeling a live multi-actor. According to Figure 1, each participant follows the same procedure to communicate with the moderator, who hosts a virtual meeting.

Primarily, the browser provides access to the connected camera by ID in the capturer module. In addition, the size, resolution, and mode are specified as properties of the input. The result is a series of ground-truth images captured by a general camera.

In the tracking stage applying the DeepLab technique, the sequences of captured images must be analyzed to extract the target, which can be any shape or object based on the trained model. However, in the present context, the system only aims to filter a complete human body out of the background. DeepLab offers the mask information of an image, which can be used to filter out the necessary pixels. The untargeted parts of the image can be dropped and transformed into transparency.

The process of the subsequent spatial mapping stage requires some variables such as position, orientation, and scale, as discussed in section 4.2, "Spatial Mapping," to discover where the live-actor model must be located. Here, each live-actor model has its own spatial mapper for the MAR scene. Only the moderator can control the configurations in terms of assigning values for the variables.

Next, to apply the segmented image on a box model for virtualization in the MAR scene, the image must be attached as a texture mapped to the model. Additionally, the model can be fully customizable, corresponding to the properties of a box model defined in ThreeJS. After the image has been applied with segmentation and formed as a live-actor model, the live-actor model can exist in the MAR scene in real time as it appears in the captured images.

As Figure 4 illustrates, the process depends on whether the computer proceeds to the moderator or participant. However, the workflows of the process are quite similar for the purpose of forming a live-actor model in the MAR scene. Noticeably, at the moderator's side, the operation is executed in the same machine from the beginning to the end. In contrast, as explained in Figure 3, the participant needs to establish a peer-to-peer connection through signals by simply exchanging the identity of devices on a signaling server for media streaming with WebRTC [Johnston and Burnett 2012], which allows full access to devices across browsers without being limited to voice, video, or data. This process works in real time to enable live interactions among users.

Therefore, a canvas is created in the browser and used to render the wholly constructed MAR scene including multiple live actors. Likewise, the MAR scene can be rendered in other connected machines via the implemented media streaming solution, WebRTC [Johnston and Burnett 2012].

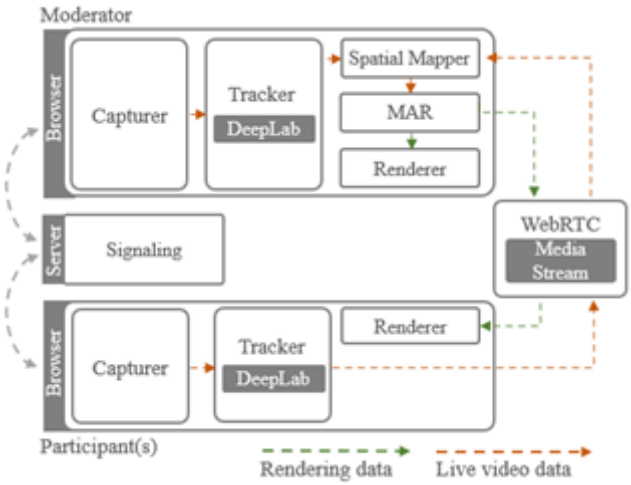

Figure 4: Detailed architecture supporting multiple live actors in a virtual conference.

As shown in Figure 5 because DeepLab provides several pretrained models that can be used in this system, a trained model with 80.25% (mIOU) at the "mobilenetv2_coco_voc_trainval" checkpoint name has been selected and used for current development. In addition, this model was trained with a 2012 PASCAL VOC dataset for image segmentation. Because this implementation is still in development, some information on it is unavailable.

The development is in the testing stage, and the entire architecture in Figure 4 is running in the same machine. However, the implementation respects the design of the architecture. The segmentation process is handled by a server with an i7-9700KF CPU

running at 3.60 GHz, 16 GB RAM, and a GeForce RTX 2060 GPU. This system continuously generates sequences of images and transfers them to the frontend at 3 fps. Therefore, the speed of analyzing the targeted object is low. However, in the final development stage, every connected computer must manage the segmentation process separately. Thus, the framerate performance will be significantly improved. Nevertheless, the testing system can demonstrate multiple live actors in virtual communication in real time.

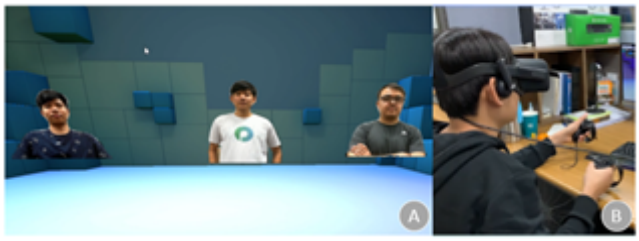

Figure 5: (A) Three live actors displayed in a virtual conference room. (B) The use of an HMD device to virtualize the entire virtual room.

In Figure 5(A), three different 2D-based live actors are shown in a 3D space functioning as a virtual conference. This is achieved through a connection between computers to share images onto a decorated space on the moderator's side. Figure 5(B) shows the use of a head-mounted display (HMD) device to realize the immersive experience involving the virtualization of the entire 3D scene. The system is implemented with WebXR [WebXR 2019], which is described in Tin et al. [Hok et al. 2019].

6 CONCLUSIONS

A system such as the one proposed in this paper can handle a virtual conference with 2D images of live actors in a 3D world. Such virtual conferences would be of great practical utility, providing a wide variety of benefits. Since the modeling of an actor still occurs in the 2D form, this system may be considered a gateway to reform a model into a completely 3D base in the near future. The overall concept proposed in this paper will be improved further in the final stage of development to perform seamless 3D-object modeling and to increase the operation speed of image segmentation.

ACKNOWLEDGMENTS

This research is based upon work supported by the Ministry of Trade, Industry & Energy (MOTIE, Korea) and Korea Evaluation Institute of Industrial Technology (KEIT)(No.10085589), and by the MSIP(Ministry of Science, ICT and Future Planning), Korea, under the ITRC(Information Technology Research Center) support program (IITP2018-2015-0-00448) supervised by the IITP(Institute for Information & communications Technology Promotion).

REFERENCES

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. 2018. "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation," arXiv:1802.02611 [cs], Feb. 2018

Ricardo Cabello. 2010."three.js – JavaScript 3D library." https://threejs.org/ (accessed Oct. 22, 2019)

Alan B Johnston and Daniel C Burnett. 2012. "WebRTC," WebRTC. https://webrtc.org/ (accessed Jul. 03, 2020)

Kwan-Hee Yoo. 2016. "Standard Model for Live Actor and Entity Representation in Mixed and Augmented Reality," Journal of Broadcast Engineering 21(2):192-199, vol. 21, pp. 192–199, Mar. 2016

Hok Tin, Min Borin, Seunghyeon Kang, Ga-Ae Ryu, and Kwan-Hee Yoo. 2019. "Real-Time Embedding Live Actors into an Immersive World," ICCC, 2019

Vuthea Chheang and Kwan-Hee Yoo. 2017. "Information technology — Computer graphics, image processing and environmental data representation — Live actor and entity representation in mixed and augmented reality," ISO/IEC JTC 1/SC 24 N 18040, 2017

WebXR API. 2019."WebXR Device API." https://immersive-web.github.io/webxr/ (accessed Oct. 22, 2019)