Multimodal Technologies and Interaction

MDPI

Article

ODO: Design of Multimodal Chatbot for an Experiential Media System

Ravi Bhushan [1,†], Karthik Kulkarni [1,†], Vishal Kumar Pandey [1,†], Connor Rawls [2], Brandon Mechtley [2], Suren Jayasuriya [1,2] and Christian Ziegler [2,*]

1 School of Electrical, Computer and Energy Engineering, Arizona State University, Phoenix, AZ 85281, USA; rbhushan@asu.edu (R.B.); kkulkar4@asu.edu (K.K.); vkpande1@asu.edu (V.K.P.); sjayasur@asu.edu (S.J.)

2 School of Arts, Media and Engineering, Arizona State University, Phoenix, AZ 85281, USA; cwrawls@asu.edu (C.R.); bmechtley@asu.edu (B.M.)

* Correspondence: cz@movingimages.de

† These authors contributed equally to this work.

Received: 24 July 2020; Accepted: 15 September 2020; Published: 23 September 2020

Abstract: This paper presents the design of a multimodal chatbot for use in an interactive theater performance. This chatbot has an architecture consisting of vision and natural language processing capabilities, as well as embodiment in a non-anthropomorphic movable LED array set in a stage. Designed for interaction with up to five users at a time, the system can perform tasks including face detection and emotion classification, tracking of crowd movement through mobile phones, and real-time conversation to guide users through a nonlinear story and interactive games. The final prototype, named ODO, is a tangible embodiment of a distributed multimedia system that solves several technical challenges to provide users with a unique experience through novel interaction.

Keywords: chatbots; digital theater; media arts; performing arts

1. Introduction

With the advances in artificial intelligence, especially in computer vision and natural language processing, machine agents serving as natural language user interfaces (chatbots) have become highly sophisticated. These chatbots have seen widespread use in client communication, industry development, and communication interfaces. While some of the traditional criticism of these chatbots include their limitations of connecting with human beings and expressing emotions, there has also been work to try and improve the affective computing underlying chatbots [1,2]. This has allowed them to be deployed in diverse forward-facing settings including psychological counseling and therapy services [3].

However, there has been a dearth of research on enabling chatbots for artistic installations, particularly chatbots that can interface with existing media systems such as distributed audio, video, and illumination, among others. This is primarily due to two main challenges: (1) existing chatbots can rarely integrate multimodal sensory inputs such as combined audio and video to perform their functions, and (2) existing experiential media systems typically feature primitive or non-intelligent signal processing or algorithmic controllers, which rarely can adapt to users in semantically meaningful ways.

Further, there is interest in making these systems energy-efficient to reduce the carbon footprint and enable new use cases for the technology. Edge computing devices have recently gained sophistication in their capability to handle complex workloads including that of image and audio processing necessary for embedded machine learning. This is an emerging paradigm that can be leveraged by experiential multimedia systems for maximum impact.

1.1. Contributions

Inspired by these challenges, we have endeavored to create a novel, multimodal chatbot that can successfully integrate with an experiential media system. The goal of this system is to provide a new, interactive experience for multiple users, and designed for digital and interactive theater. Our chatbot is able to perform crucial vision and natural language processing to communicate with end users, while still communicating and actuating a distributed media system to change the environment and ambiance settings.

A key challenge for realizing this chatbot was integrating the different media components, including the capabilities for distributed computing and the system challenges at both a software and hardware level. In this paper, we explicate the design of our chatbot and its interface with the larger system, and discuss the key insights into artistic deployment of such chatbots. In particular, we focus on the implementation details of the multimodal system and how the engineering was accomplished to create the theatrical character.

Our specific contributions include the following:

- Design of a conversational interface that facilitates storytelling and interaction through a deep learning-based universal encoder [4] and utterance-intent modeling.
- A vision system consisting of face detection as well as emotion classification [5] using neural networks.
- Custom user position tracking via mobile devices to enable crowd index and other aggregate motion information for the system.
- A communication protocol via Open Sound Control (OSC) [6] and Transmission Control Protocol (TCP) to embedded computing platforms.

In this paper, we highlight the implementation details for this multimodal system and leave a more formal evaluation of the human–computer interaction and user study to future work when the system is deployed in a theatrical production.

1.2. ODO Character and Story

"No Body lives here (ODO)" is a digital theater production with a chatbot, replacing a human actor on stage with an AI agent, as shown in the conceptual diagram in Figure 1. ODO is both the name of the production and the main theatrical character: a multimodal chatbot who conducts conversations, generates poetry on the fly, and plays interactive games with the audience. The chatbot "senses" people clusters and estimates a separation index, measuring the behavior of the "audience body" during the performance. Various multimodal sensing environments along with deep learning, computer vision, and natural language processing (NLP) come together in a soft real-time system, communicating with an audience on several levels.

ODO is a physical system, which uses sensor data of users to interact in real physical environments, generating an immersive physical interface for multimodal communication via text, image, sound and physical motion. The chatbot ODO's "gestalt" is an architecture of kinetic LED lights, hung in a matrix of 26 lights. The performance is programmed to communicate with five members of the audience at the same time through users' mobile phones with a voice-to-text app. In addition to voice and language data, the chatbot also receives gestural and position data of the audience members through the API of the mobile phones. At the same time, two 160 degree cameras in the center of the stage are tracking faces of the audience, analyzing and collecting emotional data through facial expressions. All of this sensory and user data is fed to a series of four connected computers and one central computer host that is the brain of ODO. ODO is envisioned as an immersive communication system using theater technologies and chatbot conversation techniques for the 21st century.

ODO's story is based on Plato's allegory of the cave: ODO desires to escape its cave to see the blue sky for the first time. The audience is in the room with ODO to help it achieve its goal. The interactive

performance is a post dramatic performance system [7], multimodal, and is both a radio play and an immersive installation.

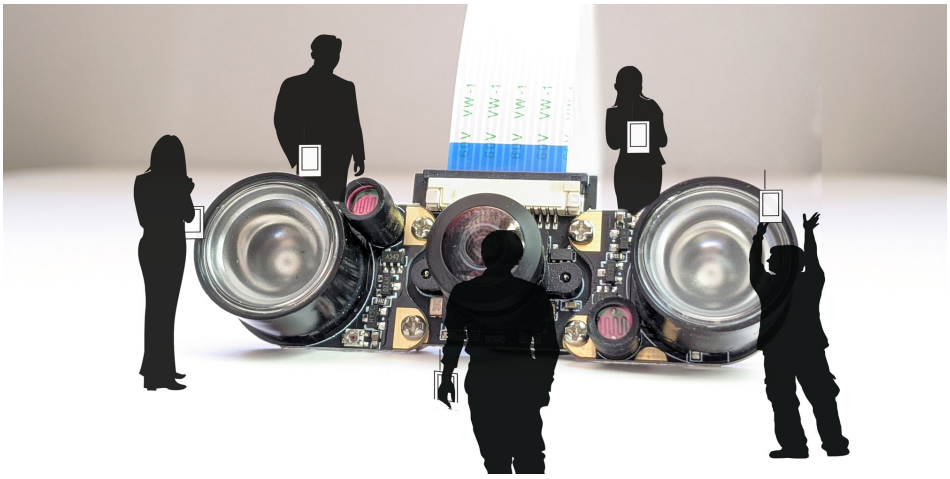


Figure 1. "No Body lives here (ODO)" is a digital theater production featuring a chatbot as a central character on stage. An experiential media system consisting of a motorized LED stage, edge computing hardware, and mobile apps is built to engage users in the performance while they interact with ODO.

1.3. Related Work and Context

Our work draws upon a vast literature in the areas of intelligent personal assistants, experiential media systems, and the deployment of computer characters/actors in theater productions. Comparing similarities and differences, we situate our work with ODO at the intersection of these three areas.

Intelligent personal assistants. There have been several recent developments in the area of intelligent personal assistants (IPAs) [8] include the creation of Amazon Alexa and Google Assistant. In this paper, we will refer to these IPAs as chatbots for ease of use. There exist many currently available frameworks for chatbots including OpenDialog [9], Google Dialogflow [10], IBM Watson [11], and Microsoft Bot Framework (MBF) [12]. While all of these options feature advanced functionality especially in their NLP capabilities, we decided to implement our own chatbot with its own custom dialogue management system in the Python programming language. Our reason for doing so is maintaining an easy-to-use, simple code interface that could interact with our multimodal stage. Our chatbot was heavily modeled after the Amazon Lex framework [13], but we needed to bypass the Amazon Web Services (AWS) database in order to do custom TCP/IP communications for our system. Recently, the use of deep learning and neural networks within these chatbots has led to improvements in state-of-the-art performance [14–17]. We also augment our chatbot with deep learning networks, particularly in computing word similarity for improved NLP performance, but also adding vision capabilities including facial detection/recognition and emotion classification.

The advances in chatbot technology has, in turn, spurred a renaissance of research on the topic [18], including human–computer interaction [19–21]. Many studies have focused on the affective components of interacting with chatbots [2,22] and how humans language use changes when interacting with them [23]. In our paper, we are focused on the implementation details of our chatbot. A full HCI study analyzing the efficacy of our system for user engagement is the subject of future work.

Experiential media systems. We follow the framework introduced by [24] of experiential media systems. Such systems build on the foundation of experiential computing [25,26], and typically feature sensing, computation, and multimodal feedback to the users in the system. Experiential media systems have been built for purposes such as biofeedback for stroke rehabilitation [27] and studying coordinated group activity [28]. One potentially similar experiential media system is the "Sound and Light" festival held on the Island of Rhodes, where multimodal interaction uses stereophonic light and

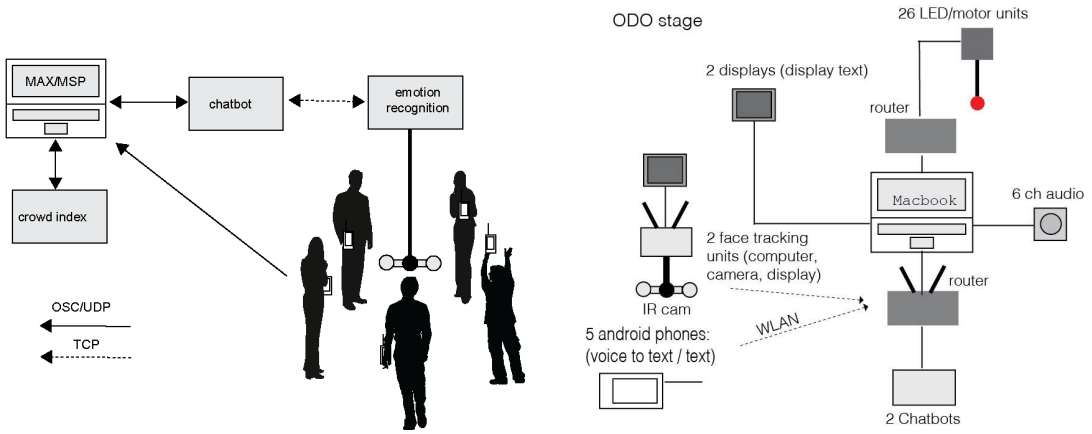
sound to tell the medieval history of Rhodes without actors. Our system differs from this particular example as we involve human audience participants to interact with an AI chatbot as part of the performance. ODO is an experiential media system that utilizes a physically-instantiated matrix of 26 LED lights that can move up/down on the stage coupled with displays and speakers to give users an interactive theater experience. In particular, the feedback given to the user is driven by the chatbot, thus yielding a level of personalized feedback that is rare among typical experiential media systems.

Autonomous computer characters in theater. There has been an active thread of research using autonomous computer characters in theater productions. In [29], the first computer character was introduced to interact with a human actor on stage. In this work, the computer uses gesture recognition to analyze a human actor's movements, and then digitally projected a virtual character onto a screen to advance the story. Following this line of work, there have been several more uses of robots in theater [30–32] including exploring improvisation and non-scripted behavior [33]. All these examples were designed to give the automated character presence on the stage, helping to draw the audience's attention and engage them.

In contrast to these previous works, in our production, ODO is an invisble character that is not represented by any one device, but which manifests itself only through voice and ephemeral events. Twenty-six lights moving in space form a cloud, which triggers visual associations utilizing a LED stage. Since the resolution of 26 lights is not high enough to generate a display system, the imagination of the audience is requested to fill in the gaps. Rather than a limitation of the system, this is an intentional benefit to trigger associations for the audience and also to avoid representational issues with giving the chatbot a type of distributed embodiment rather than a single body/terminal, which limits user interaction. One of the main aims of post-dramatic theater is to question the central role of the text or script in a performance. Rather, other elements of theater include movement, image and sound, and the new post dramatic theater dramaturgy puts all media elements of the stage on the same equitable level with no hierarchy. With this, ODO can be described as a post-dramatic digital character, using all media elements on stage including the underlying chatbot to realize its presence.

2. System Overview

Our system consists of several key components including the chatbot itself, its computer vision sub-system for face detection/recognition and emotion classification, and the crowd tracking through mobile phones. All of this is wrapped together through a communication protocol designed for real-time responsivity and feedback with the users. A full diagram of the system architecture can be seen in Figure 2b. In the following subsections, we describe the design choices for each of the sub-systems that make up the entire architecture.



(**a**) System Architecture (**b**) Hardware Connectivity

Figure 2. System architecture for ODO including the high level block diagram in (**a**) and the hardware components and connectivity in (**b**).

2.1. Chatbot

Designs of chatbots can range from the conventional, knowledge-based architectures to more free-ranging, open models. In our case, the design requirements of the chatbot include: (1) the ability to interpret user behavior and inputs to provide appropriate responses, and (2) a "character" or dramatic identity to aid the theater performance. Thus we built our own chatbot from scratch, inspired by the design of the Amazon Lex chatbot.

In particular, we leverage the idea of giving the chatbot a character concretely in the form of stories, similar to a script one gives an actor on stage. This script is not necessarily prescriptive or governs all responses of the chatbot deterministically, but rather highlights guideposts and stepping stones in the story arc that the general narrative should take. Thus, this resulted in two main design tasks: mapping user utterances to intents that move the story along, and adopting a conversational style in the gaps or open periods of the performance where the chatbot is left to "improvise" or create its own narrative possibilities.

Intent modeling for storytelling. Stories are stored as a collection of intents and each intent is a pair of utterances and responses. Utterances are a set of sentences that a user can probably utter to the chatbot. These utterances will trigger its associated responses, which are nothing but the story lines. Depending on the utterance, different story lines are presented to the user. For instance when the user utters words/sentences like "plane", "go for plane", "fix the plane", it triggers the response, which is one of the dialogue "let's fix the plane and then search water". These both are wrapped in one collection called intent (in this case intent name is "plane"), which basically checks the intention of the user behind uttering their words. In this way, a conventional chatbot is converted into a theatrical story teller.

While simple keywording spotting could give some base functionality for our chatbot, we implemented a deep learning network for natural language processing to help ensure that the system could robustly identify synonyms and other utterances that are similar to the desired intents. To process the words/sentences uttered by the user and map them to the utterances that we have stored, the chatbot uses a pre-trained Deep Learning model—Universal Sentence Encoder-USE [4]. This takes the user input, performs word embedding (namely word2vec [34,35]), combines the words using composition function and then finds similarity between the two sentences. The one which has the highest similarity is chosen. For instance, a sentence like "repair the plane" and "fix the plane" basically have the same meaning. This similarity is understood by using USE and this way the user does not have to utter the exact words/sentences, giving them the flexibility to converse with the chatbot. For out of domain utterances, if the similarity score was below a certain threshold, a standard response was selected, which reset the conversation or returned back to the previous intent in the conversation.

Conversational strategies. Conventional chatbots are either convergent or task-based (e.g., Amazon Lex, query systems) or divergent/theme-based (e.g., Pandorabot's Mitusku [36]). Both these conversational strategies do not fit to a conversation in theater, where the main goal is to develop a conversation based on a given story or dramaturgy.

We tested branching hierarchies and nonlinear storytelling strategies until we found a combined system of convergent and divergent conversational strategies that managed to both involve an audience in a conversation and also drive the story. A chatbot uses utterances, intents and entities or slots to structure a conversation. The utterance is the human to machine conversation to find out an intent of a conversation, which leads to the entities or slots to set cornerstones of a conversational path. In a theatrical conversation theses paths have to be both divergent and convergent, but they are set in specific times, which develops a specific chatbot dramaturgy. It resembles a road map connecting cities, where a main road branches out into meandering smaller streets entering a city. Inside the city the map is two dimensional and nonlinear. Leaving the city the streets need to converge back into a linear system of one main road to connect the next city or hub. We visualize this strategy in Figure 3.

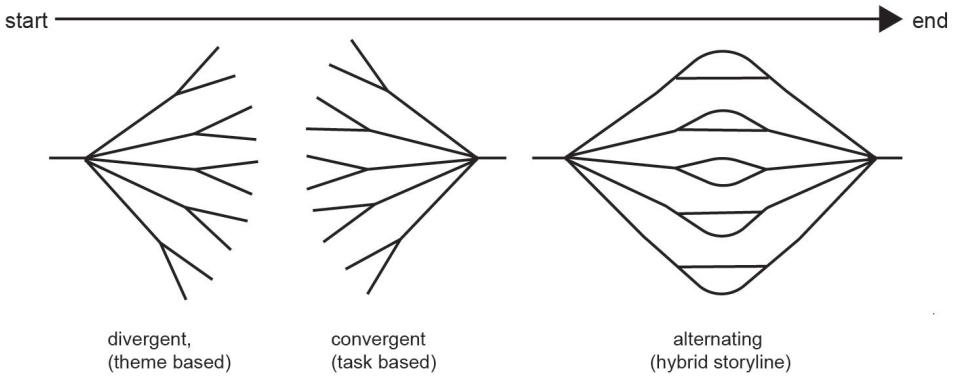


Figure 3. Conversational Strategies for Chatbots.

For the nonlinear conversation part, the chatbot was given the ability to interact with the audience through physical games and interaction with the system, or recite haikus to the users. This gave a whimsical nature to the chatbot that is refreshing and charming, leading the user through the story without feeling forced or rushed towards the end goal, at least in design.

2.2. Vision Sub-System

The key to ODO's design is the ability to visually sense the users it is engaging with. We primarily use two video feeds from 160 degree cameras and design the system to accommodate five users at a time. We chose five users because this was the maximum number of users that we could accommodate for our interaction games, which is a main focus of ODO's performance. Our vision sub-system consists of the main task of (1) detecting the faces of these five users, and (2) performing emotion classification to help aid the chatbot in performing affective decision making in its responses to user utterances.

Reading images coming off a live camera stream, the system first performs face detection using Haar feature-based cascade classifier [37] built into the OpenCV library. After face detection, emotion estimation is performed using a trained deep learning emotion recognition model based on a convolutional neural network [5]. Every detected emotion is classified in one of angry, disgust, scared, happy, sad, surprised, and neutral. While we acknowledge such a coarse, discrete categorization is not indicative of the range of human emotions and subtleties that can occur in a theatric performance, we found this enabled a tractable computational platform to perform decision making for the chatbot.

The software archives the detected emotion of the audiences and shares this information using a client/server architecture and Transmission Control Protocol/Internet Protocol (TCP/IP) protocol. Each of the vision sub-systems is identified by their name and the chatbot can request the emotion of a particular vision system having a specified field of view or the consolidated emotion estimate of all the vision sub-systems. Moreover, the chatbot can request the current emotion index or the average emotion for a specified period. In our current deployment we have two vision sub-systems to capture and estimate the emotion of the audience.

2.3. Crowd Tracking via Mobile Devices

In addition to visual data collected of the users, the chatbot also collects gestural and movement data via users' mobile devices. This yields a lot of opportunities for the chatbot, including the possibility to play movement games and other embodied practices within the context of the physical system that the chatbot is embedded in. To do so, we develop the notion of a crowd index system.

The crowd index system communicates with the camera using the OSC protocol. The crowd index continuously receives the location data captured by the user device, via an intermediating software known as Max/MSP [38]. The crowd index system uses this data to make an estimate of the crowd behavior and shares it with the max system whenever it is requested. The system leverages the density-based clustering (DBSAN) algorithm to estimate the number of clusters formed by the

audience and to identify the presence of isolated audience members [39]. Moreover, the software computes the velocity and acceleration on a per audience basis, based on their real-time location data to get an estimate of the energy and agility among the audience.

2.4. System Communication via Max/MSP

The entire system is necessarily complex and thus needs to be coordinated with a global protocol and communication scheme to be effective. A key design consideration for us is that we need the system to be interactive rates (only a few seconds of latency), so that users do not get frustrated when interacting with the chatbot.

The Max/MSP system is the backbone for communication in ODO. It receives user chat, location and other sensory information and forwards it to the other destination systems for processing. The chatbot, along with vision sub-system and crowd index sub-system, processes the Max input and communicates back to Max with the result metric.

Max/MSP system uses OSC (Open Sound Control) for communication [6]. OSC adds a layer on top of a UDP (User Datagram Protocol) to standardize information transfer between media systems. Being a connection-less protocol, it decouples the chatbot from the Max/MSP, making the system modular. The vision sub-system communicates with the chatbot over TCP/IP (Transmission Control Protocol/Internet Protocol). This has been done to closely integrate the system with the chatbot, as the vision sub-system will only start after the chatbot. The crowd index system is loosely coupled with the chatbot via the OSC interface.

Figure 4 depicts the overall system call flow diagram. Upon starting, the chatbot triggers a request to the vision system to start the camera and receives an acknowledgement. Next the chatbot system sends a trigger to the Max system to start the conversation and receives the acknowledgement. The max system receives the user input over the OSC interface and forwards it to the chatbot to get a response. The response is then shared with the user. Additionally, the max system and the chatbot can request the vision system to get the emotion estimate of the audience and receives the response. Finally, whenever required, the max system requests for the crowd index system for the mobility related metric and receives a response.

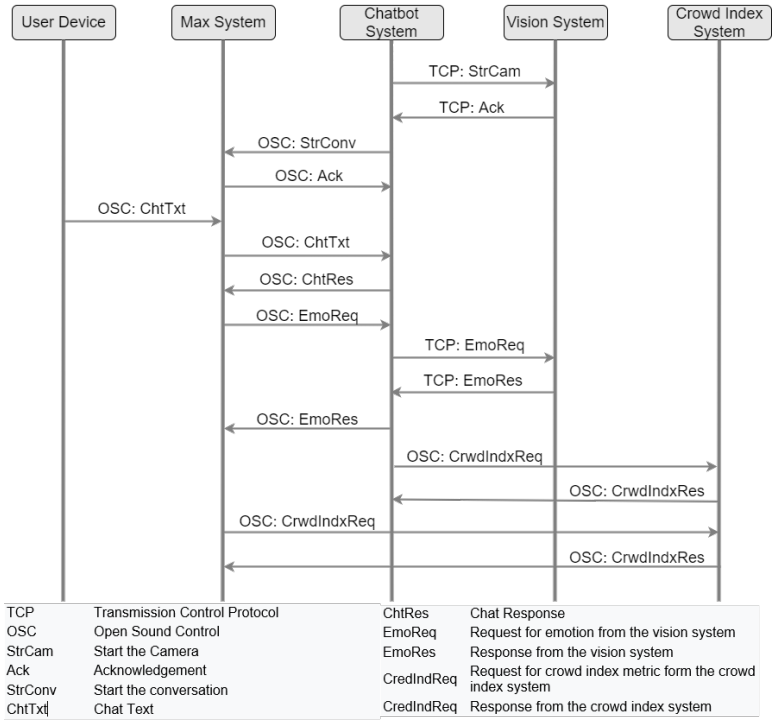


Figure 4. Call flow among the components of the ODO.

3. Implementation

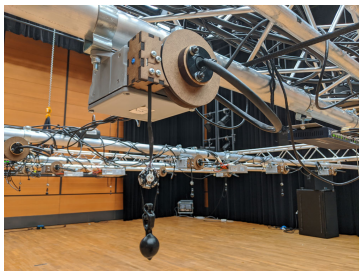
In this section, we discuss the technical implementation of the system and how we achieve the desired design goals for our chatbot. The overall system performs all of its computing locally on either the computers or the NVIDIA Jetson Nanos rather than user's mobile phones. This avoids the latency due to round trip back and forth from a cloud server due to streaming video, which we experienced in a previous iteration of the system using the Amazon DeepLens for our vision processing. Finally, we utilize LAN based communication to set up our own internal network in which to run the entire system. We now discuss the particulars of our system implementation below.

3.1. Stage Hardware

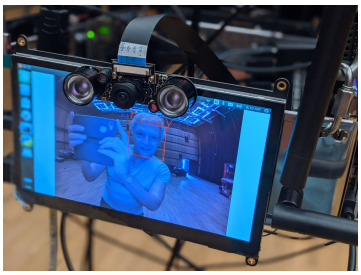
For the chatbot, we used 4 NVIDIA Jetson Nanos (NVIDIA Jetson Nano Developer Kit Item Model Nr 945-13450-0000-000, Arm Cortex A57) along with two 160 degree IR cameras (Waveshare IMX219-160 IR Camera). Each Nano is used to perform a certain task and is distributed to avoid overloading of computation on devices. The chatbot runs on the master Jetson Nano, which encapsulates the code for the chatbot, communication code for the 2 cameras, and communication code with the MAX system. Each of the cameras was connected to a separate Nano, which captures and processes emotions of the crowd. We also monitored movement of the crowd though a mobile application that communicates with the 4th Nano. Clustering of the crowd was calculated using their relative positions on the stage. Both crowd emotions and crowd clustering were then sent to the chatbot, where the chatbot engaged with the crowd depending on the crowd behavior.

Multimodal stage. The chatbot was embodied physically in a system on a black-box theatrical stage. The stage size was 8 m $\times$ 6 m with a height of 5 m. Motorized lights were hung from a scaffolding and truss system as shown in Figure 5a. The entire stage was covered with these double sided LED lights, connected to motors to move the lights vertically, which is shown in Figure 6, which immerses the audience in light and movement. This stage is similar to the forest3 stage by Ziegler et al. used in the dance production COSMOS [40]. The lights were controlled with ARTNET, a network protocol for theatrical control over UDP/IP [41]. The Max/MSP light and motor control software used GPU accelerated rendering techniques to compute the output animation of the physical LED units with minimal latency. All of the animations could be mixed before the final ARTNET output to the lighting network, affording artists a degree of variability in visual design without the need to implement or add new components to the software. This dedication to efficiency and flexibility enables rapid prototyping for designers of exhibitions and modalities of interaction.

Computers and cameras were hung from the center of the truss construction. The audience's position and movement data were sent to a central host, which relayed the data to the chatbot network. A network of 4 NVIDIA Jetson Nanos with one Mac computer communicated with Open Sound Control (OSC) [6] via LAN and WLAN. Five Android phones with a custom app sent text and motion data from each user to a host computer, which was coordinated through the Max/MSP system, as shown in Figure 5c. Two camera-based Linux systems fed face tracking data to the chatbot, which sent text and metadata of the conversation to the central host. An additional external crowd index computer permanently analyzed motion and position data from the audience and sent movement vectors and velocity data of each audience member to the host computer. The central computer fed two main displays on stage with "agility" information and a cluster index of the audience as a crowd with a cluster and separation index.



(**a**) Motorized LED



(**b**) Display with cameras at stage center

(**c**) Max software for communicating with mobile app

Figure 5. Stage hardware and software from the prototype system constructed.

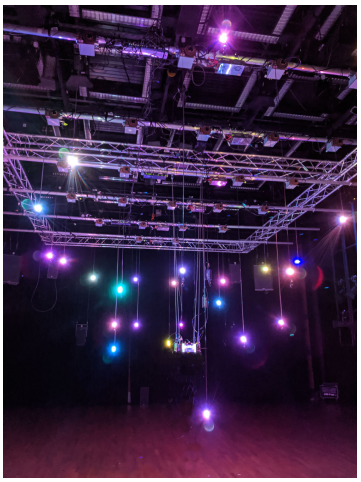


Figure 6. Moving LED Stage.

3.2. *Algorithm Details*

NLP architecture. Stories were divided into a set of different intents, which consists of a pair of utterances and responses. Utterances are the possible words/sentences that a user can use to converse with the chatbot, which triggers responses from the chatbot, which are story lines from different sections of the story. We utilized Google's Universal Sentence Encoder-USE [4] to understand users inputs. Universal Sentence Encoder provided the similarity index between the input and utterances. If a similarity index above the threshold (we have used 70%) is calculated, then the corresponding response of the matched utterance is triggered, otherwise the chatbot will request for a new input relevant to the story. This way, conversation is focused on a given topic of story and does not meander out. Figure 7 shows the correlation between user input text and saved utterances using the network. The utterance with the highest correlation coefficient is selected and the corresponding story intent is executed.

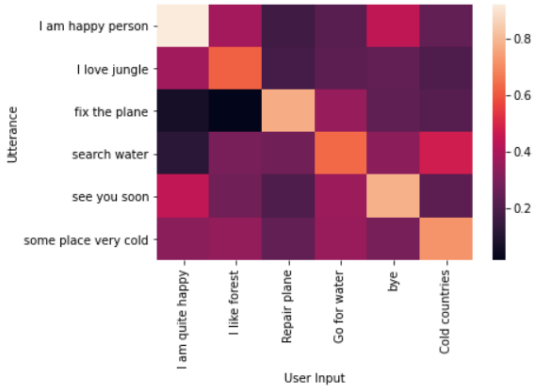
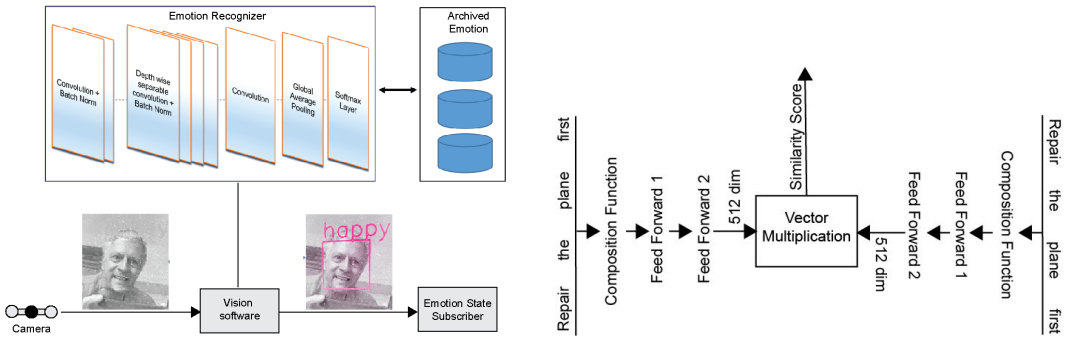


Figure 7. Sentence matching using the universal sentence encoder.

Universal Sentence Encoder uses Deep Averaging Network (DAN) [42] for matching saved utterances with user chat input. User sentences are converted in to word embeddings (low level mathematical representation of a word). DAN then combines all the words using composition function and calculates a vector average of these embeddings. These averages are then passed through one or more feed forward layers to learn the representation of the sentence and finally vectors are multiplied to get a similarity score. Figure 8b represents this architecture in detail.



(**a**) Vision (**b**) NLP

Figure 8. Vision and Natural Language Processing (NLP) architectures for the chatbot.

Since there can be the same/similar utterances for different intents corresponding to the different story-lines, it is difficult to identify the right intent as this scenario can trigger multiple intents. To avoid this, we utilized weighted intents to uniquely identify the correct intent to trigger in the story. As the story progresses, weights are updated and is tracked, this way the story always proceeds forward and does not loop back into previous story lines, even utterance matches.

Vision architecture. To detect the emotion we used a trained fully convolutional neural network-based mini-Xception model [5,43]. The mini-Xception architecture as proposed in [43] comprises of a fully convolutional neural network that comprises 4 residual depth-wise separable convolutions [44] with batch normalization [45], ReLU activations, and a last layer of global average pooling and a soft-max activation function to get the prediction of the emotion. Owing to elimination of the fully connected layers and use of depth-wise separable convolutions, the mini-Xception model has low latency in detecting the emotion, which is paramount in detecting emotion in a live video feed, and hence we use it for our vision sub-system. The architecture has approximately 60,000 parameters, which is much less compared to other architectures and henceforth makes it lightweight and fast in the prediction. The model was trained on the FER 2013 data set [46] and the accuracy was reported to be 66 percent. We found this accuracy to be suitable for our purposes as emotion data would only be periodically queried by the chatbot in the performance, and thus any mistakes in classifications were sparse in the actual performance. Future work could try to improve the emotion recognition

by finetuning the network on captured data from the system if absolute accuracy was desired from the performance.

Once the software detects the emotions for a frame it is archived along with the current timestamp. This archived information is used to make a response to the request for an emotion metric. To facilitate this, the vision sub-system software hosts a TCP/IP server and makes a response to the client by using the request–response messaging pattern. When the server receives a request from the chatbot, it shares the emotion metric with the chatbot.

Crowd index system. As depicted in Figure 4, the crowd index system has software running on an Nvidia Jetson Nano. It receives the real-time sensor information captured by the User Device. The received data comprises of the audience's real-time location. The crowd index receives this captured location of the audiences in real time. To compute the crowd index, we used the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [39] clustering algorithm, which is characterized by two parameters: ϵ and MinPoints. ϵ defines what point should be considered as within a proximity and MinPoints defines the minimum numbers of points required to form a cluster. We chose the distance metric as the Euclidian distance and the value of the ϵ is based on the dimension of the stage and a reasonable definition of proximity distance. The MinPoints were chosen as 2 and thus a cluster could be formed by two or more people. The noise points are used to identify an isolated audience member. In addition, we also calculated the velocity and acceleration of each user from their phone, which was also forwarded to the main system via OSC.

There may be some error introduced into the location estimation algorithms based on noise in the smartphone sensing. However, since our crowd index measure uses density-based clustering, the overall algorithm is resilient to individual errors in the localization, and the chatbot further only uses the crowd index to help guide the audience towards completing the interaction games at a coarse location granularity (meter scale in our implementation). Thus we did not find that noise significantly affects the system performance.

4. System Performance

4.1. Interaction and Games

One of the main user interactions that ODO performs is playing games with the audience members. Three-dimensional games usually are programmed for user interfaces, performing in virtual space. Using the LED lights described in the implementation section, the following games are implemented in the system using position and motion data from five users. Each of these games are triggered periodically in the story based on the playwright's script, as ODO is led out of the cave with the help of the audience.

- POINT Game: One user controls the z/height of one light by using position and z/height information via magnetic/pressure sensor information.
- LINE Game: Two users generate a line of lights between them controls the z/height of both ends of the line.
- WAVES game: The acceleration (x,y,z) information of all users is averaged and mapped to the speed of a Perlin noise generator controling motion movement of all 26 lights.
- PLANE game: Three users control the 3D orientation of a flat plane of lights. The host receives the x and y position of each user and z/height information via magnetic/pressure sensor information.
- LANDSCAPE game: Up to five users can move into the center field of stage between the 26 lights and build valleys and mountains using a terrain builder algorithm, which uses the user's x and y position and z/height information via magnetic/pressure sensor information in the APP to shape a 3D landscape terrain model.

4.2. ODO Conversation

ODO uses its multiple sensing modalities synergistically to interact with up to five given audience members at a time. A text-based conversation is limited like every other natural conversation to one partner, but in games and physical interactions the chatbot can interact with up to five users at the same time. At the start of the conversation, ODO asks each audience member for their names. ODO leads the conversation dialog. The audience members use either a text-based or voice-to-text application on their mobile phones to send the system input. These are stored in the order in which they are sent, and are responded to in turn by the system. Since ODO has multiple responses for any particular intent, there is a very low probability of repeated responses from ODO due to multiple similar inputs. However, having ODO respond to each participant in turn does limit the ability of the system to have long, extended conversations with each user. At the same time as the conversation, the apps also send movement and position sensor information continuously. Most of the script for ODO tends to encourage somatic and physical games for the audience leading through a story of a journey, while the chatbot cannot leave the stage physically. The play is based on the premise that ODO is aware of being stuck on stage. The chatbot needs to communicate and to play with the audience to grow as a character. At the end of the performance ODO thanks the audience for sharing time and space to "create a world together".

The vision cameras track up to five faces and perform emotion recognition on these faces. Typically the five emotion values are combined together via majority vote so that a single emotion value is held for a particular time period, which the system uses to inform its responses as a particular intent. The audience is perceived as one audience body. At the same time, the crowd index system delivers an estimate of the crowd activity, which is periodically sent every minute to the chatbot that will trigger unique responses in the output during the performance. Thus the chatbot alternates between responding to user input, emotions received from each vision system, and the crowd index with a combination of asynchronous (as they arrive) as well as synchronous (periodic emotion queries, etc.) behavior in the order in which these intents are triggered. In Figure 9, we show an example of the output of the system for a sample conversation of ODO. We can see ODO exhibiting several of the conversational strategies described earlier, including working with the emotion data to inform its responses at the end.

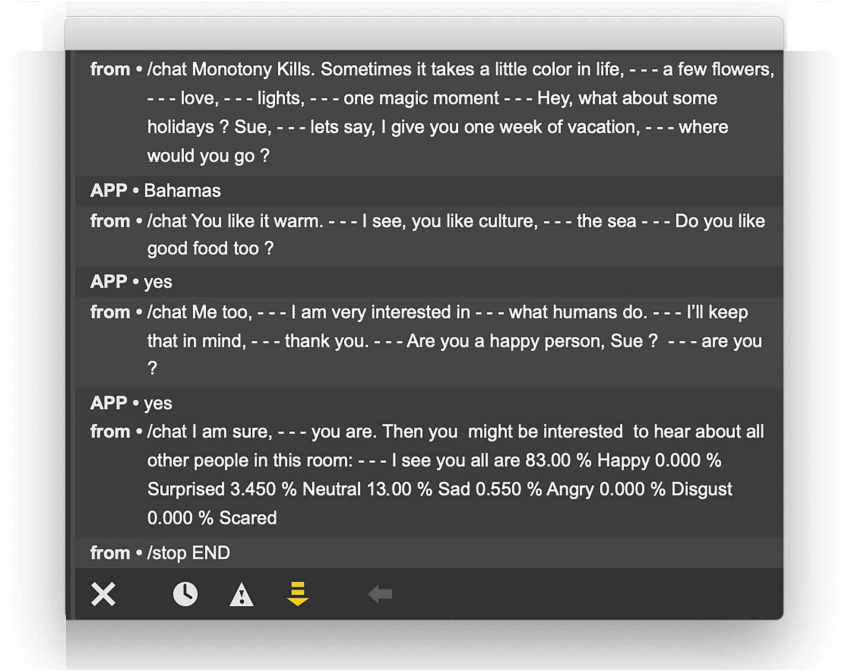


Figure 9. Example of the conversations.

4.3. Latency

We did experiments to quantify the latency of the system. We have used iPerf [47] to calculate the available bandwidth among the components of the overall system. We observed the available bandwidth to be 930 ± 5 Mbps among all the system components that need to communicate in the ODO's intranet. Since all the computing happens on the edge, we do not need internet communication. The latency including the request and response time is computed and is depicted in Table 1. As we show, the latency of most of the sub-systems are at interactive frame rates, and thus are not significantly noticeable for the human users. While it is difficult to compare these numbers to any benchmarks or baseline numbers, we note that milliseconds are well within the expectation for an interactive system, and the crowd-index, the only operation that is longer than a second, is only periodically queried every minute in the background by the system.

Table 1. Latency of different sub-systems in the chatbot. * includes sleep time of 1000 ms.

Average Latency and Bandwidth of the System	
Max-Chatbot	10 ms
Max-Sentence Matching	02 ms
Max-Haiku Poem	60 ms
Max-Emotions	42 ms
Max-Crowd Index	1470 ms *

5. Discussion

ODO is a theater production with a multimodal stage, which challenges the audience to play an active role as listeners and participants. In this production the audience plays a similar role compared to the chorus of ancient greek theater. The chorus is positioned in the "orchestra", which is the stage area between the audience and the center of the stage. The chorus of the ancient theater is the "translator" from the imagined world on stage to the world of the audience.

Our multimedia system has a lot of design implications for future work. We note that the primary novelty of this paper is not the individual components, as we use off-the-shelf vision, NLP, and mobile applications to build the system. However, the culmination of the components, implemented in the system, realizes a full interactive experience for users, with sensory input and feedback in terms of motion and lighting. The design of the chatbot and its conversational strategies forms an intermediate between convergent/task-based and divergent/open-ended designs, which can help inform the next generation of chatbots. The resulting system design has been achieved on low-cost edge computing hardware (NVIDIA Jetson Nanos) with minimal latency, and the actual physical mechanisms of the LED stage with trusses is portable and mobile, able to be installed in multiple venues with little effort. We believe this system can serve as an inspiration for artistic installations that leverage state-of-the-art AI and physical interaction into the system.

As we stated earlier, the contributions of this paper are to outline the design and implementation of ODO. Future work is needed to fully evaluate the efficacy of ODO as an experiential media system, and how it engages users who participate in the production. Putting the audience into an active role is a difficult task, because the audience expects an experience, watching a performance from a distance. To offer the reader some sense of the feedback given by the audience, we did informally ask their opinions after interacting with the system. The audience gave a mixed feedback on their experience, mentioning that doing active tasks during the performance also makes it difficult to enjoy the "linear" parts of the story. The experience, playing an active role was new to them. Some of them responded to have enjoyed a new role in this performance. They saw it as a new and different experience compared to a traditional theater watching experience. Both groups felt better prepared experiencing other interactive productions. This is good feedback as we go through the iterative design

process for this theatrical performance in the future, and work to evaluate the system from an HCI perspective. The theater installation "No Body lives here (ODO)" premiered July 2020 in Munich and also will be presented on the "Artificial Empathy" exhibition at the Center for Art and Media ZKM Karlsruhe, Germany in Fall 2020.

Author Contributions: Conceptualization, C.Z.; funding acquisition, C.Z.; investigation, R.B., K.K. and V.K.P.; project administration, S.J. and C.Z.; software, R.B., K.K., V.K.P., C.R. and B.M.; supervision, S.J. and C.Z.; writing—original draft, R.B., K.K. and V.K.P.; writing—review and editing, S.J. and C.Z. All authors have read and agreed to the published version of the manuscript.

Funding: The production was supported by the ZKM Karlsruhe (Hertz Laboratory), Arizona State University/Synthesis Center, the support program of the Landesverband Freier Theater Baden Württemberg e.V. with funds from the Ministry of Science, Research and Art, the Department of Culture of the City of Munich and the special program Konfigurationen of the Fonds Darstellende Künste Berlin. S.J. was supported by NSF grant EEC-1830730, as well as NEH grant AKA-265705-19.

Acknowledgments: We acknowledge the help of folks at the Synthesis Center at ASU for helpful discussions about this work.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Catania, F.; Spitale, M.; Fisicaro, D.; Garzotto, F. CORK: A COnversational agent framewoRK exploiting both rational and emotional intelligence. In Proceedings of the IUI Workshops, Los Angeles, CA, USA, 20 March 2019.
2. Zhou, L.; Gao, J.; Li, D.; Shum, H.Y. The design and implementation of xiaoice, an empathetic social chatbot. *Comput. Linguist.* **2020**, *46*, 53–93. [CrossRef]
3. Yin, J.; Chen, Z.; Zhou, K.; Yu, C. A Deep Learning Based Chatbot for Campus Psychological Therapy. *arXiv* **2019**, arXiv:1910.06707.
4. Cer, D.; Yang, Y.; Kong, S.Y.; Hua, N.; Limtiaco, N.; John, R.S.; Constant, N.; Guajardo-Cespedes, M.; Yuan, S.; Tar, C.; et al. Universal sentence encoder. *arXiv* **2018**, arXiv:1803.11175.
5. Arriaga, O.; Valdenegro-Toro, M.; Plöger, P. Real-time convolutional neural networks for emotion and gender classification. *arXiv* **2017**, arXiv:1710.07557.
6. Freed, A. Open sound control: A new protocol for communicating with sound synthesizers. In Proceedings of the International Computer Music Conference (ICMC), Thessaloniki, Greece, 25–30 September 1997.
7. Lehmann, H.T. *Postdramatic Theatre*; Routledge: Abingdon-on-Thames, UK, 2006.
8. Sarikaya, R. The technology behind personal digital assistants: An overview of the system architecture and key components. *IEEE Signal Process. Mag.* **2017**, *34*, 67–81. [CrossRef]
9. Greenshoot Labs. Available online: https://www.opendialog.ai/ (accessed on 1 May 2020).
10. Sabharwal, N.; Agrawal, A. Introduction to Google Dialogflow. In *Cognitive Virtual Assistants Using Google Dialogflow*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 13–54.
11. Sabharwal, N.; Barua, S.; Anand, N.; Aggarwal, P. Bot Frameworks. In *Developing Cognitive Bots Using the IBM Watson Engine*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 39–46.
12. Mayo, J. *Programming the Microsoft Bot Framework: A Multiplatform Approach to Building Chatbots*; Microsoft Press: Redmond, WA, USA, 2017.
13. Williams, S. *Hands-On Chatbot Development with Alexa Skills and Amazon Lex: Create Custom Conversational and Voice Interfaces for Your Amazon Echo Devices and Web Platforms*; Packt Publishing Ltd.: Birmingham, UK, 2018.
14. Gao, J.; Galley, M.; Li, L. Neural approaches to conversational ai. In Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, Ann Arbor, MI, USA, 8–12 July 2018; pp. 1371–1374.
15. Serban, I.V.; Sordoni, A.; Bengio, Y.; Courville, A.; Pineau, J. Building end-to-end dialogue systems using generative hierarchical neural network models. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016.
16. Sordoni, A.; Galley, M.; Auli, M.; Brockett, C.; Ji, Y.; Mitchell, M.; Nie, J.Y.; Gao, J.; Dolan, B. A neural network approach to context-sensitive generation of conversational responses. *arXiv* **2015**, arXiv:1506.06714.

17. Vinyals, O.; Le, Q. A neural conversational model. *arXiv* **2015**, arXiv:1506.05869.
18. Dale, R. The return of the chatbots. *Nat. Lang. Eng.* **2016**, *22*, 811–817. [CrossRef]
19. Brandtzaeg, P.B.; Følstad, A. Why people use chatbots. In Proceedings of the International Conference on Internet Science, Thessaloniki, Greece, 22–24 November 2017; pp. 377–392.
20. Følstad, A.; Brandtzæg, P.B. Chatbots and the new world of HCI. *Interactions* **2017**, *24*, 38–42. [CrossRef]
21. Purington, A.; Taft, J.G.; Sannon, S.; Bazarova, N.N.; Taylor, S.H. "Alexa is my new BFF" Social Roles, User Satisfaction, and Personification of the Amazon Echo. In Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems, Denver, CO, USA, 6–11 May 2017; pp. 2853–2859.
22. Ciechanowski, L.; Przegalinska, A.; Magnuski, M.; Gloor, P. In the shades of the uncanny valley: An experimental study of human–chatbot interaction. *Future Gener. Comput. Syst.* **2019**, *92*, 539–548. [CrossRef]
23. Hill, J.; Ford, W.R.; Farreras, I.G. Real conversations with artificial intelligence: A comparison between human–human online conversations and human–chatbot conversations. *Comput. Hum. Behav.* **2015**, *49*, 245–250. [CrossRef]
24. Sundaram, H. Experiential media systems. *ACM Trans. Multimed. Comput. Commun. Appl.* **2013**, *9*, 1–4. [CrossRef]
25. Jain, R. Experiential computing. *Commun. ACM* **2003**, *46*, 48–55. [CrossRef]
26. Davis, M. Theoretical foundations for experiential systems design. In Proceedings of the 2003 ACM SIGMM Workshop on Experiential Telepresence, Berkeley, CA, USA, 7 November 2003; pp. 45–52.
27. Chen, Y.; Sundaram, H.; Rikakis, T.; Ingalls, T.; Olson, L.; He, J. Experiential Media Systems—The Biofeedback Project. In *Multimedia Content Analysis*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 1–34.
28. Johnson, G.L.; Peterson, B.J.; Ingalls, T.; Wei, S.X. Lanterns: An Enacted and Material Approach to Ensemble Group Activity with Responsive Media. In Proceedings of the 5th International Conference on Movement and Computing, MOCO' 18, Genoa, Italy, 28–30 June 2018. [CrossRef]
29. Pinhanez, C.S.; Bobick, A.F. "It/I": A theater play featuring an autonomous computer character. *Presence Teleoperators Virtual Environ.* **2002**, *11*, 536–548. [CrossRef]
30. Knight, H. Eight lessons learned about non-verbal interactions through robot theater. In Proceedings of the International Conference on Social Robotics, Amsterdam, The Netherlands, 24–25 November 2011; pp. 42–51.
31. Hoffman, G.; Kubat, R.; Breazeal, C. A hybrid control system for puppeteering a live robotic stage actor. In Proceedings of the RO-MAN 2008—The 17th IEEE International Symposium on Robot and Human Interactive Communication, Munich, Germany, 1–3 August 2008; pp. 354–359.
32. Mazalek, A.; Nitsche, M.; Rébola, C.; Wu, A.; Clifton, P.; Peer, F.; Drake, M. Pictures at an exhibition: A physical/digital puppetry performance piece. In Proceedings of the 8th ACM Conference on Creativity and Cognition, Atlanta, GA, USA, 3–6 November 2011; pp. 441–442.
33. Meyer, T.; Messom, C. Improvisation in theatre rehearsals for synthetic actors. In Proceedings of the International Conference on Entertainment Computing, Eindhoven, The Netherlands, 1–3 September 2004; pp. 172–175.
34. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*; Neural Information Processing Systems Foundation, Inc.: Lake Tahoe, NV, USA, 2013; pp. 3111–3119.
35. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
36. Pandorabot. *Mitsuku*. Available online: https://www.pandorabots.com/mitsuku/ (accessed on 1 May 2020).
37. Viola, P.; Jones, M. Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001, Kauai, HI, USA, 8–14 December 2001; Volume 1, p. I.
38. Cycling74. Max/MSP. 1997. Available online: https://cycling74.com/ (accessed on 22 May 2020).
39. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the SIGKDD Conference on Knowledge Discovery and Data Mining, Portland, OR, USA, 2–4 August 1996; Volume 96, pp. 226–231.
40. Ziegler, C.; Sayaka Kaiwa, T.S.; Paquete, H. *COSMOS*. Presented at ZKM Center for Art and Media, Karlsruhe in 2018. Available online: https://zkm.de/en/event/2018/03/cosmos (accessed on 1 May 2020).
41. Art-Net. 2020 Artistic Licence Holdings Ltd., United Kingdom. Available online: https://art-net.org.uk/ (accessed on 1 May 2020).

42. Iyyer, M.; Manjunatha, V.; Boyd-Graber, J.; Daumé, H., III. Deep unordered composition rivals syntactic methods for text classification. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China, 26–31 July 2015; Volume 1: Long Papers, pp. 1681–1691.
43. Ayman, O. Emotion-Recognition. 2019. Available online: https://github.com/omar178/Emotion-recognition (accessed on 1 May 2020).
44. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
45. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv* **2015**, arXiv:1502.03167.
46. Goodfellow, I.J.; Erhan, D.; Carrier, P.L.; Courville, A.; Mirza, M.; Hamner, B.; Cukierski, W.; Tang, Y.; Thaler, D.; Lee, D.H.; et al. Challenges in representation learning: A report on three machine learning contests. In Proceedings of the International Conference on Neural Information Processing, Daegu, Korea, 3–7 November 2013; pp. 117–124.
47. Tirumala, A.; Qin, F.; Dugan, J.M.; Ferguson, J.A.; Gibbs, K.A. iPerf: TCP/UDP Bandwidth Measurement Tool. 2005. Available online: https://iperf.fr/ (accessed on 1 May 2020).