March 04, 2019

Projection Device for Generating Augmented Reality User Interfaces

Melissa Daniels

Nhat Vu

Tianchun Yang

Michael DelGaudio

Jiyoung Ko

Projection Device for Generating Augmented Reality User Interfaces

Inventors: Melissa Daniels, Nhat Vu, Tianchun Yang, Michael DelGaudio, and Jiyoung Ko

Summary

Aspects of the present disclosure are directed to a projection device that projects images onto surfaces and detects interactions with the images on the surfaces. For example, the projection device can include a projector that can project images onto a surface, a camera that can capture imagery of user interactions with the images on the surface, and a processing system (e.g., that includes one or more machine-learned models) that can process the imagery to understand the user interactions, thereby receiving user input. The images projected onto the surface can include still images, moving images, or a user interface that includes elements with which a user can interact. For example, the projection device can receive data descriptive of the user interface of a mobile device (e.g., smartphone) and can project the mobile device's interface onto the surface of a table, thereby allowing a user to interact with the mobile device interface on the surface of the table.

By projecting content in this fashion, the projection device makes the projected information more convenient, glanceable, ambient, and delightful. In particular, the projection device explores the relationship between surfaces, content, and objects and can serve, among other applications, as an augmented reality user interface which turns tables, shelves, floors, and/or ceilings into surfaces to paint in with glanceable, meaningful information and interfaces. The projection devices described herein take augmented/digital technology, apply it to the physical world, and translate this logic back in real-time.

As one example application, the projection device can be used to play a drawing game in which an artificial intelligence system attempts to identify an object drawn by a user. For example, the projection device can project a word describing an object onto a piece of paper or other surface

and prompt a user to draw a picture of the object described by word on the paper. The projection device can then use a camera to capture imagery of what the user draws on the paper. The projection device can analyze the drawing using a machine-learned model that has been trained to recognize objects. The projection device can then provide a guess with respect to which class of object was drawn by the user. The user can provide feedback regarding the accuracy of the guess generated by the projection device.

As other examples, the projection device can be used to provide a user interface for each of a number of different "channels," which can correspond to lightweight applications that can be configured through a companion application, hook into data feeds online, and display a projected user interface. One example channel which can be referred to as "Now Playing" can display information associated with music that is being ambiently played (e.g., by a Cast-enabled speaker). For example, the projection device can discover Cast devices on a local network using mDNS discovery and connect via the Cast protocol over the local network. As an example, the information associated with the music can include song/artist/album information, a music video associated with the music, an interactive equalizer, a visualizer or the like. The information can, for example, be projected onto a surface near the speaker(s) that are playing the music.

Another example channel includes a "Weather Caustics" channel that projects visualizations of information associated with live weather data at the user's location. For example, the companion application can send the location to the projection device when setting up. The projection system can then pull weather data from OpenWeatherMap and display the appropriate visualization. One example visualization includes ambient water motion reflections that react to the local weather.

Yet another example channel includes a “Space Porthole” that uses the user’s local latitude/longitude and a database of stars and constellations to project the night sky in real time on the user’s ceiling, thereby illustrating the stars that would visible to the user from their current location on a clear night. The visualized stars can include various information overlays like constellation maps, brightness/distance information, etc.

Another example channel includes an “Augmented Clock” channel that makes the classic wall clock smart. A user’s calendar can be added to the projection device to see the day’s events as an overlay around a studio clock. For example, the companion application can be used to add a publicly-available calendar URL. The projection device can pull the information periodically and show events for the next 12 hours. The events can be projected onto a portion of the surface that is adjacent to the time of the event as shown on the clock.

In another example, channels can be user-configured or user-generated. Thus, the projection device can be part of a platform that allows users to build more customized channels, in addition or alternatively to the built-in channels.

As such, the projection device can operate as an interface to a number of different applications such as, for example, a guessing game in which the projection device employs a camera and machine-learned model(s) to guess what a user has drawn on a piece of paper or other medium. The projection device can include light projector to project images onto surfaces, a camera to detect interactions with the projected images, and a machine-learned model that can recognize user interactions with the projected images. Furthermore, the projection device can be built using an embedded operating system platform that can interact with Internet of Things devices.

The input to the projection device can be received from any number of different devices such as various IoT devices included in an environment with the projection device. Thus, in one example, the projection device can serve as a callable interface for any number of different applications and/or devices scattered about an environment such as a user's home. This may be particularly beneficial for small, embedded IoT devices which do not themselves have displays or other interface components. In some embodiments, the IoT devices can interact with the projection device using an application programming interface (API) to enable the projection device to serve as an input/output interface for the IoT device to interact with a user. As one example, on some embodiments, a separate IoT device (e.g., a home security device) can take the initiative to interact with the projection device to cause the projection device to project a user interface associated with the home security device (e.g., to alert a user that a door has been left ajar). In another example, the user can interact with the companion application to cause the projection device to project the user interface associated with a user-specified IoT device. For example, a user can request the projection device to project the home security system user interface to assess the security status of his home.

In addition, in some embodiments, a built-in accelerometer can be used to identify an orientation of the projection device. In such fashion, different projections/interfaces can be provided as the projection device is oriented to face different surfaces. Thus, a user can change the channel of a projection device simply by rotating the projection head. In some embodiments, the visual recognition system of the projection device can be applied to recognize objects (e.g., a clock) in a current projection field of view of the projector and the projection device can determine which of a number of available projections/channels is appropriate in response to the recognized objects (e.g., the Augmented Clock channel can be projected in response to detection of the clock).

Thus, aspects of the present disclosure are directed to a system that includes a projector, sensors, and a machine-learned model to create an interface that enables human-machine interactions. Furthermore, in some implementations, machine-learning techniques can be used to improve the effectiveness with which the projection device processes user input into the interface such as, for example, recognizing what a user has drawn on a piece of paper.

Example Figures

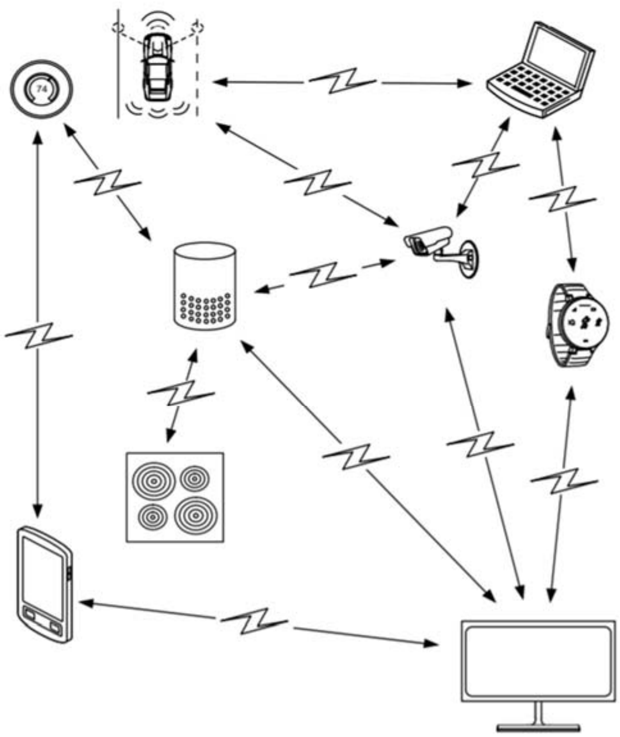

Fig. 1

Fig. 3

Detailed Description

The present disclosure is directed to a projection device that can project interactive images onto surfaces and which can be used as the input and output device for an application (e.g., an application that leverages a machine-learned model to process received user input). Further, the projection device can operate as an Internet of Things (IoT) device within an IoT environment.

Referring first to Fig. 3 which is included in this document, example embodiments of projection devices are described herein. The projection device can include a computing device that uses an operating system that can operate on embedded hardware of the projection device. Further, the projection device can include a projector that projects an image onto a surface and a camera that detects an area including the image that is projected onto the surface. In some embodiments, the projector can include a lamp that has been combined with a 3D-printed housing and off-the-shelf components such as a Raspberry Pi, laser projector, an accelerometer, and various cables.

The projection device can receive an input from an external device (e.g., a smart phone) that can provide images for the projection device. For example, a smartphone can be connected to the projection device via a wireless or wired connection (e.g., HDMI cable) through which a user interface can be projected using the projection device.

In a particular example embodiment that enables the user to participate in a drawing game, the projection device can project an arrow pointing to a physical button and text indicating that the user should “Push the button to start.” In this example, the instructions to the user are projected onto a pad of paper and the button the user is directed to press is able to communicate with the projection device and signal when the button has been pressed, which launches an application (e.g., a drawing game).

In Fig. 4, the projection device is shown projecting an image of a hand and a button as well as text indicating the title of a game (“Quick, Draw) and a prompt for the user to “Push the button to start". In this example, the instructions to the user are projected onto a pad of paper and the button the projection device can detect the user's interaction with the images and text on the pad of paper.

In Fig. 5, the projection device is depicted projecting a prompt for the user to "Draw a carrot". The projection device can detect any images drawn by the user via the camera which is directed to the surface of the pad of paper on which the user can draw with the pencil. The images drawn by the user can be used as an input for the machine-learned model of the projection device.

In some implementations, the projection device can include a machine-learned model that has been trained to perform various operations including object and/or handwriting recognition. For example, the projection device can be trained to recognize objects in images as well as movements (e.g., pen strokes or finger gestures including pressing and sliding the surface) including movements performed in the process of producing an image. Based on recognition of the image or movement, the projection device can perform operations including displaying still images and text, displaying moving images and text, and performing operations associated with a user interface including detection of user interaction with user interface elements (e.g., drawing, writing, pressing, sliding, and hovering over graphical user elements associated with performance of some operation by the projection device or a device associated with the projection device). Thus, the projection system can include a vision system that leverages machine learning techniques to process captured imagery to understand human interactions with the projected interface.

Fig. 7 depicts the output of the machine-learned model of the projection device after being provided with the input of an image of a carrot drawn by the user. As shown, the projection device has provided an indication of what the projection device determined to be the image drawn by the user. In this example, the projection device has correctly determined that the user drew an arrow on the pad of paper.

Fig. 8 depicts a high-level representation of the way in which the machine-learned model (e.g., a neural network) of the projection device parses the image drawn by the user into a simplified figure comprising a set of points and straight lines. Based on the geometry of the simplified figure, the machine-learned model can determine that the proportions of the user drawing correspond to those of a carrot. Thus, in some embodiments, the system can provide an illustration of "What the machine saw" (e.g., as part of a digital takeaway described below). This information can assist the user in understanding which aspects of the user's drawings led to the guess made by the projection device. For example, as shown in Fig. 8, the user is provided with an illustration that shows which aspects of the doodled carrot led to the machine understanding that the doodle depicted a carrot. Thus, aspects of the present disclosure can enable enhanced human understanding of and collaboration with machine learning systems.

In some embodiments, during and/or after the user drawing, the projection device can depict a QR code projected onto the pad of paper. The QR code can link to a website that provides a digital takeaway including further information about the projection device. For example, the QR code can link to a website including details on how to build the projection device, other interesting applications for the projection device, and further information about how the projection device operates (e.g., details of how the machine-learned model operates). One example of this digital takeaway is shown in Fig. 9.

From a technical perspective, some embodiments of the projection system described herein can be built using OpenCV. The camera can track pen tip movement and/or actual ink (or other media) on the paper. The movement tracking can reduce the possibility of hand blockage and the ink detection makes sure certain movements end up with ink on the paper. The parsed ink then used as input of the Quick, Draw! doodle recognition machine learning model.

Figs. 10-13 show example depictions of the Now Playing, Weather Caustics, Augmented Clock, and Space Porthole channels, respectively.

Altogether, the projection system demonstrates combining physical reality with augmented reality in a playful, mixed experience. It also showcases that not only can machine learning models can be run on-device with Android Things, they can be combined with cool projection capabilities and exploring the use of projected UI in the Internet of Things.

Referring now to Fig. 1, Fig. 1 depicts a block diagram of an example IoT environment according to example implementations of the present disclosure. As illustrated in Fig. 1, in some implementations, the IoT environment includes a plurality of different devices, each of which can be referred to as an IoT device. An example IoT device can be an intelligent, environmentally-sensing, and/or network-connected device configured to communicate with a central server or cloud service, a control device, and/or one or more additional IoT devices to perform any number of operations (e.g., in response to received commands). IoT devices can, in some instances, also be referred to as or include "smart" devices and/or "connected" devices.

Each IoT device can be a stand-alone physical device or can, in some instances, be an embedded device that is embedded within a larger device or system. Each IoT device can include electronics, software, sensors, actuators, and/or other components, including various components that sense, measure, control, and/or otherwise interact with the physical world. An IoT device can further include various components (e.g., a network interface, antennas, receivers, and/or the like) that enable the IoT device to send and/or receive data or other information from one or more other IoT devices and/or to a central system.

In particular, the various IoT devices can be communicatively connected to one another via one or more communications networks. The networks can be wide area networks, local area networks, personal area networks, piconets, cellular networks, other forms of networks, and/or combinations thereof. The networks can be wired and/or wireless. The networks can be private and/or public. As examples, two or more of the IoT devices can communicate with one another using a Wi-Fi network (e.g., a home network), Bluetooth, Bluetooth Low Energy, Zigbee, Radio-Frequency Identification (RFID), machine to machine connections, inductive communications, optical communications, infrared communications, other communications techniques or protocols, and/or combinations thereof. For example, an IoT device might communicatively connect with a first nearby device using Bluetooth while also communicatively connecting with a second nearby device using Wi-Fi.

In some implementations, each IoT device can have a unique identifier. For example, the identifier for each IoT device can include and/or be based on an Internet Protocol address associated with such IoT device, a manufacturer associated with such IoT device, a location at which such IoT device is positioned, a model number of such IoT device, a functionality of such IoT device, and/or other device characteristics. In some implementations, a given IoT device can locate and/or communicate with another IoT device based on its unique identifier. In some implementations, the identifier assigned to an IoT device can be modified by a user and/or owner of such IoT device.

In particular, in some implementations, a user can assign one or more identifiers to the IoT devices within a device topology representation. The device topology representation can describe and/or organize a group of IoT devices (e.g., based on location with one or more structures such as one or more homes, offices, vehicles, and/or the like). The identifiers can be chosen by the user and associated with the respective IoT devices within the device topology representation. The identifier(s) can include but are not limited to names, nicknames, and/or aliases selected for the IoT devices by the user. In this manner, the identifiers can be names or aliases of the respective IoT devices that the user is likely to use when identifying the IoT devices for requested control or command operations (e.g., "turn on the kitchen lights").

An IoT device can be mobile or can be stationary. In some implementations, an IoT device can be capable of autonomous or semi-autonomous motion.

In some implementations, an IoT device can be controlled or perform operations in response to communications received by the IoT device over a network. For example, an IoT device can be controlled by a control device that is communicatively connected to the IoT device. The control device can communicate directly with the IoT device or can communicate indirectly with the IoT device (e.g., over or using a mesh network). The control device can itself be an IoT device or the control device can be a device that is not considered part of the IoT environment. For example, the control device can be a server device that operates as part of cloud computing system. The commands can be in response to or generated based on a user input or can be generated without user input.

Thus, in one example, an IoT device can receive communications from a control device and the IoT device can perform operations in response to receipt of such communications. The performed operations can be internal operations (e.g., changing an internal setting or behavior) or external operations (e.g., interacting with the physical world in some way). The IoT device and the control device can be co-located or can be remotely located from each other.

As an example, the control device can be or include a user device such as a smartphone, tablet, computing device that is able to be worn, laptop, gaming console or device, virtual or augmented reality headset, and/or the like. As another example, the control device can be a server computing device. As another example, the control device can itself be an IoT device. For example, the control device can be a so-called “smart speaker” or other home control or automation device.

In some implementations, a user can interact with a control device (e.g., which can be an IoT device) to input data into or otherwise control the IoT environment. For example, the control device can include and execute a software application and/or other software programs that provide a user interface that enables entry of user input. The software applications can be executed entirely at the control device or can be web applications where some portion of the program or functionality is executed remotely (e.g., by a server connected over the Internet) and, in some implementations, the client-side logic runs in a web browser. Thus, in some implementations, a web server capable of sending, receiving, processing, and storing web pages or other information may be utilized.

In some implementations, a cloud service may be used to provision or administer the IoT devices. For example, a cloud computing system can enable or perform managed and/or integrated services that allow users to easily and securely connect, manage, and ingest IoT data from globally dispersed IoT devices at a large scale, process and analyze/visualize that data in real time, and/or implement operational changes and take actions as needed. In particular, in some implementations, the cloud computing system can employ a publication subscription model and can aggregate dispersed device data into a single global system that integrates seamlessly with data analytics services. An IoT data stream can be used for advanced analytics, visualizations, machine learning, and more to help users improve operational efficiency, anticipate problems, and/or build rich models that better describe and optimize the user’s home or business. The cloud system can enable any number of dispersed IoT device to connect through protocol endpoints that use automatic load balancing and horizontal scaling to ensure smooth data ingestion under any condition.

In some implementations, the cloud system can include or implement a device manager. For example, the device manager can allow individual IoT devices to be configured and managed securely in a fine- or coarse-grained way. Management can be done through a console or programmatically. The device manager can establish the identity of a device and can provide the mechanism for authenticating a device when connecting. The device manager can also maintain a logical configuration of each device and can be used to remotely control the device from the cloud.

In some implementations, an IoT device can include an artificial intelligence-based assistant or software agent. A user can interact with the artificial intelligence-based assistant via a control device, directly through the IoT device, or any other method of interaction. The artificial intelligence-based assistant can perform tasks or services based on user input and/or contextual awareness (e.g., location awareness), including acting as a control device to control other IoT devices. In some implementations, an IoT device (e.g., an artificial intelligence-based assistant on such device) can access information from a variety of online sources (such as weather conditions, traffic congestion, news, stock prices, user schedules, retail prices, etc.).

The artificial intelligence-based assistant or software agent can be stored and implemented by a single device (e.g., a single IoT device) or can be spread across multiple devices and implemented by some (e.g., dynamically changing) combination of such multiple devices.

In some implementations, an IoT device can include (e.g., as part of an artificial intelligence-based assistant) one or more machine-learned models that assist in understanding user commands, determining context, and/or other actions. Example machine-learned models include artificial neural networks such as feed-forward neural networks, recurrent neural networks, convolutional neural networks, autoencoders, generative adversarial networks, and/or other forms, structures, or arrangements of neural networks. Additional example machine-learned models include regression models, decision tree-based models (e.g., random forests), Bayesian models, clustering models, linear models, non-linear models, and/or other forms, structures, or arrangements of machine-learned models. Machine-learned models can be trained using supervised learning techniques or unsupervised learning techniques. Machine-learned models can be stored and implemented on the IoT device or can be stored and implemented in the cloud and the IoT device can leverage the models by communicating with cloud devices. Feedback or other forms of observed outcomes can be used to re-train models to improve their performance. Models can be personalized to one or more users or environments by re-training on data specific to such users or environments.

In some implementations, the artificial intelligence-based assistant can perform concierge-type tasks such as, for example, making dinner reservations, purchasing event tickets, making travel arrangements, and/or the like. In some implementations, the artificial intelligence-based assistant can provide information based on voice input or commands (e.g., by accessing information from online sources). In some implementations, the artificial intelligence-based assistant can automatically perform management or data-handling tasks based on online information and events, including, in some instances, without user initiation or interaction.

In some implementations, a control device (e.g., which may be an IoT device) can include components such as a mouse, a keyboard, buttons, knobs, a touch-sensitive component (e.g., touch-sensitive display screen or touch pad), and/or the like to receive input from the user via physical interaction.

In some implementations, the control device can include one or more microphones to capture audio signals and the device can process the audio signals to comprehend and respond to audio commands (e.g., voice commands) provided by a user or by some other device. Thus, in some implementations, the IoT devices can be controlled based on voice commands from a user. For instance, a vocalization from a user can be received by a control device. The vocalization can be a command spoken by a user proximate to the control device. The control device can control itself and/or one or more of the IoT devices based on the vocalization.

In some implementations, one or more vocalization(s) may be used as an interface between a user and an artificial intelligence-based assistant. For example, a user may vocalize a command which the artificial intelligence-based assistant may identify, process, and/or execute or cause to be executed. The vocalized command may be directed at the artificial intelligence-based assistant.

As one example, the vocalization may indicate a user desire to interact with or control another IoT device (e.g., lowering a thermostat setting, locking a door, turning off a light, increasing volume, etc.). The artificial intelligence-based assistant may communicate the command to the desired IoT device which can respond by executing or otherwise effectuating the user command. As another example, the vocalization can include a user commanding the artificial intelligence based assistant to perform a task (e.g., input an event into a calendar, retrieve information, set a reminder, make a list, define a word, read the first result of an internet search, etc.).

In some implementations, speech recognition or processing (e.g., natural language processing) can be performed on the vocalization to comprehend the command provided by the vocalization. For instance, data indicative of the vocalization can be provided to one or more language models (e.g., machine-learned models) to determine a transcription of or otherwise process the vocalization.

In some implementations, processing the vocalization or other user input can include determining one or more IoT devices to control and/or one or more actions to be performed by the selected IoT devices. For instance, a semantic interpretation of the vocalization (e.g., a transcript of the vocalization) can be determined using one or more semantic interpretation techniques (e.g., natural language processing techniques). The semantic interpretation can provide a representation of the conceptual meaning of the vocalization, thereby also providing an interpretation of the intent of the user.

In some implementations, the interpretation of the vocalization can be determined based at least in part on the device topology representation. For instance, the device topology representation can be accessed to determine the one or more selected IoT devices and/or actions to be performed. As one example, the device topology representation can be accessed and compared against a transcription of the vocalization to determine a match between one or more terms included in the transcription and one or more terms associated with the IoT device topology representation (e.g., “kitchen lights”). In some implementation, the identity of the speaker can be ascertained and used to process the vocalization (e.g., such as to process commands that include possessive modifiers: “brew a cup of my favorite roast of coffee”).

In some implementations, the control device (e.g., which may be an IoT device) can include a vision system that includes one or more image sensors (e.g., visible-spectrum cameras, infrared cameras, LIDAR systems, and/or the like) that capture imagery. The device can process the imagery to comprehend and respond to image-based commands or other input such as, for example, gesture commands provided by the user. In some implementations, the vision system may incorporate or perform facial movement identification (e.g. lip reading) capabilities while, in other implementations, the vision system may additionally or alternatively incorporate hand shape (e.g. hand gestures, sign language, etc.) identification capabilities. Facial movement and/or hand shape identification capabilities may allow a user to give commands a control device in addition or alternatively to voice control.

In some implementations, in response to the image data of the facial or hand gesture, the control device can determine one or more IoT devices to control and/or one or more actions to be performed (e.g., by the selected IoT devices). Interpretation of image data that depicts lip reading and/or sign language may be achieved through any method of image data analysis. The interpretation can provide a representation of the conceptual meaning of the image data. In this manner, the interpretation of the image data can provide an interpretation of the intent of the user in performing the gesture(s).

In some implementations, the interpretation can be determined based at least in part on the device topology representation. For instance, the device topology representation can be accessed to determine the one or more selected IoT devices and/or the action to be performed. For example, the device topology representation can be accessed and compared against the image data to determine a match between one or more aspects of the image data and one or more aspects associated with the IoT device topology representation (e.g., the user may be pointing to a specific IoT device when providing a voice command or a gesture command).

In further implementations, gaze recognition can be performed on the captured imagery to identify an object or device that is the subject of a gaze of the user. A user command (e.g., voice or gesture) can be interpreted in light of (e.g., as applied to) the object or device that is the subject of the gaze of the user.

In some implementations, the vision system may be used as an interface between a user and an artificial intelligence-based assistant. The captured image data may be interpreted and/or recognized by the artificial intelligence-based assistant.

In some implementations, the selected IoT devices and/or the actions to be performed can be determined based at least in part on contextual data (e.g., location of user, day of the week, user data history, historical usage or command patterns, user wardrobe, etc.) For instance, in response to receiving a command from a user, a location of the user, a time of day, one or more past commands, and/or other contextual information can be determined. The location can be determined using various suitable location determination techniques. The location determination technique can, for example, be determined based at least in part on the control device to which the user provides the vocalization.

As one example, if the control device is an IoT device that is specified in the device topology representation, the user location can be mapped to the structure and/or room to which the control device is assigned in the device topology representation. As another example, if the control device is a user device not specified in the device topology representation, the user location can be determined using one or more location determination techniques, such as techniques using wireless access points or short range beacon devices associated with one or more IoT devices, and/or other suitable location determination techniques. In some implementations, the user location can be mapped to one or more structures and/or rooms specified in the device topology representation. In some implementations, the control device and/or other IoT devices can also process audio signals and/or imagery to comprehend and respond to contextual information. As examples, triangulation and/or beamforming techniques can be used to determine the location of the user based on receipt of the voice command at multiple different audio sensors. In some implementations, multiple possible user commands or requests can be disambiguated based on the contextual information.

Further to the descriptions above, a user may be provided with controls allowing the user to make an election as to both if and when systems, devices, or features described herein may enable collection of user information (e.g., information about a user's preferences, a user's activities, or a user's current location), and if the user is sent content or communications from a server. In addition, certain data may be treated in one or more ways before it is stored or used, so that personally identifiable information is removed. For example, a user's identity may be treated so that no personally identifiable information can be determined for the user, or a user's geographic location may be generalized where location information is obtained (such as to a city, ZIP code, or state level), so that a particular location of a user cannot be determined. Thus, the user may have control over what information is collected about the user, how that information is used, and what information is provided to the user.

Fig. 2 provides a block diagram of an example software stack that can be included on an IoT device. The software stack shown in Fig. 2 is provided as one example only. Various different IoT devices can have any number of different software and/or hardware configurations which may be of greater or lesser complexity to that shown in Fig. 2.

In some implementations, an IoT device can include and execute one or more computer applications (also known as software applications) or other computing programs. The IoT device can execute the application(s) to perform various functions, including collection of data, communication of data, and/or responding to or fulfilling received commands. In some implementations, the software applications can be native applications.

In some implementations, the software application(s) on an IoT device can be downloaded and installed by or at the direction of the user. In other implementations, the software application(s) can be default applications that come pre-programmed onto the IoT device. In some implementations, the software application(s) can be periodically updated (e.g., via download of update packages). The software application(s) can be closed source applications or can be open source applications. The software applications can be stand-alone applications or can be part of an operating system of the IoT device. The software applications can be embodied in computer-readable code or instructions that are stored in memory and then accessed and executed or followed by one or more processors of the IoT device.

In some implementations, the software application(s) on an IoT device can be user-facing applications such as a launcher or a browser. In other implementations, the IoT device does not include any user-facing applications but, for example, is instead designed to boot directly into applications developed specifically for the device.

More particularly, in some implementations, an IoT device can include or otherwise be implemented upon or in conjunction with an IoT platform that includes a number of elements. The IoT platform can include an operating system. The operating system can, for example, have been optimized for use in the IoT environments (tuned for faster boot times and/or lower memory footprint). The operating system and other platform elements may be able to receive secure and managed updates from the platform operator. The IoT platform can include hardware that is accessible and easy to integrate.

The IoT platform can also enable application developers to build applications using a rich framework provided by an operating system software development kit (SDK) and platform services, including, for example, the same user interface toolkit, multimedia support, and connectivity application programming interfaces (APIs) used by developers of mobile applications for larger devices such as smartphones. Applications developed for the IoT device can integrate with various services using one or more client libraries. For example, the applications can use the libraries to interact with services such as application deployment and monitoring services, machine learning training and inference services, and/or cloud storage services. The applications can use the APIs and/or support libraries to better integrate with hardware, including, for example, custom hardware. This can include support for peripheral interfaces and device management. The device can also include a number of native libraries, including, for example, C/C++ libraries, runtime libraries, core libraries, and/or the like. Updates to one or more of these components can be deployed over the air and/or automatically when updates are available.

In some implementations, an IoT device (e.g., the software applications executed thereby) can utilize APIs for communicating between a multitude of different software applications, operating systems, databases, libraries, enterprises, graphic interfaces, or any other component of the IoT environment disclosed herein. For instance, a first software application executed on a first IoT device can invoke a second software application via an API call to launch the second software application on a second IoT device.

In some implementations, the applications can run on a single or variety of operating system platforms including but not limited to OS X, WINDOWS, UNIX, IOS, ANDROID, SYMBIAN, LINUX, or embedded operating systems such as VxWorks.

The IoT device can include one or more processors and a memory. The one or more processors can be any suitable processing device (e.g., a processor core, a microprocessor, an application specific integrated circuit (ASIC), a field programmable gate array (FPGA), System on a Chip (SoC), a controller, a microcontroller, etc.) and can be one processor or a plurality of processors that are operatively connected. The memory can include one or more non-transitory computer-readable storage mediums, such as RAM, ROM, EEPROM, EPROM, flash memory devices, magnetic disks, etc., and combinations thereof. The memory can store data and instructions which are executed by the processor to cause the IoT device to perform operations. The IoT devices can, in some instances, include various other hardware components as well, including, for example, a communications interface to enable communication over any number of networks or protocols, sensors, and/or other components.

In some implementations, the IoT device can include or be constructed using one or more System on Module (SoM) architectures. Each SoM can be a fully integrated component that can be dropped directly into a final design. Modules can be manufactured separately and combined to form the device. In some implementations, the device software can include a hardware abstraction layer and kernel which may be packaged as a board support package for the modules. In other implementations, different, non-modular architectures can be used.

Example IoT devices include or can be associated with an air conditioning or HVAC system, lighting device, a television or other home theater or entertainment system, security system, automatic door or window locking system, thermostat device, home energy manager, home automation system, audio speaker, camera device, treadmill, weight scale, smart bed, irrigation system, garage door opener, appliance (e.g., refrigerator, dishwasher, hot water heater, furnace, stove, fireplace, etc.), baby monitor, fire alarm, smoke alarm, medical devices, livestock tracking devices, cameras, beacon devices, a phone (e.g., smartphone), a computerized watch (e.g., a smart watch), a fitness tracker, computerized eyewear, computerized headwear (e.g., a head mounted display such as a virtual reality of augmented reality display), other types of computing devices that are able to be worn, a tablet, a personal digital assistant (PDA), a laptop computer, a desktop computer, a gaming system, console, or controller, a media player, a remote control, utility meter, an electronic book reader, a navigation system, a vehicle (e.g., car, boat, or plane/drone) or embedded vehicle system, an environmental, food, or pathogen monitor, search and rescue devices, a traffic control device (e.g., traffic light), traffic monitor, climate (e.g., temperature, humidity, brightness, etc.) sensor, agricultural machinery and/or sensors, factory controller, GPS receivers, printers, motor (e.g., electric motor), and/or other suitable device or system.

The technology discussed herein makes reference to servers, databases, software applications, and other computer-based systems, as well as actions taken and information sent to and from such systems. One of ordinary skill in the art will recognize that the inherent flexibility of computer-based systems allows for a great variety of possible configurations, combinations, and divisions of tasks and functionality between and among components. For instance, server processes discussed herein may be implemented using a single server or multiple servers working in combination. Databases and applications may be implemented on a single system or distributed across multiple systems. Distributed components may operate sequentially or in parallel.

Figures

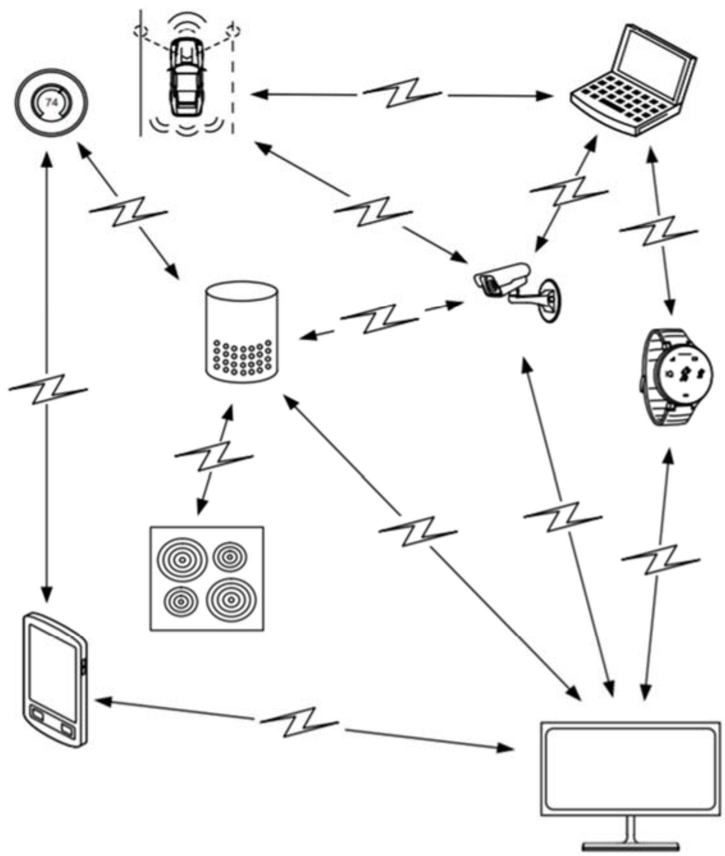

Fig. 1

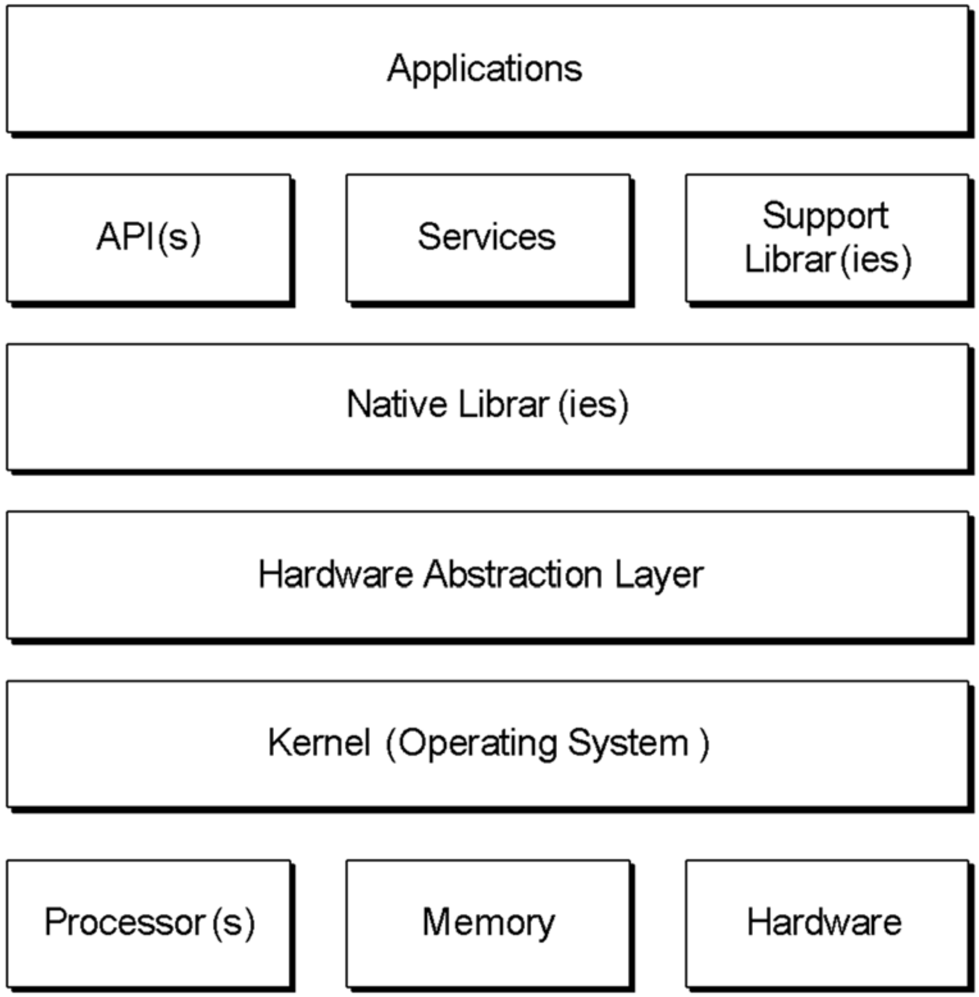
Applications
API(s)
Services
Support Librar(ies)
Native Librar(ies)
Hardware Abstraction Layer
Kernel (Operating System)
Processor(s)
Memory
Hardware

Fig. 2

Fig. 3


androidthings
QUICK, DRAW!
Push the button to start…

Fig. 4

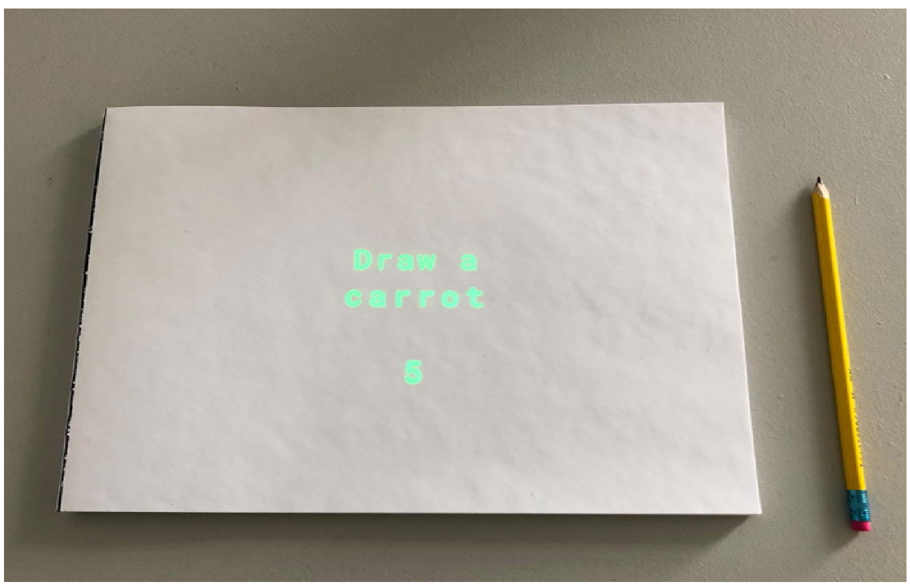
Draw a
carrot
5

Fig. 5

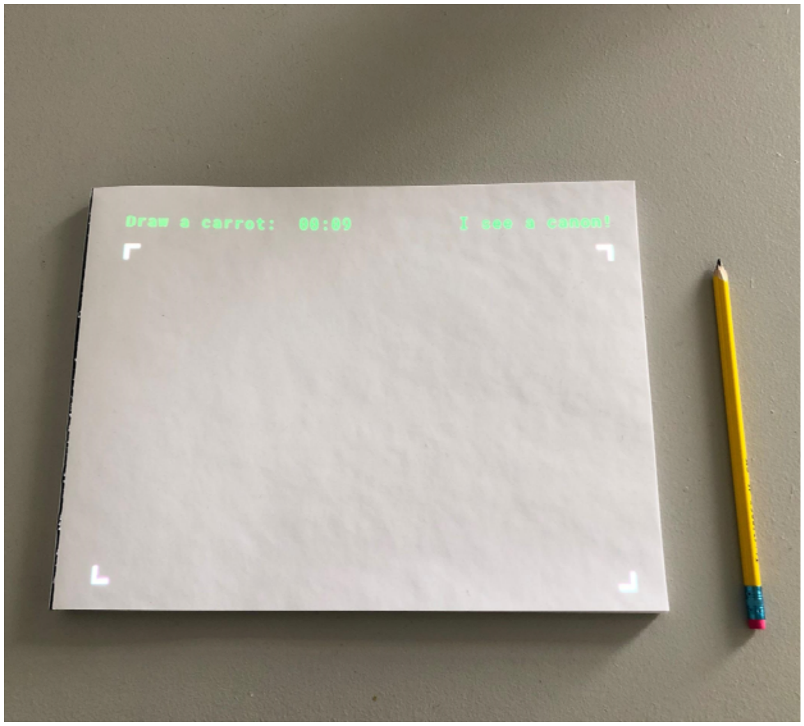

Fig. 6

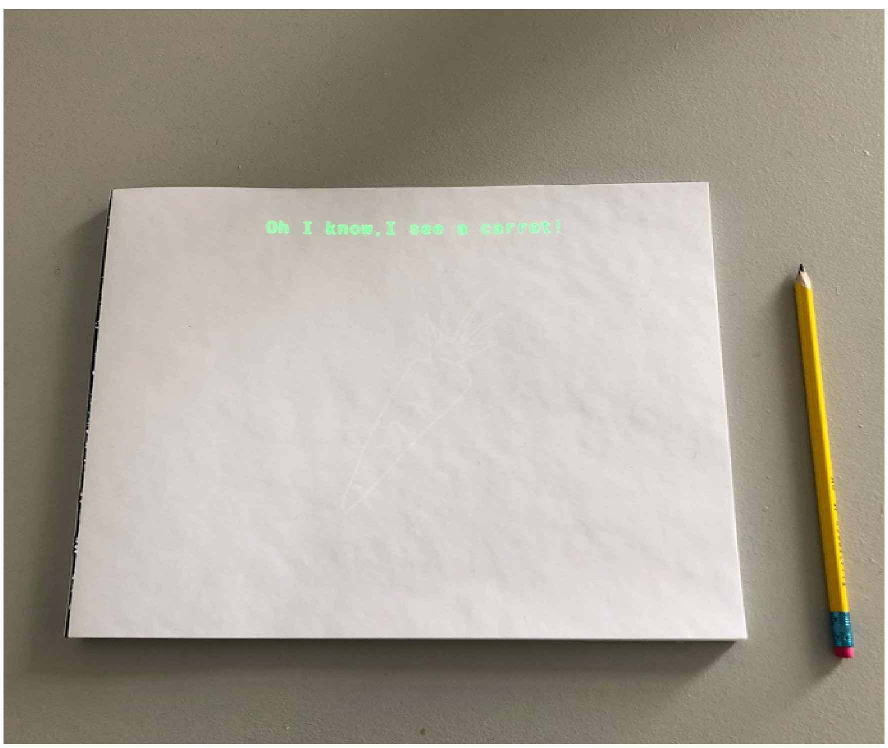

Fig. 7

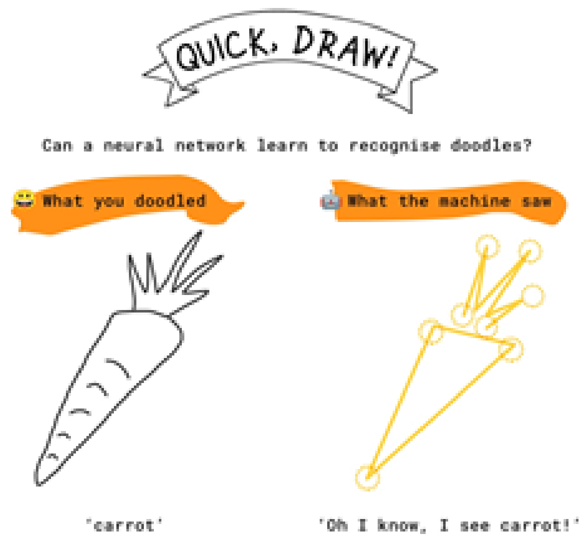
QUICK, DRAW!
Can a neural network learn to recognise doodles?
What you doodled
What the machine saw
'carrot'
'Oh I know, I see carrot!'

Fig. 8

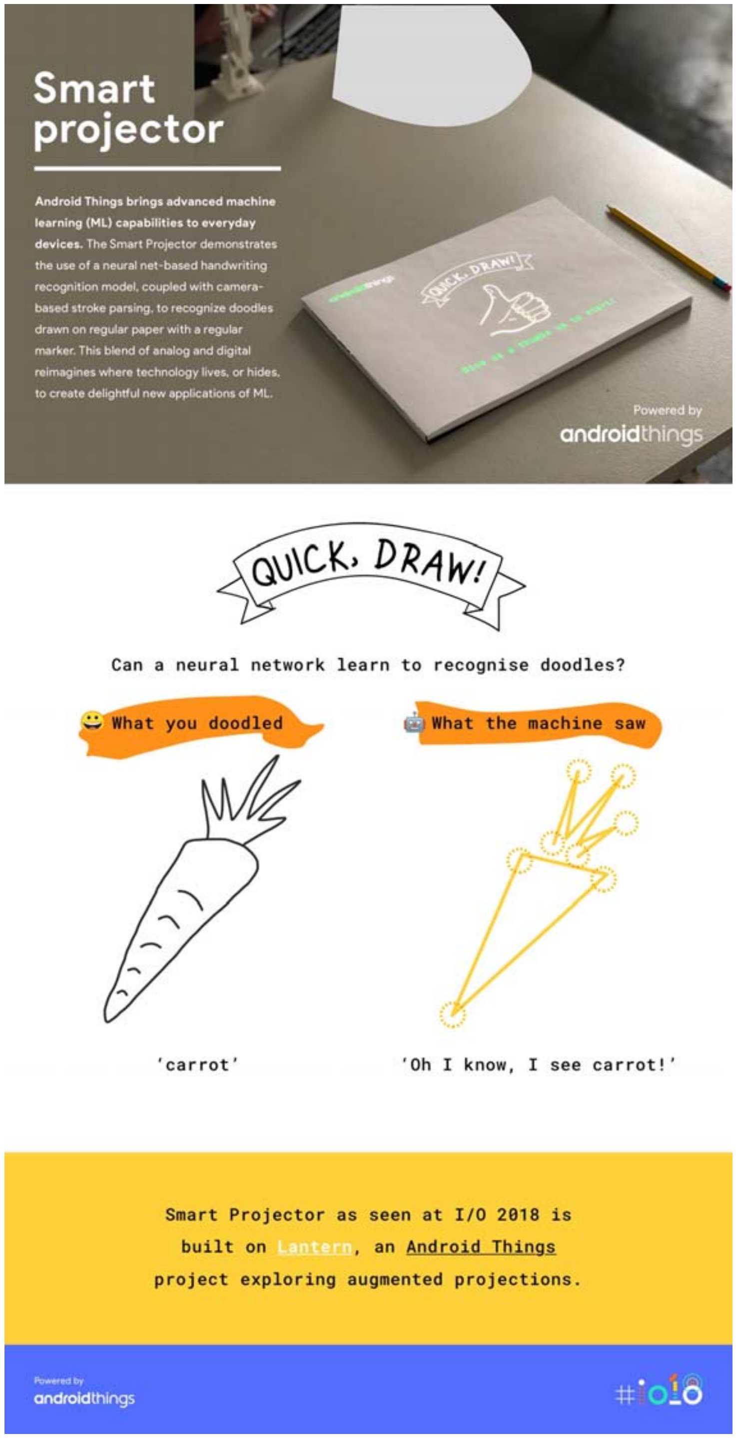
Smart projector
Android Things brings advanced machine learning (ML) capabilities to everyday devices. The Smart Projector demonstrates the use of a neural net-based handwriting recognition model, coupled with camera-based stroke parsing, to recognize doodles drawn on regular paper with a regular marker. This blend of analog and digital reimagines where technology lives, or hides, to create delightful new applications of ML.
Powered by androidthings
QUICK, DRAW!
Can a neural network learn to recognise doodles?
What you doodled
What the machine saw
'carrot'
'Oh I know, I see carrot!'
Smart Projector as seen at I/O 2018 is built on Lantern, an Android Things project exploring augmented projections.
Powered by androidthings
#io18

Fig. 9

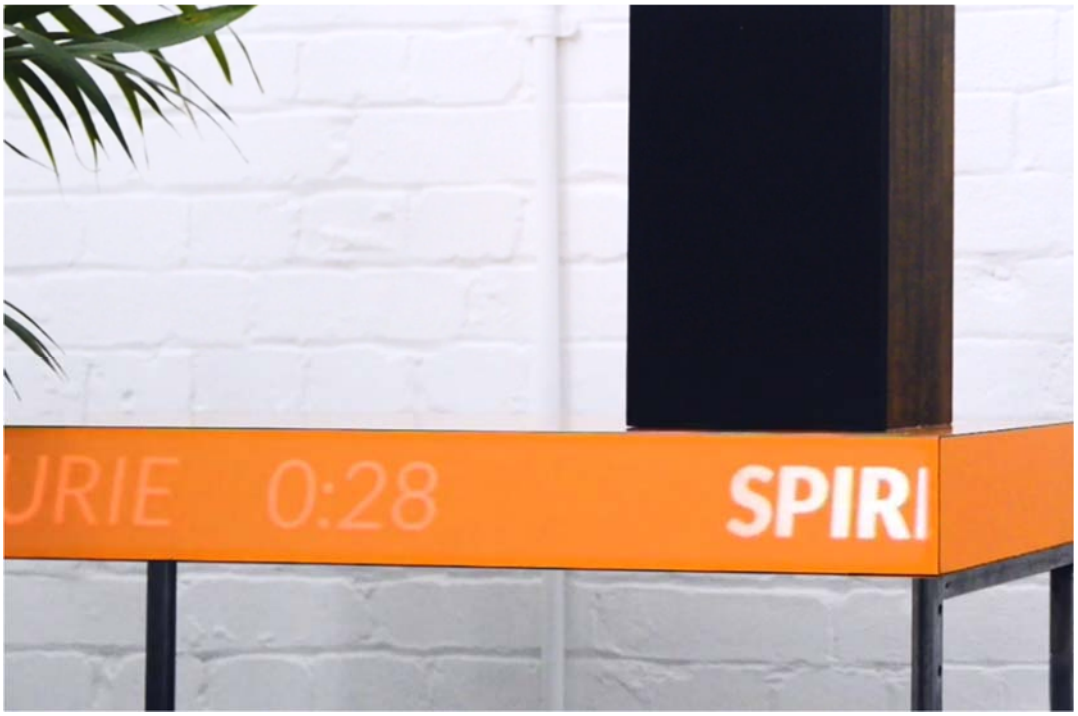

Fig. 10

Fig. 11

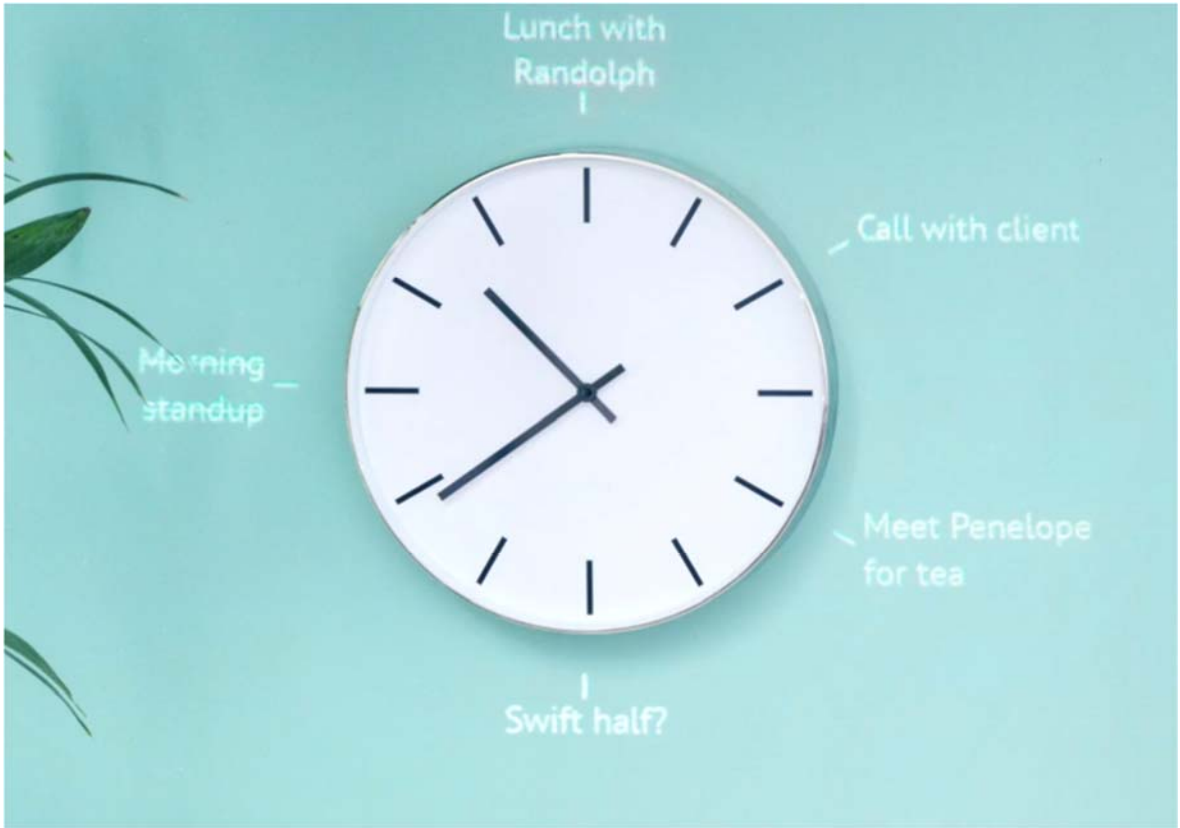
Lunch with
Randolph
Call with client
Morning
standup
Meet Penelope
for tea
Swift half?

Fig. 12

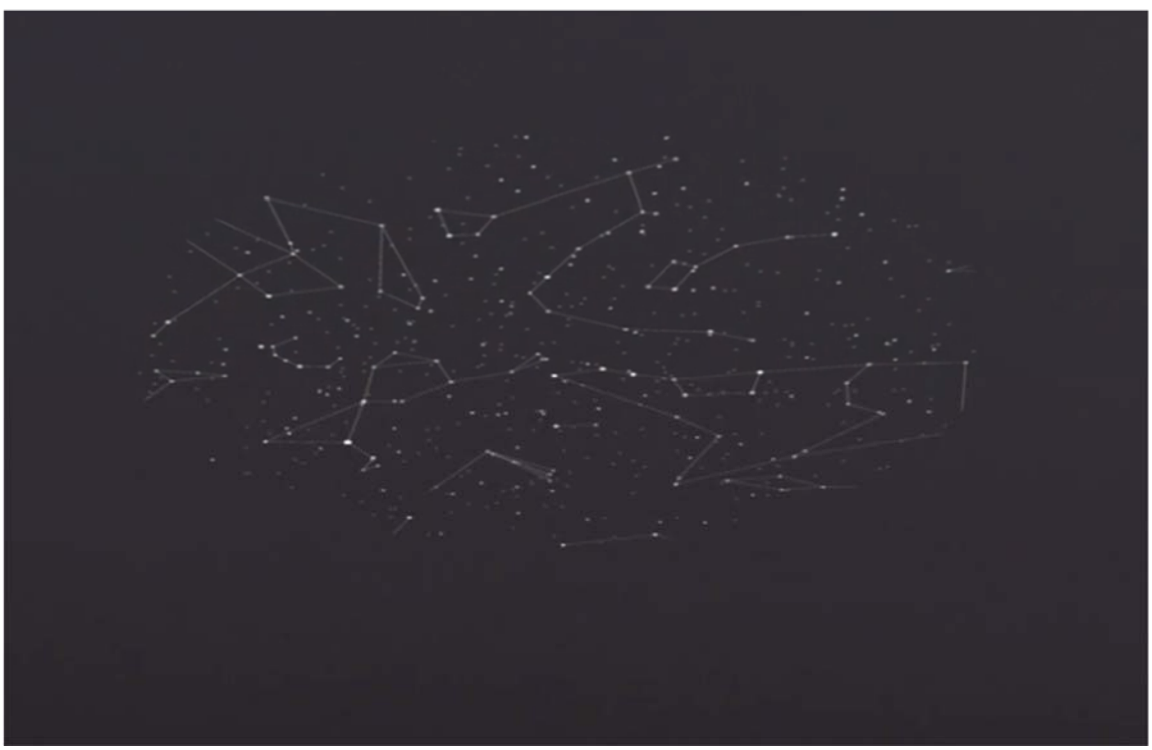

Fig. 13