Mixed Reality
MPhil Exegesis

Ehsan Tavassoli
Student Id: 1129682

AUT University
15 Feb 2017

"I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgements), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning."

Ehsan Tavassoli

15/02/2017

Abstract

The project called “mixed reality” started by exploring a number of technologies to do with augmented reality (AR), artificial intelligence (AI) and internet of things (IoT) in relation to contemporary art. Through speculation with technology, physical augmented reality (PAR) and its parallel to relational architecture was formally singled out and defined, AI based image analysis was used to encourage participation in the context of interactive art and a formal cloud based serverless microservice oriented software architecture along with all its artifacts and software were defined and implemented. The outcome was an interactive art installation that involved image analysis (e.g. age, gender) and emotion recognition in order to provide an effortless interaction by the interlocutors with the system through IoT based multi-colour light-emitting diodes (LEDs) and text projection.

Research Question

The rationale for having an art component to this project is to test in the real world of visual acuity the reality of the augmenting component, to source the expertise and visual knowledge of the arts but to bring something new to the art world, from another paradigm in line with some of the concepts of paradigm shift as propounded by Khun’s (1962) The Structure of Scientific Revolutions and his ideas of the world as a collection of facts that are contained in a theory or paradigm. The idea is to test scientific theory and practice in relation to the computer and speculate with advances in computer technologies that then are applied in various contexts. Particularly art installation contexts where it is desirable to set up ambiguous sets of stimuli about reality to engage the interlocutor of the artwork in a dilemma about that which is represented and that which is real. Enticing the viewers, into an interactive relation, with the artwork and to its authorship. Thus, the quotidian of contemporary art practice is or could be tested and augmented by the computer. Hence, demonstrating and exploring systematic understanding of the current state of knowledge within the field of theory and practice of contemporary art, which deals with the nature of the real and perception. This can be critically tested within the art aspect of the project, by viewer response and subsequent discourses, perhaps developing a direct critical response within the work itself, through various forms of interaction and heterogeneity of readings. The critical input to the project, which comes direct from scientific imperatives and methodologies such as image analysis, AI, cloud computing and IoT. These imperatives can function in the same way as phenomenological and philosophical material does for contemporary art practice now, driving the production and design of projects from an anterior or sociological perspective rather than the modernist model, which was that art was about self- analysis, or the romantic introversions of the individual genius.

The following topics will be reviewed and discussed and then applied to an artwork exhibition, which will utilize AR, IoT and AI:

1. A formal exploration of *physical augmented reality (PAR)*
2. AI as means of encouraging participation in interactive art
3. Software architecture for augmented reality and IoT implementations

Methodology

In 2007 Sha Xin Wei's Topological Media Lab started a project called "Ouija" which included several experiments with movement and technology. One of the experiments included a number of performance artists, performing a choreographed routine while a motion sensing camera analysed their movements. When the performance reached a certain threshold, a large screen projection started on stage streaming a live augmented reality form of the event.

The project was strongly criticized. The critiques argued that the performance was staged, there was the traditional theatrical separation between the performers and the audience and the interaction was only between the performers and the technology. Furthermore, "people said it was politically bankrupt because it had no 'real' interaction".

Xin Wei responded to the critique by stating that the Topological Media Lab was doing "*speculative* work with technology. That doesn't mean that we're supposed to speculate on what the technology might potentially do or who or what might want it or what it does. It means that the work itself technically speculates. Its dynamic form *is* speculative by nature. It's a speculative *event*. To speculate is to turn in on yourself. You turn in, to connect immanently with what is absolutely outside -- both in the sense of belonging to other formations monadically separated from your present world, and in the sense of what may come but is unforeseeable".

This current project, called "Mixed Reality" has come about in a similar way; through speculative research with technology. The motivation behind the project came from the rise in popularity of virtual reality, augmented reality, AI, sensor hardware technology such as Microsoft Kinect and Internet of Things (IoT) infused with their potential application to contemporary art. This was in sharp contrast with typical computer science and IT projects where the requirements are at least known at an abstract level prior to commencement of the project. The idea was to assess what the technology could do that was both aesthetic and innovative and to collaborate with artists and technology enthusiasts, while not knowing what the final exhibition would look like.

The methodology was refined over the course of the work and an Agile design/build/test methodology was adopted and applied to speculative research to cope with rapidly changing requirements and scope changes. Architectural and other technical decisions where based on the requirements that came about in the process of Agile software development. During the process, several prototypes were created in an iterative fashion and the technological and aesthetic value of each of these were assessed with the help of the supervisors and other artists and that was the fundamental deciding factor as to whether or not a technology was chosen. The design was based on an artistic requirement, build, test and then assess whether the outcome is artistically satisfactory.

Literature Review

The next few sections include literature review and a deep analysis of the intricacies of the installation including technical explanations and art and philosophical theories behind the aesthetic decisions and how the interplay of aesthetics and technical speculations lead to the final outcome.

Augmented reality

In the past few years with the help of new technologies there has been a growing body of research in the field of computer-constructed reality (Kipper & Rampolla, 2012). One of these fields is live augmented reality. That is an augmented reality system that interacts with its user/interlocutor in a real-time manner (Mullen, 2011).

There have been several attempts at defining augmented reality. Some researchers have used a broad definition, some more specific while others have created diagrams and continuums. Kipper and Rampolla (2012) describe augmented reality as "taking digital or computer generated information, whether it be images, audio, video, and touch or haptic sensations and overlaying them over in a real-time environment". Mullen (2011), defines augmented reality as "a combination of technologies that enable real-time mixing of computer-generated content with live video displays". Craig (2013) defines augmented reality as "a medium in which information is added to the physical world in registration with the world." Augmented reality technologies can be used to augment all five senses of sound, sight, touch, smell and taste. However, visual enhancement is the most commonly used medium in augmented reality (Kipper & Rampolla, 2012). Majority of the literature focus on adding or superimposing visual elements to the real-time video environments while some also include removal or replacing existing visual elements. Azuma (1997) elaborates, "It would appear to the user that the virtual and real objects coexisted in the same space".

Augmented reality is also often defined in comparison to virtual reality as a variation of the virtual environment, where the environment is not completely replaced with a computer-generated environment (Azuma, 1997). Sharman and Craig (2002) define augmented reality as a *"type of virtual reality in which additional information, otherwise imperceptible to the human system, is made perceptible and registered with the display of the physical world".* Milgram et. al. (1995) explain that augmented reality can be thought of as the "middle ground" between completely synthetic and completely real and they defined augmented reality as part of their proposed Reality-Virtuality Continuum as depicted in the figure 1 below:

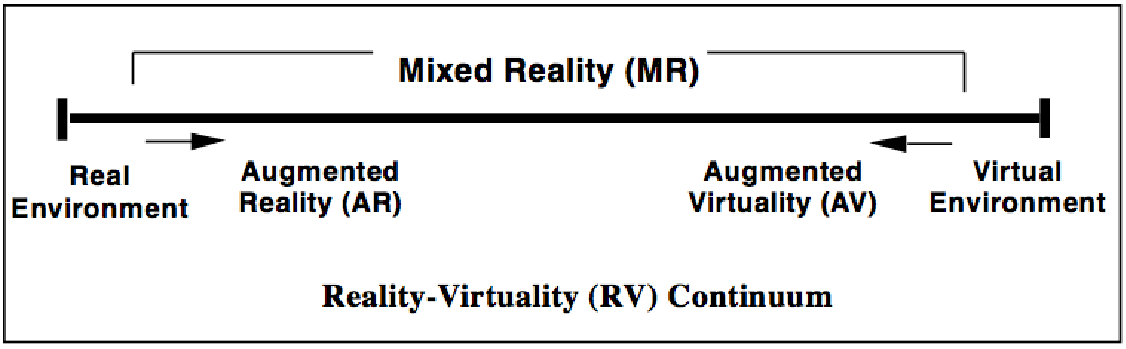


Figure 1: Reality-virtuality continuum

On right hand side of the continuum, the user is totally immersed in a synthetic world unbound to laws of physics (often wearing a headset) and on the left-hand side, the real world bound by the laws of physics. In the middle of the continuum there are concepts of augmented reality (real environment with synthetic cues) and augmented virtuality (synthetic environment with real cues).

There has been some criticism of the way augmented reality has been defined as Craig (2013) puts it "the term *augmented reality* is being bandied about very widely, and it is not clear exactly what people mean by the term". In regards to the problem of defining augmented reality Craig (2013) continues: "some sources use a very broad meaning for the term, whereas others mean something very specific and narrow". The literature review so far is also in agreement with Craig's (2013) view, with the exception of the Reality-Virtuality Continuum proposed by Milgram et al. (1995) which is also the most referenced research in augmented reality literature.

Nevertheless, as Craig (2013) also points out, even the traditional augmented reality is loosely defined. Because of this some researchers have tried to define augmented reality through its characteristics.

Azuma (1997) defines the characteristics of an augmented reality system as follows:

- It combines real and virtual
- It is interactive in real time
- It is registered in 3D

Craig (2013) defined key aspects of augmented reality as follows:

- The physical world is augmented by digital information superimposed on a view of the physical world.
- The information is displayed in registration with the physical world.
- The information displayed is dependent on the location of the real world and the physical perspective of the person in the physical world.
- The augmented reality experience is interactive, that is, a person can sense the information and make changes to that information if desired. The level of interactivity can range from simply changing the physical perspective (e.g., seeing it from a different point of view) to manipulating and even creating new information.

Craig (2013) continues with a final attempt at defining augmented reality based on the key aspects identified: *"A medium in which digital information is overlaid on the physical world that is in both spatial and temporal registration with the physical world and that is interactive in real time."*

Craig's (2013) definition makes an assumption about the "digital" and "informational" nature of the augmentation, which makes it difficult to apply to all five senses. For this reason, perhaps a broader definition of augmented reality is more suitable for example Das (1994) defines augmented reality as "augmenting natural feedback to the operator with simulated cues". This perhaps is the best definition, as it does not make any inherent assumptions about reality (natural feedback), the operator (interlocutor) or augmentation (simulated cues).

Interactive art

Interactive art is often defined as an "artwork-event" (Kluszcynski, 2010). Some like to define interactive art as "media". However, Suzanne Langer (1953), a notable aesthetics philosopher has posited that all art forms are best analyzed as experiential events instead of "media". Langer (1953) further states that all arts are "occurrent arts". That is from the phenomenology of perception it is an experiential event both that "it is happening" and then "something new transpires". Nevertheless, from theoretical and phenomenological perspectives, an interactive artwork is dynamic and its qualities are formed based on its interaction with interlocutors (Massumi, 2008).

Perception is another important aspect to explore in the context of interaction. For the purpose of this exegesis, it is sufficient to note that perception must have presentational immediacy. Perception is always "direct and immediate" and it has its own self-embracing *event* (Massumi, 2002). This is important, because perhaps this project brings something new to the mix; the language and state of the machine in interactive art. In programming languages' terminology, no interaction (between participant and computer) is possible without an event – that is the semantic meaning of an event as a predecessor to an action. In programming and hardware, it is "events" that fuel interactions. Interactions basically respond to events. For example, the click of a mouse is an event, the movement of a body is an event or change in outside temperature is an event. An event exists *before* the interaction.

Again, referencing Massumi (2008), "What is central to interactive art is not so much the aesthetic form in which a work presents itself to an audience -- as in more traditional arts like painting, sculpture or video installation art -- but the behaviour the work triggers in the viewer. The viewer then becomes a participant in the work, which behaves in response to the participant's actions. Interactive art needs behaviour on both sides of the classical dichotomy of object and viewer". The definition here, has now broadened to include aesthetics and essentially the question of what is "good" interactive art and what point we draw the line between a computer *game* and interactive art?

Massumi states: “I would encourage a rethinking of interactive art – from the premise that its vocation is to construct a situation or go into an existing situation, and open it into a *relational architecture*”.

Rafael Lozano-Hemmer was the pioneer of the term “relational architecture” in 1994. In a series of installations, he superimposed audiovisual elements over buildings and urban environments. He aimed to re-contextualise architecture and in his words “I want buildings to pretend to be something other than themselves, to engage in a kind of dissimulation”.

It is important to make a point here, a computer scientist would undoubtedly classify the works of Lozano-Hemmer as “augmented reality”. As explained later in this exegesis perhaps an unexplored class of augmented reality within computer science called “physical augmented reality” or PAR can easily contain this. That is every superimposition, be it projection, robotic or IoT, even as simple as a projector projecting on a physical wall or a change in lighting and sound, as long it has the qualities of information and interaction can be classified under PAR within the context of computer science. It is therefore, plausible to parallel PAR with relational architecture.

Massumi (2008) applies the term relational architecture to the broader aesthetic interactive art and argues that this makes *potential* “reappear at a self-abstracting and self-differing distance from routine functioning in a potentialised semblance of themselves”. He further elaborates that this is exactly what is needed to make art political and “it can push further to the indeterminate but relationally potentialized fringes of existing *situations*, beyond the limits of current framings or regulatory principles. Aesthetic politics is an exploratory politics of invention, unbound, unsubordinated to external finalities. It is the suspensive aspect of it that gives it this freedom.”

This takes us to a point that Massumi (2008) makes “If ‘please interact’ were enough to define a category, it would be gaming, not art. Beyond gaming in the strict sense, there is a gaming paradigm that has moved into other domains. You see it massively in communications, but also in marketing, design, training, education. Places where it becomes serious and useful”. Therefore, it would be incorrect to say that PAR in computer science is the same as relational architecture in arts. PAR in the context of the broader augmented reality, can have gaming and informational purposes. Relational architecture is concerned with aesthetics and potentiality and semblance. Furthermore, as Massumi (2008) puts it, “What is architecture, if not “site-specific” life-design?” This means that relational architecture could be the interior or exterior design within a space and it does not have to be an entire building or the environment.

An example of a relational architecture that utilizes PAR as means to its end is one of the works of Lazano-Hemmer in 2008. He created a work called “Pulse Park, relational architecture 14”. A biometric sensor sculpture was installed in Madison Square Park, New York. A matrix of pulsing light beams then projected a simulation of participant’s heart rate over a large ground area in the central oval field of the park. In other words, the park was reacting to the participants’ heart. In Lozano-Hemmer’s words “a poetic expression of our vital signs, transforming the public space into a fleeting architecture of light and *movement*.”

It is not the intention of this exegesis to critique the works of Lazano-Hemmer. However, in the example of Pulse Park installation described above, the participants were asked to hold the heart rate sensor/sculpture in order to interact with the work, and this takes us to the question of participation. “The uncanny double of interactivity is interpassivity.” Spectators, are “now invited to interact” with the artwork. “Such relationships have put an end to a supposed passive consumption of artworks. The spectators have been asked to shout, move, touch, hold, select, click and even put VR helmets on” Zizek (1998). This becomes an indirect effect of what Foucault (1982) calls, “regimes of power”; to impose an imperative to participate as opposed to passive consumption of artwork.

Referring back to Massumi (2008), “What is central to interactive art is not so much the aesthetic form in which a work presents itself to an audience -- as in more traditional arts like painting, sculpture or video installation art -- but the behaviour the work triggers in the viewer. The viewer then becomes a participant in the work, which behaves in response to the participant’s actions. Interactive art needs behaviour on both sides of the classical dichotomy of object and viewer”. It seems important to refer back to the language of the machine. Assuming we are speculating with technology, the statement “the work, behaves in response to the participant’s actions” can further be expanded. Perhaps the link between action and response is not so direct, especially when there exists an element of artificial intelligence. In a sense by applying artificial intelligence to interactive art, one can create an embodiment for the interplay of perception and dynamic (or *adaptive*) forms. A participant’s action, raises a series of events in the computer. It is an event that invokes a response, not the action. Furthermore, a response is not necessarily an interaction or system behavior (as opposed to action), a response can also be one or more “events” for the computer. The concept of “action”, as discussed above, raises a series of events that the machine can decipher. It is important to note that “action” must be implied because sensors (such as a camera or thermometer) that pickup these events may be responding to *inaction* (for example no movement) or what can be defined as unconscious *being* (for example emotions, gender, skin colour or facial features).

In a good installation of interactive art, the interlocutors should be *drawn* to participate and have a choice or already be participat*ing* as in more traditional passive consumption of artworks. An interactive artwork needs to have its own momentum and be “autonomous” to have true aesthetic politics. It has to be infinite. It needs to recreate itself based on its own tendencies. It has to create its own *movement* in the *dynamic form* in which it operates (Massumi, 2008).

Internet of Things (IoT)

IoT is the terminology used for a combination of existing technologies and systems such as sensor networks, cyber-physical systems, ambient intelligence and ubiquitous computing. CASAGRAS (Coordination and support action for global RFID related activities and Standardization) a project financed by the European Union, defined IoT in 2009 as “a global network infrastructure, linking physical and virtual objects through the exploitation of data capture and communication capabilities. This infrastructure includes existing and evolving Internet and network developments. It will offer specific object-identification, sensor and connection capability as the basis for the development of independent cooperative services and applications. These will be characterized by a high degree of autonomous data capture, event transfer, network connectivity and interoperability" (cited in, Minerva, Biru & Rotondi, 2015).

Gubbi et al. (2013) argues that the use of existing technologies such as "ubiquitous sensing enabled by Wireless Sensor Network (WSN) technologies cut across many areas of modern day living. This offers the ability to measure, infer and understand environmental indicators, from delicate ecologies and natural resources to urban environments. The proliferation of these devices in a communicating–actuating network creates the Internet of Things (IoT), wherein sensors and actuators blend seamlessly with the environment around us."

In 2009, Natalie Jeremijenko and Bureau for Inverse Technology created an interactive art installation by utilizing IoT called Amphibious Architecture. The installation involved submerging a number of sensors into water and measuring water quality and detecting the number of fish. Participants could then send a text message to *the fish* using their mobile phones. The fish then texted back with messages such as: "Hey there! There are 11 of us, and it's pretty nice down here. I mean, Dissolved oxygen is higher than last week..." (Jeremijenko, 2009). Brian Massumi had the following to say about the work: "The thinking-feeling-like-a-fish was the semblance in the situation of the situation, pointing beyond it. A quality of experience was built-in that could potentially lead to thoughts, sensations, and further perceptions that might fold-out, toward follow-on in other situations that neither the participants nor the artist could foresee (never having been an environmentally aware fish before)" It is "inventive in a more radical way than a technical invention in the usual narrow sense. It's not the gadgetry or setup that's creative, even if nothing like it has ever been seen before. The setup is creative to the extent that an emergent experience takes off from it, that has its own distinctive lived quality, and because of that its own self-differing momentum."

The ubiquitous vision of IoT is very broad, idealistic and immeasurable. The contemporary art aspects of this project can enable implementation of some of these ideals from a complete subjective view of the interlocutors' reality, and therefore, present an idealist and immeasurable scenario. It will be discussed further that AI plays a significant role in enabling participation and the formation of these idealistic scenarios within the interlocutor's individual phenomenological perception. In other words, through art and AI based image processing, it might be possible to create an illusionary connected physical world that appears intelligent but only within the subjective experience of participants.

A full discussion on the aesthetics of IoT in relation to art is outside the scope of this exegesis, however, there are components of IoT that intersect with PAR; Pulse Park is another IoT example of the sometimes-blurred line between technology and art. A heart rate monitor had the semblance of a sculpture (or vice-versa), while the IoT network sent signals to lights to create a relational architecture.

The cloud

For the purpose of this project, AI based image analysis was evaluated. AI based image analysis is very computation intensive and often require expensive hardware. As discussed earlier relative and representational immediacy is imperative in both augmented reality and interactive art. Especially when for aesthetical reasons, it is desirable to use small mini-computers such as the Raspberry Pi. In order to solve this problem, one can utilize the power of the cloud. Cloud computing allows for on-demand scaling of hardware to unimaginable processing and storage levels.

An interesting concept in relation to art theory can be extracted from Cloud computing. The technical setup would be minimal and limited to the decoration. Yet the processing in the cloud could give infinite dynamic forms to interactive art. Regular data feedback and self-eventing of the AI engine can create new interactions and/or reactions to the installation and make it ever changing and evolving.

The utility of IoT was discussed earlier. It happens that as of 2017, it is the cloud services that provide the IoT services along with other cutting edge technologies that can both empower and reduce the complexity of a system. In particular, IoT Hub and serverless microservices (a subset of SOA). Definitio and application of these cloud technologies will be discussed in the next few sections.

From a very high level perspective, Newcomer and Lomow (2005) describe service oriented architecture (SOA) as an architecture that “guides all aspects of creating and using business services throughout their lifecycle (from conceptions to retirement), as well as defining and provisioning the IT infrastructure that allows different applications to exchange data and participate in business processes regardless of the operating system or programming languages underlying those applications.”

SOA falls under the wider concept of service oriented computing (SOC) however, these terms are often used interchangeably. Issarny et al. (2011) state that “service oriented computing (SOC) is now largely accepted as a well-founded reference paradigm for Internet computing. Under SOC, networked devices and their hosted applications are abstracted as autonomous loosely coupled services.”

From the definitions, it becomes visible why it is important for IoT components to follow SOA.

SOA provides definitions for functional requirements such as creation of services and also none functional requirements such as underlying infrastructure. Cipolla-Ficarra (2013) in his defines the key aspects of SOA:

- Loose Coupling: services maintain a relationship that minimizes dependencies and only requires that they retain an awareness of each other;
- Service Contract: a communications agreement, as defined collectively by one or more service descriptions and related documents;
- Autonomy: services have control over the logic they encapsulate;
- Abstraction: beyond what is described in the service contract, services hide logic from the outside world;

- Reusability: logic is divided into services with the intention of promoting reuse;
- Composability: collections of services can be coordinated and assembled to form composite services;
- Statelessness: services minimize retaining information specific to an activity, and
- Discoverability: services are designed to be outwardly descriptive so that they can be found and assessed via available discovery mechanisms.

Modern cloud infrastructure is evolving rapidly and there are new trends such as microservices and serverless computing. Microservices are a subset of SOA however they differ in a way that they are hosted on their own infrastructure. Cockcroft (2015, cited in Nadareishvili et. al., 2016) defines microservices as “loosely coupled service oriented architecture with bounded contexts.” In more traditional SOA, the same infrastructure could house multiple services. The concept of serverless computing allows for running microservices as isolated functions without the need to provision any hardware.

To give an example of the power of the cloud in relation to a visual augmented reality system, rendering a complex 3D Ultra High Definition animation superimposed on a live children’s programme is impossible by using an ordinary PC or a smartphone. The rendering will take hours to complete and the children’s programme cannot be broadcasted live. In such scenario, processing time would only take a few milliseconds if rending was to be sent to the cloud, where for example 1000 processors can spin up in a matter of milliseconds and complete the processing. In this example, the animation rendering service is running in the cloud and returns the rendered animation back, where another service can perform the superimposition during the broadcast.

The project: mixed reality

“Mixed reality” has certainly come to life through the process of speculation with technology and collaboration with supervisors from AUT University and other artists, students and enthusiasts. The process started in 2013. At the beginning of the project, there was no defined final work. There were only three elements: augmented reality, artificial intelligence and visual arts. Between 2013 and 2016 a number of technologies where explored and their aesthetic potentials were assessed. For example, a number of prototypes where created for Microsoft Kinect sensor around superimposition of live characters into a virtual world and introduction of virtual elements and characters on live-stream video. It was clear at the very beginning that working with real or virtual characters was not going to produce much aesthetic innovation into the field of augmented reality and *abstract* was the way to go. Therefore, there was no longer a compelling reason to use Kinect technology. This also meant by removing the human element from the live-streams, there wasn’t much that could be done with live-streaming in terms of traditional augmented reality. This constraint opened an opportunity to explore modalities other than live-streaming augmented videos. The body of research was so preoccupied with solving technical problems such as depth and tracking, that it had left not much room to explore anything else. The literature review revealed that *physical augmented reality (PAR)* was an unexplored area of the wider augmented reality definition. PAR is not a new field as it has been defined in the literature, however, not much attention has been given to the theory behind its attributes and qualities. It is also so subtle and simple that it can be easily overlooked. PAR is when virtual elements are added to a physical space as opposed to the live-stream of the physical space. An example of this is a projection on a wall. The wall being in its *actualised* form, through *movement*, is transformed into one of its *potential* dynamic forms, and that is to become another actualised form (e.g. a screen) in the interlocators’ *perception*. It is about mixing something *virtual* with something that is seemingly real, yet virtual itself in its phenomenology of perception. Within the context of art theory, something as institutional as a wall, can turn into a *relational architecture*. Of course, if it has the necessary aesthetic and/or political qualifiers and *semblance*. Even though a simple projection on the wall augments the wall, within the scope of the technical definition of augmented reality, the augmentation alone is not enough to be considered an augmented reality system. It needs to be informational and interactive in a sense that it represents the information in relation to interlocutors. The nature of the information is also dynamic in respect to its infinite potentialised forms.

Over the years it became evident that the work will have a big intersection with category of interactive art, not in the way that augmented reality is interactive art, but in the sense that it can become a pivotal aspect of an interactive art *event*. In other words, its artistic aesthetics can come to life in the context of interactive art. This brought about a new speculation around IoT, that is devices that are connected to the internet and can send sensory information (e.g. a camera) and/or receive commands (e.g. a connected lightbulb). Initially the assumption was that IoT will provide means to interpret interlocutors’ action through its sensor technology. However, once placed in the context of interactive art, IoT can also be used in the receiving end of the interaction and for one action there could be a chain of reactions in the setup.

At this point it was narrowed down to only on three things: a camera sensor, a screen and connected light bulbs. Some event analysed by the camera would turn the lights on. It was now time to analyse the aesthetics of the space. A small dark room, with the camera and the lights in relatively close proximity. This created a new technological problem to solve. A very small camera was required and since there wouldn't be a monitor, it seemed necessary to minimalise the computer and the cabling. This resulted in the use a Raspberry Pi 3 minicomputer board that has its electronics exposed. This will give an authentic play between the necessary aesthetic quality of the space and practicalities.

The next problem to solve and perhaps the most difficult of the speculations, was in relation to the problem of participation in interactive art; how would the interlocutors interact with the work without having to put a mundane note that says "please interact". At the beginning of the project, it was assumed that artificial intelligence will be used to create the augmentation. It became an increasing difficult question whether or not AI was still relevant in the context of lights. Little was it known that AI could potentially solve the problem of participation. This meant that live input video from the camera would now need to be extensively analysed beyond simple detection and also run an AI engine in the background. Something the small Rasperry Pi 3 couldn't handle or exhibit low performance and that would break the principle of *relative immediacy* in augmented reality. Since the Rasperry Pi 3 is equipped with an internet connection, a decision was made to send all the processing to the Cloud. This way the only latency was upload and download of snapshots which was an acceptable latency within the context of the work.

An interesting concept in relation to art theory can be extracted from Cloud computing. The technical setup would be minimal and limited to the decoration. Yet the processing in the Cloud could give infinite dynamic forms to interactive art. Regular data feedback and self-eventing of the AI engine can create new interactions and/or reactions to the installation and make it ever changing and evolving. By having things in the cloud the technical direction of the project was also clear, service oriented computing by using microservices and serverless cloud computing.

The final installation will be in a small dark room. Within the room there will be a camera connected to a Raspberry Pi 3. There will be a table and a chair. There will also be a plinth, which would hold the LED multicolour IoT connected lightbulb. There will be a projector, projecting text on the screen. This text will have an agency to promulgate interaction whilst at the same time taking on a poetic and socio-political content and context in the work. The camera will be capturing a live video. Snapshots of the video will be sent to the cloud through the Raspberry Pi 3. The cloud architecture uses the choreography pattern instead of orchestration so that every component of the art installation can be autonomous and compute in parallel for maximal performance. In the cloud, an event will be raised and three serverless microservices will be notified to send the snapshots to the three AI services: computer vision, face and emotion. The computer vision service will detect visual elements found in the image for example people, context, objects and actions. The face service will extract information such as age, gender and facial features such as beards and the emotion service will detect one or more emotions (including micro emotions) from each face. The extracted data is then saved and sent to the IoT Hub which holds a register of the connected devices. Two devices are registered with the IoT Hub. One Raspberry Pi 3 that is the source of the projection of the text on the wall and another Raspberry Pi 3 that sends the signals to the light bulb. When the projection device receives the events, it starts displaying the text received. The text will include a narrative of the features that were extracted by the AI based image analysis services. For example, “you are a middle-class male, wearing glasses, and looking ridiculous” or “you are weak natured” or “why do you look upset?”. The key driver is emotion. Everything else is meant to influence emotions. When an emotion is detected, the light device receives a message which contains a colour code associated with the detected emotion. The light bulb would then change colour according to the colour of the detected emotion. For example, if anger is detected, the light will turn red. This in turn will promote further emotional responses to colour itself and so on - setting up a kind of palindrome of affect between the interlocutor and the work.

The System

From a high-level perspective, augmented reality can be broken into four aspects (Kipper & Rampolla, 2012):

- Input, this is usually cameras, microphones and other sensor and tracking devices.
- Processing, this is the stage that input is augmented and states and constructs are defined and some kind of output is created.
- Output, this is usually displays, speakers and/or other technologies such as robots.
- Participant, this is at the centre of the augmented reality world and the reason the system exists.

Figure 2 depicted below attempts to visualize Kipper & Rompolla’s (2012) work:

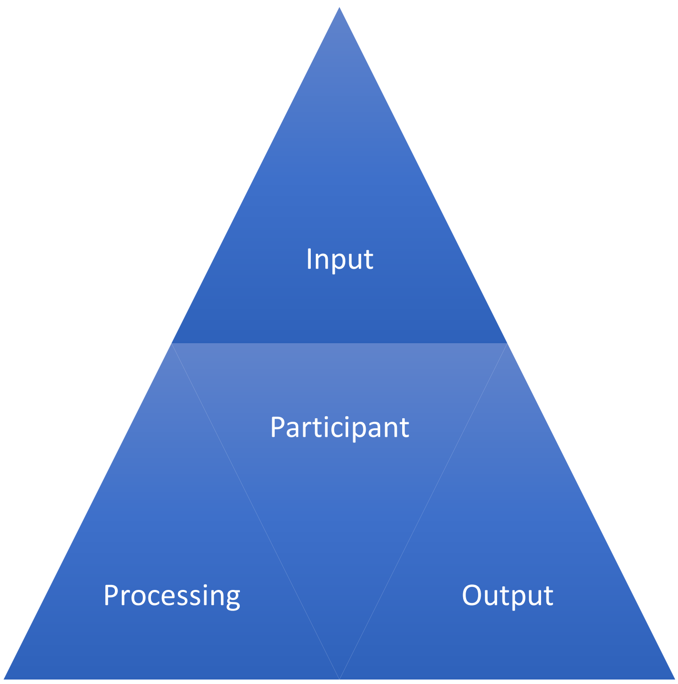


Figure 2: four aspects of augmented reality

The following sections will describe the installation in respect to the four categories described above. See Appendix A for architecture diagram.

Microsoft Azure Cloud and C# programming languages were chosen because of the author's familiarity with these technologies and their seamless integration with IoT devices. Microsoft Azure is a full-scale provider of cloud infrastructure and hosted services such as storage, queues, events, serverless services, AI and machine learning and IoT Hub among many others.

It is worthwhile to note that Amazon Web Services (AWS) was an early contender for cloud technology, and a few prototypes were created in AWS using serverless Lambdas written in Node.JS and Python programming languages in order to investigate machine learning capabilities of AWS and during this time Microsoft Azure did not offer serverless services. In late 2016 Microsoft released their version of serverless functions which offered the C# programming language (while AWS did not) and therefore the development was moved from AWS to Azure cloud for the sake of consistency of the programming language used on devices and in the cloud and better familiarity of the author with the C# programming language.

Input

A high definition (HD) camera was connected to a Raspberry Pi 3 computer board. Raspberry Pi 3 was build using Windows IoT operating system and a piece of software called "Movement" was written in C# .NET programming language to communicate with the camera. The Raspberry Pi 3, made the camera *connected* and therefore, turning it into an IoT device. The choice of camera and Raspberry Pi 3 were purely for aesthetic reasons in relation to the space and therefore a requirement of the project. Windows IoT was used because of its ability to run code written in the C# programming language and the availability of Microsoft Azure Software Development Kit (SDK) for Windows IoT.

Movement's workflow and important function have been described below.

1. Turn the camera on and start video streaming
2. Take snapshots from the camera
3. Upload snapshots to the cloud

The idea here is to keep the input layer as thin as possible. It is difficult to update IoT devices even for testing purposes. Therefore, whenever there is a centralized server, it is best to move all the logic onto that server so updates can happen without a need to upgrade individual IoT devices.

Movement uses the MediaCapture library to operate the camera. MediaCapture was picked over OpenCV because it is native to .NET. All camera operations to and from the camera are via asynchronies events. Approximately 15 frames per second are snapshotted and uploaded into the cloud using the current Thread Pool Timer. Azure SDK was used to connect to Azure Blob Storage. Azure Blob Storage was chosen because it is an affordable cloud storage service. Furthermore, Azure Blob Storage offers the unique feature of raising an *event* whenever a file is uploaded into the cloud. In the next few sections, it will be discussed how this event is picked up by the image processing services to analyse the snapshots. The Movement software decodes snapshots into a JavaScript Object Notation (JSON). JSON was chosen over XML because of its lightweight data-interchange format. It is also human readable and easy for machines to parse and generate in comparison to XML. The logic behind this conversion is so that an image snapshot can be converted to a message. This JSON message, includes some information about its surroundings and the IoT device's unique identifier. It is worthwhile to note that this project is massively distributable and reproducible meaning that it can simultaneously function in multiple geo-locations. This was not intentional but another outcome of speculating with technology. The current project's setup will only be in one location however. The content of the JSON are as described in Appendix B.

As mentioned earlier when files are uploaded into the cloud, an event is raised. This is important for the choreography of the interactive art project. Referring back to Massumi (2008) that interactive art components need to be "autonomous" and therefore, event sourcing was a requirement of the project. "Autonomy" is the fundamental reason why the architecture uses choreography pattern over orchestration pattern. In choreography, every component is self-serving and only communicates to other components through messages whist in orchestration pattern, there is a centralised command centre that orchestrates the workflow of the components and therefore the interplay of the components cannot be autonomous. Choreography is fundamentally many times more difficult to implement, however for the purpose of this project autonomy was a clear requirements. Please refer to Appendix A for the architecture diagram of how the choreography pattern can be implemented for an interactive art installation. Furthermore, through choreography and event sourcing it is possible to replay the events of the gallery in form of a series of messages (JSON) and snapshot photographs.

In the language of art theory, one could say it is a "view" of reality as observed through the lens of a camera. It is no longer reality and more of a *virtuality* or as Auslander (2006) puts it in relation to photographs and video recordings; a *performance document*. Auslander references O'Dell (1997) and theorises that a document provides evidence that an *event* actually occurred and postulates a fragmentary and incomplete record that can be *reconstructed* (Auslander, 2006 & O'Dell, 1997). In case of a live video system, there is always a "processing" or in this case an "augmenting" component that tries to reconstruct what the camera sees into pixels or other potential. Auslander (2006) further describes that the relationship between the performance (event) and the document (view) are ontological. To reference Gilbert (1998), the document "is not only representationally accurate but ontologically connected to the real world" and this "allows it be treated as a piece of the real world, then as a substitute of it."

Processing

The previous section discussed that a fragmentary record of what was perceived by the camera along with some meta-data was uploaded into the cloud as input. The project uses Microsoft Azure Cloud infrastructure to host its cloud software and therefore majority of the processing happens through a service oriented architecture.

The processing aspect of this project is primarily about AI assisted image processing. When correctly applied, AI can give much more accurate results in comparison to feature extracting techniques (He et. al., 2015). Cognitive Services API (see www.microsoft.com/cognitive-services) was used in this project, in particular the three main image processing services utillised are computer vision, face and emotion, which will be described below.

It is worthwhile to note that one of the prototypes used the OpenCV library and trained a feature-set to perform facial recognition. Although, the prototype was successful, it was later decided that its performance was not optimal and there were some lags when used in "live" video feeds. Furthermore, an attempt was made to train OpenCV through machine learning to perform emotion recognition, this proved to be a project by itself. In the process, Cognitive Services API was discovered and a quick decision was made to use the hosted service as these services are one of tools used in this project and not the aim of the project itself.

Computer Vision

Computer vision uses advanced algorithms for processing images and returns information about what it *perceives* from images. The information includes tags for more than 2000 recognizable objects, living beings, scenery, and actions. Ambiguous detections return hints to clarify the meaning of tags in the context in which they appear. There are no taxonomies or hierarchical inheritance for tags, however, the AI component is trained via the ImageNet dataset and 86-category taxonomies that classify visual features found in an image ranging from broad to specific. Examples of these categories include food, animals, people and buildings.

Cognitive services' computer vision uses deep residual learning (DRL) for image analysis as it is able to train much deeper neural networks for image classification and analysis. According to He et. al. (2015), by using DRL, they were able to obtain 28% relative improvement on Common Objects in Context (COCO) object detection dataset. The initial tests as part of this project, proved the reliability of the algorithms. However, as many other image analysis scenarios there were some errors. For example, a subject wearing a headset was misinterpreted by the computer vision to be a "person holding a mobile phone".

The margin of error within these algorithms, although may not be acceptable in some empirical situations, in art installation contexts, it is often desirable to set up ambiguous sets of stimuli about reality to engage the interlocutor of the artwork in a dilemma about that which is represented and that which is real. Enticing the viewers, into an interactive relation, with the artwork and to its authorship.

Face

The face detection service detects faces from images and returns the coordinates of each face as rectangles within the image. It also detects a set of face attributes by using machine learning to predict facial features. The facial features detected include pose, gender, age, smile and facial hair. Each face is identified by taking into account 27 landmarks for each face in the image. These landmarks can then be used for face recognition of the same face or similar faces in other images.

For the purpose of this project, due to ethics and privacy concerns neither face matching meta-data was stored nor any face recognition was used. However, within the context of the interactive art, interlocutors need not be notified of this, allowing for new potentialised forms of the art installation to manifest within the dynamics of interaction and perception.

Emotion

The emotion service uses machine learning over a large training set of facial expressions. It does not give a definitive set of emotions, but instead provides a confidence level which developers can use to interpret emotions. A higher confidence level indicates a high probability of an emotion and a lower confidence level may indicate false positives or “micro emotions” or transitioning emotions, that is when one emotional expression changes to another.

The emotion service utilises the face service behind the scenes to extract face rectangles first.

From a technical perspective once an event message (along with the snapshot) is uploaded into the cloud, it is simultaneously picked up by image analysis services that subscribe to these events. Please note that the project uses a microservices architecture and therefore it consists of many services that communicate via event messages as per the choreography pattern.

The choice of serverless microservice oriented architecture came down to best practice, effort and performance. Serverless functions are high performance and do not require any provisioned hardware in comparison to web services that are much slower and require hosting (Nadareishvili et. al., 2016).

The services that form the core of the system are listed below.

Analyse Image Service

The Analyse Image service, is notified of a new upload and starts execution. It converts the message into a dynamic object, extracts the image and the meta-data and once the message has been verified, the service calls the Image Analysis API for image analysis. The results of the images analysis are then parsed into a new JSON message and dropped in the Image Analysis Queue for further processing.

Analyse Face Service

Similar to the Analyse Image service, this service performs Face detection and parses the results into a new JSON message and uploads the new message on the Image Analysis Queue for further processing.

Analyse Emotion Service

Analyse Emotion service performs Emotion detection and creates a new JSON message for further processing. The message is then dropped into both the Emotion Analysis queue and Image Analysis queue for further processing.

The above three functions analyse the snapshots and upload them into emotion and image analysis queues. There are other functions that subscribe to these queues to pick up incoming messages.

Create Emotion Light Service

This service picks up emotion detection messages and performs a threshold analysis on the emotions. A threshold of 10% and above was used to ensure subtle and micro emotions are extracted. The service then assigns a colour code to each emotion as listed below:

Anger -> Red
Contempt -> Blue
Disgust -> Brown
Fear -> Purple
Happiness -> Yellow
Sadness -> Grey
Surprise -> Orange
Neutral -> White

This service further wraps the detected emotions and associated colours in a new JSON message and uploads them into the Light Queue for further processing.

Create Text Service

This service aggregates the emotion, face and image analysis tags and descriptions and converts them to human readable form. This is also the function that has had some aesthetic input as how to construct the text that is relevant to the art installation. For example, “you shouldn’t have married your wife” when “contempt” is detected. The text is wrapped in the new JSON message and is uploaded into the Text queue.

The above two services form the basis of processing. The light and text queue messages are later picked up and routed by Azure IoT Hub to the lights and the projector. IoT Hub is a cloud service that holds a register of IoT devices and is responsible for device security, detection, registration and communication.

Azure IoT Hub was chosen to simplify the cloud to device communication. Without the Azure IoT Hub, every device would have needed to open a direct socket to the cloud and keep it alive. IoT Hub abstract away all that complexity and also provides device security which is often overlooked in IoT.

The messages from the Light and Text queues are pickup by the following output services.

Send Light Service

The purpose of this service is to pickup the Light messages and ultimately send them to the light bulbs through the IoT Hub. This is the service that determines when and how the lightbulbs are switched on/off or change colour. The messages will be received by the “Light” software described in the Output section below.

Send Text Service

The purpose of this service is to pick the descriptive narratives and ultimately send them to the Text projector device through the IoT Hub. This service is a pass-through service and does not do much processing. The reason for this is that the “Typewriter” software described below is responsible for the aesthetic display of the text generated.

Output

There are two outputs of the system: lights and projection.

The lights are IoT connected Phillips Hue LED light bulbs that are controlled via the Hue Bridge. The Hue Bridge connects to the local Wifi or Ethernet networks and opens a bridge between the lightbulbs and the internet. In this case, the bridge is connected to a Raspberry Pi 3 computer board. Raspberry Pi 3 was build using Windows IoT operating system and a piece of software called “Light” was written in C# .NET programming language to control the lights. The Raspberry Pi 3 is registered with the IoT hub and subscribes to Light Queue messages coming from the cloud.

Light software uses the Phillips Hue SDK to connect to the bridge and translates colour codes arriving from the cloud into light commands for the light bulbs and controls colours and light intensity.

The choice of Phillips Hue due to the author being in possession of the light bulbs which were previously used in an unrelated prototype for home security.

The projector is a normal projector with a VGA input cable connected to another Raspberry Pi 3 computer board. The Raspberry Pi 3 was build using Windows IoT operating system and a piece of software called “Typewriter” was written in C# .NET programming language to stream text to the projector.

Unlike “Light” and “Movement” software that are very lightweight, Typewriter has some business logic. The reason is that it needs to orchestrate the speed of text that is projected as well as creating the user interface in which the text is displayed. Furthermore, it performs some synchronization with the light bulbs. In effect, only when an emotion is reflected as text, the light changes colour. This is however ambiguous in the context of the art installation because in the case of subtle or micro emotions, the light can change colour faster than the text.

Participants

The participants of the system are the interlocutors of the final art installation. The participants will not need to click a button or be asked to interact. The system actively looks in the room for new events and then analyses what is in the room. Even if there is no one in the room, the system will still be able to detect some objects in the room.

The projector will be displaying what it sees in the room, however as soon as an emotion is detected, an event is sent to the light bulbs to change colour based on the detected emotion, this will be synchronised with the text projection.

The text that appears on the projection will have the same colour as the lightbulb, based on the previously detected emotion, therefore creating a link between the emotion and colour, hence encouraging further participation by the interlocutors.

Discussion

“Mixed reality” started as a speculative work to assess what augmented reality, artificial intelligence (AI) and Internet of Things (IoT) could do in the context of contemporary art. The process of speculation revealed three area of interest:

1. A formal review of physical augmented reality (PAR)
2. AI as means of encouraging participation
3. A working software architecture for augmented reality and IoT implementation

The best definition of augmented reality was deductively selected during the literature review as a system that augments “natural feedback to the operator with simulated cues.” It is argued that this definition does not make any inherent assumptions about reality (natural feedback), the operator (interlocutor) or augmentation (simulated cues). From this definition, it is clear that augmented reality systems have elements of interaction and participation.

PAR has been defined as an augmented reality system that is void of live-streaming of the real environment, instead the superimposition of the virtual is on the physical space. For example, a projector projecting text on the wall, augmenting the wall to become a screen. Or changing colours of a lightbulb in a dark room, which may create a change in perception. PAR requires an informational aspect and interaction as per the wider definition of augmented reality and these are the differentiating factors in comparison to static installations.

Furthermore, interactive art theory and the application of PAR within contemporary art was explored and PAR was paralleled to the concept of relational architecture. It is argued that PAR by itself is different from interactive art, as PAR can have informational (and gaming) contexts, whilst interactive art is concerned with aesthetical form and/or politics.

In a good installation of interactive art, the interlocutors should be *drawn* to participate and have a choice or already be participat*ing* as in more traditional passive consumption of artworks. An interactive artwork needs to have its own momentum and be "autonomous" to have true aesthetic politics. It has to be infinite. It needs to recreate itself based on its own tendencies. It has to create its own *movement* in the *dynamic form* in which it operates (Massumi, 2008).

The project, through the process of speculating with technology explored the utilization of AI based image analysis. The system, constantly scans the environment. Whether an emotion is detected or not, the projected text will continue to describe what it perceives. When the interlocutors are within the range of the camera, it is argued that they are already participating in the artwork by the assumption of *being* in the room. Therefore, by applying complex machine learning based image analysis techniques, analysed in real time with the power of the cloud, it is possible to create an autonomous system that encourages participation, without the need for a "please interact" sign or routine skeleton, head or eye gesture tracking. Furthermore, by using IoT and augmenting the environment by colour changing lightbulbs, perhaps inadvertently, there is an additional dimension to participation. It unfolds during the interaction, and that is detection of emotions. Once an emotion is detected, there is a more direct response with the change in the colour of the lighting in the room. This is infused with the aesthetics of the plinth, light bulb, chair, table and the projection, which as a whole constitute the interactive art installation.

Finally, the complex interplay of the vast number of technologies that enabled this project was depicted in the architecture diagram (see Appendix A). The cloud was utilized to provide near instant processing of the environment that utilized AI based image analysis to decode the environment, detect age, gender, facial feature and emotions.

The choreography pattern was selected to ensure the system is "autonomous" and in the process, in conjunction with the power of the cloud is infinitely scalable and extensible.

Furthermore, the communication between different parts of this autonomous system was only possible through an IoT architecture. A number of seemingly disconnect physical objects (i.e. light, projector and camera), interacted in a connected world through the IoT

Hub, which held a register of connected devices and was responsible for sending commands in form messages to each device.

The analysis of what was real and what was virtual in, *mixed reality*, will be left to the interlocutors. Are the emotions real or the colours of the light? Is the text telling the truth? What happens when one light transitions to another? Did an emotion cause the change or simple movements of the muscles? Where did is originate? Were the interlocutors faking the emotions in order to interact with the system? Perhaps everything is virtual and all forms are in motion and nothing is fixed? To quote Massumi "take movement. When a body is in motion, it does not coincide with itself. It coincides with its own transition: its own variation. The range of variations it can be implicated in is not present in any given movement, much less in any position it passes through. In motion, a body is in an immediate, unfolding relation to its own non-present potential to vary. That relation, to borrow a phrase from Gilles Deleuze, is real but abstract."

This takes us back to the process of speculation during this project, in which, *abstract* was chosen over superimposition of bodies, and the uncharted *physical augmented reality* was defined and paralleled to *relational architecture* and the aesthetics of *dynamic forms* with the help of AI based computer vision has taken us, through *movement* and reflection of lights on potential forms, from one *semblance* to another yet everything remains autonomous and forms recreate themselves based on their own tendencies and the *regimes of power* are not enforced during participation.

Post-Exhibition Reflection (Amended 11 July 2017)

An informal discussion took place with a few of the interlocutors during and after the exhibition, notably artists and students of art from AUT university. Majority of the interlocutors spend in excess of 15-20 minutes interacting with the work.

The Discussion section of this exegesis posed the question and anticipated a *mixed reality*. Two facets of this term are the technological AR/VR continuum discussion and the artistic/philosophical question of the "what is real". Specifically, the Discussion section mentions:

> *"The analysis of what was real and what was virtual in, mixed reality, will be left to the interlocutors. Are the emotions real or the colours of the light? Is the text telling the truth? What happens when one light transitions to another? Did an emotion cause the change or simple movements of the muscles? Where did is originate? Were the interlocutors faking the emotions in order to interact with the system? Perhaps everything is virtual and all forms are in motion and nothing is fixed?"*

Reflecting on the questions above, the interlocutors described not being able to tell what was real and what was not, but then at times realized the system was able to read their emotions (for example when there was a lot of laughter) and there was in fact a relationship between the colour of the text, the light bulb and emotions. Another observation was post-reactions to the changes in dynamic form and self-creating nature of the art work. For example, the choice of photos and text were indented to provoke emotional reactions to the art work. Majority of the interlocutors reported the choice of photos and text were exceptionally effective in enticing different emotions.

A full qualitative analysis of the informal discussions is out of the scope of this exegesis. However, readers are encouraged to watch five short video clips captured from the interaction of two emerging artists and lecturers Moata McNamara and Azadeh Emadi publicly available on YouTube.com with their permissions. The two interlocutors in the videos spent an unusually long period of time, around 30 minutes interacting with the work. This is extremely unusual in the art world. This is another evidence for the claim in the Discussion regarding "regimes of power" and trying to solve the problem of "participation" in interactive art, which was one of the core research questions: "AI as means of encouraging participation".

The videos can be watched here:
https://www.youtube.com/watch?v=-CK5OU1K7EA
https://www.youtube.com/watch?v=rNMRRTpmogk
https://www.youtube.com/watch?v=MAQx_w3tGr0
https://www.youtube.com/watch?v=XD1l126iF1A
https://www.youtube.com/watch?v=B3TB1c4-BNU

Future Work

There is no doubt that a very complex architecture and advance technologies enabled this project (please see Appendix A: Architecture Diagram). It is anticipated that cloud and IoT architectures will continue to evolve and perhaps become more simplified in the future. It will an interesting endeavor to enable the functions achieved in this exegesis without requiring advance technical knowledge of programming and IoT architecture. In an ideal situation, an artist should be able to use a simple visual tool to orchestrate such project.

This project utilized Microsoft Cognitive Services for emotion and image analysis. These technologies are as recent as 2016. Several other services have emerged since the inception of this project for example updates to Google Tensorflow and the new Amazon Rekognition. An analysis of the effectiveness of these algorithms will be useful to future projects that may need to utilize image or emotion analysis.

Movement, motion and potentiality of form was lightly reviewed in this exegesis. The study of what is real and what is not, in relation to *movement/motion* and effectiveness a physical augmented reality experiment to bring into light, the theory of Substantial Motion by 17th century Iranian philosopher and comparing it to Bergson's theory of movement should the focus of future research.

An example of *Substantial Motion* within physical AR is a simple projection on a wall. The wall being in its *actualised* form, through *motion*, is transformed into one of its *potential* dynamic forms, and that is to become another actualised form (i.e. a computer screen) in the interlocutor's *perception*. Mulla Sadra's "theory of substantial motion, aimed at refuting the very basis of the peripatetic understanding of nature in terms of fixed substances. Mulla Sadra considers substantial motion to be a gradual transformation occurring in the very inner structure of things. Thus, a thing or substance which is now in a certain ontological state is regarded by him to be undergoing a continuous and gradual inner transformation until it reaches a new ontological state. The whole process of this inner transformation is in reality a series of passings away and recreation". "He held that the natures of this world are changing in their essences and moving with regard to their substances, while their accidents follow them in this renewal, and so receive any change which occurs in the substances. These accidents are united with their substances in actualization in the same way that differentiae are unified with genus and consequently move in accordance with the movements of substances. Only existential objects are the real subjects which change differentiae and genus. In the entire progression of existence, therefore, each preceding mode of reality becomes a genus and loses itself in the succeeding differentiae. We mean by transubstantial motion the existential motion of substantial nature which is to be a state of flux from the very beginning of its existence. Existentially, every being is a unity and a totality. Interpreted existentially substance and accident form a numerical unity in the existing individual. Nothing can affect accident without affecting substance. If accidents change, so does substance. Existentially, therefore, transubstantial motion is basic. But does such change require an existential subject which remains stable and unaffected? The only plausible candidate is prime matter. But prime matter is nothing existential, but only pure potentiality. Existentially, then, the "subject" of transubstantial motion must be the motion itself. Or, if one prefers, there is no subject of such motion understood as something itself stable and inert. If there were such a thing, then one would be forced into either an infinite regress or a denial of motion on the existential level. When we turn our attention to "time," we note that for Mulla Sadra time is not a mere accident used in measuring locomotion. Instead, for Mulla Sadra, both time and motion are the basic constituents of finite existents. Mulla Sadra makes time another dimension of physical nature so that his view of physical bodies is four dimensional" (Dehbashi, 1981).

Future research can discuss the role of AI in participation and how *potential forms* can dynamically be created in conjunction with physical AR to recreate the theory of Substantial Motion into a scientific experiment.

References

Auslander, P. (2006). The performativity of performance documentation. *PAJ: A Journal of Performance and Art*, *28*(3), 1-10.

Azuma, R. T. (1997). A survey of augmented reality. *Presence: Teleoperators and virtual environments*, *6*(4), 355-385.

Bergson H. (1990). Matter and Memory*. Zone Books*. English Ed. First published 1896.

Borenstein G. (2012). Making Things See*. Marker Media*.

Bourg D. M. & Seemann G. (2004) AI for Game Developers*. O'Reilly*.

Callan R. (2003) Artificial Intelligence*. Palgrave Macmillan*.

Cervantes, H., & Hall, R. S. (2005). Technical concepts of service orientation. *Service-Oriented Software System Engineering: Challenges and Practices. Idea Group Publishing*, 1-47.

Cipolla-Ficarra, F. V. (Ed.). (2013). *Advanced research and trends in new technologies, software, human-computer interaction, and communicability*. IGI Global.

Craig, A. B. (2013). *Understanding augmented reality: Concepts and applications*. Newnes.

Das H. (1994). *Telemanipulator and Telepresence Technologies*, SPIE Vol. 2351 Bellingham, WA.

Dehbashi M. (1981). Mulla Sadra's Theory of Transubstantial Motion: A Translation and Critical Exposition". *ETD Collection for Fordham University.*

Deleuze G. (1997). Bergsonism. *Zone Books*.

Erl, T. (2005). *Service-oriented architecture: concepts, technology, and design*. Upper Saddle River, NJ: Prentice Hall/Pearson PTR.

Foucault, M. (1982). The subject and power. *Critical inquiry*, *8*(4), 777-795.

Gilbert H. (1998), *Bodies in Focus: Photography and Performativity in Post-Colonial Theatre.* Textual Studies in Canada 10–11: 18.

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future generation computer systems*, *29*(7), 1645-1660.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770-778).

Issarny, V., Georgantas, N., Hachem, S., Zarras, A., Vassiliadist, P., Autili, M., ... & Hamida, A. B. (2011). Service-oriented middleware for the future internet: state of the art and research directions. *Journal of Internet Services and Applications*, *2*(1), 23-45.

Jansen, N., Kramer, D., Barz, C., Niewiejska, J., & Spielmann, M. (2015, May). Middleware for coordinating a tactical router with SOA services. In *Military Communications and Information Systems (ICMCIS), 2015 International Conference on* (pp. 1-7). IEEE.

Jeremijenko, N. (2009). Amphibious Architecture. *New York. Retrieved May*, *20*, 2012.

Khun T. S. (1962). *The Structure of Scientific Revolutions.* University of Chicago Press.

Kipper G. & Rampolla J. (2012). *Augmented Reality.* Syngress.

Kluszczynski, R. W. (2010). Strategies of interactive art. *Journal of Aesthetics & Culture*, *2*.

Lan, L., Li, F., Wang, B., Zhang, L., & Shi, R. (2014, December). An Event-Driven Service-Oriented Architecture for the Internet of Things. In *Services Computing Conference (APSCC), 2014 Asia-Pacific* (pp. 68-73). IEEE.

Langer, S. K. (1953). *Feeling and form* (pp. 28-47). London: Routledge and Kegan Paul.

Lozano-Hemmer, Rafael. "Relational Architecture." *Performance Research* (1999).

Massumi, B. (2002). *Parables for the virtual: Movement, affect, sensation*. Duke University Press.

Massumi, B. (2008). The thinking-feeling of what happens. *Inflexions Journal*, *1*, 1-40.

Merleau-Ponty M. (2002) *Phenomenology of Perception.* Routledge Classics 2nd Ed. - English Ed. First Published 1962.

Milanovic, N. (2011). *Engineering Reliable Service Oriented Architecture: Managing Complexity and Service Level Agreements*. Information Science Reference.

Milgram, P., Takemura, H., Utsumi, A. & Kishino, F. (1995). Augmented reality: *A class of displays on the reality-virtuality continuum*. In Photonics for Industrial Applications (pp. 282-292). International Society for Optics and Photonics.

Minerva, R., Biru, A., & Rotondi, D. (2015). Towards a definition of the Internet of Things (IoT). *IEEE Internet Initiative*, (1).

Mullen T. (2011). *Prototyping Augmented Reality.* Sybex.

Nadareishvili, I., Mitra, R., McLarty, M., & Amundsen, M. (2016). *Microservice Architecture: Aligning Principles, Practices, and Culture*. " O'Reilly Media, Inc.".

Neapolitan R. & Jiang X. (2012) Contemporary Artificial Intelligence.

Newcomer, E., & Lomow, G. (2005). *Understanding SOA with Web services*. Addison-Wesley.

O'Dell K. (1970), *Displacing the Haptic: Performance Art, the Photographic Document, and the 1970s*. Performance Research 2, 1: 73–74.

Patil, N. V., & Kshirsagar, M. C. (2015). *Enterprise Application Integration using Service Oriented Architecture with Web Service Aggregation Pattern*.

Ross, C. (2009). Augmented reality art: a matter of (non) destination. *Digital Arts and Culture 2009*.

Searle, J. R. (1980). Minds, brains, and programs. *Behavioral and brain sciences*, *3*(03), 417-424.

Sherman, W. R., & Craig, A. B. (2002). *Understanding virtual reality: Interface, application, and design*. Elsevier.

Vallverdu J. (2010). *Thinking Machines and the Philosophy of Computer Science: Concepts and Principles.* IGI Global.

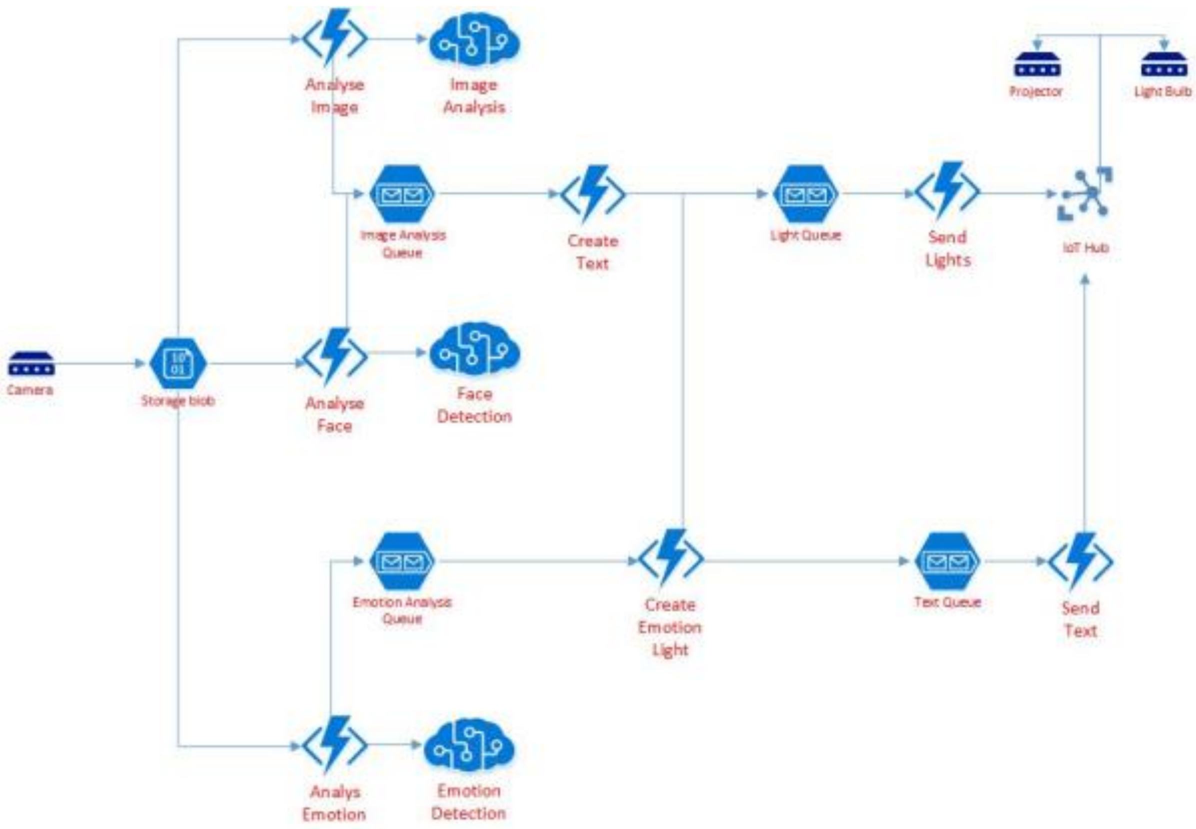
Weizenbaum, J. (1976). *Computer Power and Human Reason.* W W Freeman & Co.

Xia L., Chen C. C. & Aggarwal J. K. (2011). *Human detection using depth information by Kinect.* IEEE Computer Society Conference, 15, 20, 20-25.

Xin Wei, S. (2007). *Ouija*, Topological Lab.

Zizek, S. (1998). The interpassive subject. *Centre Georges Pompidou, Paris: Traverses.*

Appendix A: Architecture Diagram



Appendix B: JSON Message Format


```
{
  "id":"e2c3348e-3fff-4d23-9747-c071509c2bb1",
  "artifactId":"None",
  "locationId":"Main",
  "deviceId":"1",
  "timeStamp":"2016-11-07 00:36:20.995799",
  "image":"[…]"
}
```

Appendix C: Code snippets

Light Software code snippets

```
// receive messages from IoT Hub
public static async Task<string> ReceiveCloudToDeviceMessageAsync()
{
    var deviceClient = DeviceClient.CreateFromConnectionString
        (deviceConnectionString, TransportType.Http1);

    while (true)
    {
        var receivedMessage = await deviceClient.ReceiveAsync();

        if (receivedMessage != null)
        {
            var messageData = Encoding.ASCII.GetString(receivedMessage.GetBytes());
            await deviceClient.CompleteAsync(receivedMessage);
            return messageData;
        }

        await Task.Delay(TimeSpan.FromSeconds(1));
    }
}

// send messages to the light bulbs
public async Task SendCommandToLightsAsync(string message)
{
    try
    {
        dynamic emotionLightCollection =
            JsonConvert.DeserializeObject(message, typeof (ExpandoObject));

        var command = new LightCommand();
        command.TurnOn();

        ClearColours();

        if (emotionLightCollection.Anger.IsAlive)
        {
            command.SetColor(new RGBColor(emotionLightCollection.Anger.Colour));
            System.Diagnostics.Debug.WriteLine("Anger");
            AngerRec.Fill = new SolidColorBrush(Colors.Red);
        }
        else if (emotionLightCollection.Contempt.IsAlive)
        {
            command.SetColor(new RGBColor(emotionLightCollection.Contempt.Colour));
            System.Diagnostics.Debug.WriteLine("Contempt");
            ContemptRec.Fill = new SolidColorBrush(Colors.Green);
        }
        else if (emotionLightCollection.Disgust.IsAlive)
        {
            command.SetColor(new RGBColor(emotionLightCollection.Disgust.Colour));
            System.Diagnostics.Debug.WriteLine("Disgust");
            DisgustRec.Fill = new SolidColorBrush(Colors.Pink);
        }
        else if (emotionLightCollection.Fear.IsAlive)
        {
            command.SetColor(new RGBColor(emotionLightCollection.Fear.Colour));
            System.Diagnostics.Debug.WriteLine("Fear");
            FearRec.Fill = new SolidColorBrush(Colors.Blue);
```

```
        }
        else if (emotionLightCollection.Happiness.IsAlive)
        {
            command.SetColor(new RGBColor(emotionLightCollection.Happiness.Colour));
            System.Diagnostics.Debug.WriteLine("Happiness");
            HappinessRec.Fill = new SolidColorBrush(Colors.Yellow);
        }
        else if (emotionLightCollection.Sadness.IsAlive)
        {
            command.SetColor(new RGBColor(emotionLightCollection.Sadness.Colour));
            System.Diagnostics.Debug.WriteLine("Sadness");
            SadnessRec.Fill = new SolidColorBrush(Colors.Red);
        }
        else if (emotionLightCollection.Surprise.IsAlive)
        {
            command.SetColor(new RGBColor(emotionLightCollection.Surprise.Colour));
            System.Diagnostics.Debug.WriteLine("Surprise");
            SurpriseRec.Fill = new SolidColorBrush(Colors.Black);
        }
        else if (emotionLightCollection.Neutral.IsAlive)
        {
            command.SetColor(new RGBColor(emotionLightCollection.Neutral.Colour));
            System.Diagnostics.Debug.WriteLine("Neutral");
            NeutralRec.Fill = new SolidColorBrush(Colors.CadetBlue);
        }

        var lights = new List<string>() {"1", "2", "3"};

        //Send Command to lights

        await _client.SendCommandAsync(command, lights);
    }
    catch (Exception e)
    {
        //
    }
}
```

Typewriter Software code snippets

```
// subscribe to IoT hub messages
public MainPage()
{
    this.InitializeComponent();
    Task.Run(async () =>
    {
        while (true)
        {
            try
            {
                var message = await AzureIoTHub.ReceiveCloudToDeviceMessageAsync();
                await CoreApplication.MainView.CoreWindow.
                    Dispatcher.RunAsync(CoreDispatcherPriority.High, () =>
                {
                    textBlock.Text = message;
                });
            }
            catch (Exception e)
            {
                //
```

```
            }
        }
    });
}
```

Emotion Software code snippets

```
// start streaming
private void CameraStreamingButton_Click(object sender, RoutedEventArgs e)
{
    if (this.currentState == State.Streaming)
    {
        this.ChangeState(State.Idle);
    }
    else
    {
        this.ChangeState(State.Streaming);
    }
}

// If stream fails, shuts down streaming and returns to idle state
private void MediaCapture_CameraStreamFailed(MediaCapture sender, object args)
{
    var ignored = this.Dispatcher.RunAsync(CoreDispatcherPriority.Normal, () =>
    {
        ChangeState(State.Idle);
    });
}


// changes streaming state
private async void ChangeState(State newState)
{
    this.CameraStreamingButton.IsEnabled = false;

    switch (newState)
    {
        case State.Idle:

            this.ShutdownCamera();

            this.CameraStreamingButton.Content = "\xE768";
            this.currentState = newState;
            break;

        case State.Streaming:

            if (!await this.StartCameraStreaming())
            {
                this.ChangeState(State.Idle);
                break;
            }

            this.CameraStreamingButton.Content = "\xE71A";
            this.currentState = newState;
            this.CameraStreamingButton.IsEnabled = true;
            break;
    }
}
```

```
private async Task UploadToCloud(string filePath)
{
    try
    {
        var connectionString = "...";

        CloudStorageAccount storageAccount =
              CloudStorageAccount.Parse(connectionString);
        CloudBlobClient blobClient = storageAccount.CreateCloudBlobClient();
        // Retrieve a reference to a container.
        CloudBlobContainer container = blobClient.GetContainerReference("images");

        var rnd = new Random();
        var timeStamp = DateTime.UtcNow.ToString("yyyy-MM-dd-HH-mm-ss-ffffff",
              CultureInfo.InvariantCulture);
        var deviceId = "0001";
        var locationId = "0001";
        var fileName = $"{rnd.Next(1000000, 9999999)}-{deviceId}-{locationId}-
              {timeStamp}{".json"}";

        var blobName = fileName + ".json";
        CloudBlockBlob blockBlob = container.GetBlockBlobReference(blobName);

        using (var fileStream = File.OpenRead(file.Path))
        {
            ImageArtifact artifact = new ImageArtifact
            {
                id = Guid.NewGuid(),
                fileName = fileName,
                artifactId = "Hello!",
                deviceId = deviceId,
                locationId = locationId,
                timeStamp = timeStamp,
                image = GetByteArray(fileStream)
            };

            var json = JsonConvert.SerializeObject(artifact);
            await blockBlob.UploadTextAsync(json);
        }
    }
    catch (Exception e)
    {
        //
    }
}

// upload snapshots to the cloud
private async Task UploadToCloud(string filePath)
{
    try
    {
        var connectionString = "...";

        CloudStorageAccount storageAccount =
                    CloudStorageAccount.Parse(connectionString);
        CloudBlobClient blobClient = storageAccount.CreateCloudBlobClient();
        // Retrieve a reference to a container.
        CloudBlobContainer container = blobClient.GetContainerReference("images");

        var rnd = new Random();
        var timeStamp = DateTime.UtcNow.ToString("yyyy-MM-dd-HH-mm-ss-ffffff",
            CultureInfo.InvariantCulture);
```

```
        var deviceId = "0001";
        var locationId = "0001";
        var fileName = $"{rnd.Next(1000000, 9999999)}-{deviceId}-{locationId}" +
                       $"-{timeStamp}{".json"}";

        var blobName = fileName + ".json";
        CloudBlockBlob blockBlob = container.GetBlockBlobReference(blobName);

        using (var fileStream = File.OpenRead(file.Path))
        {
            ImageArtifact artifact = new ImageArtifact
            {
                id = Guid.NewGuid(),
                fileName = fileName,
                artifactId = "Hello!",
                deviceId = deviceId,
                locationId = locationId,
                timeStamp = timeStamp,
                image = GetByteArray(fileStream)
            };

            var json = JsonConvert.SerializeObject(artifact);
            await blockBlob.UploadTextAsync(json);
        }
    }
    catch (Exception e)
    {
        //
    }
}
```

Analyse Image Service code snippets

```
public static async Task Run(Stream imageArtifact, string name,
  IAsyncCollector<object> outputDocument,
  IAsyncCollector<string> outputImageAnalysisServiceBus,
  TraceWriter log)
{
  var reader = new StreamReader(imageArtifact);
  var json = reader.ReadToEnd();
  var artifact = JsonConvert.DeserializeObject<ImageArtifact>(json);
  var image = new MemoryStream(artifact.image);

  string result = await CallVisionAPI(log, image);

  if (String.IsNullOrEmpty(result)) {
    return;
  }

  dynamic document = JsonConvert.DeserializeObject(result, typeof(ExpandoObject));
  artifact.image = null;

  IDictionary<string, object> expandoObject = document;
  expandoObject.Add("artifact", artifact);
  expandoObject.Add("type", "vision");
  var message = JsonConvert.SerializeObject(expandoObject);
```

```
    await outputImageAnalysisServiceBus.AddAsync(message);
    await outputDocument.AddAsync(expandoObject);
}

static async Task<string> CallVisionAPI(TraceWriter log, Stream image)
{
    using (var client = new HttpClient())
    {
        var url = "https://api.projectoxford.ai/vision/v1.0/analyze
            @?visualFeatures=Categories,Tags,Description,Faces,
            @ImageType,Color,Adult&details=Celebrities&language=en";

        var content = new StreamContent(image);
        content.Headers.ContentType = new MediaTypeHeaderValue("application/octet-stream");
        client.DefaultRequestHeaders.Add("Ocp-Apim-Subscription-Key",
        Environment.GetEnvironmentVariable("Vision_API_Subscription_Key"));

        var httpResponse = await client.PostAsync(url, content);

        if (httpResponse.StatusCode == HttpStatusCode.OK)
        {
            return await httpResponse.Content.ReadAsStringAsync();
        }
    }

    return null;
}

public class Artifact
{
    public Guid id { get; set; }
    public string artifactId { get; set; }
    public string locationId { get; set; }
    public string deviceId { get; set; }
    public string timeStamp { get; set; }
}

public class ImageArtifact : Artifact
{
    public string fileName { get; set; }
    public byte[] image { get; set; }
}
```

Analyse Face Service code snippets

```
public static async Task Run(Stream imageArtifact, string name,
    IAsyncCollector<object> outputDocument,
    IAsyncCollector<string> outputImageAnalysisServiceBus,
    TraceWriter log)
{
```

```
    var reader = new StreamReader(imageArtifact);
    var json = reader.ReadToEnd();
    var artifact = JsonConvert.DeserializeObject<ImageArtifact>(json);
    var image = new MemoryStream(artifact.image);

    string result = await CallFaceAPI(log, image);

    if (String.IsNullOrEmpty(result)) {
        return;
    }

    dynamic documents = JsonConvert.DeserializeObject(result, typeof(ExpandoObject));
    artifact.image = null;

    foreach (var item in documents)
    {
        IDictionary<string, object> expandoObject = item;
        expandoObject.Add("artifact", artifact);
        expandoObject.Add("type", "face");
        var message = JsonConvert.SerializeObject(expandoObject);
        await outputImageAnalysisServiceBus.AddAsync(message);
        await outputDocument.AddAsync(expandoObject);
    }
}

static async Task<string> CallFaceAPI(TraceWriter log, Stream image)
{
    using (var client = new HttpClient())
    {
        var url = "https://api.projectoxford.ai/face/v1.0/detect?
            @returnFaceId=true&returnFaceLandmarks=true
            &returnFaceAttributes=age,gender,headpose,facialhair,glasses,smile";

        var content = new StreamContent(image);
        content.Headers.ContentType = new MediaTypeHeaderValue("application/octet-stream");
        client.DefaultRequestHeaders.Add("Ocp-Apim-Subscription-Key",
        Environment.GetEnvironmentVariable("Face_API_Subscription_Key"));

        var httpResponse = await client.PostAsync(url, content);

        if (httpResponse.StatusCode == HttpStatusCode.OK)
        {
            return await httpResponse.Content.ReadAsStringAsync();
        }
    }

    return null;
}
```

Analyse Emotion Service code snippets

```
public static async Task Run(Stream imageArtifact, string name,
   IAsyncCollector<object> outputDocument,
   IAsyncCollector<string> outputImageAnalysisServiceBus,
   IAsyncCollector<string> outputEmotionAnalysisServiceBus,
   TraceWriter log)
{
   var reader = new StreamReader(imageArtifact);
   var json = reader.ReadToEnd();
   var artifact = JsonConvert.DeserializeObject<ImageArtifact>(json);
   var image = new MemoryStream(artifact.image);

   var result = await CallEmotionAPI(log, image);

   if (String.IsNullOrEmpty(result)) {
      return;
   }

   dynamic documents = JsonConvert.DeserializeObject(result, typeof(ExpandoObject));
   artifact.image = null;

   foreach (var item in documents)
   {
      IDictionary<string, object> expandoObject = item;
      expandoObject.Add("artifact", artifact);
      expandoObject.Add("type", "emotion");
      var message = JsonConvert.SerializeObject(expandoObject);
      await outputImageAnalysisServiceBus.AddAsync(message);
      await outputEmotionAnalysisServiceBus.AddAsync(message);
      await outputDocument.AddAsync(expandoObject);
   }
}

static async Task<string> CallEmotionAPI(TraceWriter log, Stream image)
{
   using (var client = new HttpClient())
   {
      var url = "https://api.projectoxford.ai/emotion/v1.0/recognize";

      var content = new StreamContent(image);
      content.Headers.ContentType = new MediaTypeHeaderValue("application/octet-stream");
      client.DefaultRequestHeaders.Add("Ocp-Apim-Subscription-Key",
      Environment.GetEnvironmentVariable("Emotion_API_Subscription_Key"));

      var httpResponse = await client.PostAsync(url, content);

      if (httpResponse.StatusCode == HttpStatusCode.OK)
      {
         return await httpResponse.Content.ReadAsStringAsync();
      }
```

```
    }

    return null;
}
```

Create Emotion Light Service code snippets

```
public static void Run(string message, IAsyncCollector<string> serviceBus, TraceWriter log)
{
    var scoreThreshold = 0.1F;
    dynamic item = JsonConvert.DeserializeObject(message, typeof(ExpandoObject));

    if (item.type == "emotion")
    {
        if (item.scores.anger > scoreThreshold) {
            var emotionLight = new EmotionLightCollection();
            emotionLight.Anger.IsAlive = true;
            serviceBus.AddAsync(JsonConvert.SerializeObject(emotionLight));
        }
        if (item.scores.contempt > scoreThreshold) {
            var emotionLight = new EmotionLightCollection();
            emotionLight.Contempt.IsAlive = true;
            serviceBus.AddAsync(JsonConvert.SerializeObject(emotionLight));
        }
        if (item.scores.disgust > scoreThreshold) {
            var emotionLight = new EmotionLightCollection();
            emotionLight.Disgust.IsAlive = true;
            serviceBus.AddAsync(JsonConvert.SerializeObject(emotionLight));
        }
        if (item.scores.fear > scoreThreshold) {
            var emotionLight = new EmotionLightCollection();
            emotionLight.Fear.IsAlive = true;
            serviceBus.AddAsync(JsonConvert.SerializeObject(emotionLight));
        }
        if (item.scores.happiness > scoreThreshold) {
            var emotionLight = new EmotionLightCollection();
            emotionLight.Happiness.IsAlive = true;
            serviceBus.AddAsync(JsonConvert.SerializeObject(emotionLight));
        }
        if (item.scores.sadness > scoreThreshold) {
            var emotionLight = new EmotionLightCollection();
            emotionLight.Sadness.IsAlive = true;
            serviceBus.AddAsync(JsonConvert.SerializeObject(emotionLight));
        }
        if (item.scores.surprise > scoreThreshold) {
            var emotionLight = new EmotionLightCollection();
            emotionLight.Surprise.IsAlive = true;
            serviceBus.AddAsync(JsonConvert.SerializeObject(emotionLight));
        }
        if (item.scores.neutral > scoreThreshold) {
            var emotionLight = new EmotionLightCollection();
```

```
            emotionLight.Neutral.IsAlive = true;
            serviceBus.AddAsync(JsonConvert.SerializeObject(emotionLight));
        }
    }
}
```

Create Text Service code snippets

```
public static void Run(string message, IAsyncCollector<string>
    serviceBus, TraceWriter log)
{
    var scoreThreshold = 0.1F;
    dynamic item = JsonConvert.DeserializeObject(message, typeof(ExpandoObject));

    string emotionsString = "";
    if (item.type == "emotion")
    {
        var prefix = "you look ";
        if (item.scores.anger > scoreThreshold)
            emotionsString += prefix + "angery ";
        if (item.scores.contempt > scoreThreshold)
            emotionsString += prefix + "contempt ";
        if (item.scores.disgust > scoreThreshold)
            emotionsString += prefix + "disgusted ";
        if (item.scores.fear > scoreThreshold)
            emotionsString += prefix + "fearful ";
        if (item.scores.happiness > scoreThreshold)
            emotionsString += prefix + "happy ";
        if (item.scores.sadness > scoreThreshold)
            emotionsString += prefix + "sad ";
        if (item.scores.surprise > scoreThreshold)
            emotionsString += prefix + "surprised ";
        if (item.scores.neutral > scoreThreshold)
            emotionsString += prefix + "indifferent ";
    }
    else if (item.type == "face")
    {
        //emotionsString += item.faceAttributes.gender + " ";
        if (item.faceAttributes.facialHair.beard > 0.6)
            emotionsString += "beard ";
        if (item.faceAttributes.facialHair.moustache > 0.6)
            emotionsString += "moustache ";
        emotionsString += item.faceAttributes.glasses + " ";
        if (item.faceAttributes.facialHair.beard > 0.6)
            emotionsString += "sideburns ";
    }
    else if (item.type == "vision")
    {
        foreach (var t in item.tags)
        {
            emotionsString += t.name.ToString() + "  ";
```

```
      }
    }

    log.Info(emotionsString);
    serviceBus.AddAsync(emotionsString);
}
```

Send Light Service code snippets

```
public static void Run(string message, TraceWriter log)
{
    log.Info(message);
    serviceClient = ServiceClient.CreateFromConnectionString(connectionString);
    var commandMessage = new Message(Encoding.ASCII.GetBytes(message));
    serviceClient.SendAsync("LightMain", commandMessage);
    log.Info("End");
}
```

Sent Text Service code snippets

```
public static void Run(string message, TraceWriter log)
{
    log.Info(message);
    serviceClient = ServiceClient.CreateFromConnectionString(connectionString);
    var commandMessage = new Message(Encoding.ASCII.GetBytes(message));
    serviceClient.SendAsync("TypeWriterMain", commandMessage);
    log.Info("End");
}
```