University of Nottingham, School of Computer Science

G53IDS Interim Report: Objects as Controllers

Luke James Geeson
lxg03u@cs.nott.ac.uk
psylg1@nottingham.ac.uk
4201157

December 10, 2015

With Supervision from Michel Valstar
A Project on Machine Learning, within the School of Computer Science

Submitted in December 2015 in partial fulfilment of the conditions of the award of the degree of MSci (Hons) Computer Science.
I hereby declare that this dissertation is all my own work, except as indicated in the text:

Signature: _____

Date: _____ / _____ / _____

Contents

1 Statement of the Research Problem

1.1 Project Brief

Many systems these days are designed to work with specific devices or controllers in order to function. With developments in portable devices and wider ranging methods of control, it becomes increasingly difficult to maintain these systems whilst keeping up to date with modern interfacing trends. The aim of this project is to research, propose and use novel tracking techniques in an application that makes use of topics in Machine Learning, Computer Vision and Human Computer Interaction. The application will demonstrate effective use of existing and novel techniques in order to track moving objects in 3 dimensional space. A system will be built which will make use of these techniques in order to identify and use arbitrary objects in a given space. Once identified these objects will be tracked and their patterns of use recorded. These patterns will be recalled at a later date and use as mappings onto other programs.

1.2 Interpretation of the problem

The intent of this project is to develop a new form of system control in order to explore new means by which we can control programs without a dependence on any specific device as input. The project is composed of 3 components, which roughly correspond to the fields mentioned above: Firstly, Computer vision will be used to input and model objects and track them in 3 dimensional space. This information is then fed into the Machine learning component of the project which will be used to store and identify objects as well as recall objects for later use. An adaptive knowledge base system will be used to ensure quick retrieval of the data for each object. Once object data has been stored, the objects will be moved in real time and the gestures tracked in order to determine and learn the patterns of use. These patterns will then be used as mappings onto other components and hence used as controllers. The Human Computer Interaction portion of this project aims to unify the various technologies of these subsystems in order to create an intuitive and efficient user experience.

2 Research, Background work and Motivations

2.1 Research and Existing work

Significant work has been done on each of the respective sub-problems in each of the fields mentioned above. For instance substantial work has been done on region based variability in 3 dimensional images [1] such that algorithms exist to solve the problem of 3 dimensional tracking. Likewise, the field of machine learning is well established and provides many general solutions to problems in many areas, such as hand gesture recognition [2].

Furthermore, significant work has been done with motion tracking systems such as those present on Nintendo systems or the Microsoft Kinect, however the controllers in these cases are, in general, *directly* linked to these systems, or prove too restrictive for general use (restrictive in this case equates to *specific* implementations which could be *difficult to generalise* for other purposes without prior knowledge of the system in question). Furthermore, with a reintroduction of virtual reality systems into popular media and research, and an increasing reliance on purpose built systems [3] It becomes evident then, that there is a problem of *tight coupling* and a dependence on purpose built systems which can solved in one of two ways: Developing systems and *maintaining* them over time or else initiating the rapid and repetitive *production* of custom built systems, such as the current array of systems made by Nintendo over the last 10 years, in order to serve the growing demand to interface with reality in a novel way.

Both of the options above are complex in all senses of the word and incredibly resource intensive such that a more general solution should be considered. The solution in question involves the generalisation of the notion of a controller, such that the controller is not directly linked to the system it aims to control, but rather *the machine* learns of, models and maps the gestures made by the controller to the specific application in question. In reality, this means a system will be made that will be able to learn of new controllers (which can be arbitrary objects), new gestures (which can be specific to each object) and new mappings to new or existing programs in order to control them. In general, this solution should make it easier for more users to control a system, as it gives them more options to do so and for a longer period; but more importantly it emphasizes an approach to Human Computer Interaction whereby the *machine* learns about the trends of the user(and not vice versa), and thus produces a system that is catered to the user in every instant *without* the need to make or maintain numerous systems beyond simply adding new mappings to new systems as they are invented.

This project is part of a larger research project within the School of Computer Science at the University of Nottingham, however the aforementioned intent and intuitions are more general, and the processes could easily be reused in developing similar systems and in further work. As a result, other work exists in this larger scope, but none specifically on the ML or HCI components presented therein. It should also be noted that this project is currently in the process of being worked into a patent, and as such the information presented may be restricted, or in the least limited until otherwise stated. From now on, the larger team project which this work will be used with will be referred to as the *Unilever Augmented Mirror* or the *UAM project* for short.

Beyond the research presented in the previous section, similar systems do exist, including work in a patent by Sony [4] which largely achieves many similar features of this system, including the use of arbitrary objects as controllers. However, the details of this patent [5] suggest that the use case of such an invention is geared towards game playing, and limited therein only to that field. The aim of the system mentioned in this proposal, however, is to produce a system generic enough to be used for *any* purpose, as either specified by the user or a higher entity such as an organisation who wish for many users to interact in real time across many media at once. This is dissimilar to Sony's project in that

it is the intent that the *users* specify how to use a system, and the *machine* learns this, rather than the designers of the system itself. Similar to this, Microsoft have have been working on Natal (now known as Kinect) [6] which makes full use of motion gestures using only hands. This system is also geared towards game playing but serves as a testament to the popularity of Computer Vision. The project (defined in this dissertation), unlike those mentioned above will have a focus on the ML and HCI aspects, and more specifically the automated learning of gestures of objects in a 3 dimensional space. It must be said, that this wealth of knowledge, and the underlying academic material will be considered when looking at a way to approach this problem.

For the purposes of this project we will be using a Microsoft Kinect 2.0, due both to internal decisions (with respect to the UAM project) and the quality of this sensor and accompanying documentation. Not only has the relative success of this sensor become evident in industry [6] but it also has a wealth of support in the form of existing projects, documentation and common use today.

2.2 Motivations

Considering the above, it is possible to consider some use cases for this system, which may be useful future goals. Three of these use cases are:

1. **Alternative Controllers:** This system may be useful for those who have physical disabilities, and would as such be unable to interact with a system using the provided controllers. For instance, this system could be used in the instance where the user cannot hold a game controller, whereby the machine would learn of and use the gestures of another item that the user could hold instead.

2. **Generalised Predictive tracking:** Another case where this may be useful is in predictive object tracking, or indeed any predictive component in the Machine Learning field. This project is a special case of supervised learning (Arguably more similar to reinforcement learning [7]) whereby the user supplies the data and also reinforces it with use; and indeed a common aim of supervised learning is prediction [7]. Given this, it could be reasonable to suggest that a general system such as this could be used in the future, and combined with supervised learning such that it could be possible to produce an N-Dimensional predictive tracking system. More specifically, this system could combine predictive tracking of 3-dimensional objects, and make predictions about other features of a data point, such as temperature or sound output, in order to predict how an object may act. Such a system would also be useful in automating reactive simulators which may change as the environment does.

3. **Medical Use:** A simpler and generally useful case would be that of analysing how users interact with a system. Given the knowledge that the users determine how the system would act, this system could be used in order to find patterns of use amongst a populace. This data could be used for things such as predicting symptoms of physical disabilities (such as say, involuntary twitching) or determining patterns of use in a system in order to improve user interface design in the future.

3 Methodology

3.1 Preliminary Research

Throughout this project, it will be necessary to conduct some *user studies* in order to determine if the system being produced is intuitive enough for use. Given this project is, in part, about producing a system that can learn how users interact with it, it is also essential that the unchanging portions of the system that act in a deterministic way. This is done so as to minimise the effort of the user to learn how to teach the system gestures, but also to allow focus to stay on the automated learning of gestures in which both the user and the system designer are involved.
More research will be conducted, including but not restricted to the ideas above in order to obtain a full understanding of existing works and avoid reinventing the wheel. Therefore a portion of the project will be dedicated to *research and investigation*. This portion of the project will also be used as a time to experiment with some possible theories and solutions to the problem which may or may not be 'good' solutions and, time permitted, come up with a solution that best solves this problem in a *meaningful* way. More specifically, the research conducted will involve:

1. **Qualitative Surveys:** These will be used to sample a set number of users over the period of development for the system, this will be used to get feedback on the user experience in order to better improve it in the future.

2. **Technology development:** This will be used to learn how to use the technology required in this system, and determine the best way to solve the problem using this technology.

3. **Algorithm design:** Work will be done to produce an algorithm in order to predict the gestures of users given the relevant features of the environment. This algorithm will also be *generalised*, as far as possible so that it may be *reused* in future systems.

4. **System Implementation:** As a proof of concept and basis for the patent, the research conducted here is likely to be incorporated into the UAM project in the School of Computer Science at the University of Nottingham. As such, time will be spent in producing a system that is compatible with the larger system.

3.2 Development Process

Throughout the duration of this project, I will be the only one working on the various components of the project defined in this dissertation. As a result, it makes sense to adopt a development process that best suits a workstyle for one. The *Waterfall* process will therefore be used and the structure of this project will reflect that. The structure of the project is outlined in *Dissertation Structure*.

4 Progress Report

4.1 Progress to date

As mentioned in the previous sections, a portion of this project is dedicated to research and investigation in order to elicit the requirements of the project, the best technologies for the job and conduct some feasibility studies on past existing or novel solutions that solve part(s), or all of the problem. It is therefore important to emphasize the role these play and their sizable contribution to the progress made. The progress thus far is as follows:

1. **Dissertation Structure:** The structure of the project was outlined in terms of sections and subsections where explanations were given, describing the intended content. This is of course liable to change if the direction of the project itself changes but should serve as a general heuristic which builds upon the direction outlined in the motivations above.

2. **System Specification:** The *Functional* and *Non-Functional* requirements and specifications were determined and enumerated such that the *scope* outlined in the *Motivations* section is made explicit and thus the *objectives* of the project made clear. The elicitation of these specifications had an effect on the timeframe of the project such that it necessarily had to be adjusted to account for further research and implementation components.

3. **System Design:** Given the requirements of the system, it was necessary to formulate a design which would be followed in the implementation of the system. This also added to the *direction* of the project so that the *Feasibility Study* and *Prototyping* stages were better informed with respect to suitability. **Note:** The designs have not been included yet as they are incomplete, but discussions have been made with my Supervisor.

4. **Feasibility Study:** An evaluation of current novel and existing technologies and systems is made in order to best determine the extent to which they solved the problem(s) outlined in the *Specifications*. A decision was also made on which of these would be used in order to progress with this project and solve the problems outlined therein. **Note:** This has not been included in this report as it is incomplete thus far.

5. **Prototypes:** Basic prototyping was done following the designs of the system to trial run some of the technologies chosen for parts of the project and evaluate the scope, direction and projections of the project. **Note:** This section is incomplete but some evidence is provided of initial prototypes.

6. **Planning for the second half of the project cycle**

Details of these can be found below

4.2 Planning for the second half of the project cycle

The second half of the cycle will be composed largely of implementation steps in order to realise the design. More specifically, the following will be done:

1. **Components outlined in the Dissertation Structure**: those which are not yet present in this document or otherwise complete which have been listed in *Dissertation Structure*

2. **Software Implementation:**

    2.1. **Computer Vision Steps:** It will be necessary to make use of the computer vision technologies explored in order to do the following:

2.1.1. Vision will be required in order to identify the object in 3D space, many technologies and ideas explored in the design and research phase will be implemented in a system that can detect arbitrary objects in the hands of users of the system.

2.1.2. Vision will be required to track the object in 3D space, once the datapoint is identified by the computer vision techniques above. It will be necessary to use computer vision techniques to track this point in space. It will be necessary to implement a system that can record the locations of the object as it travels through space

2.2. **Machine Learning Steps:** Once a system is developed that can effectively identify and track an object in 3d space, it will be necessary to develop a system that can automatically identify these objects, save instances of these objects and recall them at a later point. A system will be developed which is able to automatically infer what object the user is holding. Similarly, it will be necessary to create a system that can learn the gestures made by the objects and recall them at a later date such that they may be mapped to programs when they are made.

2.3. **HCI Steps:** It will be necessary to develop a system that will do two things. an interface will be built for components in this so they may be used without the need to interface with the system using traditional methods (keyboard, mouse etc...). Secondly, it will be necessary to integrate this system with the UAM project so that the system may be used with this system in a intuitive way. More specifically:

2.3.1. Speech Recognition components will need to be developed so that object identification, learning and recall may be invoked by the user with speech commands. A full grammar will need to be produced and implemented in code using the technologies explored below.

2.3.2. Speech Recognition components will also need to be developed for the learning and recall of gestures in a similar fashion to the previous point. It will also be necessary to implement some mappings such that the gestures can be used as controllers for arbitrary programs.

2.3.3. This whole project must be integrated into the UAM project so that it can be used in an intuitive manner with the system. Code will need to be developed so that this project can be invoked from the larger system with ease, likely using voice commands.

3. **Integration and Testing Steps:** The whole system will need to be integrated into the UAM project so that it may be used with this project. This will involve developing a system hierarchy that satisfies the organisational constraints of the UAM project and integrating it into the this system so that the work may be used by others.

4. **Evaluation steps:** An evaluation of the work done will be made in order to see how what future things can be done as a result of this work. This will involve taking qualitative surveys and reviewing the system as a whole in relation to the larger project.

5 Dissertation structure

As mentioned before, this project will adopt the *Waterfall* engineering methodology and will thus proceed in a linear fashion. More specifically, the project will proceed with sequential steps, where each step will solve a particular problem in the process. The structure will thus be as follows:

1. **Abstract:** A section for summarising the goal and result of the project.

2. **Introduction/Statement of the Research Problem:** A section for introduction of the Research Problem and how it is interpreted.

3. **Research, Background work and Motivations:** An introduction to the context of the project, in order to establish scope, direction and motivations.

   3.1. **Research and Background work:** A look into some prior research done with respect to the similar problems in the areas identifed, with aims of differentiating *this* project from others that already exist. It should be noted here that this section is *not* dedicated to analysing the suitability of previous technologies to solve this problem.

   3.2. **Motivations:** A look into *why* this project is novel and some further considerations of further implications of this work.

4. **System Specification:** A section designed to make explicit the objectives of the project (and refine the scope).

   4.1. **Functional Requirements:** A section denoting the high level goals of the project

   4.2. **Non-Functional Requirements:** A section denoting the high level constraints on the project

   4.3. **Functional Specifications:** A section denoting the low level goals by which the high level goals may be achieved

   4.4. **Non-Functional Specifications:** A section denoting the low level constraints on which the *Functional Specifications* must be acheived.

5. **Feasibility Study:** A look into the suitability of previous technologies that could be used to solve the problems defined in the *Specification*.

   5.1. **Analysis of previous whole solutions:** Analysis of *existing* systems that solve *whole* or *part* of the problem in some significant way as a standalone product, be it bespoke or a general solution.

   5.2. **Analysis of previous Technologies:** Analysis of existing *technologies* that solve specific parts of the problem in whole or in part. These components may function as standalone technologies or be integrated as general frameworks on per-system basis.

   5.3. **Decisions on technologies used:** A section denoting which technologies will be used to complete this project.

6. **System Design:** A section denoting the high level design that the system will follow (making the direction explicit)

   6.1. **System Design:** A high level overview of the whole system design

   6.2. **Individual design components:** A low level look at how each individual component in the system will work

7. **Prototyping:** A section denoting to prototyping some ideas that came about from the exploration of technologies and issues in design. This section will be brief but will demonstrate some use of technologies to try and solve some of the issues in the project.

8. **Software Implementation:** A section, roughly following the sections set out in the design denoting all components implemented in the system.

9. **Integration and Testing:** A section denoted to making explicit the steps taken to integrate the system into the larger project and test it, using unit, system, integration, user and acceptance tests according to the the scope defined by the previous sections.

10. **Evaluation and a Personal Reflection:** A close evaluation of how the project went, and a personal reflection on my efforts.

11. **Conclusions and Further Work**: Lasting thoughts on the project and a look into further work that could be done.

12. **Appendices:** Additional information about tests, surveys and complimentary works completed for this project

6 System Specification

As mentioned in the Research Problem, the system being produced will be an application using existing and novel tracking technologies in order to track arbitrary objects in A 3-dimensional space. Further to this, the system should aim to learn and recognise new objects and gestures so that they may be mapped to arbitrary programs. Lastly, it is intended that this system will be worked into the UAM project and hence some plumbing is required in order properly integrate and interface this component with the larger system. In this section the *functional* requirements (the goals of the system) and the *non-functional* requirements (the constraints placed upon the system) have been made explicit and enumerated such that the scope outlined in the motivations section was made explicit and thus the objectives of the project made clear. The project itself has 3 main components: The Speech recognition component, The object identification and tracking and the machine learning system that can learn and recall the above, as such these components roughly define the main areas from which requirements can be drawn.

6.1 Functional Requirements

The requirements of the system have been specified here, these can be used later in the evaluation stages in order to determine progress.

1. **Speech Recognition:** Users will be able to control the system using speech recognition software
   - 1.1. Users will be able to teach the machine to learn about a new object
   - 1.2. Users will be able to teach the machine to recognise a gesture
   - 1.3. Users will be able to perform some image processing on the input image
2. **Object Identification:** The System will be able to detect arbitrary objects in a 3-dimensional space
   - 2.1. The system should be able to determine what hand the object is being held in
   - 2.2. The system should detect an object in the hand of the user (only one hand will be used at a time)
   - 2.3. The system should detect the size of the object
   - 2.4. The system should detect the distance the object is from the sensor
   - 2.5. The system should detect objects of an arbitrary shape and colour
   - 2.6. [OPTIONAL] The system should be able to automatically construct a 3d representation of the object
3. **Object Tracking:** The System will be able to track arbitrary objects in a 3-dimensional space
   - 3.1. The system should use the centrepoint of the object in a 3 dimensional cartesian plane in order to determine where the object is in space
   - 3.2. The system should use a sequence of previous coordinates in order to determine the direction the object is moving in
4. **Automated Learning and Inference of objects:** The System will be able to recognise and learn of arbitrary objects and recall them when prompted with the aid of user prompts
   - 4.1. The system should be able to use the information about the object identified in order to automatically construct a profile for that object. Features recorded should include:

4.2. The system should be able to store this information in an adaptive knowledge base which can be queried at a later point to recall this information

4.3. The system should be able to remove records of objects if requested by the user.

4.4. The system should be able to automatically infer which object the user is holding from the database

4.5. The user should be able to invoke the action of learning a new object via use of the speech recognition system

5. **Automated Learning and Inference of gestures:** The System will be able to learn gestures for a particular object and map them to arbitrary programs

5.1. Given an object has been identified, it should be possible for the user to teach the machine to recognise gestures made by moving the object in 3-dimensional space

5.2. The user should be able to invoke the act of learning gestures by using speech recognition

5.3. The system should track the location of the object in 3-dimensional space over time, until otherwise stated by the user

5.4. The gesture made by the user should be saved into a knowledge base system so that the pattern of use may be recorded later on

5.5. The user should be able to map a given gesture to an action or trigger, such as running a program or requesting a response from a service

5.6. The system should be able to automatically infer which gesture the user is making with a given object and invoke a trigger or action in response to this. It is assumed that the gesture has already been learnt by this system or the inference would fail.

5.7. The user should be able to map gestures to programs via the use of speech recognition

6.2 Non-Functional Requirements

The high level constraints of the system have been outlined here:

1. **Accessibility:** The system should be easy to use and readily accesible to any user of the team project. More specifically this project should be properly integrated into the UAM project such that any user of that system may also use the components of the system described here without any barriers to use.

2. **Usability and Interoperability:** This project will be worked into a system that will be used by many non-technical users, as a result the user experience should be simple, intuitive and effective in its operation. More specifically, the components of this system that will not change (all components not directly related to the ML components) should be simple to use and consistent in their operation. The end goal of the project is not to need to use any computer system directly via standard methods (keyboard, mouse, monitor etc...) such that an emphasis must be placed on the speech recognition components in order to provide a pleasant and obvious user experience.

3. **Quality:** It is the intension that this project is worked into a commercial product (the UAM project) which will distributed across the world. As a result, the quality of the end product must accurately reflect the author, the school of Computer Science and the University of Nottingham. To this end, good engineering process and practice must be followed and the system designed and implemented to a professional standard. In addition, the product must be robust, intuitive to use (see the Usability requirement), modular and efficient.

4. **Generality:** Whilst it has been stated that this project will be an application, which itself is part of the UAM project. It should be enforced that the use of this application is made as general as possible. Not only to demonstrate a system that learns how the user uses it, but also to be used as a basis on further works, such as those outlined in the *Motivations* section. As a result, the system should aim to identify and track any object (in any environment), learn and trace any gesture and map these to *any* program.

5. **Resource based constraints:** There is a number of resource based constraints on this system, most of them due to the fact that the UAM project had been going for sometime before I joined the team, the resource based constraints are as follows:

   5.1. **Use of the Kinect:** Due to internal decisions within the School of Computer Science pertaining to the nature of this project, this project is using a Microsoft Kinect v2 [6] as the primary sensor. This project should make use of this system when implementing the requirements stated above.

   5.2. **Use of the Host Operating System:** The team project is being completed on a Windows based operating system. As a result, any work completed in this project must be work on a Windows based operating system.

   5.3. **Auxiliary hardware and software:** The UAM project is using a range of technologies and hardware, it is therefore necesary that the result of this project, where it interfaces with these components, is compatibile with them so as to enable use.

   5.4. **Auxiliary structures:** The UAM project has auxiliary measures put in place in order to regulate and maintain the project so that it may be monitored, assessed and used by other members of the team. This project should aim to work within the scope of these structures such that consistency is maintained.

6. **Security:** There are no immediate security errors with this system as no personal information is being stored beyond arbitrary user names and the objects stored for each user. The UAM project will have a login system in order to enforce access rights, personal information will be stored and used in that component. However, no personal will be used in the system outlined in this project and hence it is not an issue that needs discussing.

7. **Documentation:** The system will need to be documented with both internal and external documentation. Internal documentation will be required for technical users so that they may maintain the system in the future, external documentation will be required for non-technical users who wish to use the system (as mentioned above, this will be used in a commercial setting and so this will be necessary).

6.3 Functional Specifications

This section denotes the functional specifications for the system, it uses the high level functional reqiurements as headings and adds to them such that specific detail is supplied about how these problems would be solved.

1. **Speech Recognition:** Users will be able to control the system using speech recognition software

   1.1. Users will be able to teach the machine to learn about a new object

      1.1.1. Users should be able to request that the machine should attempt to look for an object in the hand of the user

1.1.2. Users should be able to request that an object identified in the previous point is learnt by the machine by extracting features. Users must specify a UNIQUE ID by which they can identify the object later.

1.2. Users will be able to teach the machine to recognise a gesture

1.3. Users will be able to perform some image processing on the input image, including:

1.3.1. Image Segmentation by Colour and Depth,

1.3.1.1. The user should be able to perform colour segmentation on the image by speech request where the user specifies the colour in the image which they wish to detect. The set of accepted colours include: *RED, BLUE, GREEN, YELLOW,PURPLE, ORANGE, WHITE, GREY, BLACK*

1.3.1.2. The user should be able to perform depth segmentation on the image by speech request, where a depth is supplied either by the user or from a previous command

1.3.2. Image Segmentation by Skin (for the purposes of differentiation from the object)

1.3.3. The user should be able to perform image segmentation on skin of the user(s) in the image by speech request.

1.3.4. Image Segmentation of the Object held in the hand of the user (Hand detection).

1.3.4.1. The User should be able invoke object segmentation/identification by speech request.

1.3.4.2. Image processing to detect the hands of the user: The user should be able to detect and track hands in the image by speech request.

2. **Object Identification:** The System will be able to detect arbitrary objects in a 3-dimensional space

2.1. The system should be able to determine what hand the object is being held in

2.1.1. The system should be able to detect and track hands in the image, where each hand has is represented by a triple (x, y, z) corresponding to its location in the frame. This information should be calculated for each frame.

2.1.2. The system should be able to determine which hand has an object in it.

2.1.3. The system should be able to determine whether a hand has an object in it.

2.2. The system should detect an object in the hand of the user (only one hand will be used at a time)

2.2.1. The system must be able to perform image processing on *each* frame so that the object held in the hand of the user is identified and represented by a triple (x, y, z) corresponding to the center of the object in the image. The system should compute this for the *current* frame and *each subsequent* frame. From here on in this triple (x, y, z) shall be referred to as γ and known as the *centrepoint* of the object.

2.2.2. Once identified, the object should be identifiable at *any* location within *every* subsequent frame by γ (the *centrepoint* should be updated which every subsequent frame), regardless of whether the object is in the hands of the user or not. If the object is *not* in the image, the system should signify this to the user.

2.3. The system should detect the size of the object

2.3.1. The system must be able to determine the height and width of the object, to within 10cm of uncertainty on the actual size of the object in both x and y axis on the 2 dimensional plane.

2.3.2. the system must be able to determine the size of the object, once the object is identified and for every subsequent frame following that until otherwise stated by the user with a speech request.

2.4. The system should detect the distance the object is from the sensor

2.4.1. The system should be able to perform depth segmentation on the image by speech request, where a depth is supplied either by the user or from a previous command such that all objects within a small range of this depth should be highlighted and others ignored (made black on the image).

2.4.2. Depth segmentation must be used in order to determine the distance the object is from the camera.

2.5. The system should detect objects of an arbitrary shape and colour

2.5.1. The system should be able to extract the object from the image supplied and whilst taking care to *ignore any skin or background noise*.

2.5.2. The system should be able to represent the object as a *point* on a 3 dimensional cartesian plane (x, y, z) corresponding to the *centrepoint* of the image.

2.5.3. Once initially identified the object should be identifiable anywhere on the frame and any subsequent frames until otherwise specified by the user with a speech request.

2.6. [OPTIONAL] The system should be able to *automatically* construct a 3 dimensional representation of the object, which could achieve the following (also optional):

2.6.1. The system should be able to generate 3 dimensional models and assign features to them. The system should be able to determine features of the object using user aids and computer vision.

2.6.2. The system should determine which side of the object is currently visible, and automatically identify new or existing features of the object.

2.6.3. The system should determine the orientation and rotation of the object based on features visible on the object. An initial axis of rotation θ should be determined and updated as the orientation of the object is found. The orientation is defined to be a tuple R composed of a pair of points ϕ in a 3 dimensional plane (R is (ϕ, ϕ)). ϕ is defined here to be a triple (x, y, z) where x is the position in the horizontal plane, y is the position in the vertical plane, z the position in the third dimension. Given this, the system should be able to determine the axis of rotation θ in a 3 dimensional cartesian plane using any 2 arbitrary points ϕ_1 and ϕ_2 which compose R. The system should be able to determine this axis of rotation for each subsequent frame.

3. **Object Tracking:** The System will be able to track arbitrary objects in a 3-dimensional space

3.1. The system should use the *centrepoint* of the object in a 3 dimensional cartesian plane in order to determine where the object is in space

3.1.1. The system should have the facility to store previous values of γ from previous frames and use these in order to calculate the angle displacement of the object in relation to previous frames. The number of storable values of γ should be specifiable by the technical user. The value of number of storable values should be known as k.

3.2. The system should use a sequence of previous coordinates in order to determine the direction the object is moving in.

3.2.1. The system should make use of the previous values of γ in order to calculate the angle of displacement. The angle of displacement is defined to be a tuple δ composed of a pair of points φ on a 3 dimensional plane (δ is (φ_k, φ_{k+1})). φ is defined here to be a triple (x, y, z) where x is the position in the horizontal plane, y is the position in the vertical plane, z the position in the third dimension. Given this, the system should make use of the previous k values of γ in order to compute a new value γ_{k+1} which paired with the

current value of γ_k (the current frame) composes the *displacement* δ between which a line can be drawn and direction made explicit.

4. **Automated Learning and Inference of objects:** The System will be able to recognise and learn of arbitrary objects and recall them when prompted with the aid of user prompts

   4.1. The system should be able to use the information about the object identified in order to automatically construct a profile for that object. Features recorded should include:

      4.1.1. the *centrepoint* Position as defined above

      4.1.2. General Colour (A spectrum of colours present on the object). This may either be a spectrum containing the range of colours most present in the system or, if the [OPTIONAL] components of the object identification are satisfied a mapping of the colours present onto the 3 dimensional model generated for the object in question.

      4.1.3. General size, within some limits of uncertainty (owing to any uncertainty arising from use of the image processing tools to identify the image)

      4.1.4. General Shape, denoting the rough shape of the object (This may be provided by the user in order to aid the system in infering the shape)

      4.1.5. A UNIQUE ID, used to identify the object, if automatic inference fails this can be used to identify objects manually. It must be enforced that the ID for each object is *distinct* inside the database and hence cannot be used twice.

   4.2. The system should be able to store this information in an adaptive knowledge base which can be queried at a later point to recall this information

   4.3. The system should be able to remove records of objects if requested by the user.

   4.4. The system should be able to automatically infer which object the user is holding from the database

      4.4.1. If a user is holding an object and invokes the operation of recall on the object, the system should be able to automatically infer which object the user is holding from the database or else notify the user of failure if identification fails.

      4.4.2. If no object is being held by the user, then system should notify the user that this operation has failed

   4.5. The user should be able to invoke the action of learning a new object via use of the speech recognition system

5. **Automated Learning and Inference of gestures:** The System will be able to learn gestures for a particular object and map them to arbitrary programs

   5.1. Given an object has been identified, it should be possible for the user to teach the machine to recognise gestures made by moving the object in 3-dimensional space. The system should record the following values for each gesture:

      5.1.1. The system should be able to record γ for the current and previous frames against a timer, such that for each frame n (where 1 <= n <= k) a time value is stored. In other words, a sequence of $\gamma - timing$ pairs must be stored for the gesture.

      5.1.2. a UNIQUE ID must be recorded and assigned to the gesture being recorded, for the purposes of identifying the gesture when mapping it unto programs as a controller.

      5.1.3. if the [OPTIONAL] components of object identification and tracking specifications are completed, the *axis of rotation* should also be stored for each frame, thus extending the previous point so that a sequence of $\gamma - timing - \theta$ triples should be stored in this instant.

   5.2. The user should be able to invoke the act of learning gestures by using speech recognition

5.3. The system should track the location of the object in 3-dimensional space over time (reflective of the point above), until otherwise stated by the user

5.3.1. The sequence of *gesture points* recorded for each gesture should be normalised such that any gestures made by the user from that point onwards would be scaled onto a common plane (for purposes of analysis and generalising the application)

5.3.2. the user should indicate the *end* of recording by speech commands.

5.3.3. if recording of gestures is not complete by a *time limit* then it is assumed that the recording has failed, the system should enforce this.

5.4. The gesture made by the user should be saved into a knowledge base system so that the pattern of use may be recorded later on

5.5. The user should be able to map a given gesture to an action or trigger, such as running a program or requesting a response from a service

5.6. The system should be able to automatically infer which gesture the user is making with a given object and invoke a trigger or action in response to this. It is assumed that the gesture has already been learnt by this system or the inference would fail.

5.6.1. If a user is holding an object and invokes the operation of gesture recall with the object (by making a gesture), the system should be able to automatically infer which gesture the user is making from the database or else notify the user of failure if identification fails.

5.6.2. If identification has not succeeded after a given time limit, it is assumed that identification has failed

5.7. The user should be able to map gestures to programs via the use of speech recognition

5.7.1. The user should specify the name of a *gesture* and the name of a *program* onto which the gesture will be mapped. It is assumed that the gesture has already been learnt and that program exists and is stored in the location of the directory in which this application will run, the system will fail otherwise.

6.4 Non-Functional Specifications

The low level constraints on the functional specifications have been defined here, the non-functional requirements were used as headers under which more detail is specified if necessary.
NOTE: Non-functional requirements have been ommitted where it is *not* possible to specify a way in which they could be completed. Rather, the *extent* to which these have been satisfied will be discussed in the Evaluation section in the final report, or otherwise left as a decision for the reader/supervisor.

1. **Usability and Interoperability:** Grammars will be provided to use the speech recognition system, user guides will be provided for technical users. This dissertation along with commenting in code will serve as the remainder of the documentation for this project

2. **Generality:** Steps have been made explicit where generality is needed in the functional specifications above.

3. **Resource based constraints:**

3.1. **Use of the Kinect:** The kinect v2 will be used

3.2. **Use of the Host Operating System:**: Windows will be used, all code written will also be tested on these machines to ensure functionality.

3.3. **Auxiliary hardware and software:** interfaces, such as the speech recognition system will be defined where needed.

3.4. **Auxiliary structures:** The private *Git* repository will be used when significant progress has been made on the system.

4. **Documentation:** please see *Usability* above

7 Feasibility Study

This section is currently under completion and has hence not been included

8 System Design

I have been speaking with my supervisor on this to better define the direction of this project, this has not been manifest in a design on paper yet but will be done soon.

9 Prototypes

This section elaborates upon some of the prototyping done with the technologies chosen for this project. The prototyping methodology used here is in some parts *throw away*, whilst in others it is *evolutionary*, however the intuitions discovered may be translated over to later iterations. More specifically, prototypes have been done for speech recognition, object identification and the image processing involved to track the objects. No work has been done thus far on the ML component of the project as that requires that the components in the vision component are well implemented to the point where it would be possible to use them as a basis for developing the inference system.

9.1 Speech Recognition

One advantage of using the Microsoft Speech recognition API is that there is existing code which can be used as a basis for prototypes [8]. More specifically, in the SDK browser, demo code is provided which allows you to issue commands to the Kinect in order to move a Turtle around within a window [9]. The first action was therefore to construct a simple Grammar [1] and substitute the values in this demo so that the turtle could be controlled by commands such as 'Colour', 'Frame' and 'IR'. The following Context Free Grammar was thus defined:

$$
\begin{array}{lcll}
\text{CFG } \textit{Prot1} & = & \{N,\ T,\ P,\ S\} & \\
\text{where} & & & \\
N & = & \{\textit{Infrared},\ \textit{Frame},\ \textit{Colour}\} & \\
T & = & \{\text{ all alphabetical characters}\} & \\
S & = & \textit{Command} & \\
P & = & \{\text{see } \textit{Command}\} &
\end{array}
$$

$$
\begin{array}{lcl}
\textit{Command} & \rightarrow & \texttt{depth} \\
 & | & \textit{Colour} \\
 & | & \textit{Infrared} \\
 & | & \textit{Frame} \\
 & & \\
\textit{Infrared} & \rightarrow & \texttt{ir} \\
 & | & \texttt{infrared} \\
 & & \\
\textit{Frame} & \rightarrow & \texttt{skeleton} \\
 & | & \texttt{frame} \\
 & & \\
\textit{Colour} & \rightarrow & \texttt{colour} \\
 & | & \texttt{colours}
\end{array}
$$

It was then necessary to translate these rules to code, please see Appendix A for the changes to the code to reflect the grammar, in reality this allowed the use of the strings defined above to make the turtle move around the screen as it did before:

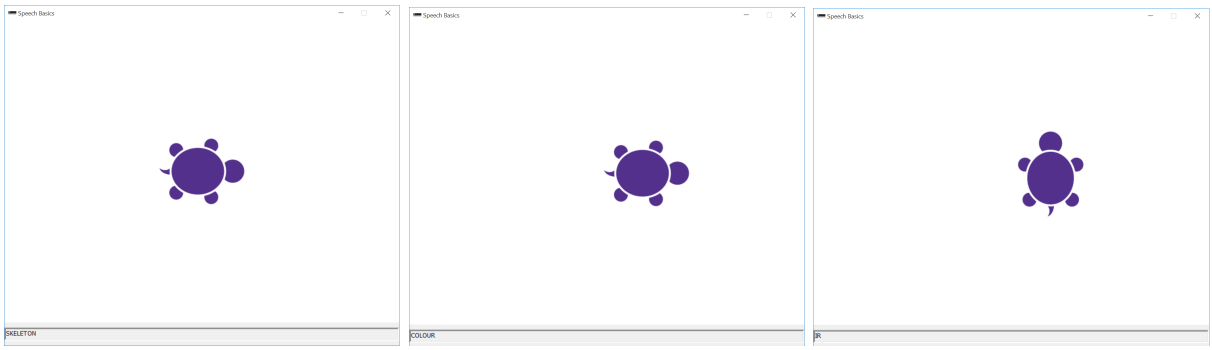

Figure 9.1: Print Screens of controlling the turtle using 'frame' to rotate 90°clockwise, 'colour' to travel forward in whatever direction the turtle faces and 'infrared' (abbreviated to IR in the image) to rotate 90°counter-clockwise

[1] I am using the formal definition of a context free grammar here as defined in [10]

Naturally this prototype is a throw-away but it serves as a good basis and to start defining other grammars on.

9.2 Object Identification

In this prototype, OpenCV was used to process streams of images in order to remove background noise, skin and clean the image of noise so that only the object remained. There were 4 problems to address here: Firstly, methods of back subtraction were explored [11] in order to find the best means by which we could remove *background* elements so that only foreground elements were changed. One suitable solution was 'a single-camera statistical segmentation and tracking algorithm' [12] which employs bayesian inferencing with Kalman filters in order to return a set of pixels defined to be 'foreground' pixels and hence negate the background. This algorithm relies on a sequence of prior frames used in order to observe the change in pixel values and calculate probabilities that something has changed at that location in the frame. Furthermore, the algorithm requires a series of prior (normally 120) frames which is exactly what is available. Lastly, this algorithm has been implemented in OpenCV [13] and is *very* efficient [12]:

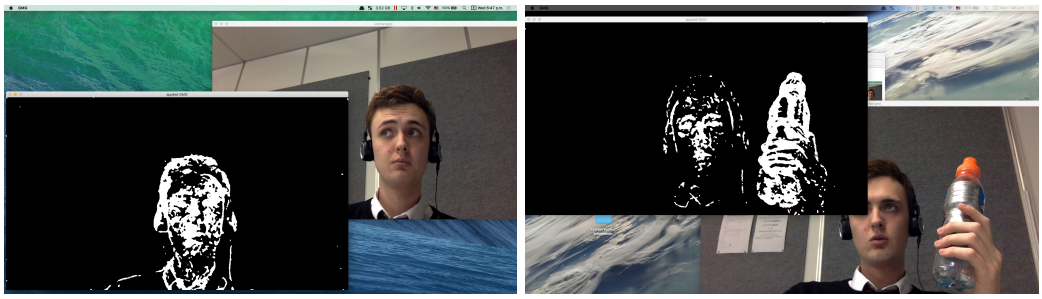

Figure 9.2: GMG in action, code supplied in Appendix B

9.3 Image processing for Object Tracking

Code for this prototype is underway but currently incomplete. Attempts are being made here to develop a simple 2d model of how the points would be tracked as they are recorded. It is exploring ways in which the direction can be determined.

Appendices

A Speech Recognition Prototype 1

This appendix includes the xml code modified to reflect the CFG *Prot1* defined in 9.1

```
<grammar version="1.0" xml:lang="en-US"
         root="rootRule"
         tag-format="semantics/1.0-literals"
         xmlns="http://www.w3.org/2001/06/grammar">
  <rule id="rootRule">
    <one-of>
      <item>
        <tag>COLOUR</tag>
        <one-of>
          <item> colour </item>
          <item> colours </item>
        </one-of>
      </item>
      <item>
        <tag>DEPTH</tag>
        <one-of>
          <item> depth </item>
        </one-of>
      </item>
      <item>
        <tag>IR</tag>
        <one-of>
          <item> infrared </item>
          <item> IR </item>
        </one-of>
      </item>
      <item>
        <tag>SKELETON</tag>
        <one-of>
          <item> skeleton </item>
          <item> frame </item>
        </one-of>
      </item>
    </one-of>
  </rule>
</grammar>
```

B GMG code using OpenCV, Object Identification Prototype 1

This section contains the code for the first prototype developed in order to test the GMG algorithm [REFERENCE] mentioned in 9.2

```
#include <stdio.h>
#include <iostream>
#include <opencv2/opencv.hpp>
#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/video/background_segm.hpp>

using namespace cv;
using namespace std;

int main(){

  VideoCapture cap(0); // connect to webcam and open
  if(!cap.isOpened()){
    printf("Unable to connect to camera, quitting\n");
    return -1;
  }

  Mat frame; //current frame
  Mat fgMaskGMG;
  Mat element = getStructuringElement(MORPH_RECT, Size(3, 3), Point(1,1) );

  Ptr<BackgroundSubtractorGMG> pGMG; //MOG2 Background subtractor
  pGMG = new BackgroundSubtractorGMG();

  //repeat until the program is closed
  for(;;){

    //capture and pass frame into Mat object
    cap >> frame;

    //apply the CMG algorithm to the current frame
    pGMG->operator()(frame, fgMaskGMG);
    morphologyEx(fgMaskGMG, fgMaskGMG, CV_MOP_OPEN, element);

    imshow("unchanged", frame);
    imshow("applied GMG", fgMaskGMG);

    imshow("webcam", frame);

    if(waitKey(30) >= 0)
      break;
  }
}
```

10 Bibliography

References

[1] P. Belhumeur and H. G.D., "Tracking in 3d: Image variability decomposition for recovering object pose and illumination," in *Pattern Analysis and Applications*, ch. 2, pp. 82–89, Springer-Verlag London Limited, 1999.

[2] R. S.S and A. A, "Manipulating objects through hand gesture recognition in virtual environment," in *Advances in Parallel Distributed Computing*, vol. 203, pp. 270–281, Springer Ltd, 2011.

[3] "Oculus rift." Website: `https://www.oculus.com/en-us/gear-vr/`.

[4] Spencer, "Sony patents a motion control system that uses ordinary objects as controllers." Article on website `http://www.siliconera.com/2009/07/02/sony-patents-a-motion-control-system-that-uses-ordinary-objects-as-controllers/`, 2009.

[5] Han, H. Seung Ju (Yongin-si, KR), B. Jae Joon (Yongin-si, KR), K. Won Chul (Yongin-si, KR), K. Do Kyoon (Yongin-si, KR), and K. Sang Kyun (Yongin-si, "Virtual world processing device and method." US Patent, available at: `http://patft.uspto.gov/netacgi/nph-Parser?Sect1=PTO2&Sect2=HITOFF&u=%2Fnetahtml%2FPTO%2Fsearch-adv.htm&r=10&f=G&l=50&d=PTXT&p=1&S1=((((((sony+AND+controller)+AND+object)+AND+playstation)+AND+recognition)+AND+motion)+AND+depth)&OS=sony+AND+controller+AND+object+AND+playstation+AND+recognition+AND+motion+AND+depth&RS=((((((sony+AND+controller)+AND+object)+AND+playstation)+AND+recognition)+AND+motion)+AND+depth)`, August 2015.

[6] "Project natal/kinect." Website `https://en.wikipedia.org/wiki/Kinect`, November 2015.

[7] C. Bishop, *Pattern Recognition and Machine Learning, Bishop*, p. 3. Springer Ltd, 2007.

[8] website = `https://msdn.microsoft.com/en-us/library/dn782040.aspx`.

[9] "Speech basics-d2d c++ sample." website `https://msdn.microsoft.com/en-us/library/hh973079.aspx`.

[10] U. J. Hopcroft John, *Introduction to Automata Theory, Languages, and Computation*, pp. 169–173. Addison-Wesley, 1979.

[11] "How to use background subtraction methods." website = `http://docs.opencv.org/master/d1/dc5/tutorial_background_subtraction.html#gsc.tab=0`.

[12] K. G. Andrew B. Godbehere, Akihiro Matsukawa, "Visual tracking of human visitors under variable-lighting conditions for a responsive audio art installation," June 2012.

[13] "Gmg algorithm implemented in c++, open cv." website = `http://docs.opencv.org/3.0-beta/modules/video/doc/motion_analysis_and_object_tracking.html#createbackgroundsubtractorgmg`.