# Keyframe Tagging: Unambiguous Content Delivery for Augmented Reality Environments

Adam James Clarkson

Ph.D. Thesis

School of Engineering and Computing Sciences

Durham University

November 2015

# Abstract

**Context:** When considering the use of Augmented Reality to provide navigation cues in a completely unknown environment, the content must be delivered into the environment with a repeatable level of accuracy such that the navigation cues can be understood and interpreted correctly by the user.

**Aims:** This thesis aims to investigate whether a still image based reconstruction of an Augmented Reality environment can be used to develop a content delivery system that providers a repeatable level of accuracy for content placement. It will also investigate whether manipulation of the properties of a Spatial Marker object is sufficient to reduce object selection ambiguity in an Augmented Reality environment.

**Methods:** A series of experiments were conducted to test the separate aspects of these aims. Participants were required to use the developed Keyframe Tagging tool to introduce virtual navigation markers into an Augmented Reality environment, and also to identify objects within an Augmented Reality environment that was signposted using different Virtual Spatial Markers. This tested the accuracy and repeatability of content placement of the approach, while also testing participants' ability to reliably interpret virtual signposts within an Augmented Reality environment. Finally the Keyframe Tagging tool was tested by an expert user against a pre-existing solution to evaluate the time savings offered by this approach against the overall accuracy of content placement.

**Results:** The average accuracy score for content placement across 20 participants was 64%, categorised as "Good" when compared with an expert benchmark result, while no tags were considered "incorrect" and only 8 from 200 tags were considered to have "Poor" accuracy, supporting the Keyframe Tagging approach. In terms of object identification from virtual cues, some of the predicted cognitive links between virtual marker property and target object did not surface, though participants reliably identified the correct objects across several trials.

**Conclusions:** This thesis has demonstrated that accurate content delivery can be achieved through the use of a still image based reconstruction of an Augmented Reality environment. By using the Keyframe Tagging approach, content can be placed quickly and with a sufficient level of accuracy to demonstrate its utility in the scenarios outlined within this thesis. There are some observable limitations to the approach, which are discussed with the proposals for further work in this area.

# Declaration of Authorship

I, Adam James Clarkson, declare that this thesis titled, 'Keyframe Tagging: Unambiguous Content Delivery for Augmented Reality Environments' and the work presented in it are my own. I confirm that no part of the material provided has previously been submitted by the author for a higher degree in Durham University or any other University. All the work presented here is the sole work of the author.

# Acknowledgements

# Contents

# List of Figures

# List of Tables

# 1

# Introduction

## 1.1 Background

The core concept of any Augmented Reality system is to allow the user to view virtual content as part of the real world, as if it were part of the real world. The realism of this experience is limited only by the content quality, display technology being used, and the stability of the content positioning within the environment. Recent research advances have provided us with highly stable tracking and mapping algorithms for Augmented Reality such that a system is aware of its position within, and relationship with a real world environment. This is an important step in ensuring that any displayed virtual content appears seamlessly and naturally within that environment when observed by the user. Several approaches exist, from novel marketing techniques requiring a user to print off a fiducial marker (a unique pattern similar to a bar code or QR code) to be held up to a webcam, right through to Head Mounted Displays which allow a user to walk around large virtual content items and perceive them in three dimensional space.

Consider a scenario in which an Emergency Response Team (ERT) are searching the site of a natural disaster. This is an inherently dangerous environment, as well as one

which must be treated sensitively. The ERT have a responsibility to ensure the safety of themselves and colleagues, while responding quickly to ensure the safety of victims of the disaster. In a situation such as this, information and communication are key. Existing technologies provide a foundation means for the ERT to talk to one another, warn one another of dangers and areas already searched. In the chaos of such an environment there is a high reliance on either the memory of ERT members, or their ability to quickly record such information in order for it to be of use. The application of Augmented Reality within a scenario such as this, could be explained as an extension of the senses.

Advanced tracking and mapping algorithms, developed from work within the robotics community mean that it is possible for a computer system to evaluate an environment from a single camera, building a three dimensional representation of it in memory which can be used as a knowledge base. The system is able to recognise when looking over a known area, and can locate the users position within the environment. By pairing this technology with a display device, such a system can be used to input information directly into the users' view, as with a Heads Up Display. In the ERT scenario, this removes the reliance on team members memory to retain information, it also removes the need for examining menu driven devices on a more traditional display technology. Colleagues positions, locations of victims, locations of hazards, and locations which still require attention are all candidates to be instantly highlighted within the users' Heads Up Display.

The tracking and mapping technology exists to begin working towards these systems, however a research area critical to this idea requires further exploration. Reliable content delivery methods for Augmented Reality, are often tied to systems which require the introduction of some known element into the environment. By placing a fiducial marker into the cameras view a system can assign content to it, similarly a virtual model of a table can be used to anchor content every time it comes into view. Some Augmented Reality systems do provide accurate content delivery without these features, though they often require you to manipulate the content while exploring the environment in real time. In the ERT scenario explained above however, these options are not feasible. The need to introduce markers into an environment, or to provide a known computerised model

involve overheads which would only slow down the ERT work rather than assisting it. Similarly the need for the ERT to manipulate content positioning while on the ground adds unnecessary overheads to people already in a dangerous situation.

By remotely adding content into such an environment, a control centre can provide the ERT on the ground with information to "extend their senses" with no requirement for additional work from anyone within a dangerous situation. By providing an image stream of the environment back to a control centre as the team explore it, the environment can be analysed and augmented with content from a remote location.

## 1.2 Research Objectives

The aim of this thesis is to present a novel means of delivering content into an Augmented Reality environment with a repeatable level of accuracy such that a user would be able to identify the content in relation to it's surroundings with no ambiguity. The thesis investigates the possibility of providing a static image based reconstruction of an Augmented Reality environment as a base for users to introduce content. This requires the provision of a means of image tagging which translates into real world three dimensional co-ordinates such that the delivered content can be displayed in a live environment within an Augmented Reality system. By providing a static image tagging basis for introducing content in this manner, the user will be able to achieve accurate object placement with more speed and ease than by manually manipulating the data within a live environment. It is expected that the approach proposed in this thesis will lead to a repeatable level of acceptable accuracy in content placement, with an improvement in speed over current methods.

### 1.2.1 Research Contributions

This thesis provides the following research contributions:

- Proposal of a novel means for delivering content into an Augmented Reality environment via a recreation of key features of that environment.

- Implementation of the proposed approach to allow for the evaluation of the method.

- Quantitative data on the accuracy of object placement achievable by the proposed approach.

- Quantitative data on the interpretation of object placement achieved by the proposed approach in a live Augmented Reality environment.

### 1.2.2   Criteria For Success

Successful completion of the investigation detailed within this thesis will be judged on the provision of answers to the 5 Research Questions (RQ) listed below in table 1.1.

**Table 1.1:** Research Questions

| RQ | Research Question |
| --- | --- |
| RQ1 | Can a user place content into an Augmented Reality environment using a photograph based reconstruction of that environment? |
| RQ2 | Is content placed using a photograph based reconstruction of the environment positioned with an acceptable level of repeatable accuracy? |
| RQ3 | Can the proposed content delivery method be used under time pressures to place content while maintaining an acceptable level of repeatable accuracy? |
| RQ4 | Can users reliably identify physical objects in an Augmented Reality environment which are highlighted by a virtual Spatial Marker? |
| RQ5 | Does the size of a Spatial Marker object relative to the physical object it is highlighting have an impact on the number of correct identifications given by users? |

## 1.3   Thesis Outline

The remainder of this thesis will follow the structure listed below in Table 1.2, which offers a brief overview of the contents of each chapter:

**Table 1.2:** Thesis Outline - Chapter Overview

| Chapter | Overview |
| --- | --- |
| 2 | This chapter discusses the challenges presented by Augmented Reality systems as a wider discipline and also looks more specifically at the challenges specifically relating to content delivery, a problem common to collaborative Augmented Reality systems. |
| 3 | This chapter discusses the proposed novel approach of Keyframe Tagging (KFT) for delivering content into an Augmented Reality environment via a static recreation of that environment. |
| 4 | This chapter discusses the implementation of the proposed KFT system as performed for this experiment, along with the reasoning for the choice of the existing tracking and mapping system used throughout the remainder of the thesis. |
| 5 | This chapter discusses the user trial conducted to assess the accuracy of placement possible when delivering content into an Augmented Reality environment using the KFT system. |
| 6 | This chapter discusses a second evaluation in which an expert user conducts a similar trial in both KFT and the existing system under two different time conditions to assess the impact on placement accuracy. |
| 7 | This chapter presents the results of an investigation to answer the question of selection ambiguity in Augmented Reality environments. Users were asked to explore a live environment which had target objects identified by virtual content. The users provided their interpretation of which objects were identified. |
| 8 | This chapter reviews the results gained from the three experiments carried out to evaluate this thesis. It draws on the scenarios from Chapter 3 in order to assess the viability of the proposed KFT approach. |
| 9 | The final chapter in the thesis presents the conclusions drawn from the development and experimental work and suggests further work which could be undertaken based on these results. |

# 2

# Augmented Reality Literature Survey

## 2.1    Introduction

Since Sutherland (1965) first introduced his work on "The Ultimate Display" (later known as "The Sword of Damocles") the field of Augmented Reality has seen many varied research projects across a wide range of fields. Both at an industrial and commercial level all fields of Computer Science have been pervaded by this research topic in some way. Indeed the rise of affordable and powerful mobile computing has provided an ideal platform for exposing every day users to the Augmented Reality paradigm, a fact noted by Duh and Billinghurst (2008) when studying the evolving trends in Augmented Reality research since the first conference was held in 1998.

### 2.1.1    Defining Augmented Reality

At its most basic, an Augmented Reality (AR) system is any system which combines virtual data with real data (Milgram and Colquhoun (1999)). While this captures the idea of what it is to "augment reality" it does not offer precision as a definition. Azuma (1997) offers a more specific definition of an AR system's need to meet three criteria;

"1) combines real and virtual, 2) is interactive in real time, and 3) is registered in three dimensions". This definition shows that the aim of an AR system is to provide additional information to the user in a real world scenario. This can take on many forms, from virtual objects to information labels or even some form of navigation aid, the element to focus upon is that the data is added to the real world, as opposed to emulating that which is real as would happen in a Virtual Reality Environment (VR).
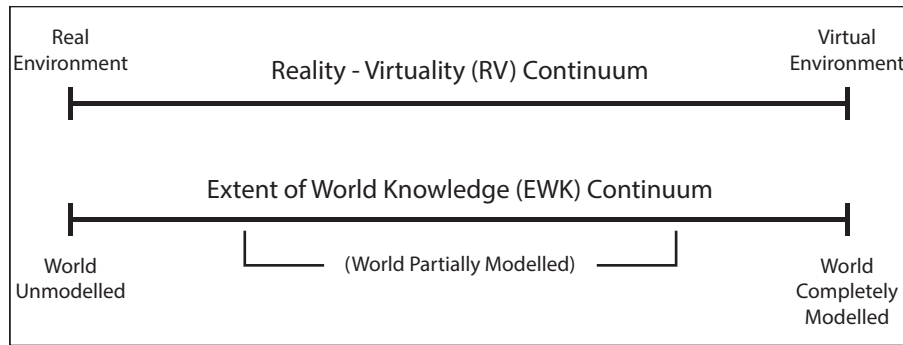


**Figure 2.1:** Reality-Virtuality Continuum alongside Extend of World Knowledge Continuum (Milgram and Colquhoun (1999))

In order to understand the difference between AR and VR environments, Milgram and Colquhoun (1999) devised the Reality-Virtuality Continuum shown in Figure 2.1. The scale between the real environment and the virtual environment is occupied by varying levels of Augmentation. In cases where virtual data is added to the real environment, the world can be observed as being "un-modelled" in terms of the Extent of World Knowledge Continuum, and so the system is termed as Augmented Reality. This is opposite to a system which adds real world data to a virtual environment, which would be known as Augmented Virtuality. Indeed, the power of Augmented Reality systems lies in the fact that they must be able to operate with more "unknowns" with regards to the environment within which they exist. The system will use cues from the environment to inform its function, rather than having complete knowledge of the environment. Feiner et al. (1993) identify the power of AR systems by identifying the scope of AR for aiding cognition during complex tasks. That is to say, they provide extra information which

would not usually be available to the user in the real world environment, and as such have powerful knowledge based potential.

Therefore in order to complete the definition of Augmented Reality systems offered by Azuma (1997), it is useful to add that AR systems will not exist in situations where the world is either completely modelled or un-modelled, such that it forms a midpoint between fully real or fully virtual environments.

## 2.2   Tracking and Mapping Environments

In order for a system to operate within the definition offered in Section 2.1.1, the system must hold the ability to inject virtual data where it is required in a real world environment. In order to do so, the system must hold some knowledge of the environment. Tracking an environment depends on the system being able to constantly detect a feature or features within an environment. This can take the form of a known object, or a physical feature within the environment, but in either case it provides a point of reference from which the system can build environmental knowledge. Mapping is the process of recording the knowledge which a system has about an environment so that it can be re-used. The level at which this is performed in Augmented Reality systems is largely dependent on whether the system uses a marker based (little mapping) or markerless (requires mapping) approach. Each of these approaches are discussed in further detail in this section. Duh and Billinghurst (2008) found AR research to be dominated by papers on Augmented Reality Tracking topics, and even though the real world applications of Augmented Reality are being explored as a growing research area, a lot of effort is still expended exploring the enabling technologies such as environment tracking.

Two approaches exist to the tracking of environments for Augmented Reality, marker based and markerless (or marker free) tracking. Marker based tracking relies on the use of fiducial[1] markers being introduced into an environment, which the system has knowledge of. In systems where this is the case, these "known" markers become anchor

---

[1]A fiducial is a point of reference within an image (or a stream of images in the case of an AR environment)

points for placing virtual data into the real world. The alternative approach is markerless tracking, which removes the "known" anchor point in the environment and relies on the ability of the system to identify pre-existing features within the environment in order to place virtual data.

## 2.2.1   Marker Based Tracking

The use of fiducial markers provides a robust means of tracking an environment with a relatively low computational cost. The popular ARToolkit library (Kato and Billinghurst (1999)) presents a system capable of estimating the camera pose within an environment in real time by identifying an artificial marker such as that shown in Figure 2.2. As with the work done by Kutulakos and Vallino (1998), the camera system does not require any calibration to be carried out before it can be used, and therefore markers can be tracked immediately. Due to the fact that the system only needs to be aware of the position of such a marker, the rest of environment is ignored and therefore saves processing power. This is possible as the marker is a three dimensional object observed within the three dimensional world, and therefore tracking the movement and orientation of such a marker is possible (assuming the marker is asymmetrical).



**Figure 2.2:** An Example Fiducial Marker from the ARToolkit Project (Kato and Billinghurst (1999)) showing subtle asymmetry

An AR system set up to track such fiducials must simply be made aware of the unique pattern within the black square and then an association of content to pattern can be

made. The advantage of such a system is that the marker patterns are easily produced, and therefore the system has some degree of scalability. Regenbrecht, Wagner and Baratoff (2002) created the MagicMeeting system using marker tracking such as this, identifying that the use of a tangible physical object within the environment aids natural interaction with the content, as it means the virtual data has a real world physical anchor which the user is more familiar with as a concept. Figure 2.3 shows how the virtual content is typically positioned upon a marker once it has been identified.



**Figure 2.3:** Example content attached to a marker in ARToolkit (Kato and Billinghurst (1999))

A draw-back of marker based tracking is that it requires a certain level of control over the environment, in that the markers have to be introduced to the environment or the system is useless. Using simple printable fiducials like this lessens the impact of this as such markers are easily "user producible" and can be printed on any home desktop printer and tracked by a webcam. This is increasingly being identified by marketing and PR companies as a novel means of advertising, however in less novel scenarios Regenbrecht, Wagner and Baratoff (2002) and Gillet et al. (2004) identify the need for the marker to be mounted onto a solid surface in order to avoid distortion of the pattern and affect the tracking ability of the system. When in a properly controlled environment impressive and immersive applications can be developed using these techniques however,

Billinghurst et al. (2000) shows an example of a multi-user table top game which uses fiducial markers as a base. Similarly, Henrysson, Billinghurst and Ollila (2005) demonstrates a face-to-face collaborative game platform with two users using a mobile device augmented with a fiducial marker to allow it to be tracked by the other player.

### 2.2.1.1   Problems with Marker Based Tracking

Several drawbacks exist with fiducial marker tracking. Introducing markers to the environment can be considered intrusive in some situations, which Park and Park (2005) attempted to address through the use of "Invisible" Infrared markers, trackable by an Infrared camera. While solving the issue of visually adding to an environment, other problems are introduced such as the lack of interaction with the marker and the possibility for people or objects unwittingly occluding the marker.

Marker occlusion is a large problem for marker based tracking systems. Because a typical fiducial marker such as that shown in Figure 2.2 relies on the identification of the black border to signal to the tracking system that it has found a marker, there is scope for occlusion issues. While Regenbrecht, Wagner and Baratoff (2002) identifies the advantage of being able to interact with a tangible marker as if the virtual content is real (in terms of rotation and translation) if the user inadvertently occludes the border the tracking will be lost, and the virtual data will disappear. This causes consistency and stability issues with such a system, and places restrictions on its application potential.

Environmental features such as lighting can also have an impact. Fiducial markers must be identified repeatedly by the tracking system, and a drastic change in the light level, or angle of reflection of light from a fiducial can cause tracking unreliability. Madritsch and Gervautz (1996) addressed this by using LED beacons to track as an alternative to a printed marker, with the camera tracking system filtering all except the red light using RGB thresholding to provide reliable tracking. By using several LEDs on each tracked "marker" the 6 degrees of freedom necessary to track movement and rotation in three dimensional space is possible. Dorfmüller (1999) offered an alternative approach using retroreflective markers illuminated by infrared light sources attached

to the camera. This offers the desirable natural interaction with the the marker (Regenbrecht, Wagner and Baratoff (2002)) missing from the solution of Park and Park (2005), while capitalising on the thresholding and filtering such as that found in Madritsch and Gervautz (1996).

## 2.2.2   Markerless Tracking

In light of the problems of marker based tracking, markerless tracking aims to remove the issues of fiducial markers. A markerless tracking system is one which aims to track its surroundings through the identification of existing features within the environment, rather than through features which have been artificially introduced. Park, You and Neumann (1998) is identified as one of the earliest implementations of such a technique, whereby the system calculates camera pose from artificial features (as with fiducial tracking) but then continuously updates this camera pose by evaluating natural features in the environment. This allowed the system to maintain a high level of tracking accuracy even when the original fiducials were not in view. However, this still required pre-preparation of the environment with the introduction of the initial artificial features, and as such could not be classed completely markerless.

Much of the research expanding into this area has a grounding within the robotics community, with common problems from this discipline being redefined in Augmented Reality as Computer Vision problems. Simultaneous Localisation and Mapping (SLAM) (Csorba (1997)) is one such concept that is key to using already existing tracking methods in Augmented Reality. Systems which rely on visual identification must be capable of identifying features which exist naturally within the environment. Comport, Marchand and Chaumette (2003) introduced a system for tracking without markers by identifying edges of surfaces within the viewport. Bay, Tuytelaars and Gool (2006) takes a similar approach by introducing "Speeded Up Robust Features (SURF)" as a means of applying image processing techniques to a video feed in order to evaluate the presence and shape of objects within a scene, as opposed to the edge detection presented by Comport, Marchand and Chaumette (2003).

Comport, Marchand and Chaumette (2003) also included model-based tracking showing how a CAD model of a target object can be used to inform a tracker of the features to look for within an environment. Wuest, Vial and Stricker (2005) expanded the CAD model tracking in order to add real time adaptation of the model to improve the robustness of the overall approach. Model-based approaches such as these still require a level of pre-preparation as is the case with the marker tracking approaches, but it is not pre-preparation of the environment, which affords a degree of flexibility more than marker based approaches. This moves towards overcoming the barriers for outdoor tracking in unprepared environments set out by Azuma (1999).

Some of the reliance on pre-prepared world knowledge is further reduced through the use of supporting sensors for the tracking system. You, Neumann and Azuma (1999) presented an early example of using inertial tracking alongside a vision system to improve orientation tracking in outdoor AR systems. More advanced approaches have also included; Combining a CAD model with gyroscope (Klein and Drummond (2003)), head tracking and gyroscope (Satoh, Uchiyama and Yamamoto (2004)), and an implementation of the SLAM problem with CAD intitialisation (Bleset, Wuest and Stricker (2006)) reduce the amount of estimation conducted by the tracking system in order maintain an accurate camera pose. Though these approaches provide valuable supporting data to the vision system, Baillot et al. (2006) identifies the fact that tracker alignment problems are "exacerbated when multiple tracking systems are used simultaneously". By providing a framework to simultaneously ground and update all sensors in one shared "world-to-base" co-ordinate system, Baillot et al. (2006) attempts to overcome this, however the addition of multiple sensors remains a high computational cost with this approach. Even projects using modern mobile phone technologies which afford the developer multiple sensors at a relatively low cost struggle to provide reliable results. Blum, Greencorn and Cooperstock (2013) found that margins of error with both the compass (10-30 degrees) and GPS location (10-30m) on modern smartphones to be too high to tolerate for reliability in general scenarios.

Several projects have focused on solving the SLAM problem without reliance on

further sensors for AR systems. Davison and Murray (2002) used a stereoscopic vision system in order to solve the SLAM problem without reliance on other sensors, and later by removing the reliance on a dual camera setup with monoSLAM (Davison, Mayol and Murray (2003), Davison et al. (2007)). Eade and Drummond (2006) also focuses on removing the dependency on multiple sensor tracking, but by implementing the FastSLAM algorithm (Montemerlo et al. (2002)) rather than the approach taken by Davison, argues that the resultant system is much more easily scalable and computationally less expensive. This makes the work of Eade and Drummond (2006) an appealing platform for further expansion.

Further expanding on the adaptation of SLAM to a Computer Vision problem, Klein and Murray (2007) builds on the monocular approach of Davison, Mayol and Murray (2003) and Eade and Drummond (2006), with the focus on removing model-based initialisation requirements. In doing so the developed Parallel Tracking and Mapping (PTAM) (Klein and Murray (2007)) moves closer to fulfilling the requirement of Azuma (1999) for AR tracking to work in completely unprepared environments. This approach is discussed in depth in Section 2.4.

With competent solutions offered to the Computer Vision SLAM problem, the focus of recent tracking research has shifted to targeting individual problems within tracking environments. As most tracking algorithms rely on edge detection and similar feature based identifications, certain properties of an environment can cause issues. Crivellaro et al. (2014) presents a demonstration of using multiple low-pass image filters to combat the issues introduced by shiny materials in a scene. This provides a more robust tracking on other stronger objects in the scene by reducing the distraction of less ideal surfaces. Other projects such as Carozza et al. (2014) take a similar approach to Klein and Murray (2007) by using a monocular camera based approach to SLAM, however in addition to creating a map of the environment, Carozza et al. (2014) also focuses on creating a 3D object representation of an environment, with buildings represented as models which can have textures applied to them. This is an advanced technique for environment recreation, which provides a good example of the work which can be done building on top of the

strong foundation knowledge of markerless SLAM solutions.

### 2.2.3    Problems with Markerless Tracking

While markerless tracking offers a high level of flexibility in its target operating environments compared to the strictly controlled nature of marker-based tracking, there are several limitations of the systems which use this tracking technology.  Problems such as occlusion and environmental factors which are common in marker-based tracking are still present within these systems, though they are different in nature.  Markerless tracking algorithms rely heavily on feature recognition, and as such environmental properties such as the lighting conditions can have a large impact on the reliability of the tracking algorithm.  Both advanced SLAM (Davison, Mayol and Murray (2003), Eade and Drummond (2006)) and PTAM (Klein and Murray (2007)) treat markerless tracking as a computer vision problem, and as such are image processing tasks at their core.  If an environment is initially explored and mapped on a brightly lit afternoon, the feature point detection will typically be higher than exploring the same environment just before dusk, or on an overcast day for example.  The systems have no way of determining the impact of shadows (or the lack of shadows) when trying to evaluate against a map they previously created.  This can have implications both if the environmental conditions change during operation, or if the created map is being reused by the system at a later date.  PTAM (Klein and Murray (2007)) shows that it is possible to constantly re-evaluate the map however, the feature point set increases in size constantly which adds strain on the tracking algorithm in situations such as this.

Occlusion can also cause issues with markerless tracking, though the impact can be reduced considerably compared to the impact occlusion has on marker-based systems. Whereby in a marker-based tracker the occlusion of a marker results in the loss of tracking, markerless trackers have redundancy built in by the nature of the system.  By tracking multiple feature points at any given time in order to provide camera localisation within the environment map, these systems can tolerate the loss of some of these feature points through occlusion.  The number of losses that can be tolerated varies from sys-

tem to system however, and occlusion must still be considered a challenge for markerless Augmented Reality.

## 2.3 Content Delivery in Augmented Reality

When considering the virtual data provided by an Augmented Reality system a wide range of options are available to the developer, depending on the application context. At its most basic, a simple mapping exists between a fiducial marker and some virtual data. At the other end of the scale dynamically created content can be placed at will into an environment that has not been augmented with fiducial markers, allowing a greater flexibility and seamless integration with the environment.

Krevelen and Poelman (2010) states that "the commercial success of AR systems will depend heavily on the available types of content", also identifying the fact that it is the presentation of commercial content to a common user which needs solving. This is a notion supported by Wu et al. (2013) with regard to learning, who state that a lack of widespread authoring tools for AR lead to content based problems "the content and the teaching sequence are fixed; teachers are not able to make changes to accommodate students"'. By facilitating the deployment of a variety of content into AR environments, more disciplines become available to AR system developers. Billinghurst and Kato (2002) identifies the fact that AR systems align well with social protocols in terms of collaboration, which allows for a wide range of applications mirroring familiar real world situations to be produced, with a low level of training needed for users to familiarise themselves with the systems.

In order to explore the varying kinds of content mapping present in existing Augmented Reality systems, a taxonomy of Content Mapping was developed, collecting the current approaches into the following categories:

- Fixed Content Mapping
- Updatable Content Mapping
- Context Dependent Content Mapping

- Remotely Controlled Content Mapping

### 2.3.1   Fixed Content Mapping

Content Mapping which happens on an application level can be considered as "Fixed Content Mapping" with regards to Augmented Reality systems. The relationship of content to its environment is based upon a mapping defined within the system, such that no other external knowledge or services are required. This means an application update of some description is required in order to alter the relationship of content to environment. This is a particularly common approach with applications based on the ARToolkit project (Kato and Billinghurst (1999)). The MagicBook (Billinghurst (2001)) provides an example of such a system, whereby the relationship between content and application code is fixed. The aim of MagicBook was to produce a physically printed book, with fiducial markers on the pages, such that when the book was viewed through a Head Mounted Display (HMD), virtual content would appear on the pages. The book paradigm which is in use here means that fixed content mapping is a suitable choice. Once a traditional printed book has been printed, the content is not updated without a new revision being released. This is precisely how fixed content mapping in Augmented Reality systems functions.

The technique is not limited to viewing a marker printed onto the pages of a book, however. Complex systems have been constructed which demonstrate the power of Augmented Reality using fixed content mapping. Schmalstieg and Wagner (2005) provides an excellent example of a content-rich application which operates purely on a fixed content mapping. The mobile application acts as a museum tour guide which displays additional information on exhibits on recognising set fiducial markers. This is a case where a sophisticated application is not restricted by the nature of fixed content mapping, as the only time the content would require updating is when exhibits change. There is a limited amount of interactivity between the user and the content as by the nature of a traditional museum exhibit, the relationship is one way, with the user consuming the content.

However, several AR projects which focus upon Human Computer Interaction (HCI) have provided advanced interfaces between the user and virtual content. This is a concept explored in the early work of Ishii and Ullmer (1997) with Tangible Bits, in which an interface system is proposed mapping a physical object to the control of virtual content. The concept is built upon and further explored in Tangible Objects Virtual TableTop (Kato et al. (2000)), and Ting Ting (Kim, Jang and Kim (2004)) with the latter implementing a gaming system built around this interface concept. A particularly immersive example of such interaction can be found in Wagner (2005), an interactive railroad in which users can control a virtual train running on a physical track. The user can build a track with physical objects and then view and control the progress of a virtual train around it with a handheld mobile device. While this provides a high level of visual feedback between the virtual and physical objects, advanced interfaces can also be used as an aid for learning. Matsutomo et al. (2012) introduce a complex interface system which allows for the visualisation of magnetic fields. Allowing users to manipulate the physical components of an electromagnet while viewing a visualisation of the resultant magnetic field proved to be a successful learning tool in this system. Even though the visualisation of the field appears dynamic, from a purely content mapping standpoint, the content is fixed internally to the application. It is the result of a formula which updates the visualisation as opposed to the content being manipulated and changed by a user.

All of these projects have a common theme in that the content delivery is purely internal though. Despite the complexity of the interface paradigms being used, the content shown to the user is intrinsically tied to its environment via a mapping defined in the application.

## 2.3.2   Updatable Content Mapping

Building on top of "Fixed Content Mapping", many AR systems offer more flexibility to users by retaining an application-level relationship between content and the environment (such as marker based tracking), but allowing the user to update the content which is used within that mapping. The MagicMeeting (Regenbrecht, Wagner and Baratoff

(2002)) application provides a collaborative meeting environment, in which a number of tracked markers are placed upon a meeting table. Through the use of HMDs the participants can see virtual content attached to these markers as part of the meeting. By allowing the participants to replace the content currently attached to a marker with an object or document from their PDA, the system allows the manipulation of the content and environment relationship. This is a paradigm employed by several other systems, with the ARTHUR (Broll et al. (2004)) system employing a similar method to review Architectural Models. While this allows a degree of flexibility and dynamic content management to an environment, it does so at the cost of requiring a highly controlled environment. The amount of control required in the environment can be reduced somewhat depending on the aims of the software. Lin (2012) presents a system which shares the idea of attaching updatable content to some marker as in Broll et al. (2004), however the markers are printed on a postcard with poetry on it which can be sent to a recipient to view. The resultant content which is shown to the recipient can be configured to display multimedia video relating to the poem. In this instance, it is not necessary to control an entire environment, simply to ensure that the user has the required equipment (webcam and a computer, and access to the viewing software) to view the content.

Other systems, such as the virtual cockpit simulator developed by (Poupyrev et al. (2002)) explore this concept further, by allowing a series of fiducial markers to be mapped to the components of an aircraft cockpit, which can be combined to make a simulator. The content attached to the markers can be swapped in and out, but despite its advanced functionality, the relationship of the content to the environment is still founded in the presence of fiducial markers. While all of these systems, and many others based on ARToolkit (Kato and Billinghurst (1999)) and similar frameworks, provide excellent examples of how user-updatable content can be brought into Augmented Reality environments, there is a fundamental limitation in that the content must be assigned to a fiducial marker. A simple mapping of content to real-world object is much more difficult, if there is no point of reference for the real-world object programmed into the system.

Lee et al. (2004) addressed this issue by providing exactly the same content authoring and execution environments for an AR system. The result is to allow the user to dynamically update the content which is viewed within in the live AR environment, therefore removing the static link between content item and environmental position as the user is allowed to create new content directly within the environment. Piekarski (2006) also explores the idea of content generation within an Augmented Reality environment, expanding the Tinmith system (Piekarski and Thomas (2001)) to support 3D modelling. Here, the user can generate any content they are able to using the provided 3D modelling tools, and anchor it within the environment.

The PTAM application (Klein and Murray (2007)) also provides User Updatable Content Mapping but in a markerless environment. By utilising markerless tracking the content is mapped to a co-ordinate system as opposed to a known marker, which allows more flexibility. The limitation of such a method is that the user must place the content while in the environment in real time, to ensure that the placement is correct. This is performed using a traditional mouse based interface, overlaid on their HMD display. While such an approach can seem cumbersome in operation compared to more sophisticated gesture based tracking, the user is instantly familiar with the metaphor of the interface, and the chance of false positive gesture detections is removed.

### 2.3.2.1   Context Dependent Content Mapping

Content flexibility can be achieved through the use of Context Dependent Content Mapping. By building on the principles which underpin systems with User Updatable Content Mapping it is possible to update the content, not with user selected content, but with contextually aware content. An early example of this using model-based awareness is found in Feiner et al. (1997), using models of the buildings on a University campus to detect a user's location and display relevant information. This is built upon in The GUIDE project (Cheverst et al. (2000)), providing a generic framework for contextual content delivery. However in both this, and the case of Feiner et al. (1997), while the content is highly dynamic there is no provision for changing the underlying content

model without customising the application.

In addition to model-based awareness, other sensors such as GPS can be used for contextually delivered content. A clear example of this can be found in the ARQuake Project (Piekarski and Thomas (2002)), in which contextual location aware information is used in order to inform the positioning of enemies within a game. The location awareness is provided by GPS and maps are used to determine the position of obstacles such as buildings, meaning that the concept is theoretically usable in any environment for which GPS maps exist. Wu et al. (2004) demonstrates another powerful example of this in which PDA users take part in a context sensitive game while exploring an area using GPS. While the utility is restricted to a narrow area (as a network infrastructure is required to disseminate game updates and facilitate player collaboration), the project serves as an example of how powerful the use of context aware sensors can be for delivering content in AR environments. However in both of these cases, there is no high-level location awareness such as that provided by environmental feature mapping or model-based approaches such as those used for markerless tracking (2.2.2).

This is an area in which commercially available Augmented Reality applications are abundant. One of the first and most popular examples is the LayAR (2010) application, which uses GPS location data to overlay tourist information onto buildings of major cities when viewed through a mobile device's camera. The content in this case is entirely dependant on context, and as such there is no requirement to allow the user to update it within the application. As well as in commercial Augmented Reality, Santos et al. (2013) provide an extension of Augmented Reality Learning Experiences, proposed initially by Billinghurst and Duenser (2012). Billinghurst and Duenser (2012) presents several different ways in which AR can be used within a classroom situation, to create learning experiences, allowing users to interact with both digital and non-digital environment content. Santos et al. (2013) took this work, and built the concept of Augmented Reality Learning Objects (ARLO), to model these experiences as a reusable entities that can be used as a component in technology supported learning. Santos et al. (2013) see these objects as having three main components; Context, Content, Instructional

Activity. In terms of the Context Dependent Content Mapping, ARLOs represent an element of content that has an environmental context attached to it. Changing the context of one of these objects (an operation handled by a teacher in a classroom) can cause the behaviour or appearance of the content to be changed as well.

### 2.3.2.2  Remotely Controlled Content Mapping

In order to provide the most flexibility for content delivery within Augmented Reality systems, some systems allow for remote control of the content displayed within the environment. In this context remote refers to something outside of the run-time environment of the AR system. An early framework for a system such as this was proposed by Spohrer (1999) with "Information in Places", proposing a planet-wide means of assigning content to places. This idea is developed further by Kooper and Macintyre (2003) who suggest a generic "browser" application for AR content, which would allow content generated by a multitude of sources to be viewed using one application. While several systems have utilised GPS to do this in a context-aware way, there is scope for a global language allowing remote content delivery into Augmented Reality systems. While this is not yet a reality, some systems are emerging which offer control over the content displayed in a remote environment to a user.

Remotely controlled content can take the form of a system which utilises external information sources in order to generate content. Li, Chuah and Tian (2014) presents a prototype system for a high school campus, which uses Augmented Reality tracking and mapping to query the school news services to chose what content should be displayed to a user. The relevance of the content can then be decided upon based on the user location on campus, or what they are currently looking at. Due to the fact that the content is completely remote from the system, it is entirely controlled by an outside entity.

Remote Mapping is also a common feature of remote collaboration systems which use Augmented Reality, in which either a shared space exists which both remote users can view, or some level of control exists from one user to another. Barakonyi, Fahmy and Schmalstieg (2004) introduces an Augmented Reality video conferencing system, which

by expanding ARToolkit (Kato and Billinghurst (1999)) allows fiducial markers to be tracked within the conversation space. This forms a base upon which remote content can be delivered by sharing marker-to-content relationships between the participants.

Julier et al. (2000) introduces a more advanced content sharing method. The Battlefield Augmented Reality system (BARS) allows a Head Mounted Display for a solider to be augmented with supplementary data, such as wireframes of buildings or locations of colleagues. By using a multitude of sensors the data can be deployed within context in the environment, with the choice of such content being made by a remote user rather than the soldier on the ground.

Höllerer (1999) developed the Mobile Augmented Reality System (MARS) system, an AR collaboration system for indoor and outdoor use. Indoor users could control the content displayed in an outdoor environment, by manipulating a virtual map. The content placements were then shown outdoors. The metaphor developed here has powerful implications for the future development of collaborative systems, but at the time of writing the work was too far ahead of the supporting tracking and mapping technology. Similar traits can be identified in Stafford, Piekarski and Thomas (2006), in which a "Godlike" collaboration metaphor is developed between a team of users indoors and outdoors. The indoor users can gesture, or place objects, within a controlled area, which appears in the sky to the outdoor users. This allows for a shared environment between the participants. This is a powerful collaboration metaphor, and one which shows a unique take on content delivery as physical objects are being recreated from one users space to another.

When considering Remotely Controlled Content Mapping such as these systems, Roesner, Kohno and Molnar (2014) makes an important observation. If the content is not only remotely controlled, but being provided by third party systems (such as social media) then careful consideration must be given to the security of data. Example are given by Roesner, Kohno and Molnar (2014) of systems which use photo recognition to provide information about users via social media profiles. While this could be useful in a number of situations, the opportunity for abusing such systems is clear.

### 2.3.3   Content Authoring

While various means of delivering content into AR environments exist, work has also been carried out in how to author the content. While some research projects which have already been discussed, such as Piekarski (2006), focus on ways to author the content within the live environment, efforts have been made to devise universal means of content authoring for Augmented Reality.

Macintyre et al. (2004) introduced Designers Augmented Reality Toolkit (DART), a framework for authoring content for Augmented Reality systems which was built upon the popular Macromedia Director software. Although this restricts users to one particular piece of software, it allows users familiar with Macromedia Director to quickly author content for AR. This is crucial to improving the state of content delivery for AR, by removing obstacles and learning curves. Similarly Ledermann and Schmalstieg (2005) provides a means of authoring content via Microsoft Powerpoint slides. Although this limits the type of content to information boards, this approach takes an almost universally recognised piece of software and opens AR content authoring to a huge user base. While both of these solutions offer a fast and familiar way to create content, limitations on the kind of content that can be created exist due to been tied to one piece of software.

By proposing an XML database as a means for AR data storage, Schmalstieg et al. (2007) provides an enabling technology for a more generic interface between AR system and content relationships. The database is part of an AR Modelling Pipeline which exists to allows both relationships between the content and scene, and also more traditional relationships between content items within the database. By utilising a database structure, the data format and metaphor is familiar to developers, and the use of XML allows in-depth descriptions of content items. Hill et al. (2010) develops a similar idea focused around the creation of AR content as HTML in the KHARMA project. The focus here is to allow for the conversion of HTML descriptions into content within an Augmented Reality environment. Ahn, Ko and Yoo (2014) extends the principles of HTML5 content creation to mobile AR content. The proposal by Ahn, Ko and Yoo

(2014) is to completely seperate the content from the application logic by allowing content to be authored based on existing web technologies such as the Document Object Model and Uniform Resource Identifiers. This is a highly distributable format which could be common across several Augmented Reality systems, allowing for content authoring on a large scale. Paired with the idea of a generic browser (Kooper and Macintyre (2003)), a powerful means of authoring and distributing content could be developed.

In highly controlled environments where marker based systems such as those in Section 2.2.1 are appropriate, Shim et al. (2014) presents a dynamic content authoring system. Users are able to chose the content to attach to a marker using a configuration system (based on a traditional GUI) before using an advanced gesture recognition system to manipulate the content live in the environment. This has the advantage of allowing the user to view the content which is being created within its target environment, during the creation phase. However it is limited by the fact that markers are required for its operation.

## 2.4   PTAM

Parallel Tracking and Mapping as introduced by Klein and Murray (2007) was designed in order to solve many of the issues arising from markerless tracking systems, in particular where the calibration and initialisation of systems is concerned. The focus was on making an Augmented Reality tracking and mapping algorithm which can operate in a completely unknown environment, as required by Azuma (1999), for Augmented Reality systems to be able to operate in 'completely unprepared environments'. While previous attempts at reducing the 'Extent of World Knowledge' (Milgram and Colquhoun (1999)) for Augmented Reality systems largely relied on the work of robotics and the use of sensors, Klein treats the problem purely in terms of computer vision. While previous advanced approaches in markerless tracking had relied on the Simultaneous Localisation and Mapping techniques (Davison, Mayol and Murray (2003), Eade and Drummond (2006)), PTAM aimed to provide markerless tracking without the need for some ini-

tialisation model, regardless of how small. The key difference in the model of tracking between PTAM and previous SLAM based solutions was the realisation that whereas the current solutions were derived from the robotics community, transferring to hand-held monocular vision systems created a much less smooth video. This introduced tracking problems more suited to algorithms arising from bundle adjustments, such as Structure-from-motion (Engels, Stewenius and Nister (2006)).

## 2.4.1   Tracking and Mapping in PTAM

In order to provide tracking functionality in truly unknown environments, PTAM shifts the focus of 'preparation' to the vision system. In order to allow the use of a monocular tracking system, the camera must be first configured to a set of known parameters. This is achieved by examining a known grid template from multiple angles in order to give the camera the planar knowledge of the relationship of a surface to the camera lens. This is particularly important as to gain the best results from the PTAM system, the user is advised to use a wide angle lens camera. As such, barrel distortion around the edges of the lens is common, which impacts the vision system and must be accounted for. Once the camera calibration system has been completed, the system is able to operate in any further environment without the need for the known grid to be present, with the only stipulation that any changes to the camera (zoom level, interchangeable lens) would require recalibration. This offers a much more flexible approach than the SLAM based systems seen in Csorba (1997), Davison, Mayol and Murray (2003), and Bleset, Wuest and Stricker (2006).

Klein and Murray (2007) states that in examining state of the art solutions to the monocular SLAM problem, a clear motivation arose to separate tracking and mapping processes into two separate threads, and deal with them as two separate processes. This has the benefit of being able to process features in batches as opposed to simply 'as they are detected' which enables 'offline' (non-real time) updates of the map data while tracking is able to continue. An overview of the main initialisation, tracking, and mapping process is offered in the rest of this section.

### 2.4.1.1    Initialisation

Upon starting the PTAM system in a new environment, rather than having to provide a known CAD mesh or other initialisation model as would be common with the SLAM technologies, the user must perform a simple initialisation. Based on the 5 point stereo algorithm (Stewenius, Engels and Nister (2006)), a user must simply press a key before moving laterally and pressing the same key again. The feature points detected in the environment upon the first key press are reevaluated upon the second, and the translation between the two informs the system of the depth data, similar to the algorithm's application in Nister, Naroditsky and Bergen (2005). Figure 2.4 shows the lateral translation between known points during this initialisation procedure, represented as a line between the two points.



**Figure 2.4:** Lateral Translation for Stereo Initialisation in PTAM

This means of initialising an environment is one of the key advantages of the PTAM system, as it enables its use in any environment where the vision system can detect fea-

ture points. In the case of PTAM, feature point detection is done using edge and corner detection algorithms, and as such the "best" calibration (resulting in accurate and stable tracking) is carried out when the scene has multiple reference points of this type. Initialising the map with only larger smooth planar surfaces in the viewport can lead to insufficient translation data, and subsequently a reduced level of reliability when using the system. Furthermore, it is important to consider that Klein states PTAM is designed to be used to track desktops and workspaces, and therefore it is pushing the limits of the software to attempt tracking of whole rooms or even larger environments.

### 2.4.1.2 Feature Point Detection

The density of feature points in the created map is a particular strength of the PTAM mapping approach, as with SLAM based systems there is a constant re-evaluation of the properties of the map and the camera pose within it at an ever increasing granularity. When first observing a new area of the environment a key frame image is created and analysed. A small number of the coarsest features are identified and used for camera pose estimation, before a search is carried out for up to 1000 points to fine tune the pose estimation (Figure 2.5 shows the detected feature points within PTAM).

Following the initialisation process described above, a basic map exists consisting of two image key frames, and feature points detected by an initial run of the corner detection algorithm on these frames. Camera pose information is derived from the initialisation algorithm and movement of the camera which the user performed in that stage. As the user explores an environment, new key frame images are created every time a set of defined criteria are met (time since last key frame created, minimum distance from nearest detected point). As each key frame is added the corner detection algorithm is applied to detect new points, however these points cannot contain depth information from only one frame. PTAM therefore uses a "Patch Search" to look for the feature in the nearest key frames (determined by camera pose) and then triangulates the depth information of the new point using this information.

"Patch Searching" is a key concept of PTAM's tracking algorithm. By looking at

small areas of an image (initially 8x8 pixels) around a detected point, it is possible to quickly search the environment for potential matches. This is not only used to provide depth information for new points, but also to keep tracking existing map information. When using the "Patch Search" to identify an already detected point, the system will look for patches which potentially match a feature point, and then further examine these patches to confirm whether they match. If a feature point is potentially detected within a scene, then the patch around it will be transformed (through a affine warp transformation) to take account of the viewport change which has occurred between the cameras current position and the key frame image in which this feature was detected. Successful patch searches result in a pose estimation calculation, which updates the camera positioning information in the map.

The result of this is that for each new area of the environment discovered, a new camera pose estimation is created based on already known patches and feature points, rather than relying on the tracking of the motion relative to some features in the map or some known initialisation model. This means that should the tracking algorithm become 'lost', there is no need to rediscover the target object from the initialisation model as each key frame provides a known set of feature points upon which the camera pose can be re-estimated.

Not only does this pose estimation provide accurate and reliable tracking as the user moves around the environment, it also reduces the amount of failure conditions where the user loses all virtual data because of a tracking issue. Additionally, the user does not have to manually intervene to correct such a situation, other than simply being prepared with the knowledge that the situation can be rectified by looking at a densely populated area of their environment to allow the system to regain the tracking.

### 2.4.1.3   Map Creation

As the user explores the environment, and the tracking and feature detection elements of PTAM enrich the dataset, an underlying map of data is maintained. This map is formed of a point cloud, and a series of images. The point cloud holds the co-ordinate and
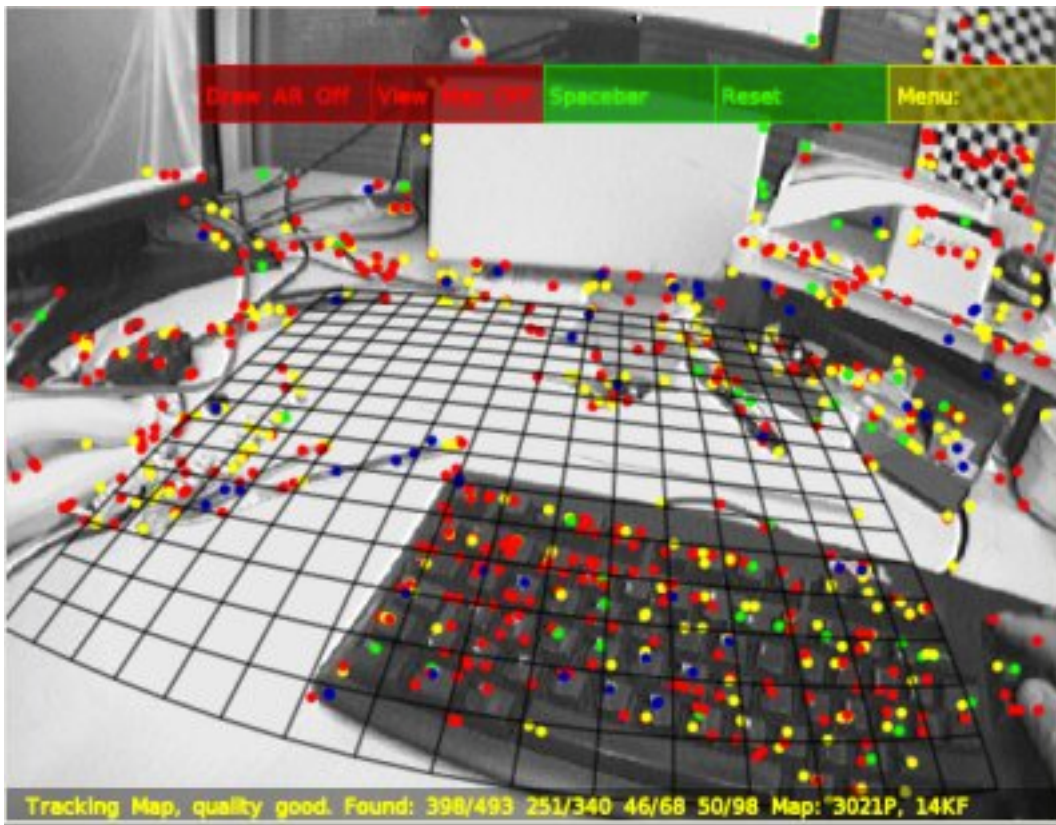
**Figure 2.5:** Detected Feature Points in PTAM (from Klein and Murray (2007))

measurement data of each detected feature point in relation to the camera position at the time the feature was recorded. These initial measurements are continuously re-evaluated with a bundle adjustment algorithm which runs in the mapping thread. This bundle adjustment adjusts the pose for key frames based on updated measurements which are created by exploring the environment. However as they are computationally expensive operations, in order to not impact on the performance of the tracking thread, when tracking is being performed only "local bundle adjustment" is allowed on the map. Local bundle adjustment simply limits the operation to the most recent key frame and it's four nearest neighbours for any given pose update. This enables the map to be kept up to date for new pose estimations, while ensuring the algorithm does not impact on the tracking performance.

This powerful tracking and mapping setup allows for content to be anchored within the environment with much more stability than other approaches. The reasoning behind this is that the position of a virtual object within the PTAM environment can be closely linked with a camera pose estimation and therefore the content is only considered for display once the current camera pose is local to this pose estimation. This removes the processing requirement of constantly evaluating the position of every piece of virtual content within an environment, allowing more power to be given to the current display and evaluation of the "visible" content.

## 2.4.2    Content Delivery in PTAM

The focus of the PTAM software as a research tool is heavily on the capability of its tracking and mapping algorithms. The anchoring of content in this environment forms an intrinsic part of this, such that the content appears stable within an environment and does not drift when viewed from several angles. There is scope to expand upon PTAM in order to improve upon the means of content delivery provided within the application in order to exploit the robust tracking and mapping for a wider range of applications.

The current means of inserting content into a PTAM environment requires the user to interact with that environment in real time. While exploring the environment, a simple mouse based interface is provided in order to facilitate the introduction of virtual objects (Figure 2.6). In order to place these objects, the user can select a position on the screen, and then fine tune the positioning with a series of $x$,$y$, and $z$ arrows. This approach allows the user to instantly review the content positioning in three dimensions and make adjustments according on what they can see, which allows for a high level of placement accuracy. However while highly accurate, this approach to content delivery has a high time requirement, as the user is physically interacting with the space whilst placing content. The user's interpretation of 3 dimensional space, and particularly depth perception can result in the content placement looking perfect from one viewpoint, before realising it is incorrect from another. The flexibility afforded by interacting with the content in its environment to correct such issues is valuable, however the time penalties
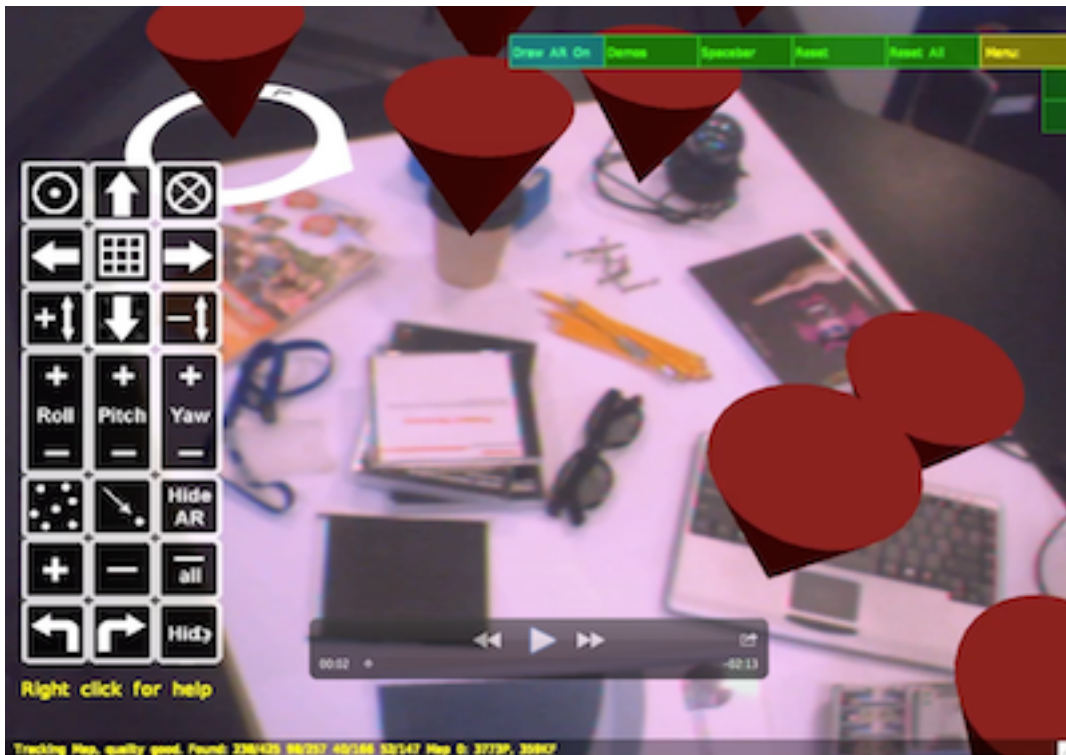
**Figure 2.6:** Content Placement Interface in PTAM

associated with such interaction are not ideal for a wide range of real world scenarios.

The content placement is stored within PTAM as a separate entity to the tracking and mapping data. While the data structure is similar (references to camera pose estimations and known locations within the map) it is not reliant on any single feature or group of features. This further adds to the robustness of the PTAM approach, as should the tracking algorithm fail to re-detect a feature point, the content can still be displayed as long as enough of the surrounding feature points are detected to trigger the camera localisation within the map. This allows for the continued operation of the system even in environments which change, without the explicit need to remap them.

Currently PTAM limits content delivery to a single user within the environment. While it would be possible to see an extension to this means of content delivery for collaborative Augmented Reality, it would still require both users to occupy the same space. This would work well for an extension of the MagicMeeting (Regenbrecht, Wagner and

Baratoff (2002)) system, where the marker-based tracking system could be replaced as all users occupy the same working environment, however it could not provide the base for collaboration in remote situations, such as the indoor/outdoor "Godlike" collaboration metaphor dealt with in Stafford, Piekarski and Thomas (2006).

While PTAM does not provide a sophisticated means for content delivery in these respects, the underlying principles form an excellent base upon which to create one. By utilising the feature point and camera pose estimation data structure maintained in each environment map, systems can be developed on top of PTAM which are guaranteed the tracking and mapping reliability and stability which has been discussed here.

## 2.5   Chapter Overview

Sophisticated tools exist for the tracking and mapping of Augmented Reality environments, both concerning fiducial marker based tracking, and more recently, markerless tracking for unprepared environments.  These technologies have also given rise to a number of collaborative systems, which distribute Augmented Reality content either across a number of users within a shared space, or in the case of some advanced systems, amongst remote users. The metaphors developed by Höllerer (1999) and Stafford, Piekarski and Thomas (2006) for remote content sharing provide a powerful means of delivering content within their own systems. There is scope however, for tools to expand these metaphors into a more generic interface for remote content delivery in Augmented Reality.

Just as research has been conducted into generic means of authoring Augmented Reality content independently of underlying applications (Schmalstieg et al. (2007), and Hill et al. (2010)), a means of delivering content independently of underlying applications is also desirable. While Julier et al. (2000) shows that a "command center" to "solider" metaphor is possible, GPS and inertial sensors are relied upon as well as the vision system. With the development of scalable monocular SLAM systems (Eade and Drummond (2006)), and the powerful PTAM (Klein and Murray (2007)) platform, a

generic means of remotely providing content for a solely vision based Augmented Reality system becomes important.

# 3

# Keyframe Tagging

## 3.1 Introduction

This chapter discusses the requirements for an image based content delivery system to work with an Augmented Reality environment in order to provide a means for introducing new content into any environment. By examining real world scenarios of use, the requirements are distilled into a novel approach for remote[1] content delivery into a previously unknown environment[2]. The implementation of the proposed Keyframe Tagging approach and evaluation of its utility will be undertaken in further chapters.

## 3.2 Content Delivery as a Problem

An Augmented Reality system is one that it is capable of real world environmental tracking, and is able to display virtual content within that environment. These are often considered as a tightly coupled problem, and many Augmented Reality systems therefore

---

[1]Remote is defined as a user not present within the live Augmented Reality environment regardless of whether the actions take place in real time or not, or whether the user is geographically remotely located.

[2]In this context an unknown environment refers to one in which neither the user nor the system have any prior knowledge.

aim to solve both at the same time. The reality of this is that the environmental track-ing portion is given the most weight, as without a reliable and stable tracking algorithm there is little point in displaying content, as it would not appear in the expected place within an environment. While this is an important observation of the overall structure of an Augmented Reality system, this thesis acknowledges that many excellent tracking systems now exist for Augmented Reality, and instead focuses on the content aspect of these systems.

In order to appreciate a content-focused approach to Augmented Reality, it is impor-tant to decouple the perception of tracking and content systems being one and the same, and look at the means of content delivery for Augmented Reality. When considering the process of selecting the position of some virtual content within an environment as an independent entity, it is possible to develop this as a problem in its own right. For the purposes of this thesis, we will consider only Augmented Reality tracking systems that map an unknown environment, as opposed to ones which require the introduction of fiducial markers or other visual signposts for them to function. The output of such a tracking system is a set of environment data, or a "map" of that environment. Figure 3.1 shows that we can consider such a map as the input to a content delivery system, which in turn provides a modified version of that map as an output.
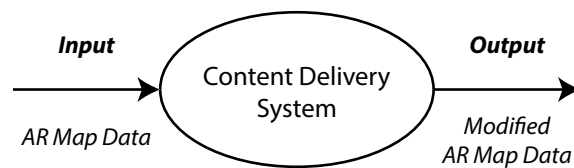


**Figure 3.1:** High Level Data Flow for Content Delivery

This thesis takes the content delivery problem and provides a solution which ad-dresses the need for scalable and flexible content delivery into an already existing envi-ronment. The proposed method is called Keyframe Tagging (KFT), which recreates an existing environment map for the user and allows them to place content within it in any location. The need for a system such as KFT is born out of the acknowledgement that

while a good tracking system is crucial to the success of any Augmented Reality application, without a scalable means of users delivering content into that environment, the utility of the tracking system becomes severely limited.

## 3.3 Scenarios

In order to understand the role which a content delivery system such as KFT can play, here several scenarios are presented outlining possible real world uses, and demonstrating the need for scalable content delivery. Each of the discussed use cases present different challenges, which gives rise to the different requirements of the content delivery aspect of Augmented Reality systems.

### 3.3.1 SN1: Virtual Office Media. A Static, Offline Environment.

There is a pervasion of technology into office life, and many of the early proposals for Augmented Reality systems were born out of meeting room and workspace situations. Such scenarios are perfect for explaining the role and utility of content delivery distinct to the tracking element of Augmented Reality. A potential use of Augmented Reality in the workplace is to allow the virtual tagging of content around an office building. This could take the form of a virtual presentation that is present on a meeting table, and viewable through any means of display technology be it Head Mounted Display or smartphone as two examples. The requirement placed on the tracking element of such a system would be to identify the correct meeting table, and place the correct presentation upon that table.

However, the content delivery aspect of such a system is required to add scalability and flexibility. Without a means of allowing users to easily customise what content is displayed on this meeting table, or where on the table it appears, the system's utility is quickly limited to one static case. Were the developer needed to make alterations to the code for each presentation then the utility of the system is reduced. With a content delivery system that puts that power into the user's hands without the need for program-

matic change, the possibility of multiple content items or locations becomes possible. By tracking the entire office building, and then having a content delivery system capable of recreating that environment for a user to explore, the potential for virtual information and knowledge sharing is huge. It would be possible to attach any notices to information boards, office walls, schedules to meeting room doors and so forth. Building upon these ideas, it would be possible for each user to see customised content in each location depending on who they were.

From a content delivery point of view the key aspect of this is that in some offline capacity, a user can manipulate the types and position of content without the need for programmatic changes to the underlying system. This is made possible due to the fact that the structure of an office environment is unlikely to change much from month to month, and so a map of such an environment can be modified independently of that environment with the confidence that it is still relevant. Additionally, as the content placement is being performed offline, the user can theoretically take as much time as they need to get the content positioned in an accurate manner.

### 3.3.2    SN2: Training Exercise. A Static, Real-Time Environment.

Building upon the content delivery requirements of Scenario SN1, there are use cases which could require content delivery into an online environment. Here "online" means changing the content which is in front of a user as they explore the environment in real time. Augmented Reality systems already have a large utility in training and simulation, from medical applications (Bichlmeier et al. (2007)) to vehicle mechanics (Henderson and Feiner (2009)), as the ability to change an environment in a non-destructive real-time way presents a unique opportunity to trainers and supervisors. Typically, training and simulation exercises are undertaken under highly controlled circumstances, which is ideal for Augmented Reality as they will likely happen in an already known, static environment. A training room for example, can be considered for the most part static, as the dimensions and structure of the room will not change, even if the exercise requires that things within the room are altered. Considering a training exercise which relies

on no physical changes to the environment (in that a new environment map can be created for each exercise) and only changes to the virtual content, this environment can be considered "static".

Therefore, the role of content delivery is to allow the trainer to explore the environment separately to the candidate and make selections about where to place virtual content as in Scenario SN1. However, in order to allow the realtime flexbility of altering the content while the candidate is in the live environment, there are certain other expectations of the system. Most importantly, the trainer must be able to access a quick overview of where content has been placed, in order to remove or change it as required. This also places a requirement on the output side of the content delivery system. In order to allow seamless integration into a real-time use case such as this, the content delivery system must be able to quickly produce a representation of the environment map reflecting the changes, in the same format as the tracking system provided it. Without the ability to do this quickly and reliably, the content delivery system would be limited to only offline changes.

A scenario such as this provides an interesting challenge when it comes to the method of content placement. The trainer must be able to review locations and place content quickly in order for real time changes to have the desired impact. This must be done while also maintaining an acceptable level of accuracy, so that the content appears where the trainer is expecting. Unlike with Scenario SN1 there is a time pressure which could affect accuracy, and in order to make the proposed content delivery system as scalable and flexible as possible, this should be taken into consideration in the systems' design.

### 3.3.3 SN3: Emergency Response Support. A Dynamic, Real-Time Environment.

In the case of Fire and Rescue, Emergency Response Teams (ERTs) will often be required to enter a building or an environment which has been changed in some way, or has become more difficult to negotiate. In the case of a natural disaster, this could include structural collapse or a similar large scale alteration to the landscape. In situations such

as this the use of any system which relies on environmental information, or landmarks is very difficult. A computer vision system that requires the tracking of a known object will struggle to function, or cease to function at all if that landmark has been removed from the environment. Therefore it is desirable to have a system which can operate in completely unknown environments. By utilising a mapping system such as the one provided in Parallel Tracking and Mapping (PTAM)(Klein and Murray (2007)), the team on the ground can be gathering a new map of the surroundings which can be transmitted to other teams and/or a command centre to provide up-to-date information on the ground.

In such a situation, it is desirable to share information between multiple teams on the ground, particularly with regards to navigation concerning rescue efforts, or potential hazards to be avoided. While this would be possible with the existing approach offered by PTAM (Klein and Murray (2007)), it is potentially dangerous to require the team on the ground to spend time placing the content manually while they are in the environment. A more efficient solution is to have a content delivery system which can operate remotely, in order to allow a command centre operative to manipulate the environment on the ground, allowing each of the EMT members to see the results.

In order for this to be possible, there needs to be a means of transmitting the mapping data from the ground back to the control centre, and then allowing for this data to be manipulated but not corrupted in order to provide the content delivery service. Considering a Disaster Relief scenario such as this, the set of requirements is largely focused around object placement speed and accuracy. Time is an important consideration, and therefore the approach that is adopted must allow a user to quickly place content. While accuracy is a consideration, there is some scope for a tolerance of inaccuracy in exchange for speed benefits in this scenario. In a chaotic disaster relief scenario, a navigation pointer being placed 50cm away from its intended target could be an acceptable trade-off in exchange for that pointer being placed quickly.

## 3.4   Keyframe Tagging

The aim of this thesis is to present the proposed novel Keyframe Tagging (KFT) method of introducing content into an Augmented Reality environment. The approach described in the thesis is generally applicable to content delivery for Augmented Reality, however the scope of the thesis is to consider an implementation built on the PTAM system described in 2.4. KFT takes into account the requirements arising from these scenarios to provide a straightforward means for a user to introduce content into an already existing Augmented Reality environment remotely, such that the content appears reliably and accurately in place within that environment. KFT considers the content delivery aspect of this independently to Augmented Reality tracking and mapping, and as such these aspects will be dealt with by another system. In terms of the data flow shown in Figure 3.1 the map data created by such a system is considered the input and output to the "Content Delivery System". KFT fills the role of "Content Delivery System", which is expanded in Figure 3.2 to show the three distinct stages which must occur within the KFT system to manipulate the input and produce the required output data.
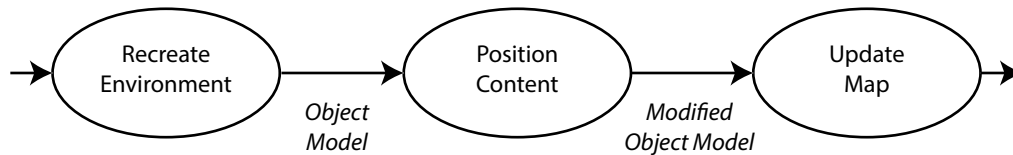


**Figure 3.2:** KFT Data Flow (Expanded from Figure 3.1)

These three key stages were identified through studying the use case of such a system. This system is designed to be used offline and possibly remotely, as opposed to live in the Augmented Reality environment. As such, it must be able to recreate the target environment in a meaningful way, allow the user to select where they want to insert content, and produce an updated version of the mapping data which is still valid and useable by the original tracking and mapping software. Additionally, the scenarios demonstrated several qualities to be included for the KFT approach to content delivery to be considered

both flexible and scalable. Scenario SN1 (Section 3.3.1) describes a use case where the user is not time pressured when placing content, giving time to ensure that their selections are accurate. This is contrasted by both Scenario SN2 (Section 3.3.2) and Scenario SN3 (Section 3.3.3) which describe use cases that require a certain level of speed for the user placing content. This gives rise to an interesting speed vs accuracy dynamic which will be explored in the evaluation of this approach, but in order to ensure the resultant system is as flexible and scalable as possible, both fast placement and accurate placement will be considered when designing and implementing the solution.

### 3.4.1   Input and Output

While Figure 3.2 shows the three main areas which are addressed by the KFT approach, this section focuses on the expected data flow both into and out of such a system, shown in Figure 3.1. As discussed previously, the KFT system is only concerned with content delivery into Augmented Reality systems that are capable of tracking unknown environments. That is, systems which are able to make a map of their surroundings and keep a reference co-ordinate set in order to establish the users' whereabouts within that environment. Such systems are commonly grouped under the Simultaneous Localisation and Mapping (SLAM) term, however it is possible that other tracking and mapping approaches could be used, as long as they adhere to the input requirements discussed here.

In examining the possible use case scenarios, in particular Scenario SN2 and Scenario SN3, a common theme is that the approach taken in the KFT system should provide the means for a user to be delivering content into a live environment in real time. When considering the input data of the system, this has an impact on how the visual representation of the environment would be best handled. While some tracking systems make use of a video feed, such a data set would be large and vulnerable to potential corruption should it be used for continuous serialisation and synchronisation between two sites. Therefore the decision was made that the KFT system would be based upon a series of photographs of the environment.

In addition to the photographs of the environment, the set of Feature Points and co-ordinates detected by the tracking and mapping system must be available. This data should include the real world co-ordinates of the Feature Point, camera position information from the time that the Feature Point was detected, and a reference to the last created photograph before the Feature Point was detected, such that each Feature Point has a "source image".

In terms of the output from KFT, the manipulation of the mapping data which occurs as part of content delivery is the injection of new co-ordinate points and their referencing to a virtual model such that the content is positioned in the correct location. In order to minimise the impact on the environment map, and therefore its validity, no other supporting data created as part of this approach will be inserted into the final map. This is to ensure that the original tracking and mapping system can still utilise the map data, producing the new content in the expected place.

### 3.4.2 Recreate Environment

One of the key requirements of a remote content delivery system is that it is able to recreate the target environment. Without this ability there would be nothing to insert new content into. As the content delivery problem is primarily one of data manipulation, the first step is to rebuild the environment data into a usable object model. In doing so, every aspect of the data is accessible in an expected format. The key components of the data are:

1. The image based representation, or "Keyframes"
2. The detected Feature Points within the environment
3. The content objects (if importing a map with already existing content)

Figure 3.3 shows key components 1 and 2 and their relationships within the environment map as an expected input format. It does not depict how the content objects (key component 3) fits into the model at this stage, as the given model is the minimal requirement for input to the KFT system. The environment map model holds a list of
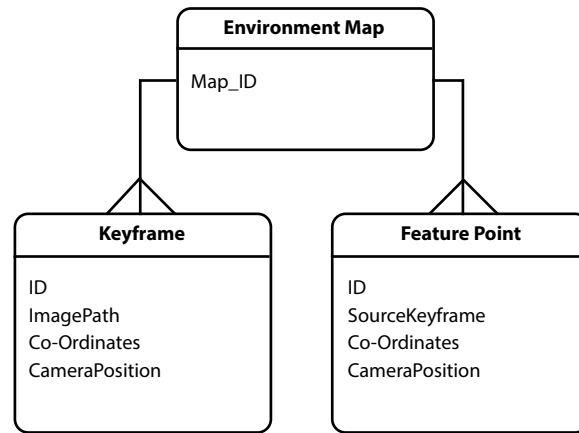
**Figure 3.3:** Entity Relationship Diagram of the Map Model

Keyframes and Feature Points, while an association is made between the two with regards to the "Source Keyframe" attribute within Feature Points, such that there is always a reference back to a Keyframe from each detected point.

In addition to the data attributes which are outlined in Figure 3.3 the environment map will likely consist of many more data attributes which are not required for the manipulations within KFT. While it is therefore not important to build them into the environment recreation model, they must not be discarded, so as to protect the validity of the output map data.

### 3.4.2.1  Image Based Representation

The utilisation of images as the base data format as opposed to a video feed has several advantages for the object model underpinning KFT. Most importantly, an image can have a unique reference easily attached to it. While this is theoretically possible through the bookmarking of a video stream, it is a much more straightforward task to assign a unique identifier to an image extracted from a video feed, and base the rest of the data structure off this information. Additionally, if the system is being tracked and mapped in realtime, updates can be sent to KFT instantaneously, by simply adding to the set of images. With video, a full resynchronisation would be required, which would add over-

heads that have a negative impact on the usability for the KFT system. In addition to this it would increase the operational requirements of such a system considerably, with large amounts of bandwidth required for a continuous stream of video to be synchronised.

### 3.4.2.2   Detected Feature Points

The detection of Feature Points is crucial for the tracking and mapping system to be able to locate the user's position within a map whilst they are exploring an environment, and display the relevant information back to them. They represent a series of points within both a Keyframe image, and the environment for which known, and reliable co-ordinates exist. They are therefore a valuable point of reference for the latter stages of content delivery, actually choosing the position for new content. It is for this reason that they form a crucial part of the object model upon which KFT is built and cannot be omitted.

In terms of recreating the environment, it is important to keep a close relationship between the Feature Point and the frame in which it was detected.

### 3.4.2.3   Content Objects

While KFT is not compatible with allowing for the manipulation of content added to the environment via other systems[3], it has to support further editing of content added by KFT. That means that a map with already existing content must be parsed, and the object model must reflect not only that content is present, but in which Keyframe image the content was originally positioned. The user must see this existing content no differently to content which was positioned in the current session, and as such the object model should treat it the same.

---

[3]KFT is not concerned with allowing the user to manipulate the positioning of objects placed with any other existing content delivery system other than itself.

#### 3.4.2.4    Presenting The Environment

Once the object model has been created and validated, the next step in the recreation of the environment is to actually present it to the user in a meaningful way. In order to achieve this the KFT system will present the user with a filmstrip of images, in chronological order as they were mapped. By scrolling through the interface in this way, it provides the user with a sense that they are looking around the environment rather than looking at distinct images. Different perspectives on the same area of the environment will be offered in logical groupings, rather than being scattered throughout a random series of images.
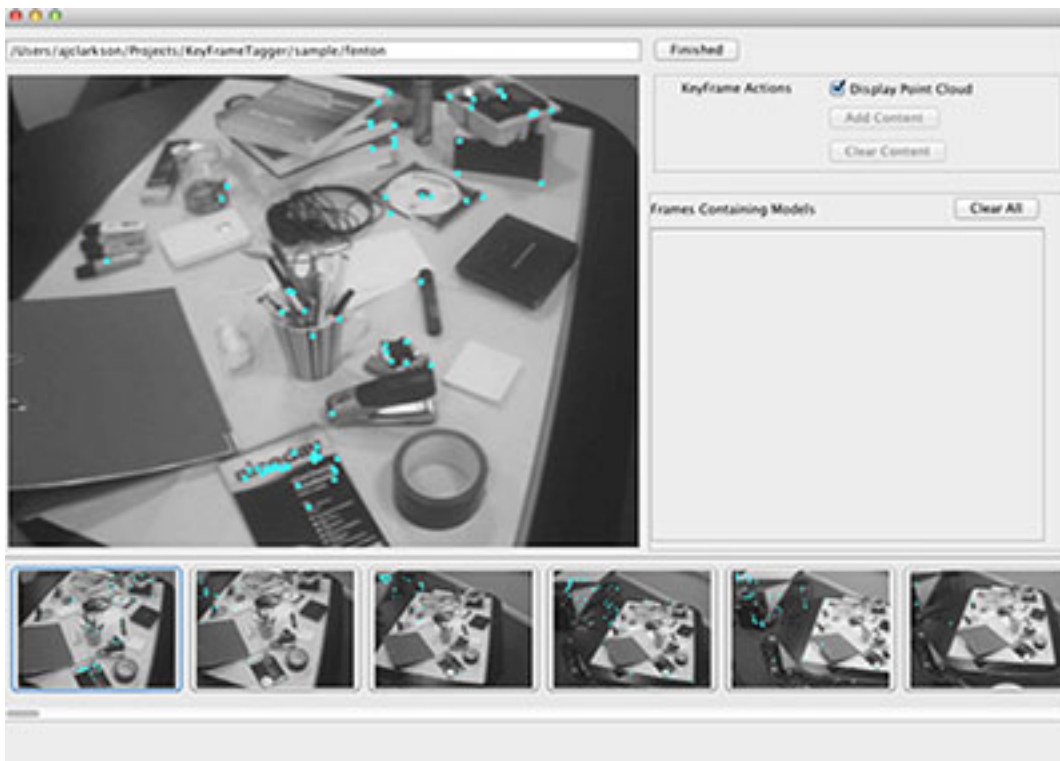


**Figure 3.4:** The KFT interface

Figure 3.4 shows the KFT interface. The chronological filmstrip of images can be seen at the bottom of the interface, with a larger version of the currently selected image displayed in the centre of the interface. Several points can be seen overlaid on both the

thumbnail and larger images, which are visual representations of the Feature Points detected by the tracking and mapping system. They are shown here overlaying the "source image" as discussed in Section 3.4.1. These Feature Points should be included in the interface in order to provide the user with a frame of reference when it comes to positioning the content, as explained in the following section.

### 3.4.3  Position Content in Environment

Allowing the positioning of content within an environment is the key role of the KFT approach. It strives to allow users to place content anywhere they desire within an environment while ensuring that the placement is accurate enough that it reliably and repeatedly appears in the correct location when passed back to the tracking and mapping system. In order to facilitate this, the "Tagging" aspect of Keyframe Tagging was conceived.

The notion of "tagging" an image is one which many users will be familiar with it because of its existence across social networks and other applications. This is an ideal paradigm to adopt, particularly as making a positional selection on a photograph is an intuitive action, even for new users. The challenge associated with "tagging" a Keyframe image is that the user is essentially being asked to make a three dimensional selection from a two dimensional image.

KFT overcomes this by making full use of the Feature Points which exist within the environment for which full real world co-ordinates are provided. These points provide a frame of reference which is vital to exploring further positioning. To tie a user to simply place content upon a detected Feature Point would result in perfect accuracy but over a severely limited number of available content positions. Therefore KFT allows for the user to stray away from these Feature Points for content positioning, while retaining their utility in inferring new depth co-ordinates based on their distance from the desired point of content placement. In doing so, the user is free to make a selection where they wish but the system supports their selection.

### 3.4.3.1    Increasing Feature Point Resolution

The KFT approach will not however, allow the user to freely select any part of a Keyframe image. While doing so initially seems like the ideal means of positioning content, this freedom afforded to the user actually has the potential to hinder the accuracy of their selection. KFT allows them to "increase the resolution" of the available Feature Points by creating several Virtual Points, grounded in the co-ordinate system of the known Feature Points. By doing so the user is required to think more carefully about where they are placing content, and the content's relationship to other known points.

These Virtual Points, while crucial to the positioning of content in the map will not become a permanent feature of it. Rather, the Virtual Points exist temporarily as an aid to the positioning of content. The reasoning for this choice is that if KFT injected every new Virtual Point into the map data and then passed this back to the tracking system, the tracker would seek each point out and include it when assessing the tracking status of the map. This could prove problematic, as the Virtual Points have no relation to physical features existing in the environment, and therefore cannot be tracked. As KFT is concerned only with content delivery into an environment, there is a responsibility to ensure that any changes do not impact the environment in a negative way.

These Virtual Points are created by calculating the centre point of a line bisecting any two selected points (Equation 3.1). This point becomes the Virtual Point, and the algorithm can be used recursively within an image to generate as many points as required by the user.

$$
\begin{aligned}
SP_1 &= (x_1, y_1, z_1) \\
SP_2 &= (x_2, y_2, z_2) \\
VP &= ((x_1 + x_2)/2, (y_1 + y_2)/2, (z_1 + z_2)/2))
\end{aligned}
\tag{3.1}
$$

Using the 'two-point' average from Equation 3.1 the co-ordinates for $VP$ are acceptable, but can be improved upon. While the $x$ and $y$ co-ordinates for both $SP_1$ and $SP_2$ can be trusted in terms of accuracy, the accuracy of the $z$ co-ordinate is harder to ascer-
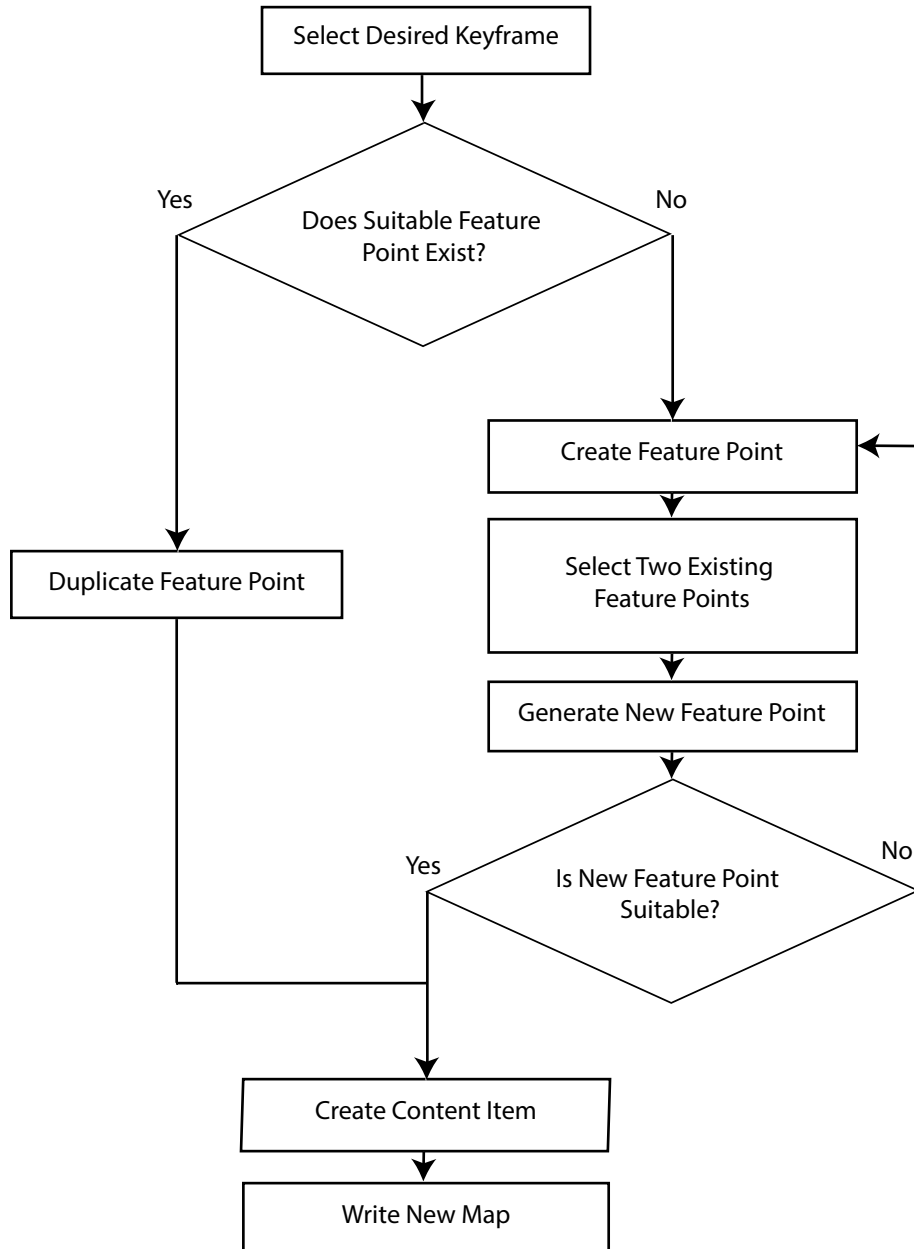
**Figure 3.5:** Content Positioning Flowchart

tain. This is due to the nature of trying to appreciate the depth of a three dimensional environment from a two-dimensional image. In order to provide more accuracy for the $z$ co-ordinate in $VP$ the KFT system draws on known co-ordinate information from the nearest neighbours of $SP_1$ and $SP_2$, within tolerances, to smooth the $z$ co-ordinate prediction.

Algorithm 1 shows a pseudo code representation of how KFT will analyse the Source Keyframe of a selected point in order to identify its nearest neighbours for z-smoothing. Two tolerances are used in order to filter the set of possible points, first looking for clusters with similar $x$ and $y$ locations, before checking that the $z$ co-ordinate is within tolerable bounds. The algorithm is then applied to each of the two selected points.

---

tolerance = filter out points that are too far away;
z-tolerance = closer filter for z-distance;
z-candidates = set of points to use for z-smoothing;
sp = selected point;
points = set of all Feature Points from this frame;
**while** *points has next* **do**
    p = current point;
    **if** *p.x - sp.x < tolerance **and** p.y - sp.y < tolerance* **then**
        **if** *p.z - sp.z < z-tolerance* **then**
            add p to z-candidates;
        **end**
    **end**
    move to next point;
**end**

---

**Algorithm 1:** Pseudo-code representation of identifying suitable nearest neighbours of a Selected Point

Following the identification of suitable nearest neighbours, the points identified in z-candidates are weighted. These are then used to calculate a weighted average z co-ordinate which will be used in generating the co-ordinates of the new Virtual Point. Algorithm 2 provides an example of calculating these weighted sets, which in turn have z-smooth (Algorithm 3) applied to them to provide weighted averages. This has the impact of placing more weight on neighbours which have z co-ordinates closer to the

selected point. The application of these algorithms provides a new $z$ value for the Selected Point which will be used in the calculations for $VP$.

```
z-tolerance = closer filter for z-distance;
z-candidates = set of points to use for z-smoothing;
close-points = calculated set of nearest neighbours;
distant-points = calculated set of distant neighbours;
close-weight = weighting given to nearest neighbours;
z = weighted z value for new point calculation;
while z-candidates has next do
    p = current point;
    if p.z <= z-tolerance then
        add p to close-points;
    else
        add p to distant-points;
    end
end
close-average = z-smooth(close-points, close-weight);
distant-average = z-smooth(distant-points, 1 - close-weight);
z = (close-average + distant-average) / z-candidates.size;
```

**Algorithm 2:** Pseudo-code representation of weighting the nearest neighbour average for z co-ordinate calculations.

```
z-candidates = set of points to use for smoothing;
weight = average weighting to use for these candidates;
z = new z value; while z-candidates has next do
    p = current point;
    z = z+p.z;
    move to next point;
end
z = z / z-smooth.size;
return z * weight;
```

**Algorithm 3:** Pseudo-code representation of the z-smoothing operation for a Selected Point

The new $z$ values associated with the Selected Points provide more confidence in the accuracy of the $z$ value of the newly created Virtual Point. In particular this has a

limiting effect on the assumption that the user can reason which two points are required to produce the desired midpoint. While the $x$ and $y$ co-ordinates of a point may appear correct on the Keyframe image, there are some circumstances in which the $z$ co-ordinate is slightly off, from a rapid camera angle change, or two objects of differing depth being very close together.

Another problem which KFT must address is that of occlusion. When faced with a two dimensional image overlaid with Feature Points, it is impossible for the user to tell whether a Feature Point is on a foreground or background object should the two intersect. This is a problem which is common in Augmented Reality systems, and while some live environment systems take surface scanning approaches to avoiding it, that is not feasible within the KFT system. Therefore the system must allow the user to review their selection, or offer alternative angles in order to ensure they are aware that their selection has not fallen victim to this problem.

### 3.4.3.2   Inserting Content into the Environment

When an ideal position has been identified for the content, the system must allow the user to place the content. While the KFT approach is independent of the type of content, visual feedback must be provided to the user in order for them to ascertain that this action has taken place. In order to support the review of user's content positioning a bookmark should be added to the appropriate Keyframe image, so that the user can quickly revisit those frames which have content attached to them and review their positioning.

Figure 3.5 shows a flowchart representation of the process required to position content through the KFT interface which has been described here. The key decisions which dictate a user's actions are whether or not the currently selected Feature Point is suitable for content placement. If a suitable point is available immediately within the Keyframe with no need to generate further Virtual Points, this Feature Point is duplicated before being used as a basis for creating a Content Item. The reasoning behind this is that an existing Feature Point within an environment map is tied to a physical object or feature within that environment. Should that feature disappear, then the associated Feature

Point will not be displayed in the live environment, and as such any associated content would be lost. By duplicating the Feature Point and converting it to a Content Item, this is not a problem as only the co-ordinate data is carried over and no relation to detected features.

## 3.4.4   Update Environment Map
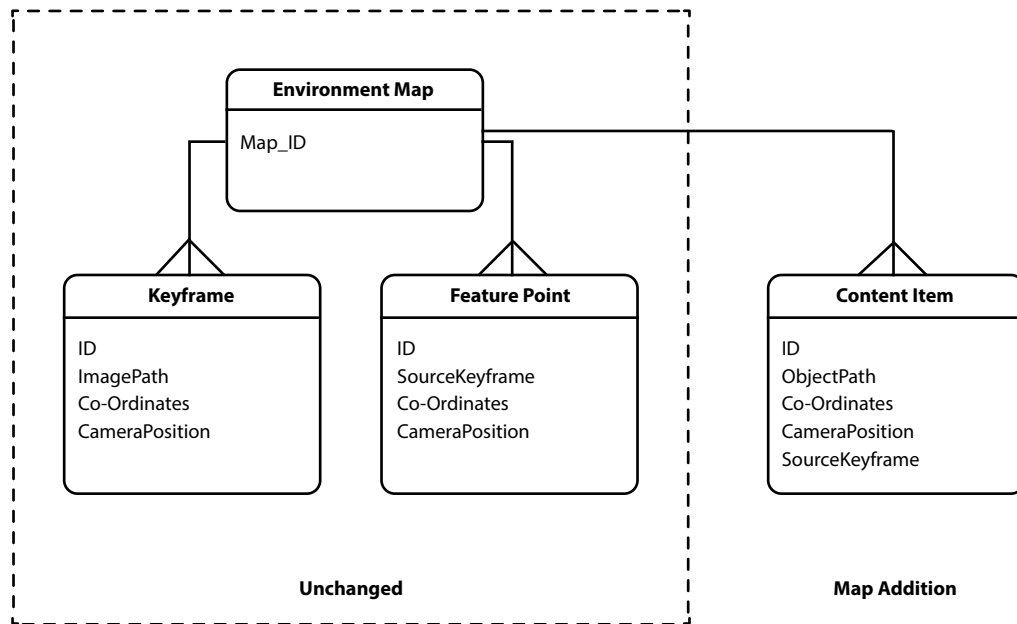


**Figure 3.6:** Entity Relationship Diagram of the Map Model with Content Item Addition

The final responsibility of KFT is to update the original map files with any changes. Any changes to the map should be as minimal as possible so as not to interfere with the integrity of the mapping files, and ensure that they are still able to be read by the underlying tracking and mapping system. Content placements will have the correct co-ordinate format by the nature of their creation, derived from one or more existing Feature Points.

Figure 3.6 shows the modified Map Model Entity Relationship Diagram, expanded from Figure 3.3. The Content Item is illustrated as the only addition to the map, and the attribute structure of the expected Content Item can be seen to closely resemble that of the Feature Points from which it was created. Only the addition of an "Object Path" is present from its base attributes, provided in order to allow a file path to the content which should be added into the environment. As discussed in Section 3.4.1 the model depicted in Figure 3.6 could now be considered as an input for future operations of the system. KFT should be capable of manipulating its own modified maps as if they were a session currently running on a data model such as that shown in Figure 3.3. It is important to ensure that any other unused fields[4] belonging to Keyframes, or Feature Points and consequently Content Items, are also carried over into the updated map and not ignored by the serialisation process. If the validity of the initial environment map is not maintained, the updated version will not be usable in the underlying tracking and mapping system.

## 3.5   Requirements of KFT

The set of key requirements which can be derived from the discussion of this approach are as follows. These requirements will serve as a base for the implementation of the software used to evaluate this method:

Table 3.1: Derived Requirements of KFT Implementation

| #   | Requirement |
| --- | --- |
| RE1 | Load an existing AR map from a data source |
| RE2 | Recreate the environment from the data source |
| RE3 | Present the user with an image-based representation of the environment |
| RE4 | Highlight detected Feature Points with known co-ordinates |
| RE5 | Allow the user to navigate through a series of images to explore the environment |

---

[4]Fields which are present in the tracking data, but are not required for the functionality of KFT.

| # | Requirement |
|---|---|
| RE6 | Allow the user to create new "Virtual Points" based on the known locations of Feature Points to increase the number of available content anchors. |
| RE7 | Allow the user to attach content to one of these Feature/Content points. |
| RE8 | Allow the user to review the positioning of their content |
| RE9 | Inject the new map data into the existing data source |
| RE10 | Ascertain the validity of the new map data |
| RE11 | Save the new map data to the existing data source for exploration |

## 3.6   Chapter Overview

By considering a set of real world use case scenarios for remote content delivery in Augmented Reality environments, the proposed novel KFT approach for content delivery is introduced. Taking into consideration the requirements which arise from each of these scenarios, a data model is proposed which can be manipulated to insert content into an environment. By looking at the data flow which happens internally with Content Delivery systems, Figure 3.5 proposes a flowchart of actions which must be facilitated by the system in order to allow for the accurate delivery of content into an area which the user chooses. Section 3.4.3 discussed several issues facing the approach of requiring users to position content in a three dimensional environment using a two dimensional reference image, before Figure 3.6 offers an example modification of the input map model to serve as output from the KFT system. Finally, a set of requirements are derived which KFT must satisfy in order to successfully facilitate the delivery of content into remote Augmented Reality environments.

# 4

# Implementing the KFT Software

## 4.1 Introduction

This chapter takes the approach discussed in Chapter 3 and the identified requirements and offers an explanation as to how the key requirements were implemented in the KFT system. After providing some context for the implementation each of the three key requirements will be discussed in further depth.

## 4.2 Implementation Context

In order to evaluate and analyse the feasibility of the Keyframe Tagging approach described in Chapter 3, a novel prototype system was developed. The KFT system provides the means for users to add content into already existing Augmented Reality environments, with a focus on ease of control and accuracy of object placement. In order to develop a testable system, it was necessary to ascertain the role of KFT and interplay with other components within a workable Augmented Reality infrastructure. As this approach is concerned only with the means for delivering content into an environment, it

was not necessary to cover existing ground, and reimplement the tracking and mapping aspects of an Augmented Reality system.

After surveying several options for tracking and mapping systems (Section 2.2) and considering them in terms of their application to the discussed usage scenarios of this thesis (Section 3.3) the Parallel Tracking and Mapping (PTAM) system by Klein and Murray (2007) was identified as the chosen base system. While it would be possible to apply the KFT approach introduced in Chapter 3 to a range of tracking and mapping systems by processing the output data into an image based representation, the PTAM system provides this by default and so provides an excellent platform to build a prototype implementation without the need for further data manipulation.

```
┌─────────────────────────────────────────┐   PTAM
│       Track & Map Environment            │   (off the shelf)
└─────────────────────────────────────────┘

                  Map XML Data

┌ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ┐
│ Keyframe Tagging Software                 │
│                                           │
│     ┌─────────────────────────────┐       │
│     │     Recreate Environment     │       │
│     └─────────────────────────────┘       │
│          Present Environment to User       │
│     ┌─────────────────────────────┐       │
│     │ Position Content in Environment│      │
│     └─────────────────────────────┘       │
│          User Selects Desired Position     │
│     ┌─────────────────────────────┐       │
│     │   Update Environment Map     │       │
│     └─────────────────────────────┘       │
└ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ┘

               Modified Map XML Data

┌─────────────────────────────────────────┐   PTAM
│   Reload Map & Track Environment         │   (off the shelf)
└─────────────────────────────────────────┘
```
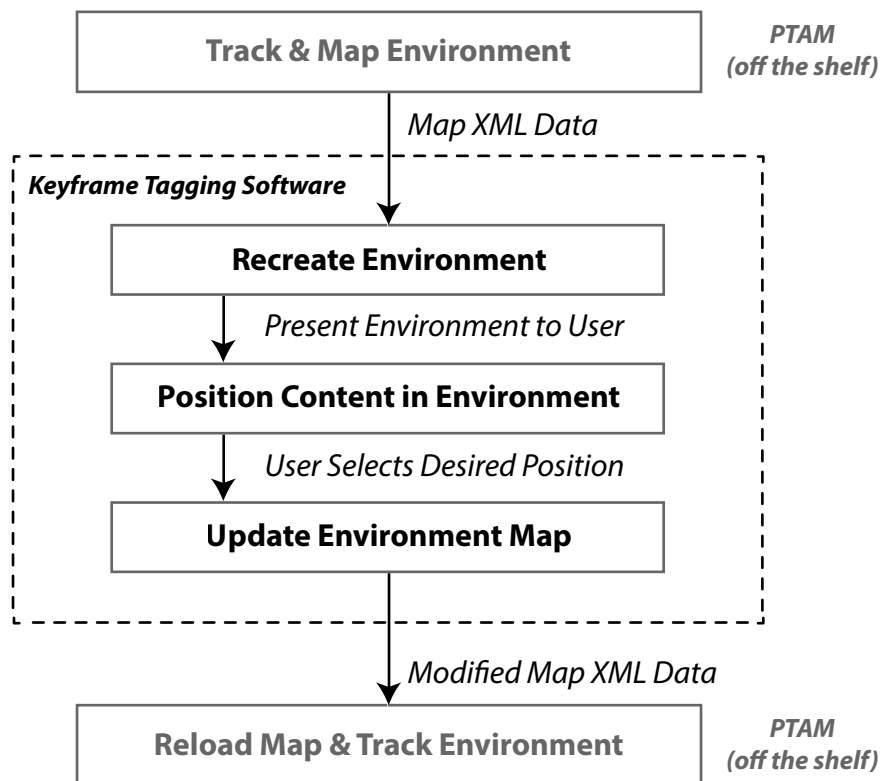
**Figure 4.1:** Content Delivery for Augmented Reality Pipeline

The pipeline shown in Figure 4.1 gives a high level overview of how the components

of the system will fit together. The distinction between which components are provided by PTAM (top & bottom) and which are provided by KFT can be seen from this. The environment mapping is provided by PTAM, before being passed into the KFT software, where the user can manipulate the data, and produce a modified map containing their content. This is then passed back to PTAM in an accepted format so that the environment can be re-explored with the content in place.

This data flow could theoretically take place in real time, with the PTAM software modified to provide real time updates to the tagging system, and receive them with new content in place. The dotted lines in Figure 4.1 illustrate the break points where this serialisation of data would need to occur. However, the focus of this thesis is on the means of content delivery, rather than the infrastructure which enables it and therefore the content delivery currently takes place with non real-time updates to both systems. In terms of the scenarios introduced in Section 3.3 the focus of this implementation follows Scenario SN1. Scenarios SN2 and SN3 would require further infrastructure setup to enable real-time synchronisation of the data, which is not essential for measuring the accuracy of content positioning in an environment.

## 4.2.1   Building on PTAM

As the proposed approach builds on top of the PTAM software, it is important to give the reader a high level overview of some of the characteristics of PTAM which make this possible.

The tracking and mapping algorithm developed by Klein and Murray (2007) for the PTAM system operates on a video feed from a single camera source. After a quick location independent calibration, the system is able to interpret stereo depth and so tracking a three dimensional environment becomes possible with the use of one lens. In order to provide all of the functionality of a Simultaneous Localisation and Mapping system, PTAM must record a series of reference points from the video feed, such that it can track its own position within the map as it is being created. In order to do this, a series of still images, termed "Keyframes" are recorded at regular intervals. Each of these

Keyframes is a black-and-white snapshot, and holds a real-world co-ordinate relative to the position of the camera when it was recorded. In addition to recording these images, a series of "Feature Points" are identified by PTAM as it scans the environment. By running edge detection and Feature Point recognition algorithms on the incoming video feed, a number of straight edges and corners are identified. These Feature Points have co-ordinates relative to the camera position, crucially including depth information provided by the initialisation procedure of the system. Each Feature Point belongs to a Keyframe, which is the last one recorded before a feature was detected.

When a Feature Point is recorded, its position relative to the camera, the Keyframe it belongs to and other positioning information are stored. When the camera revisits that position within the environment, the Feature Points can be reliably mapped over the environment to provide a means of localisation for the system. This makes it possible to leave a room and then return to see the Feature Points laid out exactly where they should be. The tracking provided by the PTAM system is sufficiently robust to serialise all of this information to a file, re-enter and recreate the environment at a later date.

The properties of these Feature Points are exactly what are required of a piece of content introduced to the environment. They exist independently of any environment knowledge, other than the map created by PTAM. As such, when a user entering an environment establishes the map tracking, the Feature Points are all reestablished in their place. PTAM has some Feature Point redundancy built in, such that if n object is removed from the scene, and that object had a Feature Point attached to it, the map can still continue to function, assuming there are still enough Feature Points to reinitialise the tracking. The finer details of the tracking and mapping algorithms within the PTAM software are not pertinent to this thesis, and the reader should consult Klein and Murray (2007) for further information in this area.

## 4.2.2   Development Technologies

The KFT system was developed using JAVA because of its ability to run cross platform and the wide range of external software libraries available to complement the feature set

of the developed tool. A custom Model View Controller (MVC) framework was developed using a modification of the MVC paradigm similar to that found in the Apple Cocoa framework (Apple (2011)). Here, the controller mediates the flow of data in both directions. This means that a change in the model is communicated, via the controller, straight to the view. This allows for more flexibility in inter-model communication updating the view components of the system interface. A model may only affect a view which is registered with it, but in allowing this two-way data flow, the model becomes completely decoupled from the view. This modified MVC is depicted in Figure 4.2.
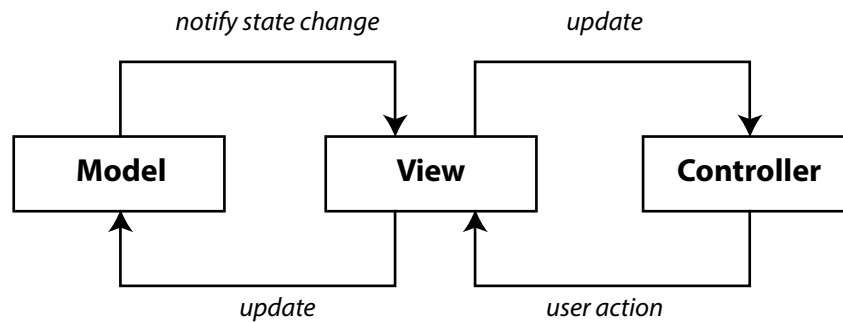


**Figure 4.2:** Modified MVC Design

In order to provide the means for this system to be further expanded to work with other underlying technologies, the main input and output format of the data will be XML. For the purposes of this thesis, the developed schema for serialisation and unserialisation is specific to the PTAM data structure, which specifies the Feature Point and co-ordinate data as part of an XML file. A script which pre-processed incoming data from other systems to match the expected format would be possible, though outside of the scope of this thesis.

## 4.3   KFT System

As outlined in Chapter 3 Figure 3.2 the three key features of the proposed approach are:

- The ability to recreate the live environment which was mapped.
- The ability to provide a means of the user selecting a Feature Point to attach content to in an environment.
- The ability to attach a content object to the selected area in a format which will seamlessly reintegrate with the chosen tracking and mapping system, such that the environment map is still usable.

This section will take these three aspects of the KFT system and explain their implementation in more detail.

## 4.3.1    Recreate Environment

The KFT system represents the environment using a series of still images, allowing the user to navigate chronologically backwards and forwards through the environment as it was explored. Due to the underlying tracking and mapping system being PTAM, these images are easily available, as they form part of its data set. Using these in addition to the recognised Feature Points from PTAM, the live environment can be rebuilt.

Due to this, the approach will take the Keyframe images straight out of PTAM, and along with a serialised data file containing all of the Feature Points and their positioning information. From this, the developed software will be able to recreate the environment in a meaningful way, and one in which the user can understand when navigating it. By keeping the positional information from the Feature Points as a central part of the reconstruction, some known co-ordinates are provided to each Keyframe which will come in useful when delivering content into the environment.

Figure 4.3 shows how the environment is represented within the KFT system. The currently selected Keyframe is presented in large form, above a thumbnail strip of the other Keyframes within the environment map. This can be scrolled through to allow the user to navigate the environment. On both the selected Keyframe and the thumbnails, small points can be seen drawn across the image. These appear cyan in colour to contrast with the black-and-white images provided by PTAM, and represent the location of the
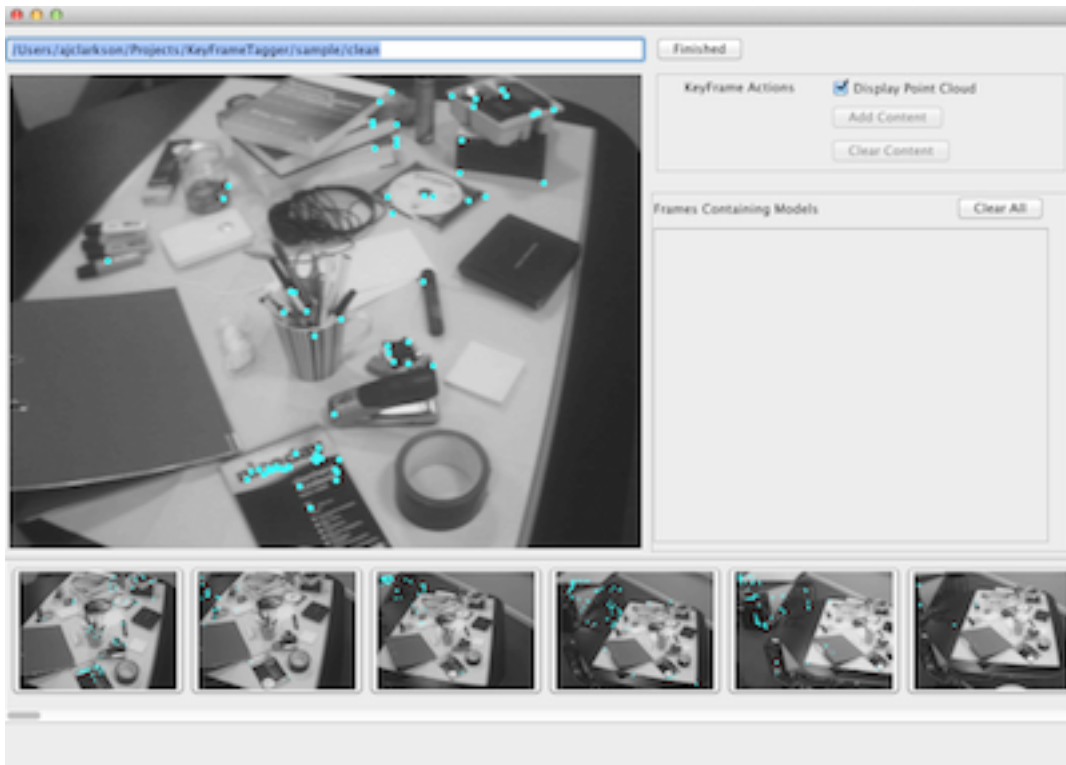
**Figure 4.3:** Environment Recreation in KFT: Keyframe Overlaid with Feature Points

known Feature Points. These Feature Points are positioned within the frame utilising their relationship between a "Source Keyframe" and the coordinate translation which PTAM offers between these Keyframes and the real world co-ordinates in relation to the camera when tracking the environment.

KFT actively filters an incoming data set to remove Keyframes from the recreation if they contain less than 5 Feature Points. In developmental testing, it was found that Keyframes with less than 5 Feature Points often offered little to no utility to the user when considering the ability to increase the resolution of these Feature Points, a process described in Section 3.4.3.1. This amount is configured within a system wide configuration file, and so is easily customisable to meet the needs of a particular environment. The feature is useful in ensuring that a user is not overwhelmed by Keyframes, as PTAM creates images at regular intervals, so the longer spent in the environment, the more

frames are recorded.

## 4.3.2   Position Content in Environment

The representation of Feature Points within any given Keyframe, allows the user to select their desired Feature Point as a target upon which to attach content. The implementation of this is straightforward, as the Feature Points are derived directly from the dataset of PTAM and therefore have a real world co-ordinate attached to them.  However, as discussed in Section 3.4.3 the KFT approach seeks to provide a means of increasing the number of points available to the user, so that content positioning is not restrictive. Therefore, the pseudo code listings which can be found in section 3.4.3 were implemented in the backend of the Keyframe Tagging system, to take advantage of nearest neighbour calculations for z-smoothing, resulting in the best available estimation for depth calcuations when providing Virtual Points. The system treats the Virtual Points as any other map point, though they are only persistent within the environment for the running time of the program. They are not serialised within the mapping data, such that any erroneous calculations from this technique do not compromise the overall stability when the map data is loaded back into the tracking and mapping software.

Figure 4.4 revisits the Keyframe and interface depicted in Figure 4.3 in order to show the same scene augmented by a number of Virtual Points.  It clearly shows the power of being able to bridge large gaps in the original Feature Point set with these new points, especially when dealing with large flat surfaces, that are typically ignored by the edge detection algorithms underpinning PTAM and other similar tracking and mapping softwares.

## 4.3.3   Update Environment Map

The final key requirement of the KFT system as identified in Chapter 3 is to actually deliver the content into the environment, once a suitable location has been identified. While one of the stipulations of KFT is that an object should be attached to a Feature Point (or created Virtual Point), the actual delivery of the content provides an object with
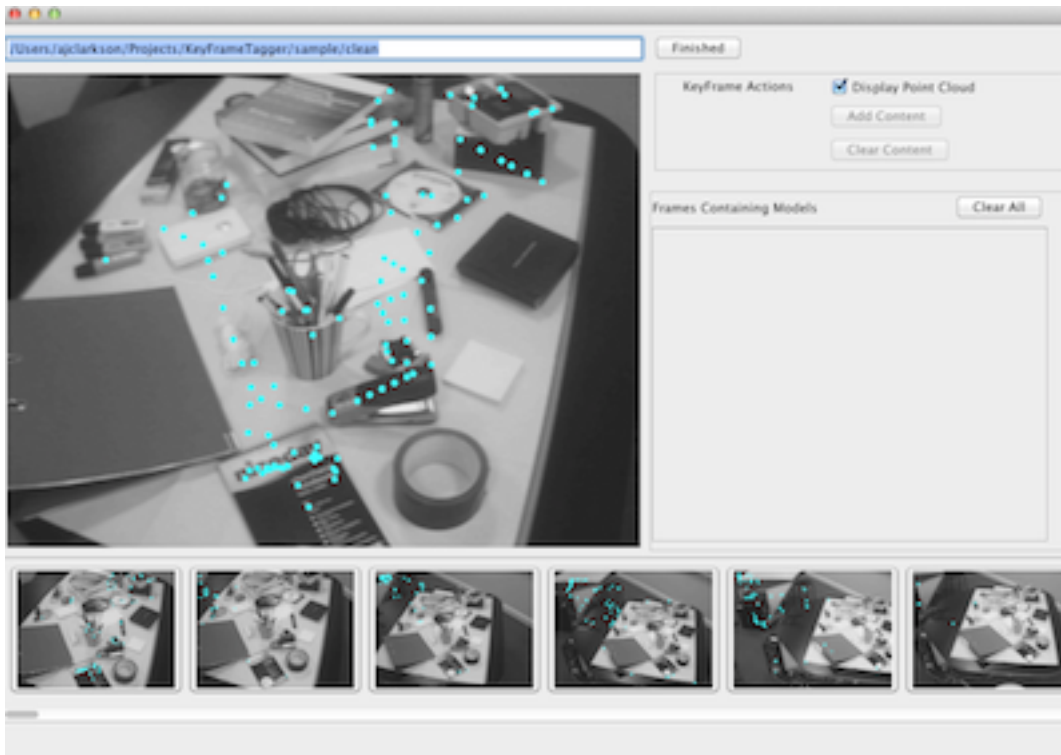
**Figure 4.4:** Increasing the resolution: New Virtual Points added to the Keyframe from Figure 4.3

a co-ordinate system and relationship independent of that. While it would be possible to link the co-ordinates to that of a point, if the point disappears from the map, the content would not be shown.

The content is attached through serialising the new object data to XML which is injected into the existing map as provided by the underlying tracking and mapping system. Similarly to the environment recreation methods described in Section 4.3.1 for the purposes of this thesis this is done using a schema designed to format the data for the PTAM system, but could be translated using a post-processing script for other applications. Once the user has attached content to a point, the Feature Point overlaid on the Source Keyframe in the KFT interface is highlighted in a different colour (orange) to distinguish the fact that it has content attached. Additionally a second filmstrip of thumbnails appears to allow the user to quickly jump to the frames that they have attached content in. This can be seen in Figure 4.5.
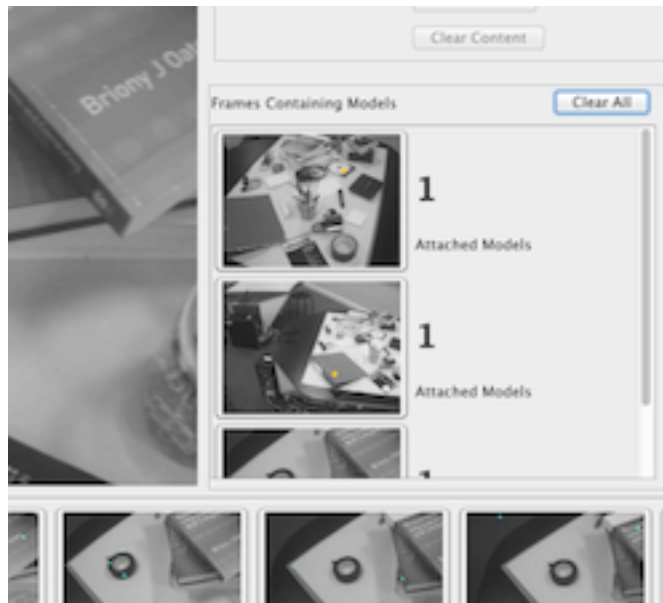
**Figure 4.5:** Additional Shortcuts to Content Attached Keyframes

The addition of content into the environment triggers the serialisation of that data into the existing map file, such that it seamlessly becomes a part of the original data map. While each user created Virtual Point exists within the run time data model, KFT has been implemented to not include these points as Feature Points within the map file. The reasoning behind this decision is that the user-created points are not tied to a physical feature within the environment as the other Feature Points are, and as such should not be treated by the tracking and mapping software as Feature Points. A tracking algorithm would waste valuable (in terms of performance) computational time trying to locate the physical position of these "Feature Points" in the environment, when one does not actually exist. The disadvantage to taking this approach is that should a user wish to reload the environment map at a later date, the Virtual Points will not be present. However, as one of the stated requirements of this approach was to make as little impact as possible on the original data, and the fact that Virtual Points should only be viewed as a content-positioning aid for the user, the benefits of their exclusion outweigh the disadvantages.

Therefore the additional content can be seen in the following XML Schema snippet

describing the "contents" part of the model as depicted in Chapter 3 Figure 3.6:

```
<xs:element name="contents">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="content" maxOccurs="unbounded" minOccurs="0">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element type="xs:integer" name="id"/>
                        <xs:element type="xs:string" name="objectPath"/>
                        <xs:element type="xs:string" name="coords"/>
                       <xs:element type="xs:string" name="cameraPosition"/>
                        <xs:element type="xs:integer" name="sourceKF"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

The fully modified schema showing the injection of this new collection of content items can be seen in Appendix A. The implementation of KFT to comply with this schema then allows for the manipulation of the output data to match the tracking and mapping algorithm selected, in this case PTAM.

## 4.4   Chapter Overview

This chapter has given insight into how the KFT system proposed in Chapter 3 was implemented in support of this thesis. Through exploration of the requirements which were stated in that Chapter, the solutions are offered here to ensure the resultant software performs as expected. The discussed XML schema for serialising the KFT data into a translatable format, can be found in Appendix A.

In terms of the requirements outlined in Section 3.5 the following Table 4.1 gives an overview of whether or not each has been completed:

**Table 4.1:** Implementation Status of KFT Requirements

| #    | Requirement | Completed? |
|------|-------------|------------|
| RE1  | Load an existing AR map from a data source | ✓ |
| RE2  | Recreate the environment from the data source | ✓ |
| RE3  | Present the user with an image based representation of the environment | ✓ |
| RE4  | Highlight detected Feature Points with known co-ordinates | ✓ |
| RE5  | Allow the user to navigate through a series of images to explore the environment | ✓ |
| RE6  | Allow the user to create new "virtual points" based on the known locations of Feature Points to increase the number of available content anchors | ✓ |
| RE7  | Allow the user to attach content to one of these anchor points | ✓ |
| RE8  | Allow the user to review the positioning of their content | ✓ |
| RE9  | Inject the new map data into the existing data source | ✓ |
| RE10 | Ascertain the validity of the new map data | ✓ |
| RE11 | Save the new map data to the existing data source for exploration | ✓ |

Requirements RE1 and RE2 were achieved through the implementation of an image loader and xml parser which was provided with a unmarshalling description to translate the PTAM data into the KFT input format. Once loaded, this allowed the Graphical User Interface shown in Figure 4.3 to be constructed with images of the environment and a filmstrip navigation between images, satisfying RE3 and RE5. By processing the relationship between detected Feature Points and their source images as described in Section 4.3.1 it was possible to generate an overlay of dots representing these Feature Points within the interface, satisfying RE4.

By performing mouse click localisation on each of the displayed feature points, it is possible to allow users to make selections on the displayed Feature Points Sections.

This, in conjunction with the Virtual Point Generation covered in Section 4.3.2, satisfy Requirement RE6. Requirements RE7 and RE8 were implemented by allowing any selected point to become a content anchor and have a bookmark created to signpost this fact to the user as described in Section 4.3.3 and Figure 4.5.

In order to serialise the newly created content to the original map (Requirement RE9), a marshalling description was provided to the implemented xml parser to ensure the data structures produced output data compatible with the PTAM system. For Requirement RE10 a validation against a schema was performed after this step to ensure the output data was valid and would not affect the operation of PTAM when using the new map file, before finally saving the data into PTAM's expected format (Requirement RE11).

The next chapter will evaluate the performance of the system when used for content delivery in an Augmented Reality environment.

# 5

# Experiment 1: Investigating KFT Object Placement Accuracy

## 5.1  Introduction

Operating in a previously unknown environment presents a unique set of challenges to an Augmented Reality tracking system, most notably the technological challenge of utilising a map for localisation whilst this map is still being created. As the proposed KFT system builds on top of such systems, a different set of challenges are taken into consideration with regards to unknown environments.

As KFT is not responsible for the tracking of the environment, the unknown aspect presents more challenges to the user of the system than it does the system itself. In order to convey context and meaning to a user, KFT recreates the operating environment in a meaningful way in order to assist the user with content delivery. In order for the system to be judged a success, the user must be able to place content with a high level of accuracy within any environment, regardless of whether or not they have prior knowledge of it. The rest of this chapter discusses the design, undertaking and results of an experimental

73

study designed to test the placement accuracy of virtual content within an environment, along with some other metrics which may impact on the final accuracy of placement.

This study was designed in order to attain answers to a subset of the Research Questions presented in the introduction to this thesis. The questions under consideration in this experiment are displayed in Table 5.1

**Table 5.1:** Research Questions Addressed in Chapter 5

| RQ | Research Question |
| --- | --- |
| RQ1 | Can a user place content into an Augmented Reality environment using a photograph based reconstruction of that environment? |
| RQ2 | Is content placed using a photograph based reconstruction of the environment positioned with an acceptable level of repeatable accuracy? |

## 5.2   Study Design

The focus of this study was not what content was being added to the environment, but rather how accurately any piece of content could be placed. Due to this the participants were simply asked to select the location in which they wish to place the content, rather than being required to select the content to add. Taking the scenarios discussed in Chapter 3 into account, an office environment was selected as the target due to the availability of such an environment for tracking and mapping purposes. In order to recreate the "unknown" element of the environment partitcipants were not introduced to the specific experimental environment in any way before they were presented with it via the KFT interface. This ensured that their performance with regards to speed and accuracy were solely due to the representation of the environment in the KFT system and not based on prior knowledge.

A PTAM map was created of a complex environment (involving more elements than a standard desktop), comprising of a desktop with assorted objects, a second tabletop, windowsill and view from the window. Several objects were contained within the scene,

all typical to an office environment such that no one object stood out as identifiably different. The environment was scanned and tracked using PTAM, and once a full map of the environment had been built, this was exported to be the map used for each participant, and for the creation of an expert benchmark.

Within the PTAM environment, the live environment tools provided by PTAM (as outlined in Section 2.4.2) were used in addition to the manual manipulation of the xml co-ordinate data by an expert user in order to place "perfect" examples of location tags for each of the ten objects. Manipulation of the xml data allowed for several tags to be placed for each object using the live tools and averaged out to give a central point for the benchmark. The mapping function of PTAM was disabled while this process was carried out, so as ensure the tag placement was carried out on the same base map as the participants would use in KFT. This placement data was then saved and considered the benchmark against which all other participants efforts using the KFT system was judged.

In addition to the "perfect" location, the live environment tools were used to provide supplementary location tags for the data analysis, which were placed at the boundaries of each object. This enabled a scale of "distance from the centre of an object, to out of bounds" to be used as a measure of accuracy.

In order to test the approach, the participants were instructed to use the KFT system to find and then tag each of the target objects. The only instruction they were given with regards to placement was that the tag should be placed as close to the centre of the top face of each target object. This was necessary in order to allow the results of each participant to be compared against created benchmark data to measure accuracy.

### 5.2.1 Hypotheses

In order to answer the research questions presented in Table 5.1, a set of hypotheses were derived for this experiment:

**HP1.1:** The average tag quality from the KFT approach will be "Good"[1]

---

[1]The accuracy ratings as defined for this experiment are shown in Section 5.2.3.

when compared with the live placement benchmark.

Given that the participant is required to interpret the approximate depth position of the desired position in a photograph, the tag location will not be accurate enough to be considered "perfect" in a straight comparison. This is due to the fact that in a live placement with no time restraints to create the "perfect" benchmark the expert user can explore the environment from all angles and make adjustments in real time, with no need to make assumptions on the $z$ co-ordinate positioning in a photograph.

*HP1.2:* Participants who review more Keyframes will have a higher level of accuracy.

By reviewing several Keyframes, a participant will see the same area of the environment from different angles. This will provide a different set of Feature Points from which to create their tags, and so allow them to place content more accurately than a participant who utilises the first Keyframe in which each target object appears.

*HP1.3:* Users who create more Virtual Points will have a higher level of accuracy.

By creating Virtual Points based on the available Feature Points the participant increases the resolution of positions available to them to attach content. They also take advantage of the $z$ co-ordinate estimations built into KFT. By creating more Virtual Points in order to deliver the content precisely where they want, the $z$ co-ordinate becomes more accurate.

*HP1.4:* The longer a participant takes to complete the experiment, the higher their overall accuracy will be.

Several of the discussed scenarios in Chapter 3 deal with the trade off between speed and accuracy. This was a low pressure environment without the need for high levels of speed, however the participants were instructed that they should spend no longer than

20 minutes on this task. With a time limit in mind, the participants who spent less time identifying and tagging each target object will have the lower positional accuracy results.

### 5.2.2 Technologies

To provide the tracking and mapping facilities required to create the initial Augmented Reality environment, the Parallel Tracking and Mapping (PTAM) system was used to create the environment map. As outlined in the implementation of the KFT system (Chapter 4), this is the base system which KFT was implemented to build on top of. The PTAM system allowed for a map of the environment to be created and saved providing each participant with an identical base point, and ruling out the possibility of any influence on their performance coming from an external system. The participant had no knowledge of the underlying tracking and mapping and they only interacted with the environment through the KFT interface as developed for this thesis.

In terms of hardware, this study was focused solely on the tagging of target objects, and therefore could theoretically take place on any desktop machine running JAVA (as the implementation language of this software) with the required external libraries installed. A MacBook Pro (2.7Ghz i7) was used as the desktop machine in order to ensure that the same machine could be used for every participant. Additionally, as this is a laptop computer, a wired three button mouse will be used as the input device for the participants in order to rule out any unfamiliarity with the use of a trackpad.

The data recording for this study was solely provided by system logging built into the KFT system for all timings and user actions. Qualitative data was gathered via online questionnaires both before and after the experiment.

### 5.2.3 Method

The participants for the study were chosen from Durham University staff and students. For this study a group of 20 took part with ages ranging from 20 to 29 years old, and an average age of 24. The participants were 65% male to 35% female, and they were not paid for participating in the experiment. Each participant was required to complete the

experimental task once, tagging the location of ten objects. This provided 200 tags with which to analyse the accuracy of the KFT approach.

The instructions given to the participant were to explore the environment and place a tag as close to the centre of the top surface of an object as possible. These tags would then be used to provide an accuracy rating based on their proximity to the tag in the benchmark map. In order to ensure the accuracy was a metric relative to the objects size, four possible levels of accuracy were outlined based on the distance away from the centre of an object as defined by the "perfect" position on the benchmark map. Where 0 is the centre of the object, the four levels were defined as percentage distances radiating from this point, and up to the boundary of the object (also defined within the benchmark data):

- Perfect = 0-25% distance from centre
- Good = 26-50% from centre
- Acceptable = 51-75% from centre
- Poor = 76-100% from centre
- Failure = Out of Bounds

The participants were required to undertake a brief training exercise in order to achieve a level of mastery in the task. This was to ensure that all participants had the same baseline when using the software, and rule out any timing discrepancy that may be introduced due to different rates of familiarisation within the participants. The training exercise initially comprises of a brief explanation and demo of the tool, before the participant was asked to complete the tagging of one object in a separate testing environment. Once the participant had completed this with satisfactory accuracy, they could continue to the main task.

During the experiment proper, the participants were be advised to complete the list of target objects in any order they chose, and they were advised that they may revisit previously tagged objects to review their positioning throughout the experiment. Once

the participant was satisfied with their tagging across all ten objects, they indicated this by selecting "finished" within the interface, and the log files were then saved and locked.

After the task, the participant was asked to complete a short survey in which they rated the positioning of their tags for each object from "Exactly where I wanted it" to "Not at all where I wanted it". The participant was free to review their positioning for each object in the KFT system while completing this task, with no impact on their data as the logging system was already locked by this point. This provides some insight into whether or not the participant was satisfied with their selections, or felt limited by the software. A comments box was also provided to encourage the participant to elaborate on any issues they encountered when using the software.

### 5.2.3.1  Controlling Other Variables

In order to ensure there are no differences in the availability of Keyframes or Feature Points within the presented environment, each participant was provided with the same cleanly initialised base map. Due to the fact that there was no dependence on the environment after the base map had been created, there was no need to control this environment until all participants had completed the task. This would have been necessary if the participants tag locations were evaluated in the live environment.

The creation of the base map took place in controlled, reproducible conditions. The tracking algorithm in PTAM can sometimes produce unexpected results with variable lighting. In order to ensure that any inaccurate tags are a result of the participants interaction with the KFT tool, environmental conditions affecting PTAM must be ruled out. Therefore, artificial lighting was used to ensure the tracking algorithm did not struggle throughout the creation of the map. Additionally access to the environment was controlled such that nothing could be moved or completely removed during the tracking phase.

The base map was created by making a single pass around the room, with time taken to look at each element in the room from different angles. This was to recreate the conditions in which a map would be made in a real environment, such that excessive

time was not spent ensuring good Feature Point detection around the target objects.

Should the Keyframe tagging interface fail at any point during the experiment the participant would have to withdraw from the experiment and be replaced by a new participant. This is to ensure that no participant has prior knowledge of the mapped environment before starting the task. No participant should be permitted to restart the task after they have started it.

The evaluation of each participant's tag placement took place independently of the PTAM environment. The XML data was compared to the expert user's benchmark map in order to remove the need for the expert user to make a judgement on the accuracy of a participant's tags.

## 5.3   Experiment Results

In order to judge the success of the KFT approach through this experiment, each participant's tag placements were analysed in comparison to a benchmark ideal placement, produced without time constraints using the content placement tools available within the PTAM system. A centre marker, and boundary markers were contained within this benchmark, and accuracy scores were then calculated from the percentage distance from this centre marker, up to the boundary of the top surface of the object. The percentage distance from centre of each object was averaged for each participant, and then inverted in order to provide an average accuracy score for each participant. These results were then considered in terms of the accuracy compared with the PTAM benchmark, and whether the number of Keyframes reviewed, Virtual Points created, and total time take (up to the allowed time limit) had an impact on the accuracy score achieved by the participant. The rest of this chapter will present and evaluate the results.

### 5.3.1   Comparing KFT Placement Accuracy to the PTAM Benchmark

The accuracy scores awarded to each participant were categorised on the following scale:

**Table 5.2:** Accuracy Rating Scale for Participants

| Distance From Centre | Accuracy | Accuracy |
|---|---|---|
| 0-25% | 100-76% | Perfect |
| 26-50% | 75-51% | Good |
| 49-75% | 50-26% | Acceptable |
| 76-100% | 25-1% | Poor |
| Out of Bounds | 0% | Failure |

**Table 5.3:** Average Accuracy Score Using KFT For Placement

| Min. Number | Max. Number | Mean | Standard Deviation |
|---|---|---|---|
| 46.20 | 72.89 | 63.61 | 7.89 |
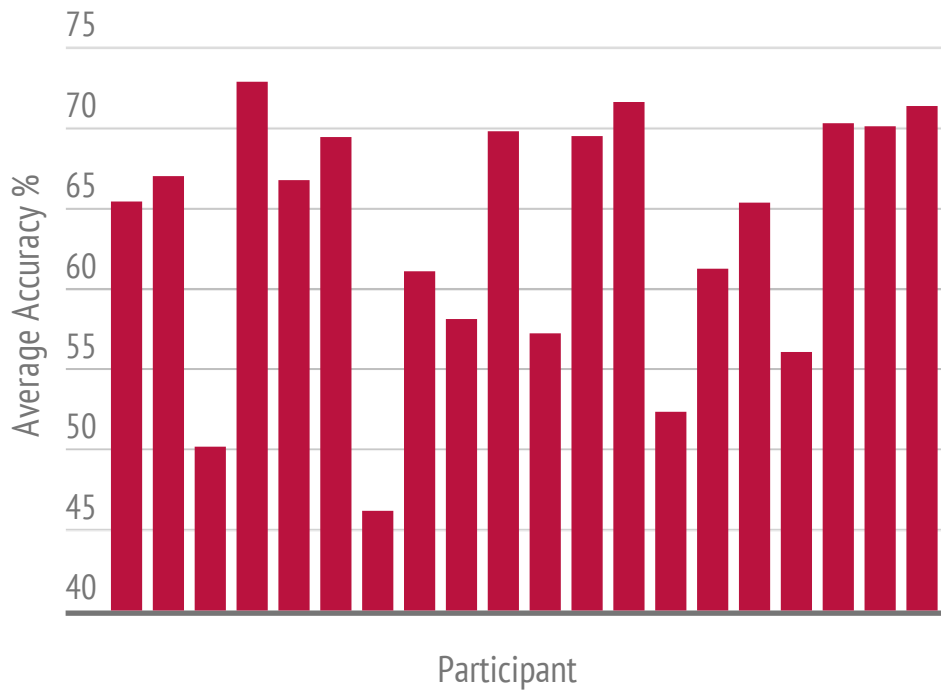


**Figure 5.1:** Average % Accuracy Scores by Participant, Compared to PTAM Benchmark

When looking at the average accuracy score achieved by all participants shown in Table 5.3, the mean score achieved was 63.61% (with a standard deviation of 7.89). This falls within the range of "Good" as predicted in the first hypothesis laid out for this experiment:

> **HP1.1:** The average tag quality from the KFT approach will be "Good" when compared with the live placement benchmark.

Figure 5.1 shows the average scores of each of the 20 participants. The lowest average accuracy score achieved by any participant was 46.20%, which falls within the "acceptable" range, whilst the highest average score was 72.89% remaining in the "Good" range. This data shows that with inexpert users it is possible to repeatedly achieve a high level of tag placement accuracy across a series of objects. Indeed, no participant placed a tag outside the bounds of an object, considered a "Failure" and only 8 tags across the set of 200 were considered "Poor"

While this data address the predictions raised in Hypothesis 1, and clearly shows that the KFT approach can be used to deliver content into an environment accurately when considered in comparison with a PTAM benchmark, a more detailed look at the approach each participant took to arrive at these accuracy scores is required to fully the performance of KFT as an approach.

## 5.3.2    Investigating Whether The Number of Keyframes Reviewed Impacts Placement Accuracy

The KFT approach allows a user to freely explore an environment by looking at multiple Keyframe photographs of it. For this study, the participants were presented with the interface and given a demonstration of its capabilities with regards to environment exploration. In order to make full use of the Virtual Point creation afforded by the KFT approach to increase the resolution of available content placement points in an environment it is advantageous to review several Keyframes before selecting a starting point for each object.

**Table 5.4:** Average Number of Keyframes Used

| Min. Number | Max. Number | Mean | Standard Deviation |
|---|---|---|---|
| 48 | 224 | 103.30 | 46.30 |

Table 5.4 shows a range of 176 between the minimum and maximum number of Keyframes reviewed by all participants. This was largely due to differing approaches taken by the participants. In observing their actions, two clear approaches were identified. One approach was to quickly scroll through the environment scanning the thumbnail images and select the first one which contained both the target object, and some nearby Feature Points. From there the participant would either reject this frame and repeat the same process, or continue with the process of placing content. The other approach was to carefully look at multiple frames containing the target object, moving between them before making a decision on which provided the best starting point for content placement. The second approach lead to many more Keyframes being reviewed, which did have an observable impact on the accuracy. Figure 5.2 shows the average accuracy score of each participant plotted against the number of Keyframes which they used.

The positive correlation between the two variables as seen in Figure 5.2 demonstrates the positive impact reviewing more Keyframes had on accuracy. The data yields a Spearman Correlation = .590 $p < 0.01$. This demonstrates a moderate - strong positive correlation of the data, in support of the second hypothesis laid out for this experiment:

> **HP1.2:** Participant who review more Keyframes will have a higher level of accuracy.

## 5.3.3 Investigating Whether The Number of Virtual Points Created Impacts Placement Accuracy

By allowing the user to create new target points for attaching content to, one of the key features of the KFT approach is the ability to increase the resolution of available
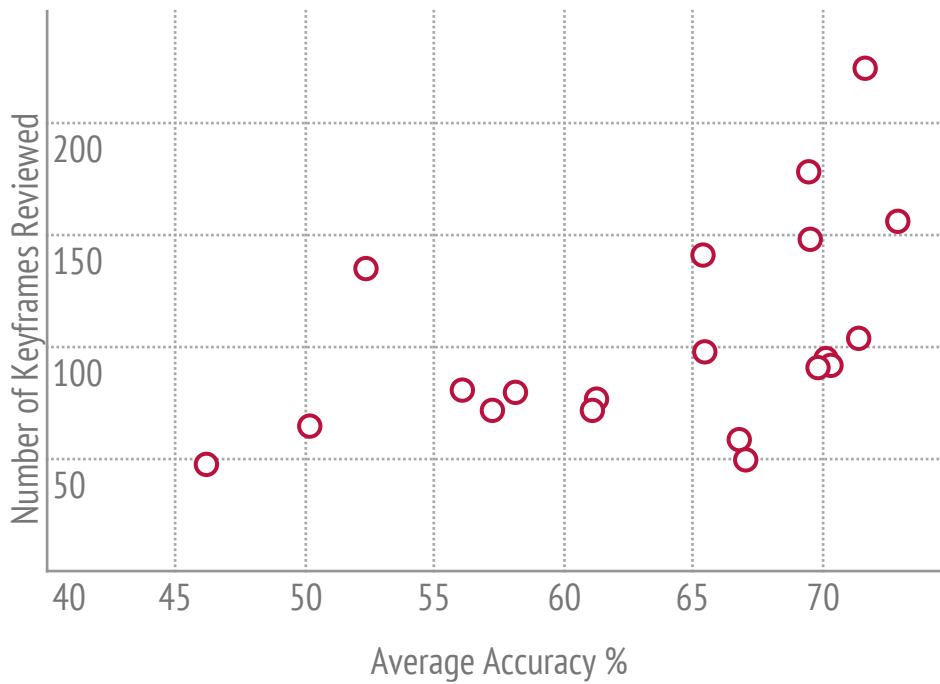
**Figure 5.2:** Impact of Number of Keyframes Reviewed on Placement Accuracy

Feature Points in any given environment, with a strong grounding in the already existing co-ordinate system. This capability had been demonstrated to the participants of this experiment in a training session, and they were encouraged to use it where necessary to improve the accuracy of their tag placements.

Table 5.5 shows the average number of Virtual Points created for this purpose across all of the participants. As with the exploration of participants' behaviour when considering the impact of reviewing multiple Keyframes, trends occurred within this data. Some of the participants in the experiment demonstrated a tendency to prefer using Feature Points that already existed in a scene if they were close to the target object. In some cases this resulted in very low levels of accuracy when compared with the participants who created several Virtual Points to home in on the centre of the object as requested in the

task.  The mean number of points created at 34.15 gives an average of 3.42 points per object within the environment.  However it is important to note that participants who created the most points to use in this experiment were often doing so by starting to tag an object in one frame and then realising that they could not achieve the desired results before moving on to a different Keyframe and starting again.

**Table 5.5:** Average Number of Virtual Points Created

| Min. Number | Max. Number | Mean | Standard Deviation |
|---|---|---|---|
| 17 | 52 | 34.15 | 10.74 |
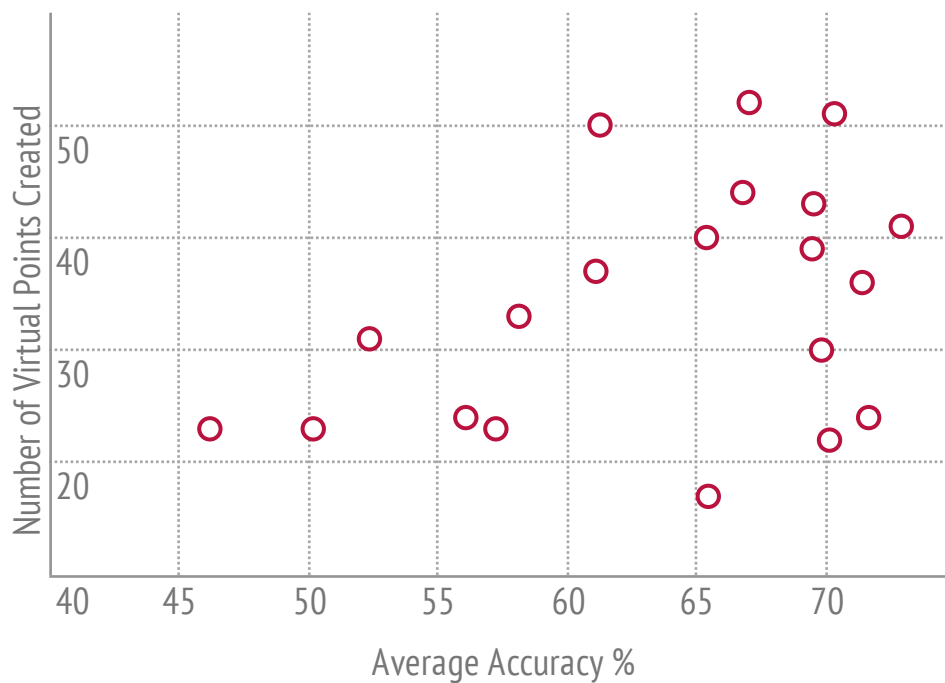


**Figure 5.3:** Impact of Number of Points Created on Placement Accuracy

Figure 5.3 shows the average accuracy for each participant plotted against the number of Virtual Points which they created in the experiment.  There is a weak towards a

positive correlation between the two variables, although not one as clear cut as that seen when considering the impact of number of Keyframes on accuracy (Figure 5.2 Section 5.3.2)). The Spearman Correlation for this data set also demonstrates this fact scoring .332 p = .152. While this a positive correlation can be observed on the dataset, it is not statistically significant. The data suggests support for the third hypothesis laid out for this experiment, though it cannot be said to be fully in support:

> **HP1.3:** Users who create more Virtual Points will have a higher level of accuracy.

## 5.3.4    Investigating Whether The Participants Total Time Impacted Placement Accuracy

The participants were advised that an upper time limit of 20 minutes would be enforced on the experiment, though they were given no other indication of how long they should spend on the task. The time taken between participants was largely dependant on the approach that they took with regards to reviewing large numbers of Keyframes or creating Virtual Points as discussed with the previous two result sets.

**Table 5.6:** Average Time Taken by Participants

| Min. Secs | Max. Secs | Mean | Standard Deviation |
|-----------|-----------|--------|--------------------|
| 413 | 963 | 695.25 | 161.98 |

Table 5.6 shows the maximum time to 963 seconds (16 minutes 3 seconds) which is comfortably below the imposed time limit, as such no participant had to be excluded from the results set for exceeding the time limit. With a mean time of 695.25 seconds (11 minutes 35 seconds) this indicates that the participants spent a little over a minute (69.5 seconds) on each object. In practice however this time was divided into searching for each target object within the scene and then tagging it, so it would be incorrect to say that this time was wholly spent on tagging positions.

Figure 5.4 shows the average time taken in seconds for each participant plotted
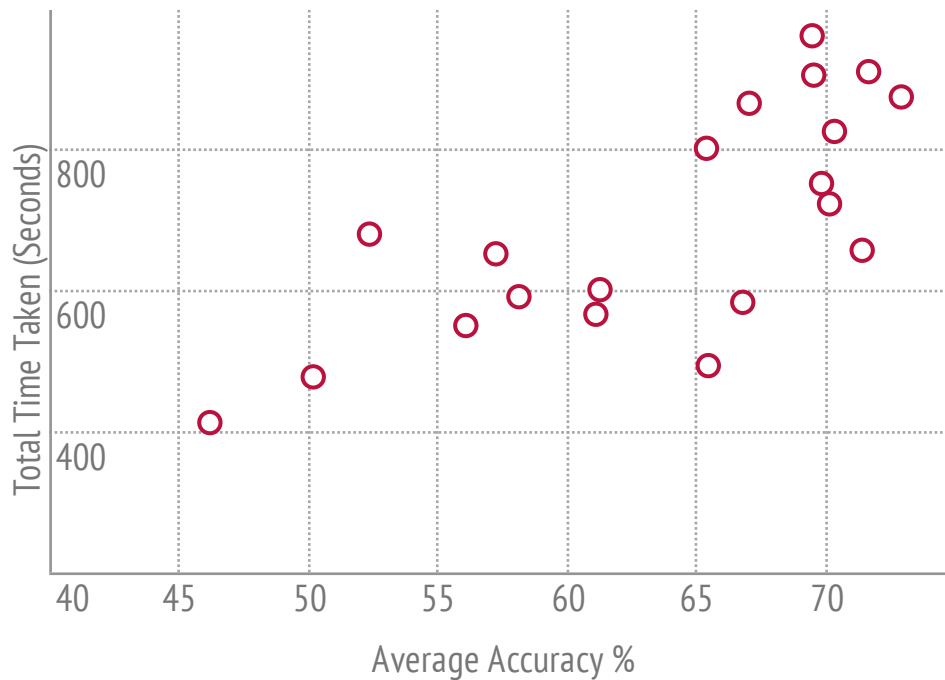
**Figure 5.4:** (Impact of Time Taken on Placement Accuracy )

against their overall average accuracy score. There is a very strong positive correlation between the variables, clearly showing that the participants who spent longer on the experiment achieved a higher level of accuracy. The Spearman Correlation for these results was .728 $p < .01$. Therefore this data can be said to be fully in support the fourth hypothesis laid out for this experiment:

> **HP1.4:** The longer a participant takes to complete the experiment, the higher their overall accuracy will be.

The participants who achieved the highest level of accuracy here were the ones who took the longest time to complete the experiment. This was predicted due to the fact that these participants spent more time exploring the environment and determining which Keyframes to use, and how to best create new Virtual Points. The overriding indication

from the data is that in almost every case increasing the speed of content placement reduces the accuracy. This result is of interest when considering the accuracy of object placement via the KFT system and the PTAM benchmark, and the thesis will now go on to test the impact of time on accuracy in both systems with an expert user.

## 5.3.5    User Satisfaction

In addition to collecting the quantitative data presented in this chapter, users were also asked to complete a post-session questionnaire detailing the level of satisfaction they had for each tag placement. The users were asked to rate their satisfaction on the following scale:

1.  Not at all where I wanted the content (*Very Unsatisfied - 1*)
2.  Not where I wanted the content (*Unsatisfied - 2*)
3.  In the correct area (*OK - 3*)
4.  Where I wanted the content (*Satisfied - 4*)
5.  Exactly where I wanted the content (*Very Satisfied - 5*)

While completing this survey, the users were allowed to review their placements, but were not allowed to make any further changes. This was in order to ensure they were not simply answering the survey from memory, but by considering each tag placement individually.

**Table 5.7:** Summary of User Satisfaction Scores

|                                      | User Satisfaction |
| ------------------------------------ | ----------------- |
| Lowest Score (over 10 objects)       | 3.32              |
| Mean Score (over 10 objects)         | 4.15              |
| Highest Score (over 10 objects)      | 4.9               |

The figures in table 5.7 show that even the lowest average satisfaction score awarded by a participant across all ten objects was above the middle "OK, in the correct area"

option, at 3.32. This suggests that there was an acceptable level of satisfaction across all participants and across all of the objects. This is supported by the fact that the overall average satisfaction was 4.15, falling in the second highest category. Considering satisfaction ratings on an individual object level, two objects shared the lowest rating (3.75). Interestingly while candidates consistently flagged these objects as the most problematic, they did not have the lowest overall accuracy scores in the collected quantified data.

One of these objects was an A4 ring binder folder, laid flat on the table. This object provides the perfect test bed for the KFT approach, as it has a large flat surface which yields few Feature Points from the underlying tracking software. Therefore, participants must create Virtual Points in order to position content in the center of the object. One explanation for the low satisfaction rating is offered in a comment from a participant; *"The angle of the folder in the picture made it hard to tell where the center was"*. This highlights one of the issues KFT set out to solve. In these circumstances users can review multiple Keyframes to chose a more suitable angle, and while this was stressed to the participants, the decision to do so ultimately rests with them. One participant demonstrated understanding this process when commenting about a difficulty they had with a dense population of Feature Points; *"When the dots were too close together, I struggled to click on the one that I had determined to be the best one. Had to go to a different photo to recreate the point."*

The majority of comments on problems arose from smaller objects where the Feature Points were more densely concentrated; *"When there was already a collection of points it made it rather difficult to select"*, *"I had problems making selections when two points were very close together"*. From these and similar comments it was also discovered that when a participant tried to create a Virtual Point with two Feature Points that were very close together the click registration of the system sometimes prevented them from accurately selecting two points within a tight cluster, and is an aspect of the system which could be improved.

In terms of exploring the environment, 12 of the participants provided comments supporting the approach. These comments largely focused on the selection of suitable

Keyframes, such as; *"Easy to explore the room from one photo to the next"*, *"Having dots on the small images makes it easier to scan through and choose a picture to work on"*, *"Easy to find different angles for each item"*. Only one participant noted an issue with exploring the data; *"Many, many, photos to sort through to find what you are looking for, seems overwhelming"*, though it was mentioned by another; *"some pictures only had a few dots on the screen"*. Despite this, all participants completed the task comfortably within the time limit set, which suggests that this is an issue of user satisfaction as opposed to a hinderance of user performance. The remaining seven participants did not give a comment relating to exploring the environment. This shows that the majority of participants understood the environment recreation offered, and found the provided interface features (such as the thumbnail filmstrip) to enhance the usability of the system. The one negative comment comes despite all Keyframes with less than five detected Feature Points being removed by the software, it could therefore be beneficial to consider giving the user control over this threshold to filter the data set in real time.

The comments and satisfaction ratings derived from the post-session questionnaire provide insight into the participants' opinion of the software performance alongside the quantifiable data discussed in this chapter. The critical comments gathered regarding the system relate to areas where improvements could be made within the interface, and at no point did the software prevent a user from successfully completing the task. The participants understood the process of selecting a Keyframe and creating Virtual Points, and clearly felt that this approach provided a suitable environment for positioning content, as shown by the average satisfaction rating of 4.15 out of a possible 5.

## 5.4   Chapter Overview

This chapter has presented a study which was designed to test the viability of the KFT approach as a means for introducing content accurately into an Augmented Reality environment. In terms of the hypotheses laid out for this experiment, Table 5.8 gives an overview of whether or not they were proved by this experiment.

**Table 5.8:** Summary of Experimental Hypotheses

| # | Hypothesis | Confirmed? |
|---|------------|------------|
| HP1.1 | The average tag quality from the KFT approach will be "Good" when compared with the live placement benchmark | ✓ |
| HP1.2 | Participant who review more Keyframes will have a higher level of accuracy | ✓ |
| HP1.3 | Users who create more Virtual Points will have a higher level of accuracy | *partial* |
| HP1.4 | The longer a participant takes to complete the experiment, the higher their overall accuracy will be | ✓ |

Through careful examination of the hypotheses laid out at the start of the chapter, the conclusion can be drawn that KFT is a viable means for introducing content. The overall accuracy levels when compared to the PTAM benchmark fell well within the acceptable means outlined as part of this study.

When considering the impact of the time taken to complete the experiment, the results produced the predicted outcome. The longer a participant spent, the more accurate results were produced. In order to fully explore the effects of this, another experiment will be discussed in the next chapter to find how the accuracy in KFT compares to that in the PTAM system.

Some qualitative data has been provided along side the experimental results, collected in post-session questionnaires. While largely subjective, the comments and placement satisfaction ratings provide an insight into how participants viewed the performance of the software, and also highlighted possible areas for improvement in the overall usability of the interface.

# 6

# Experiment 2: KFT vs PTAM for Accurate Object Placement

## 6.1 Introduction

Having proved that the KFT system is capable of allowing users to accurately deliver content into an Augmented Reality environment, this chapter will discuss the evaluation of the KFT approach against an already existing means of content delivery. In order to evaluate the accuracy with which users placed content using the KFT system in Chapter 5 the users' results were compared against a benchmark created in the live PTAM environment by an expert user. In doing so, a comparison was made between a normal user in KFT and an expert user in PTAM, the fact that the resulting accuracy could be categorised on average as "Good" shows the utility of KFT in content delivery. However in order to fully explore the utility of it KFT against the PTAM benchmark, an experiment was devised in which an expert user in both systems performed the same task before comparing the results. This allows conclusions to be drawn not only on the comparative accuracy of the two systems, but also on other metrics such as the time

taken to accurately tag an environment.

By taking into consideration the comparison of KFT against an already existing system in this way, this study was designed specifically to provide an answers for the research questions identified in the introduction of this thesis. Table 6.1 shows the question under consideration for this experiment.

**Table 6.1:** Research Questions Addressed in Chapter 6

| RQ | Research Question |
| --- | --- |
| RQ3 | Can the proposed content delivery method be used under time pressures to place content while maintaining an acceptable level of repeatable accuracy? |

## 6.2   Study Design

As with the previous experiment, the focus of this study was on the accuracy with which content can be placed, rather than the actual content, which is irrelevant here. In order to compare the accuracy of content delivery in PTAM and the KFT approach, an expert user performed a tagging task using both systems. The task involved tagging 10 unique objects in an environment repeated across four different environments. Each environment was also tested under two time constraints. This resulted in a dataset of 10 tags for each environment for KFT (Long Trial), KFT (Short Trial), PTAM (Long Trial), and PTAM(Short Trial). The long trials were conducted with 16 minutes allowed for the task completion as it represented the longest amount of time taken by any participant in the user study, to the nearest minute. The short trials allowed 8 minutes for the task completion to allow for evaluation of the impact of reducing the time available by half.

### 6.2.1   Hypotheses

The experimental hypotheses listed below were derived from the Research Question to be addressed by this experiment as can be found in Table 6.1:

*HP2.1:* The overall average tagging accuracy in each trial will not be as accurate with KFT as with PTAM.

Due to the fact that the tag placements in PTAM are observable in real time, in place within a live environment there is more scope for the user to examine the specifics of their placement. By being able to move around the object and interact with it in real time a more accurate placement will be achieved than with KFT alone.

*HP2.2:* The KFT approach will allow for a "Good" level of tagging accuracy in both the long and short timed trials.

Despite the fact that the KFT is not expected to achieve as high a level of accuracy as those found with the PTAM system, it is predicted that the KFT approach will allow an expert user to achieve a "Good"[1] level of accuracy regardless of whether used for the long or short trial.

*HP2.3:* The imposition of a stricter time penalty (short trial) will have more of an impact on the attainable accuracy of the PTAM system than that of the KFT system.

The exploration of the live environment for content placement in PTAM offers a faster rate of target object identification by its nature. However, manipulating the positioning of the content takes more time due to the need for a user to view the adjustments from multiple angles in the environment to confirm their placement.

## 6.2.2   Technologies

Both the KFT and PTAM software were used in order to undertake this experiment. Both systems were run on the same machine in the interest of ruling out any performance issues. The hardware used was a Macbook Pro (2.7Ghz i7). In order to facilitate the exploration of the PTAM environment, and the initial mapping of it a Head Mounted

---

[1] The accuracy ratings as defined for this experiment are shown in Section 5.2.3.

Display was also used. The specific choice was a Vuzix iWear VR920 HMD with a resolution of 1024x768 which is above the operating resolution of PTAM, this was paired with a Unibrain fire-i wide angle camera.

Data for this study was performed using logs and generated data files from both KFT and PTAM. Timing was conducted on a separate stop watch with an alarm to alert the user when to stop using each tool.

### 6.2.3   Method

PTAM was again used as the tracking and mapping system for this experiment, and a detailed map was created for each of the four environments. These four environments consisted of a large table top covered with items that could be reasonably expected to be found in a office working environment. The tables were all the same size, and had the same number of objects placed upon them in the interests of fairness.

Once this map had been created and locked from providing any further tracking and mapping data, and as with the previous experiment the live environment tools provided by PTAM (as outlined in Section 2.4.2) were used in addition to the manual manipulation of the xml co-ordinate data by an expert user in order to place "perfect" examples of object center and boundary tags. This benchmark map then formed the basis of comparison for all tests in this study.

The participant for the study was an expert user in both the PTAM and KFT interfaces, and as such was deemed to have similar familiarity with each, removing any preference bias. The instructions for the task were to place a marker at the centre of the top surface of each target object. The user was aware of the time restriction for each trial and was encouraged to take the full amount of time available reviewing and modifying the placement of each object. This was an important feature of the experimental design as it was crucial to the ability to test the impact of time on accuracy.

### 6.2.3.1   Controlling Other Variables

It was crucial to ensure that no modifications were made to the office environment between making the original map, which KFT would use, the benchmark PTAM maps against which the trials would be judged and the live trials taking place within the PTAM environment. In order to ensure that this was the case, the trials were undertaken in a heavily controlled environment to which no admittance other than the participant was allowed until the end of the experimentation. This meant that nothing could be added, removed, or modified in respect to the positioning of objects in the environment that may undermine the PTAM tracking.

In addition to controlling the physical positioning of objects in, and the access to the environment other environmental properties were controlled. In previous trials and test studies with PTAM the sensitivity to changing light conditions have caused issues with the stability of the tracking algorithm over the course of a day using a prepared map that has been locked to prevent any further tracking from taking place. In order to ensure that this was not a factor in the results of this study all natural light sources were obstructed and only artificial light was used for the study.

Should either of the pieces of software fail at any point, a new environment would have be created and the test repeated within that environment to ensure no extra time is afforded in any sense to either piece of software. The evaluation of the results would also take part independently of the PTAM environment using programatic means.

To remove any influence of the ordering in which these trials were conducted, the ordering shown in table 6.2 was used. This ensured that no advantage would be gained by overfamiliarity with one environment giving a benefit to the system used, or affecting the performance under either time constraint.

**Table 6.2:** Experiment Ordering for Each Trial (Environment - Time - System)

| T | | | | |
|---|---|---|---|---|
| 1 | E1-SK | E4-LK | E3-LP | E2-SP |
| 2 | E2-LK | E4-SP | E3-SK | E1-LP |

| T | | | | |
|---|---|---|---|---|
| 3 | E3-SP | E2-LP | E1-LK | E4-SK |
| 4 | E4-LP | E2-SK | E1-SP | E3-LK |

## 6.3    Experiment Results

The purpose of this study was to compare the performance of KFT against the already existing solution (PTAM was chosen in this case) in terms of accuracy of object placement. It also examined the impact of time on the accuracy in each system, to see whether KFT provides the time savings as predicted. The results for accuracy were considered on the same scale as used in the previous experiment, shown below in Table 6.3

**Table 6.3:** Accuracy Rating Scale for Participants

| Distance From Centre | Accuracy | Accuracy |
|---|---|---|
| 0-25% | 100-76% | Perfect |
| 26-50% | 75-51% | Good |
| 49-75% | 50-26% | Acceptable |
| 76-100% | 25-1% | Poor |
| Out of Bounds | 0% | Failure |

**Table 6.4:** Comparison of KFT and PTAM Placement Accuracy - Long Trial

| Environment | KFT | PTAM |
|---|---|---|
| 1 | 72.26% | 80.24% |
| 2 | 77.58% | 78.69% |
| 3 | 74.20% | 77.47% |
| 4 | 75.49% | 81.94% |
| Average | 74.88% | 79.59% |

The figures in Table 6.4 show that working to a 16 minute time limit across four environments PTAM is the more accurate system, $F(1,78) = 16.20$, $p < .001$. The average

outcome of the PTAM accuracy results at 79.59% places it within a "Perfect" rating on the accuracy result scale. However, the tagging performed with the KFT system achieved 74.88% rating it as "Good" as was the overall case in the user study (Table 5.3). The overall accuracy differential from KFT to PTAM was small, at -4.71% and as such assumptions can be drawn that the two systems performed at a similarly acceptable level under the 16 minute time constraint.

**Table 6.5:** Comparison of KFT and PTAM Placement Accuracy - Short Trial

| Environment | KFT | PTAM |
|---|---|---|
| 1 | 64.50% | 52.49% |
| 2 | 60.01% | 53.65% |
| 3 | 65.78% | 57.01% |
| 4 | 61.47% | 51.52% |
| Average | 62.94% | 53.67% |

Table 6.5 show the accuracy figures for the experimental trials which took place with an 8 minute time limit. In this case, the results show that the KFT system was more accurate when working under the increased time pressure, $F(1,78) = 33.60$, $p < .001$. KFT retains a "Good" rating on the accuracy scale shown in Table 5.3, while PTAM drops from a "Perfect" rating in the long trial, and attains a lower "Good" score than the KFT system. The overall accuracy differential from KFT to PTAM was +9.27%, a similar sized difference to the long trial but in the opposite direction.

**Table 6.6:** Summary of Trial Length Impact on Placement Accuracy

| Trial Length | KFT | PTAM |
|---|---|---|
| 16 minute | 74.88% | 79.59% |
| 8 minute | 62.94% | 53.67% |
| Accuracy Diff. | 11.94% | 25.92% |

Observing the accuracy difference when halving the time available to the user across both systems shows a stark contrast. Table 6.6 shows that while KFT accuracy dropped

11.94% ($t$(39) = 17.88, p < .001), PTAM saw a much larger 25.92% drop ($t$(39) = 8.73, p < .001).  In both of these cases, statistical significance was demonstrated by observing the before and after effect of imposing a time constraint.  The face that both results provide p < .001 shows that the null hypothesis may be rejected.  Observing this result in the context of the accuracy differences shown in Table 6.4 and Table 6.5 this demonstrates that while PTAM achieved a higher level of accuracy on the longer trial, the advantages of the system are drastically hampered by the imposition of a time penalty. When considering these results in light of the hypotheses laid out at the start of this chapter, there are mixed results.

*HP2.1:* The overall average tagging accuracy in each trial will not be as accurate with KFT as with PTAM.

PTAM was expected to be more accurate than KFT in both cases, this was not found to be the case.  Despite the fact that the user can explore the content placement from multiple angles when using the PTAM system, the imposition of a much stricter time restriction caused this to become a hindrance, and in the short trial the KFT system out performed PTAM. However, PTAM was the more accurate system with the longer more relaxed trial.

*HP2.2:* The KFT approach will allow for a "Good" level of tagging accuracy in both the long and short timed trials.

As predicted, despite the fact that KFT was less accurate than PTAM in the longer trial, the system managed to yield results in all cases that were considered "Good" on the accuracy scale shown in Table 6.3. Though not reflected in the accuracy scale, KFT performed at a higher level than expected in the longer trial, with the average accuracy only 1.12% under the threshold for a "Perfect" rating, as was achieved by the PTAM system. This high performance meant that despite suffering a 11.94% accuracy hit in the shorter trial, the overall result of 62.94% was still comfortably within the bounds of a "Good" rating.

***HP2.3:*** The imposition of a stricter time penalty (short trial) will have more
of an impact on the attainable accuracy of the PTAM system than that of
the KFT system.

The third hypothesis predicting that the time restriction would have a larger impact
on the PTAM system was also supported. An accuracy reduction of 25.92% as opposed
to 11.94% for KFT clearly shows that the time available to the user has more of an
impact on the PTAM system than the KFT system.

## 6.4   Chapter Overview

This chapter introduced a study designed to test the impact on the overall accuracy of
object placement achievable in KFT when compared to an already existing system (in
this case PTAM). When tested for two different lengths of time, the KFT approach
demonstrates that it is the more accurate system when under time pressures. In terms of
the hypotheses laid out for this experiment, Table 6.7 gives an overview of whether or
not they were proved by this experiment.

Table 6.7: Summary of Experimental Hypotheses

| # | Hypothesis | Confirmed? |
|---|---|---|
| HP2.1 | The overall average tagging accuracy in each trial will not be as accurate with KFT as with PTAM | ✗ |
| HP2.2 | The KFT approach will allow for a "Good" level of tagging accuracy in both the long and short timed trials | ✓ |
| HP2.3 | The imposition of a stricter time penalty (short trial) will have more of an impact on the attainable accuracy of the PTAM system than that of the KFT system | ✓ |

The results presented here show KFT to be a viable approach for quickly tagging
environments with an acceptable level of accuracy, something which is outlined as an
important factor when considered in the context of the scenarios outlined in Section

3.3.

# 7

# Experiment 3: AR Environment Object Selection Ambiguity - Using KFT

## 7.1   Introduction

Having shown that the KFT interface is capable of providing users with a means of placing content into an Augmented Reality environment with an acceptable level of repeatable accuracy, a further experiment was conducted to examine the ability of users to recognise the positioning of content within a live Augmented Reality environment. An experiment was designed to ascertain whether users could reliably identify physical objects highlighted by a virtual "Spatial Marker", and whether the size of that marker had any impact on the correct identifications made by the user. This was examined by having the users explore live environments which had been pre-populated with Augmented Reality content in a scenario such as that described in Chapter 3 Scenario SN1 (Section \ref{sn1}). The pre-population of the environment was carried out using the KFT tool in order to collect data on the accuracy of content placement achievable with the tool. While the experiments presented in Chapters 5 and 6 show that a level of accuracy was

achieved in comparison to a benchmark, this experiment explores the performance of the map generated by KFT.

This study was designed to provide answers to the remaining Research Questions as outlined in Chapter 1. Table 7.1 shows the questions which were taken into consideration.

**Table 7.1:** Research Questions Addressed in Chapter 7

| RQ | Research Question |
| --- | --- |
| RQ4 | Can users reliably identify physical objects in an Augmented Reality environment which are highlighted by a virtual Spatial Marker? |
| RQ5 | Does the size of a Spatial Marker object relative to the physical object it is highlighting have an impact on the number of correct identifications given by users? |

## 7.2   Study Design

This study concentrates on a user's navigation of a live environment, observing Augmented Reality content which has been placed using the KFT method. The principle aim is for users to observe Spatial Markers which have been introduced to the scene, and report back a physical object in the environment which the Spatial Marker is highlighting. Figure 7.1 shows the model that was used for the Spatial Markers in this study. The model was designed to satisfy the need for a non-verbal cue that has no inherent link to a particular type of object, so as not to bias selection. For this study the different sizes of the marker and whether this affects the level of specificity with which a user identifies the object of interest were also observed. For the purposes of this study the environment constructed was an office scenario (as in Section 3.3.1) with several common office objects placed upon a desktop. The participant was given no further instruction than to select the objects which were being highlighted by the Spatial Markers.

The virtual object that was chosen to represent a Spatial Marker was an upturned cone. This object was chosen as in terms of three dimensions it has a y-axis directional

**Figure 7.1:** Spatial Marker used in the system

implication but its size is proportional around the whole object therefore removing any suggestion of direction on either of the other axes. The same can not be said about a traditional arrow model, or other selections which could be considered to have 'billboard' properties and so are only properly interpretable from the correct positioning and angle along the z-axis. The cone object also lends itself well to being resized proportionally, that is to say the width and height can be increased equally with no alteration to the model. This was important for the study as increasing the height of the marker but not its width would lead to a large marker having a narrow footprint from above, which impacts the visual congestion of the scene.

## 7.2.1 Hypotheses

The following list of hypotheses were derived in order to investigate and provide answers to the Research Questions to be examined within this study, as listed in Table 7.1.

   ***HP3.1:*** *Small Spatial Markers will lead to a faster object identification time,*

>   *and a higher level of selection accuracy than when using Large Spatial Markers,*
>   *due to differing amounts of visual congestion.*

The visual congestion in a scene caused by the use of small Spatial Markers will be considerably less than that of a scene containing large markers. Therefore smaller markers will allow the user to quickly and accurately identify the object in question. Large markers however will require more time and perseverance on the part of the user leading to a longer identification time. The occlusion present due to the large amount of visual congestion will also make it more difficult for the participant to accurately identify which objects are to be selected.

>   **HP3.2:** *Proportionally sized Spatial Markers will provide the highest levels of*
>   *selection accuracy of the marker types, however the time required to complete the*
>   *task will be higher than that for small markers.*

Proportionally sized markers are defined by the footprint area of the object which they are identifying. The marker size is matched to this area resulting in a smaller object having a smaller marker. This will lead to the participant obtaining a higher correct identification score. The time required to complete the task will be increased as some of the markers will be large, and will therefore create more visual congestion in the scene than will be observed when using the small markers. Inversely Proportionally sized markers will perform better in terms of score and time taken than the large ones, though they will be less effective than the small and proportional markers.

>   **HP3.3:** *The use of large Spatial Markers will yield the lowest level of object*
>   *specificity.*

Large Spatial Markers inherently suggest to the user that the object of interest is a group, or complete object as opposed to a single member or part of the whole. This will lead to a lower level of specificity when the user is asked to select objects of interest using these markers.

*HP3.4: Altering the size of the Spatial Marker relative to the object of interest's size will yield a higher level of specificity in object selection, leading to a higher correct identification score than other Spatial Marker types.*

Variable size Spatial Markers suggest a relationship with the object that they are highlighting, and therefore if the sizes are relative the user is more likely to make a more specific choice. Choosing part of an object, or a single member in a group of objects rather than when faced with large markers for example.

## 7.2.2 Technologies

The software that was chosen as the base system was PTAM (Parallel Tracking and Mapping) developed by (Klein and Murray, 2007). The reasoning behind this choice is that PTAM excels at creating Augmented Reality worlds in previously unknown environments. By adopting the methods developed in monocular SLAM technologies (Davison et al., 2007) and Parallel Tracking and Map building (Klein and Murray, 2007) the system can be quickly calibrated for stereo vision using one camera, before creating a virtual map of the environment as the user explores it.

As the initial map will be identical for each participant, tagged using the KFT system by an expert user, the dynamic environment mapping properties of the PTAM system will be disabled for this experiment so as to ensure each participant is operating with precisely the same base map to ensure consistency across all of the trials.

With regards to the hardware requirements of this study, the PTAM software is reasonably intensive on processing power, however the most important factor was mobility in order to allow the participants to explore each environment without feeling constrained by the equipment. In order to facilitate this, a MacBook Pro (2.7GHz i7) was used as the main machine, with an attached Vuzix iWear VR920 Head Mounted Display (HMD). The HMD had a resolution of 1024x768, which when paired with a Unibrain fire-i wide angle firewire camera provided a field of view suitable for exploring an Augmented Reality environment. The equipment setup can be seen in Figure 7.2.

**Figure 7.2:** Hardware Equipment used in the trials

All data recording was conducted using an external video camcorder, as tests with screen recording software introduced noticeable lag to the graphics system. This is unacceptable in Augmented Reality systems, particularly when conducting experiments with new users, as the delay creates a tension between what the user expects to see and what they actually see.

## 7.2.3    Method

The participants for the study were chosen from Durham University staff and students. For this study a group of 16 took part with ages ranging from 20 to 32 years old. 16 participants were chosen with an average age of 23 and 68.75% males to 31.25% females. Participants were not paid for taking part in the experiment. The table (B.5) which shows the experimental ordering can be found in Appendix B. Each participant was required to complete a task in four different environments, which involved them wearing a Head Mounted Display (HMD). The four environments were all typical of the context for this study and consisted of a series of objects placed upon a desktop in dense population[1]. From these, ten were objects of interest and so were highlighted by

---

[1]No object was positioned any more than 10 centimeters away from another. This measure was derived from several dry runs to ensure that the distance between objects did not create an environment

the Spatial Markers. The positioning of the Spatial Markers was determined by an expert user creating delivering the content into the environment by use of the KFT system.

The decision to choose ten objects of interest was based on the optimal number of objects that could be placed in each environment while obeying the dense population measure. In dry runs of the experiment setup, twenty objects were found to be the optimal number. Therefore half of these were selected as objects of interest in order to make sure there were enough non objects of interest in the environment to allow the participant to make identification mistakes. The participant was asked to identify which objects were highlighted by pointing them out with a laser pointer and giving a verbal confirmation[2]. By requiring the participant to speak aloud what they were selecting, the data could be analysed to see the difference between the selection of a stack of books, or the top book of the stack. This is something which could not have been derived from simple pointing alone.

Within the group of ten objects were a mix of single objects, members of a group of objects and objects that had a distinct feature or 'part' which could be picked out. Figure 7.3 shows the range of objects in one of the environments. This allowed the experiment to be designed to demonstrate the level of specificity with which participants interpreted the object selections. For example, a marker could identify the handle of a coffee mug, rather than just the fact that it was a coffee mug. By defining a list of correct answers[3] (Appendix B) for each object in each environment, it was possible to come up with a score for each participant, which would be used in order to judge how well each marker type worked. By controlling how many of each type (single, group member, part) were in the environment, and keeping this constant across the tasks it was possible to evaluate

---

where object identification was too simple.

[2]To give *verbal confirmation*, the participant was asked to report verbally which object they were selecting, in as much detail as possible. See section 7.2.3.2 for more information on the experiment script.

[3]The *correct answer* was implemented by design in the experiment. Each Spatial Marker was specifically targeted to an object, or part of object. After experiment dry runs testing other aspects of this system, the correct answers were explained to the test participant to check whether or not they felt they were realistic as a means of validating the expectations. These test participants were not used in the main experiment.

**Figure 7.3:** Environments - Range of objects set out on four desktops

how well each Spatial Marker type works at the different levels of specificity as well.

A training exercise was used for each participant in order for them to achieve a level of mastery before proceeding to the tasks. This was to make sure that the participant was familiar with Augmented Reality as a concept and the technology being used. This was a simple matching game, where the participant would see four numbered virtual tiles hovering over a desktop and they had to place physical tiles underneath them matching the orientation and the position. The training exercise introduced key concepts for Augmented Reality, such as overcoming the difference in hand eye coordination when a HMD vision system is used, and the fact that virtual objects always occlude physical ones. Once the participant could complete this task in less than 60 seconds they moved on to the trials. This time was chosen based on the amount of time it took an expert to complete the task, while allowing for adjustments to the system for new participants.

## 7.2.3.1   Spatial Marker Properties

For this study the only property of the Spatial Marker that was varied was the size of the model.  Each participant will complete the same task across the four environments and will see a different size of Spatial Marker in each environment.  In any one environment the participant will always see the same property.  The four different Spatial Marker sizes that were used are small (fixed size), large (fixed size), proportionally sized, and inversely proportionally sized.
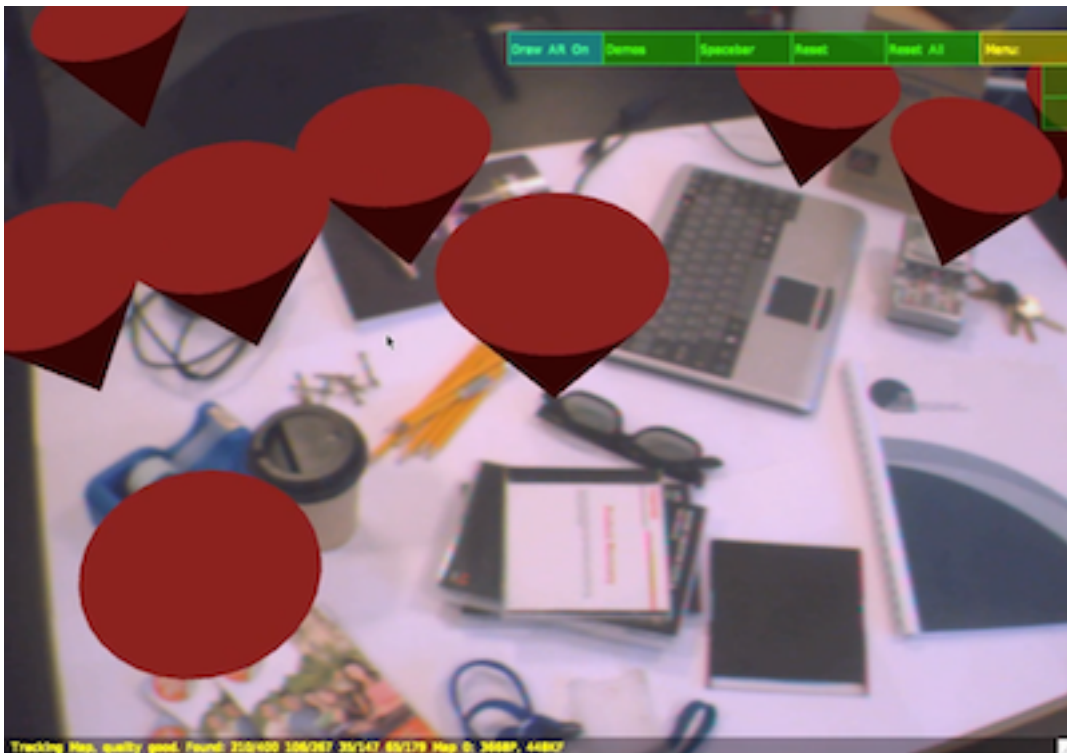


**Figure 7.4:** Spatial Markers - A desktop augmented with Spatial Markers

The small and large Spatial Marker sizes were derived during in the system design phase with several options being explored for each extreme.  The sizes used for the experiment were selected to ensure two factors; 1) The small markers were not so small that they could easily be lost against the visually noisy background of the desktop environment.  2) The large markers were not so large that when used in dense population

the whole environment became totally obscured by the marker models. Once the size of each was determined this was fixed across all of the environments to ensure consistency. An example of the markers augmenting a desktop environment can be seen in Figure 7.4. These sizes were also chosen as the ends of the scale to be used for proportionally sizing Spatial Markers as well. Proportionally sizing the Spatial Markers was done based on the "size" of the object of interest. For the purposes of this experiment, the "size" of an object was determined as being the area of desktop which it occupied in terms of its footprint.

From these measurements a scale was devised, grouping the objects into sizes based on 10 centimetres of their footprint. Each of these groups then had a marker size assigned to it along the scale from the smallest to largest marker size. The 10 centimetre groupings for this scale were derived from tests during the system development phase. It was found that this provided enough groupings to ensure that the markers varied on object size, while the variance was not so small that it was indistinguishable, even to an expert user. This scale was kept consistent across all of the environments and all of the trials.

### 7.2.3.2   Controlling Other Variables

In order to ensure that the only factor which was varied in this study was the size of the Spatial Markers, several other factors had to be observed and controlled to make sure they were not having any influence on the selection of objects by the participants. The factor which had the most potential to influence selections here was the distance of the marker from the object. For each object in each environment the distance between the top most surface of the object and the bottom of the marker was controlled and maintained as exactly the same distance. This was vital to ensure that smaller markers were not closer to the object by some order of relative sizing, meaning that they are easier to interpret than a large marker on the same object.

While the colour of the Spatial Marker is not considered important for this study in terms of conveying meaning, all of the marker had the same matte dark red shade with no lighting effects added. This was chosen for its ability to stand out against the busy

background of the scene, and previous tests with other colours did not allow for as clear distinction. In the event that a participant has colour blindness, the training exercise uses the same colouring in order to establish whether or not the participant has an issue with distinguishing the markers.

In the trials which required participants to interpret large Spatial Markers, the scene was densely populated visually, and therefore high levels of occlusion were present both between markers, and between markers and physical objects. In order to minimise the impact of this, and ensure that the participant could accurately identify all of the objects in question, they were encouraged to freely explore the environment. This was made possible through the use of a laptop and the HMD which made the whole system portable (Figure 7.2), and removed any consideration about the varying heights of participants having an impact on how the scene was viewed.

It was important to ensure that every participant was given the same introductory instructions for using the system. This meant controlling the terminology used particularly when explaining what verbal confirmation was required. No prompting or prior information was given to the participant with regards to how specific they should be, in order to avoid biasing the result towards the more specific regardless of the marker used. This was done by asking the participant to verbally describe "exactly" what they were selecting. This avoided asking for a part of an object, for example, therefore relying solely on the participant's interpretation.

## 7.3 Results

### 7.3.1 Investigating the Differences in Correct Identifications Between Spatial Marker Types

Every participant was awarded a "score" for each task, which shows the number of correct identifications made in each environment with regards to object selection. These correct identifications were measured in respect to the lists detailed in Appendix B, whereby every described object has a prescribed correct identification. The box plot in Figure 7.5

shows the average number of correct identifications a participant made for each marker size. The means and standard deviations of which can be seen in Table 7.2.
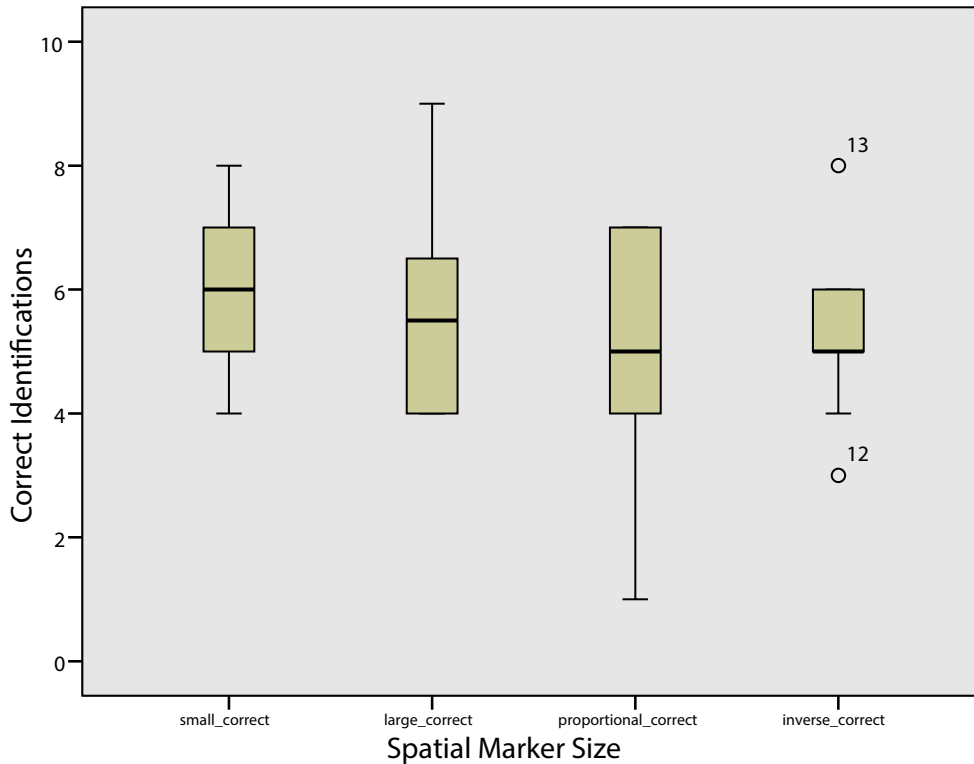


**Figure 7.5:** Correct Identifications by Spatial Marker Size

The box plot of the scores shown in Figure 7.5 shows only a small level of variance in the number of correct identifications for each Spatial Marker type. However, it also shows that the distribution of the data was not normal. The absence of adjacent values for 3 of the columns indicates a weighting of the results towards the ends of the scales. For example, when considering "Large Correct" the results were heavily weighted towards a smaller number of correct answers, with the mean raised through the existence of a small number of higher scores. A Wilks' Lambda multivariate test did not indicate statistical significance, with a value of .74, $F$ (3, 13) = 1.538, p = .25. The fact that the mean values seen in Table 7.2 are so close is another indicator of a lack of statistical significance in

these results.

**Table 7.2:** Descriptive Statistics for Correct Identifications by Spatial Marker
Size

| Marker Size | N | Mean | Standard Deviation |
| --- | --- | --- | --- |
| Small | 16 | 6.00 | 1.37 |
| Large | 16 | 5.56 | 1.59 |
| Proportional | 16 | 5.25 | 1.66 |
| Inverse | 16 | 5.25 | 1.13 |

Figure 7.5 suggests that the participants performed slightly better with the small Spatial Marker type, with almost even distribution of the range of scores either side of the mean. The outliers shown for participant 12 and 13 in Figure 7.5 have not been removed based on further inspection of the data set. They have not been identified as extreme outliers, and a closer inspection of the scores achieved by participant 12 and 13 show that they were not problematic across the three other marker conditions.

One concern with repeating tasks in a short space of time such as this is that fatigue and complacency can set it with participants, which could have had an impact on the amount of effort each person put into these tasks. Figure 7.6 demonstrates that this was not the case with this experiment, as there is no identifiable downward or upward trend across the trials.

Nothing more than an inference of support can be drawn for the first hypothesis laid out for this experiment:

> **HP3.1:** *Small Spatial Markers will lead to a faster object identification time, and a higher level of selection accuracy than when using Large Spatial Markers, due to differing amounts of visual congestion.*

The small Spatial Markers marginally outperformed the large markers, with participants achieving a higher mean correct score in the environments with the lower amounts of visual congestion. While this suggests that the occlusion introduced by having large
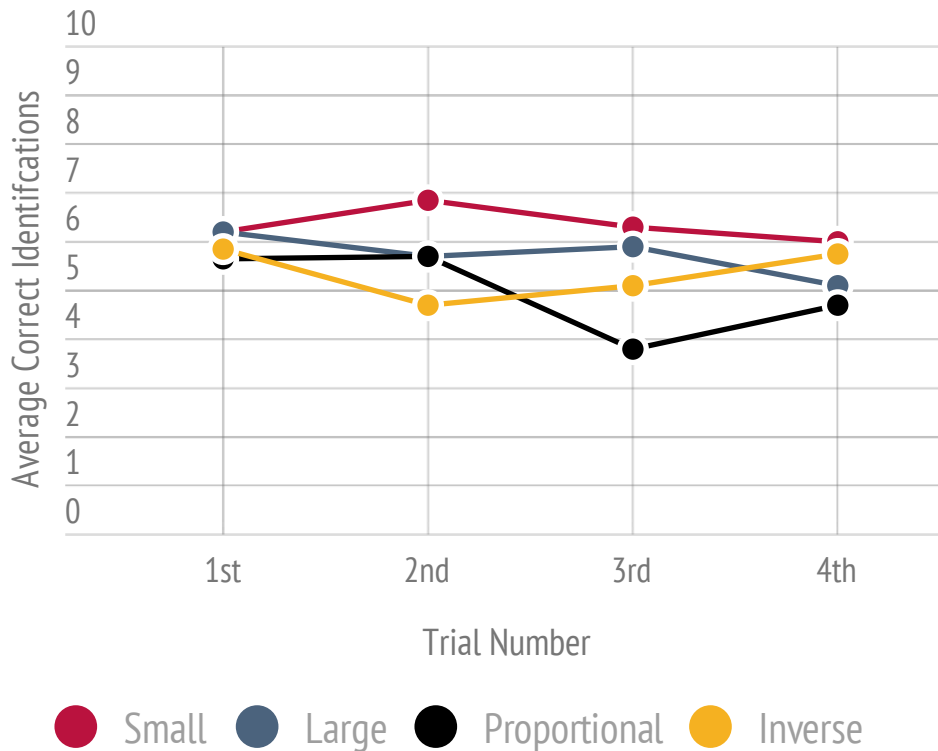
**Figure 7.6:** Effect of Experiment Ordering on Correct Identifications

markers obscuring both the physical objects in the scene, and other virtual marker ob-
jects has an impact on the ability to select objects, there was not statistical support for
this claim. When considering the second hypothesis, these findings are not agreement:

> **HP3.2:** *Proportionally sized Spatial Markers will provide the highest levels of*
> *selection accuracy out of the marker types, however the time required to complete*
> *the task will be higher than that of small markers.*

The proportionally sized markers performed worst overall, tied with the inversely
proportional markers in terms of mean correct score. This suggests that the inherent link
between marker size and object size was not picked up on by the participants in the way
predicted. It was expected that this would yield the highest accuracy in object selection,
despite the fact that there would be a higher level of visual congestion than when using

all small markers. It was also predicted that the inversely proportional markers would perform worse than the proportional markers which again, was not demonstrated to be the case through these results. Due to this it is possible to argue that the disadvantages of large markers, with regards to occlusion, have more of an affect than the advantages of the small markers when considering correct scores.

### 7.3.2 Investigating the Total Time Required Between Pointer Types

When considering the total time required to complete the task across the different marker types, it can be observed that there was a small amount of variance between the different marker types. The mean times and standard deviations for the results can be seen in Table 7.3. Running a one-way repeated measures ANOVA test on these results showed that the Wilks' Lambda test did not provide statistical significance, with a value of .884, $F$ (3, 13), p = .252. Figure 7.7 plots the descriptive statistics, and shows small markers had the shortest mean time required to complete the task, at 1074 seconds. This was considerably lower than the other three, at 1215, 1127, and 1134, albeit with a higher standard deviation (Table 7.3).

The first stage of these experiments was to get the participants to complete a training exercise in order to ensure that all participants were at the same level. This was designed in order to ensure that the first trial that the participant undertook was not their first experience of Augmented Reality, and therefore would not lead to vastly increased completion times. Figure 7.8 shows the effect of the experiment ordering (Appendix B) on the total completion time for each cue size. From this, it is possible to see that there was no impact on the timing created by the ordering. While some of the plots, particularly that of the small marker are quite erratic, there is no general trend which can be applied to all of the conditions. This means it cannot be said that the time required decreased as more trials were completed, or that the time required increased as more trials were completed. When considering the effect of ordering between subjects on the time required, the significance rating of the first trial is .608 when testing the between-subjects effects. Therefore this shows that there was not a statistically significant impact on the results
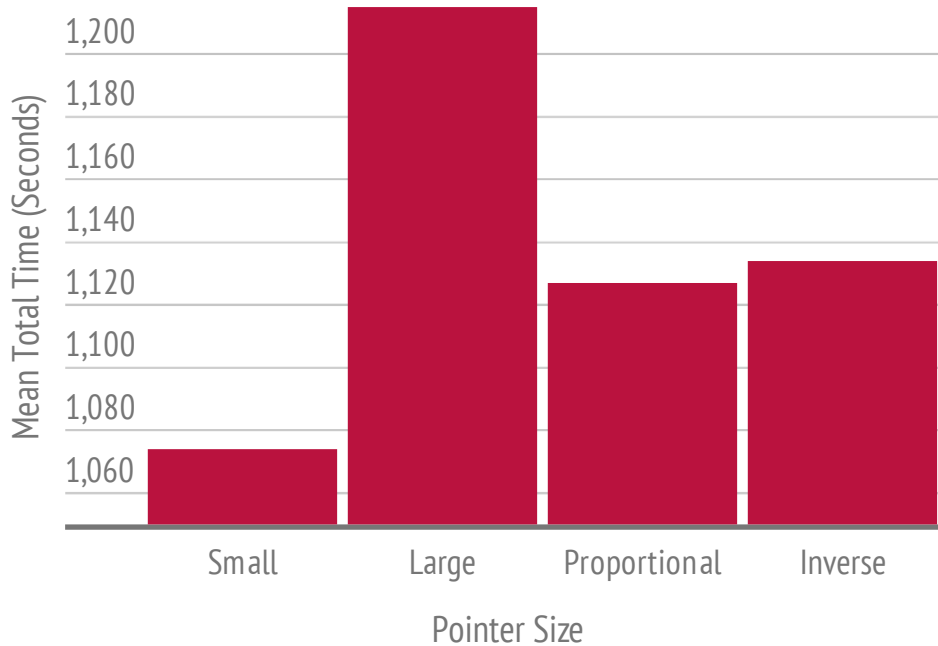
**Figure 7.7:** Total Time Taken for Completion by Spatial Marker Size

based on which trial was completed first.

Though the null hypothesis cannot be rejected the total time taken for each of these marker types tends towards support of the first two hypotheses laid out for this experiment:

*HP3.1: Small Spatial Markers will lead to a faster object identification time, and a higher level of selection accuracy than when using Large Spatial Markers, due to differing amounts of visual congestion.*

*HP3.2: Proportionally sized Spatial Markers will provide the highest levels of selection accuracy out of the marker types, however the time required to complete the task will be higher than that of small markers.*
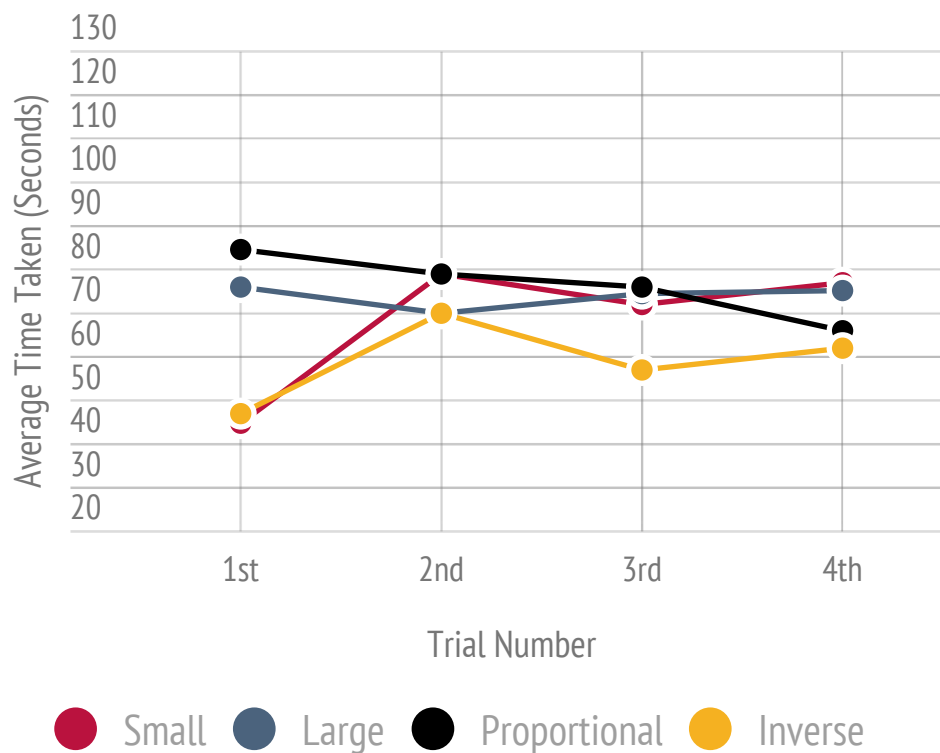
**Figure 7.8:** Effect of Experiment Ordering on Time Taken

It can be seen that the lack of visual congestion provided by the small markers allowed the participants to make the fastest object identifications, with the longest times coming from the large markers. This shows that the participant was forced to spend more time persevering with the scene when only large markers were present. This is supported by the fact that while proportionally and inversely proportionally sized markers were very similar in time required, they were faster to complete than the large marker environments. This suggests that the time saving advantages of small markers had more of an impact than the negative effect on time apparent in the large marker types.

**Table 7.3:** Descriptive Statistics for Total Time Taken by Spatial Marker Size

| Marker Size | N | Mean | Standard Deviation |
|---|---|---|---|
| Small | 16 | 67.13 | 27.89 |

| Marker Size | N | Mean | Standard Deviation |
|---|---|---|---|
| Large | 16 | 75.94 | 24.275 |
| Proportional | 16 | 70.44 | 20.03 |
| Inverse | 16 | 70.88 | 21.75 |

## 7.3.3  Investigating the Impact of Time Taken on Correct Identifications

As well as considering the impact of the cue size on the number of correct identifications, we can consider the impact of the time taken to complete the task on the score given for that task. Figure 7.9 shows the total number of correct identifications across each marker type plotted against the total time required by all participants to complete that task.

It can be seen from Figure 7.9 that the time taken did not generally reflect the correct identifications, with the small marker size clearly providing the most correct identifications with 96, in the shortest total time of 1074 seconds. Contrast this with the large markers which yielded 89 correct identifications in 1215 seconds which was the longest time, and no correlation can be drawn with regards to time impacting accuracy. Once again, it can be seen that both the proportional and inversely proportional marker sizes are clustered closely together and provide similar results both in terms of time taken and correct identification, suggesting that the variable marker sizes gave similar performance regardless of how they were varied.

As can be seen in the first hypothesis laid out for this experiment, it was predicted that small markers would yield both high accuracy, and fast identification times, particularly when compared with the large marker sizes:

> **HP3.1:** *Small Spatial Markers will lead to a faster object identification time, and a higher level of selection accuracy than when using Large Spatial Markers, due to differing amounts of visual congestion.*
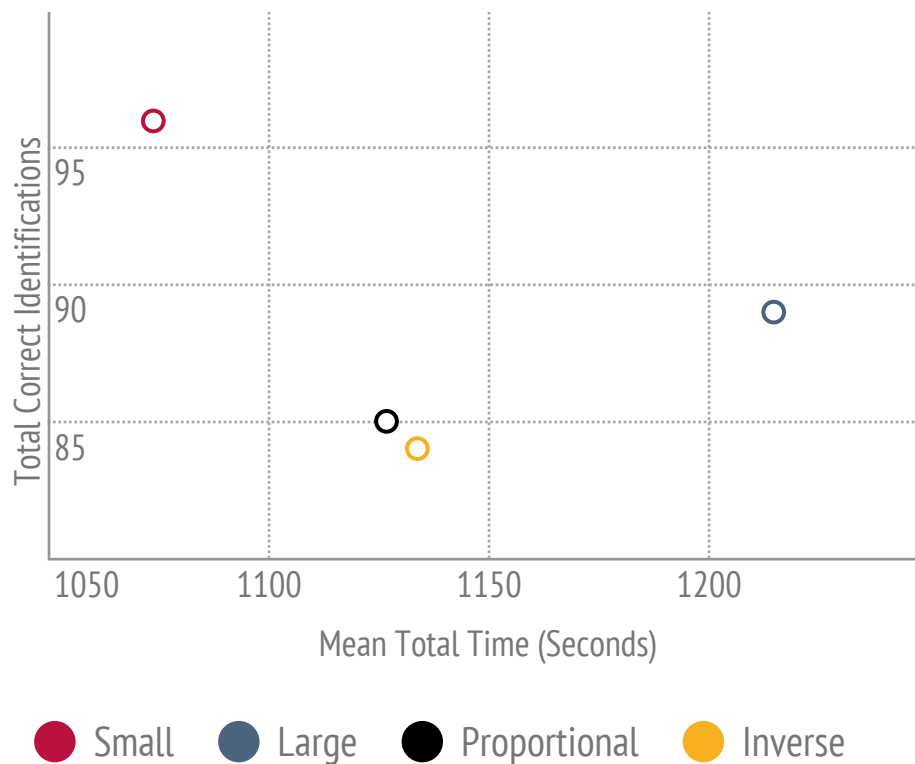
**Figure 7.9:** Total Correct Identifications and Total Time Taken by Spatial Marker Size

This is demonstrated by these results, however the second hypothesis predicted that the proportionally sized markers would give the highest accuracy with a slight time penalty:

> **HP3.2:** *Proportionally sized Spatial Markers will provide the highest levels of selection accuracy out of the marker types, however the time required to complete the task will be higher than that of small markers.*

While the time penalty is correct, and has been demonstrated in Section 7.3.2, the accuracy was not as high as the same task performed with small markers. The reasoning for this has already been explored in Section 7.3.1, however it would suggest that while participants took slightly longer to complete the experiment with these markers, it did

not improve the accuracy. This means that the perseverance required to deal with the visual occlusion caused the participant to take longer over a decision, but this extra time did not allow them to be more accurate with a decision. This shows that the predicted inherent link between object and marker size was not picked up on by the participant. This is compounded when looking at the first hypothesis with regards to large markers. While it is true that they took the longest time to complete, again, the accuracy was also lower than with smaller markers. Once again, here the participant has not taken advantage of the extra time to focus on the accuracy of the marker placement.

## 7.3.4    Investigating the Impact on Selection Specificity by Spatial Marker Size

As outlined in Appendix B, each object in the environment was assigned a level of specificity; Single Object, Group Member, or Part of Object. This was done in order to test the performance of each Spatial Marker size on varying kinds of object. Table 7.4 shows the percentage of correct identifications for each level of specificity.

**Table 7.4:** Percentage of Correct Answers For Each Level of Specificity.

| Marker Size | Single Object | Group Member | Object Part |
|---|---|---|---|
| Small | 100.00% | 53.13% | 43.75% |
| Large | 100.00% | 50.00% | 35.94% |
| Proportional | 96.88% | 50.00% | 32.81% |
| Inverse | 100.00% | 51.56% | 29.69% |

It can be observed that as expected the Single Object correct identifications were at close to 100% for every marker type[4]. In the case of group members, and object parts however, there is a large drop off, with participants more easily identifying group members than object parts. The performance of candidates across all marker types when identifying group members was very consistent, with a slight improvement with small markers. This is also seen when considering the identification of object parts rather than

---

[4]The expectation placed on single object specificity can be found as part of Appendix B.

the object as a whole. Figure 7.10 shows the mean correct identification scores for each object type by marker size.
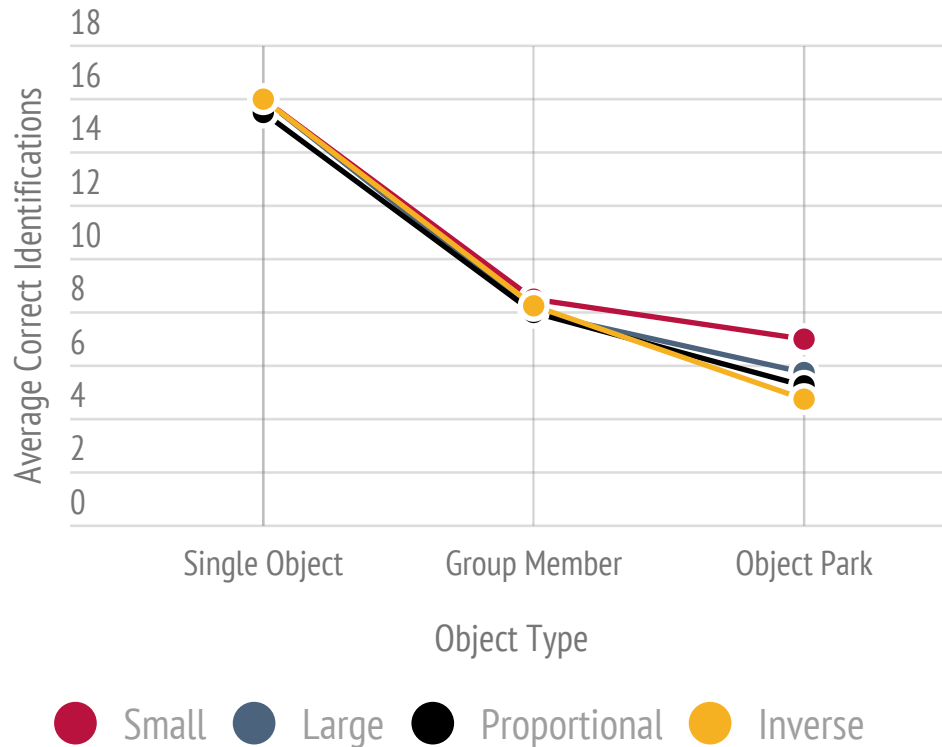


**Figure 7.10:** Mean Correct Identifications for each Level of Specificity by Marker Size

**HP3.3:** *The use of large Spatial Markers will yield the lowest level of object specificity.*

**HP3.4:** *Altering the size of the Spatial Marker relative to the object of interest's size will yield a higher level of specificity in object selection, leading to a higher correct identification score than other Spatial Marker types.*

Hypothesis 3 and 4 predicted the impact of marker size on object selection specificity. Hypothesis 3 deals with large markers, and the prediction that the participant would not

distinguish the levels of specificity when faced with a large marker for every object. This was proved to be the case in this experiment.

Hypothesis 4 predicted that the proportional markers should yield the highest levels of specificity in this experiment. By looking at the figures in Table 7.4 it can be seen that this was not the case. Small markers provided the most correct answers at each level of specificity, showing that the participant has not linked the size of the marker to the selection required.

## 7.4    Chapter Overview

In terms of the hypotheses laid out for this experiment, Table 7.5 gives an overview of whether or not they were proved by this experiment.

**Table 7.5:** Summary of Experimental Hypotheses

| # | Hypothesis | Confirmed? |
|---|---|---|
| HP3.1 | Small Spatial Markers will lead to a faster object identification time, and a higher level of selection accuracy than when using Large Spatial Markers, due to differing amounts of visual congestion | ✓ |
| HP3.2 | Proportionally sized Spatial Markers will provide the highest levels of selection accuracy out of the marker types, however the time required to complete the task will be higher than that of small markers | *partial* |
| HP3.3 | The use of large Spatial Markers will yield the lowest level of object specificity. | ✓ |
| HP3.4 | Altering the size of the Spatial Marker relative to the object of interest's size will yield a higher level of specificity in object selection, leading to a higher correct identification score than other Spatial Marker types. | ✗ |

These results have demonstrated that the participants did not make the cognitive link between marker size and selection specificity that was predicted. One of the main con-

tributing factors to this could be the context of the experiment. These hypotheses were tested in a desktop environment, which typically features a number of objects that are similar sizes. While this provides an excellent boundary case test for the object selection accuracy, it may be the case that the sizes were too close together to be distinguished by new users.

By repeating this experiment in a different environment more drastic results may come forward, or it could be observed that the effect of these results is just found again with more impact. When looking at the overall number of correct identifications regardless of specificity, there was a observable effect demonstrating that small Spatial Markers provided the best result, something that was also reflected across the time taken to complete trials. This suggests that in a desktop environment such as this, the best option would be to provide the user with all small markers. There are scenarios however that would benefit from having more obvious markers that could lead to fast localisation of content rather than identification, and these results will be discussed in context of those scenarios in Chapter 8.

# 8

# Discussion

## 8.1 Introduction

Chapters 5 and 6, introduced the experimental studies which were carried out in order to evaluate the KFT method for content delivery proposed in this thesis. In addition to this Chapter 7 introduced an experimental study which saw participants interact with an Augmented Reality environment that had been populated with content by the KFT tool in order to observe the success of the approach in a real scenario. By identifying a number of different hypotheses designed to test the Research Questions posed in Chapter 1, several areas of Content Delivery for Augmented Reality were tested. This chapter looks at the results presented within these experimental study chapters in the context of the scenarios introduced in Chapter 3, and in terms of the Research Questions outlined as the Criteria for Success for this thesis.

## 8.2 Experimental Results

The Pipeline for Content Delivery in Augmented Reality (Figure 8.1) first introduced in Chapter 4 can be used to show the coverage and relationships of the three experiments

in this thesis. The initial experiment in Chapter 5 examined the accuracy with which a
participant could place content, using the KFT interface. Content positioning is the key
concept to KFT, and was the underpinning principle of this experiment. By comparing
the accuracy with which a participant could place content with a benchmark set by an
expert user, the experiment provides clear data on the success of the KFT approach in this
area. The collected data showed full agreement with Hypothesis HP1.1, which stated
that the average tag quality across all participants would be "Good" (within 51-75%
accuracy), with the average being 63.61%. This data also shows that the recreation of
the environment was of a high enough standard to allow the users to understand and
correctly interpret an environment which they had not seen before, the first key step of
the KFT approach as identified in Figure 8.1.
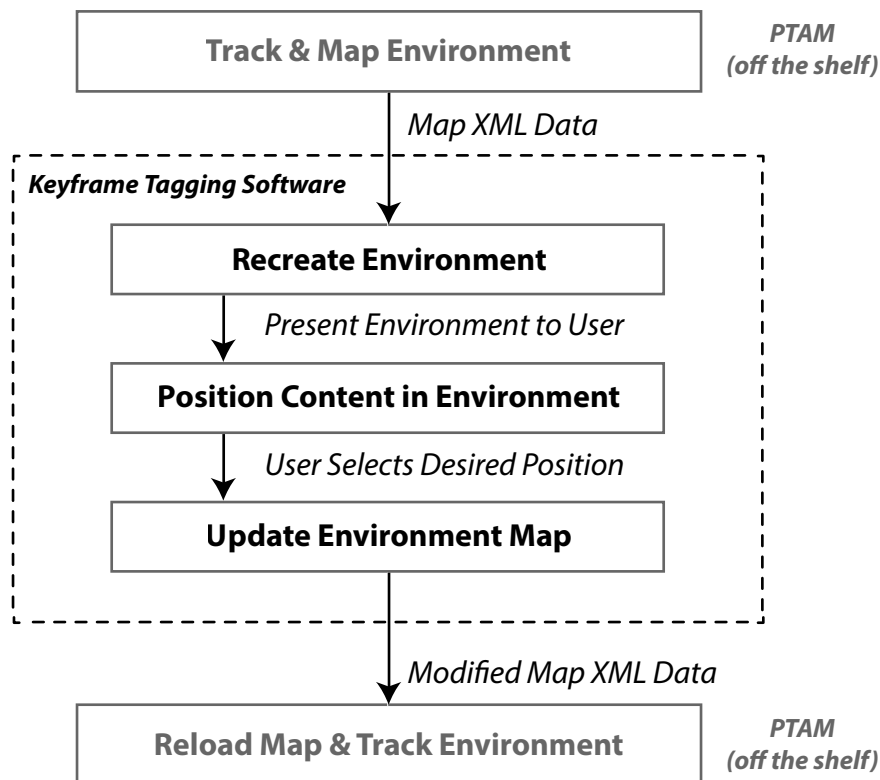


**Figure 8.1:** Content Delivery for Augmented Reality Pipeline

In addition to establishing the accuracy of content positioning against an expert benchmark, Experiment 1 (Chapter 5) offers insight into how choices made by the participant affected their accuracy scores. The results showed that participants taking more time to review several Keyframes performed the best overall, with the highest accuracy score an impressive 71.63%, a score falling only slightly short of the highest possible "Perfect" accuracy score defined for this experiment.

The data collected in Experiment 1 (Chapter 5) positively reinforced the method of environment recreation and content positioning through the use of Feature Points increasing Virtual Points (Chapter 4). Experiment 2 (Chapter 6) was designed to test the performance of an expert using the system compared with an already existing solution (PTAM (Klein and Murray (2007))). While this experiment covered the same elements of the Content Delivery Pipeline (Figure 8.1), it provided data on the potential performance of participants given a level of familiarity with the system. While Experiment 1 (Chapter 5) focused on users who had never used the system before, with an unknown environment, in order to examine the quality of the environment recreation, Experiment 2 (Chapter 6) purely focused on finding the highest possible accuracy within a given time frame. The expert user repeated the same tasks in KFT and the PTAM software, and the results showed the accuracy performance of KFT to be more consistent than PTAM with the introduction of a time restriction. When halving the available time an accuracy reduction of 25.92% in the PTAM system compared with a reduction of just 11.94% clearly shows KFT to be a viable means of content positioning. As with the results of Experiment 1, this is in direct support of the quality of environment recreation and the accuracy of the underlying algorithms for content positioning in the KFT system.

The success of Experiments 1 and 2 in showing KFT to be a viable means of accurate content placement informed the design of Experiment 3 (Chapter 7). Experiment 3 provides data covering a different aspect of the Content Delivery Pipeline (Figure 8.1) in that the participants interacted with the final step of this process "Reload Map & Track Environment", using the PTAM software rather than KFT. Experiment 3 was designed to show whether an environment map tagged by an expert using the KFT interface could

provide accurate navigation information to participants exploring that environment in real time. In order to test this a series of navigation cues were placed into the environment to highlight certain target objects to the participants. The participant then had to specify which objects had been highlighted, allowing the evaluation of the accuracy of object placement.

## 8.3    Addressing the Scenarios

In terms of the proposed scenarios for the operating environments of the KFT system developed in this thesis, a number of interesting problems arose. Scenario SN1 described a relatively straightforward use case which would afford the user a great deal of time with regards to content placement, as the environment was largely static and the tagging could be performed without pressure. The investigation carried out in Chapter 5 showed a clear correlation between the amount of time taken by a user positioning content and the accuracy of that content's positioning, which would suggest that KFT used in this scenario could provide a high level of accuracy. With no time pressure on the user, a high number of Keyframes would be able to be reviewed, along with the possibility for a high number of Virtual Points to be created. Both of these factors were demonstrated to have a positive effect on the overall accuracy of content placement.

Scenario SN2 introduced a time constraint along with the need for a high level of accuracy. In terms of affecting a live training environment at the same time that a participant is interacting with it, the timing becomes the key factor. While the results discussed above provide the ideal support for accurate tagging with no time pressures, this is not the case with Scenario SN2. The experiment conducted in Chapter 6 tested the KFT approach against strict time constraints in order to see whether or not there was still an acceptable level of accuracy gained with regards to content placement. The results show KFT to outperform the existing system (PTAM), with a 11.94% drop in accuracy. While this still fell within the "Good" range of our determined scale, in cases where hazardous materials could be the target object, this may need some consideration.

Despite this fact, KFT was demonstrated to maintain a higher level of accuracy than PTAM with the introduction of the time penalty.

In terms of Scenario SN3 it offers some similarities to the conditions presented by Scenario SN2 as it requires real time interaction with a live environment. This means that time pressures are still very much a factor. However, due to the fact that the proposed operation is in support of Emergency Response highlighting potential dangers or locations of interest, there is actually a more relaxed requirement of accuracy. That is not to say that a low level of accuracy would be accepted, but the importance of achieving "Perfect" as opposed to "Good" is not as crucial. As was demonstrated in the Object Selection Ambiguity experiment presented in Chapter 7 participants were able to reliably correctly identify objects using Spatial Markers in an environment that had been tagged using the KFT system. Despite the fact that in Chapter 6 the expert user with the KFT system never achieved a result which could be considered "Perfect" on the proposed accuracy scale, even when applying time constraints the level of accuracy remained at least "Good" for the KFT system which has some meaning with regards to Scenario SN3. As long as an Emergency Response Team member could understand the positioning of the Spatial Marker, it is more important to get the content into the environment quickly than it is to take concern with slight differences in accuracy.

## 8.4   Answers to Research Questions

This thesis presented Criteria for Success in the form of a series of Research Questions that should be answered by the experimental methodology. This section examines each of these Research Questions individually, drawing conclusions from the results presented within the experimental chapters of the thesis.

### 8.4.1    RQ1:  Can a user place content into an Augmented Reality environment using a photograph based reconstruction of that environment?

By developing the Keyframe Tagging software tool as part of this thesis, a means was provided for users to place content into an Augmented Reality environment using a photograph based reconstruction of that environment.  In accordance with the set of requirements outlined in Chapter 3, the implementation focused on how an environment can be meaningfully recreated from a series of photographs and co-ordinate data (Chapter 4).

The fact that results were obtained for the experiment examining KFT Object Placement Accuracy in Chapter 5 answers this question.  It was possible to develop a system based on the set of requirements outlined in Chapter 3 and implemented as described in Chapter 4 in order to allow users to place content into an Augmented Reality environment using a photograph based reconstruction of that environment.

While KFT does not deal with the creation of the environment, or the display of the content back in the live environment, it has been demonstrated to be able to parse maps created by the chosen tracking and mapping system (PTAM) in a non-intrusive manner, such that any additional content injected by KFT does not corrupt the map, and functions when loaded back into the tracking and mapping system.

### 8.4.2    RQ2: Is content placed using a photograph based reconstruction of the environment positioned with an acceptable level of repeatable accuracy?

While the answer to RQ1 has ascertained that it is possible to place content into an Augmented Reality environment using a photograph-based reconstruction of that environment, this is of little practical use without measuring the accuracy with which this content could be placed.  In order to test the usefulness of the KFT approach, the experiment outlined in Chapter 5 required participants to place content items into an

unknown environment with a repeatable level of accuracy. The results from this experiment show that the average accuracy score that was achieved by the participants was 63.61%. This result converts to a "Good" rating on the scale outlined in Section 5.2.3.

While looking at the average across all 200 tags and 20 participants offers an answer to this research question, looking deeper into the collected data shows that with a lowest score of 46.20% ("Acceptable" on the accuracy scale) KFT provides an acceptable level of accuracy even in the worst case according to the results of this experiment. This is supported by the fact that no tags from the 200 fell outside of the bounds of the target object and thus none were considered to be incorrect.

The experiment outlined in Chapter 5 allowed for data collection to give more than a simple answer to this question however. By examining the number of Keyframes that each participant reviewed, and the number of Virtual Points they created in order to position content, it is possible to build a picture of the factors of the KFT approach which have an impact on the overall accuracy score gained by the participants. The experiment was conducted with no coercion towards a particular technique (e.g. review many Keyframes, create many points) in order to achieve accurate placements. It was found that when participants did review large numbers of Keyframes in order to identify the location for their content placements, they achieved higher levels of accuracy, and a significant positive correlation was identified between these two factors. When looking at the impact of the number of virtual points created, there was a tendency towards a positive correlation, though it could not be classed as statistically significant for this sample size.

These results show that inexpert participants came up with "Good" results. Considering that the two factors which increased the accuracy levels of the participants are factors which could be easily introduced through instruction or training, the potential for this method to attain a higher average accuracy score than that gained in this experiment is clear to see. Therefore it can be said that the KFT approach for content delivery does offer a repeatable level of acceptable accuracy for content delivery in Augmented Reality environments.

### 8.4.3  RQ3:  Can the proposed content delivery method be used under time pressures to place content while maintaining an acceptable level of repeatable accuracy?

Taking the results from Chapter 5 in response to RQ2 into account, it has been proven that all participants accurately placed the content within the environment within the allowed time scale. However, and by design for this experiment, the time scale was not designed to put the participant under any time pressure; The participants were simply afforded an upper time limit which had been determined from pilot studies as a comfortable amount of time to complete the task. This allowed for the collection and analysis of data on the impact of the time taken by a participant on their accuracy score. The results of the experiment in Chapter 5 clearly show a strong correlation between the time taken the accuracy score. This is strongly related to the nature of the task, as the more time spent assessing whether the 2 dimensional representation of a content tag location is where the participant wanted it in 3 dimensional space, then the more likely they were to arrive at a satisfactory conclusion. The same holds for the pre-existing software solution within the PTAM system.

The means of placing content in PTAM requires the user to do so live in the Augmented Reality environment, which has the advantage of them being able to fully explore the 3 dimensional positioning by moving around the object, something which by the nature of the 2 dimensional image based KFT approach is simply not possible for KFT. This offers clear advantages when considering the accuracy of content placement, but it is also time consuming. In light of this a further experiment was carried out, outlined in Chapter 6. The aim of the experiment was to directly address RQ3, and test the performance of KFT under time pressures when compared with the benchmark PTAM software. As predicted within the hypotheses for this experiment the KFT system maintained a "Good" level of accuracy when put under time pressures. However, under strict time penalties KFT was actually found to be more accurate than the PTAM software, which was not predicted and further supports the fact that KFT can allow a user to op-

erate under strict time pressures whilst still maintaining an acceptable level of accuracy.

### 8.4.4   RQ4: Can users reliably identify physical objects in an Augmented Reality environment which are highlighted by a virtual Spatial Marker?

Having proved in Chapters 5 and 6 that the KFT approach is suitable for placing content, a further experiment was presented in Chapter 7. This experiment was concerned with the ability of participants to explore a live (PTAM) Augmented Reality environment and provide feedback on a number of physical items which had been highlighted as target objects by a virtual Spatial Marker. The fact that the participants overwhelmingly selected the correct objects is in direct support of the fact that they can identify objects highlighted by a Spatial Marker.

However in addition to the fact that this experiment proved that participants can use Spatial Markers to identify objects within an Augmented Reality environment, it also provided further validation to KFT as a content delivery method. The trial environments for this task had been tagged by an expert user using the KFT method of content delivery, in order to test participants understanding and appreciation of the positioning of the virtual content. The correct identifications and fast response times which are explained in Chapter 7 offer further validation of the technique by showing that the results of the content placement are actually viable in a real world application.

### 8.4.5   RQ5: Does the size of a Spatial Marker object relative to the physical object it is highlighting have an impact on the number of correct identifications given by users?

The results presented in Chapter 7 show that there isn't a significant impact on the number of correct identifications given by participants when the size property of the Spatial Marker is adjusted relative to the physical object it is highlighting. While these results went against the initial hypotheses of the experiment, they provided an inter-

esting insight into the behaviour of participants with regards to the interpretation of virtual navigation cues. It was expected that the participant would alter the specificity of their answer (e.g. a single pen in a group of pens) when presented with either smaller or proportionally sized markers (as defined in Section 7.3.1). However, the results of the experiment showed that the difference in performance from the participants across differently sized markers were not varied enough to a show a statistically significant impact on the results.

It is likely that the reasoning why there seemed to be a better overall performance for small markers in each category outlined within Table 7.4 to be related to issues of occlusion and general visual congestion of the scene. By using smaller markers, the participant can see more of the physical environment without having to look around the marker, however there is a tradeoff when considering the use of small makers within a larger environment, as the participant may struggle to initially locate the marker before proceeding to the object identification stage.

## 8.5   Chapter Overview

The set of Research Questions were outlined in Chapter 1, against which the success of the proposed approach would be judged. A summary of these questions can be found in Table 8.1

Table 8.1: Summary of Research Questions Addressed in this Thesis

| RQ | Research Question | Considered? |
|---|---|---|
| RQ1 | Can a user place content into an Augmented Reality environment using a photograph based reconstruction of that environment? | ✓ |
| RQ2 | Is content placed using a photograph based reconstruction of the environment positioned with an acceptable level of repeatable accuracy? | ✓ |

| RQ | Research Question | Considered? |
|---|---|---|
| RQ3 | Can the proposed content delivery method be used under time pressures to place content while maintaining an acceptable level of repeatable accuracy? | ✓ |
| RQ4 | Can users reliably identify physical objects in an Augmented Reality environment which are highlighted by a virtual Spatial Marker? | ✓ |
| RQ5 | Does the size of a Spatial Marker object relative to the physical object it is highlighting have an impact on the number of correct identifications given by users? | ✓ |

# 9

# Conclusions and Further Work

## 9.1 Introduction

This thesis has contributed to the domain of both Augmented Reality and Computer Supported Collaboration by introducing novel means of content delivery for unknown Augmented Reality environments (Chapter 3), with the particular aim of satisfying a set of requirements derived from real world use case scenarios (Section 3.3). There are several well established projects which support rapid and ubiquitous content delivery in Marker Based Augmented Reality systems (Haringer and Regenbrecht (2002); Schmalstieg et al. (2002); Behzadan and Kamat (2005)), however these require a certain level of pre-preparation of their environment, most commonly through the addition of known pattern based markers. When considering unknown Augmented Reality environments a excellent work has been conducted into tracking and mapping solutions (Davison (2003); Davison et al. (2007); Klein and Murray (2007)) and with this in mind the aim of this thesis was to marry the two domains.

## 9.2    Research Contributions

By examining the means of adding content into environments in such existing software, a tradeoff was found between flexibility and accuracy.  The PTAM (Klein and Murray (2007)) (and subsequently PTAMM(Castle and Klein (2008))) software provided a high level of flexibility for content addition in an unknown environment, however its limitation is that content is either added programmatically or using an interface while exploring the Augmented Reality in real time. While this allows for a high level of accuracy (Chapter 6) it is impractical for a wide range of the scenarios presented in Section 3.3.

In order to address this tradeoff, a novel means of delivering content into unknown Augmented Reality environments was designed. The Keyframe Tagging approach has a particular focus not only on being able to work with unknown environments (Chapter 3), but also in providing the user with a fast and accurate means of introducing their content.  In doing so a flexible method with a repeatable level of accuracy was created and is presented within this thesis.

The overall approach taken in the Keyframe Tagging method is defined in Chapter 3, with detailed entity relation diagrams to represent the data model which underpins the technique.  The content positioning technique is discussed in Section 3.4.3, with Figure 3.5 presenting a high level overview of the algorithmic process.  This discussion is further expanded in Chapter 4 which details how the Keyframe Tagging software was implemented.  While KFT as a whole is a novel means of delivering content, Section 3.4.3.1 describes an algorithm designed to increase the resolution of available Feature Points upon which a user can place content. This is key to the approach as a whole, as it provides the flexibility for a user to place the content where they wish, while still having a reliable co-ordinate system to ensure accurate placement.

Three experimental studies were designed to test the proposed approach against a set of Research Questions which are outlined in Chapter 1.1. The first tested the accuracy with which novice users could place content within an environment when compared

with an expert benchmark (Chapter 5). This was then developed in the second experiment (Chapter 6) to examine the time vs accuracy tradeoff when used by an expert user in comparison with the existing PTAM software(Klein and Murray (2007)). The final study tested whether participants could reliably identify objects within an environment which were highlighted using a virtual Spatial Marker, in an environment which had been marked up using the Keyframe Tagging software (Chapter 7).

The results from these experimental studies demonstrated KFT to be a viable means of content delivery into unknown Augmented Reality environments. Chapter 8 presents a discussion of these results in the context of both the scenarios considered in the system design (Chapter 3.3), but also in answer to the Research Questions (Chapter 1.1) which the thesis can be judged against.

## 9.2.1   Summary of Research Contributions

Table 9.1 restates the Research Contributions outlined in Chapter 1.2.1, providing an overview of which areas of the thesis deal with each one.

**Table 9.1:** Summary of Research Contributions and Thesis References

| # | Research Contribution | Thesis Reference |
|---|---|---|
| RC1 | Proposal of a novel means for delivering content into an Augmented Reality environment via a recreation of the key features of that environment. | Chapter 3 describes the proposed approach with focus on the recreation of the environment and positioning of content. |
| RC2 | Implementation of the proposed approach to allow for the evaluation of the method. | Chapter 4 outlines the implementation of KFT. |
| RC3 | Quantitative data on the accuracy of object placement achievable by the proposed approach. | Chapters 5, 6 present the quantitative data gathered on object placement accuracy. |
| RC4 | Quantitative data on the interpretation of object placement achieved by the proposed approach in a live Augmented Reality environment. | Chapter 7 contains quantitative data relating to the users' interpretation of object placement within a live environment. |

## 9.2.2   Criteria For Success

Table 9.2 provides an overview of the Criteria for Success discussed in Chapter 1, providing a reference to which area of the thesis deals with each one.

**Table 9.2:** Summary of Criteria for Success and Thesis References

| RQ | Research Question | Reference |
|---|---|---|
| RQ1 | Can a user place content into an Augmented Reality environment using a photograph based reconstruction of that environment? | Chapter 5 Experiment 1 |
| RQ2 | Is content placed using a photograph based reconstruction of the environment positioned with an acceptable level of repeatable accuracy? | Chapter 5 Experiment 1 |
| RQ3 | Can the proposed content delivery method be used under time pressures to place content while maintaining an acceptable level of repeatable accuracy? | Chapter 6 Experiment 2 |
| RQ4 | Can users reliably identify physical objects in an Augmented Reality environment which are highlighted by a virtual Spatial Marker? | Chapter 7 Experiment 3 |
| RQ5 | Does the size of a Spatial Marker object relative to the physical object it is highlighting have an impact on the number of correct identifications given by users? | Chapter 7 Experiment 3 |

## 9.3   Limitations

As a novel approach, the KFT method and developed software system set out to achieve a number of goals which are discussed in Chapter 3. While the set of requirements were implemented successfully (Chapter 4) in order to allow the experimental studies and data collection for this thesis to go ahead, there are certain limitations to the approach. As such each of the experimental studies have threats to their validity which are discussed in their respective Chapters (5-7).

The fact that the creation of the environmental data using a tracking and mapping software had to be conducted under "perfect" conditions was unavoidable in view of experimental design. In order to ensure the repeatability of the studies, and fairness across each experiment steps were taken to ensure that lighting levels and camera calibration were not a factor - though they may be in a real world use of this method. Therefore there is a limited scope to the results collected, however they do provide a solid foundation upon which further research could be conducted. Which is discussed in the following section.

## 9.4    Further Work

While this thesis has successfully answered all of the Research Questions posed (as shown in Chapter 8), there is scope for further work to be carried out as a result of the development and experimental results of this project. The further work can be separated into Technical Development and Further Research.

### 9.4.1    Technical Development

The technical implementation and development of the Keyframe Tagging software was limited to the resources available (a single developer) and within time constraints. While the functionality matched the requirements, the development and evaluation of this prototype highlighted potential areas for further improvement.

While it was not deemed necessary for this project, where the focus was on fast, simple and accurate object placement through the KFT interface, in use case scenarios where the speed is not deemed to be important it would be possible to augment the interface with further functions. For example having the user select 10 positions for one tag across multiple frames and then taking the average positioning across them all. This would potentially result in a greater accuracy, however it would also incur a heavy time cost even in highly trained users.

As part of the tool's development, a series of pilot studies tested certain aspects of its design. For the needs of this project it was decided that Keyframes with less than 5 Feature Points should be discarded from the interface. This decision was taken in order to ensure only Keyframes which could be used as a starting point for increasing the Feature Point resolution were included. The comments provided by one participant suggested that this may still leave too many Keyframes; *"Many, many, photos to sort through to find what you are looking for, seems overwhelming"* (Section 5.3.5). In scenarios which afford more relaxed time constraints it would be possible to allow the user to increase or decrease this amount depending on the point density of the map they were working on.

Another issue which could be further explored with technological development was raised in several participant comments, regarding the selection of points when the scene becomes too crowded. A typical example of this is: *"When there was already a collection of points it made it rather difficult to select"*. The KFT approach could therefore be expanded to allow a form of image zooming on a Keyframe when a large density of Feature Points occupies a single area. This would allow for easier point selection, and potentially further improve the speed with which users can position content.

## 9.4.2   Further Research

Having identified that the KFT method can be used to accurately tag an environment under experimental lab conditions it would be possible to build on top of this research and further explore real world scenarios. In cases of disaster relief such as that explored in Scenario SN3.3.3 there could be technical barriers to overcome such as intermittent internet connections with data loss or other communication issues. While taken into account whilst designing this system, to test such conditions experimentally was outside the scope of this thesis.

Building on the results of Experiment 3 (Chapter 7) it would be possible to conduct further studies to combine several marker properties and observe the effect on Object Selection Ambiguity. While it was found that large markers were ineffective for close up identifications, they would undoubtedly provide faster localisations of an object in a large

environment. By developing the tracking and mapping aspect of such a system to work in the proposed scenarios with which KFT was designed, by switching the properties of markers depending on their distance from the user, further results could be collected.

## 9.5   Conclusion

This thesis has presented a novel means for content delivery in Augmented Reality environments, which can be used by untrained users to attain a repeatable level of acceptable accuracy. Three experiments were carried out to evaluate different aspects of the system; accuracy, performance under time constraints, interpretation of objects' position placed using the system. It is clear from the results of these experiments that the proposed Keyframe Tagging method can be used to place content into an Augmented Reality environment accurately.

With an average accuracy score of 64% which can be categorised as "Good" against a scale explained in Chapter 5 the approach matched the expectations detailed in the Research Questions. KFT exceeded expectations when operating under time constraints in comparison with pre-existing software, demonstrating its suitability for use across a wide range of scenarios, some of which are described in Chapter 3.3.

By placing navigational content into an environment using KFT and allowing participants to feedback on certain tasks, the approach provided an insight into how accurate the placements are perceived to be, by a non-expert user. This study also examined whether or not the style of navigational content impacted on the participants' actions and while the expected cognitive links between marker properties and object identification did not surface in this experiment, the participants reliably made correct identifications across several different trials.

In addition to providing a novel means of content delivery, this thesis has provided a solid foundation for conducting further research in both still image based content delivery, and object selection ambiguity. While the software developed is extensible and could be taken in a number of different research directions in the future, the develop-

ment and evaluation of this new method leads this thesis to conclude that Keyframe Tagging is a viable, accurate means of delivering content into unknown Augmented Reality environments.

# A

# XML Schema for KFT Data Model

The following is a code listing of the XML Schema for the output model produced by the Keyframe Tagging software, as discussed in Chapter 4.3.3.

```xml
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="map">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="keyframes">
          <xs:complexType>
            <xs:sequence>
             <xs:element name="keyframe" maxOccurs="unbounded" minOccurs="0">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element type="xs:integer" name="id"/>
                    <xs:element type="xs:string" name="imagePath"/>
                    <xs:element type="xs:string" name="coords"/>
                    <xs:element type="xs:string" name="cameraPosition"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="features">
          <xs:complexType>
```

```xml
                <xs:sequence>
                <xs:element name="feature" maxOccurs="unbounded" minOccurs="0">
                    <xs:complexType>
                      <xs:sequence>
                        <xs:element type="xs:integer" name="id"/>
                        <xs:element type="xs:integer" name="sourceKF"/>
                        <xs:element type="xs:string" name="coords"/>
                        <xs:element type="xs:string" name="cameraPosition"/>
                      </xs:sequence>
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="contents">
              <xs:complexType>
                <xs:sequence>
                <xs:element name="content" maxOccurs="unbounded" minOccurs="0">
                    <xs:complexType>
                      <xs:sequence>
                        <xs:element type="xs:integer" name="id"/>
                        <xs:element type="xs:string" name="objectPath"/>
                        <xs:element type="xs:string" name="coords"/>
                        <xs:element type="xs:string" name="cameraPosition"/>
                        <xs:element type="xs:integer" name="sourceKF"/>
                      </xs:sequence>
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
          <xs:attribute type="xs:integer" name="id"/>
        </xs:complexType>
      </xs:element>
</xs:schema>
```

# B

# Experiment 3 Supporting Data

## B.1  Target Object Lists by Environment

For each environment, a list of the expected answers are provided here. As candidates were asked to highlight the object with a laser pointer as well as verbally stating which object they had chosen, sometimes the distinction of "left" or "right" or position within a group was noted at the time of the experiment. The level of specificity for each object is also shown, as discussed in Chapter 7.

### B.1.1  Environment 1

**Table B.1:** Environment 1 Target Object Descriptions

| # | Correct Object/Description | Specificity |
| --- | --- | --- |
| 1 | A4 Ring Binder Folder | Single |
| 2 | Stapler | Single |
| 3 | Roll of Tape | Single |
| 4 | Tub of Paperclips | Single |
| 5 | Top Book (Software Architectures) (from off center stack) | Group Member |

| # | Correct Object/Description | Specificity |
|---|---|---|
| 6 | Phone (within packaging, not total package) | Group Member |
| 7 | Scissors (from mug along with other stationary) | Group Member |
| 8 | Blue Highlighter (2nd from left, from group of four) | Group Member |
| 9 | CD (disc in open case) | Object Part |
| 10 | Embossed Logo on External Hard Drive | Object Part |

## B.1.2     Environment 2

**Table B.2:** Environment 2 Target Object Descriptions

| # | Correct Object/Description | Specificity |
|---|---|---|
| 1 | Sellotape Dispenser | Single |
| 2 | Webcam | Single |
| 3 | Bunch of Keys | Single |
| 4 | Wallet | Single |
| 5 | Striped Coffee Mug (from set of 2 adjacent) | Group Member |
| 6 | CD in clear case (in group with 2 other cased discs) | Group Member |
| 7 | Labelled Pencil (in group of identical pencils, one with label) | Group Member |
| 8 | Center Leaflet (in fan of 5 identical leaflets) | Group Member |
| 9 | Large Blue Logo on Notebook cover | Object Part |
| 10 | Mouse Trackpad (on combined Keyboard and Trackpad) | Object Part |

## B.1.3     Environment 3

**Table B.3:** Environment 3 Target Object Descriptions

| # | Correct Object/Description | Specificity |
|---|---|---|
| 1 | Spool of CDs | Single |
| 2 | Scotch Tape in Box | Single |
| 3 | Toner Box labelled 'Refill' | Single |
| 4 | Roll of Masking Tape | Single |
| 5 | Silver Remote Control (from group of 3) | Group Member |

| # | Correct Object/Description | Specificity |
|---|---------------------------|-------------|
| 6 | Black Coloured Pencil (inside clear pack of pencils ) | Group Member |
| 7 | Large Bulldog Clip (adjacent to small clip) | Group Member |
| 8 | Smallest Screwdriver (in group of 3 different sizes) | Group Member |
| 9 | 'Return' Key on USB Keyboard | Object Part |
| 10 | 'Niceday' logo on small notebook | Object Part |

## B.1.4   Environment 4

**Table B.4:** Environment 4 Target Object Descriptions

| # | Correct Object/Description | Specificity |
|---|---------------------------|-------------|
| 1 | Tub of Drawing Pins | Single |
| 2 | Stack of Books (neatly piled, no overlaps) | Single |
| 3 | Black Logitech Plastic Case | Single |
| 4 | HTC Box | Single |
| 5 | Red Dry Wipe Pen (from pair of pens) | Group Member |
| 6 | Rightmost Post-it Notes (from group of 3) | Group Member |
| 7 | Left Speaker (from set of 2 speakers) | Group Member |
| 8 | Left DVD (from 2 upturned discs) | Group Member |
| 9 | Logo on Precision Screwdriver Case | Object Part |
| 10 | Large Picture (in open magazine article) | Object Part |

# B.2   Trial Ordering to Remove Variable Effects

In order to ensure that the order in which participants completed the experiment had no bearing on the result, the following trial ordering (Table B.5) was enforced. The four trial environments are shown E1 through 4, and the 4 Spatial Marker sizes (S - Small, L - Large, P - Proportional, I - Inversely Proportional)

**Table B.5:** Participant Ordering for Experiment 3

| P | Trial 1 | Trial 2 | Trial 3 | Trial 4 |
|---|---------|---------|---------|---------|
| 1 | E1-S | E2-L | E4-I | E3-P |
| 2 | E1-L | E2-P | E4-S | E3-I |
| 3 | E1-P | E2-I | E4-L | E3-S |
| 4 | E1-I | E2-S | E4-P | E3-L |
| 5 | E2-S | E3-L | E1-I | E4-P |
| 6 | E2-L | E3-P | E1-S | E4-I |
| 7 | E2-P | E3-I | E1-L | E4-S |
| 8 | E2-I | E3-S | E1-P | E4-L |
| 9 | E3-S | E4-L | E2-I | E1-P |
| 10 | E3-L | E4-P | E2-S | E1-I |
| 11 | E3-P | E4-I | E2-L | E1-S |
| 12 | E3-I | E4-S | E2-P | E1-L |
| 13 | E4-S | E1-L | E3-I | E2-P |
| 14 | E4-L | E1-P | E3-S | E2-I |
| 15 | E4-P | E1-I | E3-L | E2-S |
| 16 | E4-I | E1-S | E3-P | E2-L |

# References

Ahn, S., Ko, H. and Yoo, B. (2014), "Webizing Mobile Augmented Reality Content", *New Review of Hypermedia and Multimedia*, Vol. 20 No. 1, pp. 79–100.

Apple. (2011), "Cocoa Core Competencies", available at: `https://developer.apple.com/library/mac/documentation/General/Conceptual/DevPedia-CocoaCore/MVC.html`.

Azuma, R.T. (1997), "A Survey of Augmented Reality", *Presence: Teleoperators and Virtual Environments*, Vol. 4 No. August, pp. 355–385.

Azuma, R.T. (1999), "Tracking in Unprepared Environments for Augmented Reality Systems", *Computers & Graphics*, Vol. 23 No. 6, pp. 787–793.

Baillot, Y., Julier, S., Brown, D. and Livingston, M. (2006), "A Tracker Alignment Framework for Augmented Reality", in *The Second IEEE and ACM International Symposium on Mixed and Augmented Reality, 2003*, pp. 142–150.

Barakonyi, I., Fahmy, T. and Schmalstieg, D. (2004), "Remote Collaboration Using Augmented Reality Videoconferencing", in *Proceedings of Graphics Interface 2004*, pp. 89–96.

Bay, H., Tuytelaars, T. and Gool, L.V. (2006), "SURF : Speeded Up Robust Features", *Computer Vision*, Vol. 3951, pp. 404–417.

Behzadan, A. and Kamat, V. (2005), "Visualization of Construction Graphics in Outdoor Augmented Reality", in *Proceedings of the 37th Conference on Winter Simulation*, pp. 1914–1920.

Bichlmeier, C., Wimmer, F., Heining, S.M. and Navab, N. (2007), "Contextual Anatomic Mimesis Hybrid In-Situ Visualization Method for Improving Multi-Sensory Depth Perception in Medical Augmented Reality", *Symposium on Mixed and Augmented Reality*, pp. 1–10.

Billinghurst, M. (2001), "The MagicBook: a Transitional AR Interface", *Computers*

*& Graphics*, Vol. 25 No. 5, pp. 745–753.

Billinghurst, M. and Duenser, A. (2012), "Augmented Reality in the Classroom", *Computer Magazine*, Vol. 45 No. 7, pp. 56–63.

Billinghurst, M. and Kato, H. (2002), "Collaborative Augmented Reality", *Communications of the ACM*, Vol. 45 No. 7, pp. 64–70.

Billinghurst, M., Poupyrev, I., Kato, H. and May, R. (2000), "Mixing Realities in Shared Space: An Augmented Reality Interface for Collaborative Computing", in *Proceedings of ICME 2000*, pp. 1641–1644.

Bleset, G., Wuest, H. and Stricker, D. (2006), "Online Camera Pose Estimation in Partially Known and Dynamic Scenes", *ISMAR '06*, pp. 56–65.

Blum, J., Greencorn, D.G. and Cooperstock, J.R. (2013), "Smartphone Sensor Reliability for Augmented Reality Applications", *Mobile and Ubiquitous Systems: Computing, Networking, and Services*, pp. 127–138.

Broll, W., Lindt, I., Wittk, M., Novotny, T., Fatah, A., Mottram, C., Strothmann, A., et al. (2004), "ARTHUR : A Collaborative Augmented Environment for Architectural Design and Urban Planning", *Virtual Reality*, Vol. 1 No. 1, pp. 1–10.

Carozza, L., Tingdahl, D., Bosche, F. and Gool, L. (2014), "Markerless Vision-Based Augmented Reality for Urban Planning", *Computer-Aided Civil and Infrastructure Engineering*, Vol. 29 No. 1, pp. 2–17.

Castle, R. and Klein, G. (2008), "Video-rate Localization in Multiple Maps for Wearable Augmented Reality", in *12th IEEE Symposium on Wearable Computers*, pp. 15–22.

Cheverst, K., Davies, N., Mitchell, K. and Friday, A. (2000), "Experiences of Developing and Deploying a Context- Aware Tourist Guide : The GUIDE Project", in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, pp. 20–31.

Comport, A., Marchand, E. and Chaumette, F. (2003), "A Real-Time Tracker for Markerless Augmented Reality", *ISMAR '03*, pp. 36–45.

Crivellaro, A., Yannick, V., Kwang Moo, Y., Fua, P. and Lepetit, V. (2014), "Tracking Texture-less Shiny Objects with Descriptor Fields", in *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 331–332.

Csorba, M. (1997,August), *Simultaneous Localisation and Map Building* (PhD Dissertation), University of Oxford.

Davison, A., Mayol, W. and Murray, D. (2003), *Real-time Localization and Mapping with Wearable Active Vision*, IEEE Computer Society, pp. 18–27.

Davison, A.J. (2003), "Real-Time Simultaneous Localisation and Mapping with a Single Camera", in *Proceedings of Ninth IEEE International Conference on Computer Vision, 2003*, pp. 1403–1410.

Davison, A.J. and Murray, D. (2002), "Simultaneous Localization and Map-building using Active Vision", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24 No. 7, pp. 865–880.

Davison, A.J., Reid, I.D., Molton, N.D. and Stasse, O. (2007), "MonoSLAM: Real-time Single Camera SLAM.", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29 No. 6, pp. 1052–67.

Dorfmüller, K. (1999), "Robust Tracking for Augmented Reality using Retroreflective Markers", *Computers & Graphics*, Vol. 23 No. 6, pp. 795–800.

Duh, H.B.-L. and Billinghurst, M. (2008), "Trends in Augmented Reality Tracking, Interaction and Display: A Review of Ten Years of ISMAR", *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pp. 193–202.

Eade, E. and Drummond, T. (2006), "Scalable Monocular SLAM", in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 469–476.

Engels, C., Stewenius, H. and Nister, D. (2006), "Bundle Adjustment Rules", in *Photogrammetric Computer Vision*.

Feiner, S., MacIntyre, B., Haupt, M. and Solomon, E. (1993), "Knowledge-Based Augmented Reality", *Communications of the ACM*, Vol. 36 No. 7, pp. 53–62.

Feiner, S., MacIntyre, B., Hollerer, T. and Webster, A. (1997), "A Touring Machine: Prototyping 3D Mobile Augmented Reality Systems for Exploring the Urban Environment", *Digest of Papers. First International Symposium on Wearable Computers*, Vol. 97, pp. 74–81.

Gillet, A., Sanner, M., Stoffler, D., Goodsell, D. and Olson, A. (2004), "Augmented Reality with Tangible Auto-fabricated Models for Molecular Biology Applications", *IEEE Visualization 2004*, pp. 235–241.

Haringer, M. and Regenbrecht, H.T. (2002), "A Pragmatic Approach to Augmented Reality Authoring", *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, Vol. 0, p. 237.

Henderson, S.J. and Feiner, S. (2009), "Evaluating the Benefits of Augmented Reality for Task Localization in Maintenance of an Armored Personnel Carrier Turret", in *Proceeding of IEEE International Symposium on Mixed and Augmented Reality (ISMAR '09)*, pp. 135–144.

Henrysson, A., Billinghurst, M. and Ollila, M. (2005), "Face to Face Collaborative AR on Mobile Phones", in *International Symposium on Augmented and Mixed Reality*, pp. 80–89.

Hill, A., MacIntyre, B., Gandy, M., Davidson, B. and Rouzati, H. (2010), "KHARMA: An Open KML/HTML Architecture for Mobile Augmented Reality Applications", *9th IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 233–234.

Höllerer, T. (1999), "Exploring MARS: Developing Indoor and Outdoor User Interfaces to a Mobile Augmented Reality System", *Computers & Graphics*, Vol. 23

No. 6, pp. 779–785.

Ishii, H. and Ullmer, B. (1997), "Tangible Bits : Towards Seamless Interfaces between People , Bits and Atoms", in *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems.*, ACM, pp. 234–241.

Julier, S., Baillot, Y., Lanzagorta, M., Brown, D. and Rosenblum, L. (2000), "BARS : Battlefield Augmented Reality System", in *NATO Symposium on Information Processing techniques for Military Systems*, pp. 9–11.

Kato, H. and Billinghurst, M. (1999), "Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System", *IWAR*, pp. 85–94.

Kato, H., Billinghurst, M., Poupyrev, I., Imamoto, K. and Tachibana, K. (2000), "Virtual Object Manipulation on a Table-top AR Environment.", in *Proceedings of the 2nd International Symposium on Augmented Reality*, ISAR, pp. 111–119.

Kim, S.-J., Jang, M.-S. and Kim, H.-S. (2004), "A Development of Interactive game 'ting ting' using Real and Virtual Objects", in *IEEE International Conference on Systems, Man and Cybernetics*, IEEE, pp. 1191–1197.

Klein, G. and Drummond, T. (2003), "Robust Visual Tracking for Non-Instrument Augmented Reality", *ISMAR '03*, pp. 113–122.

Klein, G. and Murray, D. (2007), "Parallel Tracking and Mapping for Small AR Workspaces", *Symposium on Mixed and Augmented Reality*, pp. 1–10.

Kooper, R. and Macintyre, B. (2003), "Browsing the Real-world Wide Web: Maintaining Awareness of Virtual Information in an AR Information Space", *International Journal of Human-Computer Interaction*, Vol. 16 No. 3, pp. 425–446.

Krevelen, D.W.F. van and Poelman, R. (2010), "A Survey of Augmented Reality Technologies, Applications, and Limitations", *The International Journal of Virtual Reality*, Vol. 9 No. 2, pp. 1–20.

Kutulakos, K. and Vallino, J. (1998), "Calibration-free Augmented Reality", *IEEE*

*Transactions on Visualization and Computer Graphics*, Vol. 4 No. 1, pp. 1–20.

LayAR. (2010), "LayAR", available at: `https://www.layar.com`.

Ledermann, F. and Schmalstieg, D. (2005), "A High-level Framework for Creating Augmented Reality Presentations", in *In Proceedings of the 2005 IEEE Virtual Reality Conference*.

Lee, G., Nelles, C., Billinghurst, M. and Kim, G. (2004), "Immersive Authoring of Tangible Augmented Reality Applications", *Third IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 172–181.

Li, D., Chuah, M.C. and Tian, L. (2014), "Demo: Lehigh Explorer Augmented Campus Tour (LACT)"", in *Proceedings of the 2014 Workshop on Mobile Augmented Reality and Robotic Technology-based Systems*, pp. 15–16.

Lin, H. (2012), "Interacting with Visual Poems Through AR-Based Digital Artwork", *Turkish Online Journal of Education Technology*, Vol. 11 No. 1, pp. 123–127.

Macintyre, B., Gandy, M., Dow, S. and Bolter, J.D. (2004), "DART: A Toolkit for Rapid Design Exploration of Augmented Reality Experiences", in *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*, pp. 197–206.

Madritsch, F. and Gervautz, M. (1996), "CCD-Camera Based Optical Beacon Tracking for Virtual and Augmented Reality", *Computer Graphics Forum*, Vol. 15 No. 3, pp. 207–216.

Matsutomo, S., Miyauchi, T., Noguchi, S. and Yamashita, H. (2012), "Real-time Visualization System of Magnetic Field Utilizing Augmented Reality Technology for Education", in *IEEE Transactions on Learning Technologies*, pp. 531–534.

Milgram, P. and Colquhoun, H. (1999), "A Taxonomy of Real and Virtual World Display Integration", *Mixed Reality: Merging Real and Virtual Worlds*, pp. 5–30.

Montemerlo, M., Thrun, S., Koller, D. and Wegbreit, B. (2002), "FastSLAM : A Factored Solution to the Simultaneous Localization and Mapping Problem", in *Proceedings of the AAAI National Conference on Artificial Intelligence*, pp. 593–598.

Nister, D., Naroditsky, O. and Bergen, J.R. (2005), "Visual Odometry", in *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 652–659.

Park, H. and Park, J.-I. (2005), "Invisible Marker Based Augmented Reality System", in *Visual Communications and Image Processing*, Vol. 5960, pp. 59601–59601.

Park, J., You, S. and Neumann, U. (1998), "Natural Feature Tracking for Extendible Robust Augmented Realities", *IWAR*.

Piekarski, W. (2006), "3D modeling with the Tinmith mobile outdoor augmented reality system", *Computer Graphics and Application*, Vol. 26 No. 1, pp. 14–17.

Piekarski, W. and Thomas, B. (2001), "Tinmith-evo5 An Architecture for Supporting Mobile Augmented Reality Environments", in *International symposium on augmented reality*, pp. 177–195.

Piekarski, W. and Thomas, B. (2002), "ARQuake: The Outdoor Augmented Reality Gaming System", *Communications of the ACM*, Vol. 45 No. 1, pp. 36–38.

Poupyrev, I., Tan, D.S., Billinghurst, M., Kato, H., Regenbrecht, H. and Tetsutani, N. (2002), "Developing a Generic Augmented Reality Interface", *IEEE Computer*, Vol. 35 No. 3, pp. 44–50.

Regenbrecht, H., Wagner, M. and Baratoff, G. (2002), "MagicMeeting: A Collaborative Tangible Augmented Reality System", *Virtual Reality*, Vol. 6 No. 3, pp. 151–166.

Roesner, F., Kohno, T. and Molnar, D. (2014), "Security and Privacy for Augmented Reality Systems", *Communications of the ACM*, Vol. 57 No. 4, pp.

88–96.

Santos, M.E.C., Yamamoto, G., Taketomi, T., Miyazaki, J. and Kato, H. (2013), "Authoring Augmented Reality Learning Experiences as Learning Objects", in *IEEE International Conference on Advanced Learning Technologies*, pp. 506–507.

Satoh, K., Uchiyama, S. and Yamamoto, H. (2004), "A Head Tracking Method Using Bird's-Eye View Camera and Gyroscope", *ISMAR '04*, pp. 202–211.

Schmalstieg, D. and Wagner, D. (2005), "A Handheld Augmented Reality Museum Guide", in *IADIS International Conference on Mobile Learning*.

Schmalstieg, D., Fuhrmann, A., Hesina, G., Szalavari, Z., Encarnacao, L.M., Gervautz, M. and Purgathofer, W. (2002), "The Studierstube Augmented Reality Project", *Presence: Teleoperators & Virtual Environments*, Vol. 11 No. 1, pp. 33–54.

Schmalstieg, D., Schall, G., Wagner, D., Barakonyi, I., Reitmayr, G., Newman, J. and Ledermann, F. (2007), "Managing Complex Augmented Reality Models", *IEEE Computer Graphics and Applications*, IEEE Computer Society, No. August, pp. 48–57.

Shim, J., Kong, M., Yang, Y., Seo, J. and Han, T.D. (2014), "Interactive Features Based Augmented Reality Authoring Tool", in *IEEE International Conference on Consumer Electronics*, pp. 47–50.

Spohrer, J.C. (1999), "Information in Places", *IBM Systems Journal*, Vol. 38 No. 4, pp. 602–628.

Stafford, A., Piekarski, W. and Thomas, B. (2006), "Implementation of God Like Interactions Techniques for Supporting Collaboration Between Outdoor AR and Indoor Tabletop Users", *ISMAR '06*, pp. 165–172.

Stewenius, H., Engels, C. and Nister, D. (2006), "Recent Developments on Direct Relative Orientation", *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 60 No. 4, pp. 284–294.

Sutherland, I. (1965), "The Ultimate Display", *Multimedia: From Wagner to Virtual Reality*, pp. 506–508.

Wagner, D. (2005), *Towards Massively Multi-user Augmented Reality on Handheld Devices*, Springer Berlin Heidelberg.

Wu, H.K., Lee, S.W.Y., Chang, H.Y. and Liang, J.C. (2013), "Current Status, Opportunities, and Challenges of Augmented Reality In Education", *Computers & Education*, Vol. 62, pp. 41–49.

Wu, M., Mitchell, K., McCaffery, D., Finney, J. and Friday, A. (2004), "Real Tournament – Mobile Context-aware Gaming for the Next Generation", *The Electronic Library*, Vol. 22 No. 1, pp. 55–64.

Wuest, H., Vial, F. and Stricker, D. (2005), "Adaptive Line Tracking with Multiple Hypotheses for Augmented Reality", *ISMAR '05*, pp. 62–69.

You, S., Neumann, U. and Azuma, R.T. (1999), "Orientation Tracking for Outdoor Augmented Reality Registration", *Ieee Computer Graphics And Applications*, No. December, pp. 36–42.