# Linking Spatial Video and GIS

Paul Lewis, BSc.

July 2009

NUI MAYNOOTH

Ollscoil na hÉireann Má Nuad

Dissertation submitted in fulfilment of the requirements
for candidature for degree of

Doctor of Philosophy

Department of Computer Science &
National Centre for Geocomputation
National University of Ireland Maynooth
Maynooth
Co. Kildare
Ireland

Supervisor: Dr A.C. Winstanley
Supervisor: Prof. A.S. Fotheringham

embark initiative
Investing in People and Ideas
John & Pat Hume Scholarships
NCG
National Centre for Geocomputation
An tIonad Náisiúnta Geoiríomhaireachta
sfi
science foundation ireland
fondúireacht eolaíochta éireann

# Abstract

Spatial Video is any form of geographically referenced videographic data. The forms in which it is acquired, stored and used vary enormously; as does the standard of accuracy in the spatial data and the quality of the video footage. This research deals with a specific form of Spatial Video where these data have been captured from a moving road-network survey vehicle. The spatial data are GPS sentences while the video orientation is approximately orthogonal and coincident with the direction of travel.

GIS that use these data are usually bespoke standalone systems or third party extensions to existing platforms. They specialise in using the video as a visual enhancement with limited spatial functionality and interoperability. While enormous amounts of these data exist, they do not have a generalised, cross-platform spatial data structure that is suitable for use within a GIS. The objectives of this research have been to define, develop and implement a novel Spatial Video data structure and demonstrate how this can achieve a spatial approach to the study of video.

This data structure is called a Viewpoint and represents the capture location and geographical extent of each video frame. It is generalised to represent any form or format of Spatial Video. It is shown how a Viewpoint improves on existing data structure methodologies and how it can be theoretically defined in 3D space. A 2D implementation is then developed where Viewpoints are constructed from the spatial and camera parameters of each survey in the study area. A number of problems are defined and solutions provided towards the implementation of a post-processing system to calculate, index and store each video frame Viewpoint in a centralised spatial database.

From this spatial database a number of geospatial analysis approaches are demonstrated that represent novel ways of using and studying Spatial Video based on the Viewpoint data structure. Also, a unique application is developed where the Viewpoints are used as a spatial control to dynamically access and play video in a location aware system.

While video has been to date largely ignored as a GIS spatial data source; it is shown through this novel Viewpoint implementation and the geospatial analysis demonstrations that this need not be the case anymore.

# Acknowledgements

Support for such an undertaking as a PhD. comes in many forms, principle amongst them the emotional, practical and technical help. While the experience and personal journey was rewarding it was also fraught with many disheartening low points.

To my wife, Caroline, the greatest gratitude possible has to be acknowledged as without your unyielding support and energy I would not have been able to complete this thesis. You provided the motivation, drive and time it required and hopefully I can reward that commitment. My daughter, Eva, is also a huge resource of emotional support and motivation, unfortunately, at two years old she doesn't realise that yet.

To my supervisor Adam, I thank you whole heartedly, as without your support over the last seven and a half years, many of my personal achievements since I came back to college in 2001 could not have been realised; most importantly, I would never have undertaken this PhD in 2005. Whether reading document drafts, writing references or advising on my work, you have always been thorough and motivating, especially in the low moments.

To my second supervisor, Stewart, I also thank you with much gratitude for the trust and confidence you placed in me. Your advice, draft reading and financial support, from the Science Foundation Ireland Research Professorship grant 03/RP1/I382, have all been great morale boosting aspects of my PhD. experience.

To my family, both in-laws and immediate, especially my parents Noreen and Ian, your help and support have been immense. Rallying round to renovate a house, raise a baby, provide sustenance and accommodations, etc., etc.; it was all appreciated beyond any gratitude that can ever be repaid, thank you.

I would also like to acknowledge the EMBARK initiative, through which I received my principle postgraduate funding scholarship, IRCSET. Thanks also go to NUI Maynooth for a multitude of reasons, principally the John & Pat Hume research Scholarship.

Last but not least, NCG, NIRSA and Computer Science are sources of much inspiration and friends. Both current and present staff and student members have all contributed in some way to my NUIM academic experience and I thank you all collectively; an individual list would be just far too long!

# Contents

## Appendices:

# List of Tables

## Chapter One:

## Chapter Two:

## Chapter Three:

## Chapter Five:

## Chapter Six:

## Chapter Seven:

# List of Figures

## Chapter Two:

## Chapter Three:

## Chapter Four:

## Chapter Five:

## Chapter Six:

## Chapter Seven:

# List of Equations and SQL Queries

## Chapter Five:

## Chapter Seven:

# List of Acronyms

| | |
|---|---|
| 2D | Two Dimensions |
| 3D | Three Dimensions |
| µs | Microsecond |
| AOV | Angle Of View |
| API | Application Programming Interface |
| ASCII | American Standard Code for Information Interchange |
| ASF | Advanced Systems Format |
| AVC | Advanced Video Coding |
| AVI | Audio Video Interlaced |
| CCD | Charged-Coupled Device |
| COC | Circle of Confusion |
| CPU | Central Processing Unit |
| DBMS | Database Management System |
| DEM | Digital Elevation Model |
| DFXP | Distributed Format Exchange Profile |
| DGPS | Differential Global Positioning System |
| DOD | Department Of Defence |
| DOF | Depth of Field |
| DTS | Decode Time Stamp |
| DV | Digital Video |
| DVD | Digital Video Disk |
| EG | Engineering Guidelines |
| ESRI | Environmental Systems Research Institute |
| ETRI | Electronics and Telecommunications Research Institute |
| FOV | Field of View |
| FPS | Frames Per Second |
| FSK | Frequency Shift Key |
| GI | Geographical Information |
| GIS | Geographical Information System |
| GNSS | Global Navigation Satellite System |
| GPRS | General Packet Radio Service |
| GPS | Global Positioning System |
| GSM | Global System for Mobile communications |
| HD | High Definition |
| IEC | International Electrotechnical Commission |
| IEEE | Institute of Electrical and Electronics Engineers |
| IMU | Inertial Measurement Unit |
| INS | Inertial Navigation System |
| ISO | International Standards Organisation |
| ISR | Intelligence, Surveillance and Reconnaissance |
| JVC | Victor Company of Japan |
| KLV | Key Length Value |
| LBS | Location Based Service |
| LIDAR | Light Detection and Ranging |
| MISB | Motion Imagery Standards Board |
| MIT | Massachusetts Institute of Technology |
| MPEG | Moving Pictures Expert Group |

| | |
|---|---|
| MXF | Material Exchange Format |
| NATO | North Atlantic Treaty Organisation |
| NMEA | National Marine Electronics Association |
| NUIM | National University of Ireland Maynooth |
| OGC | Open Geospatial Consortium |
| OWS | OGC Web Services |
| OSI | Ordnance Survey Ireland |
| PCM | Pulse Code Modulation |
| PTS | Presentation Time Stamp |
| RIFF | Resource Interchange File Format |
| RTK | Real Time Kinematic |
| SD | Standard Definition |
| SMPTE | Society of Motion Picture and Television Engineers |
| SQL | Structured Query Language |
| STANAG | STANardisation Agreements |
| TIN | Triangulated Irregular Network |
| TS | Transport Stream |
| TT | Timed Text |
| TTI | Text and Timing Information |
| UAS | Unmanned Air System |
| UAV | Unmanned Aerial Vehicle |
| US | United States |
| WAV | WAVeform |
| WMV | Windows Media Video |
| XML | eXtensible Markup Language |
| XMT | eXtensible MPEG4 Textual |

# Chapter One: Introduction

The use of Spatial Video, as a visually enriched GIS data source, is improving rapidly and is proving to be useful in aerial (Livingstone *et al.*, 1998), terrestrial (Mc Loughlin *et al.*, 2005; Ó Riain *et al.*, 2006A) and marine (Rzhanov *et al.*, 2000) survey and analysis. However, the capture, storage, processing and analysis of Spatial Video is usually done using bespoke hardware and/or software implementations that have narrowly focused application areas. It would be useful to integrate Spatial Video into a GIS framework that would enhance its use through existing spatial data functionalities and interaction with other GIS data sources. This chapter outlines the following central research question: can a Spatial Video index and query data model be designed, developed and implemented using existing GIS data structures and methods? A detailed list of research targets and objectives is also given; followed by a layout of the structure of the thesis.

## 1.1 GIS and Spatial Video

As defined by Worboys *et al.*, (2004A) '*A Geographical Information System (GIS) is a computer-based information system that can be used to capture, model, retrieve, share, manipulate, analyze and present geographically referenced data*'. While accepting this as a general definition of the ideal set of GIS functionalities, considering different types of GIS data in terms of these functions can also be useful. In this study, this approach is taken where a spatial data source is fitted to a GIS data structure and demonstrates its GIS suitability based on these functionalities. The spatial data source is geographically referenced videographic data, or Spatial Video. The properties of this spatial data source, to date, have not been defined for their amenability to support the functions identified by Worboys *et al.*, (2004A) in an integrated GIS role and the underlying aim of this thesis is to provide a simplified model of Spatial Video to support such functionality.

A study of any one of these GIS functions could form a research area in its own right, even when using Spatial Video as the only data source. However, in this work, the possibility of using Spatial Video as a usable data source for all these functions is considered. While not all the functions are considered in detail, some have practical implementations performed as part of a simulated set of detailed case studies.

To facilitate this discussion, a broader re-definition of geographically referenced videographic data is necessary for both clarity and ease of generalisation. For geographically referenced videographic data the term *Spatial Video* will be used. This is defined and explained in the following sections.

## 1.2 Spatial Video

In general, videography is a well-understood concept that can be defined as the process and/or set of methods and operations used to capture a sequence of moving images, (Kiger, 1972). It has existed for many years and is ever-present in our daily lives in numerous capture and display formats. Spatial video is a specific extension to any of these numerous video formats where spatial attributes are applied to some or all of the images/frames within the captured sequence. In general terms, spatial attributes can include any number of different descriptors that can help define a video's image/frame location, time, altitude, orientation or other spatial attribute. The methods are varied when acquiring Spatial Video, in the types of sensors and equipment used, but also in the forms of integration and recording of the video and spatial properties, (Foy *et al.*, 2007) is an example. Table 1.1 loosely identifies the general relationships between a typical video image and various spatial data sensors.

| VIDEO FRAME ATTRIBUTE | EQUIPMENT/SENSOR |
|---|---|
| Image Capture | Video Camcorder |
| | Progressive Scan Camera |
| Position | GPS (Global Positioning System) |
| | Assisted GPS – DGPS (Differential GPS) |
| Orientation | Altimeter/Compass/Inertial Navigation Unit |

Table 1.1 Types of equipment used to capture video and its spatial properties.

The collection and use of spatial video is typically linear and involves a number of operational stages:

1. Acquisition
2. Processing
3. Storage
4. Distribution
5. Analysis

Each of these stages typically requires further subsets of heterogeneous operations that have been developed using numerous different technologies and methods to handle specific objectives or applications. A typical example of a Spatial Video application area is Aerial Videography, (McCarthy, 1999). This is a distinct research field in its own right, with many organisations and agencies using aerial-based Spatial Video to acquire planar views for their respective domains. However, this thesis specifically considers the near-orthogonal form of Spatial Video that has been captured for rail, road and other terrestrially based infrastructural management and assessment projects.

### 1.3 Spatial Video Challenges

The existing inherent nature of Spatial Video is as a bespoke data acquisition solution in many mapping, survey and environmental analysis projects. This diverse need has resulted in a number of problems and challenges when trying to define a more general GIS model for Spatial Video, especially with previously captured data sets. Chief amongst these problems is a broad understanding of Spatial Video's place as a data source for a GIS. Typically, Spatial Video is collected for a specific reason, used to provide a particular solution and, very likely, never used again. Thus, its place has been to satisfy an immediate need where no further usage or applications of the data are either conceived or implemented. This has resulted in enormous amounts of Spatial Video being collected and then stored in various distributed archives, in many formats, to largely differing (and undocumented) levels of visual and spatial detail and quality, with no further usage.

These problems centre on there being no centralised or generalised structure to index and search this amount of video data from a GIS perspective. No common structure exists that defines Spatial Video and the sort of GIS geospatial analysis that could be relevant to and/or performed on it. The large amount of retrospective Spatial Video that exists has no standardised video or consistent spatial format. Probably, every possible video storage format has been used, including legacy formats no longer supported, to record the Video sequences. Even very basic image sequence directory structures are semantically considered Spatial Video in nature even though they do not conform to any standard video container format specifications. The quality and accuracy of the spatial data are also very variable, especially in the retrospective GPS accuracy, which not only contends with existing errors but also contains larger errors in the pre-2000 selective availability era.

The central (and obvious) factor encompassing all these formats is that video provides a visual perspective of the environment. This aspect has always defined the application areas for which video is used in a GIS and involves any of the following objectives:

- enhancing the GIS with recorded digital imagery of the cartographic environment (Hirose *et al.*, 1998; Kawasaki *et al.*, 1999; McCarthy, 1999)
- using the visual information to generate other spatial data sets through supervised or unsupervised visual analysis of the image content (Mc Loughlin *et al.*, 2005; Ó Riain *et al.*, 2006A; Ó Riain *et al.*, 2006B)
- using the video's geographical content and spatial parameters to segment or semantically describe the video (Hirose *et al.*, 1998; Nobre *et al.*, 2001; Navarrete *et al.*, 2002).

Thus, Spatial Video has a number of specific approaches in its application and also numerous practical uses. However, a generalised GIS-constrained semantic definition is lacking. With such a definition, any Spatial Video sequence should be easily modelled, based on its geography, for easy visual playback, content analysis or indexing.

### 1.4 Modelling Spatial Video

In this study a general solution to the various problems and challenges of Spatial Video is defined through modelling its content in a geographical context. Two approaches to this solution are possible based on the available knowledge of Spatial Video implementations.

1. Video-centred Solution:

   By investigating all the existing video-centred methodologies and the possibilities of extending these to provide a solution is considered. This is the standard Spatial Video approach where individual frames or groups of frames are spatially indexed in an embedded or associated file format. Various video file formats and standards exist that are applicable to storing and indexing video with spatial properties. This methodology has a distinct advantage over other types of indexing as data capture devices and platforms can be managed in a predefined and calibrated fashion. This makes for a consistent and reliable Spatial Video solution where output video and spatial data streams are encapsulated in well-understood formats that contain all the relevant survey data in a single source location. A disadvantage of this approach is the lack of any detailed GIS context description of the video sequences' spatial elements.

2. GIS-centred Solution:

   By investigating a spatial-extent context where the object space of a video sequence is modelled in a GIS-centred manner. Effectively, the space rather than the video itself is linked with relevant frames or sequence indexes. There are a number of possibilities for providing a solution using this approach where a semantic understanding of the spatial content of video can be defined. One of these possible solutions is discussed and implemented with emphasis on defining the data structures to facilitate and support existing Spatial Video data sets in a GIS framework while also providing extensibility for future development. An advantage of this approach is its wider applicability and cost effectiveness through further use of existing Spatial Video surveys. A disadvantage is the work in organisation and assimilation of existing Spatial Video streams into a single coherent GIS data source and, probably, the post-processing requirements for inclusion with existing and future data sets.

To define this GIS-centred solution a new model is developed where the geographical space, as captured by each image, is constructed as a GIS primitive data type that defines the spatial extent of each video frame. This model allows complete flexibility in dealing with the range of geographic precision issues that any particular Spatial Video data frame may present. In many cases, very basic empirical knowledge of a Spatial Video data set will provide the minimum set of control parameters that are necessary to construct a maximal Viewpoint sequence. In other cases, the Viewpoint model can be extended to implement geographical extents of higher levels of precision using alternative data sources as controls. These could be collected by other sensors during the survey, such as Light Detection and Ranging (LIDAR) device, and/or applied during a post processing operation, such as models of topography or the built environment.

## 1.5 Research Objectives

While Spatial Video is a very useful visual GIS data source, any single bespoke application is usually restricted to a specific project that is based on the content of the visual information. However, while the output of such projects normally results in other forms of reusable spatial data being created, the original video element is often ignored and/or forgotten as it is so specific that no further use seems possible. Thus, the main objective of this research is to show that a generalised Spatial Video data structure can be constructed and implemented to enhance its GIS interoperability. This goes beyond treating the video in a bespoke technical format that provides a source of visual information for a specific project, but treats it in a generalised form that has a much wider set of possible applications.

This objective has one principle aim; to retrieve logical video streams or images from a Spatial Video data index, based on well-understood GIS geospatial analysis techniques involving non-video spatial data sets. Two distinct approaches, both of which concern GIS geospatial analysis capabilities, are considered. Firstly, GIS operations on diverse and distributed Spatial Video data sets should be possible using basic GIS queries, such as those based on point locations, buffers around objects and/or other spatial measurements, which can return appropriately composed video streams. Secondly, more complex GIS operations can enable interactions between Spatial Video and non-video spatial data sources, such as standard raster or vector files.

Effectively this will enable a maximum level of GIS support for any type of Spatial Video that has been defined by the structures developed in this project. To achieve this, a number of elements are important:

- Theoretical investigations should consider a broad range of existing Spatial Video implementations but devise solutions that are independent of any specific type.

- Spatial Video indexing structures should incorporate and support a wide range of possible video and spatial data structure combinations.

- GIS geospatial analysis techniques should be considered broadly, based on their GIS data structure requirements.

- A large amount of diverse Spatial Video data sets should be acquired or collected to demonstrate a number of GIS operational capabilities that will prove the broad applicability of this research.

In undertaking this research a number of specific objectives are outlined; a brief discussion is also included with each.

### 1.5.1 Indexing Video with Spatial Data

What methods and standards exist for the internal indexing of video formats to incorporate both spatial and camera parameters? Do these methods and standards provide a feasible and/or efficient indexing solution to support generalised Spatial Video GIS geospatial analysis?

Based on existing data sets, solutions to this type of indexing are varied and bespoke, reflecting the lack of metadata standards which are only beginning to develop in this area. However, one standard that is currently implemented in aerial video capture systems specifically targets spatial parameter metadata indexing. This is the MISB (Motion Imagery Standards Board) engineering guidelines 104.5 (Long, 2006). Other solutions include, audio encoded frequency modulation, video timestamp indexed subtitle file formats, MPEG (Moving Pictures Expert Group) 7 and 21 standards implementations. However, many solutions still fail to specify a universal video geo-indexing standard, although the MISB is currently undergoing a review and update of its 104.5 standard to facilitate a more flexible and broader application.

### 1.5.2 Decoding Audio Indexed Spatial Video Streams

One existing form of real-time spatial indexing of video is hardware-encoded frequency-modulated audio-streams. Previously, this form of indexing could only be hardware decoded; however, a software solution has been developed to decode the spatial parameters from the video's audio stream. Analysis of the efficiency of this indexing method and the accuracy of the image-frame to spatial-location relationship is also discussed.

### 1.5.3 Theoretically Extending ViewCones to Viewpoints

Using existing OGC (Open Geospatial Consortium) implementations of the Geo-Video Service, (Lewis, 2006), a Viewpoint data structure is introduced that extends the Aerial Video ViewCone implementation of this standard. A Viewpoint is a three dimensional construct that defines the camera's capture point location separately from a GIS data structure polyhedron of the image's geographical extent. Using a Depth-of-Field calculation to define a near and far field acceptable level of image sharpness, the Viewpoint model can be used to model the orthogonal nature of the terrestrial Spatial Video.

### 1.5.4 Two Dimensional Viewpoint Implementation

A two dimensional Viewpoint implementation is a simplified proof-of-concept experiment based on the three dimensional theoretical model. Using a number of images captured in similar circumstances to that of standard Spatial Video data streams, the model will be calculated based on the Viewpoint theory presented. A number of physical restrictions and technical assumptions have to be defined to achieve this objective. These must consider the retrospective nature of the existing Spatial Video data that are being used. This reflects the difficulty in determining the original survey system calibration parameters that define the camera’s projection matrix and/or the spatial variables positional accuracies. Thus, the implemented model is an approximation based on post-data-collection analysis, rather than having control over the data collection methodology.

### 1.5.5 GIS Database Modelling of the Viewpoint Data Structure

Implementing a fully operational Spatial Video-enabled Viewpoint database involves the registration and calculation of all camera, spatial and Viewpoint parameters into an indexed and searchable structure. Standard GIS database technologies will be used with emphasis on the types of OGC standards support and Geospatial Analysis operations that are available.

### 1.5.6 Spatial Video GIS Analysis Queries

Using point, line and polygon spatial data types to perform a number of simulation queries on the Viewpoint database, logical spatial video sequences are composed and retrieved, or spatial analysis results determined. This involves defining the search queries that will perform the database operations by properly handling the underlying semantic understanding of the Viewpoint data structure and the search space query data. These queries should return all possible video frames and sequences that the search location is concerned with or generate spatially orientated analysis results.

### 1.5.7 Viewpoint refinement based on non-video Spatial Data Queries

The design, development and implementation of a Spatial Video Viewpoint assumes a maximal, unobstructed geographical extent. Using non-video spatial data sets, specifically building profiles vector data, can the accuracy of the Viewpoint structures geographical coverage be improved?

### 1.5.8 Dynamic Spatial Video Player

Typical video players are designed to play complete video streams or contiguous parts of them. However, using the Spatial Video Viewpoints GIS database model, multiple sections of differing format video footage could be returned. This requires a dynamic video player to be developed that can play and combine the many different formats of video at only the sections contained in the query result.

## 1.6 Research Limitations

Ultimately, only two significant limitations materialised during this research. Firstly, acquisition of different types of Spatial Video data sets from the various commercial providers proved difficult. In some cases, detailed information and specifications on the data structures and spatial data accuracy was almost impossible to obtain. Obviously, these problems directly relate to commercial considerations. However, where data sets could be acquired, but spatial data structures were unknown, empirical knowledge could be acquired through simple testing and analysis of the Spatial Video data files and surveyed data content.

Secondly, the Spatial Video Viewpoint model that is theoretically developed in chapter four is three-dimensional in nature; however the base case implementation in subsequent chapters is two-dimensional. This is because the increase in implementation complexity of a three-dimensional model is computationally prohibitive when the same result can be proved in the simpler two dimensional cases.

## 1.7 Thesis Structure

This thesis consists of a further seven chapters structured and detailed as follows:

*Chapter 2* presents an overview of the video formats, spatial data standards, GIS video implementations and applications that relate to Spatial Video in general. Particular emphasis is placed on both the academic and commercial roles of Spatial Video with background development, application areas and interoperability relationships being discussed.

*Chapter 3* describes the general approaches to indexing video streams with spatial metadata as opposed to an approach of building a spatial index of a video stream. Particular emphasis is placed on an approach where video frames are indexed with GPS spatial data parameters using an embedded Spatial Video data structuring model. The model is discussed through an analysis of an existing commercial implementation of this methodology and the development of an audio software decoder system for this system. Importantly, it is shown that these models are not appropriate solutions to the main research objectives of this project.

*Chapter 4* introduces and develops the Viewpoint model as a Spatial Video GIS data structure. It will be shown how this model is a theoretical extension of existing viewable region models that include Isovist, Viewshed and Frustum structures. Viewpoints are defined as very simple GIS data structures that are calculated from a video frame’s known location and the video camera’s operational parameters. This computational form closely models the View Frustum structure used in 3D computer graphics, but is introduced into a geo-spatial domain. While this concept is a generalised and simple idea, implementing it accurately as a 3D form in a global coordinate system poses a number of considerable challenges. These challenges are discussed as part of a complete model that could be extended into 3D GIS modelling environments.

*Chapter 5* presents a Viewpoint implementation of a Spatial Video image frame into a GIS-compatible data structure based on the theoretical developments of chapter four. It will be shown how this implementation is an extension of the Open Geospatial

Consortium ViewCone as defined in their Geo-Video Service specifications (Lewis, 2006). ViewCones define very simple GIS data structures as calculated from a video frame’s known location and the video camera’s operational parameters. To construct Viewpoints from first principles, and based on extending the ViewCone model, a defining set of assumptions is discussed in relation to the retrospective Spatial Video data sets available. Also, the precise parameters that should be recorded for any ongoing Spatial Video data collection are also investigated. What this ‘base case’ implementation will use is both empirically generated and accurately collected camera and spatial properties to construct a generalised and maximal Viewpoint spatial extent on retrospectively collected data.

*Chapter 6* describes the processing procedures, problems and solutions involved in populating an implementation of a Spatial Video Viewpoints database. The spatial database used is of a Database Managements Systems (DBMS) centred approach where the Viewpoints data structure, developed in earlier chapters, defines the geometry for a Spatial Video frame indexing system. Algorithmic solutions are discussed that attempt to deal with a number of spatial data problems and are introduced and developed in terms of defining accurate operational parameter data set representations of a Spatial Video survey.

*Chapter 7* contains discussions on the semantic nature of how relevant GIS operations on Spatial Video Viewpoints should behave. It also describes implementations of practical examples based on these assertions. Spatial operations discussions will highlight a number of issues relating to the GIS functionalities introduced in chapter one and how they can be achieved through Spatial Video interaction using Viewpoints. As part of this study, a Location Based Services system, that can dynamically stream Spatial Video footage, based on the Viewpoint database model, is also discussed. This system highlights a laboratory demonstration of how a location aware video player can be dynamically controlled using Viewpoints.

*Chapter 8* discusses the main conclusions of this research by summarising the work completed, by discussing how the research questions have been answered and by detailing some future directions for extending this research area. The contributions to knowledge achieved through the application of a Spatial Video Viewpoint model are discussed, followed by some final remarks.

# Chapter Two: Video, Spatial Data and GIS

This chapter presents an overview of the video formats, spatial data standards, GIS video implementations and applications that relate to Spatial Video in general. Particular emphasis is placed on both the academic and commercial roles of Spatial Video with background development, application areas and interoperability relationships being discussed.

## 2.1 Introduction

Creating an interoperable relationship between the specific areas of Video, Spatial Data and GIS Geospatial Analysis is not a trivial exercise. Considerable commercial and academic research and investment has seen many different formats and systems develop in these areas for the acquisition and use of Spatial Video and these are now discussed.

While numerous video formats exist, none is specifically designed to deal with geographical spatial metadata tagging or indexing. Emphasis is placed on both internal and external indexing methodologies where spatial data is either stored within the video file or separately in a different associated-file format. A large number of video-related indexing systems exist that can both internally (Joung *et al.*, 2003; Tae-Hyun *et al.*, 2003) and/or externally (Ardizzone *et al.*, 1997; Jiang *et al.*, 1998) index individual video frames or sequences. However, these methods semantically describe the video content, and possibly the content geography, but not its geographical extent as a geospatial entity. Of these two distinct approaches (internal and external indexing); a discussion emerges as to what is the appropriate method for indexing Spatial Video for geospatial analysis operations. Thus, video formats and indexing in relation to data structures and metadata storage possibilities for Spatial-Video-based geospatial analysis are discussed.

Spatial Data formats are well defined in various standards development, (OGC Simple Features, 1999; ESRI, 2003; Geo Community, 2006). However, while these formats are either vector or raster based and suitable for digital imagery analysis, none facilitates the increased use and importance of digital video imagery in a GIS role. This point is not unique to GIS as video in general has not been well supported by metadata indexing, although some standardisation processes are beginning to improve this situation. Discussions of the various spatial data elements relevant to this study and the possible formats appropriate to video use in this role are included.

In general, Geographical Information Systems provide very comprehensive platforms for the analysis of geographically referenced data. A very large number of analysis techniques, methods and operations exist that can manipulate many different types of data, application systems and output requirements. However, Spatial Video is not normally included to any great extent in any of these. Only a subset of these spatial operations may ever be applicable to Spatial Video; thus, discussing GIS usage of video data in terms of its applications to both commercial and research domains that exist is included.

## 2.2 Video Formats, Standards and Indexing

While the spatial content of Spatial Video provides the critical location element for this data source, and without which the video becomes somewhat meaningless in a GI context, it is the video's ability to capture the environment visually that provides the GI enhancement. Discussed here are the Digital Video (DV) and Moving Pictures Expert Group (MPEG)2 video container formats used by the video-capturing equipment in this research. Also discussed are the video, standard and high-definition, image formats used and the conversions to alternative container formats where control-of-frame indexing is easier.

### 2.2.1 MPEG Format Conversions

MPEG, (2006), is a working group of the International Standards Organisation and the International Electrotechnical Commission, (ISO/IEC, 2006). The MPEG group is charged with defining and developing video and audio-coded standards; a number of which have been widely implemented. In particular, the use of the MPEG2, MPEG4 and MPEG7 specification schemes is considered.

In general, the MPEG2 standard is the default video container format for high definition (HD) video recording equipment and in particular for the HD video footage collected in this study. It is an incremental standard based on the MPEG1 specification and was developed to overhaul the inadequacies and inefficiencies within this standard, especially in an increasingly digital television broadcast industry. Its main drawback, for this study, is its lack of any inherent linear frame indexing scheme. This is because of the image frame compression techniques that are employed. Three types of frame compression are used in MPEG2: Intra-coded frames (I-frame), a compressed version of the original captured image; Predictive-coded frames (P-frame), where compression is further reduced based on the previous I-frame or P-frame content; and a Bidirectionally-predictive frame (B-frame), which is also further compressed but depends on both previous and subsequent I and/or P-frames. This structure is not frame-time stamped or indexed and is essentially just a stack of dependent compression encoding relationships.

To navigate to any specific frame, a calculation is required based on the video start time, frames per second and the correct handling of the frame storage bit rate. The bit rate for each frame is not a consistent value as it depends on the video content detail and its rate of static or dynamic change. This method only provides an estimated file byte location of where the required frame is and then it needs to be decoded based on many possible dependencies on any number of previous or subsequent frames. This is not typically a problem for MPEG2 footage being played in a normal linear situation over a streaming digital television channel or in another compatible media player. Even a video player seek control bar is delineated by the video byte amount, thus a point half way into the video is generally defined as the point at which half the content bytes have been viewed and not as the half way point in running time of the video.

To counter this problem it was decided to convert MPEG2 formats to the more modern MPEG4 standard. This was achieved using Xpress Pro, (Avid, 2007), which is a professional-standard video-editing software suite that has implemented most of the MPEG4 specifications. While, compression happens in a similar manner to MPEG2 it does provide an easily accessed and reliable frame index. Also, at a broader applicability level, television broadcasting standards have earmarked High Definition (HD) as the format of maximum quality going forward (Ive, 2004; Wood, 2004), with MPEG4 being the broadcast standard being most capable of delivering this quality, (OOC, 2008). In many instances MPEG4 has already been identified to replace, or has replaced, MPEG2 as the choice of video container on HD digital networks, (Wiegand *et al.*, 2003; Marpe *et al.*, 2006). As such, MPEG4 not only has a broad and future use advantage based on its specifications but also comes with an implicit linear frame index.

In this study, the MPEG2 container format footage is of HD standard, captured in 720p/25 video mode. This mode provides for an image with 720 vertical scan lines or 720 pixels high by 1280 pixels wide at 25 frames per second. It is progressively scanned where each image is a full representation of the detail as it appears on the recorder’s sensor surface. The transfer to MPEG4 maintained these video format standards and image quality throughout each video stream. The other reason for using the MPEG4 format is its direct relationship with the MPEG7 specifications. MPEG7 is not an audio and/or video encoding standard, but a multimedia content description standard. It can exist separately from the video-encoding format but directly links to it using the video’s timecode. However, it is particularly suited to MPEG4 as it can exploit this format’s space and time object-based content description representations (MPEG7, 2006). MPEG7 is discussed in chapter three based on its intrinsic Spatial Video encoding possibilities.

### 2.2.2 Digital Video (DV) Format Conversions

The Digital Video (DV) standard is the most widely used set of specifications in video capture equipment. This standard forms two parts, the codec container format and the physical storage media, i.e. DV tapes. Developed in the 1990's the DV specifications developed from a collaboration of standards used by over 60 commercial entities. DV is now used by all the major camera manufacturers as the default video data storage specification format. Although not as efficient in data storage as the MPEG standards, they have significant advantages over previous commercial formats in quality terms. At a consumer level, the DV standards have facilitated a considerable market for reasonably priced video capture equipment and playback technologies, especially at non-professional levels. This has been mainly led by the development of the mini-DV tape standard which in turn enabled development of lighter, more compact and affordable camcorders.

The DV format defines both the hardware media storage equipment and software data structure for most consumer-quality camcorders. The software data structure is based on the Society of Motion Picture and Television Engineers, (SMPTE, 2007), specification 383M (*Material Exchange Format (MXF) -- Mapping DV-DIF Data to the MXF Generic Container*, p.2), as referenced in (NDIIPP, 2008). The DV Digital Interface Format (DIF) data is stored in 80 byte blocks and is usually wrapped into a DV Audio Video Interleaved (AVI) distributable format, although other formats are possible. In this study the Spatial Video captured in this format was stored in a DV-AVI type1 format, at standard definition quality at 25 frames per second. DV-AVI video formats of type 1 store the video and audio in the original multiplexed setup but as a single stream inside the AVI file video structure. Standard definition format is 720 horizontal scan lines or 720 pixels wide by 576 pixels high, in a 4:3 aspect ratio setup.

The DV-AVI Spatial Video image formats are interlaced where each frame contains either the odd or even sensor scan lines. Essentially, no interlaced video frame image is complete in progressively scanned terms; it is only half the representation of the scene captured by that frame. This is a legacy methodology where the video image display on all analogue television broadcast networks refreshes at high enough rates that the human visual system cannot differentiate the subtle changes between different frames.

The DV-AVI distributable container format allows DV video playback on many different software platforms, especially Microsoft Windows operating systems as the AVI container format is a Microsoft-defined specification. It uses the Resource Interchange File Format (RIFF) where the file data is stored in tagged chunks. This makes the file format very adaptable to storing data that are already contained in another container format; however file sizes are nearly always larger when this is done. However, while working with this format for this study the data structure has been very difficult to generalise for efficient indexing of Spatial Video. As such, it was decided to convert this footage to another format similar to the process followed in the previous section. The DV-AVI formats were converted to Windows Media Video (WMV) format, which improved the generalisation of the Spatial Video indexing software.

The Windows Media Video (WMV) format is another Microsoft audio and video container format. Its underlying file format is defined by the Advanced Systems Format (ASF) which is principally designed for Internet media streaming. Similar to the AVI object storage system, media data are stored in serialized objects which are tagged with an identification number. This standard has been formalised by (SMPTE, 2007) in the 421M (*VC-1 Compressed Video Bitstream Format and Decoding Process*) specifications. This format and SMPTE standard are direct competitors to the MPEG4 standards in HD video distribution on both HD-DVD and Blue-Ray disk. Although the Blue-ray disk technology is now the exclusive HD video disk storage technology, only the MPEG2, MPEG4 Advanced Video Coding (AVC) or WMV VC-1 codec is supported. Thus, a choice between MPEG4 and WMV depends on other issues, such as the levels of codec support in the available user playback software, because either format is a viable existing and future technology platform for the storage of Spatial Video.

### 2.2.3 Waveform (WAV) Audio Format

This format is used as the base container format for the decoding work performed in chapter three. It is an audio-only container format and is used to store the spatially encoded audio streams that are embedded in the DV-AVI Spatial Video format files. It is a Pulse Code Modulation (PCM) bit stream format where the data structure adheres to the Resource Interchange File Format (RIFF), (Microsoft, 1992, 2007). Audio data and meta-information are stored in tagged chunks with the specifications having been jointly developed by Microsoft and IBM, (Wilson, 2003). One advantage is its simple data structure which allows software to be easily developed to decode audio embedded spatial data from Spatial Video files.

### 2.2.4 Video Format Discussion and Problems

A conversion of WMV to MPEG4 is also possible, as in the previous section; however MPEG4 is not, as yet, a widely supported codec on software and hardware video players. While it is optimised for HD video and can handle SD video perfectly, its lack of existing support on a broad range of implementation platforms decided a WMV solution should also be considered for inclusion in this study. Alternatively, converting MPEG4 to WMV is also possible through the WMV-HD format; however this format is essentially an MPEG4 implementation anyway so another conversion was deemed unnecessary.

Many other video container formats exist that are usually proprietary in nature and are designed for specific technological situations. Any of these formats would probably suit the objectives of this study; however they would require bespoke implementations of the concepts, theories and solutions. Converting to every possible format and then testing and implementing the objectives were beyond the scope of this study.

Both MPEG2 and DV-AVI file formats were used as the initial Spatial Video storage formats based on the codec container formats used in the respective video equipment. A number of issues arose when initially implementing these formats as the frame indexing structures for different video content in the same container formats was not consistent or easily navigable. In the development of a normal video player this is not a problem as frame rendering can be controlled based on a Decode Time Stamp (DTS), Presentation Time Stamp (PTS) and player clock reference. This process is important for correctly synchronised video and audio playback. In the case of Spatial Video, the objective is to control the frame images based on location-based spatial indexing. Thus, a simple methodology of linear indexing was desirable for queries and operations performed in this study. The conversion to a MPEG4 or WMV format negated these synchronisation control mechanisms as these formats provided a linear frame indexing. Table 2.1 provides a diagrammatic view of these conversions. A Spatial Video query would return a list of result frames that could be ordered and easily located in the appropriate video file.

| **Video Format** | **Original File Container** | | **Converted File Container** | |
|---|---|---|---|---|
| | File Extension | Codec Standard | File Extension | Codec Standard |
| High Def. | *.mpeg | MPEG 2 | *.mp4 | MPEG 4 |
| Standard Def. | *.avi | Microsoft RIFF | *.wmv | Microsoft ASF |

Table 2.1. Spatial Video file format conversion details.

Sample comparisons of identical Spatial Video file frame indexing are shown in tables 2.2 and 2.3. These tables show the same Spatial Video content sections just analysed in different container formats. On the left is the original equipment capture format while on the right is the converted one. In Table 2.2 the MPEG2 format has a frame index based on the DTS column, but only where the DURATION was 3600 and this only pointed to a byte location that required consideration of the PTS value. A simple single solution was not achieved for this format; however converting to MPEG4 provided a very simple linear model where the PTS value was incremental from 1 to the end of file frame where the DURATION is equal to 1.

| Time | MPEG2 Format | | | | MPEG4 Format | | | |
|---|---|---|---|---|---|---|---|---|
| | DTS | PTS | DURATION | SIZE | DTS | PTS | DURATION | SIZE |
| 48:06.8 | 30321 | 30321 | 2160 | 672 | 0 | 0 | 1 | 17577 |
| 48:06.8 | 32481 | 32481 | 2160 | 672 | 0 | 0 | 0 | 1000 |
| 48:06.8 | 34641 | 34641 | 2160 | 672 | 0 | 0 | 0 | 18 |
| 48:06.8 | 26722 | 30321 | 3600 | 98885 | 0 | 0 | 0 | 16 |
| 48:06.8 | 30321 | 41121 | 3600 | 93859 | 1 | 1 | 1 | 8323 |
| 48:06.8 | 33921 | 33921 | 3600 | 78723 | 2 | 2 | 1 | 9078 |
| 48:06.8 | 37521 | 37521 | 3600 | 64611 | 3 | 3 | 1 | 10937 |
| 48:06.8 | 41121 | 48321 | 3600 | 62548 | 4 | 4 | 1 | 11118 |
| 48:06.8 | 44721 | 44721 | 3600 | 51387 | 5 | 5 | 1 | 10719 |
| 48:06.8 | 48321 | 51921 | 3600 | 97689 | 6 | 6 | 1 | 10493 |
| 48:06.8 | 51921 | 62721 | 3600 | 87263 | 7 | 7 | 1 | 10455 |
| 48:06.8 | 55521 | 55521 | 3600 | 77892 | 8 | 8 | 1 | 9879 |
| 48:06.8 | 59121 | 59121 | 3600 | 83057 | 9 | 9 | 1 | 9057 |
| 48:06.9 | 36801 | 36801 | 2160 | 672 | 10 | 10 | 1 | 7938 |
| 48:06.9 | 38961 | 38961 | 2160 | 672 | 11 | 11 | 1 | 7708 |
| 48:06.9 | 41121 | 41121 | 2160 | 672 | 12 | 12 | 1 | 9233 |
| 48:06.9 | 62721 | 73521 | 3600 | 85538 | 13 | 13 | 1 | 7677 |
| 48:06.9 | 66321 | 66321 | 3600 | 68266 | 14 | 14 | 1 | 7757 |
| 48:06.9 | 43281 | 43281 | 2160 | 672 | 15 | 15 | 1 | 6560 |
| 48:06.9 | 45441 | 45441 | 2160 | 672 | 16 | 16 | 1 | 7294 |
| 48:06.9 | 47601 | 47601 | 2160 | 672 | 17 | 17 | 1 | 7130 |
| 48:06.9 | 69921 | 69921 | 3600 | 70320 | 18 | 18 | 1 | 7199 |
| 48:06.9 | 73521 | 80721 | 3600 | 88799 | 19 | 19 | 1 | 6147 |
| 48:06.9 | 49761 | 49761 | 2160 | 672 | 20 | 20 | 1 | 5741 |
| 48:06.9 | 51921 | 51921 | 2160 | 672 | 21 | 21 | 1 | 6773 |
| 48:06.9 | 54081 | 54081 | 2160 | 672 | 22 | 22 | 1 | 6154 |
| 48:06.9 | 77121 | 77121 | 3600 | 67051 | 23 | 23 | 1 | 6206 |

Table 2.2. Spatial Video frame index comparison of default MPEG2 format and converted MPEG4 format.

The same situation is shown in table 2.3 where no single simple solution to storing and navigating through a linear frame format was achieved for the AVI file type. No consistency in the PTS, DTS or DURATION relationship was determined where a reliable frame controlled player could be developed. Conversion to the WMV format did provide a simpler linear control; where the DURATION equals 0 all frames incremented by a value of 40 based on the first frame’s value. As an example in table 2.2, the first frame is 1579; therefore frame 200 is 1579 + (200 * 40) which equals frame number 9579. This method works reliably across all the Spatial Video files in this format.

| Time | AVI Format | | | | WMV Format | | | |
|---|---|---|---|---|---|---|---|---|
| | DTS | PTS | DURATION | SIZE | DTS | PTS | DURATION | SIZE |
| 53:46.4 | 0 | 0 | 1 | 144000 | 1579 | 1579 | 186 | 3716 |
| 53:46.4 | 0 | 0 | 1201 | 5124 | 1718 | 1718 | 186 | 3716 |
| 53:46.4 | 0 | 0 | 1201 | 5124 | 1904 | 1904 | 186 | 3716 |
| 53:46.4 | 1 | 1 | 1 | 144000 | 1579 | 1579 | 0 | 9207 |
| 53:46.4 | 1200 | 1200 | 1201 | 5124 | 1619 | 1619 | 0 | 1276 |
| 53:46.4 | 1200 | 1200 | 1201 | 5124 | 2089 | 2089 | 186 | 3716 |
| 53:46.4 | 2 | 2 | 1 | 144000 | 1659 | 1659 | 0 | 4111 |
| 53:46.4 | 2401 | 2401 | 1201 | 5124 | 1699 | 1699 | 0 | 8827 |
| 53:46.4 | 2401 | 2401 | 1201 | 5124 | 2229 | 2229 | 186 | 3716 |
| 53:46.4 | 3 | 3 | 1 | 144000 | 1739 | 1739 | 0 | 4920 |
| 53:46.4 | 3602 | 3602 | 1201 | 5124 | 1779 | 1779 | 0 | 13194 |
| 53:46.4 | 3602 | 3602 | 1201 | 5124 | 1819 | 1819 | 0 | 57 |
| 53:46.4 | 4 | 4 | 1 | 144000 | 1859 | 1859 | 0 | 10991 |
| 53:46.4 | 4803 | 4803 | 1201 | 5124 | 1899 | 1899 | 0 | 1441 |
| 53:46.4 | 4803 | 4803 | 1201 | 5124 | 1939 | 1939 | 0 | 263 |
| 53:46.5 | 5 | 5 | 1 | 144000 | 1979 | 1979 | 0 | 114 |
| 53:46.5 | 6004 | 6004 | 1201 | 5124 | 2019 | 2019 | 0 | 95 |
| 53:46.5 | 6004 | 6004 | 1201 | 5124 | 2059 | 2059 | 0 | 93 |
| 53:46.5 | 6 | 6 | 1 | 144000 | 2099 | 2099 | 0 | 94 |
| 53:46.5 | 7205 | 7205 | 1201 | 5124 | 2139 | 2139 | 0 | 93 |
| 53:46.5 | 7205 | 7205 | 1201 | 5124 | 2179 | 2179 | 0 | 94 |
| 53:46.5 | 7 | 7 | 1 | 144000 | 2414 | 2414 | 186 | 3716 |
| 53:46.5 | 8406 | 8406 | 1201 | 5124 | 2507 | 2507 | 186 | 3716 |
| 53:46.5 | 8406 | 8406 | 1201 | 5124 | 2647 | 2647 | 186 | 3716 |
| 53:46.5 | 8 | 8 | 1 | 144000 | 2219 | 2219 | 0 | 27331 |
| 53:46.5 | 9607 | 9607 | 1201 | 5124 | 2259 | 2259 | 0 | 13938 |
| 53:46.5 | 9607 | 9607 | 1201 | 5124 | 2299 | 2299 | 0 | 1613 |

Table 2.3 Spatial Video frame index comparison of DV-AVI format and converted WMV format.

## 2.3 Video Spatial Data

The spatial data aspects of Spatial Video are discussed in this section. All types of video could be considered spatial as they invariably capture space to some extent. Theoretically, any generalised video data set could be converted to Spatial Video formats by some form of location association to each frame. This study specifically considers Spatial Video which has been tagged with a frame location relationship at the time of data capture. Also, the Spatial Video footage has been captured from moving platforms travelling along terrestrial road networks. Outlining these spatial data sources relevant to Spatial Video and describing the captured base set of parameters is included. These parameters are collected from civilian standard GPS receivers where the spatial data are encoded into the video's audio signal. The other

GPS source described is Real Time Kinematic (RTK) GPS that was used in the test data sets described in chapter five.

### 2.3.1 Global Positioning System (GPS)

The primary and central technology for acquisition of geographical location information is the Global Positioning System (GPS). This is a Global Navigation Satellite System (GNSS) developed and maintained by the United States Air Force for the Department of Defence. There are approximately 25 to 30 operational GPS satellites in orbit at any one time. Terrestrial location is determined by precise measurements of each satellite’s timing signals, its orbital position and the approximate orbits of the other satellites in the network. It is possible to ascertain a terrestrial location using a GPS receiver with a minimum of three signal viewable satellites, but this would be to a high degree of error unless augmented. At least four satellites are required to achieve accuracy levels of five to seven meters on average (Tiberius, 2003). Many GPS augmentation techniques have been developed to improve this level of accuracy, in some cases to millimetre values.

In this study, standard GPS receivers recorded the Spatial Video frame location data at one hertz frequency. Dedicated commercial hardware encoded these data onto the video’s audio track; a process discussed in detail later. The GPS signals were recorded to the National Marine Electronics Association (NMEA) 0183 serial data transfer sentence specification for GPS data. Two forms of these sentences were used, the $GPGGA and the $GPRMC with an example and description shown in table 2.4. These are simple, comma separated, plain text ASCII strings. The GPS receivers acquired the GPS satellite time and orbital data from which the receiver’s distance from each satellite can be calculated. The spatial location variables can then be determined and directly output through the NMEA message. Most other variables such as the speed and azimuth values are calculated on previous recorded positions during the receiver’s operation.

<table>
<tr><th colspan="2">GPRMC SENTENCE<br>$GPRMC,120241.00,A,5323.2228,N,00635.1483,W,21.8,221.6,260206,,*1A</th></tr>
<tr><td>120241.00</td><td>Coordinated Universal Time (UTC) – Atomic time standard</td></tr>
<tr><td>A</td><td>Status of Navigation receiver. A = OK</td></tr>
<tr><td>5323.2228,N</td><td>North Latitude 53° 23’ 13.36800”</td></tr>
<tr><td>00635.1483,W</td><td>West Longitude 6° 35’ 8.89800”</td></tr>
<tr><td>21.8</td><td>Speed over ground in knots</td></tr>
<tr><td>221.6</td><td>Azimuth, true north course.</td></tr>
<tr><td>260206</td><td>Date: 26 February 2006</td></tr>
<tr><th colspan="2">GPGGA SENTENCE<br>$GPGGA,120242.00,5323.2179,N,00635.1483,W,1,09,1.5,138.6,M,,M,,*69</th></tr>
<tr><td>120242.00</td><td>Coordinated Universal Time (UTC) – Atomic time standard</td></tr>
<tr><td>5323.2179,N</td><td>North Latitude 53° 23’ 13.36800”</td></tr>
<tr><td>00635.1483,W</td><td>West Longitude 6° 35’ 8.89800”</td></tr>
<tr><td>1</td><td>Quality of GPS fix,</td></tr>
<tr><td>09</td><td>Number of signal viewable satellites.</td></tr>
<tr><td>1.5</td><td>Horizontal Dilution of Position</td></tr>
<tr><td>138.6</td><td>Altitude in meters.</td></tr>
</table>

**Table 2.4. Examples of GPS NMEA message captured and tagged to the Spatial Video data streams used in this study. Only sentence variables that were used in later chapters are described.**

The GPGGA sentence defines the GPS fix data and its quality. It contains the latitude, longitude and altitude for determining a three dimensional fix. Accuracy levels are very variable as reported in this data string, especially if the height of the geoid is missing which will significantly degrade the horizontal altitude variable value. The quality of the GPS signal is also shown which can be used to determine the level of reliability in the accuracy of the location information. The GPRMC sentence defines the recommended minimum GPS location and orientation data variables. It provides two dimensional X and Y positional values as well as orientation and velocity calculations.

### 2.3.2 Real Time Kinematic (RTK) GPS

Real Time Kinematic (RTK) can be used as a GPS augmentation technique and is used in survey situations where high degrees of accuracy are required. It operates based on measurements of the GPS carrier phase rather than the data contained within the signal. The carrier phase alignment measurements are refined based on re-broadcasted phase signals from static terrestrial base stations. In Ireland the accuracy using this method is better than six centimetres when using a geodetic RTK receiver and GPRS/GSM mobile phone communication, (Bray, 2006). There are currently sixteen such base stations in Ireland that provide national RTK coverage. This coverage is provided from an older, existing, GPS ground station network that provides a web-based GPS download source for more accurate post-processing of survey GPS data.

Based on this older network, a centralised computer system maintained by Ordnance Survey Ireland (OSI) continuously receives GPS signals from each base station over a broadband network. These signals are processed in real time and relayed over the mobile phone network to any device that is enabled for RTK and is registered with and logged into the OSI RTK network. The calibrated test Spatial Video implementations in chapter five are based on this GPS data source.

## 2.4 Spatial Video in a GIS

In these sections discussion is divided into the two distinct application areas of Spatial Video: commercial and research. For this section Spatial Video is a generalised data source and not necessarily the specific version that is used in this study. The methods used to collect video with spatial data are independent of the concept of video frames being location and/or orientation tagged. Thus, what is considered is the use of spatially tagged video in any form in any GIS application area.

As previously mentioned, the majority of the applications of Spatial Video are as a visual enhancement, used for improved analysis and/or spatial reasoning, within a GIS. As such, it is generally regarded that the roots of Spatial Video stem from academic research led by Andrew Lippman at MIT in 1978 based on the Aspen

Movie-Maps project, (Lippman, 1980). This project incorporated camera imagery into a user-orientated information enhancement tool for computer presentation and interaction. A multitude of intermediate stages have seen development of many of the different aspects that this project introduced, from the types of mobile mapping systems used to capture the data through to the processing, storage, analysis, usage and presentation of Spatial Video to an end user. The current most widespread application area for spatially tagged imagery is, obviously, the Internet where Amazon released A9.maps Block View in 2005, (A9 Maps, 2006). However this service has since been withdrawn and it can only be surmised that the market dominant Google Street View, (Google, 2007), and Microsoft Live Maps, (Microsoft, 2005), has forced its demise.

The following sections will highlight the application areas and progressive development of Spatial Video between these initial investigations at MIT in 1978 and today’s Internet-based Google and Microsoft applications.

### 2.4.1 Commercial Applications

A number of commercial applications areas have developed based on Spatial Video usage within a GIS environment. Spatial Video is used by many different types of public and commercial entities for a multitude of reasons. Commercial and Public concerns that use Spatial Video include, Government and Local Authorities, Utilities contractors, Defence and Emergency services, transportation and service companies. The uses Spatial Video are put to typically involve remote management where road network asset inventories, validation and auditing, planning and engineering assessment can be preformed based on visual inspections of the environment without individuals having to be in the survey region. The systems looked at here range from high accuracy survey entities that offer dedicated, bespoke Spatial Video hardware and software systems, through to Internet-based standards and free service offerings.

#### 2.4.1.1 Routemapper

Routemapper is a marketing brand of the IBI group, a Canadian based international consultancy company, (RouteMapper, 2007). Routemapper markets consultancy, survey and software services for the collection, analysis and use of Spatial Video for road and rail survey projects. Its primary software applications are Routemapper Desktop, that is available as a Lite or Ultra version, and Routemapper interactive which is an Internet browser service that is only available to registered clients for viewing their own surveys. A three-stage process is involved where a bespoke survey vehicle acquires video and positional data for the survey area. This is followed by a data validation and quality control stage to process the video and spatial data for any errors or inconsistencies. Finally, the survey area Spatial Video can be integrated into the Routemapper software browser.

The Routemapper browser is a very powerful bespoke GIS specially designed for Spatial Video integration and analysis. Video can be controlled both temporally through normal video player style controls and spatially through a cartographic interface of relevant raster or vector data sets. Advanced photogrammetric techniques can be applied to the video footage to take real world measurements in both 2 and 3 dimensions. This detail can then be stored in spatially enabled databases for export and use in other GIS applications. The web browser version has similar visual functionality but currently lacks the more detailed database and photogrammetric possibilities. Sample images of both browsers are shown in fig. 2.1.

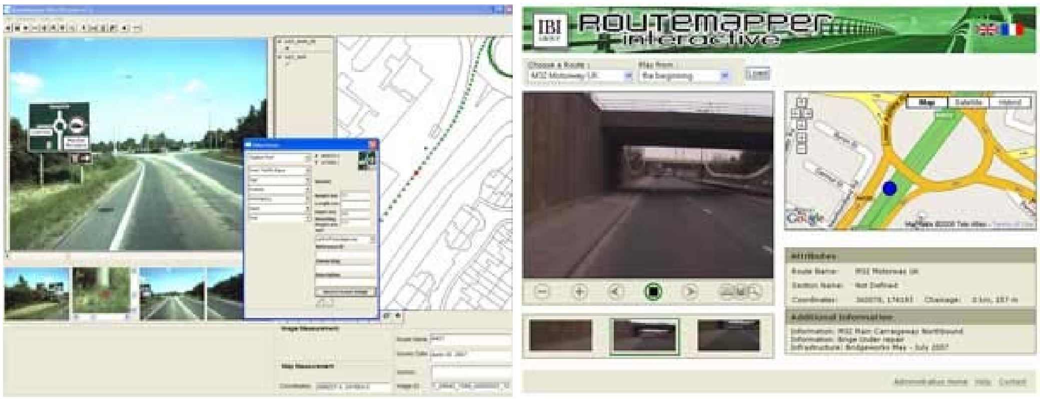


Fig. 2.1. IBI Group Routemapper desktop browser (left) and Internet browser (right).

#### 2.4.1.2 Red Hen Systems

Similar to Routemapper is Red Hen Systems who offer a complete range of Spatial Video collection, processing and analysis services and products, (Red Hen Systems, 2005). They are not a dedicated survey service, although they can perform the video surveys, but provide a full range of hardware and software solutions. Their two main Spatial Video profiles include a suite of hardware products for data acquisition and software applications for analysis and mapping. The hardware offering includes a range of GPS and video data logging devices for integration with consumer quality camcorders and GPS devices. These hardware implementations have also been patented to define custom solutions to encode video with spatial data. This includes hardware designs for miniature cameras such as those used in an Unmanned Aerial Vehicle (UAV).

Their software portfolio includes both desktop and Internet solutions. The Internet offering is known as MediaMapper Server and is similar to what Routemapper offer in that user-related spatial video can be accessed and searched. However, this is without the detailed functionality available in the desktop systems. There are two main desktop systems: MediaMapper and GeoVideo. MediaMapper is their standalone desktop solution containing a standard suite of GIS video-related controls, such as data and feature logging, temporal and spatial video searching, image and map measurement and industry standard spatial file formatting. A less powerful version is their GoeVideo software extension to the ESRI ArcGIS version eight or higher software.

One underlying but significant difference between these commercial Spatial Video vendors is that Red Hen systems can log and handle multiple video streams. They can edit and splice these different streams based on user-defined video and/or geographic sequences which can then be exported to a new video stream. Routemapper currently does not have this capability; however it can handle different types of video format as long as they are pre-processed through the validation stage where the video is frame grabbed and spatially tagged into self-contained Spatial Video sequences. Red Hen Systems requires all the Spatial Video formats to be in a DVD format which requires separate conversion procedures.

#### 2.4.1.3 OGC Geo-Video Service and Standards

The Open Geospatial Consortium (OGC) is an international collaborative organisation that collates input on a broad range of geospatial issues from more than 360 organisations that include government, private and public sectors. The core objectives are the definition of a set of standards that determine the interoperability of all aspects of geospatial data access, collection, organisation, storage, usage etc. In 2005 the OGC Web Services phase three (OWS-3) initiative defined a number of working areas that included a set of software profiles for the development and enhancement of a Geo-Decision Support Service (GeoDSS). GeoDSS was tasked with extending the ability to access and exchange geographic information across many different profiles through the use of standards specifications. Directly related to this study is one particular GeoDSS subtask: the implementation of a Geo-Video Service that can standardise access to video that includes geo-location information, (OGC OWS-3, 2005).

This service is still only in draft document stage, (Lewis, 2006), but is very comprehensive none-the-less. It defines an extensive range of service profiles from underlying architectures to access protocols and database structures to variable requirements for GeoVideo web service calculations. The set of concepts contained in this document is the most closely aligned set of specifications and implementations that complement those developed in this study. The core similarity is the geo-Video Service ViewCone concept. This is a two dimensional geometric shape that defines the viewable geographic extent or spatial extent bounding box of each frame of video within a Spatial Video file and is shown in fig 2.2. It is computed based on calibrated camera parameters and recorded spatial variables. These similarities and, more importantly, the principle differences between the OGC Geo-Video Service ViewCone and this study’s implementation are discussed in later chapters.

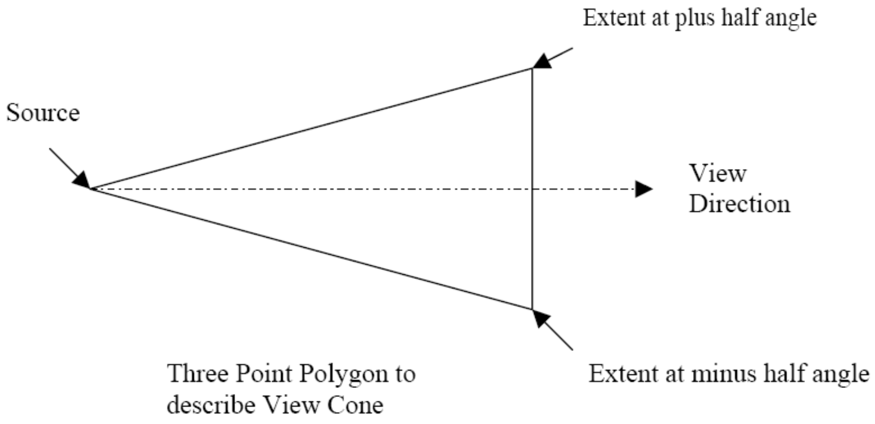

Extent at plus half angle
Source
View Direction
Three Point Polygon to describe View Cone
Extent at minus half angle

Fig. 2.2. OGC Geo-Video Service minimum spatially tagged video frames ViewCone construct.

This work was undertaken by Intergraph Corporation in 2005/2006 as part of the OGC working group on the Geo-Video service. At the time of writing, Intergraph Corporation has undertaken the development of these standards into a commercial product. This product will implement the concepts that are central to these standards and a number of others related to separate Spatial Video topics. Some of these other topics have also been considered in this study and include the areas where spatial data are internally coded to the video data.

### 2.4.1.4 Immersive Media

Immersive Media, (2006), have developed a hardware and software tool set called Immersive Video. The video data collection hardware system is known as Dodeca 2360 and comprises 11 camera lens embedded in a single unit with a 360° horizontal and 290° vertical coverage. This camera system captures video data at 30 frames per second that can be post-processed through an automatic mosaic application to any desired output frame rate. The output data format can support multiple types of metadata tagging, including geo-spatial data tagging. Two software applications are available in a similar fashion to Red Hen Systems. One is a bespoke desktop application that can play back any video footage while a standard GIS interface can load other spatial data to augment the video view. The other toolset is a software extension plug-in for ESRI’s ArcMap application. The video player is completely dynamic where the mouse can be used to alter the viewing angle as the video is playing.

The most notable use of this form of Spatial Video is through the Street View Google Maps interface, (Google, 2007). Immersive media were originally contracted by Google to survey a number of US cities. However, Google have since acquired the necessary equipment and now manage the data capture themselves. Extensive investment in data collection, post-processing and GIS interface development has brought actual street level digital imagery to both Google Earth and Maps. Based on hundreds of survey vehicles driving through major urban centres around the world, video in thousands of cities has been captured and is now available online free of charge from the Google maps products. Fig.2.3. shows an example of the Dodeca 2360 hardware, its capabilities and Interface implementations.

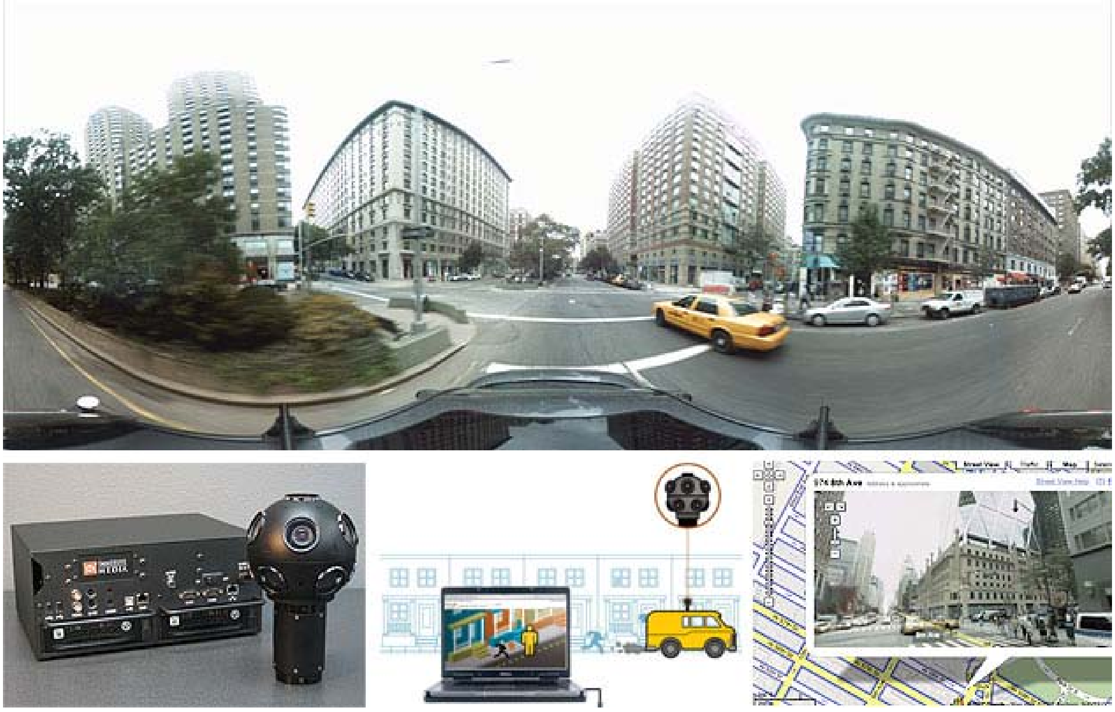


Fig. 2.3. Immersive Video 360° implementation of Spatial Video and its uses in relation to the Google Maps Street View

### 2.4.2 Research Applications

In this section academic contributions to the development and use of Spatial Video are considered with particular concentration on data structuring and GIS interfacing. Spatial Video data collection is not a concern here as the multiple methods and techniques of capturing video and spatial data are incidental to the methods of indexing, searching or using them in a GIS. While a large amount of literature exists for the multiple mobile platform methods that have been developed for the collection of Spatial Video, along with multiple algorithms and techniques for post and real time video frame-to-spatial data indexing, no significant amount of work has been completed that considers a broader theoretical or practical GIS context for the Spatial Video data generated. No single piece of academic work identifies Spatial Video as the data source from which a generalised data structure or set of spatial operations can be defined.

Three significant points of view should be considered in relation to the literature relevant to this study:

1. The methods of indexing and storing video with spatial data.
2. A theoretical model for Spatial Video in a GIS, particularly three dimensional.
3. The use of Spatial Video concepts in GIS-based operational queries.

Research on these topics overlaps in many instances although some research is self-contained and only relates to GIS modelling or video frame spatial data indexing.

O'Connor *et al.*, (2008), have implemented one specific example of a methodology for the storage, indexing and retrieval of video based on spatial metadata. They highlight a system where the MPEG7, (MPEG7, 2006), and MPEG21, (Bormans *et al.*, 2002), video file multimedia metadata standards implementations are used to provide a complete and extensible video frame indexing system. By using these standards, not only can spatial data be associated with each frame, but multiple types of metadata can extend the searchable functionality of the video streams that are defined. This point has been considered in this study as two distinct approaches where an internal video file indexing or centralised database indexing is more appropriate to requirements.

O’Connor *et al.*, (2008), also develop a user interface to query a Spatial Video database. However, only where a GPS tag has been recorded is the video key frame indexed so video images form the indexing control for return-of-video sequences. Also, the spatial queries involving region based operations only return all the video key frame images inside the region as defined by the GPS location of where the image was taken. Based on the Spatial Video data structure implementation here and redefining some spatial operations semantics, this type of spatial operation has a more precise meaning. This has enabled the system determine the difference between a video frame that was captured within a region but does not visualise it from one that visualises a region but may not have been captured within it.

Nobre *et al.*, (2001), is one of the first pieces of research to introduce the notion of a geographical space being captured in each video frame image where a GIS data structure can be used to model  this space. In this case a decision support system is developed for retrieval of video sequences based on user interest spatial queries. This system is heavily dependent on manual user calibration based on visual image analysis. Captured video is geo-referenced based on GPS data, followed by equal division of the line that the video traverses to represent the points where each frame is located. Each frame can then be queried and manually geo-referenced to determine the view frustum object space. This is based on manual adjustment of key images that are calibrated based on visual inspection of real world object projections onto the image plane. Using this methodology an accurate measurement of the camera frame object space can be achieved based on arbitrary calibration. This process assumes static camera conditions, i.e. no automatic focus or change in zoom. This assumption is also made here as internally stored video change parameters do not exist in existing video data file structures.

VideoGIS is a system defined in work by Navarrete *et al.*, (2002), where Spatial Video segmentation is based on geographical content segmentation. A data schematic, process and structuring is described which includes details of implementations based on OGC standard GIS data structures. However, no detail is provided as to the automatic creation or usage of these data structures in a Spatial Video context. Both this piece of work and Nobre's systems touch on some of the core concepts in the development of a Spatial Video GIS query data model. They introduce either the concept of modelling video frame object space as a geographical extent or using GIS data structures for this purpose; however the objectives of these papers are not to define these points in detail as only sparse information of their implementation, structure or use is presented.

A number of papers from the Electronics and Telecommunications Research Institute (ETRI) in Korea detail a VideoGIS system called GeoVideo. The literature defines an incremental development of firstly, a systems specification, (Kyong-Ho *et al.*, 2003A) for a Spatial Video system; secondly, a data structure for metadata tagging of the video (Tae-Hyun *et al.*, 2003); and finally, an implementation of the concepts in a mobile location based service, (Kyong-Ho *et al.*, 2003B; Qiang *et al.*, 2004). In these papers they present a final product called MediaGIS where a fully implemented and complete system from the point of data collection through to data distribution to end users is detailed.

The metadata spatial storage mechanisms use the MPEG7 data structures for video frame annotation which includes the spatial variables. Work also performed at ETRI specifically developed an implementation of this data structure based on MPEG standards in (Joung *et al.*, 2003). Centralised 3D databases form the backbone of the spatial queries that return the relevant image or Spatial Video sequence. Upon a successful user query, a viewing frustum can be assumed based on pre-processed image spatial and orientation data where the perspective projections for transfer from 2D image space and 3D object space are calibrated in relation to existing 3D city models. This study does not assume availability of such rich data sources and as such only assumes and improves the viewing frustum based on empirical testing and modelling.

In Hirose *et al.*, (1998), an interactive system of video imagery navigation has been completed based on a multi-view Spatial Video data collection, processing and query system. This work has since generated quite a number of extension research projects that use multiple cameras to capture Spatial Video which can be played back in dynamic video players where the viewpoint control is only limited by the degrees of freedom in the video capture system. In Hirose, a 360° horizontal viewing system is possible based on eight cameras calibrated with positional and orientation sensors. In further work such as in Neumann *et al.*, (2000), the same result is achieved with an array of digital sensors instead of individual cameras. Ultimately, this sort of work has lead to the Immersive Video systems highlighted in the previous section.

Highlighted in Cho, (2007), is the ability to define accurately a camera's location based on the calculation of its viewing frustum. This work constructs 3D imagery from 2D camera pictures without the spatial location of the camera being known. This location can be determined by solving a number of well-documented systems of equations in computer vision, (Hartley *et al.*, 2003). In this work, this is achieved based on at least six reference points that relate 3D Lidar data to 2D image points. Such a calibration can define the viewing frustum parameters to back calculate the camera location. Conceptually, this study reverses this process as the camera location is known and the camera parameters can be assumed to an acceptable error range. However, if Lidar data were available for the video sequences captured, the procedures in this paper would produce very accurate viewing frustums for each video frame.

### 2.5 Conclusions

This chapter has discussed and detailed the role of various video container formats, spatial data and GIS usage of Spatial Video. The discussion on video container formats avoided listing the multiplicity of software data formats available for the storage of video data. Instead it concentrated on the core physical process of container format usage; basically the format defined by the video capture equipment is pre-determined by the hardware and not user or usage considerations. It is the following steps that are important where a suitable data format is used that provides the required system functionality in the subsequent workflow. Two types of video capture equipment are used, providing two types of output video format both of which were converted to an MPEG4 and WMV format, respectively, to provide logical and easier video frame control in the subsequent spatial data implementation work.

Spatial data were considered in terms of the data collected and availability to this study. In later sections of this thesis more detailed discussions are developed on these topics that relate to various problems and errors that were encountered in GPS track data usage. Finally, overviews of the various application areas that Spatial Video has been developed for, from both a commercial and academic perspective, are considered. These discussions provide the context for the work described in this thesis based on the following core concepts:

- Methodologies towards internal or external frame indexing with metadata, specifically spatial data decoding from video audio streams.
- Methodologies for describing the geographical spatial extent of a video frame image that can be extended to describe complete video streams.
- Conceptual implementation of GIS operations based on Spatial Video. This covers a range of operations such as: in what way can this spatial data structure be improved? And what constitutes logical queries or usage of Spatial Video?

# Chapter Three: Spatial Video Data Structures

This chapter describes the general approaches to indexing video streams with spatial metadata as opposed to an approach of building a spatial index of a video stream. Particular emphasis is placed on an approach where video frames are indexed with GPS spatial data parameters using an embedded Spatial Video data structuring model. The model is discussed through an analysis of an existing commercial implementation of this methodology and the development of an audio software decoder system for this system. Importantly, it is shown that these models are not appropriate solutions to the main research objectives of this project.

## 3.1 Introduction

Two areas of work are developed in this chapter, both relating to internally indexed Spatial Video. In the first section current standards and implementations are discussed where spatial data are stored with or within the video capture file format at the time of collection. This topic is examined in relation to its obvious advantages where less post-processing is required to control and store the various video and spatial data files from multiple surveys. Also discussed are the disadvantages of this methodology in relation to the broader objectives of the study where multiple Spatial Video data streams can be indexed and searched independently of the video file format. A number of standardisations exist for generalised video metadata indexing; however only those related to spatial data indexing are highlighted.

In the second section is described an existing commercial application that indexes video files with spatial data. This system is based on a hardware encoder and decoder developed by NavTech Systems in the United Kingdom, (NavTech, 1995). This hardware system encodes GPS NMEA data into the audio channel of the video file; it will also decode the spatial data in a post-processing procedure when connected to a computer. A software version of the hardware decoder is developed as a novel part of this study and is also described in this chapter.

### 3.2 Spatial Data Indexing of Video

In this section four distinct approaches to creating a video stream with a spatial data index are described. These approaches are based on a study of video format technical specifications and standards, where some limited examples of actual software implementations involving spatial data have been included. The approaches are:

1. Encoding the spatial data into the video file audio track through the use of dedicated hardware.
2. Encoding the spatial data into the video file based on SMPTE standards that define Key Length Value (KLV) metadata formats.
3. Based on MPEG7 video file metadata standards, spatial data can either be stored in a file associated with the video or embedded within the video file dependent on its format.
4. Storing the location data in a separate spatial data file associated with the video.

These approaches are discussed in terms of both actual and theoretical applications where these methods are used in commercial and research roles. They are discussed at an abstract level of applicability rather than at low levels of technical detail. This is because a technical implementation of any approach would probably be bespoke based on the system requirements. Also, getting actual implementation details and data structures for existing methods was impossible in the case of the commercial and military systems; however certain knowledge can be assumed about these systems based on their underlying schemas and standards.

The four approaches fall into three types of data structuring categories (table 3.1). Type 1 has known commercial and military implementations; type 2 has been used in academic work, and was mentioned in chapter two; while type 3 is theoretically introduced here as a standardisation approach for other known indexing methodologies. The principal concern of these approaches is linking the spatial variable to a video stream; however the accuracy of this link is not always high. This stems from the difference in signal frequencies between the video stream and the spatial variables. Video capture frequencies are usually in the range of 24 to 60 frames per second while spatial variable frequency is dependent on the spatial sensor; in the case of civilian GPS this is at one or two hertz. For high accuracy survey systems this frequency rate relationship has to be tightly calibrated; however most systems do not require this level of detail and thus accept a higher level of spatial error.

| Type No. | Video spatial data indexing method. | Technical result of each method. |
|---|---|---|
| 1 | • Audio Track format.<br>• SMPTE KLV format. | Embedded within the video file container format. |
| 2 | • MPEG7 format. | Can be embedded in a MPEG4 format file or stored as a separate file for other video formats. |
| 3 | • Associated video file format. | Separately indexed file based on arbitrary video frame links. |

Table 3.1. Resulting data structure effects of the four methods of spatially indexing video files.

### 3.2.1 Audio Track Indexing

This method of indexing serves two important purposes: to combine the spatial data with the video footage and to provide frame location synchronisation. Introduced in chapter two are Red Hen Systems and NavTech Systems who use this methodology, (NavTech, 1995; Red Hen, 2005). They build dedicated hardware to enable GPS data to be stored in the video files audio stream. Also from chapter two, RouteMapper, (2007), can interface its Spatial Video browsers based on video captured using the NavTech GPS hardware encoder. However, this low cost approach has a trade off of low accuracy; nevertheless other aspects of their business can survey and implement higher levels of accuracy based on systems using different bespoke methodologies. Ostensibly this methodology is a simple way of storing and combining the collected spatial data into the same file as the associated video. For this study this was achieved through dedicated hardware that encodes the GPS signal, sampled at one hertz, onto the video camcorder’s audio track (NavTech CamNav, 2004). This is discussed in more detail in section 3.4.

Using this methodology, video frame-to-spatial location synchronisation accuracy is achieved in a number of ways and is usually dependent on the survey requirements. At its worst, synchronisation uncertainty is a function of two systems: the level of GPS error and the spatial location-to-video frame signal delay offsets. The GPS error uncertainty is dependent on many other variables separate from the audio encoding process, such as multipath effects, low quality GPS signal, etc. and can only be improved by the choice of GPS source. Thus, augmented GPS such as Differential GPS (DGPS) or Real Time Kinematic (RTK) GPS would immediately improve the accuracy of the spatial data during the encoding stage. Also, and similar to this topic, is an awareness of the physical location of the spatial sensor and the video equipment image plane offset. While variable levels of spatial accuracy may be achieved for the GPS sensor, it is measuring the location of the image plane that is important and as such the final calculations should contain an offset variable to adjust the spatial accuracy accordingly.

The internal video file structure contains an audio signal synchronisation that relates an audio and video image timestamp; however, due to signal propagation errors this relationship is not reliable. This type of signal propagation error is systematic to the encoding process, if it is quantifiable, and as such can be reliably adjusted for its effect at a post-processing stage. Propagation errors are both algorithmic and physical in nature. Definable CPU clock cycles are required to convert the GPS ASCII data sentences into the audio stream byte orders. Also, there is a measurable delay in the capture, transfer and writing of these spatial data across the electrical circuitry and serial data transfer cables. In McCarthy (1999) an analysis of these error rates is given.

Thus, this methodology is predominantly an initial spatial data storage mechanism that has many inherent synchronisation errors in the audio byte ordering of the spatial data and the correct video image ordering. In the post-processing stage many different techniques can be applied to deal with these problems. The spatial data can be decoded separately from the video and processed to improve their accuracy based on any number of techniques from simple visual point location verification and editing through to more sophisticated differential correction, track smoothing or point interpolation. Also, based on a calibration of the video timestamp with the GPS signal time, a post-processing correlation can be defined and used to improve the system's final synchronisation accuracy.

This methodology has a number of general and distributed usage drawbacks. Firstly, in the RouteMapper implementation, the spatial data are processed, post-data capture, into a separate text file which has to be maintained in a specific directory file structure in relation to the associated video file. This structure is then recorded by the mapping and visualisation software for reference when future playback, analysis and viewing are required. Thus, when using another Spatial Video browser, with any specific Spatial Video file, its distribution is hindered by either performing another processing step to produce the spatial data file again or porting the details of the previous spatial processes details into the next browser's settings. Secondly, this also means that any of the synchronisation and calibration parameters mentioned earlier, that have been calculated for any specific Spatial Video process stage, have to be reproduced as they are not stored in the audio stream indexing procedure.

Ultimately the accuracy and applicability of this methodology is determined by the post-processing stages. The lack of post-processing will create a usable but inaccurate Spatial Video stream that has acceptable spatial detail and applicability at certain levels of cartographic scale. More intelligent and sophisticated levels of post-processing will result in much higher levels of spatial accuracy which are normally not relevant except in survey grade requirements situations.

### 3.2.2 Spatial Data File Associated with a Video Stream

The indexing methodology where a spatial data file is associated with a video stream can be formalised through the implementation of a merged set of standards based on existing GIS and subtitling file format specifications. The basis for this proposal is the implementation diversity of this methodology where the spatial data for a Spatial Video sequence is stored in a bespoke manner that has a logical association to the video structure. In most Spatial Video bespoke applications this sort of methodology is implemented where the spatial data file is generated from a post-data-capture processing technique. Many different examples of this process exist with the usual objective being to produce a spatial data file, in a GIS enabled format, that has bespoke properties relating to the original video stream.

Two contextual representations are introduced here that both involve the same objectives but from different perspectives. Current Spatial Video systems either tag video frames with spatial data or tag spatial data with video frame details, both in separate files. Neither perspective supports one aspect without sacrificing the inherent data semantics of the other. Thus, where GIS data standards can define file formats for representing the video's spatial data through such data types as points, lines, polygons, etc., these can be considered and handled as a homogeneous GIS data structure. Metadata, such as video frame details, can then be associated with each spatial entity to define the link between the video and spatial data. However, this forms a heterogeneous video format link as no video systems will inherently understand this association. Alternatively, video file description standards can define frame level timing descriptors where metadata can be tagged to these descriptors. This process would define a format where the metadata describes the spatial variables while the descriptor tag determines the associated video file frames or sequences. In this situation the associated video frame data are stored in a well-understood format that is homogeneous to video file systems but heterogeneous to the spatial data GIS structures.

The point data type, (Geddes, 2005), is the simplest GIS data structure to be used when representing the Spatial Video indexing scheme. It is typically this data structure that stores the video frame spatial location information and can be defined, very simply, through any standard table or spreadsheet implementation. Ultimately, this structure is usually converted to an ESRI shape file that is well-supported by GIS, (ESRI, 1998). In a Spatial Video context this structure can then be populated with metadata including video stream linkage information. Normally, this is a frame number or timestamp. The disadvantage of this methodology is that a standardised relationship between the video file and its associated spatial data is not present. Only user-defined metadata are stored in the GIS data file format, not a universally accepted descriptor list. This is the predominant approach to handling the frame location points from Spatial Video streams when separate spatial data files are implemented.

The context for recommending an improvement in standardisation of spatial data and video file linkage is based on video subtitling practices. While different sections of this thesis discuss video metadata structures that can define frame level video properties, these are not orientated towards a separate video metadata-structure. In this context it is proposed that the spatial data structures adhere to a formalised understanding of their video file linkages. Two recognised sets of standards provide a methodology where timing intervals are the principle descriptor for text subtitling in video files. A subtitle file format is defined in Waters, (1991), where Text and Timing Information (TTI) blocks can be used to detail a video file’s subtitle content. Adams, (2006), defines a Distributed Format Exchange Profile (DFXP) for the transcoding and exchange of Timed Text (TT) over distributed systems. These standards and file structures provide a valid methodology for indexing video with spatial variables based on the link between the frame and its capture location. This would then provide a fully compatible file format that a video system would understand; however it would not be easily usable in a GIS.

The problem is highlighted in the previous section where it was mentioned that video and spatial data are usually sampled at differing rates, thus a number of possible video and spatial data linkages could occur. These range from indexing every video frame with a captured or interpolated spatial variable or indexing a spatial variable with a range of video frames or over a sequence. The level of required spatial accuracy will usually define the method, but the later approach would tend towards an inaccurate Spatial Video representation suitable only for situations where rough visualisation is acceptable. Spatial Video captured at low velocities would affect the distance between each captured GPS point, resulting in a smaller distance travelled for a given number of video frames. This means less obvious visual change across the video frames between any two spatial locations. In higher accuracy requirements or on higher velocity capture platforms it would be more appropriate to index each frame with interpolated spatial data as the distance per frame ratio will be larger.

An actual implementation could be based on either a standards implementation of video structures into a GIS file format or GIS data types into video subtitling formats. The perceived implementation would centre on a GIS data format providing specifications for a set of TTI or TT type standards. This is based on the spatial data aspect where the video frame link is predominantly for GIS application analysis and visualisation reasons. This would maintain well-recognised GIS data format compatibilities with existing GIS systems as well as providing a standardised video file-frame timing relationship. Video systems could then be adapted to utilise the subtitling formats as contained in the GIS data file structure as they conform to recognised standards.

### 3.2.3 Key Length Value Metadata Format

The applicability of this methodology is based on military applications where video spatial parameters are collected and recorded in real time into a dedicated video output file format. Primarily based on the SMPTE (2007) Key Length Value (KLV) data encoding protocol, amongst others, both the NATO STANAG (1995) and MISB AMIWG (2000) standards organisations have defined methodologies for encoding spatial data into video streams. The importance of video imagery in military related situations is increasing rapidly, especially with the greater usage of unmanned vehicles that relay operational data to remote operators. This has lead to a development and implementation of the KLV protocols to store all relevant sensor data for distribution and analysis in intelligence, surveillance and reconnaissance roles.

As mentioned, these are military related organisations where NATO STANAG (1995) is an international cooperation that ratifies numerous sets of protocols and procedures for common distribution of technical military requirements. Each signatory military organisation implements the centrally ratified standards independently, based on requirements. They range in objectives from technical specifications of software, hardware and systems to procedures for administrative and logistical communications and organisation. STANAG 4609 (2007) is a specifications guide that defines the protocols for the implementation and distribution of motion imagery. In section 3-1 of this document it expressly states the importance and role of Spatial Video where *‘the difference between commercial domain and Intelligence, Surveillance and Reconnaissance (ISR) applications is the vital importance of dynamic geo-localisation metadata’*. Also, in this document’s appendix an application note considers the more detailed implementation of this methodology on the United States Predator Unmanned Aerial Vehicle (UAV).

MISB AMIWG (2000) is a working group of the Motion Imagery Standards Board which was directly created as the standards implementation authority for all motion imagery captured and used by the United States (US) Department of Defence (DOD). Specifically relating to Spatial Video, it is well known for its original analogue 104.5 Engineering Guidelines (EG) for video and spatial data integration on the Predator UAV. These have since been developed into formal standards that extend to digital motion imagery in the form of the (MISB, 2008) 0601.2 Unmanned Air System (UAS) data-link local metadata sets. This standards document draws from a number of previously tried and tested system specifications that include:

- Engineering Guideline 104.5 – Predator UAV Basic Universal Metadata Set.
- SMPTE 336M-2001 – Data Encoding Protocol Using Key-Length Value.
- SMPTE 335M-2001 – Metadata Dictionary Structure.
- MISB Recommended Practice 0605 – Inserting Timecode and Metadata in High Definition Uncompressed Video.
- MISB Recommended Practice 0103.1 – Timing Reconciliation Universal Metadata Set for Digital Motion Imagery.

A KLV data protocol defines a methodology for embedding metadata in a video file. It is a binary data stream format where a key determines the data segment, the length specifies the amount of metadata and the value holds the metadata bytes. In its original implementation the Predator UAV used a calibrated system that can process aircraft telemetry, video camera, GPS and Inertial Measurement Unit (IMU) data into a single KLV formatted MPEG-2 or MPEG-4 Transport Stream (TS).

This methodology provides the broadest usage of international standardisations that have been specifically developed to generate Spatial Video. Unfortunately, the precise implementations of actual systems were impossible to acquire due to their military connections.

### 3.2.4 MPEG7 Format

As was briefly mentioned in chapter two, MPEG7 (2006) is a multimedia content description standard. It does not define an audio and video encoding format like its predecessors MPEG2 and MPEG4, but it does complement these formats by allowing metadata to be tagged. It does this through an eXtensible Markup Language (XML) schema where the metadata are associated with the video file timecode. This methodology is defined through Descriptors, Description Schemes and a Description Definition Language. These are a hierarchical set of properties where the Description Definition Language specifies the set of syntax rules for the use, interaction and modification of a Descriptor Scheme or Descriptor. A Description Scheme specifies the structural representations and relationships of both Descriptor Schemes and Descriptors, while a Descriptor is the metadata feature representation. In Nack *et al.* (1999) it is highlighted that MPEG7's application areas would include GIS as an important usage area for audio and video multimedia information.

An MPEG7 format would normally be implemented as a separate data file structure similar to those discussed in section 3.2.2. This has a useful advantage as it allows video metadata to conform to formalised standards even if the video container format itself is not MPEG-related. However, where MPEG7 can perform an important internal Spatial Video indexing role is when it is used in conjunction with MPEG4 video files. In Joung *et al.* (2003) a fully operable system has been developed that defines an MPEG7 metadata scheme embedded in an MPEG4 file format. This is achieved by extending the eXtensible MPEG4 Textual (XMT) format which is itself an extension of the XML language. Using this method, MPEG-7 formatted XML is embedded within the MPEG4 data stream. This facilitates a complete video search capability where no video associated files are required to provide the query repositories. This implementation has no specific objectives relating to GIS or Spatial Video but it does highlight the possibilities of another internal metadata storage methodology.

Extending this implementation Tae-Hyun *et al.* (2003) develop a complete VideoGIS LBS that uses MPEG7 as the Spatial Video metadata storage mechanism. The MPEG7 schema is used to provide a metadata repository for 3D geographical object searches and their related video content sequences. This methodology is used to control a wireless network LBS system that can return content based on an objects appearance in the video-object-space; an example would be an Internet hyperlink for a restaurant that is located in the area captured on spatially indexed video footage.

This methodology has significant advantages over any of those mentioned previously as it combines numerous adaptability possibilities. Various levels of video metadata description schemes could exist that would have to adhere to MPEG7 standards and would include specialised spatial data profiles. Incorporating this structure internally in the Spatial Video data file or separately is also possible. This would provide a standardised structuring for a choice of either method.

## 3.3 Decoding Audio Encoded Spatial Video

In this section a technical piece of work that directly relates to the audio encoding Spatial Video methodology mentioned in section 3.2.1 is described. This work was performed as part of a process to build automated modules for acquiring spatial data from Spatial Video files. In the following sections the commercial vendor links, encoding process and data structures that define the audio-encoded Spatial Video that needs to be decoded is introduced. The decoding process and its results are then discussed.

### 3.3.1 Encoding GPS into a Audio Stream

NavTech (1995) have developed an encoding/decoding hardware system that enables one hertz GPS NMEA data strings to be inserted into the audio stream of a consumer standard video camcorder. The system is shown in fig 3.1. Embedding the GPS data onto the audio track is achieved using a Frequency Shift Key (FSK) modulation as defined by Miyagi (1968), where the technique streams encoded audio data, through the camcorder microphone input connection during video recording. The GPS data are decoded using the same hardware which can then be post-processed into a user required file format. This can then be synchronised with the video stream based on the video and GPS time link.

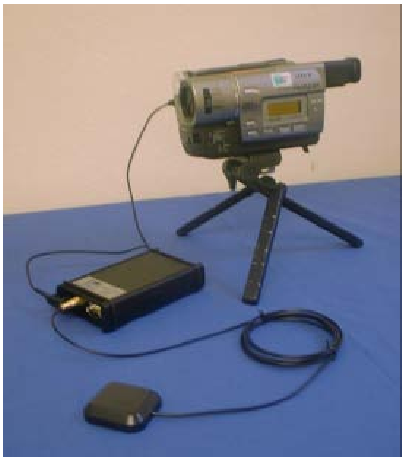
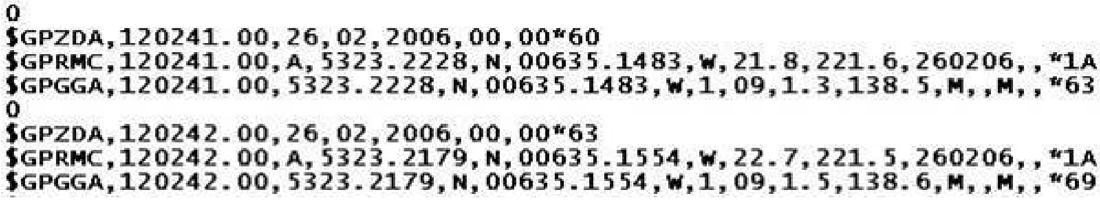


Fig 3.1. (NavTech, 1995) CamNav GPS encoding and decoding hardware shown with consumer camcorder and GPS antenna. Image taken from NavTech CamNav brochure.

The encoding processes store the resulting audio data structures based on GPS NMEA sentences. Example descriptions are detailed by Commlinx Solutions (2003) and are discussed in more detail in section 2.3.1. Fig 3.2 shows an example of these sentences along with a bespoke trigger character that can be initiated by user interaction during the survey process. Navtech systems provided support for this work by supplying a CamNav specification document (NavTech CamNav, 2004) that defines the audio GPS data structures as they would be encoded through the hardware. Contact with one of the original developers of the system was also available in an advisory and results interpretation capacity.



```
0
$GPZDA,120241.00,26,02,2006,00,00*60
$GPRMC,120241.00,A,5323.2228,N,00635.1483,W,21.8,221.6,260206,,*1A
$GPGGA,120241.00,5323.2228,N,00635.1483,W,1,09,1.3,138.5,M,,M,,*63
0
$GPZDA,120242.00,26,02,2006,00,00*63
$GPRMC,120242.00,A,5323.2179,N,00635.1554,W,22.7,221.5,260206,,*1A
$GPGGA,120242.00,5323.2179,N,00635.1554,W,1,09,1.5,138.6,M,,M,,*69
```

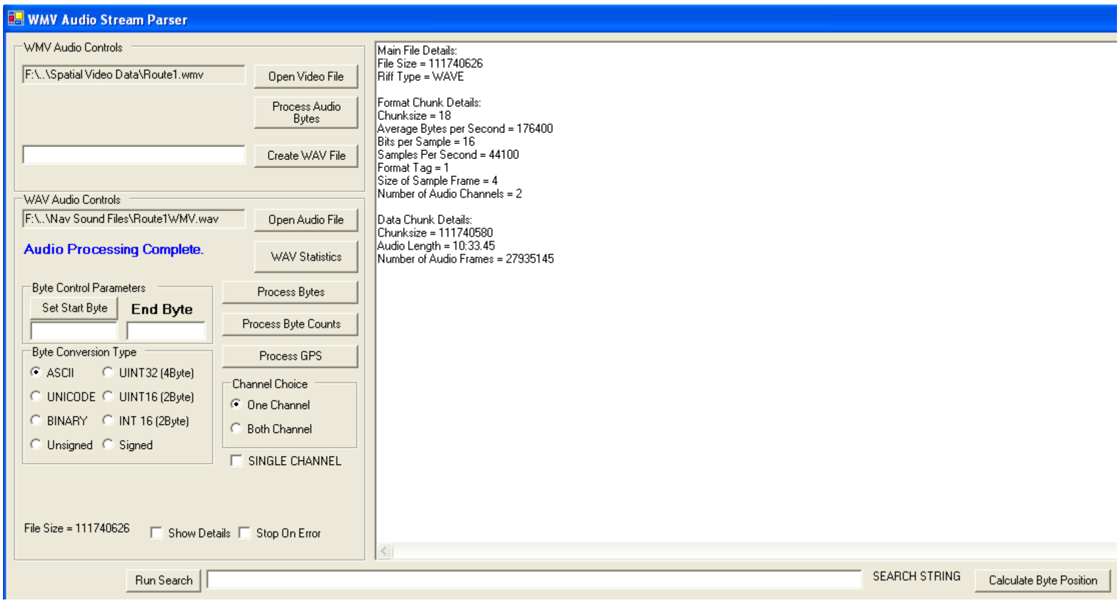
Fig. 3.2 Sample set of GPS NMEA messages with intermittent user definable trigger characters.

This FSK wave form is a continuous, ninety six byte, transmission structure that has an overall duration of 192ms (milliseconds) per audio data frame. All frames begin with byte zero as the frame synchronisation byte and end with byte ninety five as the check sum for one whole frame structure. This data frame is encoded as a continuous stream of GPS symbols represented as ASCII binary strings with each bit having a 250µs (microseconds) duration in the CamNav data frame of the audio channel. Each zero binary symbol is defined by an audio signal inversion of 250µs, while a one binary symbol has a second signal inversion at an interval of 125µs. Based on these data structure specifications some basic calculations and assumptions can be made that include determining the data frame structure, frame byte positions and symbol partitions. Firstly, the symbol bits are encoded Least Significant Bit (LSB) first by the encoder which necessitates the resultant binary string representation to be reversed for visualisation and decoding. Secondly, this frame structure can only facilitate encoding ninety four GPS data string characters per hardware processing cycle.

### 3.3.2 Developing a Decoder

Almost any digital video capturing equipment can be used with CamNav as long as it has an audio input connection with which the hardware can stream the encoded GPS NMEA sentences onto the audio channel. Based on the hardware-specific video codec that is implemented on the survey camcorder, any one of many different video formats could be encountered in the decoding process. Therefore, a generalised approached was adopted where all Spatial Video sets that were captured using the GPS audio encoded methodology went through two decode pre-processing steps. Firstly, the Spatial Video was uploaded to a PC and converted to a Microsoft Windows Media Video (WMV) format which generalised the next step. Secondly, the audio stream was isolated and separated from the video file and stored in a Waveform Audio Format (WAV) file. This second step was performed to separate the development and testing logic of the audio decoder from a video stream format that was little understood during the initial stages of development. Ultimately this second step would not be required in a final system.

The WAV file format data structure is detailed in two technical source documents, (Microsoft, 1992; Bosi *et al.*, 2003), and is a well understood and supported format through numerously available open source analysis systems. However, because of its simplified data structure, some basic bespoke software was developed to enable the WAV file creation, as mentioned in step two above, and facilitate detailed bit and byte level analysis of the file's audio data layout. This component allowed any WAV file to be analysed based on a number of binary data structures which included ASCII, Unicode, Signed and Unsigned Integer formats. The file could be traversed from any user controlled starting and ending byte locations. It also comprised a number of search and output modules for identification of binary string representations of specific sections of encoded GPS data. Fig 3.3 shows a screen shot of the software with an example of a WAV file identification and contents header being decoded based on a zero to forty byte elements search and display setting of ASCII format.



F:\..\Spatial Video Data\Route1.wmv

F:\..\Nav Sound Files\Route1WMV.wav

Audio Processing Complete.

● ASCII ○ UINT32 (4Byte)
○ UNICODE ○ UINT16 (2Byte)
○ BINARY ○ INT 16 (2Byte)
○ Unsigned ○ Signed

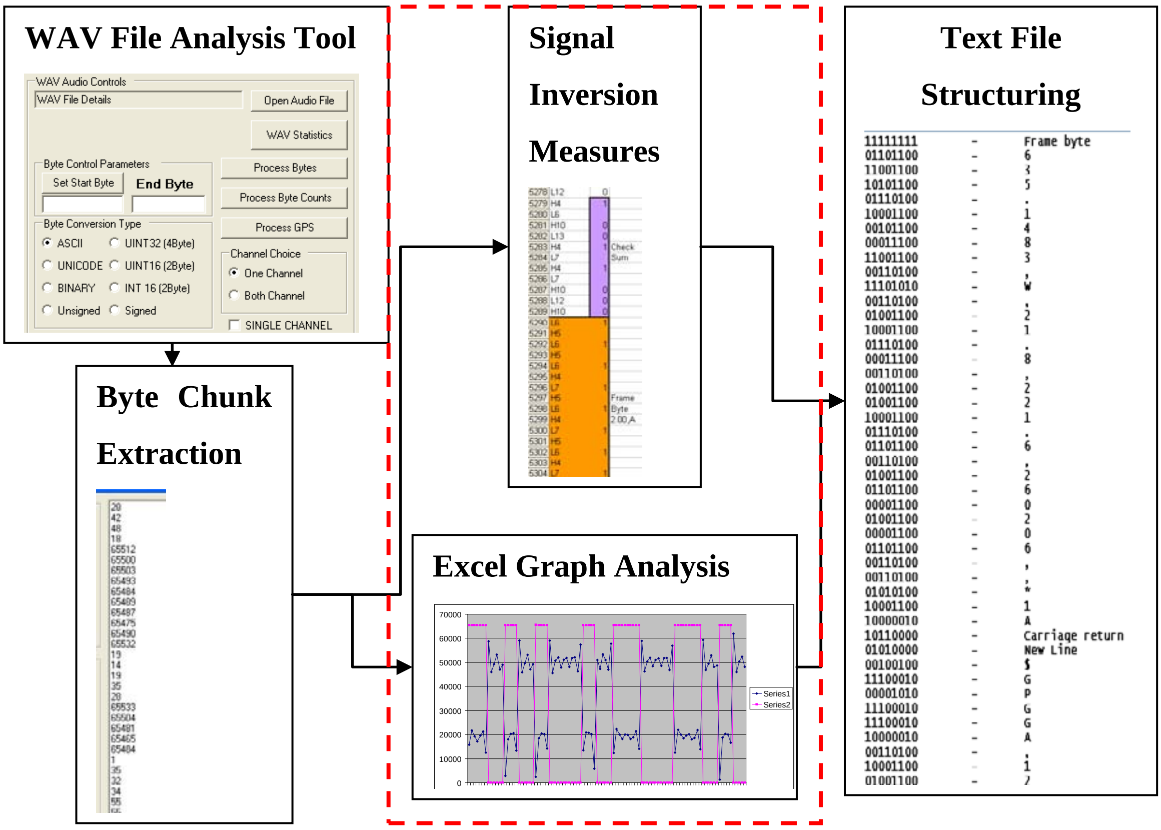
● One Channel
○ Both Channel
□ SINGLE CHANNEL

File Size = 111740626 □ Show Details □ Stop On Error

```
Main File Details:
File Size = 111740626
Riff Type = WAVE

Format Chunk Details:
Chunksize = 18
Average Bytes per Second = 176400
Bits per Sample = 16
Samples Per Second = 44100
Format Tag = 1
Size of Sample Frame = 4
Number of Audio Channels = 2

Data Chunk Details:
Chunksize = 111740580
Audio Length = 10:33.45
Number of Audio Frames = 27935145
```

Fig. 3.3. Screen shot of the WAV file detailed analyser software developed for this research work.

The decoding process comprised four distinct stages and a total of five different operations that began with the WAV file analysis software and ended with an extension module being included in this software to decode a complete GPS stream. Fig 3.4 summarises the various stages and operations involved in the decoder development process. Stage one involved using this analysis tool to understand and gain familiarity with the WAV file data structures. The various project data files were objectively analysed to ascertain similarities in byte positioning and data contents. Stage two involved the extraction of audio data byte chunks. These were extracted as signed integer audio sample values, for stereo and mono formats. Stage three comprised two operations where these data sets were examined to determine wave structure and location of signal inversion points. These operations proved very time consuming as large numerical data sets were processed and measured for inversion points at numerous points in the audio sample files. Finally, the fourth stage entailed processing of signal binary symbols so that these could be structured into byte-sized group structures that could then be converted to their ACSII character representations, as taken from the IEEE Long Island (2005) tables.



```
20
42
48
18
65512
65500
65503
65493
65484
65489
65487
65475
65490
65532
19
14
19
35
20
65533
65504
65481
65465
65404
1
35
32
34
55
```

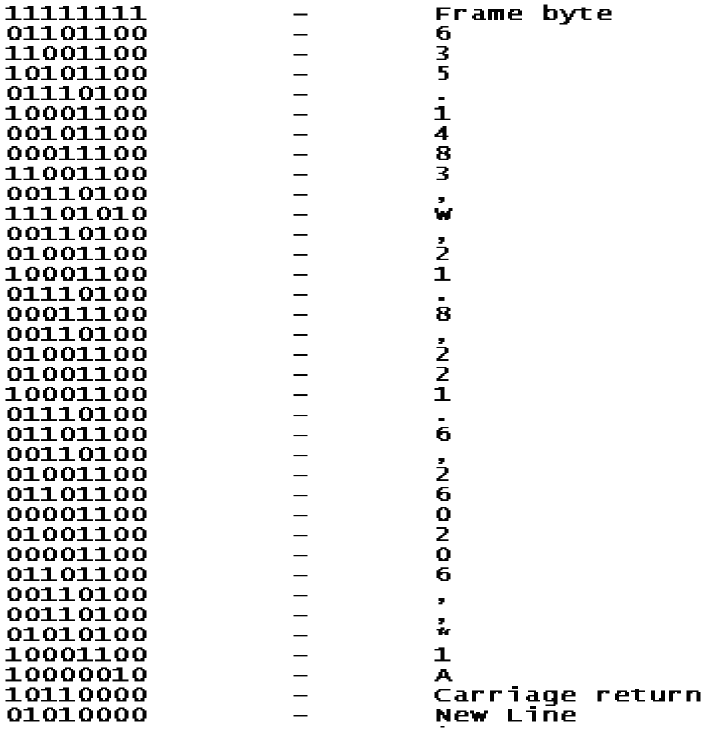
```
5278 L12   0
5279 H4    1
5280 L6
5281 H10   0
5282 L13   0
5283 H4    1  Check
5284 L7       Sum
5285 H4    1
5286 L7
5287 H10   0
5288 L12   0
5289 H10   0
5290 L6    1
5291 H5
5292 L6    1
5293 H5
5294 L6    1
5295 H4
5296 L7    1
5297 H5       Frame
5298 L6    1  Byte
5299 H4       2.00,A
5300 L7    1
5301 H5
5302 L6    1
5303 H4
5304 L7    1
```

```
11111111    -    Frame byte
01101100    -    6
11001100    -    3
10101100    -    5
01110100    -    .
10001100    -    1
00101100    -    4
00011100    -    8
11001100    -    3
00110100    -    ,
11101010    -    W
00110100    -    ,
01001100    -    2
10001100    -    1
01110100    -    .
00011100    -    8
00110100    -    ,
01001100    -    2
01001100    -    2
10001100    -    1
01110100    -    .
01101100    -    6
00110100    -    ,
01001100    -    2
01101100    -    6
00001100    -    0
01001100    -    2
00001100    -    0
01101100    -    6
00110100    -    ,
00110100    -    ,
01010100    -    *
10001100    -    1
10000010    -    A
10110000    -    Carriage return
01010000    -    New Line
00100100    -    $
11100010    -    G
00001010    -    P
11100010    -    G
11100010    -    G
10000010    -    A
00110100    -    ,
10001100    -    1
01001100    -    2
```

Fig. 3.4. Decoder development process flow chart. Stage threes two analysis operations are highlighted in the dashed red box.

### 3.3.3 Implemented Decoder Results

Analysis of the encoded frame structures for a number of different video segments showed an average of ninety two GPS data string encodings instead of the expected ninety four. The cause of this is the GPS signal propagation delays which resulted in the hardware encoding the missing byte symbol structures with the 0x00 padding or ballast bytes. It has also been shown that the software decoder method output at least one more complete GPS data string from the audio files at the beginning of the decode process than that produced through the hardware version. Based on the hardware specifications document, it is assumed that either the synchronisation procedures used in decoder mode or the serial port data handling architecture could be the reason for some GPS NMEA messages being dropped when the decode process begins. Fig 3.5 displays a sample of the manual decode work preformed, with all the symbol groupings and complete ASCII conversions shown. This sample can be clearly related to fig 3.2 which was produced through the use of the hardware decoder and matches the GPS characters contained in the second half of line three and the first half of line four.



```
11111111        –        Frame byte
01101100        –        6
11001100        –        3
10101100        –        5
01110100        –        .
10001100        –        1
00101100        –        4
00011100        –        8
11001100        –        3
00110100        –        ,
11101010        –        W
00110100        –        ,
01001100        –        2
10001100        –        1
01110100        –        .
00011100        –        8
00110100        –        ,
01001100        –        2
01001100        –        2
10001100        –        1
01110100        –        .
01101100        –        6
00110100        –        ,
01001100        –        2
01101100        –        6
00001100        –        0
01001100        –        2
00001100        –        0
01101100        –        6
00110100        –        ,
00110100        –        ,
01010100        –        *
10001100        –        1
10000010        –        A
10110000        –        Carriage return
01010000        –        New Line
```

Fig. 3.5. Sample taken from the manual decode processing files that defines the binary structures determined from the audio signal data structures.

## 3.4 Conclusions

Based on the work performed in understanding, analysing and implementing various aspects of these Spatial Video data structures, it was realised that the initial research objectives could not be efficiently realised through any of these methods. All these methods involve constraints because of the video centred approach where each video's spatial data is either internally contained or part of a related set. This results in videos that could be stored in distributed locations, in many differing formats and using bespoke spatial indexing systems. The problem is in providing an efficiently centralised indexing methodology that is separate from any constraints relating to the video and its spatial data.

What is needed is an efficient Spatial Video data structure that provides extended GIS functionality through indexing and searching based on the video's spatial properties. To translate this objective to all the data structures presented here, a static approach is preferred as a dynamic one would be highly inefficient. This is because of the disparate number of system implementations that would be required but also the access and processing time needed to build the Spatial Video GIS model on the fly. To dynamically process the numerous different types of Spatial Video data structures a process of operations would be required that includes:

- Accessing all video files in the distributed system to discover their inherent structures, i.e. file formats, spatial indexing etc.
- Retrieve the varying spatial data implementations and process them based on type, accuracy, geo-referencing and interpolation requirements.
- Perform the required operations on the generated spatial model and form a result which could include defining a number of different video sequences.
- Traverse the distributed system, again, to access these video sequences.
- And, possibly, having to replicate the same processes, again, every time the data, operation or access location requirements change.

While this dynamic process is possible it is not logical. Therefore a static method is devised where the Spatial Video GIS model is defined post-capture and pre-integration with a GIS. In most Spatial Video cases a post-capture processing is already required to verify and validate the video and spatial data relationships. Thus, using this approach, access to the Spatial Video structure will only be required once where a centralised system could be efficiently implemented and populated.

This chapter is important in understanding the core Spatial Video concepts and highlighting some significantly diverse methodologies for Spatial Data handling and storage. These could be formalised in other future research projects. On the other hand, this chapter’s significant contribution is that it determined that a centralised spatial video-data indexing approach is required rather than a video spatial-data one. It is this concept that is developed through the rest of this thesis.

# Chapter Four: Introducing Viewpoints

This chapter introduces and develops the Viewpoint model as a Spatial Video GIS data structure. It will be shown how this model is a theoretical extension of existing viewable region models that include Isovist, Viewshed and Frustum structures. Viewpoints are defined as very simple GIS data structures that are calculated from a video frame’s known location and the video camera’s operational parameters. This computational form closely models the View Frustum structure used in 3D computer graphics, but here it is introduced into a geo-spatial domain. While this concept is a generalised and simple idea, implementing it accurately as a 3D form in a global coordinate system poses a number of considerable challenges. These challenges are discussed as part of a complete model that could be extended into 3D GIS modelling environments.

## 4.1 Introduction

The core concepts in the development of a general Viewpoint Spatial Video data structure are grounded in both the Space Syntax Theory fields of Architectural Isovist and GIS Viewshed analysis. In general, Space Syntax encompasses the theories and techniques for the analysis of spatial configurations across many different research fields, (Hillier *et al.*, 1976). Utilising these approaches to understanding space in terms of digital imagery is simply an extension of these concepts to include various aspects of Computer Science and Electronic Engineering. In particular, Spatial Video data sets provide a visualisation platform that is inherently a geometrically oriented and definable representation of an accessible geographical space. While the more mainstream aspects of Space Syntax Theory concentrate on the analysis and study of space, the Viewpoints concept for this thesis is largely an exercise in how this space can be defined, modelled and studied in a video context. Thus, the Viewpoint approach is introduced in this chapter as a theoretically definable 3D construct where the various relevant concepts and aspects are discussed. This then begins a further series of chapters that continue this discussion and further develops the concept through the use of well-understood Computer Science technologies and methodologies in a practical implementation and application of this data structure.

## 4.2 Modelling a Viewable Environment

Historically, the steps and stages in modelling a viewable geographical environment began with the construction and definition of an Architectural Isovist. This has subsequently been refined and introduced into GIS in the form of Viewsheds. Both concepts basically define the '*set of all points visible from a given vantage point in space with respect to an environment*' (Benedikt, 1979). Modelling Spatial Video in the form of Viewpoints geographically expands this environment definition by utilising the operational parameters of the video recording equipment to approximate the viewable region in the form of a viewing Frustum. These aspects that contribute to the theoretical development of the Viewpoint model are briefly introduced in the following sections.

### 4.2.1 Architectural Isovist

The originator of the Isovist model, as surmised by Turner *et al.*, (2001), is Tandy (1967) although it has a long history as an application and analysis technique in architecture. In its basic form it provides a very simple 2D plan-view model of the viewable environment. It is typically modelled from a defined location in space where a full 360$^{\square}$ viewing rotation about this area is determined. As such an Isovist can be generated from a number of diverse spatial situations from a point to an area in space. Architecturally it is usually used to model larger objects in space, such as buildings, for possible line of sight or impact to the visible environment analysis, as can be seen in fig. 4.1.

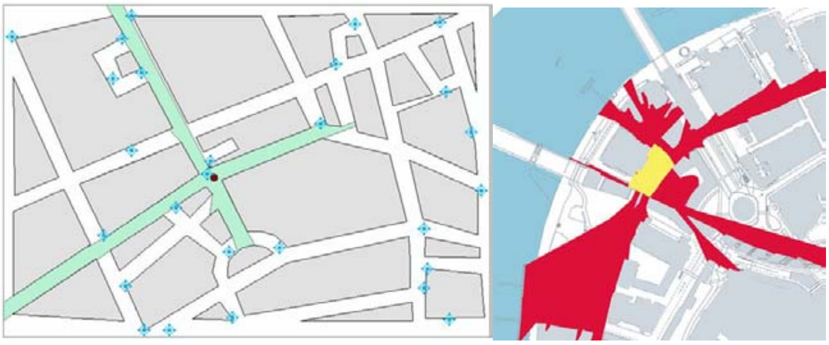


Fig. 4.1. Sample images of Architectural Isovist implementations. The left hand image shows the viewable environment from a point in space, while the right hand image defines the impact of a building footprint on the viewable environment. Images taken from http://www.spatialanalysisonline.com and http://www.spacesyntax.com respectively.

### 4.2.2 GIS Viewshed

A Viewshed can be considered a specific implementation of the Isovist model in a purely geographical or GIS context. The same Isovist principles are applied in the construction of a Viewshed; however bounding limits are typically applied on the rotational axis of view which defines a restricted spatial extent. In 3D GIS modelling of Viewshed extents, digital elevation models usually form the intersecting layer that will define the viewing boundaries, restrictions or occlusions, an example of which is shown in fig. 4.2.

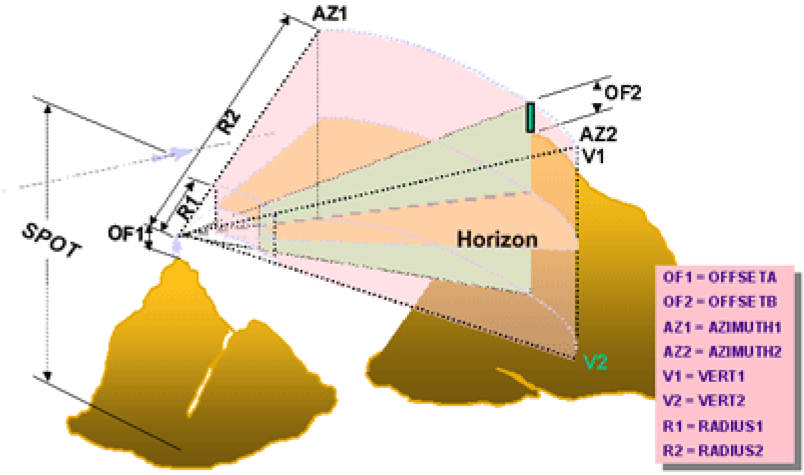


Fig. 4.2. This 3D model highlights the variables required to calculate a Viewsheds spatial extent. A number of optional parameters are shown that include two azimuth variables to determine the horizontal field of view and a minimum and maximum radius to define the projection distances. Image taken from http://webhelp.esri.com .

From Fisher, (1999), Worboys *et al.*, (2004B), summarises Viewshed models as either probable or fuzzy in a GIS context. They conclude that either a well defined Viewshed boundary can be constructed to probabilistic location accuracy or that the region will be defined by a fuzzy boundary that is both broad and graded. These points are important in the theoretical development of Viewpoints as a high accuracy geographical extent boundary is not possible without employing very accurate spatial measurement equipment. A typical Viewpoint boundary defined in this study is an approximation of the possible extent.

### 4.2.3 3D Viewing Frustum

A Viewing Frustum is a viewable region modelling approach used in 3D Computer Graphics. It is a computer-based screen rendering methodology built on the concept of a frustum, which is a geometrical shape formed from a pyramid structure bounded by a plane parallel to the base and the base itself, as shown in fig. 4.3. In Computer Graphics research a viewing frustum model is the potential volume of space that can be rendered on screen regardless of its containing occlusions. Further research in this view perspective approach attempts to account for occlusions through view-frustum culling techniques.

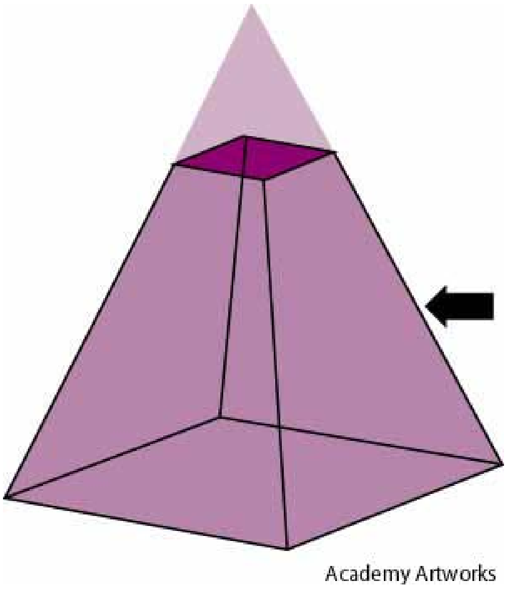


**Fig. 4.3. This example of a frustum is the bottom portion of the pyramid once the top portion is removed. Image taken from http://content.answers.com/main/content/img/ahd4/A4frustr.jpg**

Such Viewing Frustum models are defined by various parameters that depend on the camera's position, orientation and optical settings. The frustum structure is usually a rectangular pyramidal shape, as shown in fig 4.4.; however, applying this model to a real world scenario would involve a number of adjustments of this symmetrical shape to account for various camera and environmental corrections as well as lens distortion or atmospheric refraction.

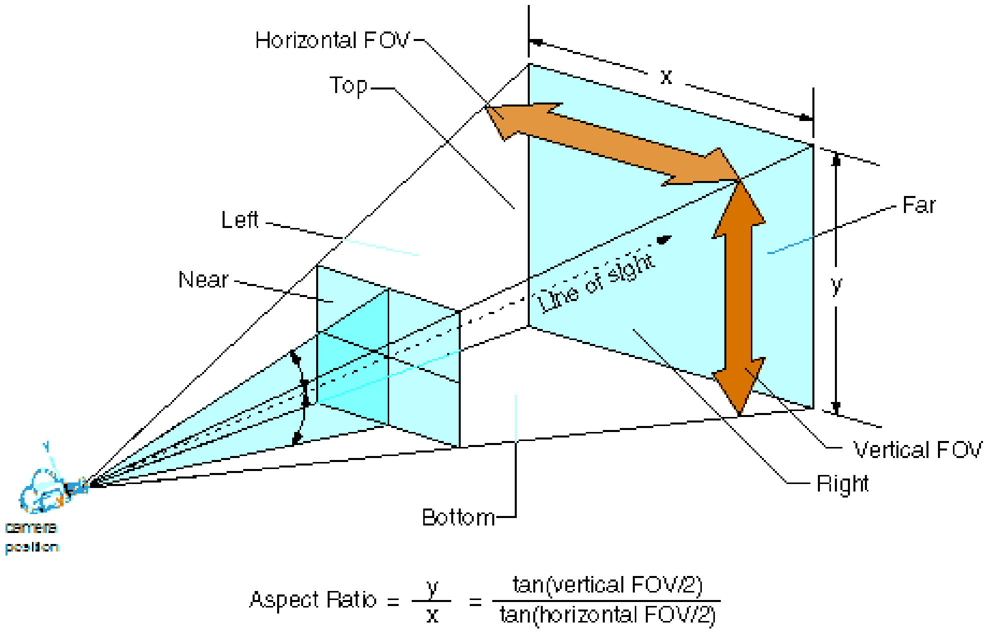


Fig. 4.4. This example highlights the various elements of a Viewing Frustum structure as would be applied to a Computer Graphics task. It shows the near and far depth-of-field planes which are also bounded by a top, bottom, left and right plane. All of which simulate the internal containing characteristics of the computer graphics monitor viewable regions. Image taken from http://wwweic.eri.u-tokyo.ac.jp .

### 4.3 GIS and Photogrammetry

The previous section briefly outlined and developed the set of environmental modelling areas that provide the theoretical basis for implementing the Viewpoint concept in a GIS. Alternatively, various research fields of photogrammetry specifically deal with image rectification and image object space geo-referencing. Normally, the speciality techniques that are well-defined for aerial photogrammetry are, in the case of oblique terrestrial images, extended and/or redefined to handle the different photogrammetric problems and requirements presented. The underlying principle is that an image is correctly processed and adjusted to represent the environment it is capturing, such as Orthophotos. The approach of a Viewpoint is to conceptually reverse this process and define the image object space as a GIS entity.

Briefly highlighted in this section are some of the methods used for image object space geo-referencing, rectification and triangulation; however these are techniques applicable to post-image capture analysis and require precise and time consuming attention. This study is modelling video which does not easily lend itself to such slow and exacting methods of image verification and rectification. Thus, while these techniques provide an accurate basis for image object space geo-referencing and/or rectification, they would need to be adapted to be used as a basis for the Viewpoint definition described in this study.

### 4.3.1 Aerial Image Techniques

The basic geometry of any aerial image is defined by its optical axis and whether its exposure station is vertical or near vertical. Practically, a true vertical photograph is almost impossible because aircraft attitude and surface contours will prevent this. However, the tilted photographs can still be analysed with vertical geometry equations to acceptable levels of error given a tilt angle of approximately ±1° from vertical. Moreover, image correction for lens distortion, atmospheric refraction and earth curvature may also require consideration in digital image geo-processing. Modern digital cameras have lens distortions very finally calibrated and only require adjustment calculations in the most precise of analytical circumstances. Atmospheric refraction is directly affected by the height and angle of the image; this requires Snell’s law to be solved for the light rays based on the proportional change in the refraction index. Correction for earth curvature is primarily a concern for imagery captured at very high altitudes and those exposed to large contour and elevation changes. Because of the many known problems with this type of adjustment an alternative approach is to employ a 3D orthogonal object space coordinate system.

Determination of an image’s scale can easily be realised by the same measured relative distances in both the image and over the captured surface. These measures can be calculated through a number of different techniques that determine the scale based on the vertical properties of the image and the type of terrain captured. The simplest flat terrain model defines a direct relationship between the same measured corresponding distances in both the image and on the surface. The variable terrain model will define a variable image scale as portions of the image that show higher elevations will have increased scale while those showing lower elevations will have decreased scale. In this case it is often advantageous to provide an average scale for the whole image over the entire terrain.

By the use of an arbitrary ground coordinate system, calculations of ground coordinates from an aerial image can be determined. This can be used to determine any number of the terrain points that appear in the image. By the use of some simple geometry and access to image scale, height and focal length parameters, the X and Y terrain coordinates can be calculated. Adjustments to these methods are necessary in the case of calculations where relief displacement is present. This is caused when the relief of an object is either above or below the selected reference datum; examples of this would include the effect tall buildings can have in obscuring objects.

Digital image geo-referencing in photogrammetry, also known as ground registration, involves aligning the image rows and columns with a ground coordinate system. Two steps are involved that require the computation of a 2D coordinate transformation of the image to the surface and building an alignment array that relates the image pixels to ground locations (Wolf *et al.*, 2000). The underlying process is dependent on the identification of a number of ground control points that relate the surface and the image. Once identified their conversion from a ground coordinate system to image coordinates defines' the alignment of the subsequent rectangular grid cells that are comparable to the digital images pixels.

A tilted image can be described by two sets of parameters that define both its spatial location and its angular orientation. All the previous techniques become more complicated and need to be adjusted to account for an image's tilt based on these parameter sets and other derived variables such as the lens focal length. Either Tilt-Azimuth-Swing or Omega-Phi-Kappa angular orientation systems can be used to express the tilted images rotations. Based on these fundamentals the process of converting a tilted image to its equivalent vertical one is known as ortho-rectification and results in an Orthophoto image.

### 4.3.2 Terrestrial Image Techniques

Essentially, terrestrial images have the same properties as aerial images but are captured at extreme tilt angles. There main characteristics are that they are horizontal or near horizontal, oblique, to the ground surface. In the same way that aerial images have a tilt angle based on the vertical ideal, a terrestrial image has an elevation or depression angle based on the ideal horizontal. Many techniques available to aerial photogrammetry are also relevant here, although atmospheric and earth curvature corrections are typically not necessary unless an image of large panoramic landscapes is captured. Typically, images consist of much smaller geographical extents and contain numerous occlusions and obstacles like buildings and vegetation.

If not recorded at the time of image capture, the angle on inclination of the camera axis can be calculated in certain circumstances. As with aerial photogrammetry the identification of image properties is the key to accurate calculations, in this case vertical and horizontal linear features such as window or footpath edges. If these features are present, perspective geometry principles can define either the camera axis elevation or depression angle. Given this and knowledge of the camera focal length, both horizontal and vertical angles can then be calculated for other image points. The converse of this is also possible where the image capture point can be ascertained through a three-point resection of the image. This method does require the image to contain at least three horizontal control points and the inclination angle to be known.

Lastly, stereoscopic image analysis provides the most conclusive set of techniques, methods and accuracy levels in determining the geographic context of objects captured in an image. It is usually used to measure or analyse objects in the image content rather than geo-reference them. Stereoscopic imagery adds the advantage of depth perception, gained from multiple views of the same object from different angles, and facilitates higher levels of accuracy when determining object distance or size. While depth perception in monoscopic image analysis is possible, it is only an approximation and usually intuitive knowledge of the scene determines result accuracy.

### 4.4 Spatial Video in GIS

This section discusses how a Spatial Video Viewpoint is theoretical defined as a GIS spatial entity. Also discussed are the geographical properties of Spatial Video in order to determine how these can be used to describe the components of a Viewpoint model. The relationships between the captured image's spatial location, orientation and geographical extent are discussed in terms of the GIS data types that are most applicable to constructing each Viewpoint component. The principle consideration is to use existing, well understood, GIS data type primitives. However, actual Viewpoint calculations are not performed as part of this introductory chapter.

A Viewpoint is a GIS data type representation of a single Spatial Video image frame. As a short summarisation, a Viewpoint is defined by both point and polyhedral GIS data types that form a one-to-one association. The spatial variables of the video image are used to define the point structure, which represents the camera's location in space when the image was captured. The geographical space that has been captured in the image can be semantically represented in many different forms, at varying levels of complexity; the representation used in this thesis is a generalised polyhedron. This is implemented through GIS polygon data types and facilitates a simpler visual and written discussion platform. These concepts are expanded in the following sections.

Chapter five implements this theory in a 2D space while chapter six defines the physical data structure as implemented in a spatial database. This database defines each Viewpoint using point and polygon geometry data-types as described in the table 6.2 schematic.

### 4.4.1 Video Camera Spatial Data

A GIS point data type is used to define the camera's location in space when the image was captured. Based on OGC simple feature standards (Herring, 2006), a point data type can be defined by four variables X, Y, Z and M. Used in this model are X, Y and Z to define the latitude, longitude and altitude of the camera sensor. The M variable is used to store a true north azimuth of the cameras direction of travel. This is calculated based on the current and previous point location variables; all these variables are shown in fig 4.5.

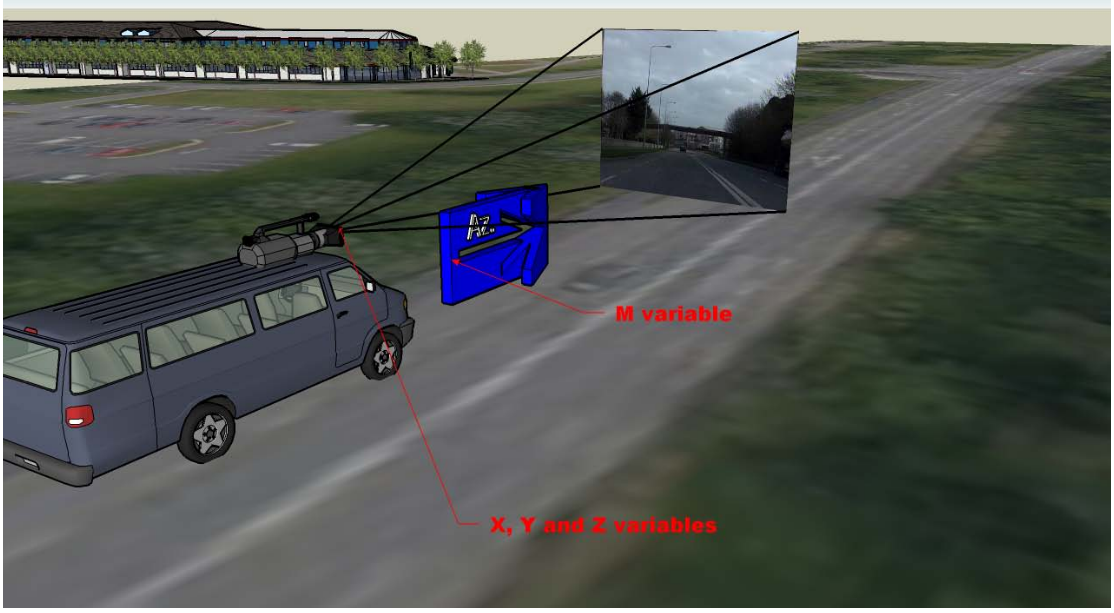


Fig 4.5. This figure visualizes a Spatial Video frame image and its associated point variable relationship. X, Y and Z define the camera sensor plane spatial location. M defines the true north heading of the camera.

It is important to consider the M variable, as it only stores the azimuth direction that the camera is travelling in along the surface. In this context this variable is used to project and calculate the geographical extent as captured in the image. However, it cannot be assumed that the camera's sensor plane will be both orthogonal to the traversal surface and coincident with the azimuth. In practical operational scenarios, a Spatial Video camera can be orientated in any non-orthogonal position that may capture other oblique views of the environment that are disjoint from the camera's direction of travel. Thus, another data structure is required to handle this particular situation and would need to be incorporated into the final system.

For this theoretical implementation, a set of roll, pitch and yaw angles can be used to define the camera's orientation. The reference axis systems origin would be the camera sensor plane principle point. Thus, the yaw angle would be the difference between the azimuth and the sensor's orthogonal orientation in the surface. The pitch angle would define the degree of difference between the sensor plane in an orthogonal position and its actual position. The roll will define the angle of difference between the sensor plane being parallel to the surface and its actual position. All these angles will have to be incorporated into the spherical geometry model that calculates the subsequent geographical extent; these angles are shown in fig 4.6.

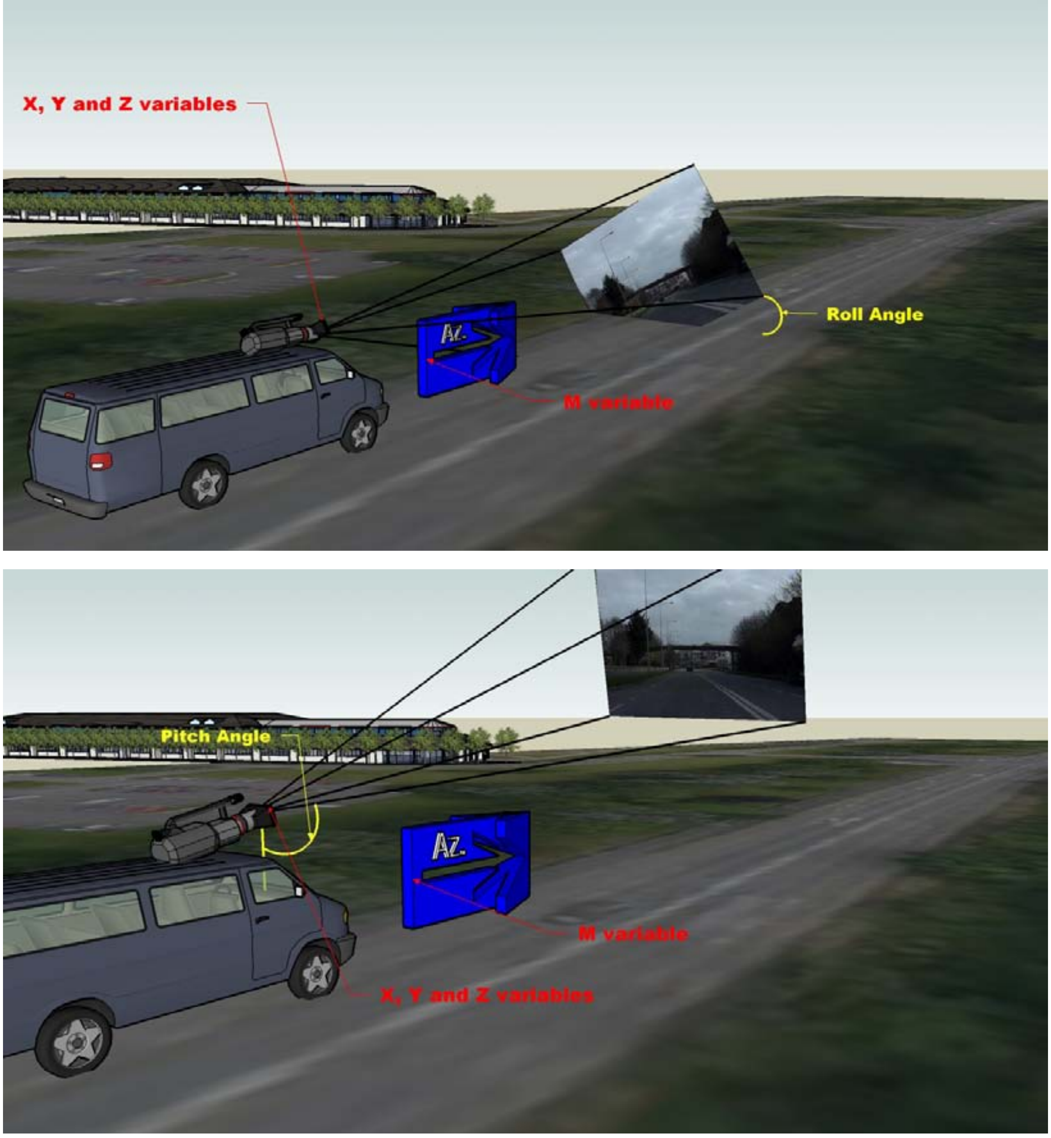


Fig 4.6. Continued on next page.

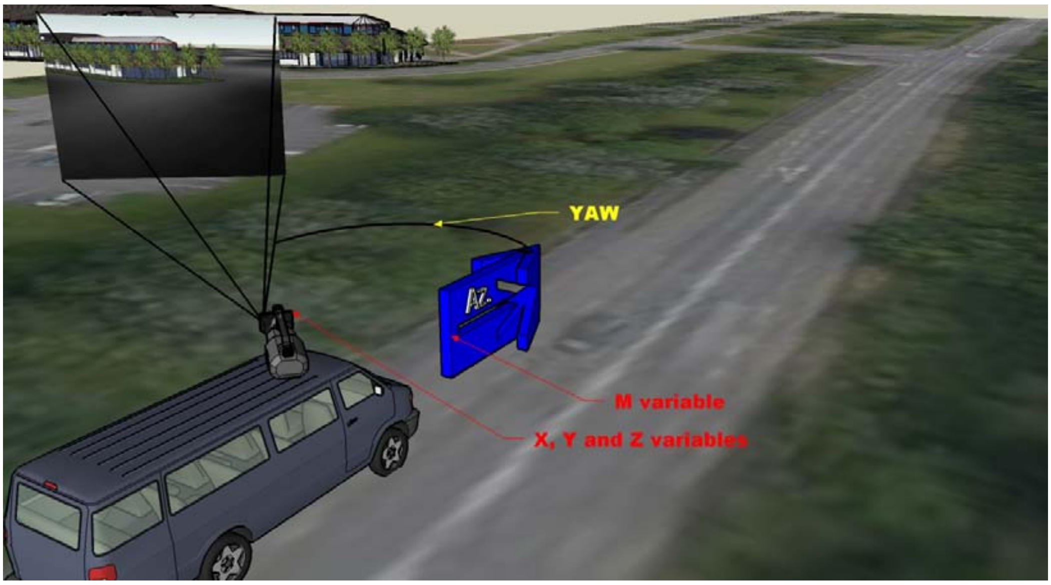


Fig 4.6. This series of images shows the Roll, Pitch and Yaw angles in the Viewpoint Spatial Video model.

The point data type has many important uses for spatial entity representations in a GIS; however a Viewpoint implementation has utilised the M variable in such a way that the point has a conceptual restriction placed on it. It is important to understand that using the GIS point data type to define an image's capture location and orientation is to re-define or restrict its original concept to a semantically different one. The subsequent calculations that create the image's geographical extent are defined by the projections and translations about the M defined axis. This means that any logical spatial query involving the use of a Viewpoint point data type will need to consider the M variable as a logical control. This is discussed further in the following sections.

### 4.4.2 Video Image Spatial Data

Here, a viewing frustum concept is used to model the geographical extent of a Spatial Video image. The basic structure that defines a viewing frustum is a geometric pyramidal polyhedron, as is shown in fig. 4.4. This principle has been implemented in 2D through the ViewCone data structure of the OGC Geo-Video Service, (Lewis, 2006). This construct could be extended to the 3D domain using the same principles contained in the following sections; however its use would not necessarily be relevant because of the nature of aerial imagery. A correctly calibrated aerial camera will have its focal length and aperture set for an infinite far depth-of-field. This easily facilitates the capturing of surface imagery from above where all the relevant object space is on the surface. Thus, using a digital elevation, ellipsoidal or other appropriate surface model, the far depth-of-field extent can be calculated, as shown in fig 4.7.

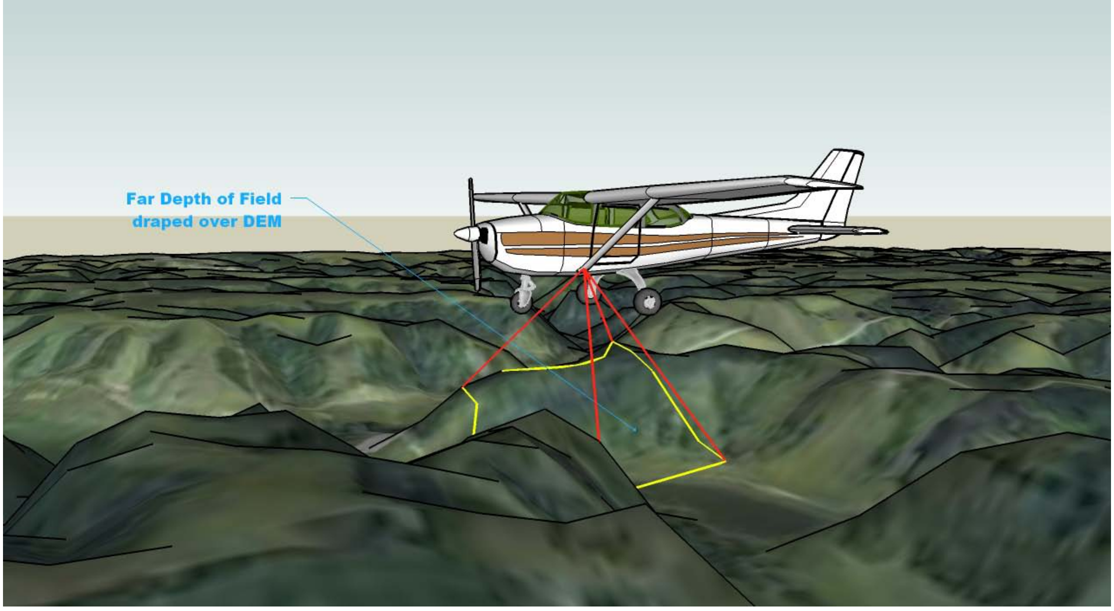


Fig 4.7. This figure simulates an aerial image draped over a DEM where the calculations only need to determine a far depth-of-field, i.e. where the image lines-of-sight intersect the ground.

However, in the Viewpoint model introduced here near and far depth-of-field planes are defined for a number of reasons:

1. Oblique terrestrial imagery will capture objects at widely varying perspective depths, not just at the far extent of the focus range.
2. Near or far objects may appear blurred in captured images where the video camera focus has a fixed setting.
3. Far field views may not contain definable geographical object space, i.e. atmosphere.
4. Where video camera footage has been captured with automatic focus, near and far geographical extents will change and vary between frames.

Because of the oblique terrestrial nature of the video images, normally not all of the object space light rays of the image intersect with the terrain. It is more likely that only a small portion of the image plane’s object space will intersect with the terrain, as shown in fig 4.8. Portions of the image will capture above the horizon and thus the object space will be the atmosphere, considered infinity. In other cases, like large panoramas of landscape imagery, definable objects like mountain ranges will appear in focus at very large distances at many tens of kilometres distance from the spatial location of the camera.

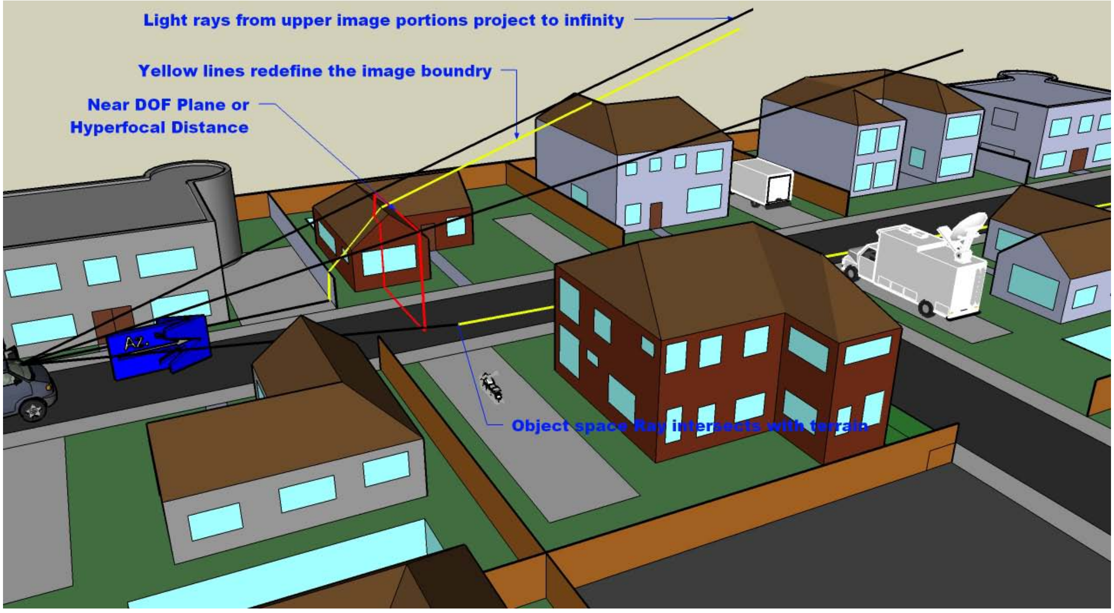

Light rays from upper image portions project to infinity
Yellow lines redefine the image boundry
Near DOF Plane or
Hyperfocal Distance
Object space Ray intersects with terrain

Fig 4.8. Spatial Video image captured in an urban environment has a number of problems directly related to both near and far Depth-of-Fields.

So, introducing the depth-of-field is not to define the exact image boundaries in spatial terms but to approximate them for information that is theoretically relevant. This is because of the multiple situations that can exist in collecting Spatial Video where geographical extents are either not fully realisable or not realistically relevant. Two particular situations represent this issue, firstly, for video footage that captures spatially sequential images that are roughly coincident with the camera's direction and secondly, for sequences that capture images with a lot of extreme far field object space and/or infinite indefinable object space.

In the first case, video sequences of a forward-facing moving camera will create multiple images, each containing portions of the same geographic space as previous or subsequent images. The portion of geographical overlap will depend on the vehicle's forward momentum and the camera's frames per second rate. Thus, by defining the depth-of-field parameters they can optimise the geographical extent for each individual image. When a spatial query is applied it will be possible to return video sequences or individual images where the query item is visually optimised.

In the second case, video sequences of whole or partial open spaces will have each image register object space over large distances. A Hyperfocal Distance camera lens setting will define the object space of these image sequences as the distance at which the camera lens can be focused such that all subjects from half that distance to infinity will be in acceptable focus, (Derr, 1906). However, distant objects will have a very low resolution and thus be visually inefficient, thus, modelling this space may be irrelevant. In a 3D context, the vertical axis of the image object space may capture large amounts of atmosphere which is infinitely expansive and impractical to model in real terms. Alternatively, for non-Hyperfocal distances, a calculable far extent will exist anyway, where all objects beyond this distance will appear blurred in the image.

Beyond these spatial appreciation concerns, the variable and dynamic image content of terrestrial Spatial Video must also be considered. It is not simply a case of introducing the depth-of-field plane at a measurable near distance and relevant content far distance. These planes themselves may require spatial adjustment due to the many different types of occlusions and visual restrictions present in Spatial Video. Examples include buildings, other moving objects like vehicles or pedestrians, network infrastructure like signage or lampposts. Both near and far depth-of-field planes may require complex adjustments to account for these issues. Visually they could appear as holes or segments in the plane having to be warped or stretched around the object. Fig 4.9 shows an oblique image containing occlusions and its hypothesised 2D plan Viewpoint geographical extent. Fig 4.10 shows a simple 3D example of this image space occlusion situation where more complex ViewCone structures would be determined by these effects.

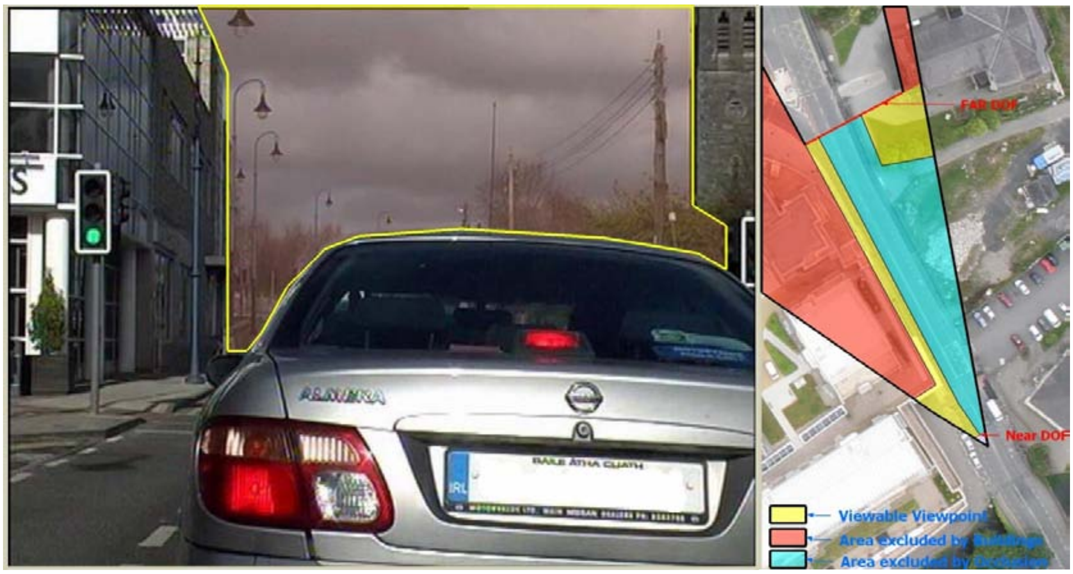


Fig 4.9. The left hand image displays an obliquely captured Spatial Video frame image with the Far DOF defined behind the occlusion. The right hand image is a 2D plan representation of the various Viewpoint geographical extent restrictions that would be imposed at differing spatial altitudes.

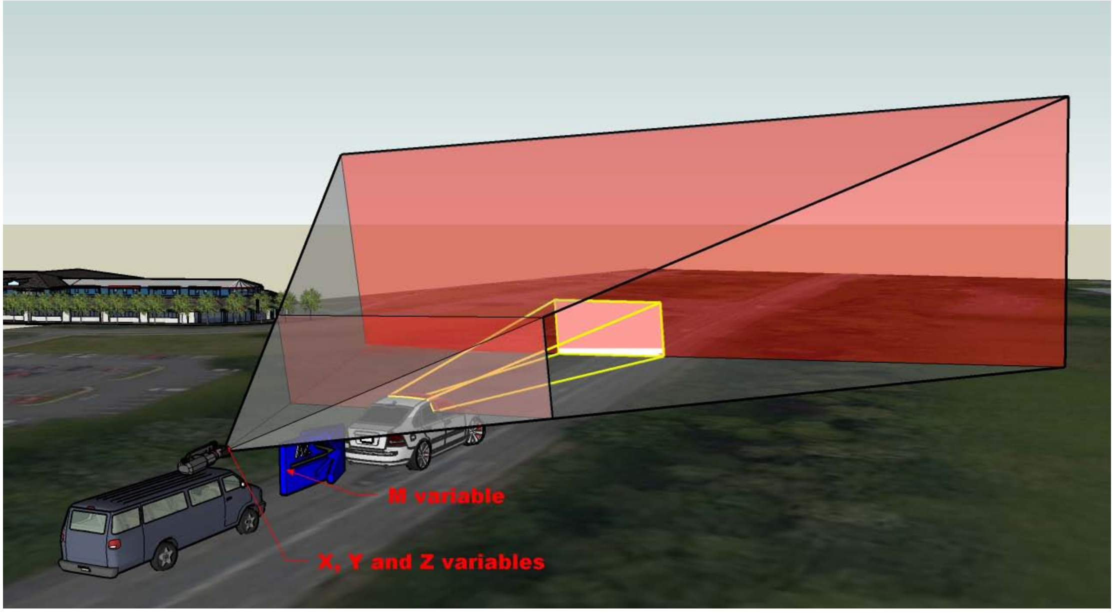


Fig 4.10. Simple 3D visualization of a Spatial Video frame image and its geographical extent restrictions based on a single occlusion.

## 4.5 Viewpoints in 3D GIS

Creating a Viewpoint model in a 3D GIS environment requires many considerations and will probably involve a number of solutions. To begin, a Viewpoint is a single defined representational structure of a Spatial Video frame image. The model creates two very different, yet dependent, geometric entities. Firstly, modelling the spatial location of the camera where the image was captured is simple and well-supported in GIS modelling, visualisation and analysis domains. As mentioned previously, the X, Y and Z variables of the point data type can fully represent this requirement. All the other attributes of this point structure are important reference variables that define the link to the geographical extent structure of the image's object space.

Secondly, to model the geographical extent, a context for choosing a solution is important as many different methods that define a 3D entity are possible. This context will define the objectives which will further determine the optimal solution. Examples of context would include the type of questions that could be asked of or problems solved by building this model of Spatial Video. The subtle, yet important, distinction is; does a Spatial Video, Viewpoint based, geographical analysis query need to return the relevant video sequences, geography or other set of results and in what form, visually, textually etc.? The answer to this type of question will define whether the model needs to construct a 3D geometry to serve a visualisation result or just return a set that defines the appropriate Spatial Video sequences.

By making this distinction the technical implementations that define the Viewpoints ViewCone geometry are determined. The very basic implementation would be no pre-processed geometric shapes, just the Viewpoint control point and the entire set of associated spatial and camera variables. From this set, the geometric structure could be calculated on the fly as per requirements. Alternatively, pre-processed geometric structures could be created based on any one of many; image processing, photogrammetric and GIS techniques that are available. Also, the choice of storage data structures for 3D objects is extensive within GIS and GI database environments. The chosen implementation depends on the complexity of the resultant geographical extent and the intended application specific area of use. This in turn depends on the complexity and detail of the camera and spatial data, as not every Spatial Video data set will contain easily definable and accurate parameter sets.

In the very basic visualisation case, a maximum geographical extent could consist of six GIS polygon data types constructed as a geometrically closed polyhedron. As is shown in fig 4.11, two of the polygon planes, in red, will be the near and far depth-of-fields while the remaining four will be the left, right, top and bottom planes. All these planes can be calculated and defined based on the base set of video camera and spatial parameters mentioned so far. In further chapters it will be a 2D version of this data structure that will define the Viewpoints and implementation operations that are applicable.

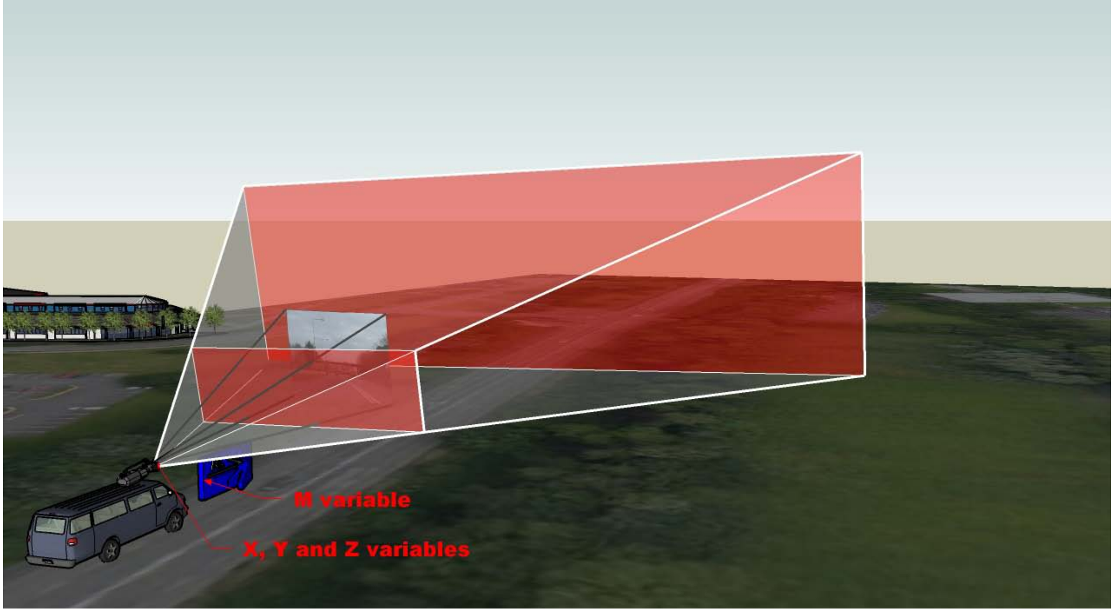


Fig 4.11. Viewpoint data structure constructed from a minimal set of Spatial Video parameters to define a maximal geographical extent.

## 4.6 Conclusions

So far the theoretical Viewpoint construct developed here consists of a point that defines the camera's location and orientation, and a ViewCone viewing frustum that defines the geographical extent of the image. This Viewpoint structure defines one video frame image stored as two separate data structures. However, these two structures are linked as a single dependent unit because one is calculated based on the properties of the other. To represent this in a GIS it is useful to use primitive data types, principally to ensure maximum applicability over as broad a range of GIS functionalities as is possible. However, 3D GIS data structures have still not been standardised. Both Zlatanova *et al.*, (2002) and De Floriani *et al.*, (1999) maintain that the multiplicity of 3D GIS data structures that have been developed provide relevant bespoke solutions to particular problems, however no consensus has yet emerged on a complete global 3D GIS data structure. For now, many GIS vendors implement multidimensional features based on best practice, reasonable requirements and the most likely future extensions of existing 2D structures that are already well defined. OGC and ISO standards have still not developed a complete guideline on multidimensionality for geo-spatial information. Thus, a number of options are possible, with the implemented solution for any specific Spatial Video sequence being a requirements-determined implementation rather than a single global solution.

In GIS modelling of the world, multidimensional approaches come from two distinct types of requirements: the modelling of solid objects and structures and the modelling of surfaces. In defining a geometric Viewpoint in 3D another important semantic difference should be discussed. Dependent on the requirements, either GIS modelling option could provide a viable solution. The difference is in the definition where a surface structure could warp the image to its geographical extent while a solid representation would be a variably complex solid.

For example, using a surface concept, a complex parabolic structure that represents an image’s geographic extent could be defined using a triangulated irregular network (TIN), as is shown in the simple image in fig. 4.12. This would be an oblique implementation of the structure as opposed to its more normal usage as an earth surface modelling concept. The surface implementation would also not be a closed object with a volume representation. This could, theoretically, lead to more complex problems with conceptually understanding this structure’s Viewpoint representation in a GIS geospatial analysis environment. Calculating surface area video coverage values would be an example.

Alternatively, defining the Viewpoint geometry as a solid object could be achieved through many methods. However, typically GIS multidimensional entity representations are of well-formed and regular objects, such as buildings. While a Viewpoint could be of varying levels of complexity that would reflect the detail of geographical accuracy, it is its closed form structural representation that is important. Hence a multidimensional structure that is semantically different to existing GIS usage of solids needs to be defined. In the context of Spatial Video, this is the object space of the image that does not contain the objects which would need to be converted into a 3D volumetric entity.

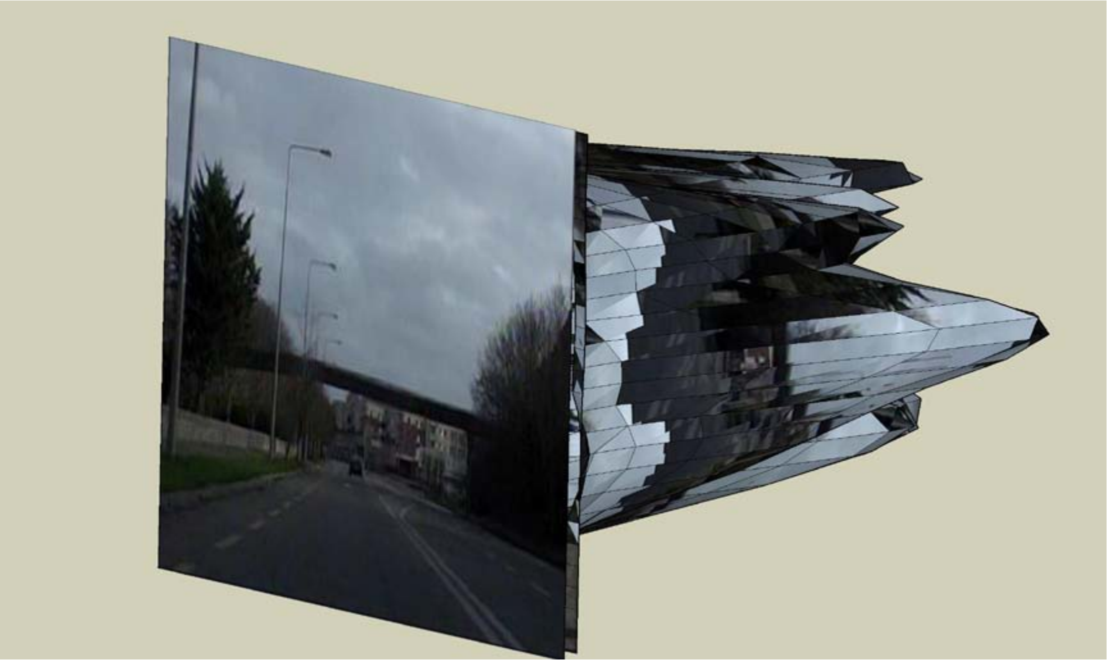


Fig 4.12. Conceptual GIS multidimensional surface representation of a Spatial Video Image using a parabolic TIN data structure. Here the image is warped to represent the surface.

Another existing, but not yet widely used, technology is mobile LIDAR which can define a 3D model of the survey environment. This data source creates a point cloud which could be used to form a very accurate model of the video's image object space. Coupling this data source with the general Viewpoints concept is another methodology that has future potential when building a spatial metadata source for an associated Spatial Video stream.

Discussed in the subsequent chapters are the implementations and applications of this theory in terms of its realistic calculation, implementation, analysis, improvement and usage.

# Chapter Five: Viewpoint Implementation

This chapter presents a Viewpoint implementation of a Spatial Video image frame into a GIS-compatible data structure based on the theoretical developments in chapter four. It will be shown how this implementation is an extension of the Open Geospatial Consortium ViewCone as defined in their Geo-Video Service specifications (Lewis, 2006). ViewCones define very simple GIS data structures as calculated from a video frame's known location and the video camera's operational parameters. To construct Viewpoints from first principles, and based on extending the ViewCone model, a defining set of assumptions is discussed in relation to the retrospective Spatial Video data sets available. Also, the precise parameters that should be recorded for any ongoing Spatial Video data collection are also investigated. What this 'base case' implementation will use is both empirically generated and accurately collected camera and spatial properties to construct a generalised and maximal Viewpoint spatial extent on retrospectively collected data.

## 5.1 Introduction

Two sets of experiments are presented in this chapter that provide proof-of-concept results towards a viable Viewpoint modelling approach for Spatial Video. The approach taken to this work is to implement the theory and measure the results based on a control set of images and camera parameters that are realistically representative of a normal Spatial Video image. The same process was then followed on images taken from retrospectively collected Spatial Video data images. A number of important assumptions are also stated regarding the accuracy and inherent uncertainty contained in this model.

## 5.2 Viewpoint Implementation Assumptions

The assumptions that are highlighted in this section represent restrictions placed on this model by the nature of the data being defined. In general, the large amounts of retrospective Spatial Video data that exist and, specifically, are available to this study have a large error range. Except for one video data stream that is available, this error range is due to the low quality GPS that was used in the data collection stage. Also, the camera equipment used for each video stream, while known, has not been provided with a calibrated operational parameters set. Other errors originate from the physical setup of the Spatial Video equipment where the offset values for the GPS unit and the camera sensor plane cannot be easily known.

The post-analysis usage of these Spatial Video data sets could afford to ignore such accuracy errors as the primary purpose was to provide an enhanced visualisation data source for a bespoke GIS. Accurate geo-referencing of the images was neither performed nor necessary. Thus, in these sections both the spatial and camera parameter setup assumptions that are relevant to the subsequent Viewpoint calculation sections are defined. While these restrictions simplify this model to a base case implementation and geographical extent approximation, a principle of a Viewpoint model is that such assumptions should be capable of being incorporated into the system at any level of accuracy. The direct result of varying levels of parameter accuracy would be in the bounding error of the geographical extent approximation being better defined. In a fully adaptive model, both the accuracy of the parameter sets that define the camera and spatial variables, and the adjustments to the geometric orientations, would only require better knowledge of the physical Spatial Video data collection setup.

### 5.2.1 Camera Calibration Model

In defining the set of assumptions for the camera model, both the camera operational parameter set and the orientation need to be considered. In all cases, the cameras used to collect the Spatial Video footage for this study are known, and listed in table 5.1. From the camera specification manuals, the following three operational parameter ranges for both the lens and camera body are acquired and used in the Viewpoint calculations:

1. Charged-coupled Device (CCD) array Sensor Size.
2. Lens F-Number Range.
3. Lens Focal Length Range.

With these parameters, estimations of the camera models as they were used in the past collection of data can be determined; however no defined camera calibration for the data collection period is available to improve or verify these estimations. Empirically derived comparisons of the image's calculated Viewpoints and the visual geographical extent of each image are the closest evaluations of these estimations.

| Equipment Type | Camera Model |
|---|---|
| Consumer Quality Mini DV Camcorder | Panasonic NV-GX150EB |
| Consumer Quality Mini DV Camcorder | Panasonic NV-GS180EB |
| Film Industry High Definition Movie Camera | JVC GY-HD111/Fujinon TH16 - 5.5 BRMU lens |
| Consumer Quality Digital Camera | Canon Powershot A85 |

Table 5.1. List of Spatial Video data capture devices used in the data collected and available to this study.

The cameras' operational parameter ranges are used to estimate the variables required in the Viewpoint models Field of View (FOV) and Depth of Field (DOF) plane equations; detailed in later sections. Define here are the assumptions in deriving the image sensor size and Circle of Confusion (COC) parameters. In general terms the calculation of these parameters is subjective, but sufficiently accurate for this test case implementation. They are subjectively defined because accurate parameters would require detailed calibration knowledge which is not available for the retrospective Spatial Video files available.

All the video footage that is used in this study was recorded using Charged-Coupled Device (CCD) digital camera sensor plane technology. The image sensor size is important as it is the physical plane that records the object space light rays. Currently, the specification documents that defined digital camera sensor sizes are based on legacy standards. These relate to video camera tube technologies that were pre-CCD development. As a result, accurate sensor sizes can only be ascertained by assuming the measurements based on acquired tables, (Bockaert, 2008; Kerr, 2008), manufacturer information, (Victor Company Of Japan (JVC), 2006; Panasonic Corporation, 2008; Sony Corporation, 2008), or physically dismantling and measuring the device. Using the former approaches, table 5.2 contains the sensor size calculations that were compiled for the list of cameras used in this study.

| **Camera Model** | **Datasheet Sensor Size** | **Horizontal(mm) (pixel size x effective pixel)** | **Vertical (mm) (pixel size x effective pixel)** | **Diagonal** $\sqrt{Hor^2 + Ver^2}$ **(mm)** |
|---|---|---|---|---|
| Panasonic NV-GS180EB & NV-GX150EB | 1/6” | 2.4 | 1.8 | 3.00 |
| JVC - GY-HD111 With Fujinon lens TH16-5.5 BRMU | 1/3” | 4.89 | 3.69 | 6.12 |
| Canon Powershot A85 | 1/2.7” | 5.27 | 3.96 | 6.59 |

Table 5.2. Spatial Video digital equipment sensor sizes. Specification data sheets only provide a height and width parameter, the diagonal parameter is calculated using Pythagoras theorem. Source: http://www.sony.co.jp/~semicon/english/90203.html and http://industrial.panasonic.com.

The second assumed variable calculation involves the Circle of Confusion (COC). A COC measurement defines the maximum permissible blur circle for an image and directly affects depth of field calculations. This parameter is subjective in so far as it should be considered in a calibration for depth of field and will also vary dependent on the final image output size and/or magnification. Its size has a relationship to the human visual system and display format which can be perceived or represented by the maximum resolution that appears sharply in focus. Any larger a COC and the image points will appear out of focus and blurred. A number of detailed discussions are available on this topic with most settling for a human vision related non-dimensional calculation of 1/1500 of the image sensor diagonal, (Evens, 2003; Wheeler, 2003; Conrad, 2006; Kerr, 2006B).

In a more complete assessment, related specifically to digital cameras, Lyon, (2006), suggests an implementation that considers the COC to pixel size ratio. From this it can be deduced that a relative COC to pixel size constant can be calculated based on the sensor pixel diagonal to COC ratio. For an aspect ratio of 4:3 this would be a constant of .67, for 16:9 it is 1.59 and for 5:4 it is 1.49:

$$c = sensor\_diagonal/(aspect\_ratio\_constant\sqrt{pixel\_count}) \qquad (5.1)$$

Thus, for any megapixel value at this aspect ratio a relative COC can be defined. Table 5.3 details both implementations for the known list of cameras and shows the minor difference in the resulting values; all further calculations use a COC to three decimal places.

| Camera Model | Format Size pixels | Megapixels in pixels | Equation 5.1 | (Sensor Diagonal) / 1500 |
|---|---|---|---|---|
| Panasonic NV-GS180EB & NV-GX150EB | 720x576 | 414,720 | 0.00300mm | 0.00200mm |
| JVC - GY-HD111 With Fujinon lens TH16 - 5.5 BRMU | 1280x720 | 921,600 | 0.00401mm | 0.00408mm |
| Canon Powershot A85 | 2272x1704 | 3,871,488 | 0.00523mm | 0.00439mm |

Table 5.3. Circle of Confusion calculations using both literature recommendations for the known list of cameras.

### 5.2.2 Camera Spatial Model

Two assumptions define the camera spatial model in this section, firstly, the coordinate system for the calculations and, secondly, the camera's spatial orientation. The spatial model coordinate system for these experiments will remain geometrically spherical and use decimal latitude and longitude variables. While a transformation to simpler planar geometric coordinate systems is easily done, it is felt that the methods and algorithms that are readily available for accurate calculations in the spherical model are as appropriate an implementation methodology as projections into a planar coordinate system and back again. Where necessary, viewpoints are projected into a planar coordinate system for easier visualisation.

When defining the camera spatial model orientation two assumptions are made about the image plane: firstly, it is vertically perpendicular to the traversal surface, and, secondly, the optical axis is coincident with the spatial azimuth or direction of travel, as is shown in fig 5.1. These assumptions simplify the geometric calculations in the model; however, incorporating orientation adjustment parameters is only an exercise in redefining the geometric calculations into a more complex model. The result of this assumption is a more accurate approximation of the viewpoint geographical extent in the controlled experimental model as opposed to the larger error bound on the retrospective Spatial Video data set calculations.

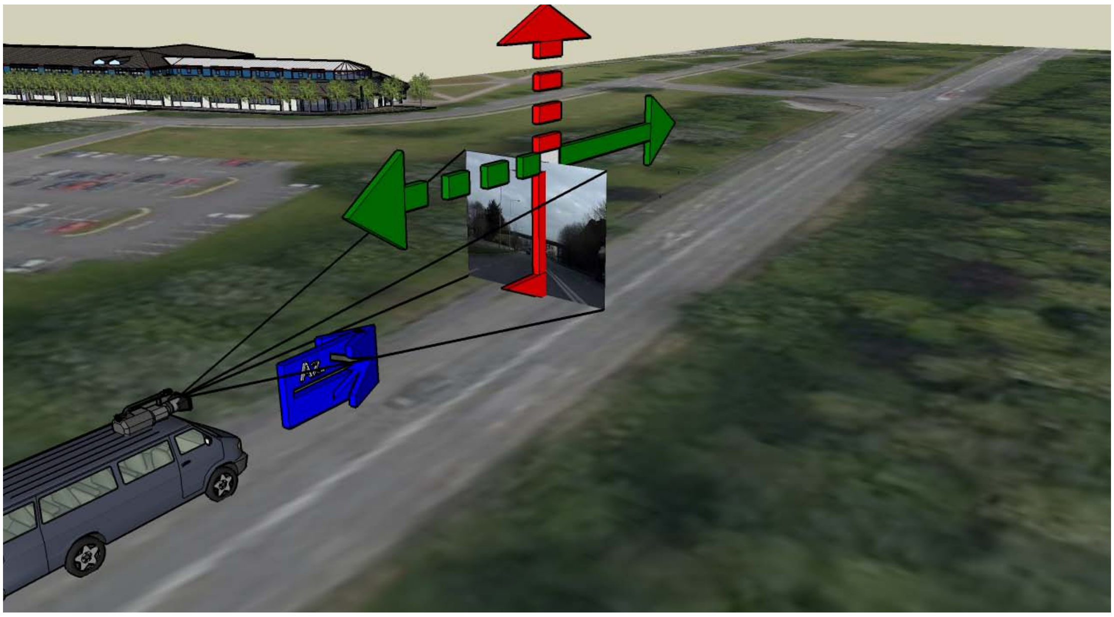


Fig. 5.1. Visualisation of Camera Model spatial assumptions. The Red axis is the image axis perpendicular to the road surface; the Green axis is the optical axis coincident with the (blue) Azimuth direction of travel.

In a complex fully calibrated Spatial Video system, these spatial orientation assumptions would be reduced or eliminated by inclusion, as an example, of an Inertial Navigation Systems (INS) sensor. The relative roll, pitch, yaw and velocity parameters from such a device would help define the Viewpoint structure of the camera image and could easily be incorporated into the geometrical model. Given that the spherical geometry implementations are based on a geodetic model that is a locally optimised ellipsoid, extrapolation of Viewpoint geometry will be consistent across the spatial content of the video. However, the robustness of the system is dependent on these assumptions as camera orientations outside these restrictions will not have their Viewpoints implemented correctly. This is an important point for a broader implementation of the Viewpoint theory; however, given the retrospective Spatial Video that was available and used in this project such modelling of a fully rotational geometrical space was unnecessary.

Finally, the rays of light that define the captured image in the camera model have other spatial aspects that are ignored which include lens distortions and light diffraction. While modern lens systems are highly accurate and minimise distortions significantly to measurements of approximately $5\mu m$, (Wolf *et al.*, 2000), they are normally only modelled in high accuracy analytical photogrammetric situations. Light refraction is also not modelled in this implementation as the distances that define most Viewpoint calibrations are insignificantly small; however they would be important in a higher accuracy calibrated system, especially if the camera is recording internally on the survey vehicle behind glass.

## 5.3 Camera Model Equations

Defined in this section are the principle camera concepts, and associated formulae, that are used to calculate an approximate geographical extent for any given Spatial Video frame. The equations and discussions used in all sub-sections here are taken from (Kerr, 2006A, 2006B), they are standardised across all the literature which is widely available from many other sources. An in-depth discussion of each of these concepts is not necessary in terms of its applicability in forming a Viewpoint as it is only one possible implementation methodology from the many relevant photogrammetric ones available. What these concepts do provide is a very simple methodology that uses camera parameters to approximate the Viewpoint data structure.

Typically, these equations are used by photographers in a setup context where they will maximise the camera’s parameters for any particularly desirable shot. This normally involves being able to define and/or measure various scene parameters to implement these equations accurately. In particular, subject distance is important to most equations for image optimisation. In a retrospective Spatial Video context this is not easily possible, especially in high frame rate video capture situations where determining the camera's optimal subject distance would be very difficult. To accommodate this; camera, rather than scene-specific, parameters are used to approximate the camera calibration equations.

### 5.3.1 Angle of View

In general, Field-of-View (FOV) is a definable measure based on the maximum viewable extent of a visual system. It has a direct relationship to the amount of a scene that is viewable from a point; as an example the human visual system has approximately a 180° range. When implemented in a photogrammetric context, FOV can be confused with an Angle-of-View (AOV). A FOV has linear dimensions where regions are typically defined in terms of width, height, feet, meters etc., and is very useful for photographic setup. In calculating an FOV, knowledge of the subject distance is required. Alternatively, an AOV represents the camera lens properties as an arc angle. This is a dimensionless representation of the images object space, i.e. it does not define a metric type measurement relevant to a specific scene. This angular measure is not as useful to practical photography as a FOV description of a scene. Trying to describe, for example, that 80° of the scene will be captured from this point as opposed to saying a 100 meter wide object 20 meters from here will be captured is less intuitive. Given that the object space focusing distance for every given Spatial Video frame image is not explicitly known, using the AOV approach is preferable.

An AOV for a camera lens and rectilinear sensor setup defines three angles; the horizontal, vertical and diagonal; which can be calculated using the following equation:

$$A = 2\arctan\frac{d}{2f} \tag{5.2}$$

Equation 5.2 defines the angle A of the object space whose apex is the centre of the entrance pupil of the lens. The camera sensor size is *d* and represents either the horizontal, vertical or diagonal measure, while *f* is the focal length of the lens. In the case of Spatial Video, lens focus is nearly always over large distances which make this equation appropriate. Otherwise, the *f* term would need to be replaced by the distance from the lens' second nodal point to the focal plane for images where the focal plane is close to the camera lens. Implementing an AOV allows us to calculate the approximate geographical space boundary extents for the top, bottom, left and right Viewpoint planes.

### 5.3.2 Depth of Field

In a discussion on Depth-of-Field (DOF) two important concepts arise which add subjective quality issues to a measurable quantity range. A DOF will represent an image focus range as a set of two distance measures in front of the camera lens. As light rays from the object space converge on the image plane they each focus on differing points. Only light from one object space plane will see all these points resolved to a precise image plane point. All other object space points will form imprecise points known as blur spots or, as pointed out earlier, Circles of Confusion (COC). So, any objects located inside this DOF range will be captured on the sensor image plane in focus, i.e. sharply focused, while objects outside this range will be blurred. Also, this range assumes the final image will be viewed under normal conditions determined by the setup calibration, i.e. the images will not be magnified or viewed too closely.

Thus, subjectivity surrounds the definitions of what is sharply focused, or what are normal conditions. It is not within the purview of this study to address these questions. However, the Viewpoint implementation does assume that the video will be viewed in its original capture context and so avoids normal viewing condition problems. It is also assumed that the COC is constant across all Spatial Video collection sets for any particular camera which at least introduces a consistent if not highly accurate result. Remembering the section on COC and its inherent subjectivity, more indiscriminate considerations are examined in order to define a measurable DOF range. However, given that the objective is to define a geographical extent for a large amount of Spatial Video frames where spatial indexing and searching in a GIS context is facilitated, then this approach is acceptable.

In its correct context a DOF is primarily used to calibrate an image setup where all these questions can be answered with relative assurance that the image will be focused where it matters for the object space. Under these setup conditions, and like the FOV, object distance knowledge is important to perform accurate calculations. As mentioned before, this knowledge is not easily determined over such a large data set for every Spatial Video frame, so the approach taken here is to implement a Hyperfocal Distance DOF calculation.

#### 5.3.2.1 Hyperfocal Distance

A Hyperfocal Distance defines a measurable distance in front of the camera lens from which point to infinity the DOF extends. Based on this, any object light ray will be considered focused onto the image plane, within the bound of acceptable sharpness that has been defined through the COC, if it originates from any point in space beyond half this distance. In aerial survey situations the camera lens is normally calibrated for infinity focus, which is perfectly appropriate as the object space will nearly always be at the far extent of the lens focus range. In the oblique terrestrial situation a lens will normally have its focus determined by the survey requirements and would very seldom be set to infinity as this would minimise the DOF range. Thus, a number of different DOF ranges could be appropriate to not only different Spatial Video surveys but also to different sequences within any given survey. Therefore, given the camera's close range to the terrestrial survey surface and the variability in terrain distances from the camera lens, it is more appropriate to assume a maximum DOF range. This is achieved by applying a Hyperfocal Distance measurement across the whole Spatial Video data stream.

A Hyperfocal Distance is calculated using the following equation:

$$D_h = \frac{f^2}{nc} + f \qquad (5.3)$$

The Hyperfocal Distance $\boldsymbol{D}_{\mathrm{h}}$ given in equation 5.3 has parameters *f* being the actual lens focal length, *n* is the lens aperture as an F-number and *c* is the COC diameter limit. To calculate the near limit of the Hyperfocal Distance $\boldsymbol{D}_{\mathrm{nh}}$ the following equation is used:

$$D_{nh} = \frac{D_h}{2} \frac{1}{\left(1 - \frac{nc}{f}\right)} \tag{5.4}$$

In implementing these equations the following situations should be considered. Firstly, most Spatial Video surveys will have the video capture equipment set to auto-focus where the aperture F-number value could be dynamically changing from one frame to the next. The result of this would be a near limit Hyperfocal Distance change for frames where auto-focus has changed the lens aperture. In this set of test implementations, accurate acquisition of this parameter value is trivial. However, in a complete Spatial Video data set this is not easily measurable in a dynamic context.

An empirically defined far focus limit is also generated based on the Hyperfocal Distance near focus limit. The context for this is in the chapter four discussions on the logical reasons why this would be desirable in a Spatial Video context. As has been mentioned earlier, scene setup and object distances are not known from frame to frame so the normal DOF near and far focus limit equations are not appropriate. Thus, the Hyperfocal Distance per frame is implemented and extended to an arbitrary far focus limit. This far focus limit is defined simply as:

$$D_{nh} + FAR\_FOCUS\_DISTANCE(mtrs) \tag{5.5}$$

Finally, implementing these near and far Hyperfocal Distance limits allows us to define the Viewpoint model's near and far geographical bounding planes. When incorporating these planes with those calculated by the AOV equations a polyhedron is constructed, as described in chapter four and shown in Fig 4.11, which approximately represents the maximum possible geographical extent based on a minimal set of image capture parameters. The final stage in building a Viewpoint is relating these planes to the geographical space.

## 5.4 Spatial Data Parameters and Equations

Discussed in this section are the final elements that are required to determine a Viewpoint model. So far a camera parameters model that provides a polyhedral data structured of a video frame's image object space has been defined. To represent this in a geographical context the model needs to be incorporated into a spatial domain by using known spatial parameters for each frame to solve projective spherical geometry systems that will define the Viewpoint's spatial extent. Global Positioning System (GPS) data provide the known location information which is then used in a geodetic spherical geometry model to solve and construct the Viewpoints in geographical space.

### 5.4.1 Global Positioning System Data

The minimum level of spatial data collected with a Spatial Video stream is a GPS NMEA message list. This source of spatial data provides a number of parameters in the form of a formatted sentence string. Each string begins with a sentence identifier followed by a comma delimited list of data fields. All the variables used in this study are taken from the $GPRMC and $GPZDA strings. CommLinx (2003) is a source for these parameter string descriptions. In this experimental case the GPS latitude, longitude and azimuth are used. The altitude could be used in the 3D implementation, however, appropriate consideration for the much larger error bounds present in this vertical axis are required. These variables are modelled as a GIS point data type, as described in chapter four.

In the experimental cases described here two levels of GPS were used. For the retrospective Spatial Video image test the standard civilian GPS signal was used while the calibrated survey test used Real Time Kinematic (RTK) GPS. In the RTK GPS case the accuracy of the positional parameters is approximately 1cm horizontally, thus it provides very robust calibration data for the survey area being modelled through the Viewpoint concept. Countering this is the levels of inaccuracy inherent to the civilian GPS signal where the Spatial Video test data required supervised adjustment to attain an accurate Viewpoint representation of the geographical space. This is discussed in more detail in the test results section.

### 5.4.2 Spatial Extrapolation Steps

Here the methods that will be followed to extrapolate the spatial locations of the Viewpoint polygon data structure extents in 2D space are discussed. The following steps are performed:

1. Adjust the GPS coordinates to be coincident with the principle point of the camera image plane.
2. Calculate an adjusted Hyperfocal Sharpness Distance to the eight Viewpoint plane intersection points.
3. Calculate an adjusted azimuth.
4. Use these results to solve the geodetic forward algorithm as defined in (Vincenty, 1975) through code available from the National Geodetic Survey USA, (2006).

In a 3D context only one other step needs to be included where a calculation to adjust the altitudes of the Viewpoint plane intersection points is required.

Based on the assumptions mentioned in section 5.2.2, the 2D test implementations will define a planar slice through the 3D Viewpoint as shown in fig. 5.2. Using these assumptions for the first step, an adjustment of the GPS coordinates to the camera image plane is required. In these test cases this process is only necessary in the latitude and longitude planes; an altitude adjustment would also be required for the 3D calculations. For the survey test data set no adjustment is performed as an RTK GPS reading was captured from the camera location point. In the retrospective data test a very simple planar Cartesian adjustment is applied.

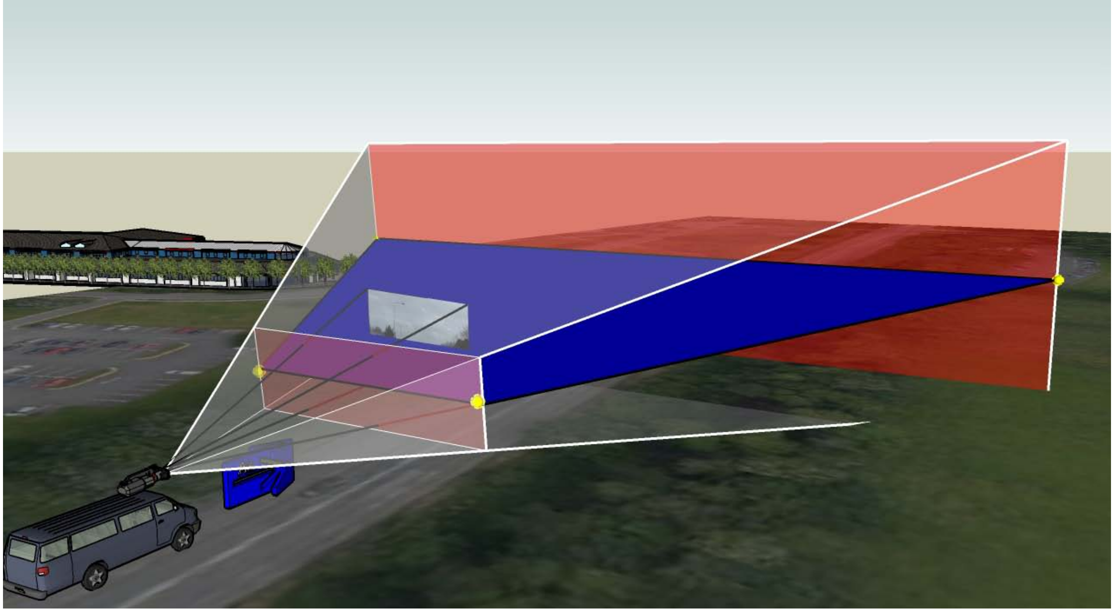


Fig 5.2. 3D Spatial Video Viewpoint with its modeled ViewCone highlighting the blue 2D planar slice which this Chapters test implementation defines. Also shown are four yellow spheres which are the spatial parameters defining the ViewCone slice.

Step two calculates a slight adjustment of the Hyperfocal Sharpness distance to accurately measure the distance to the ViewCone boundary plane intersection points, the yellow spheres shown in fig. 5.2. As the Hyperfocal distance calculates the distance from the lens apex to the centre of the near focus limit plane it is necessary to perform this calculation. The calculation is based on a right angle triangle and half the horizontal AOV. A sine rule equation is applied and can be visualised in fig 5.3. This method is also applied to calculate the far focus limit plane and its intersection points by utilising the Hyperfocal Sharpness Distance plus the *far_focus_distance* constant.

Step three is a simple adjustment of the azimuth based on half the horizontal AOV. This is either an addition or subtraction of the azimuth’s value depending on the true north orientation of the defining angle. The only other consideration here is an adjustment where the angle exceeds 360° or falls below 0°.

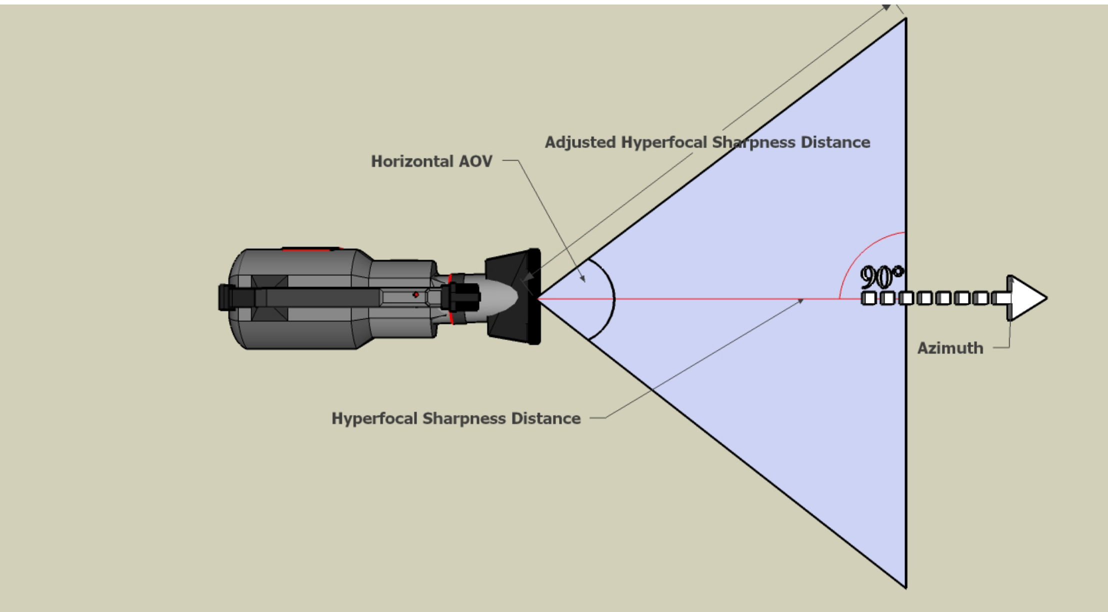

Adjusted Hyperfocal Sharpness Distance
Horizontal AOV
90°
Azimuth
Hyperfocal Sharpness Distance

Fig 5.3. Calculation of Hyperfocal Sharpness Distance.

Finally, step four uses the geodetic direct extrapolation formulae to define the spatial locations of the four 2D Viewpoint plane intersection points, (Vincenty, 1975). These formulae have been implemented in a Fortran program written at the National Geodetic Survey USA (2006), however this has been converted to Visual C# for this project; it is listed in appendix one. This algorithm is accurate to 0.5mm on the earth ellipsoid being used over any distances as long as they are not antipodal; however this accuracy will not hold unless locally adjusted ellipsoids are used. In this study an Airy modified Ellipsoid for Ireland was used. The direct algorithm extrapolates the Viewpoints ViewCone latitudes and longitudes based on the known location of the camera image plane, the adjusted Hyperfocal Sharpness distance to each point and the adjusted azimuth.

## 5.5 Calibrated Image Data Test

In this test a Spatial Video survey setup and collection scenario is replicated. The difference in this case is that the setup variables can be determined and recorded before image collection rather than having to be empirically determined as happened in the retrospective test. Because this is a test case scenario images were collected from a static location rather than video footage from a moving platform. Each image shot was setup such that the operational parameters of the camera are known before image exposure. A visual alignment of the GPS device and the image object space extreme points, as seen in the camera viewfinder, was used to record the geographical locations of the proposed Viewpoint calculations.

The camera used was a Canon Powershot A85 and was setup on a levelled tripod. Table 5.4 shows this camera’s spatial and operational range. The RTK GPS unit was used on a levelled survey rig to record the camera location and the image object space boundary points. These data sets were then used as a set of controls for the hypothetical Viewpoint extent calculations. The Viewpoints for these control images were calculated based on the recorded parameters collected when the image exposure was taken. Results were then tabulated for comparison and mapped for visual analysis.

| **Focal Length** | **Aperture F-Num** | **Image Sensor Size** | **Latitude** | **Longitude** | **Azimuth** |
|---|---|---|---|---|---|
| 5.4 to 16.2 | 2.8 to 4.8 | 1/2.7 inch | 53° 23' 4.5924" | 6° 36' 4.2912" | $346.43^{0}$ |

**Table 5.4. Canon Powershot A85 operational parameter ranges and RTK GPS spatial location and orientation data.**

### 5.5.1 Test Setup and Data Collection

This test took place on the football playing fields at NUIM’s north campus. A set of camera maximum and minimum zoom images were recorded for analysis and comparison against the Viewpoint calculations. Using the camera viewfinder, the RTK GPS unit was positioned at the image object space boundaries and recorded. This sequence of image capturing and camera/GPS parameter recording is shown in the following sequence of images and variables in table 5.5.

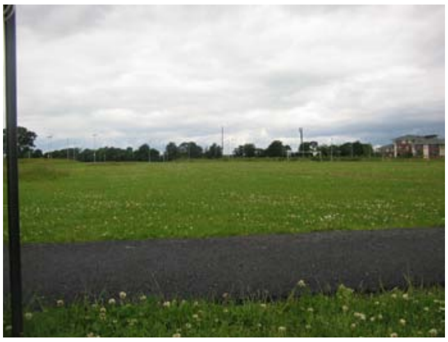
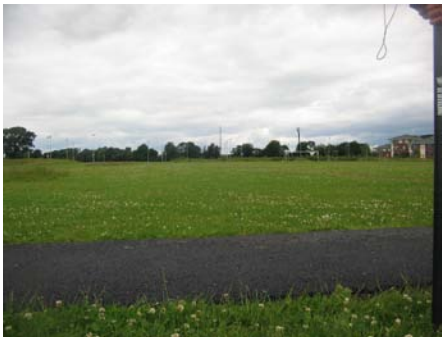
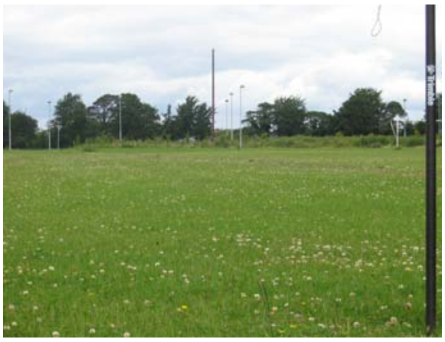
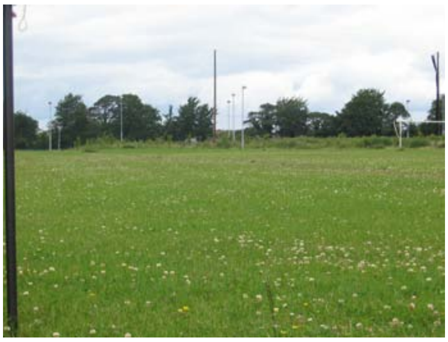
| | | |
|---|---|---|
| **Image: Test003.jpg**<br> | **Focal length** | 5.4 |
| | **F-Number** | 3.2 |
| | **Object Space GPS Location** | Bottom Left |
| | **RTK GPS Latitude** | 53° 23’ 4.6308” |
| | **RTK GPS Longitude** | 6° 36’ 4.3444” |
| **Image: Test004.jpg**<br> | **Focal length** | 5.4 |
| | **F-Number** | 3.2 |
| | **Object Space GPS Location** | Bottom Right |
| | **RTK GPS Latitude** | 53° 23’ 4.6441” |
| | **RTK GPS Longitude** | 6° 36’ 4.2729” |
| **Image: Test005.jpg**<br> | **Focal length** | 16.2 |
| | **F-Number** | 4.8 |
| | **Object Space GPS Location** | Bottom Right |
| | **RTK GPS Latitude** | 53° 23’ 4.7223” |
| | **RTK GPS Longitude** | 6° 36’ 4.3089” |
| **Image: Test006.jpg**<br> | **Focal length** | 16.2 |
| | **F-Number** | 4.8 |
| | **Object Space GPS Location** | Bottom Left |
| | **RTK GPS Latitude** | 53° 23' 4.7135" |
| | **RTK GPS Longitude** | 6° 36' 4.3799" |

Table 5.5. Continued on next page.

| | | |
|---|---|---|
| **Image: Test007.jpg**<br> | **Focal length** | 5.4 |
| | **F-Number** | 3.2 |
| | **Object Space GPS Location** | Middle Left |
| | **RTK GPS Latitude** | 53° 23' 5.1226" |
| | **RTK GPS Longitude** | 6° 36' 4.9645" |
| **Image: Test008.jpg**<br> | **Focal length** | 5.4 |
| | **F-Number** | 3.2 |
| | **Object Space GPS Location** | Middle Right |
| | **RTK GPS Latitude** | 53° 23' 5.4028" |
| | **RTK GPS Longitude** | 6° 36' 3.9236" |
| **Image: Test009.jpg**<br> | **Focal length** | 16.2 |
| | **F-Number** | 4.8 |
| | **Object Space GPS Location** | Middle Right |
| | **RTK GPS Latitude** | 53° 23' 5.5752" |
| | **RTK GPS Longitude** | 6° 36' 4.3260" |
| **Image: Test010.jpg**<br> | **Focal length** | 16.2 |
| | **F-Number** | 4.8 |
| | **Object Space GPS Location** | Middle Left |
| | **RTK GPS Latitude** | 53° 23' 5.4886" |
| | **RTK GPS Longitude** | 6° 36' 4.8476" |

Table 5.5. List of captured images, operational parameters and RTK GPS recorded positions.

Fig 5.4 shows a plan view of these RTK GPS point locations on the NUIM Campus.

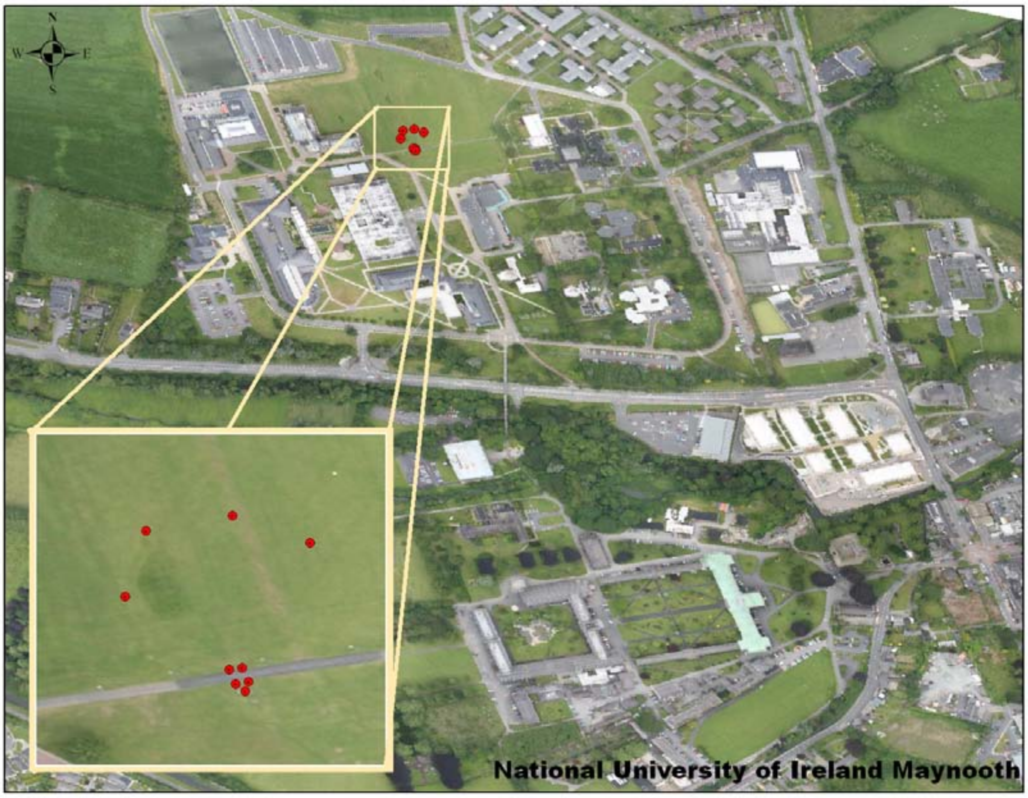


Fig. 5.4. Calibrated Image test survey area RTK GPS points overlaid on a NUIM Orthophoto.

### 5.5.2 Viewpoint Calculations

The calculations of the 2D Viewpoint parameters are shown in this section. These are based on the equations and procedures mentioned previously. Two sets of data are presented here that represent an amalgamation of the eight images into sets based on matching camera parameters as follows:

1. Viewpoint one image set:
    - Near Field - Test003.jpg and Test004.jpg
    - Far Field - Test007.jpg and Test008.jpg

2. Viewpoint two image set:
    - Near Field - Test005.jpg and Test006.jpg
    - Far Field - Test009.jpg and Test010.jpg

The resulting Viewpoint calculation parameters and ViewCone spatial locations' variables are presented in table 5.6, where the Hyperfocal Distance far focus field constant is 100 meters, the Sensor Diagonal is 6.59mm and the COC is .004mm.

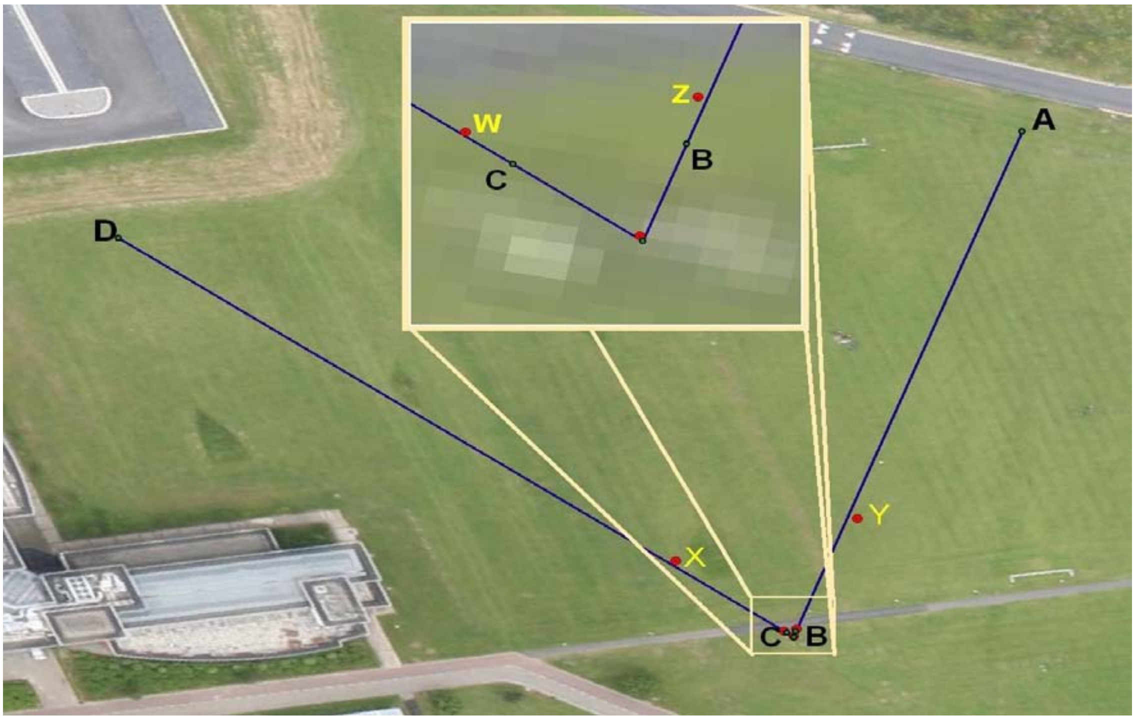
| **Operation** | | **Viewpoint One** | **Viewpoint Two** |
|---|---|---|---|
| Horizontal Angle of View | | 52.02° | 18.48° |
| Hyperfocal Sharpness Distance (HSD) | | 1.042mtrs | 6.237mtrs |
| Adjusted Near HSD | | 1.159mtrs | 6.319mtrs |
| Adjusted Far HSD | | 112.43mtrs | 107.633mtrs |
| Adjusted Left Azimuth | | 320.42° | 337.19° |
| Adjusted Right Azimuth | | 12.44° | 355.67° |
| Near Left Viewpoint | Latitude | 53° 23' 4.61882" | 53° 23' 4.77816" |
| | Longitude | 6° 36' 4.32987" | 6° 36' 4.42236" |
| Near Right Viewpoint | Latitude | 53° 23' 4.62653" | 53° 23' 4.79353" |
| | Longitude | 6° 36' 4.27652" | 6° 36' 4.31579" |
| Far Left Viewpoint | Latitude | 53° 23' 7.39295" | 53° 23' 8.06165 " |
| | Longitude | 6° 36' 8.16585" | 6° 36' 4.72983" |
| Far Right Viewpoint | Latitude | 53° 23' 8.14139" | 53° 23' 7.79935" |
| | Longitude | 6° 36' 2.97955" | 6° 36' 6.54747" |

**Table 5.6. Resultant Viewpoint parameters based on collected data calculations.**

### 5.5.3 Results

In this section both visual and tabulated results for the calculated Viewpoint parameters are presented. These results show the distance of the recorded RTK GPS points from the lines projecting from the camera's location that pass through the Viewpoint ViewCone near and far focus plane points. Whether the control points fall inside or outside the Viewpoint points is also shown in tables 5.7 and 5.8. A complete table of all the calculations and detailed results is available in appendix eleven.

**Viewpoint One**

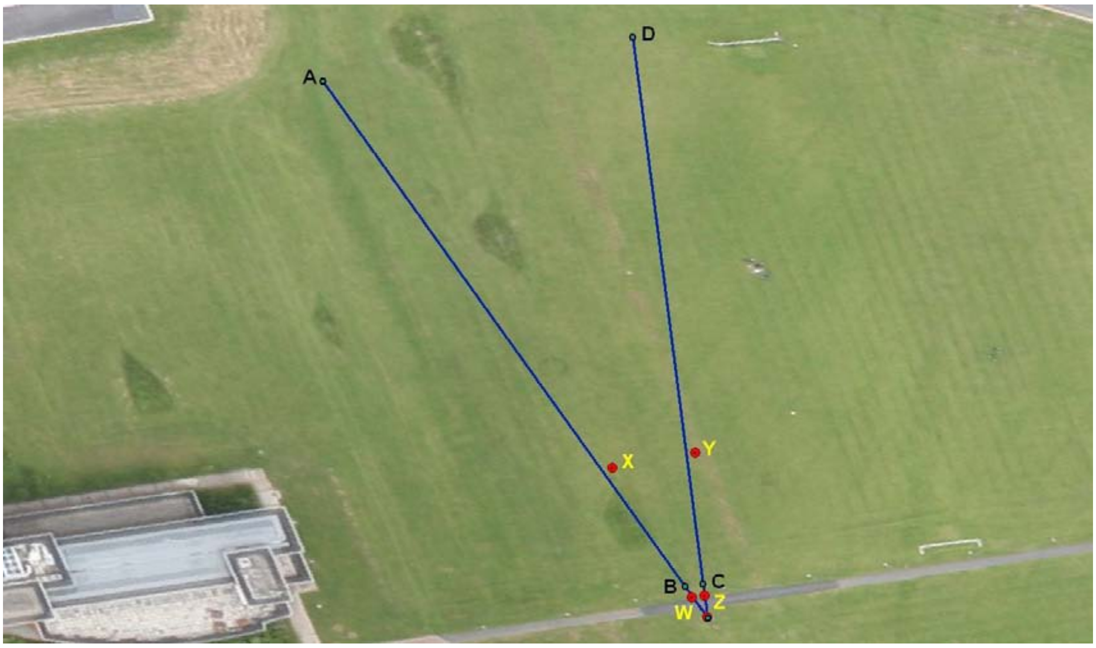


| Line | Point | Point to Line Distance in Meters | Result |
|---|---|---|---|
| C to D | X | 0.798mtrs | Inside |
| C to D | W | 0.026mtrs | Inside |
| A to B | Y | 1.066mtrs | Outside |
| A to B | Z | 0.051mtrs | Inside |

Table 5.7. Viewpoint One Plan View Results.

The average offset distance difference in the Viewpoint One results is 0.485mtrs. The near focus limit is defined by the plane created from point C to B and is clearly in front of the RTK GPS points W and Z. While the far focus limit is defined by the plane created from point D to A. The points W and Z define the bottom corner points where the image object space intersects with the terrain surface. The C to B plane defines the near focus limit where captured geographical space in this image is in focus from beyond this point; however this plane does not intersect with the terrain surface at this point. Thus, in a 3D context, the geographical space captured between the C-B and W-Z planes is above the terrain and represents space where its bottom plane is at a definable altitude above the traversal surface.

**Viewpoint Two**



| Line | Point | Point to Line Distance in Meters | Result |
|---|---|---|---|
| A to B | X | 1.219mtrs | Inside |
| A to B | W | 0.049mtrs | Outside |
| D to C | Y | 0.972mtrs | Outside |
| D to C | Z | 0.039mtrs | Inside |

Table 5.8. Viewpoint Two Plan View Results.

The average offset distance difference in the Viewpoint Two results is 0.57mtrs. Conversely to the situation in Viewpoint One, the near focus limit defined by the B-C plane is clearly beyond the RTK GPS image bottom corner points W and Z. This focus limit has determined that the geographical space captured between the W-Z and B-C planes as not being sharply focused. Thus, viewing images Test005.jpg and Test006.jpg and determining if the near geographical terrain, which is captured in this portion of the image object space, is in focus is a subjective answer.

## 5.6 Spatial Video Image Data Test

In this test the camera operational parameters are defined empirically based on the Viewpoint implementation principles. These parameters are then used as the base Viewpoint calculation variables for all further frames in the associated Spatial Video stream. Two Spatial Video test data sets were used, one for each camcorder, which are defined in table 5.9.

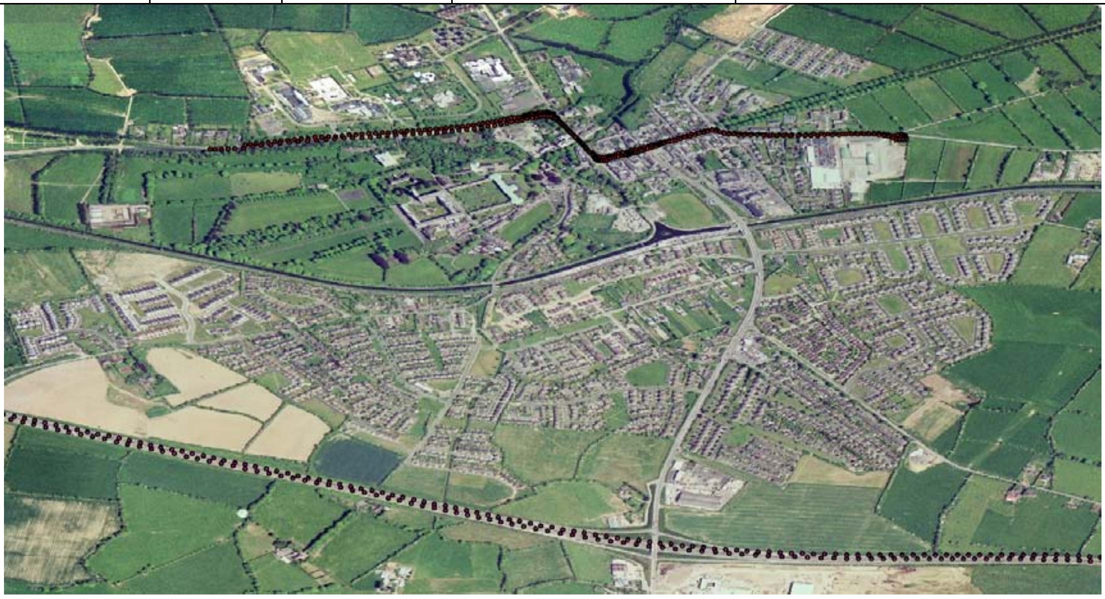
| Camera | Focal Length | Aperture F-Num | File and Format | Route Description |
|---|---|---|---|---|
| Panasonic NV - GX180EB | 2.45 to 24.5 | 1.8 to 4.8 | Route2.wmv – Windows Media Video Interleaved | Kilkock Road – Maynooth Main Street – Leixlip Road – Tesco Roundabout – Return. |
| JVC GY – HD111 | 5.5 to 88 | 1.4 to 16 | Front.mp4 – MPEG4 High Definition | Kilkock M4 Interchange – Along M4 – Leixlip West Interchange – Return. |
|  | | | | |

Table 5.9. Spatial Video streams route descriptions and plane view; the upper track is the Panasonic one while the lower track is a portion of the JVC one. The associated camcorder operational ranges used to define the parameters in this test are also included.

The process that was followed here was to take sample images and compute Viewpoints from varying camcorder parameters. Based on a simple visual appreciation of the video's footage, it was easily determined if the angle of view was wide or narrow at the time of data collection. In both cases a wide angle setting was assumed for the initial Viewpoint calibration tests followed by successively narrower approximations. Each parameter set was refined until an acceptable representation of the geographical space captured in the sample image's object space was achieved in the Viewpoint extent that defined this space.

### 5.6.1 Test Errors

The inherent Spatial Video errors that are present in any of the data sets used in this test all relate to GPS accuracy. It was assumed at the outset of this test that an approximate fifteen meters error range would be present in the civilian standard GPS used. However, in the test video data sets this error range did not exceed five meters in any measured situation. Fig 5.5 shows a GPS error correction of 4.2 meters for the Panasonic camcorder image shown in table 5.10.

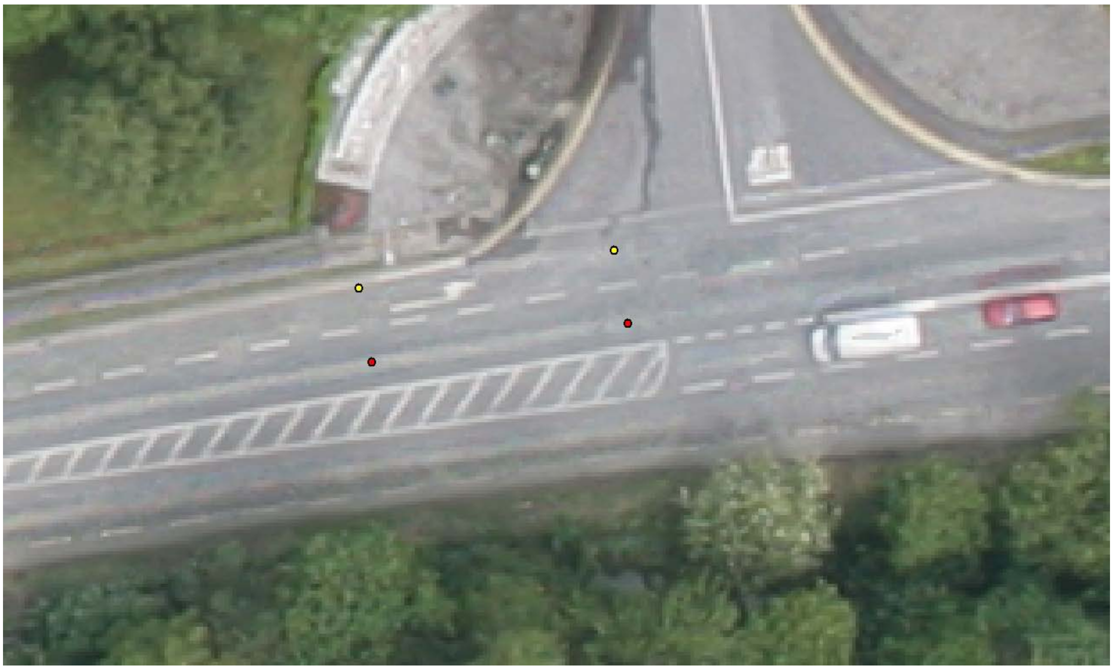


Fig 5.5. Orthophoto of NUIM entrance with original GPS positional errors shown as the yellow points and the corrected points shown in red.

In each video stream situation a line was defined for the track over an appropriate Orthophoto based on the manual visual sampling of the footage. The video associated GPS track was then snapped to this line with the resulting GPS point offsets providing the basis for all Viewpoint calculations. The other source of error was the spatial and video frame association. In the Spatial Video systems used in these data collections the video was captured at 25 hertz while the GPS was acquired at 1 hertz. However no synchronisation information is provided or calibration stage performed to determine this relationship. In McCarthy *et al.* (2008) a timing test was performed on this type of data collection with conclusions determining a 1.51 meter error when travelling at 100km/hr.

It is known that this sort of error is systematic for a given data set and as such is relatively easy to factor into the experiment. For the test images chosen from the two data sets, the capture locations and Viewpoint extents subjectively represent the geographical space satisfactorily. Thus, if the Viewpoint calibration does have an error in the video frame GPS spatial variable relationship, then this error is the same for all subsequent and previous frames given. This assertion is dependent on the Spatial Video data set being consistent in its composition where no frame or GPS rate change happens.

### 5.6.2 Viewpoint Calculations and Results

As has been mentioned earlier, the determination of the camcorder operational parameters was performed on an incremental basis starting with the settings that define the camcorder at its widest angle of view. This angle was progressively reduced, i.e. the focal length and F-Number increased, and visually compared to the chosen set of random video frames. Once it was determined that the calculated Viewpoint provided an accurate representation of each image’s geographical space, these operational parameters were recorded. They would subsequently be used to define the Viewpoints for all frames in the associated Spatial Video data stream.

Table 5.10 contains two of the random Spatial Video images and the camcorder operational parameters that have been empirically tested to provide an accurate Viewpoint for any video frame image in their respective data sets.

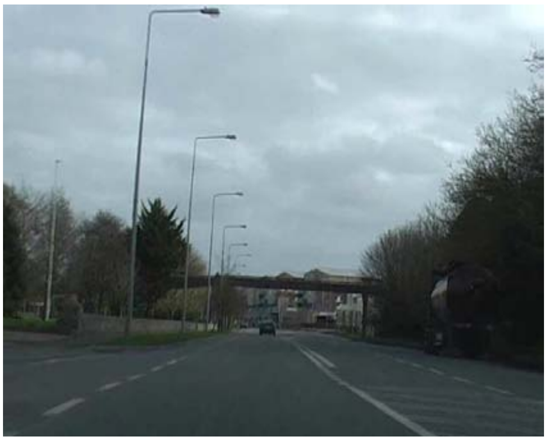
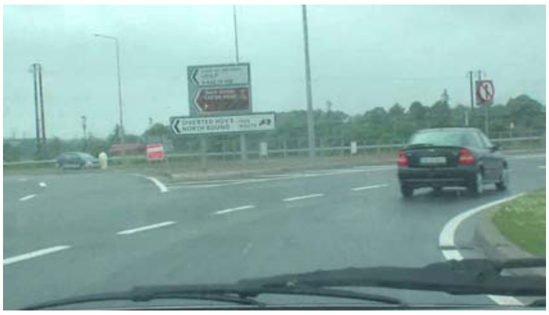
| | | |
|---|---|---|
| **Panasonic NV-GX180EB Image**<br> | **Focal length** | 4.5 |
| | **F-Number** | 1.8 |
| | **Azimuth** | 82.61° |
| | **GPS Control Latitude** | 53° 22' 54.88799" |
| | **GPS Control Longitude** | 6° 36' 6.14400" |
| **JVC – GY-HD111 Image**<br>**(High Definition)**<br> | **Focal length** | 10.5 |
| | **F-Number** | 4.8 |
| | **Azimuth** | 47.80° |
| | **GPS Control Latitude** | 53° 21' 37.49400" |
| | **GPS Control Longitude** | 6° 31' 58.89000" |

Table 5.10. Empirically tested camcorder operational parameters for two of the Spatial Video data set images sampled.

The Viewpoint calculation parameters are define in table 5.11 with figures 5.6 and 5.7 showing the plane view of the resultant Viewpoints for visual comparison against the images shown in table 5.10. While a close visual comparison is subjective in determining these results it has been shown to be effectively accurate for the subsequent chapter’s geo-spatial analysis operations sections.

| **Operation** | | **Panasonic Viewpoint** | **JVC Viewpoint** |
|---|---|---|---|
| Sensor Diagonal | | 3.00mm | 6.12mm |
| Circle Of Confusion | | 0.002mm | 0.004mm |
| Horizontal Angle of View | | 29.86° | 26.27° |
| Hyperfocal Sharpness Distance (HSD) | | 2.536mtrs | 2.823mtrs |
| Adjusted Near HSD | | 2.596mtrs | 2.900mtrs |
| Adjusted Far HSD | | 105.037mtrs | 105.573mtrs |
| Adjusted Left Azimuth | | 67.68° | 34.69° |
| Adjusted Right Azimuth | | 97.54° | 60.91° |
| Near Left Viewpoint | Latitude | 53° 22' 54.91988" | 53° 21' 37.57093" |
| | Longitude | 6° 36' 6.01411" | 6° 31' 58.80098" |
| Near Right Viewpoint | Latitude | 53° 22' 54.91988" | 53° 21' 37.53950" |
| | Longitude | 6° 36' 6.00481" | 6° 31' 58.75332" |
| Far Left Viewpoint | Latitude | 53° 22' 56.17834" | 53° 21' 40.30189" |
| | Longitude | 6° 36' 0.88721" | 6° 31' 55.64078" |
| Far Right Viewpoint | Latitude | 53° 22' 54.44198" | 53° 21' 39.15439" |
| | Longitude | 6° 36' 0.51067" | 6° 31' 53.90148" |

Table 5.11. Parameters that define both the Spatial Video camcorder random test image Viewpoints.

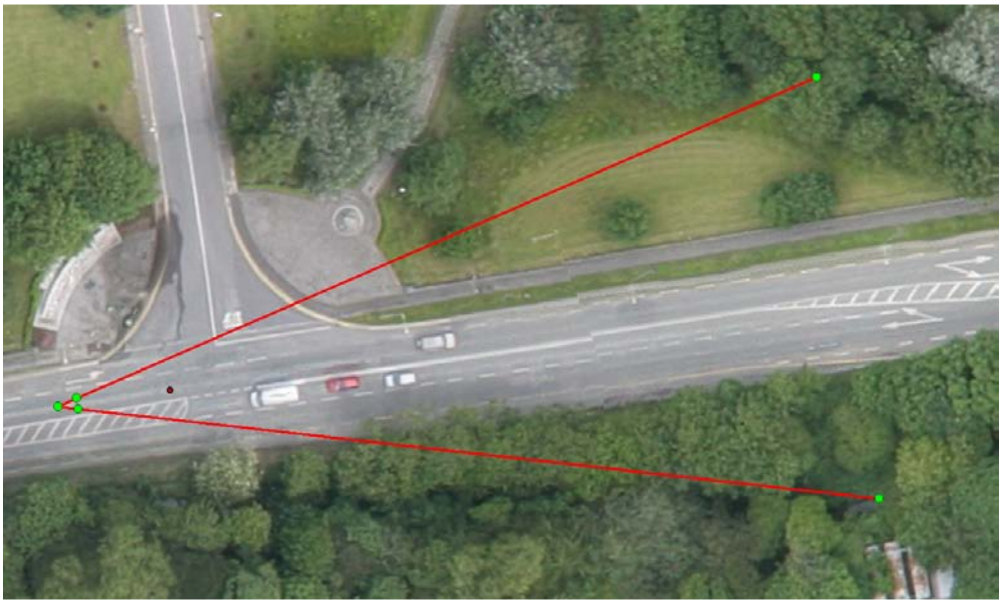


Fig 5.6 Calibrated Viewpoint for the Panasonic camcorder video frame image.

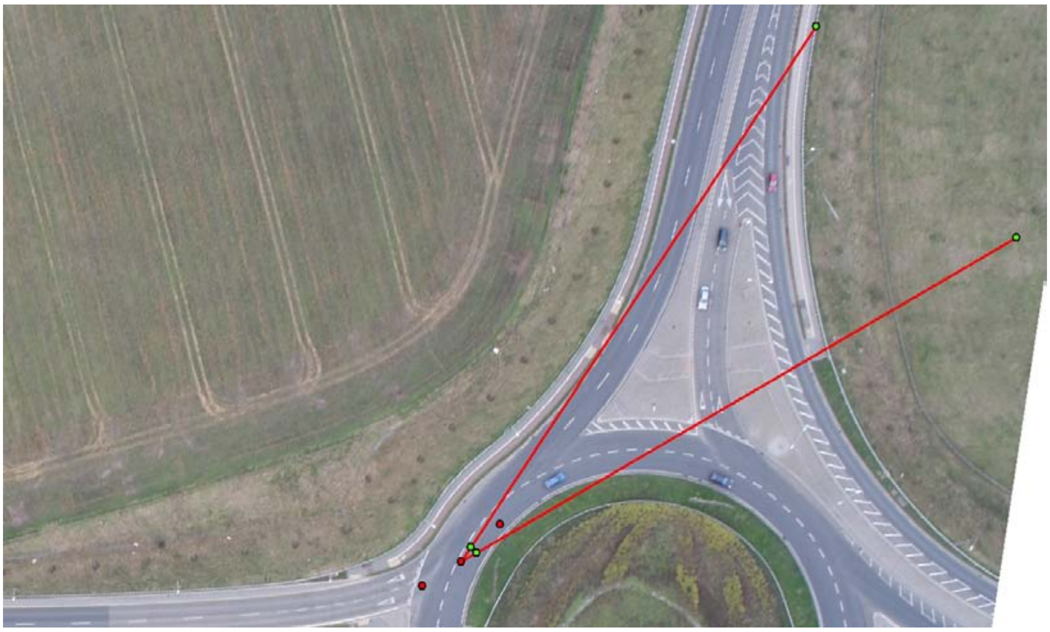


Fig 5.7. Calibrated Viewpoint for the JVC camcorder video frame image.

## 5.7 Conclusions

This chapter describes, based on two experiments, that a very simple and minimum set of camera and spatial parameters are required to approximate the geographical space as captured in a Spatial Video image’s object space. Of course this does not take into account occlusions that may be present in the captured image; some of which are handled through a GIS approach detailed in chapter seven. While the first experiment modelled the viewpoint correctly it was an approximation that was accurate to approximately half a meter. The second experiment, using the Spatial Video images, still required a supervised fitting of the Viewpoint to achieve a set of calculation parameters that could be used for the whole video stream. However, what these experiments did show is that this approach is valid when calculating Spatial Video Viewpoints. Nonetheless, this approach definitively supports Fishers (1999) assertions that a Viewsheds boundaries can only be estimated, as is the case from these experiments. While this point is true for the retrospective, inaccurate, nature of the Spatial Video used in this project it may not be the case for highly calibrated and tested systems where near to optimal Viewpoints should be theoretically possible.

Also, a number of assumptions were discussed which in practical terms add subjectivity to this modelling approach. This is further compounded by the inherent subjectivity in determining Viewpoint parameters for retrospective data sets based on visual comparisons. However, as this is a base case proof of concept implementation of the simplest form of the Viewpoint model, it can be easily surmised that more complex models of the Viewpoint theory are simply a case of defining adjustments to existing variables or achieving accurate parameter acquisition in future Spatial Video collection processes.

# Chapter Six: Viewpoints Database and Problems

This chapter describes the processing procedures, problems and solutions involved in populating an implementation of a Spatial Video Viewpoints database. The spatial database used is of a Database Managements Systems (DBMS) centred approach where the Viewpoints data structure, developed in earlier chapters, defines the geometry for a Spatial Video frame indexing system. Algorithmic solutions are discussed that attempt to deal with a number of spatial data problems and are introduced and developed in terms of defining accurate operational parameter data set representations of a Spatial Video survey.

## 6.1 Introduction

Building on the theoretical GIS Viewpoint framework introduced in chapter four and the test implementations of chapter five, a spatial database is developed to store these data structures which index the video frames. The basis for the type of video frame access index has been discussed in chapter two. A bespoke software component was developed to facilitate this implementation where a video and spatial data synchronised index could be defined along with the camera parameters that will determine the Viewpoints structures. This tool was developed as an automated post-survey processing tool and is partly based on software written at earlier stages in the research for video frame and audio spatial data capturing.

A number of problems with this process are discussed in relation to the levels of accuracy and inherent quality of the spatial data. These problems materialised as the viewpoint geometries were being viewed and analysed and basically highlighted the uncertainty in the accuracy of the Viewpoint structure and its geographical content. A number of solutions to these problems are discussed in relation to their implementation in an attempt to improve these issues. While, nominally successful, ultimately a hybrid approach to solving these problems is realised.

Fig 6.1 highlights an overall system architecture for the various elements introduced in this and previous chapters. It details each significant stage in the process from acquiring the Spatial Video through to preparing it for use in a GIS context; which is discussed in the next chapter.

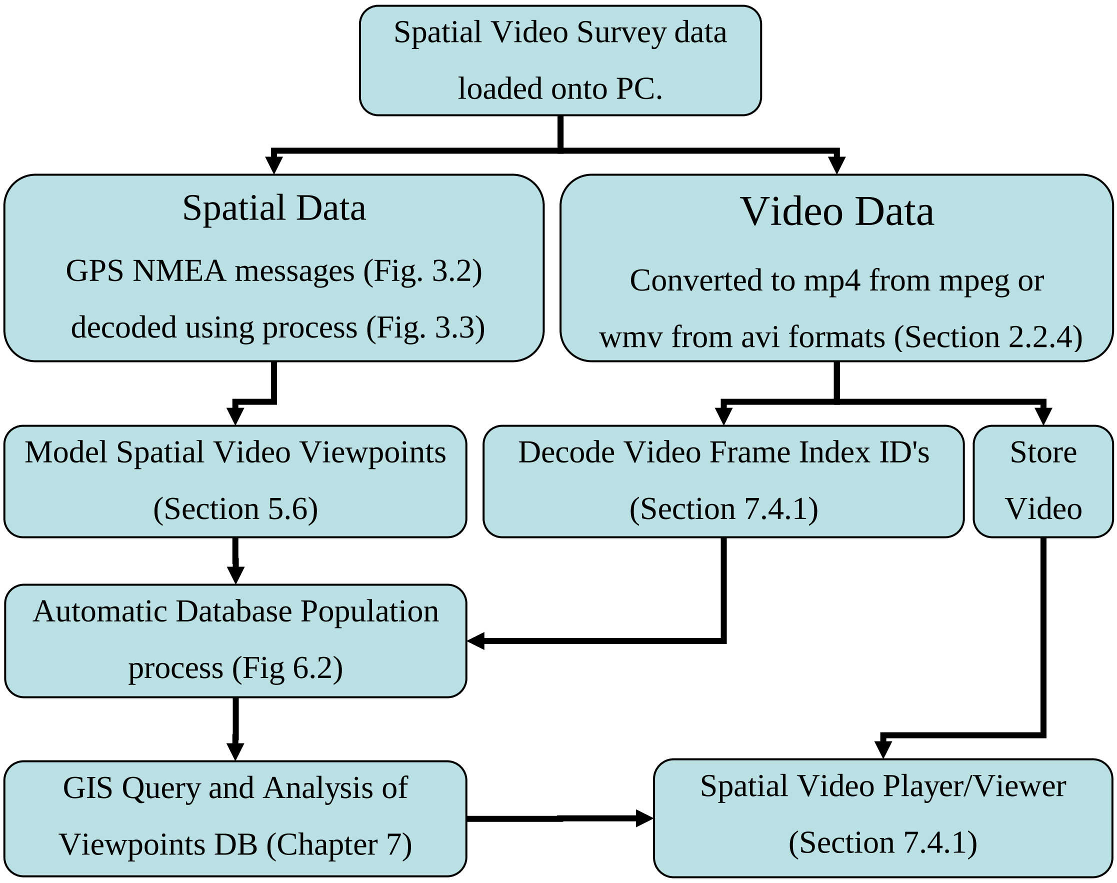

Spatial Video Survey data loaded onto PC.
Spatial Data
GPS NMEA messages (Fig. 3.2) decoded using process (Fig. 3.3)
Video Data
Converted to mp4 from mpeg or wmv from avi formats (Section 2.2.4)
Model Spatial Video Viewpoints (Section 5.6)
Decode Video Frame Index ID's (Section 7.4.1)
Store Video
Automatic Database Population process (Fig 6.2)
GIS Query and Analysis of Viewpoints DB (Chapter 7)
Spatial Video Player/Viewer (Section 7.4.1)

Fig. 6.1. Overall system architecture for all the distinct processes involved from the point of downloading the Spatial Video survey data to querying it in a GIS through the Viewpoints modelling approach.

### 6.2 Viewpoints Spatial Database

Discussed in this section is a PostGIS (2001) spatial database implementation of the Spatial Video Viewpoints model. PostGIS is just one of many possible spatial database systems that could have been used to complete these objectives; this one was chosen because of familiarity. Bespoke software has been developed to populate this database with Spatial Video Viewpoints based on input data sets and operating environment parameters. Chapter five detailed these parameters, defined here is the database, requirements for its population, problems with and solutions to accurate spatial representations and implementation/flexibility concerns.

### 6.2.1 GIS Database Support

PostGIS (2001) is an open source spatial database extension to the PostgreSQL object relational database. In Viqueira *et al.*, (2005) it is listed amongst the Data Base Management Systems (DBMS) centric approaches, as its spatial database development is defined through data management qualities with extensions for spatial data type and method support. An alternative approach is a GIS-centric methodology where spatial data and topological functionalities are extended with associated data management capabilities or spatial metadata properties. The former approach suited the objectives of this work as PostGIS has developed its spatial support through well typed data formats which have broad support across a number of GIS platforms. PostGIS in particular was used as its implementations are recognised by the Open Geospatial Consortium (OGC, 1994) for their full compliance with the Simple Features for SQL standards, (OGC Simple Features, 1999). This point is important as the ViewCone data structures defined in (Lewis, 2006) are part of the OGC Open Web Services, (OGC OWS-3, 2005), which are mentioned in chapter five and on which this modelling is built. Also, in Khuan *et al.*, (2007) the suitability of PostGIS for future modelling in 3D is shown through implementation of 2D surfaces. However, in OGC 3D, (2001) new abstract specifications have provided for the proposed development of future 3D features standards.

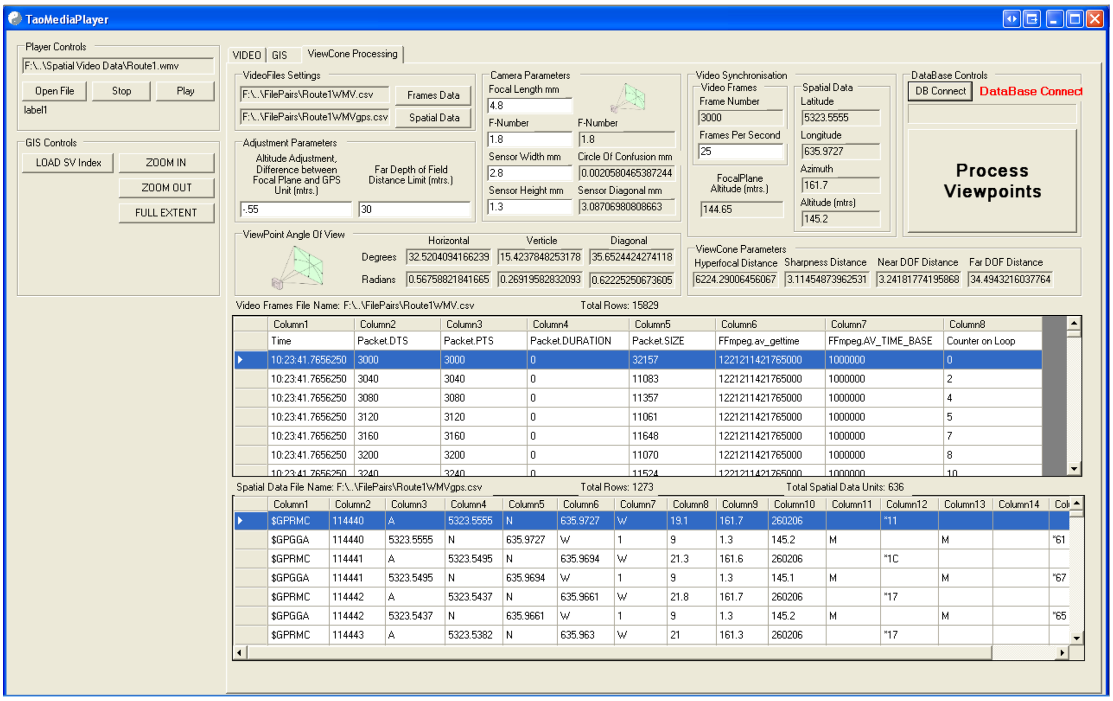
PostGIS also supports and implements numerous Geospatial analysis operations both natively and through the Open Source Geometry Engine (GEOS). Use of the Viewpoints data structure is demonstrated through these operational capabilities along with discussions of some necessary changes to redefine some Spatial Video semantic understandings of spatial operation functionalities. Some of these operational capabilities are summarised in table 6.1.

| **Class of Spatial Operations** | **Example Operations Used** |
|---|---|
| Management Functions | • AddGeometryColumn()<br>• UpdateGeometrySRID() |
| Geometry Constructors | • ST_PointFromText()<br>• ST_PolygonFromText() |
| Spatial Relationships and Measurements | • ST_Within()<br>• ST_Azimuth() |
| Geometry Processing Functions | • ST_Intersection()<br>• ST_SymDifference() |
| Linear Referencing | • ST_Line_Interpolate_Point() |

**Table 6.1. Sample list of PostGIS Geospatial operations as defined by an operation class and which were used in this chapter.**

### 6.2.2 Viewpoints Database Methodology

In designing a Spatial Video Viewpoints database a number of options were possible in the range of implementation levels for calculation of a dynamic spatial extent for each frame. The consequences of these options are discussed in more detail later; however, in this section only one Viewpoint calculation procedure is defined and implemented; that being the least dynamic approach where each Viewpoint is identical in its area. This required the Viewpoints database structure, record contents and populating procedures to be statically pre-defined. The processing procedures would then calculate all geometric spatial objects based on the default Spatial Video's system characteristics. In fig 6.2 a screenshot of the Viewpoints database calculation and processing software interface is shown.



F:\..\Spatial Video Data\Route1.wmv

F:\..\FilePairs\Route1WMV.csv

F:\..\FilePairs\Route1WMVgps.csv

Focal Length mm: 4.8
F-Number: 1.8
F-Number: 1.8
Sensor Width mm: 2.8
Circle Of Confusion mm: 0.0020580465387244
Sensor Height mm: 1.3
Sensor Diagonal mm: 3.08706980808663

Altitude Adjustment, Difference between Focal Plane and GPS Unit (mtrs.): -.55
Far Depth of Field Distance Limit (mtrs.): 30

|  | Horizontal | Verticle | Diagonal |
|---|---|---|---|
| Degrees | 32.5204094166239 | 15.4237848253178 | 35.6524424274118 |
| Radians | 0.56758821841665 | 0.26919582832093 | 0.62225250673605 |

Frame Number: 3000
Frames Per Second: 25
FocalPlane Altitude (mtrs.): 144.65
Latitude: 5323.5555
Longitude: 635.9727
Azimuth: 161.7
Altitude (mtrs): 145.2

| Hyperfocal Distance | Sharpness Distance | Near DOF Distance | Far DOF Distance |
|---|---|---|---|
| 6224.29006456067 | 3.11454873962531 | 3.24181774195868 | 34.4943216037764 |

Video Frames File Name: F:\..\FilePairs\Route1WMV.csv Total Rows: 15829

|  | Column1 | Column2 | Column3 | Column4 | Column5 | Column6 | Column7 | Column8 |
|---|---|---|---|---|---|---|---|---|
|  | Time | Packet.DTS | Packet.PTS | Packet.DURATION | Packet.SIZE | FFmpeg.av_gettime | FFmpeg.AV_TIME_BASE | Counter on Loop |
| ▶ | 10:23:41.7656250 | 3000 | 3000 | 0 | 32157 | 1221211421765000 | 1000000 | 0 |
|  | 10:23:41.7656250 | 3040 | 3040 | 0 | 11083 | 1221211421765000 | 1000000 | 2 |
|  | 10:23:41.7656250 | 3080 | 3080 | 0 | 11357 | 1221211421765000 | 1000000 | 4 |
|  | 10:23:41.7656250 | 3120 | 3120 | 0 | 11061 | 1221211421765000 | 1000000 | 5 |
|  | 10:23:41.7656250 | 3160 | 3160 | 0 | 11648 | 1221211421765000 | 1000000 | 7 |
|  | 10:23:41.7656250 | 3200 | 3200 | 0 | 11070 | 1221211421765000 | 1000000 | 8 |
|  | 10:23:41.7656250 | 3240 | 3240 | 0 | 11524 | 1221211421765000 | 1000000 | 10 |

Spatial Data File Name: F:\..\FilePairs\Route1WMVgps.csv Total Rows: 1273 Total Spatial Data Units: 636

|  | Column1 | Column2 | Column3 | Column4 | Column5 | Column6 | Column7 | Column8 | Column9 | Column10 | Column11 | Column12 | Column13 | Column14 | Col |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ▶ | $GPRMC | 114440 | A | 5323.5555 | N | 635.9727 | W | 19.1 | 161.7 | 260206 |  | *11 |  |  |  |
|  | $GPGGA | 114440 | 5323.5555 | N | 635.9727 | W | 1 | 9 | 1.3 | 145.2 | M |  | M |  | *61 |
|  | $GPRMC | 114441 | A | 5323.5495 | N | 635.9694 | W | 21.3 | 161.6 | 260206 |  | *1C |  |  |  |
|  | $GPGGA | 114441 | 5323.5495 | N | 635.9694 | W | 1 | 9 | 1.3 | 145.1 | M |  | M |  | *67 |
|  | $GPRMC | 114442 | A | 5323.5437 | N | 635.9661 | W | 21.8 | 161.7 | 260206 |  | *17 |  |  |  |
|  | $GPGGA | 114442 | 5323.5437 | N | 635.9661 | W | 1 | 9 | 1.3 | 145.2 | M |  | M |  | *65 |
|  | $GPRMC | 114443 | A | 5323.5382 | N | 635.963 | W | 21 | 161.3 | 260206 |  | *17 |  |  |  |

**Fig. 6.2. Viewpoints processing software where Spatial Video files, Camera parameter and synchronization utilities statically define the Viewpoints database geometries.**

This software component was developed to perform a post-processing procedure, used after the Spatial Video data has been captured, that populates a database with its Viewpoint geometries. It was implemented to an integrated design philosophy that incorporated previously developed software tools which include the audio decoder described in chapter three and a video frame analyser algorithm developed as part of a spatially controlled video player and the analysis work described in chapter two. In fig 6.3 a process flowchart defines the various data elements, processing procedures and software components that define the Spatial Video Viewpoints database steps.

In summary, this processing software required that a complete Spatial Video survey file be loaded, the camera recording equipment and any spatial adjustment parameters be set and a synchronisation be defined between the first video frame and spatial data point. The software would process the Spatial Video file by storing its video frame index and spatial reference data in a PostGIS database record. This was done for every frame index in the Spatial Video file. If a captured spatial data string was not available for a given video frame it was interpolated based on the last and next known GPS points and the video frame rate. The geometric Viewpoint was calculated as each frame and GPS synchronisation was completed and added to the spatial database record. More detail on each individual process is contained in the following sections.

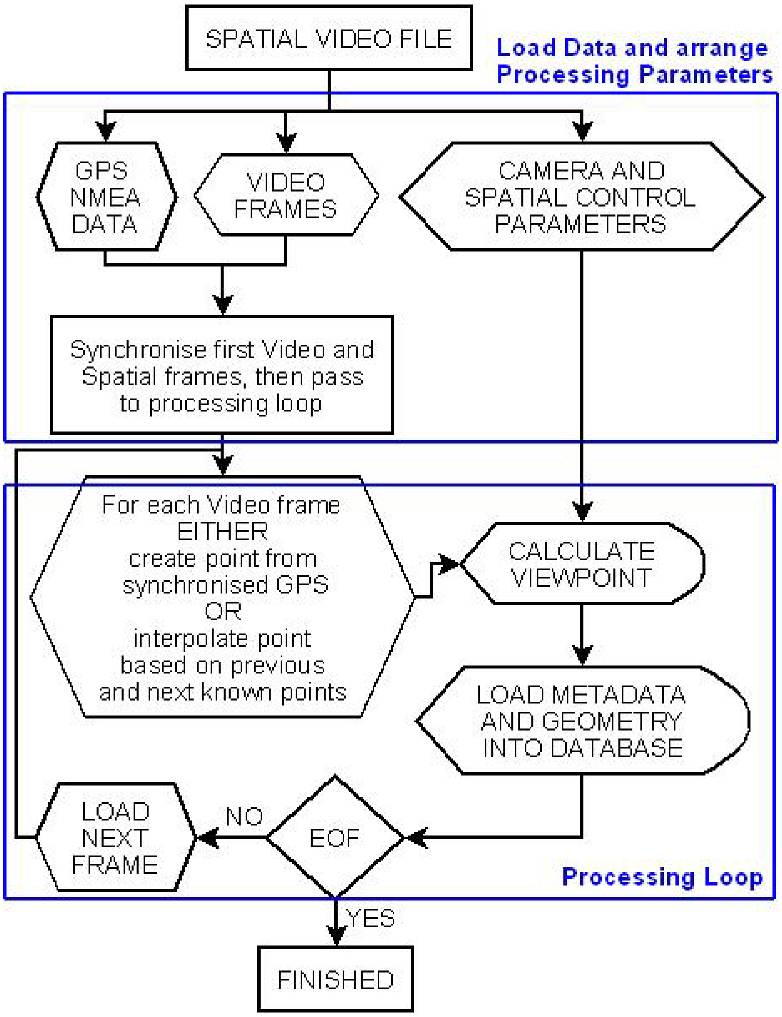

SPATIAL VIDEO FILE
Load Data and arrange Processing Parameters
GPS NMEA DATA
VIDEO FRAMES
CAMERA AND SPATIAL CONTROL PARAMETERS
Synchronise first Video and Spatial frames, then pass to processing loop
For each Video frame EITHER create point from synchronised GPS OR interpolate point based on previous and next known points
CALCULATE VIEWPOINT
LOAD METADATA AND GEOMETRY INTO DATABASE
LOAD NEXT FRAME
NO
EOF
YES
Processing Loop
FINISHED

Fig. 6.3. Process flowchart for Spatial Video Viewpoints database population.

### 6.2.2.1 Spatial Video Survey Data Preparation

This section briefly describes the Spatial Video data loading and parameter definition sections of the Viewpoints processing application. This involved methods to load the video's frame and GPS spatial variable indexes; also to set the camera operating parameters and any spatial or processing adjustment constraints. Included in this are two spatial adjustment parameters which define the GPS antenna's location in relation to the camera image plane, and the far depth-of-field distance used to calculate the Viewpoints ViewCone maximum extent.

Two possibilities exist for loading a Spatial Video file into the application and depend on the format in which the spatial data are stored. For a Spatial Video data set captured with the CamNav equipment, (NavTech, 1995), the audio decoder, as described in chapter three, could be used to process the spatial data from the audio channel. Alternatively, some Spatial Video data sets contain the GPS NMEA sentences in a separate file which could be loaded individually. Both methods loaded the GPS NMEA sentences into an editable data grid. In this format the GPS data could be manually manipulated to correct any problems or errors. A second data grid was loaded with the video frame index as well as a variable set to the video's frame rate.

The objective of these procedures is to provide a method that creates a starting frame-to-GPS point synchronisation. This was achieved by selecting a video frame index point and a GPS NMEA sentence from the respective data grids. Once set for the start point, the Viewpoints calculation procedures were further controlled based on the video frame rate. Fig 6.4 shows an outline visualisation of the setup objective for this procedure. The subsequent processing, based on a frame rate control, is known to be not optimum because of GPS accuracy errors. For example, a video capturing at 25 frames per second and a GPS recording location at one point per second cannot be relied on that every 25th video frame will be spatially coincident with all subsequent GPS points, after the initial synchronisation. A more sophisticated approach would need to be implemented as each video sequence captured between GPS points may bear no close relationship to the frame rate. A solution to this problem has not been defined in this study; however, while spatial frame rate synchronisation still presents a significant problem, in practice it is manageable as the GPS coverage was of a high enough standard to facilitate a systematic offset between spatial and video indexes, resulting in a close to optimal relationship for such a system to work accurately.

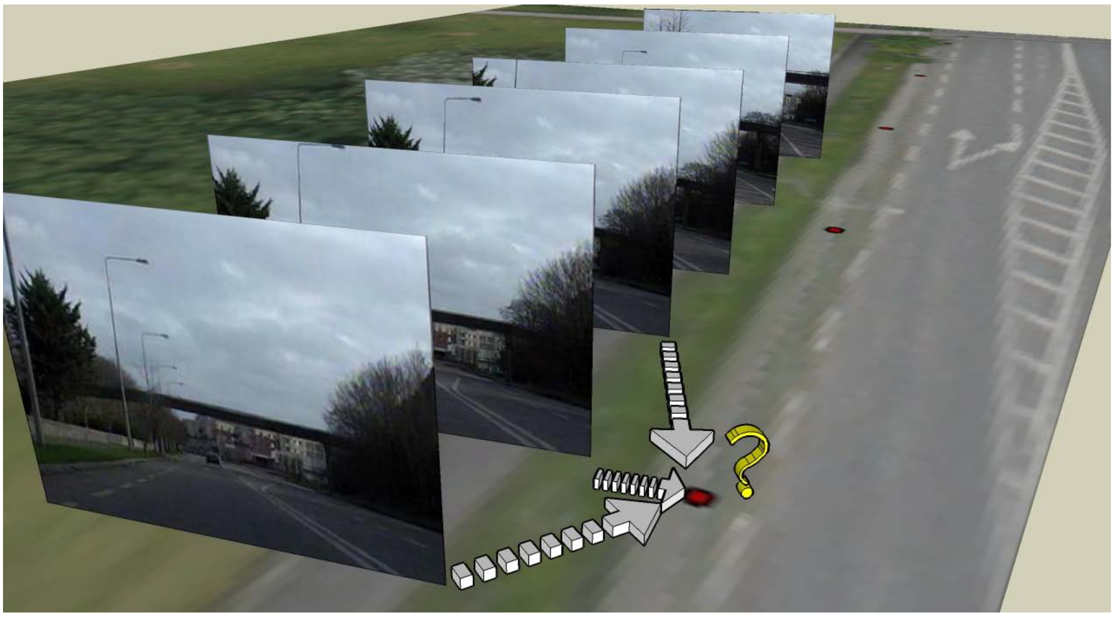


Fig. 6.4. Outline visualisation example of video frame to GPS point synchronisation objectives.

Video capturing equipment specifications and operational parameters were also defined at this stage. The camera's Charge-Coupled Device (CCD) width and height dimensions are set in millimetres along with the focal length and aperture F-number. Fig 6.5 shows a flowchart of how these parameters are used in defining the video camera's operational model. Chapter five defines all these parameters and their respective formulas; it also details the steps involved in empirically defining them.

A significant problem in this implementation methodology is the static nature of these parameter sets. They are defined through manual empirical testing and measurement, but only on a random sample set of video frames taken from each Spatial Video data stream. This approach has been shown, in chapter five, to provide an accurate overall representation of the object space of a video frame as defined through a Viewpoint. However, this method is only as accurate as the Spatial Video survey constraints and the amount of random video sampling in a manual calibration. In the survey data sets captured for this study the survey constraints included fixed focus calibrations which determined these parameters to be consistent across all video frames. Surveys that alter, either manually or automatically, the focus or zoom of the camera lens will require dynamic adjustments to be applied to the camera model and Viewpoint calculations as the algorithms proceed. Also, some automated system for detecting these video frame changes would need to be implemented as they are likely to have a very high transition rate.

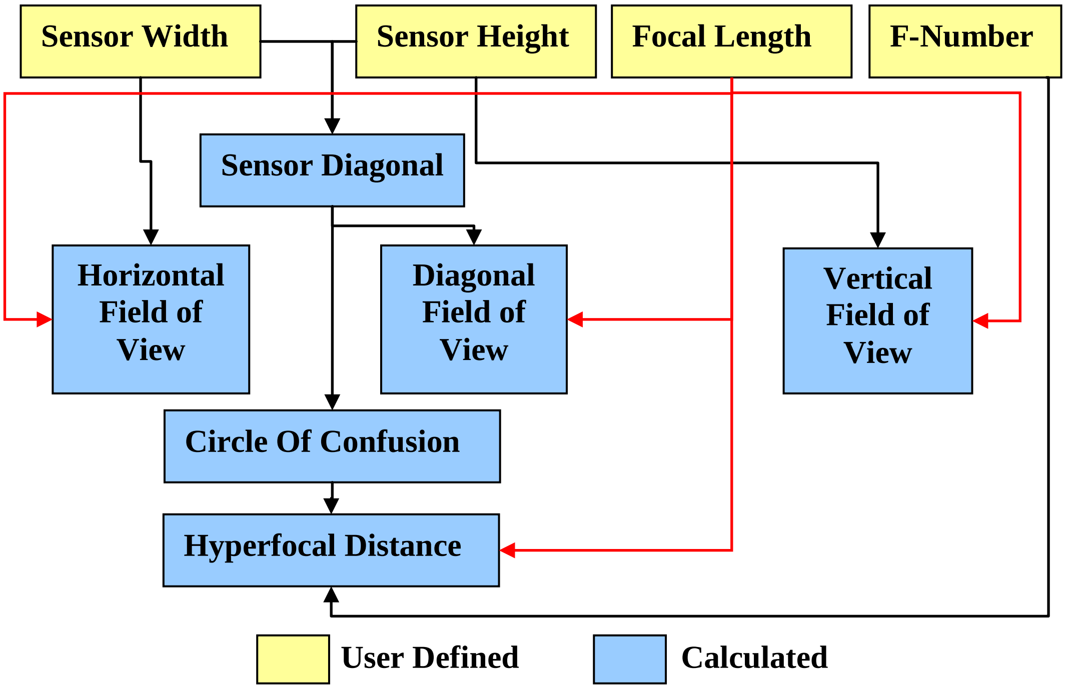


Fig. 6.5. Camera model parameter calculation flowchart. Yellow shaded parameters are user defined based on empirical testing and specification documents, while all blue parameters are derived from this base set.

Two parameters for spatial adjustment are also defined at this stage. Firstly, as was introduced in chapter four and implemented in chapter five, a far focus depth of field distance for all Viewpoint ViewCone calculations is defined. This parameter is user-defined and simply extends the geographical space representation of the ViewCone to a defined distance beyond the Hyperfocal sharpness. Equation 5.5 is used to calculate this parameter and its implementation is discussed in sub-section 5.3.2.1. Once again, this parameter could be considered in a more dynamic context – empirical testing of spatial operations has shown that a longer distance is more appropriate where Spatial Video footage captures wide open space. A shorter setting is more practical in a confined survey setting such as an urban environment or where road boundaries have high elevations. This happens where ViewCone boundaries intersect occlusions but also overlap subsequent ViewCone coverages to such an extent that a large portion of the same space is captured on different frames. Fig 6.6 shows an example of this.

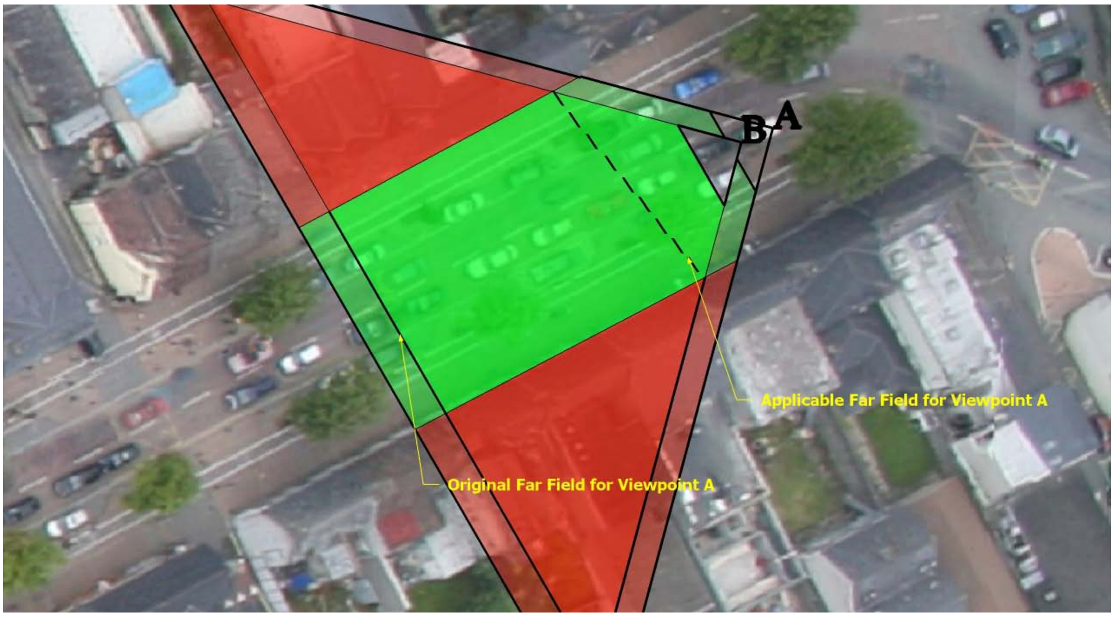

B
A
Applicable Far Field for Viewpoint A
Original Far Field for Viewpoint A

Fig. 6.6. This urban Viewpoint overlay shows how a far field distance can be affected in its coverage extent by occlusions. In this case, a line of buildings (streetscape) on the survey route can exclude any gain of a wide angle FOV and longer far focus field extent.

Secondly, the initial intention was for a spatial adjustment parameter set to be input that would define the GPS antenna offset in relation to the camera image plane as a survey calibration distance and azimuth adjustment parameter pair. This required an initial known antenna location to be defined in terms of the offset distance from the GPS unit to the camera focal plane and an azimuth adjustment to align it with the camera’s orientation. This is because the subsequent Viewpoint calculation represents the location of the captured image and the geographical space it encompasses from the camera and not the spatial location. However, it was realised that this parameter set is only useful in a situation where very high accuracy GPS can be collected and that the adjustment is reliably systematic across the whole video stream. This is not the case for the survey data sets; thus, this parameter set became more useful in correcting for GPS error as this could be incorporated and corrected for in the same processing procedure using the algorithm presented in appendix one. Unfortunately, it was a static implementation that assumed this error adjustment ranged across the whole video stream. A dynamic version is required to implement this concept properly; but defining an automated process would be difficult and poses a significant problem.

An example of the GPS error present in the survey area's Spatial Video data is shown in fig 5.5 of chapter five. This was a simple manual adjustment in the context of that Viewpoint test case; the parameter effectively moved the GPS track orthogonally to its direction of travel to coincide with the video camera's focal point. The more accurate dynamic system would provide a better representation which would allow forward, backward and offset angle adjustments of GPS points to achieve the appropriate video frame-to-spatial capture point accuracy. This highlights the inherent uncertainty that exists when dealing with GPS error and its spatial synchronisation to a video frame. This is highlighted in an exaggerated fashion in fig 6.7.

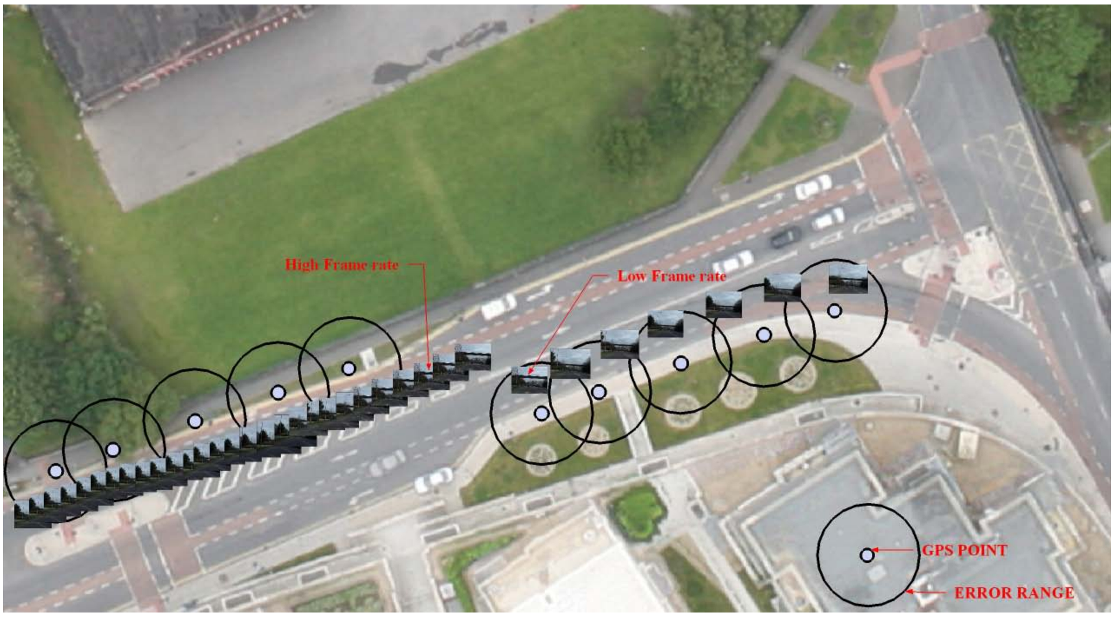


Fig. 6.7. Exaggerated example of the spatial and video-frame data synchronization problem. In this example each GPS point needs to be aligned to a video frame based on a geometric adjustment. For the higher frame rate stream (left) a simpler GPS orthogonal adjustment is possible as more frames fall into the point error range. The lower frame rate stream (right) requires a greater degree of error adjustment as fewer video frames can be coincidently aligned to.

Ultimately, the degrees of freedom in correctly aligning the spatial and video data will partly be a function of the video frame rate and the GPS error range and quality. In a low frame rate scenario, the spatial variability would be higher and thus require greater geometric degrees of freedom to define a precise synchronisation. A high frame rate video stream could satisfy the system as implemented in this software, as it may require only a simple orthogonal adjustment to the closest video frame. This could be achieved by a line drawn through the video frame's actual spatial sequence where each GPS point is snapped to it to coincide with any video frame, regardless of frame rate position.

### 6.2.2.2 Consecutive Viewpoint Spatial Problems

As shown in the flowchart of fig 6.3, the Viewpoint calculations are performed for every Spatial Video frame in a processing loop. This begins with the video-frame and GPS-point that have been synchronised through the procedure described in the previous section. The Viewpoint calculation is performed based on the methodologies described in chapter five, section 5.4.2. Discussed in this section are a number of methods used to handle the following list of spatial data problems:

1. Interpolation of extra GPS points is necessary to define the locations of all frames captured between the spatial data intervals; this is based on the differences in video and spatial data capture frequencies.
2. Smooth the GPS track data to better represent the real world survey vehicle route more accurately.
3. Average and adjust for GPS drift when the survey vehicle was stationary, is shown in fig 6.8.

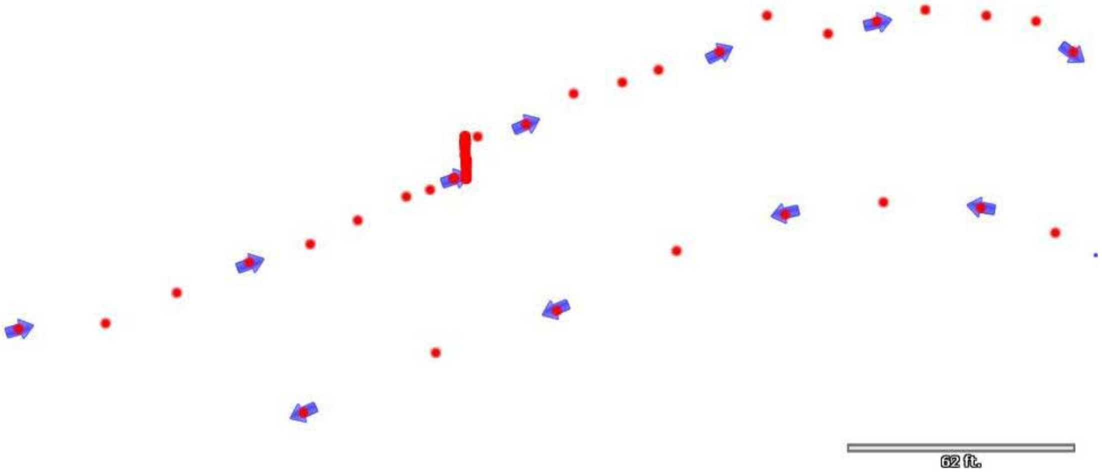


Fig. 6.8. Original GPS track from one of the Spatial Video urban routes. This highlights GPS drift where the vehicle was stationary.

A number of attempts at dealing with these problems were made with each producing unsatisfactory results. Three incremental sets of implementations were developed with each one resulting in an improvement on the previous. However, they did not provide a comprehensive solution to the calculation and deployment of a smooth Viewpoints Spatial Video stream. The first method implemented a simple linear GPS track with equal intervals interpolation based on the video's frame rate. The second method implemented a bespoke azimuth smoothing also with equal intervals interpolation. The third method used third party software, (TopoFusion, 2002), to generate a post-survey and pre-Viewpoint processing spatial data file. This software implements a piecewise cubic Bessel interpolating spline. The first two methods were implemented as part of the Viewpoint processing algorithm but were removed when the pre-processed GPS track spline method was employed. This only required the processing algorithm to simply read the spatial data from the pre-prepared GPS file, except in cases of GPS drift.

Depending on the GPS data rate, (one hertz for CamNav data sets), and the video frame rate, (25 hertz for Panasonic and JVC camcorders), it is clear that many more video frames will exist than will spatial data points. Ideally, a systematic relationship should exist based on this knowledge, i.e. for every 25th video frame a GPS point in the spatial data stream can be assumed to be coincident with it. Thus, an equal distance calculation can be implemented to divide the distance between two subsequent GPS points based on the video frame rate. It is this methodology that was assumed when the two bespoke viewpoint processing algorithms were implemented. As mentioned in the previous section; it is not ideal to make this assumption because of GPS error ranges, the levels of complexity in the alignment procedures and video frame rates. However, the methods that were implemented for interpolating a spatial point to represent each inter-GPS point video frame performed adequately for the requirements.

The first methodology implemented was a simple linear track where interpolation was calculated based on an equal division of the distance between any two consecutive GPS points and the video frame rate. The algorithm was implemented as a planar Cartesian calculation based on two known GPS coordinates in an X and Y axial plane.

This distance was divided by the video frame rate to determine the interval distances between each video frame. The first known GPS point was then used to extrapolate the next video frame location by adding the subdivision distance. An example is performed in the following steps, while the result is shown in fig 6.9.

1. Known Points calculations, (all performed in Radians):
    a. Latitude Formula: (Latitude1 – Latitude2) / (Frames Per Second – 1)
    b. Example: (53.382 – 53.381952) / 24 = 0.000002
    c. Longitude Formula: (Longitude1 – Longitude2)/(Frames Per Second-1)
    d. Example: (-6.582525 - -6.582505) / 24 = -0.00000083
2. Successively interpolate points by incrementing, using the graduated distance:
    a. Known latitude: 53.382 + 0.000002 = 53.381998
    b. Known longitude: -6.582525 + -0.00000083 = -6.582524167

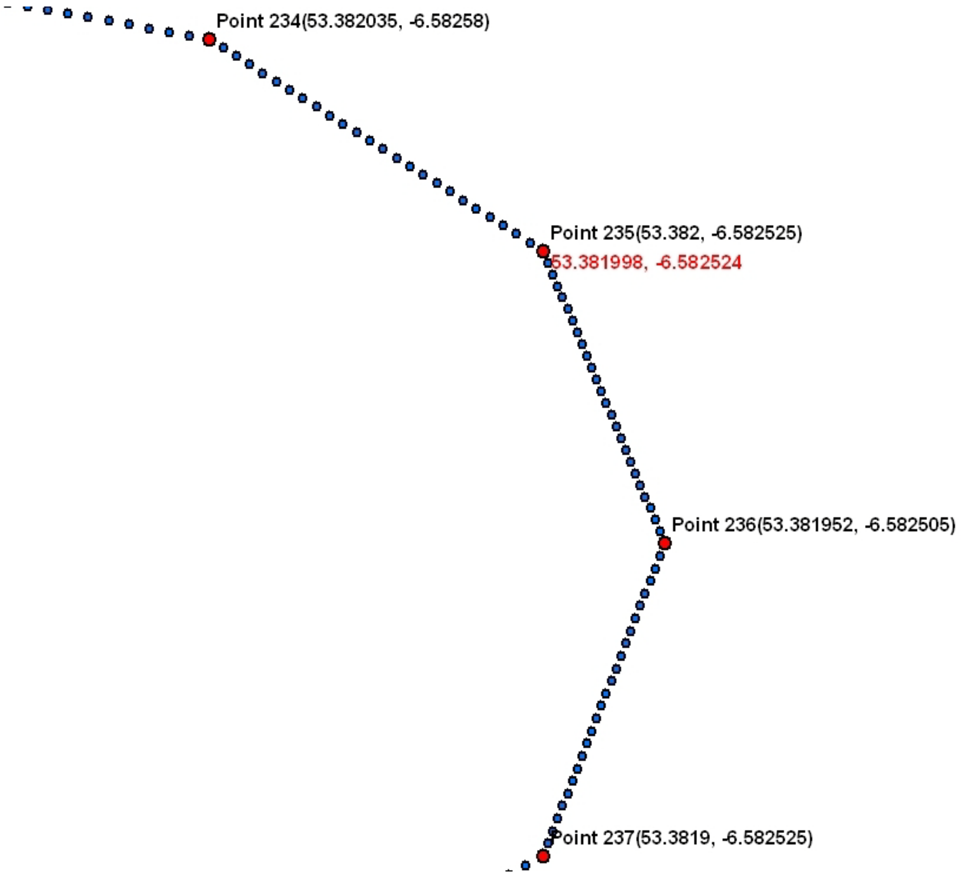

Point 234(53.382035, -6.58258)
Point 235(53.382, -6.582525)
53.381998, -6.582524
Point 236(53.381952, -6.582505)
Point 237(53.3819, -6.582525)

Fig. 6.9. Linear interpolation applied to a segment of GPS track data.

It is clear from fig 6.9 that linear interpolation has resulted in smooth distance interpolation between points; however the transition to the next segment is not a smooth representation of a real piecewise road survey track. Thus, in the second method a bespoke implementation of a geodetic azimuth algorithm was developed to smooth these transitions. To determine the interpolation a similar methodology as before was used where a distance was calculated and used to generate the successively graduated GPS points. Instead of Cartesian calculations, Vincenty's (1975), inverse geodetic formula was applied, based on Gavaghan's (2007), implementation, which calculates an ellipsoidal distance between GPS points. This algorithm is shown in appendix two. Also, to achieve a better smoothing between segments, the difference between the azimuth measured at the first point and that at the second point was calculated using an adapted version of Vincenty's formula, based on Gavaghan's implementation, (Gavaghan, 2007). This algorithm is shown in appendix three.

The interpolation distance intervals were calculated based on the video's frames per second rate. The azimuth angle adjustment interval was based on either the video's frames per second rate or a user-defined factor. The latter approach was tested and implemented using half the azimuth at each step and resulted in a shallow smoothing as opposed to the frame rate which resulted in a larger smoothing gradient. The procedure is as follows, with the result shown in fig 6.10.:

1. Calculate distance intervals between known GPS points 235 and 236 using appendix two's algorithm. Formula: Distance_Calculate(Frames per Second).
    a. Result = 0.220216 meters per interval.

2. Calculate normalised difference between the azimuths determined from GPS point 234 to 235 and from GPS point 235 to 236, divided by video frames per second.
    a. Using appendix four's function to normalise the azimuths calculated using appendix three's algorithm and define the interval.
    b. Result: (29.22590936° / 25) = 1.169036374°

3. Interpolate intermediate GPS points using appendix one's algorithm where each subsequent point is extrapolated based on the previous points input, the distance interval and the adjusted forward azimuth. All calculations were performed in radians.
    a. Result Latitude = 53.3819985305
    b. Result Longitude = -6.582522783

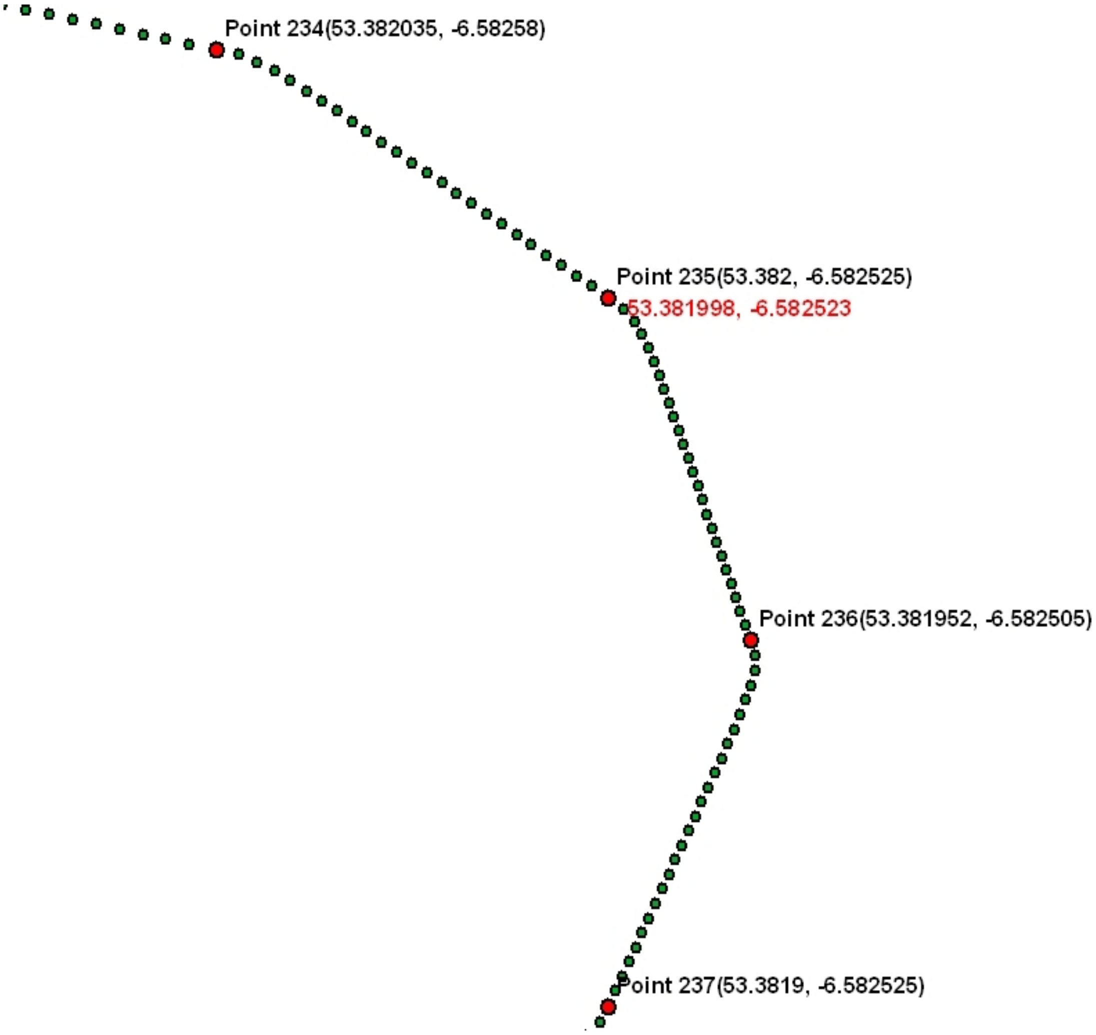

Point 234(53.382035, -6.58258)
Point 235(53.382, -6.582525)
53.381998, -6.582523
Point 236(53.381952, -6.582505)
Point 237(53.3819, -6.582525)

Fig. 6.10 Geodetic FPS interpolation applied to a segment of GPS track data.

This methodology did smooth the transition between segments, however it failed to provide a continuous piecewise smooth curve that could be considered a reasonable representation of the road survey route. While the transition was smoother at the beginning it quickly tended towards a linear finishing sequence. To solve this problem a spline approach was investigated where a number of approaches were considered and investigated which include, (Douglas *et al.*, 1973; Jupp, 1978; Maolin *et al.*, 2009; Sun *et al.*, 2009). However, these GPS smoothing algorithms deal with the spatial data only, whereas in the context of this project the objective is to model the video. This means that the GPS track should become a smooth representation of the visual environment captured in the video thus an approach with the objective of preserving the original GPS data points was implemented. This is because of the spatial adjustments that have already been considered in the previous sections operations for dealing with GPS uncertainty. One possibility that was considered was used by McLoughlin *et al.*, (2008) for piecewise road network smoothing. It was based on an approximating spline using the smoothing functions defined by Reinsch (1967). Here two criteria are minimised using weighting factors to reduce the slope of the spline and to ensure the curve came reasonably close to the original GPS points.

Alternatively, another approach is based on an interpolating spline algorithm as is implemented in the TopoFusion software, (TopoFusion, 2002). A brief description of this approach is taken directly from the software's specifications documents where it defines that an interpolating spline differs from an approximating one in that the original data points are preserved. This is achieved using a piecewise cubic Bessel interpolation that ensures the continuity of slope by employing Hermite boundary conditions. The Bessel functions first derivative is estimated by fitting a parabola through three consecutive GPS points. TopoFusion implements this methodology specifically for GPS track smoothing, but also includes a facility for user control of the number of interpolating points. This software was used to interpolate the GPS routes in a pre-Viewpoint processing procedure using the relevant video frame rate to generate the extra frame control GPS points. Fig 6.11 shows a comparison between the linear, geodetic half difference, geodetic FPS difference and TopoFusion interpolation.

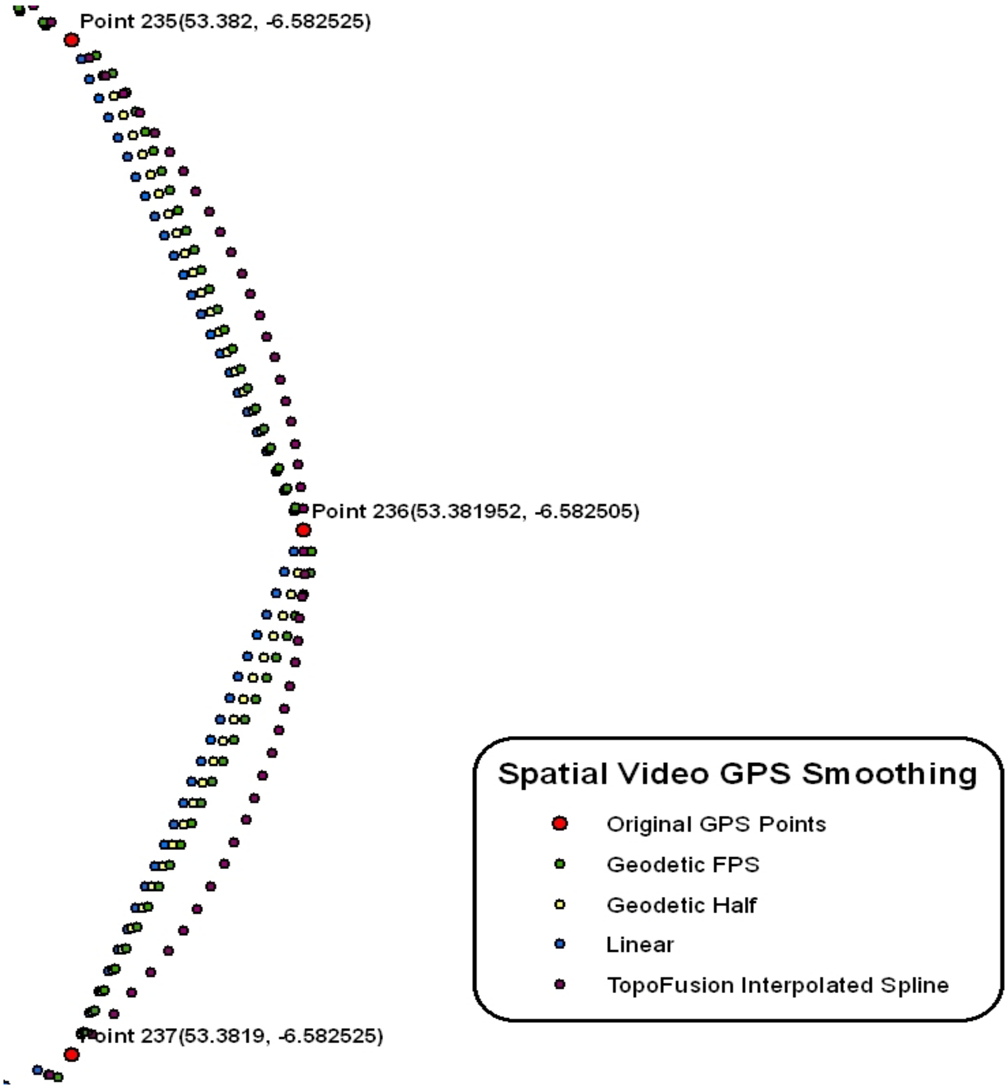


Fig. 6.11. Comparison between bespoke GPS track smoothing implementations and the TopoFusion interpolated spline.

Clearly a smoother track has been generated by the interpolated spline; however a number of issues still pose some problems. While these three methodologies provide a solution to the video frame interpolation issue, it is only the spline method that solves the problem for a reasonably accurate spatial location estimate for each video frame. This also leads to the conclusion that it defines the most accurate piecewise representation of the original road network. Unfortunately these conclusions are negated by the bespoke nature of the original methods which were also designed to account for the GPS drift problem. The proprietary nature of the spline software did now allow for this situation to be handled and as is clear from fig 6.12 the bespoke methods significantly improved the track smoothness for GPS drift.

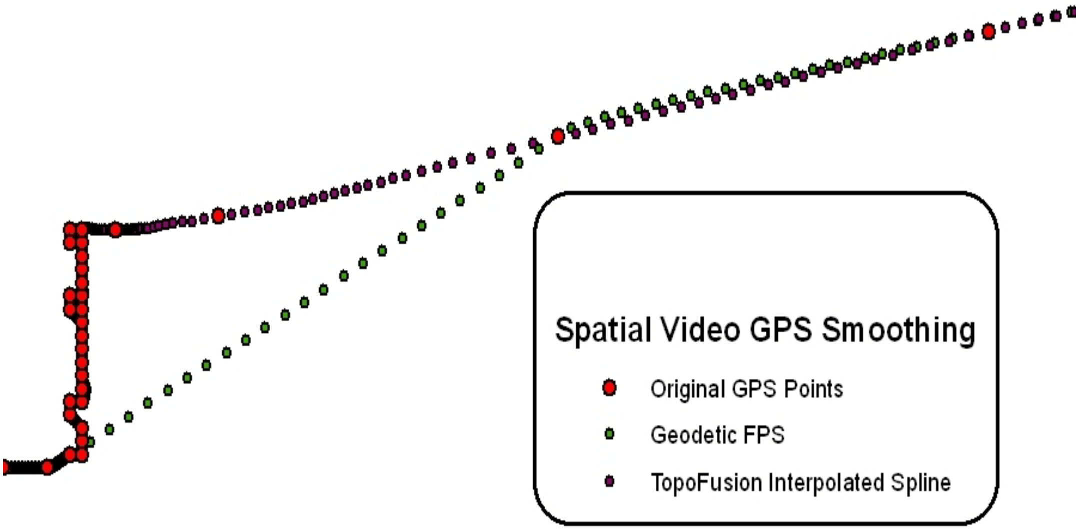


Fig. 6.12. Visualisation of GPS drift handling in the geodetic smoothing compared to the spline.

It is realised that a bespoke implementation of the spline algorithm could have this special case condition incorporated; however the final solution incorporated both the geodetic and spline approach. The spatial data set was firstly spline processed followed by a geodetic interpolation where GPS drift was detected. This involved measures of GPS based velocity and azimuth change. A gradual reduction in GPS velocity, below 5 knots, that tends towards to zero, (zero is not likely to be achieved as the GPS drift will contain a velocity measure), flagged the possibility of drift being introduced. This flag then resulted in a track history to be recorded on the average azimuth over the successive points. A check on the extent of change in the average azimuth of preceding points and the next consecutive GPS point's azimuth caused the algorithm to redefine the track. This was achieved by relocating all consecutive points to coincide with the last point before the rapid change in azimuth. This process continued until the preceding azimuth history value was re established, within a tolerance range, by a consecutive point azimuth calculation. This solution was defined and shown to work for the tracks collected for this study; however it has not been tested on a broad range of GPS track smoothing data sets. Ultimately, this procedure should not be required as most modern GPS units now contain SiRF controllers which allow static navigation settings that eliminate drift. Garmin have introduced this technology into their receivers since 2005.

### 6.2.3 Database Tables and Records

A PostGIS spatial database, as mentioned in the previous sections, has been populated with Spatial Video Viewpoint geometries and survey metadata parameters. This database contains a single table where the survey's technical metadata consists of the full set of parameters used in the processing procedures. It also contains two geometry data type fields to hold the Viewpoint geometries; one contains a point data type that stores the spatial location for the associated frame while the other contains a polygon data type that stores the Viewpoint ViewCone. The Viewpoint static geometries construction parameters are stored for reference but also for further processing based on any dynamic improvements that can be applied to the system. In a later discussion, various theoretical improvements to some of the problems and practical issues will show why it would be required that various parameters be redefined in the database. For now it is enough to point out that the initial record structure defines an exact matching of the metadata parameters and the geometries they define.

From the survey data sets used in this study, approximately one hour and fifty six minutes of footage has been processed into the Spatial Video database. This is comprised of 70,296 Viewpoint records where each record defines the geographical extent of a video frame and the variables that were used in the calculations. In table 6.2 the database table structure schema is defined along with a sample Viewpoints record. The Polygon_geom field contains five geodetic latitudes and longitudes that define a five-point ViewCone with the first and last points closing the polygon. A non-intersecting polygon configuration is paramount in the automated database population process as a geometry integrity checking system is built into the PostGIS table configuration.

Table 6.2 defines the Viewpoints database which is called *svindex* when used in chapter sevens SQL operations. Table 6.3 formalises this description and that of all the other spatial-data tables that have been set-up for use in chapter seven.

| **Field** | **DataType** | **Sample Data** |
|---|---|---|
| File_Source | Text | *Directory_String*\Route2.wmv |
| Frames_Per_Second | Numeric | 25 |
| Frame_Number | Numeric | 1579 |
| Focal_Length | Numeric | 4.5 |
| Aperture | Numeric | 1.8 |
| Sensor_Width | Numeric | 2.4 |
| Sensor_Height | Numeric | 1.8 |
| Far_DOF_Limit | Numeric | 50 |
| Focal_Plane_GPS | Text | 0.55,22 |
| Date | Date | 260206 |
| UTC_Time | Time | 111558 |
| HDOP | Numeric | 2.1 |
| Altitude | Numeric | 141 |
| Geoid_Height | Numeric | 0 |
| Speed_Knots | Numeric | 21.4 |
| Azimuth | Numeric | 79.7 |
| Point_geom | Geometry | POINT(-6.60422666666667 53.3816633333333) |
| Polygon_geom | Geometry | POLYGON((<br>-6.60418706586609 53.3816744911111,<br>-6.60418303217517 53.3816612198078,<br>-6.60371793498222 53.3816386908733,<br>-6.60376496199081 53.3817934203863,<br>-6.60418706586609 53.3816744911111)) |

Table 6.2. Spatial Video Viewpoints database table schema and one sample populating record.

| **Database Table Name** | **Field Name** | **Field Type** |
|---|---|---|
| *svindex* | sv_id | serial NOT NULL |
| | file_source | character varying(254) |
| | frames_per_second | bigint |
| | frame_number | text |
| | focal_length | numeric |
| | aperature | numeric |
| | sensor_width | numeric |
| | sensor_height | numeric |
| | far_dof_limit | numeric |
| | focal_plane_gps | text |
| | date | bigint |
| | utc_time | bigint |
| | hdop | numeric |
| | altitude | numeric |
| | geoid_height | numeric |
| | speed_knots | numeric |
| | azimuth | numeric |
| | point_geom | geometry (point type) |
| | polygon_geom | geometry (polygon type) |
| | | |
| *PointOfInterest* | gid | integer NOT NULL |
| | description | character varying (80) |
| | the_geom | geometry (point type) |
| | | |
| *PointOfView* | gid | integer NOT NULL |
| | location | character varying (80) |
| | the_geom | geometry (point type) |

Table 6.3. Continued on next page.

| Database Table Name | Field Name | Field Type |
|---|---|---|
| *smallareas* | ogc_fid | serial NOT NULL |
| | wkb_geometry | geometry (polygon type) |
| | sa_code | character (80) |
| | land_use | integer NOT NULL DEFAULT 1 |
| | area_name | text |
| | land_use_descritpion | text |
| | | |
| *LineOfView* | gid | integer NOT NULL |
| | route_description | character varying (80) |
| | the_geom | geometry (line type) |
| | | |
| *model* | gid | integer NOT NULL |
| | building | character varying (80) |
| | the_geom | geometry (polygon type) |

Table 6.3. Database schematic for all the spatial data tables built for this chapter and used in chapter seven.

### 6.3 Conclusions

The content presented in this chapter has predominantly detailed the various problems that had to be overcome in order that a Viewpoints database could be implemented. The approaches taken to building this system were to define an automated Viewpoints database population procedure where a simple video frame to spatial location index could be defined. However, the variability in GPS quality, contained in the Spatial Video survey data, presented quite a number of indexing problems. While this chapter details these problems and provides implemented solutions, it is realised that a hybrid approach involving a number of GIS and Computer Vision techniques would need to be developed to give a robust solution. This is because the source of visual information contained in the video is ignored, which could be used to help provide spatial clues towards more accurate track estimations.

This is because of the nature of research into the GPS/track smoothing where spatial approaches predominate. As highlighted in these implementations, a purely spatial data improvement process was pursued before applying these results to the video indexing. The initial implementations and set-up decisions were handled statically where systematic relationships are assumed to persist throughout the process. However, this is not true based on the testing in the development of this system and on the operations work performed in the chapter seven, which is also based on this Viewpoints database. However, this realisation has not negated these approaches as none of the GPS problem handling methodologies produced unreasonable results. While not highly accurate on any individual video frame to spatial location level, the operational functionalities of the Viewpoints database remained viable.

In this study a hybrid spatial solution to GPS data smoothing was implemented and should be considered viable in a video context where the image frame rates are high enough to allow spatial alignment. This is regardless of the actual relationships between spatial and video data collection frequencies. Ultimately though, a single solution to the GPS to video frame alignment procedures and to the GPS track smoothing should involve an element of image analysis that can automatically determine critical changes in spatial situations that will affect the video frame indexing procedures or be augmented with another spatial data source such as a compass. Either method can help validate the associated GPS track data and provide the required corrections to reflect the level of smoothing that is necessary, i.e. is it a GPS error or a real movement of the survey vehicle.

While a lot of work has been performed in the area of road detection using image analysis of aerial imagery, in some cases augmented with geographical information, (Auclair Fortier *et al.*, 2000), no work appears to have been undertaken using oblique terrestrial imagery to define improved GPS track representations. Morris (2002) uses a edge detection technique called the snake algorithm to determine mountain tracks from aerial imagery. These are then used to smooth a GPS track by snapping it to the closest detected image edge. While this approach could be used to smooth the Spatial Video GPS to aerial imagery road networks, it does not consider the oblique imagery content of the video. This content may contain visual evidence of route obstacles that caused a GPS track to deviate legitimately.

Thus the hybrid approach could possibly involve a combination of systems where both image analysis and GIS controls are defined. A GPS track could easily be spatially aligned to existing sources of road network data; however this does not guarantee accuracy as the video footage reveals many examples of where normally determined spatial irregularities are actually accurate spatial representations of the survey path. This is based on many example situations involving video footage that shows the survey vehicle having to alter its course, to varying degrees, to avoid everyday road traversal obstacles. These could include road works, cyclists or other stationary vehicles temporarily parked. Thus, a proposed system could analyse aerial imagery and the video's terrestrial imagery for reference objects or changes in spatial alignment where spatial distortion is detected in the GPS track. In some cases what may be present as a multipath error may be a genuine avoidance of some obstacle by the survey vehicle.

The analysis procedures that determined these problems were visual in that the resulting database of Viewpoints produced a clearly incorrect spatial representation of the video track object space. Fig 6.13 gives an example of every tenth Viewpoint calculated for a Spatial Video stream in the survey area. Clearly two distinct situations can be concluded based on a visual analysis of this sequence. Firstly, where continuity in the GPS track is stable, a satisfactory linear ViewCone alignment can be seen. Secondly, four ViewCones (one is slightly occluded) are calculated that bear no realistic resemblance to the video camera's orientation as this survey was defined as forward orientated and coincident with the azimuth direction. The root of these ViewCones is a set of points that have resulted from a GPS drift problem. The ViewCones where calculated based on the incorrectly assumed GPS azimuth orientation being consistent, however the sharp change in azimuth was not initially detected and corrected.

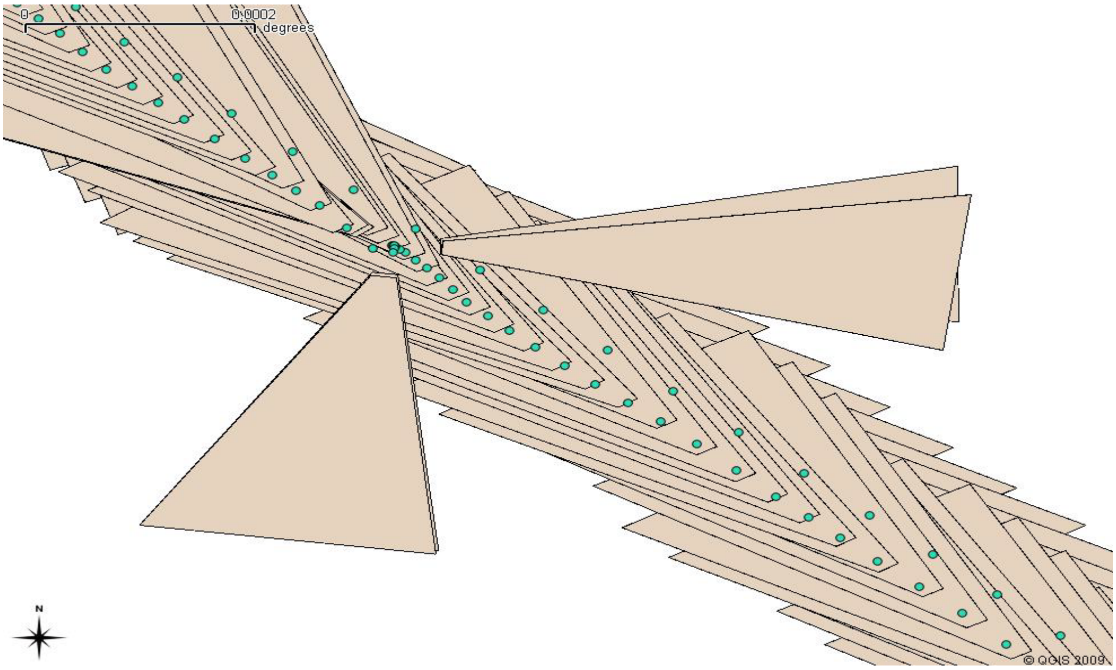


Fig. 6.13. Viewpoints database frame anchor points and associated ViewCones viewed in a GIS. This is a clear example of the automated spatial calculation problems generated from incorrectly detected GPS drift error.

The same method was employed to determine a satisfactory track smoothing in both the point to line representation as is shown in fig 6.10, but also in the ViewCone alignment situation. Fig 6.14 shows an example of the unsatisfactory representations of the ViewCones before track smoothing was employed. The transition between actual captured GPS points and the successive linearly interpolated ones defined a sharp change in the geographical extent representations as determined by the calculated ViewCones.

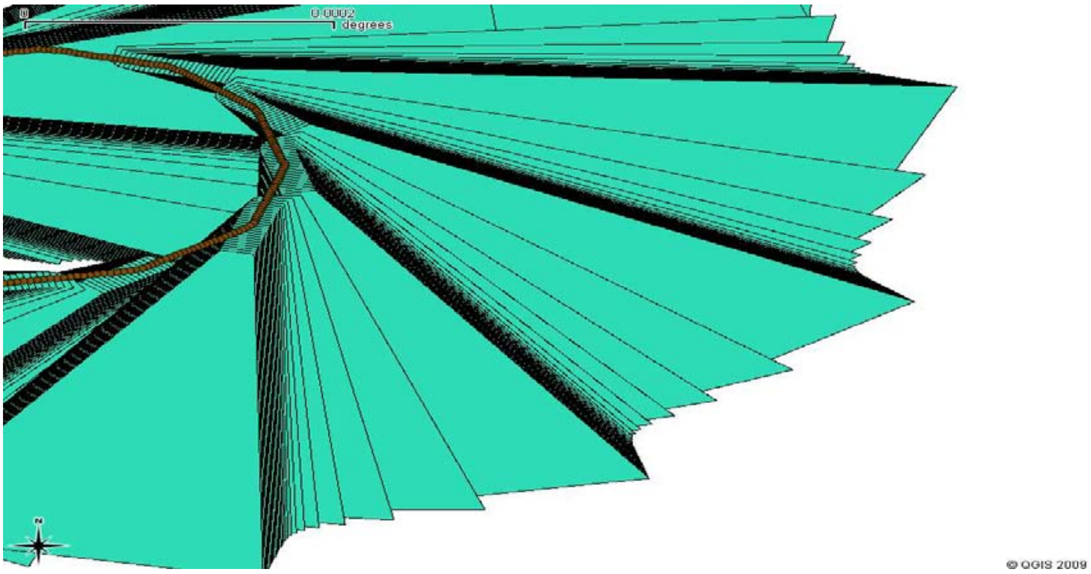


Fig. 6.14. This sample of the spatial data problems highlights the unsatisfactory transition between actual GPS capture points and the linearly interpolated ones.

The problems described in this chapter are predominantly related to a number of the generally well-known GPS errors inherent to any track data collection procedures and the processes necessary to synchronise and interpolate different data stream collection rates. However, most of these problems only materialised when performing the work detailed for chapter seven because of various irregularities that appeared in the query results. Once realised the Viewpoints database processing procedures where reworked to adjust or account for these spatial problems. Thus, a number of the important spatial data problems were discovered and solutions provided, which are explained in this chapter. While these solutions may not necessarily be optimal, they are certainly sufficiently accurate to achieve the overall objectives, as the details of chapter seven should show.

# Chapter Seven: Viewpoint Geospatial Operations

This chapter contains discussions on the semantic nature of how relevant GIS operations on Spatial Video Viewpoints should behave. It also describes implementations of practical examples based on these assertions. Spatial operations discussions will highlight a number of issues relating to the GIS functionalities introduced in chapter one and how they can be achieved through Spatial Video interaction using Viewpoints. As part of this study, a Location Based Services system, that can dynamically stream Spatial Video footage, based on the Viewpoint database model, is also discussed. This system highlights a laboratory demonstration of how a location aware video player can be dynamically controlled using Viewpoints.

## 7.1 Introduction

The process of developing a GIS-based model for Spatial Video has passed through a number of stages, from developing a fundamental understanding of video formats through to the creation of a functional GIS Viewpoints database. It is logical then that this chapter should finalise the process by discussing the types and methods of geospatial analysis that can be applied to use, retrieve and study Spatial Video from a number of different perspectives. Unfortunately, a definitive set of fundamental geospatial operations does not exist, as is evident in the numerous different operations sets available across many GIS and in the literature. Albrecht (1997) defines a comprehensive list of twenty universal analytical GIS operations that are data structure independent which, he suggests should form the basic building blocks of any GIS application. Fig 7.1 provides a graphical overview of this list. This work is highlighted in an environmental modelling and chart-based software interface, (Albrecht, 1996). A basic prototype tool, VGIS, was implemented that modelled applications of these operations using real GIS data. However, this is a conceptually high level set of processes which are in contrast to the OGC orientated approach used here. OGC's approach is originally grounded in lower-level SQL based specifications defined in (Egenhofer *et al.*, 1991; Egenhofer *et al.*, 1993).

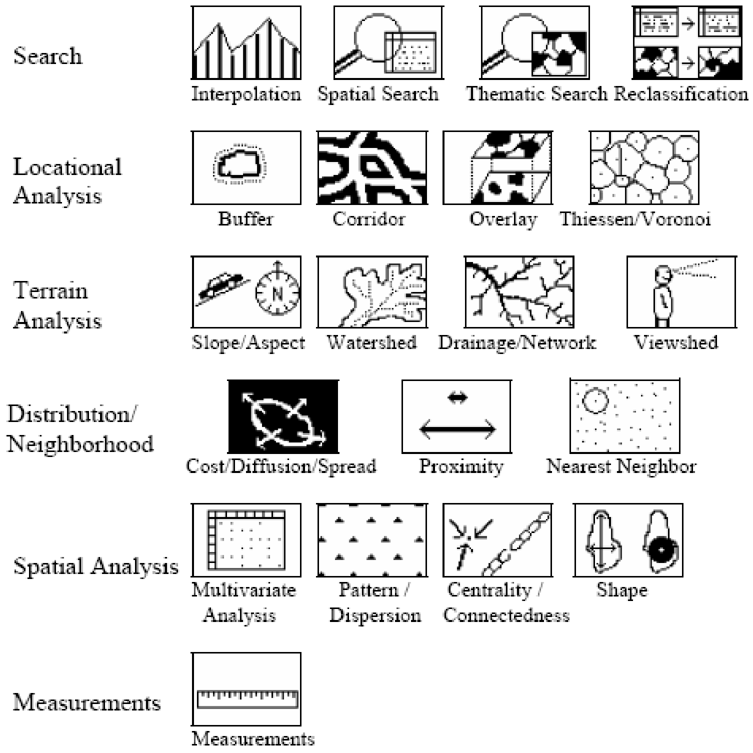


Fig 7.1. Universal Analytical GIS operations as defined in (Albrecht, 1997).

Longley *et al.* (2001) also define six broad spatial analysis headings, although they differ from Albrecht in that they are further divided into two groups where one concentrates on query, measurement and transformation while the second deals with statistical data mining, optimisation and hypothesis testing methodologies. While a large overlap with Albrecht's assertions exists at a functional level, this is not the case at a generalisation level. Thus, the approach mentioned in chapter one, section 1.1 is brought back into focus where the geo-spatial analysis testing is defined in terms of GIS functionality. Instead of trying to determine a definitive list of fundamental GIS operations that may be relevant to Spatial Video analysis, testing is performed on a Spatial Video Viewpoints database based on it being able '*to capture, model, retrieve, share, manipulate, analyse and present geographically referenced data'* in a realistic GIS context (Worboys *et al.*, 2004A).

In considering these seven functionalities it can be determined that the first two, *capture* and *model*, have been completed. *Capture* in terms of the acquisition and storage methodologies that have been detailed in earlier sections along with the many alternative systems available commercially and in the literature. *Modelling* has been defined in previous chapters through the Viewpoints implementation. The *retrieve, share, manipulate* and *analyse* aspects are concentrated on mostly using spatial SQL operations. The *present* aspect involves two considerations: how the Viewpoints structure displays in a GIS; and the video playback in a video player.

A number of changes to spatial operations need to be enforced because of the nature of Spatial Video and the Viewpoint data structure implementations. These are presented in a GIS context where three basic approaches to forming queries and determining results from the Viewpoint structures are investigated. Firstly, how should operations that relate to querying and retrieving multiple video sequences or images be performed based on a geographical search space? Secondly, how geo-spatial analysis operations should be used to study the Viewpoints spatial data structures? Thirdly, how non-video spatial data interaction should be introduced to improve a Viewpoints geographical representation? Importantly, these discussions provide a semantic context to this work as certain standard GIS spatial operations cannot properly perform a Spatial Video query without redefining the search context. Based on the complexity of the spatial objects and the expected video results, the changes may be distinct yet subtle refinements to the various standard operations.

To reinforce these points, a number of practical examples of Viewpoint-based GIS operations are detailed through applications of various spatial operations that are available as part of the PostGIS (2001) database schema. To perform these operations the Viewpoints database has been populated with a number of Spatial Video surveys. Thus, the study area in Maynooth is defined along with the Spatial Video survey footage details. These data sets will also be used, through the Viewpoints database, in a laboratory demonstration of a typical commercial style application for a Location Based Service (LBS) in-car satellite navigation system. This is based on a spatially controlled dynamic format video player.

Other examples involve non-video 2D spatial data sets; one is a buildings model of Maynooth town centre, the other is a land classification model. These are used to simulate spatial queries where video sequences or content information about video sequences is determined based on the non-video spatial data. The buildings model is also used in a polygon re-definition operation that shows the applicability of improving the Viewpoints geometric accuracy based on GIS analysis rather than image analysis.

### 7.2 Study Area

The study area that was surveyed for this chapter is the town of Maynooth in Ireland. The survey routes chosen consisted mainly of primary or secondary network routes that are used regularly in travelling through the town. A limited amount of the environs were also surveyed as was the M4 motorway that bypasses the town, between junctions six at Celbridge West / Leixlip West and junction eight at Kilkock. These surveys were defined over five separate routes that overlap with another route at least once. Fig 7.2 provides an overview of the survey routes.

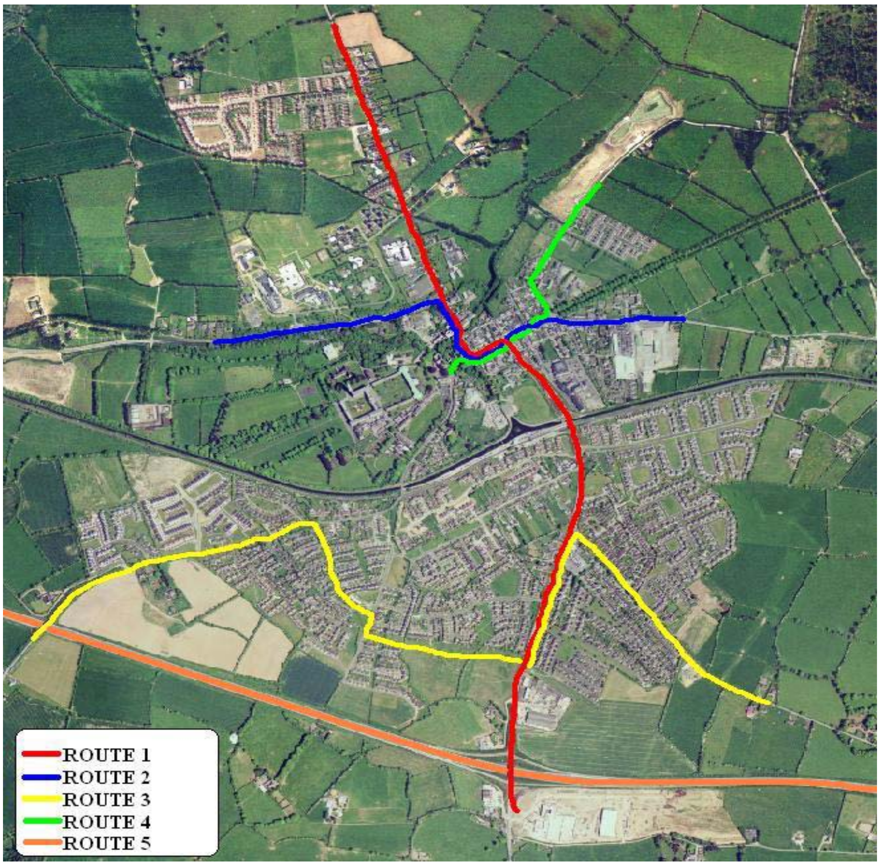


Fig. 7.2. Overview of the Maynooth area and the Spatial Video survey routes. The complete Route 5 is not shown in this image.

### 7.3 Spatial Video Data

The five survey routes were captured using a CamNav GPS hardware encoder and a video camcorder, (NavTech, 1995). Three models of video camcorder were used to capture the video footage; a Panasonic 150EB and 180EB and a JVC HD-111, which are described in chapter five, tables 5.1 and 5.2. The CamNav was connected to the camcorders through their audio input jack and the spatial data were encoded to the video audio stream using the methodologies detailed in chapter three, section 3.2.1. This hardware set-up was then mounted in a survey vehicle which travelled the routes in the order highlighted in Fig 7.2 and detailed in table 7.1.

| **Route Details** | | **Route Description** |
|---|---|---|
| **Route 1 (RED)** | | Started on Moyglare Road, through Maynooth Main St to business campus on Straffen Road. Returned by same route. |
| **File Format** | Windows Media Video | |
| **File Size** | 154 MB | |
| **Duration** | 10.33 minutes | |
| **Resolution** | 720 X 576 SD | |
| **Route 2 (BLUE)** | | Started on Kilkock Road, through Maynooth Main St to shopping centre on Leixlip Road. Returned by same route. |
| **File Format** | Windows Media Video | |
| **File Size** | 126MB | |
| **Duration** | 8.42 minutes | |
| **Resolution** | 720 X 576 SD | |
| **Route 3 (YELLOW)** | | Started on Celbridge Road, continued left onto Straffen Road then right onto Rathcoffey Road. Returned by same route. |
| **File Format** | Windows Media Video | |
| **File Size** | 176MB | |
| **Duration** | 10.43 minutes | |
| **Resolution** | 720 X 576 SD | |

Table 7.1. Continued on next page.

| | | |
|---|---|---|
| **Route 4 (GREEN)** | | Started on Dunboyne Road, through Maynooth Main St to NUIM South Campus Entrance. Returned by same route. |
| **File Format** | Windows Media Video | |
| **File Size** | 80MB | |
| **Duration** | 4.50 minutes | |
| **Resolution** | 720 X 576 SD | |
| **Route 5 (ORANGE)** | | Start at Kilkock M4 Interchange eight, surveyed along M4 to Leixlip West / Celbridge West Interchange six. Returned back along M4 to end at the Pylon Bridge after Maynooth Interchange seven. |
| **File Format** | MPEG-4 | |
| **File Size** | 184MB | |
| **Duration** | 12.51 minutes | |
| **Resolution** | 1280 X 720 HD | |

Table 7.1. Five Spatial Video survey routes detailed and described. SD is Standard Definition, HD is High Definition.

The camcorder frame rates operated at twenty five frames per second while the CamNav GPS encoder processed the spatial data at one hertz. The camcorder used on route five was a JVC High Definition (HD) progressive scan quality camera in which each video frame is a full resolution image recorded from all the sensor’s scan lines. This represents as complete and accurate a representation of that spatial location as the camera and conditions allow. This camcorder was also set for fixed zoom and fixed focus which has a direct impact of the simplicity of automated processing of the Viewpoint calculations. For every video image in the survey, the operational parameters and resultant ViewCone dimensions are theoretically identical. This assumption is dependent on the image object content as this will directly influence the real-world ViewCone dimensional representations.

The Panasonic camcorders operated at an output frame rate of twenty five frames per second while the resolution was of a Standard Definition (SD) interlaced quality. Thus, each video frame is composed of odd and even camera sensor scan lines that have been captured at a rate of fifty scans per second, i.e. each scan only forms half an image which is composed with the next half scan to form a full frame at twenty five frames per second. This method results in a degraded image quality when viewing a frame statically and has its historical roots in the display of video imagery on Cathode

Ray Tube (CRT) devices. This introduces a temporally dependent spatial issue, especially where Spatial Video is concerned, as the video is captured while the camcorder is moving, but also while objects it images are moving. Two considerations develop from this; firstly, to retrieve a full frame in this format a de-interlacing approach would need to be implemented, of which there are many known solutions. Secondly, defining the correct spatial data for an interlaced frame would theoretically require two location points to be calculated that are dependent on the velocity of the capturing equipment. The greater the velocity at the time a frame was recorded the greater the distance will be between these points for the same video frame.

These concerns point to a number of changes that should be considered for future modelling efforts based on the Viewpoint concept. Chapter six concentrates on defining an alignment between the spatial data and the video frames, without consideration for the frame content or how the frame is composed. This approach provided a sufficiently accurate implementation to justify the processes that were applied, and remain fully applicable to HD video formats. However, in an interlaced video stream each frame can contain spatial displacement. Therefore, by introducing a survey data type and quality level consideration an alternative context to the spatial alignment approach would be relevant. High accuracy alignment of SD video frames would need to implement approaches that account for the spatial location difference contained in the same frame data, except in cases of stationary captured frames.

These points have not been implemented in this study beyond a theoretical appreciation that indicates future work is needed to define these requirements. Two possible solutions are envisaged, one involving higher interpolation rates while the other would define an alternative to the Viewpoint geographical ViewCone structure. Firstly, interpolation at twice the frame rate for SD video may be a possible solution to accurate alignment; however this would require frame de-interlacing to be applied with a resultant much higher overhead on Viewpoint calculation and storage. It would also only need to be applied within a realistic accuracy range based on survey velocity; on stationary or very low velocity frames where any perceptible change in spatial content is minimised, no de-interlacing or double spatial interpolation will be required.

Alternatively, the Viewpoint model could extend its geographical space structures to define more complex spatial representations of a SD frame. This would involve changes in the depth of fields and angles of view. Here the representations could determine an optimal captured space as the union of the two spaces defined by the spatial differences of the odd and even scan lines.

### 7.4 Viewpoints and Geospatial Analysis

This section uses the Viewpoints database, populated with the survey data presented in the previous section, to discuss and perform a number of geo-spatial queries. This database was filled based on the procedures detailed in chapter six. These queries will demonstrate how the functionalities of *retrieval, sharing, manipulation* and *analysis* of Spatial Video can be performed in a GIS application. Using a combination of both simulated and actual geo-spatial data in the form of point, line and polygon data sets a number of queries are performed on the Spatial Video Viewpoints database. These data sets, used in various operations, will mix search, analysis and extensibility approaches to interacting with the Viewpoints. This approach highlights some semantic issues, practical implementations and relevant results, in such a way that all of these functionalities are essentially achieved or included in some form.

An important aspect of this section also includes a semantic approach to the Viewpoint data structures meaning. A number of operational cases highlight why both the Viewpoints point and polygon structures need to be considered in a manner that is different from the normal understanding associated with these data types. Essentially, the Viewpoints spatial structures cannot be used as independent units in spatial operations but must be considered as dependent relationships where each unit influences the other through a logical constraint based on the spatial operations goal. Also, these operations are discussed and performed in a sequence of lower to higher levels of complexity in both the spatial requirements and the Viewpoints structure determination. Initially, the original Viewpoint implementations are used in the query processes; however, more complex operations will define better geographical representations of the Spatial Videos object space.

### 7.4.1 Spatial Video Access Software

To complete these spatial operations a separate video access tool has been developed to visually test the results. This tool can dynamically access any of the queried video footage to the frame level and output these data as individual image files. It is written in the C# programming language and is based on the Tao framework Application Programming Interface (API), (Tao, 2006). This framework is only a C# interface to two other core components both of which are written in C, (SDL, 1998; FFmpeg, 2007). FFmpeg is the video access, decoding and navigation functionality component, while the SDL component provides the visual display and output functionality. This software was also developed for a number of other uses, which includes the video file analysis and frame index determination work highlighted in chapter two and a location aware video player detailed later in section 7.5.

### 7.4.2 Spatial Video Geo-Spatial Operations

The objective of this section is to perform geo-spatial search and analysis operations on the Spatial Video Viewpoint database. These include video image and sequence searches based on point, line and polygon data type interactions, but also some simple analysis such as coverage calculations. Specifically, Point-In-View/Point-Of-Interest, Point-Of-View, Line-Of-View, Polygon-In-View, Dissolving Polygon Boundaries and Thematic Coverage operations are looked at. Some of these will form result sets involving video frames, individually or as sequences, from spatial searches across multiple Spatial Video surveys. Subjective evaluation of these results is performed where entire sets of query-returned video frames are compared to the GIS overview, but also the image content accuracy is reviewed. For completeness a practical number of preceding and subsequent frames were also viewed based on their spatial proximity to the query objects, however these normally only confirm the fuzzy boundary nature of Viewshed type analysis in these modelling circumstances, as mentioned in Worboys *et al.* (2004B).

#### 7.4.2.1 Point-In-View/Point-Of-Interest Search

Used in this operation is a set of four arbitrary Points-Of-Interest in the study area region to perform a spatial search of the Viewpoints database. Fig 7.3 provides an overview of the four points and their position in relation to the Spatial Video tracks. The points were chosen to test the spatial operations in the following circumstances:

- One of the points should be unique to one route only.
- Two of the points could be viewable from multiple survey routes.
- One of the points should not be viewable from any survey route.

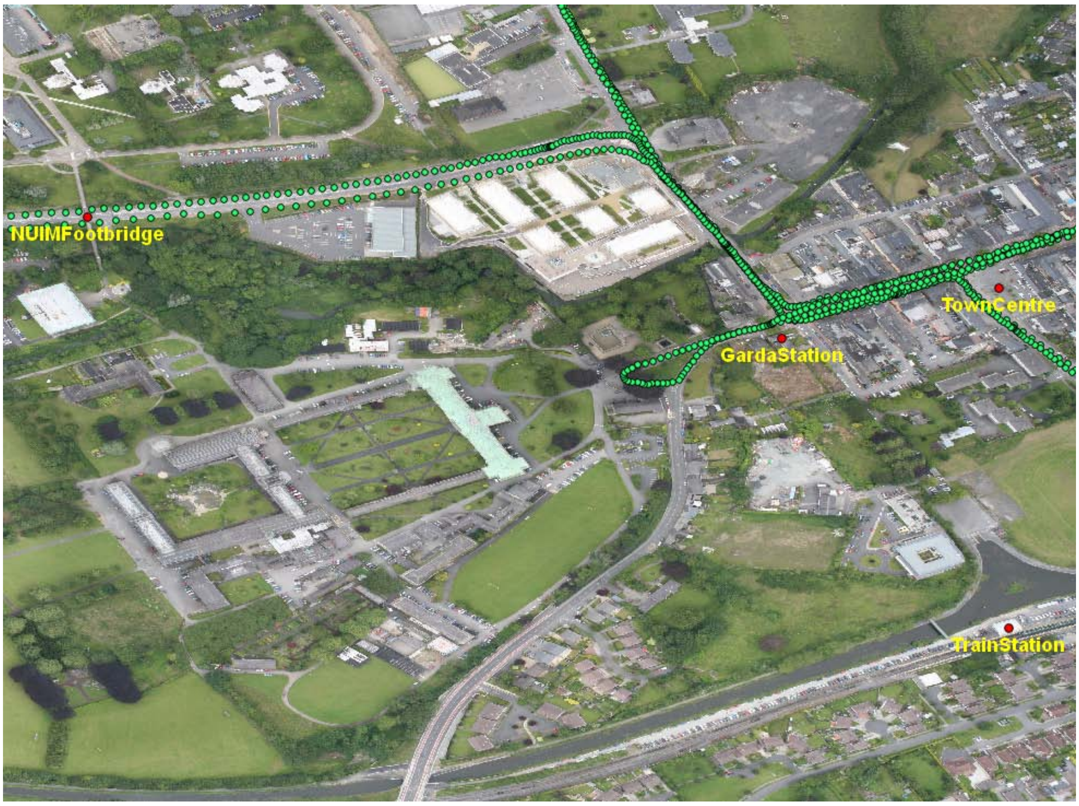


Fig. 7.3. Overview of the study area with four points of interested highlighted in relation to the Spatial Video routes. Image copyright National Centre for Geocomputation, NUIM.

The spatial SQL used to perform this query is:

*SELECT DISTINCT ON (svindex.sv_id) svindex.* FROM svindex INNER JOIN PointOfInterest ON ST_Within(PointOfInterest.the_geom, svindex.polygon_geom); (7.1)*

This query statement defines an SQL inner join operation to combine the records from the Spatial Video frame index table and the point-of-interest database. The join is controlled using the *ST_Within(geometry A, geometry B)* method which returns true where geometry defined in A is completely inside that of B. In this case A is the collection of point-of-interest points which are entirely contained within any Spatial Video stream Viewpoint as defined by the ViewCone geometry B.

The results from this query are summarised in table 7.2 and listed in more detail in appendix five. Subjective analysis of the results consisted of two steps, firstly, mapping the Viewpoints in a planar GIS environment to determine if the spatial extents proved reasonable based on empirical knowledge of the survey area. Secondly, the frames returned from the query were visually inspected to determine if the relevant point-of-interest object appears in the video images.

| Point-Of-Interest | Number Of Frames | Number and List of Surveys |
|---|---|---|
| NUIM Footbridge | 86 | 1 – Route 2 |
| Garda Station | 273 | 3 – Route 1, Route 2, Route 4 |
| Town Centre | 25 | 1 – Route 1 |
| Train Station | 0 | 0 |

Table 7.2. Results of spatial search operation based on Point-Of-Interest query of the Viewpoints database.

### 7.4.2.2 Point-Of-View Search

For this operation an arbitrary point is chosen from a route corridor centre line to search the Spatial Video database for all Viewpoints captured from or near this location. The chosen point-of-view is on Maynooth Town Main Street as this maximizes the search space over as many route sections with a common survey sector as possible. Three levels of incremental complexities are demonstrated as follows:

1. A simple point-in-space search is performed to find all coincident Viewpoints captured from this location.

2. A buffer is defined around the point-in-space to extend the relevant Viewpoints search space.

3. A directional view parameter based on Azimuth is included to control the point-of-view orientation in the return query.

The first level query tests for all points coincident with the point-of-view location by using the *ST_Equals(geometry A, geometry B)* method. This method returns true if the two geometries are spatially equal. The result of the query returned no frames as the route centre line is not coincident with the Spatial Video route track which is centred on the driving lane. The spatial SQL used to perform this query is:

```
SELECT DISTINCT ON (poly.sv_id) poly.* FROM svindex AS poly INNER JOIN
        PointOfView AS pt ON ST_Equals(pt.the_geom, poly.point_geom);        (7.2)
```

The second level query creates a five meter buffer around the point-of-interest using the *ST_DWithin(geometry, unit distance)* method. To facilitate easier use of this function, by passing in the unit distance in meters, a coordinate transformation to Irish National Grid was implemented using the *ST_Transform(geometry, SRID)* method. The results from this query returned 106 Spatial Video frames from two survey routes, Route 2 and Route 4, and are shown in fig 7.4. Appendix six highlights the start and end video frame sequences for these results. The spatial SQL used to perform this query is:

```
SELECT DISTINCT ON (poly.sv_id) poly.* FROM svindex AS poly INNER JOIN
    PointOfView AS pt ON ST_DWithin(ST_Transform(pt.the_geom, 29903),
              ST_Transform(poly.point_geom, 29903), 5.0);                    (7.3)
```

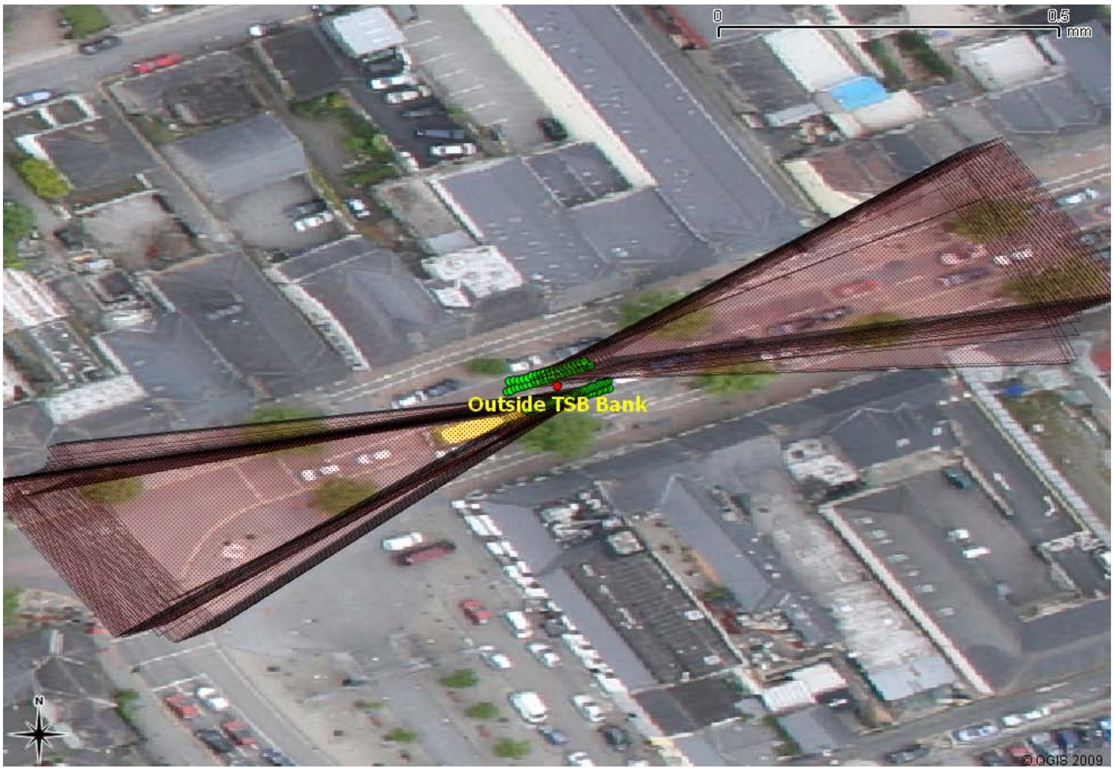


Fig 7.4. Point-Of-View spatial search results for SQL query 7.3. The five meter buffer around the point-of-view returned 106 Spatial Video Viewpoints. The viewpoint locations are shown as the green points while each associated ViewCone is shown as the transparent overlaid polygons.

The third level query also uses a five meter buffer around the point-of-interest using the *ST_DWithin(geometry, unit distance)* method; however the operation is constrained with a requirement for an orientation parameter. To achieve this, two approaches are possible; firstly, a simulated azimuth range can be passed to the query or, secondly, the azimuth can be calculated using the *ST_Azimuth(geometry pointA, geometry pointB)* method. This method operates in radians so conversion to degrees is required for interaction with the Viewpoints database. Also, the method is used by generating a point-of-interest in the chosen viewing direction as viewed from the point-of-view. A four degree tolerance is also defined for the azimuth in this operation as the course between any two GPS track points can vary. The results from both these query options returned 37 Spatial Video frames from two survey routes, Route 2 and Route 4. A sample set of the video frame image results are shown in appendix seven. Fig 7.5 displays a GIS overview of these results.

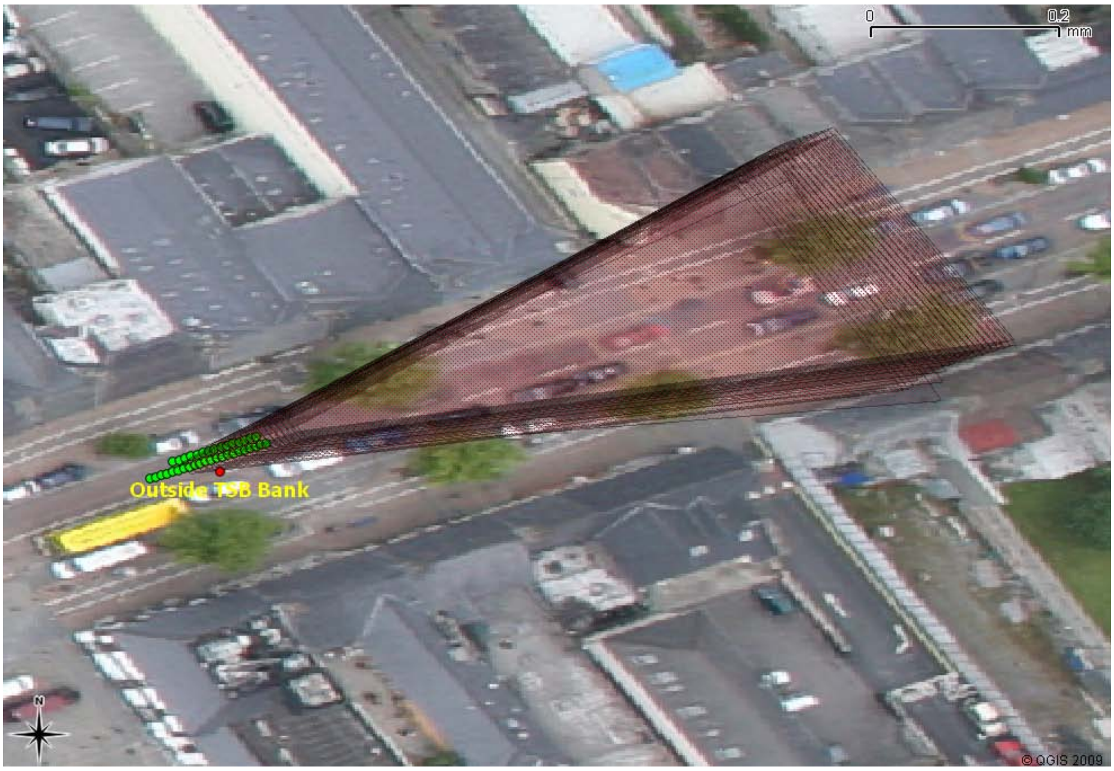


Fig 7.5. Point-Of-View spatial search results for SQL query 7.4 and 7.5. The five meter buffer around the point-of-view had an azimuth orientation constraint in the range of 62 to 71 degrees and returned 37 Spatial Video Viewpoints. The viewpoint locations are shown as the green points while each associated ViewCone is shown as the transparent overlaid polygons.

The two spatial SQL statements used to perform these queries are:

```
SELECT DISTINCT ON (poly.sv_id) poly.* FROM svindex AS poly INNER JOIN
PointOfView AS pt ON ST_DWithin(ST_Transform(pt.the_geom, 29903),
ST_Transform(poly.point_geom, 29903), 5.0) AND poly.azimuth BETWEEN 62
AND 71;                                                            (7.4)
SELECT DISTINCT ON (poly.sv_id) poly.* FROM svindex AS poly INNER JOIN
PointOfView AS pt ON ST_DWithin(ST_Transform(pt.the_geom, 29903),
ST_Transform(poly.point_geom, 29903), 5.0) AND poly.azimuth BETWEEN
(ST_Azimuth(pt.the_geom, ST_PointFromText('POINT(-6.5903305 53.3817925)',
4326))/(2*pi())*360)-4 AND (ST_Azimuth(pt.the_geom, ST_PointFromText('POINT(-
6.5903305 53.3817925)', 4326))/(2*pi())*360)+4;                   (7.5)
```

SQL query 7.5 approximately calculates the same azimuth search bounds as used in query 7.4. This method was chosen to simulate an interactive query performed in a

GIS interface where a user would generate the viewing orientation. This operation could take many forms but would essentially involve the point-of-view forming an axis point around which the orientation azimuth could be calculated based on another point selection in an interactive interface.

#### 7.4.2.3 Line-Of-View Search

The context for this operation is to extend the principles implemented in the point-of-view section into a search along a line such as a road network route or centre line. Performing this operation is straight forward for a basic query using a line geometry that represents the search space. In this case the SQL query 7.3 point geometry (*pt.the_geom*) can substituted with the line geometry. This results in a very large data set of all Spatial Video frames, across multiple streams, which are within a five meter distance of the line-of-view. While this is a valid practical result, semantically, it is difficult to spatially interpret other than to define that the result routes have different temporal metadata. It is also more difficult to visualise as a playable video stream because of the number of different surveys returned in the query. They highlight all routes, from all available viewing orientations, for all overlapping route sections. The results are overviewed in fig 7.6, while detail of the differences is highlighted in table 7.3 and fig 7.7.

| **Number of Routes = 3** | | **Number of Viewpoints (Frames) = 7683** | |
|---|---|---|---|
| **Sequence Set** | **Survey Route** | **Frame Start** | **General Orientation** |
| 1 | Route 1 | 149280 – 163880 | Right – Left |
| 2 | Route 1 | 499080 – 517960 | Left – Right |
| 3 | Route 2 | 267259 – 412419 | Right – Left |
| 4 | Route 2 | 174259 – 207059 | Left – Right |
| 5 | Route 4 | 63680 – 132040 | Right – Left |
| 6 | Route 4 | 143120 – 243320 | Left – Right |

Table. 7.3. Line-Of-View search results from the Viewpoints database summarised.

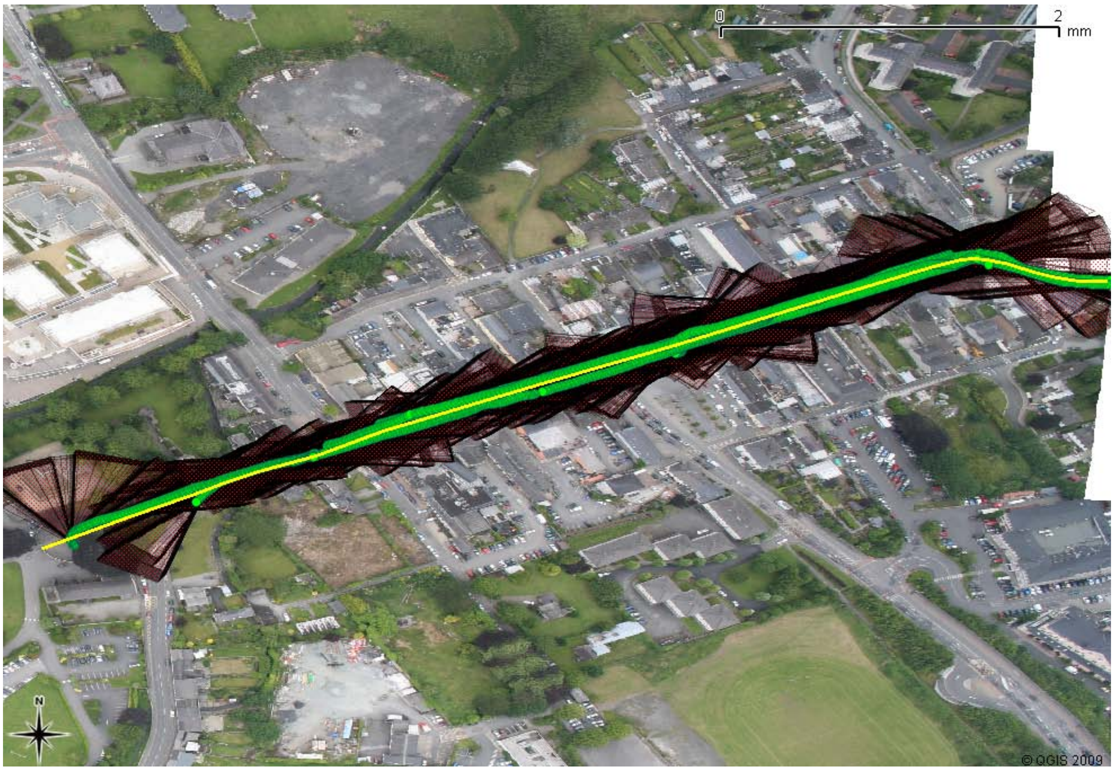


Fig. 7.6. Line-Of-View (yellow line) search overview for all Spatial Video Viewpoints captured along the road centre line that defines Maynooth Main Street. The green points are the ViewCone capture points while the red polygons are the associated polygons.

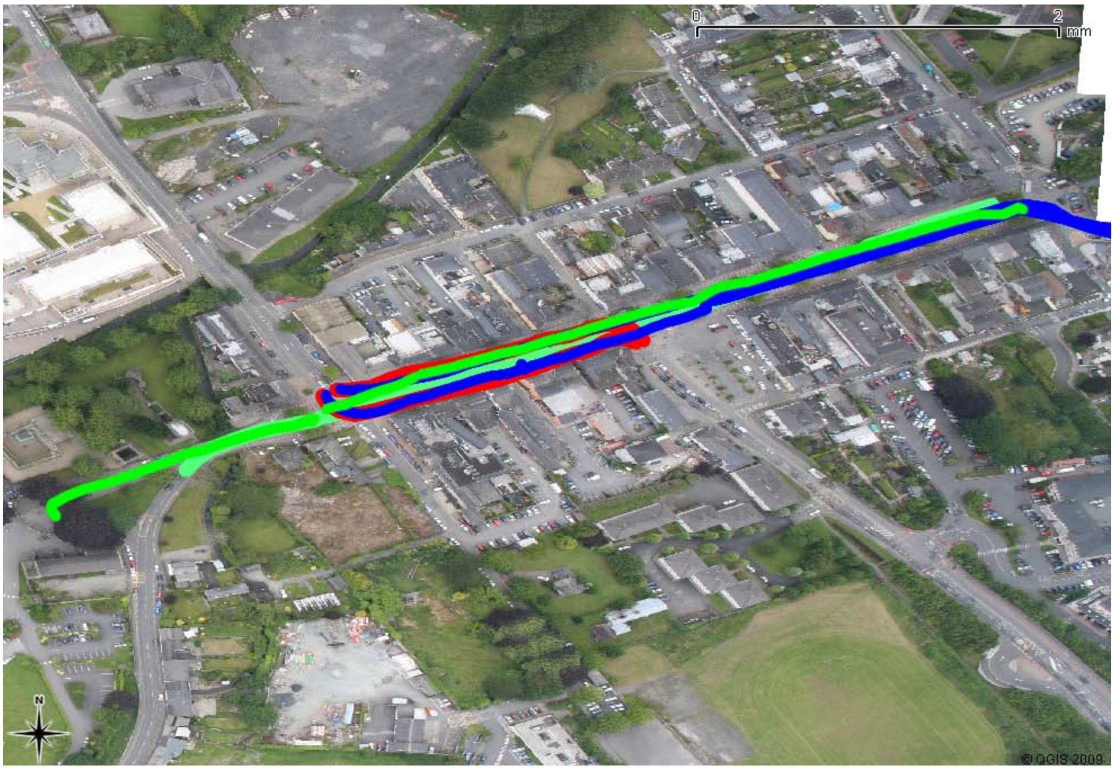


Fig. 7.7. This figure highlights the Line-Of-View search for all the sections returned from the basic query where both practical and semantic comprehension of these many different routes is difficult.

A more realistic option would involve more specific requirements which could include query constraints relating to specific routes by any conceivable metadata variable such as temporal, route id, survey equipment, etc. parameters. Alternatively, spatial constraints would seem more appropriate as the nature of the data is visual recording of the space. Thus, the Spatial Video footage search could be constrained such that it is aligned with the direction of the line, i.e. an extension of the point-of-view orientation in SQL query 7.5; or, that the returned footage maintains consistency across routes, i.e. only one route gets returned that maintains the maximum continuity from as small a number of surveys as possible. Extending this type of query operation to this more complex search requirement is not possible through a single standard SQL operation. Instead, a scripted SQL query has been written that is shown in appendix eight. This script firstly compiles a list of all possible survey routes along the line and orders them by highest quantity to determine the priority survey routes. Viewpoints are only returned for the survey that is most prominent at each line segment. Secondly, the line is deconstructed into its constituent sections, where each section is used to search against the Spatial Video database for all Viewpoints that fall within a buffer distance of the line. The results are also constrained to have an azimuth orientation that falls within a definable range of the line segment azimuth. The results of this operation are shown in fig 7.8 and the video detail attribute results are shown in table 7.4.

| **Number of Routes** | | **Number of Viewpoints (Video Frames)** | |
|---|---|---|---|
| 2 Routes | | 1789 | |
| **Sequence Set** | **Survey Route** | **Frame Start** | **General Orientation** |
| 1 | Route 4 | 145120 – 244080 | Left – Right |
| 2 | Route 2 | 202619 – 207059 | Left – Right |

Table. 7.4. Video route profile results for the constrained Line-Of-View search.

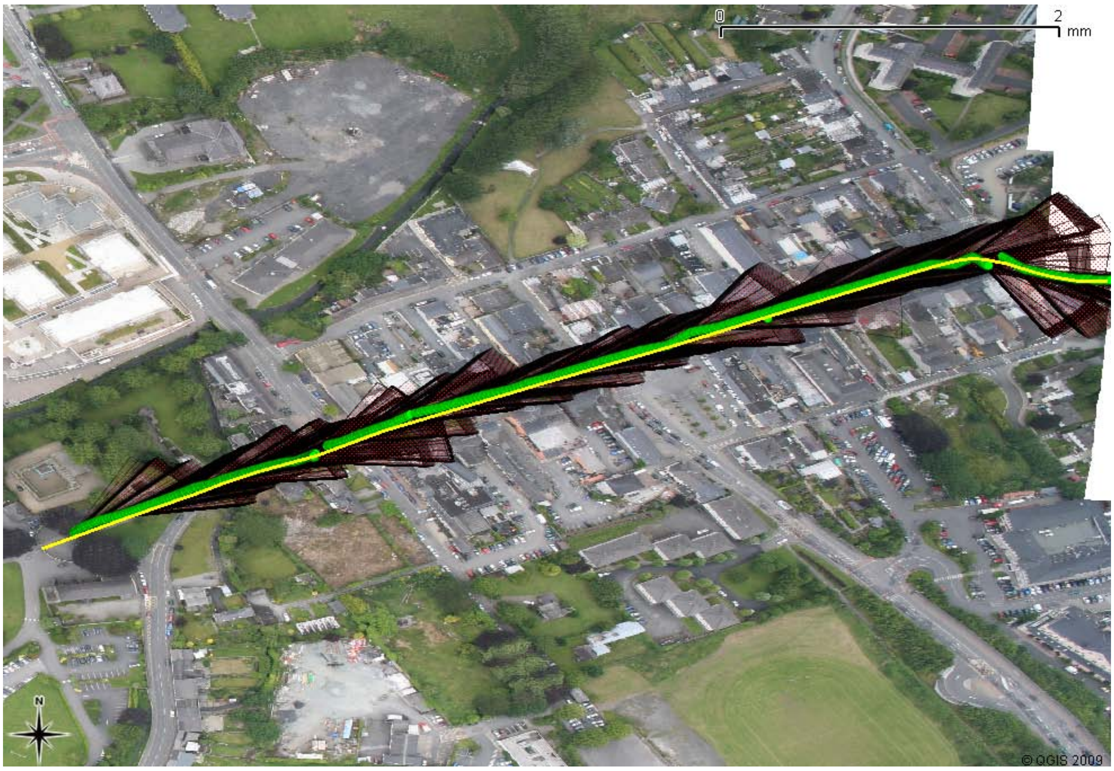


Fig. 7.8. Line-Of-View (yellow line) search constrained on azimuth and consistency of survey. The line direction has been defined from left to right so the query only returned routes captured while the survey vehicle captures frames orientated or travelling east.

The results of this constrained operation provide a more meaningful video replay experience as the footage is continuous over one route for most of the line-of-view and switches to only one other route for the last section.

### 7.4.2.4 Polygon-In-View Search

Finalised in this section are the spatial search operations by including the polygon data type. A census district polygon data set for the Maynooth area is used to query the Viewpoints database for all video frames that view an individual region. In this simulated case the Clane Road polygon (bottom left corner region) is chosen as the query region as its location in relation to the surrounding areas and the available Spatial Video routes provides the greatest variability in Viewpoint capture scenarios. The Maynooth census area polygon data overlay is shown in fig 7.9.

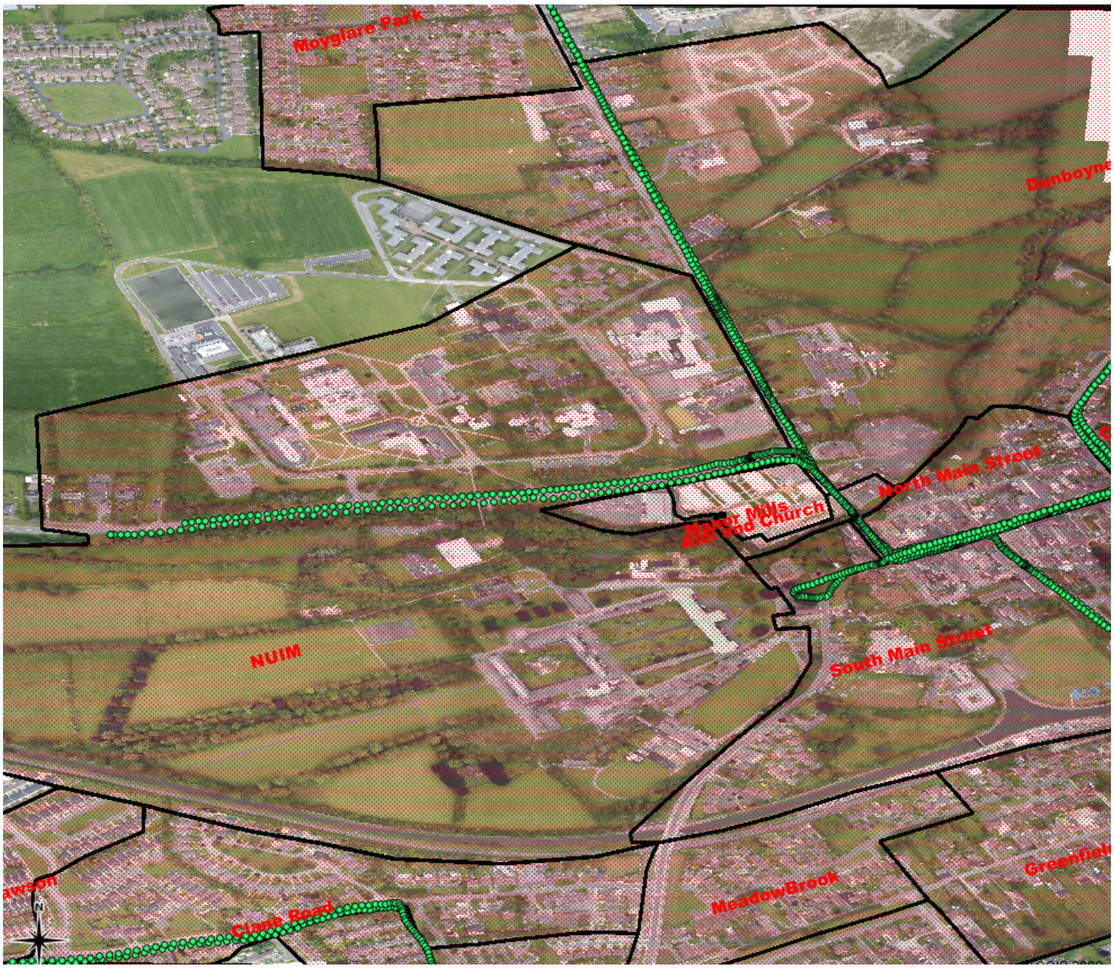


Fig. 7.9. Census data polygon districts overlaid on the Maynooth area orthographic photo with some of the Spatial Video surveys tracks highlighted in green. The Clane Road region is bottom left in this image.

This operation raises an issue with the semantic understanding of the Viewpoint data structure and what constitutes a view of a polygon. In this case a simple point-in-polygon spatial data operation will only retrieve what Viewpoints were captured within the polygon region. For many Viewpoints this would work correctly; however, for some of the points captured in the polygon, many of their associated ViewCones would extend outside the region. Therefore, these Viewpoints do not provide a view of the query polygon so should be excluded from the result set. A ViewCone coverage control, based on its intersection with the polygon, is required. Setting this control should be possible through reasonable polygon coverage ranges where a ViewCone that is classed as capturing the query space can be returned based on its percentage of coverage area. This concept will also extend to Viewpoints captured outside the polygon where an associated ViewCone captures geographical space within the search region.

SQL query 7.6 provides a solution where all Viewpoint ViewCones are either fully contained or have more than 60% of their structure contained within the query polygon. The fully contained operation is achieved using the *ST_Within(Geometry A, Geometry B)* method. Two steps are required to perform the greater than 60% intersection and are achieved using the *ST_Area(geometry)* and the *ST_Intersection(geometry A, geometry B)* methods. Firstly, the average ViewCone area is calculated for all Viewpoints as the base for a reference control range. Secondly, intersecting geometry areas are calculated and measured against 60% of the reference control to determine if a Viewpoint is valid. The results are shown in table 7.5 and in fig 7.10 and highlight how certain Viewpoints that were recorded outside the search space but capture geographical space within it are included. The converse is also true, but not shown in fig 7.10, where Viewpoints recorded within the search space are excluded because they capture geographical space that is not representative of the query polygon.

```
SELECT sv_id,file_source,frame_number,point_geom,polygon_geom FROM svindex
WHERE (ST_Area(ST_Transform(ST_Intersection(polygon_geom,
(SELECT wkb_geometry FROM "smallareas" WHERE ogc_fid = 7)),29903))
>= (SELECT AVG(ST_AREA(ST_Transform(polygon_geom, 29903)))/100*60
FROM svindex)) ORDER BY file_source, frame_number ASC;                    (7.6)
```

| Polygon-In-View | Survey Route | Frame Sequence | Number of Frames |
|---|---|---|---|
| Clane Road | Route 3 | 231680 – 277240 | 910 |
| | | 403600 – 433720 | 751 |

Table. 7.5. Video route profile results for the constrained Line-Of-View search.

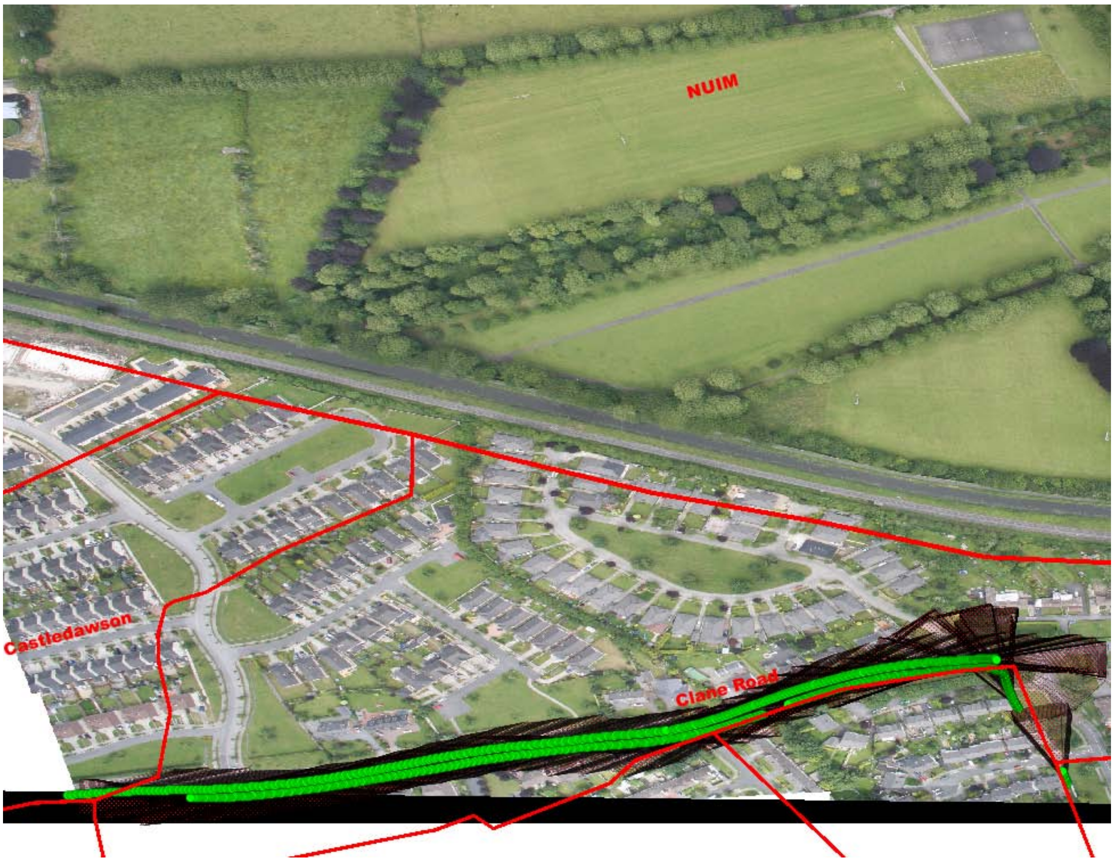


Fig 7.10. Results of Polygon-in-View operation where the Clane Road query polygon is the search space. The result Viewpoints are shown as the green points with transparent red polygons. The geographical extent of each Viewpoint is either fully contained within the search space or more than 60% of the structure is contained within it.

### 7.4.2.5 Calculating Coverages

The previous section isolated all Viewpoints where greater than 60% of the area they capture is contained within the query polygon. In this section some basic analysis is undertaken by calculating the coverage area of these results in square meters. This is achieved using the *ST_Union(geometry)* method which unions all the Viewpoint ViewCone geometries that intersect the query polygon by 60%, or greater, of their area into a single polygon. The geometry can then be converted into an Irish grid locale coordinate system and calculate the area in square meters using the *ST_Area(geometry)* method. The result from this operation calculated a Spatial Video captured coverage area of 18,145.75m$^2$. This is shown in fig 7.11, while the SQL query is shown in equation 7.7.

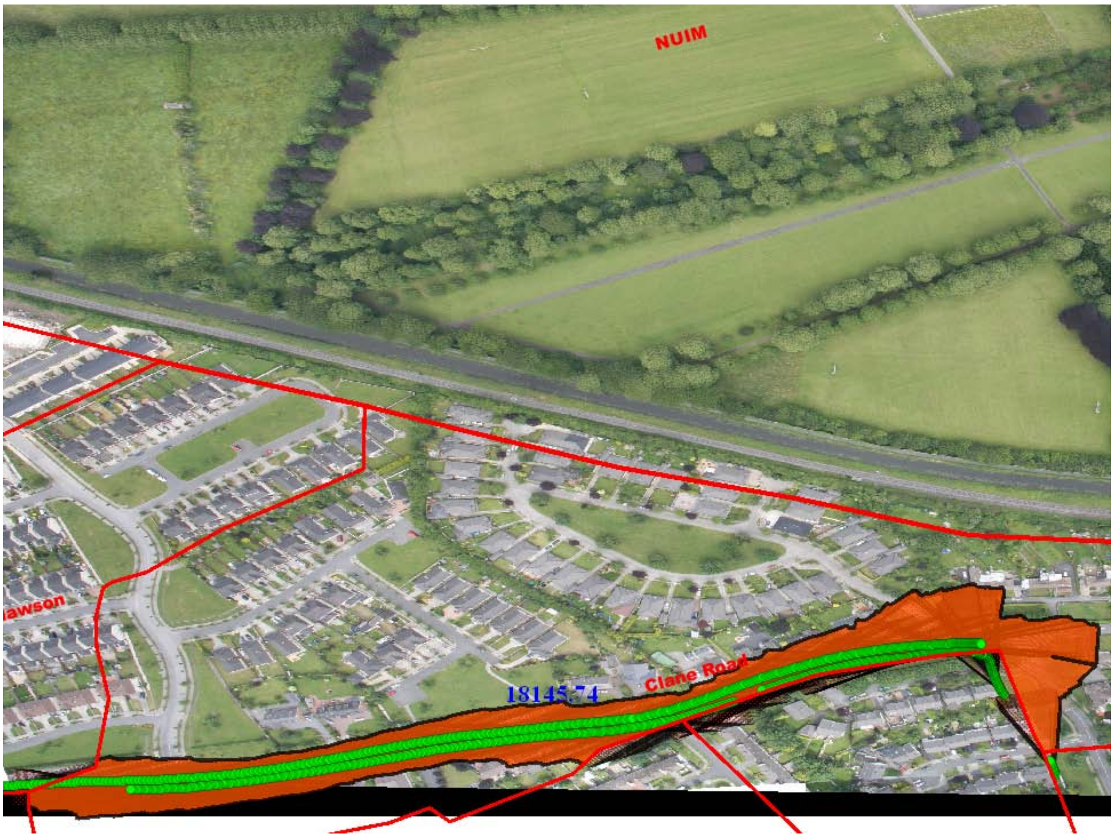


Fig. 7.11. Calculating coverage areas for the polygon-in-view operation results. The Union of all query polygon intersecting Viewpoints is shown in orange overlaid on the original search query results.

```
SELECT coverage.geom, round(CAST (ST_Area(ST_Transform(coverage.geom,
29903)) AS numeric), 2) FROM (SELECT ST_Union(ST_Intersection(polygon_geom,
(SELECT wkb_geometry FROM "smallareas" WHERE ogc_fid = 7))) as geom.
FROM svindex WHERE (ST_Area(ST_Transform(ST_Intersection(polygon_geom,
(SELECT wkb_geometry FROM "smallareas" WHERE ogc_fid = 7)),29903))
>=(SELECT AVG(ST_AREA(ST_Transform(polygon_geom, 29903)))/100*60 FROM
svindex))) as coverage;                                            (7.7)
```

While this operation provides accurate results given the input data, once again it is not a completely realistic representation of the search area. Much of the video footage in this search area is constrained by residential properties and high banked hedgerows which have never been considered in the original Viewpoint construction model. One approach to improve this would involve computer vision based image analysis as part of the process; however, in the subsequent sections it will be shown how this could also be achieved in a spatial data domain.

### 7.4.2.6 Thematic Coverage Operations

As is highlighted in Kraak *et al.* (2003), thematic mapping has many varied applications for the communication of GIS information from a geo-spatial analysis operation. Used in this sample operation is a thematic metadata assignment for each land parcel from the previously used polygon data, to simulate a land-use cartographic layer, as shown in fig 7.12. This thematic mapping environment can then be used to query the Spatial Video Viewpoints' database for many different spatial analysis reasons. In this case two objectives are defined; one is to calculate the quantity of geographic space captured in each polygon, across all Spatial Video surveys, while the second is to quantify the video content by the type of geographic coverage it contains.

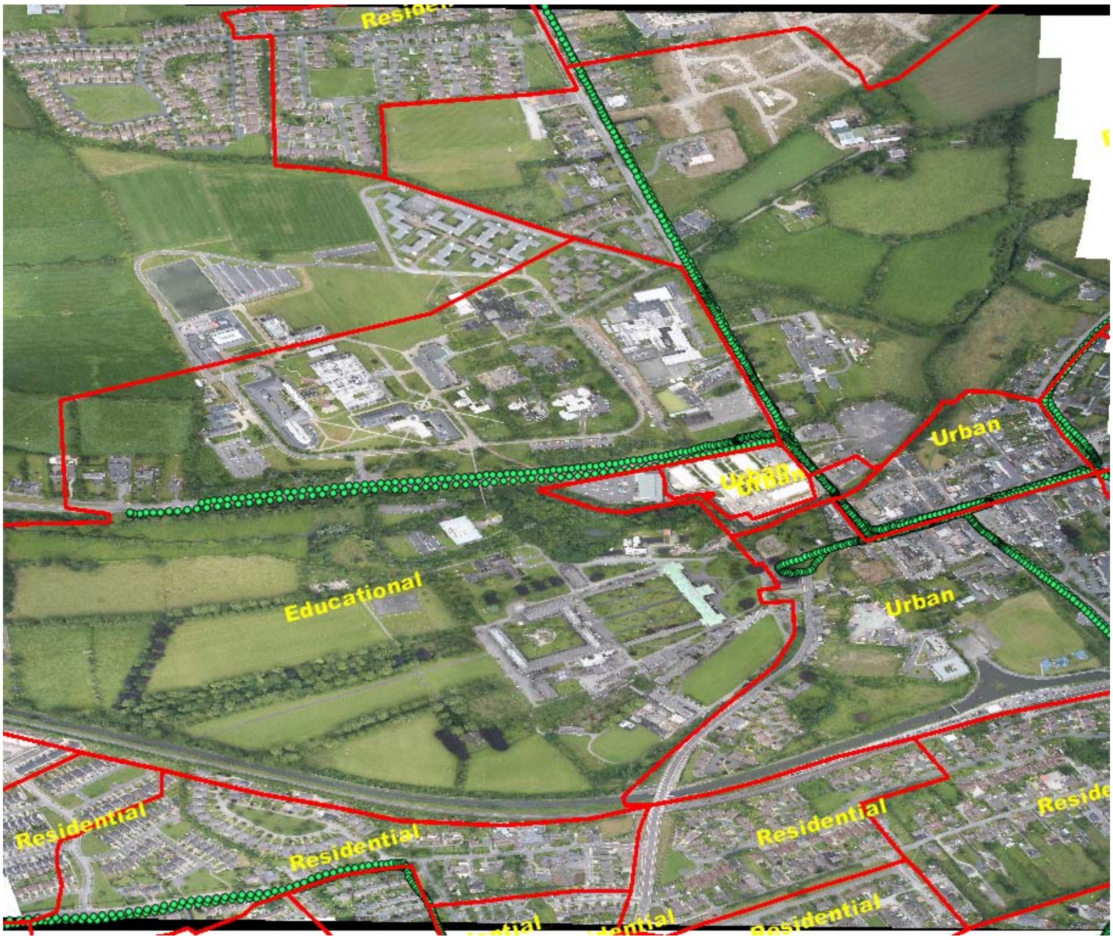


Fig 7.12. Previously used census area polygon layer defined by a simulated land usage. Geo-spatial analysis operations on this data set include calculating volumetric Viewpoint coverage's and quantifying video content by coverage area.

Detailed in fig 7.13 is the spatial extent of the coverages calculated with the SQL script listed in appendix nine. This script draws on previous operations where it sequentially processes each query polygon against the Spatial Video database to retrieve all Viewpoints where the ViewCone has a 60% or greater geographic capture space. Each Viewpoint, or portion, is aggregated into a single polygon that represents the total geographic space captured for each polygon from across all the Spatial Video surveys. This is achieved using the spatial union, *ST_MemUnion(gemoetry)*, method which defines a single output geometry from multiple input geometries. Based on these results, the coverage area and percentage coverage area of each polygon can be calculated and is shown in result table 7.6.

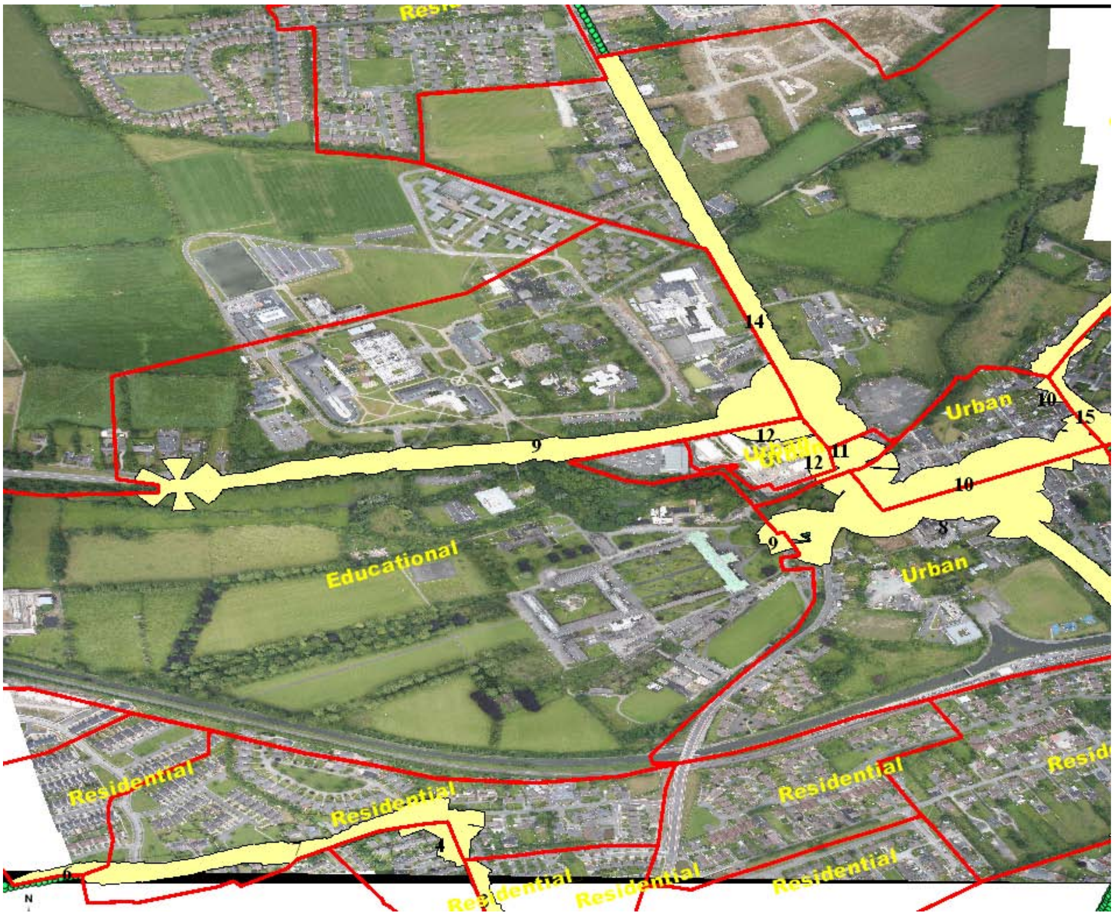


Fig. 7.13. Shown here is an overview of the spatial union of all the Viewpoints that represents the total coverage per polygon captured in the Spatial Video surveys.

Three of the polygon areas have a zero coverage result, which means that no Viewpoints from any of the Spatial Video survey routes capture these areas, as is shown in fig 7.13 and the coverage values in table 7.6.

| Area ID | Area Name | Land Use | Area m² | Coverage % |
|---|---|---|---|---|
| 1 | Greenfield | Residential | 504.38 | 0.5 |
| 2 | Meadowbrook | Residential | 0 | 0 |
| 3 | Beaufield | Residential | 5,701.93 | 12.9 |
| 4 | College Green | Residential | 8,788.08 | 21.4 |
| 5 | Kingsbury | Residential | 0 | 0 |
| 6 | Castledawson | Residential | 2,086.11 | 3.9 |
| 7 | Clane Road | Residential | 18,145.74 | 14.9 |
| 8 | South Main Street | Urban | 38,106.12 | 17.5 |
| 9 | NUIM | Educational | 38,803.22 | 4.3 |
| 10 | North Main Street | Urban | 18,721.96 | 42.1 |
| 11 | Aldi and Church | Urban | 3,568.42 | 23.5 |
| 12 | Manor Mills | Urban | 5,286.77 | 32.5 |
| 13 | Moyglare Park | Residential | 0 | 0 |
| 14 | Dunboyne/Moyglare Road | Rural | 40,365.74 | 5.5 |
| 15 | Carton Avenue | Residential | 5262.79 | 12.4 |

Table 7.6. In this table the total area of the Viewpoints' geographical capture space, per polygon thematic region, across all Spatial Video surveys is calculated. Also shown is the percentage of each polygon that is captured.

Lastly, based on the result of this operation some Spatial Video analysis can be extrapolated by approximately determining the percentages of video thematic content based on the polygon's land use metadata. In this case aggregate results are produced; however, by implementing some minor changes to the script the video duration, survey routes or sequence frames could also be calculated instead. The results are shown in table 7.7 and are based on a total coverage of all viewpoints being calculated as 810,478.47m². This calculation of total coverage proved to be a non-trivial task, as standard geometric union operations are iterative and computationally expensive in the PostGIS environment. The Spatial Video Viewpoints database contains approximately 75,000 overlapping polygons and using the standard union methods the algorithm had not completed its task after sixteen hours. A cascade union method was implemented based on Springmeyer's (2008) algorithm and provided the solution in six minutes.

The different approaches to implementing the spatial union methods is generally well known with the more efficient algorithms being implemented in newer versions of the PostGIS spatial operations toolset, which were not available at the time of writing. However the main differences in how the operations work involve both memory management and data structure manipulation. The standard union simply joins the first two polygons, and then the result is joined to the third polygon with this process continuing for all polygons in the union set. The cascade algorithm builds an R-tree data structure where the union of the end nodes of the tree, which are the smallest subsets of the polygons, determines a result for each parent node. Each level of the tree is recursively processed until the final union of all polygons is produced.

| **Coverage Type** | **Area m² Total** | **% of total Spatial Video** |
|---|---|---|
| Residential | 40,489.03m² | 5.0 |
| Urban | 65,683.27m² | 8.1 |
| Educational | 38,803.22m² | 4.8 |
| Rural | 40,365.74m² | 5.0 |
| Undefined | 625,137.21m² | 77.1 |

Table 7.7. Aggregate results of the Spatial Video database, where the video content is determined to contain various percentages of thematic geographical content based on the polygon data sets metadata.

The undefined element in table 7.7 is the area of all Viewpoints that are not classified into any of the other coverage types. This is because the Spatial Video survey area covers a larger spatial extent than the thematic polygon data set does. Therefore, 77% of the Spatial Video does not intersect with the thematic coverage data set.

### 7.4.3 Spatial Video Accuracy and Extensibility Operations

This section introduces another non-video spatial data set for use in an operation that improves the spatial accuracy of the Viewpoint geographical extent. The original model defines a maximum Viewpoint extent calculated from the camera and spatial parameters; however this approach does not consider the video content such as any terrain based occlusions like buildings, hills, and infrastructure elements. These all have a direct impact on the line-of-sight from a video capture point which should be reflected in the Viewpoints geographical representation. As has already been mentioned, the current Viewpoint calculation model is defined very simply and that adding an image processing approach could lead to improved Viewpoint representations for each frame. However, it is not unreasonably to perform this improved accuracy objective by using existing spatial data to determine a more realistic geographical space. This approach can be used as a novel way of demonstrating how a higher level of accuracy in the Viewpoint shape can be achieved. It will also highlight the extensibility and flexibility of the Viewpoints data structures and their interaction within a GIS.

#### 7.4.3.1 Spatial Difference

The objective here is to model the Viewpoint's ViewCones more accurately based on relevant spatial data from alternative, non-video, sources. For this example a buildings footprint spatial data layer of the Maynooth town centre buildings has been defined. This is shown in fig 7.14 where the buildings data layer has been overlaid onto the complete Spatial Video Viewpoints coverage layer, as calculated for the previous section. To achieve this objective visibility analysis has been used where a Viewshed operation has been developed that generates an adjusted geometric visibility polygon from each Viewpoint that intersects any of the buildings in the buildings layer. This represents a re-calculation of the Viewpoint ViewCone based on the obstacles that logically restrict the viewing field line-of-sight.

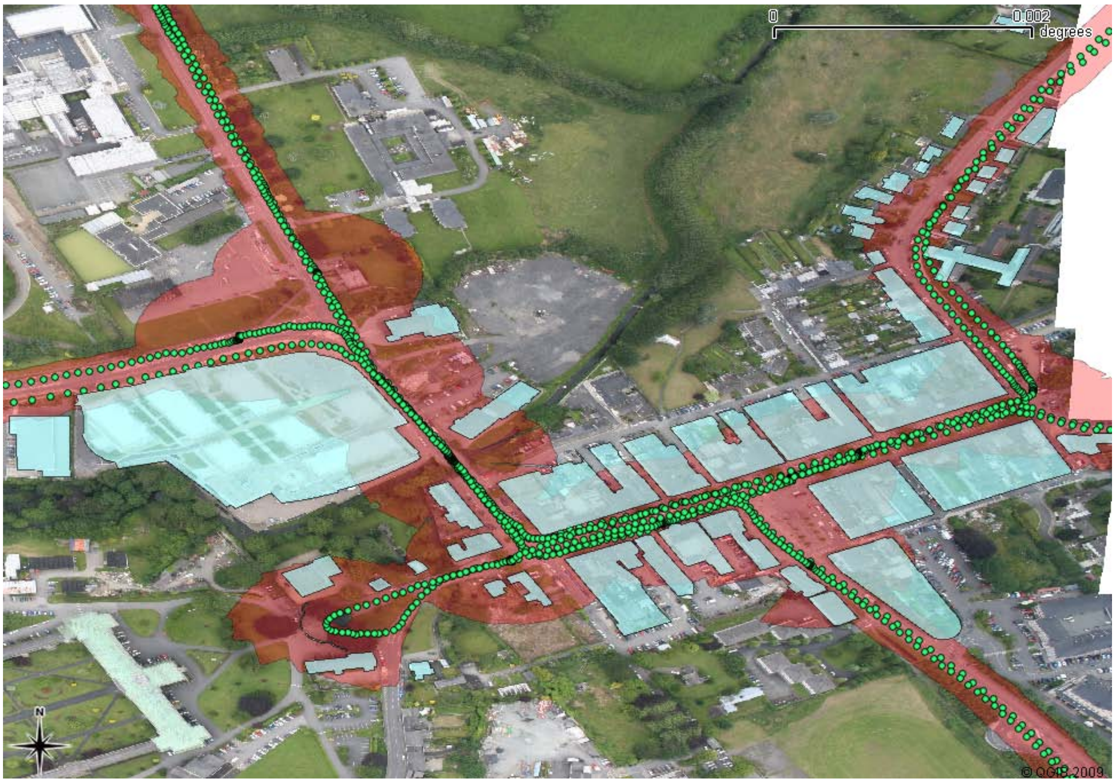


Fig. 7.14. Viewpoint accuracy intersection model. The green points are the survey routes track data, the red polygon is the total Viewpoint coverage space and the blue polygons are the building footprints for all the buildings in Maynooth town centre.

There are a number of established methods and software interfaces for performing geometric visibility analysis based on line-of-sight intersection testing, that includes both vector and raster data type support, (GRASS, 1982; Turner, 1998; Rana, 2001; Obermeyer, 2006; ESRI, 2007). However, none of these sources provided suitable options for implementing this type of analysis under these conditions. None of the sources facilitated both dynamic access to a PostGIS spatial database and processing of spatial geometry in vector format. They also typically define their operations based on variables already established in the Viewpoint structure, such as the field-of-view, direction-of-view and point-of-view. Operations are performed based on a point-in-space, the viewpoint, from which all the line-of-sight intersection computations are calculate. However, as has been mentioned in chapter four, the video frame image does not capture space from a point but from a definable line, the near depth-of-field.

In the PostGIS spatial database environment no Viewshed operation is defined for this type of problem. Fig 7.15 shows an overview of the operations requirements for a single Viewpoint. PostGIS does define spatial difference operators for overlaid geometries; however when used in this context normally result in incorrect viewable regions that are often disjoint. In the case of fig 7.15 it defines the yellow and red regions, which is a correct spatial difference operation but definitely not a correct Viewshed result. Thus, a bespoke PostGIS operation was developed to achieve the objective. This is listed in appendix ten.

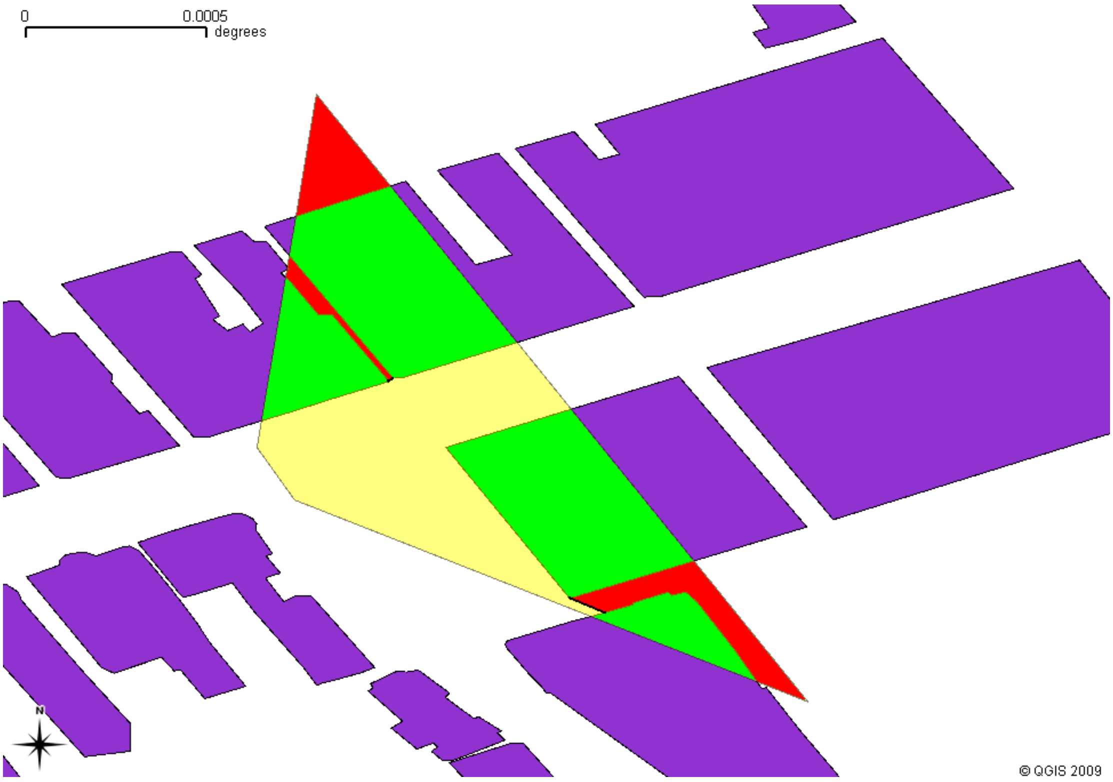


Fig. 7.15. Single Viewpoint Viewshed analysis operational requirements. The algorithm is developed to calculate the yellow polygon region based on intersections with the purple buildings layer that will exclude the green and red obstacle space.

The Viewshed algorithm deconstructs each Viewpoint ViewCone into definably spaced lines that represent lines-of-sight from the base near depth-of-field to the far depth-of-field. The algorithm in appendix ten has this spacing set at one hundred scan lines-of-sight, constructed from left to right for each processing ViewCone. The first point of intersection between each scan line and the obstacle's objects is recorded and all such points are then converted to a Viewshed polygon on completion of all scan line processing. The processing of this operation involved adjusting 15984 Viewpoints that intersected with the buildings layer and took over 39 hours of computation.

However, it is recognised that this implementation is not optimal and processing time could be significantly reduced. Fig 7.16 shows the results of this operation where the green coverage represents the resultant viewable regions compared to the original full Viewpoints in red with the heavy black outline. The Viewshed obstacles are based on the light blue building footprints and the NUIM college entrance wall.

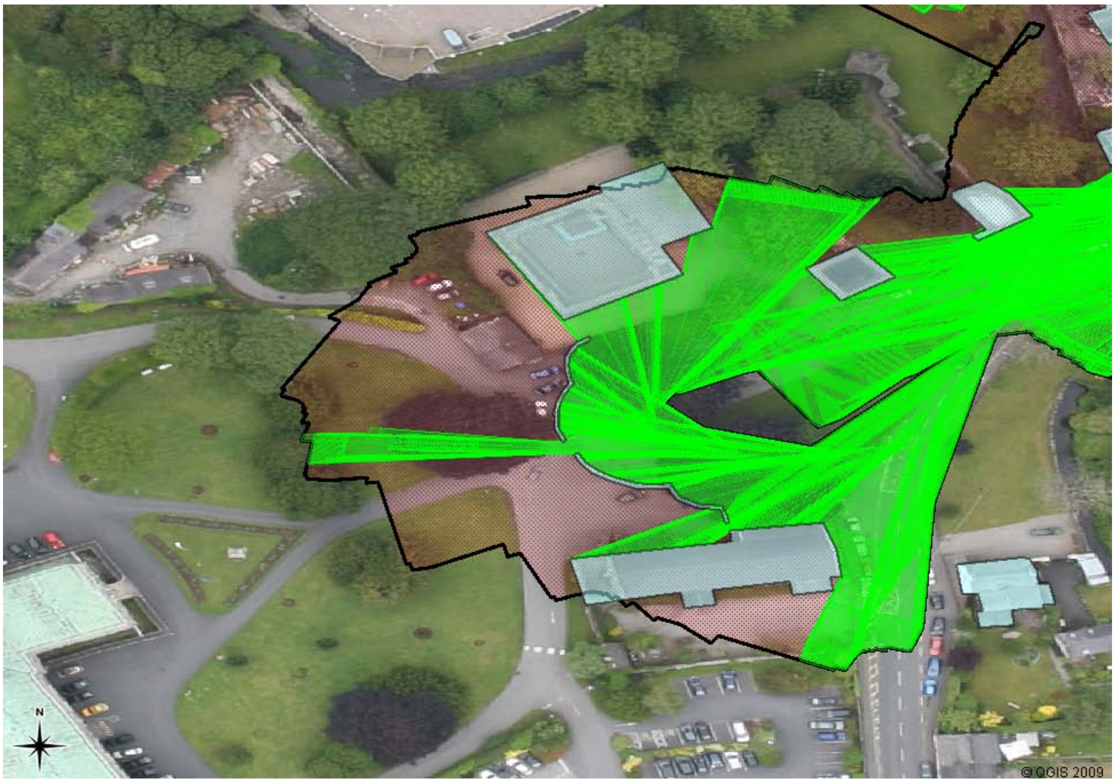


Fig. 7.16. Viewpoint Viewshed processing results where the original Viewpoint assumed geographical space is shown in red with a heavy black outline. The Viewshed obstacle layer is shown in light blue and incorporates a number of buildings and boundary wall. The light green represents the results where all the original Viewpoints have been recomputed based on the Viewshed visibility analysis.

Implementation of this operation has demonstrated the flexibility and extensibility characteristics of the Viewpoints model by facilitating dynamic generation of altered ViewCones based on the spatial intersections of line-of-sight lines with non-video spatial obstacles data. This enables more accurate representations of the geographical space captured in each video frame to be modelled.

## 7.5 Location Aware Video Player

This section presents an alternative context-aware view of video file playback. The context in this case is location, where a GPS coordinate can be used to manage the video footage in a spatially aware manner. This is facilitated through the use of the video access software introduced in section 7.4.1, the Spatial Video Viewpoints database from chapter six and certain aspects of the spatial operations described in this chapter. A simulated experiment is presented where GPS routes are used to control the video playback based on the route's current location data. This involves being able to dynamically shift between different video survey streams and to different points within a video stream based on the location information being received by the player. Two routes were collected using a different GPS device from that used when collecting the Spatial Video survey data, but set on similar settings where the spatial data were collected at one point per second. The collected routes are shown in fig 7.17 and cover a number of base case situations when testing dynamic control of the spatial player. These situations include some sections of the various simulated tracks having the following characteristics:

- Tracks should cross Spatial Video survey sections with multiple capture streams. This requires the player to differentiate current player stream data to maintain consistency in the viewing stream.
- Tracks should cover network area not captured in any survey data. Thus the player has to determine when stream data is unavailable and adjust appropriately.
- Tracks should encompass routes where video stream switching is required. Thus the player has to access and load another stream dynamically.
- Tracks should have sections where Spatial Video network data are queried in both directions. Thus the player has to determine the simulated track and survey video orientations and adjust to play the video stream in the correct direction.

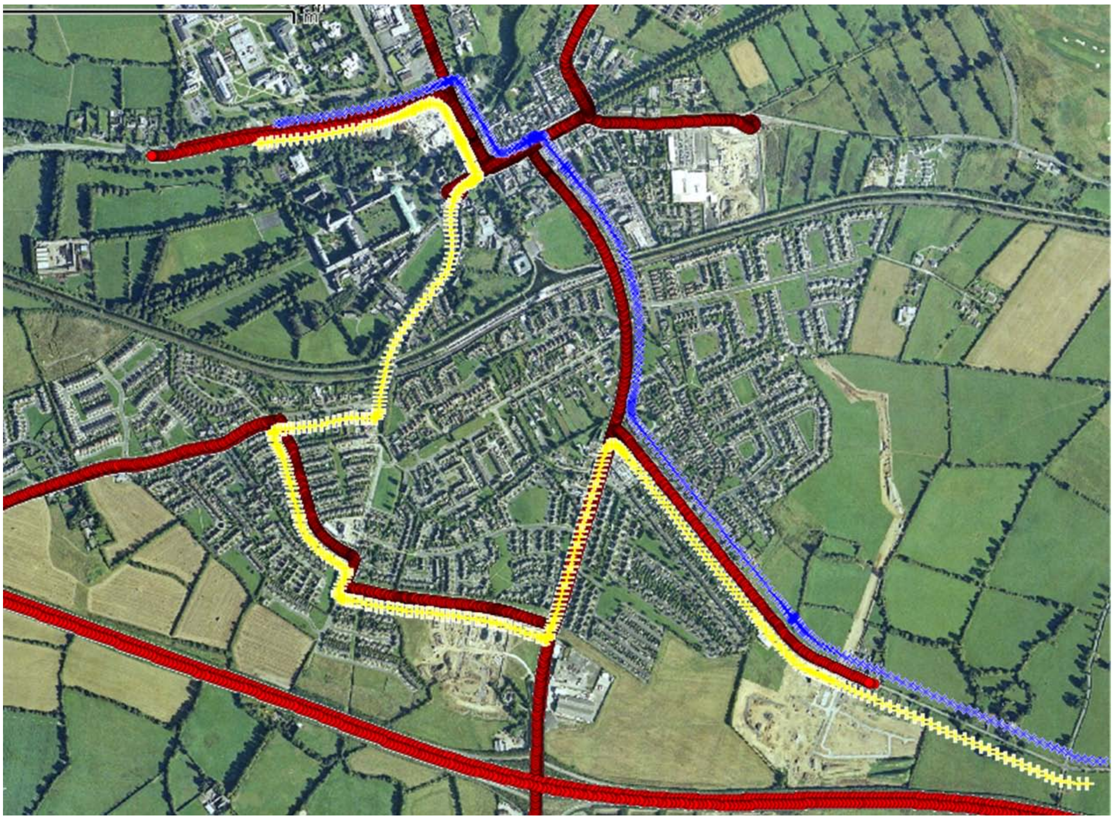


Fig. 7.17. Spatial Video Player experimental location control tracks. Track one is detailed in light blue while track two is yellow. Both tracks are slightly offset for easier visualisation.

The location aware video player was developed as an extension to the existing video playing and analysis tool previously developed. The laboratory experiment was designed to take as input a simulation route track data file and begin searching the Spatial Video Viewpoints database for footage relevant to the search space. This was achieved by using a similar point-of-view search method as that used in section 7.4.2.2 using SQL query 7.5. The simulated track route orientation was calculated between every two track points and the Viewpoints database searched for a Spatial Video file to load and a frame index to start playing from. Once a video file and stream point had been established, the video was loaded and began playing in the viewer.

From that point on every subsequent simulated track point was passed to the algorithm at approximately one second intervals to mimic the GPS points being polled from a real GPS input unit. The one second interval is approximate because it was defined ambiguously. Empirical knowledge of GPS unit polling is that it cannot be relied on to accurately output at this exact rate. While the GPS time, unit time or player system time could all be used to generate accurate measures, this implementation needed the player to deal with the location information as soon as it gets it and adjust the video stream accordingly. This approach led to a spatial displacement problem where the video was being played with no rendering speed controls.

The player would correctly spatially re-locate the relevant video file and stream based on the input track simulation data; however it could jump, either forward or backward, with noticeable spatial displacement because the video was being played from the last search point with no time or distance controls. Thus, on a fast processing system the video would render frames faster than a slower system and could, virtually, move spatially faster through the video footage than the velocity of the captured track simulation data was determining. In this case the video stream would be reset to a previous frame from that being currently viewed resulting in some frames being replayed. The opposite was also possible where the video could spatially fast forward to catch up on the next track data point, which would also cause a distinguishable change in the video's spatial context.

Also, the inherent GPS polling and Spatial Video errors were sources of various problems in the accuracy of the video playback and in the Viewpoint search operations. A generous search range in both orientation; with a range of +/- 6°, and buffer distance; with a range of 8 metres, had to be allocated in order that a suitable Viewpoint index could be established for the player. It also required careful ordering by stream file source as visual continuity in the video streams could be affected by regular switching between different survey video sources.

This problem was also present in another form where unavoidable switching between stream sources was required, i.e. only one database video stream is available for the next search section. This resulted in issues of visual disparity in switching from one video stream to another. Situations developed where occlusions suddenly appeared from one frame to the next or the video footage would suddenly go from day to night or clear to raining. However, the results were generally satisfactory and created a correctly processed location aware stream of video footage from multiple points within files and over different file sources and formats. Improvements could certainly be applied through an implementation of a velocity consideration where the video does not just play as fast as the system can handle, but plays at a rate that reflects the query GPS track route.

### 7.6 Conclusions

The requirements of this chapter laid out objectives to demonstrate how a Spatial Video Viewpoints model could be used to generalise and define its interaction in a standardised GIS functionality scenario. Using the Viewpoints database as the base query source and the PostGIS spatial extensions as the processing environment it was possible to develop a number of bespoke query routines. This process involved sequential stages where the discussions about the nature of the problem each operation was attempting to demonstrate, determined the correct approach, implemented a suitable algorithm or query and analysed the results. Ultimately the list of operations that were implemented is only a subset of many possible queries or questions that could be applied to a Viewpoints database for analysis or problem-solving, through spatial data approaches to video access, analysis and profiling.

Compiling and implementing an authoritative list of all Spatial Video query implementations is not reasonably possible for a single thesis chapter. However, what has been achieved is to define a summary overview of the most relevant data specific operations. This involved using all types of 2D primitive spatial data types; points, lines and polygons; to formulate a number of queries that interact with the Spatial Video based on spatial and/or video objectives. This showed how standard spatial operations can be applied, with minimum alteration, to fit the objectives requirements and determine results that are reasonable, based on the visual context of the video.

One significant aspect of the Viewpoints model that has not been discussed in detail is spatial buffering which, when it has been used has been applied carefully. This is because a Viewpoint representation of a video frame is not a normal dynamic spatial model but a visually constrained one. The Viewpoint represents a geographical space of an image; therefore, performing a query with a buffer on the Viewpoint that extends this space needs to be carefully considered. The visual content of the image cannot change; therefore a buffer on its Viewpoint is irrelevant. Artificially creating a larger extent Viewpoint to search for captured space would, obviously, be ineffective as the video frame that captures this space will not reflect this buffer. Space defined to exist in the buffer space will not appear in the video frame.

Also, using a buffer on a query object could produce the same irrelevant video frame results. As an example, in a point-in-view operation, buffering the point to query Viewpoints that contain this point would likely produce inaccurate results. As a result it was necessary to ensure that the point-in-view was defined as being contained within the Viewpoints ViewCone. The far depth-of-field case is the only context situation where both of these buffer operations would work. Here a Viewpoint could be forward buffered or the search space is buffered to extend into the Viewpoint forward captured space. This is because it has already been defined, and discussed, that each Viewpoint has had a far depth-of-field limit applied for practical reasons. In some of the operations performed in this chapter the result frames that represent a successful query are nearly always approximations as a number of the previous video frames could also be included. They may capture the search space effectively, however not to the same visual detail, resolution or scale as the result frames; highlighting the reason for the far depth-of-field inclusion.

Finally, described in this chapter are a large number of low level spatial query routines that could be easily incorporated into a more user friendly application environment. This could form the basis of a generalised Spatial Video handling and interaction system that could enable dynamic processing and analysis of this data from a number of spatial and visual perspectives. In this thesis a number of disparate data handling and processing applications were used to produce the results of these operations and it is recognised that a single Spatial Video modelling and interaction environment does not currently exist.

# Chapter Eight: Conclusions

This chapter discusses the main conclusions of this research by summarising the work completed, by discussing how the research questions have been answered and by detailing some future directions for extending this research area. The contributions to knowledge achieved through the application of a Spatial Video Viewpoint model are discussed, followed by some final remarks.

## 8.1 Summary of Work

A number of Spatial Video surveys are the basic data source for this project, although it has been the implementation of a generalised GIS-based model that has formed the context for much of the work. Specifically, the Spatial Video used for this process has been retrospective data sets captured at a near-orthogonal orientation to the terrestrial surface, which has been traversed by a road network survey vehicle. The modelling process started with a multitude of existing Spatial Video surveys and the basic data capturing equipment; this began an examination of the possible opportunities and/or improvements that could be developed to utilise these data in a GIS-constrained framework.

Two broad areas were initially considered as possible approaches for storing a GIS data-type-constrained model of Spatial Video. These are internally and externally generated video-frame spatial-data stream indexes. It was quickly realised that an internally indexed method was not currently feasible as video storage formats and GIS software systems have no well defined and developed Spatial Video specific frameworks. In a video context, spatial data are not defined, while in a GIS context video has no fundamental support.

Existing Spatial Video systems were discussed with broad reference to their integration within a GIS. In all cases, bespoke GIS software systems exist to handle the respective Spatial Video data streams; there also exist software extensions to a number of the popular GIS platforms which can also handle these data sources in a bespoke manner. However, as their bespoke or extension software profile demonstrates, Spatial Video is not intrinsically integrated into GIS. Vector and raster data formats are well defined in any GIS, yet video, which is just many raster images stored in a specific file format, cannot be handled within a GIS. Raster operations can only be performed following a video frame grabbing operation, vector data cannot be overlaid on the video stream and video frames cannot be ortho-corrected while in a video format, to emphasize but a few of the shortcomings of current GIS. These points highlight a number of future research directions in this area as raster support for video in a GIS has not been described in this thesis. This research concentrates on the vector data representation and operations domain of the video's spatial data characteristics.

Existing Spatial Video processes that encompass operations from data collection through to analysis have been discussed where it has been shown that current software systems have generally developed for bespoke application specific areas. As such, Spatial Video data generalisation and interoperability across multiple GIS has not been well defined or discussed in literature. The context for existing uses of Spatial Video has always been video-centric where the visual properties enhance the GIS environment with playback functionality that supports some basic GIS interactivity. However, all this is achieved with minimal use of the captured spatial data. The approach developed here considers extensions in the spatial domain where the definable geographical space captured in each video image is central. Thus, this research is approached from a GIS perspective were the Viewpoints model is primarily based on the Spatial Video spatial data. It also considers alternative sources and methods of spatial data interaction with the Viewpoints model, while being constrained to a well defined GIS software environment.

Thus, the existing internal indexing models of Spatial Video are substituted for a new and novel centralised database modelling approach where a GIS data structure for Spatial Video can be more easily implement in a standardised way. To achieve this, a Viewpoint construct is defined and developed to represent the core spatial properties of the survey video. Theoretically the Viewpoint was defined in a 3D context but experimentally implemented in a 2D form. A number of problems developed based on this work relating to frame and location point synchronisation, interpolating location points for higher frame rate video, correcting the GPS track data to more correctly represent the video's survey route and detecting changes to the video capture parameters. None of the problems was solved with a definitive solution; however the inaccuracy of the model and the data being modelled facilitated an implementation that approximated acceptable results.

Further extending this into the operational domain a number of algorithms and tools were developed to demonstrate the broad flexibility of the Spatial Video Viewpoint model to many different geo-spatial analysis operations. This included defining how the operations should be generated, the results considered and the SQL queries or scripts composed. These example operations were finalised with some work that could generate more accurate Viewpoint geographical space representations using a bespoke Viewshed line-of-sight analysis algorithm. This research showcased how some of these examples could be used in a practical situation by developing a location-aware video player. The player could play back video based on a regularly updated stream of GPS locations dynamically altering the video files being streamed.

## 8.2 Main Contributions to Knowledge

Discussed in this section are the main contributions to knowledge that the implemented solutions have achieved based on the objectives stated at the outset of this research project. Each objective is briefly discussed in terms of its implementation suitability and feasibility as in some cases the solutions that were defined were not necessarily optimal.

### 8.2.1 Indexing Video with Spatial Data

Video storage formats currently do not have standardised metadata definitions that specifically support spatial data. Although this can be done through existing video data structure metadata or encoding procedures, it is a format-specific and bespoke solution to a problem that needs to be generalised. It is argued that standardisation of spatial data within the core video storage data structures as part of future standards development is necessary. This could be similar to the standards defined for spatial data in photographic EXIF data, (Metadata Working Group, 2009), however video-specific considerations would need detailed thought.

Existing formats for the storage of spatial data in or with a video file use bespoke methods that include logging GPS data in the audio stream, using subtitling formats or writing to SMPTE Key Length Value (KLV) metadata specifications. Audio formats are where the camcorder audio input jack is connected to a GPS encoder. This format is not frame-index specific and will usually require post-processing to define the index and improve accuracy. As such its use is widespread and cost effective as an inaccurate consumer-ready and affordable system that can be implemented easily. However it was discussed that this method was not feasible for the requirements of this research as access to the spatial data required dynamic processing of the video streams.

Subtitling specifications facilitate an easily constructed video stream metadata format that is also well supported in visually enhanced players where the location information can be overlaid easily in the video window. However this is not necessarily an internal indexing method as some formats define an associated video file system as opposed to internal indexing, although internal indexing does exist for certain formats. Not all video players support all subtitling formats and using this format to store frame-level indexing would be counter-productive from the perspective of visualisation of the subtitles. This is because frame rates of twenty five frames per second would make the viewing of location information impractical.

A more complex, and expensive, system that has been implemented on UAVs by the United States Department of Defence through Intergraph Corporation encodes the spatial data as KLV metadata variables in the video capture stream. This has required a very well calibrated system specification to be defined where electronic signal propagation delays and equipment accuracy are all determined before surveys are commenced. This project has broadly taken elements from this work in the form of the OGC standards ViewCone data structure storage definitions and used this as a base to develop the ideas on the retrospective data that are available. However, the difficulty in using standard ViewCones in this case is related to the practical implications of the data source capture domain and its retrospective nature. System calibrations can only be assumed and/or approximated because accurate survey setup information is either unknown or has never been defined, especially on lower accuracy consumer systems, which would be required to accurately implement the ViewCone data structure.

Ultimately, this research determined that internal indexing of video with spatial data is completely unsuitable for the overall objectives. It is shown that no standards or generalised frameworks exist that support any interoperability of spatially indexed video files in a cross platform, cross format or GIS-enabled manner. Thus, the project defined and developed a centralised, post-processed, spatially defined indexing as the most effective methodology for completing the objectives of the thesis.

### 8.2.2 Decoding Audio Indexed Spatial Video Streams

As part of the research into existing Spatial Video indexing methods a software decoder was developed for a specific spatial data audio encoding device. Typically, the encoding hardware is required to decode the spatial data in tandem with a video frame-grabbing card; however as this project progressed this decoding functionality was developed in software to ease the impractical requirements of dealing with image list directories and associated spatial data files. This improved the overall usability of the Spatial Video surveys captured with this device as no video post-processing would be required and spatial data could be determined without the hardware encoder. The software decoder was shown to work effectively and more robustly than the hardware version as extra spatial records could be retrieved based on the hardware synchronisation steps not being required. However, its software implementation could be improved as decoding the spatial data required an intermediate process of creating a sound file containing the encoded spatial data. This separate file generation is unnecessary as it is recognised that this step could be removed and a dedicated spatial data decoding algorithm developed to access the video audio stream.

### 8.2.3 Theoretically Extending ViewCones to Viewpoints

Based on the unsuitability of the internal indexing methods, an alternative theory was defined the developed that centred on an external indexing approach. This facilitated a shift to using well established GIS-based software platforms to define, develop, implement and demonstrate a centralised indexing system that could be used to study Spatial Video. The core idea was to generate a video frame level GIS data type representation of each Spatial Video survey stream where the index is centralised in a spatial database. This can then be used to directly access any of the video streams, in different formats, across distributed locations, to any frame-index access point. It was also important that this GIS data type representation should support spatial operations in a very broad context. This was achieved by theoretically defining a 3D model approach where a data structure was developed called a Viewpoint.

A Viewpoint is a theoretical 3D extension of the OGC 2D ViewCone structure as implemented in their Geo-Video service, (OGC OWS-3, 2005). The key differences in the Viewpoint model relate to the smaller geographical scales captured in the surveys and the oblique nature of the image content where objects are captured, in detail, throughout the depth-of-field range. Thus, it is argued that the OGC data structure, where the capture point defines the origin of a viewable polygon region, should be changed to represent the terrestrial nature of the Spatial Video data characteristics. This change consists of a point that defines the capture location but does not define the viewable extent origin. The ViewCone should be calculated as a disconnected polyhedron that represents the captured geographical space with a set of field-of-view extent planes and a near and far depth-of-field plane.

This construct is introduced in a minimal form of complexity where the maximum spatial extent of the video-frame image capture space could be defined based on some easily determined survey system parameters. Significant practical difficulties are recognised in the determination of an accurate Viewpoint model which is based on an oblique terrestrial orientation of the Spatial Video data. To develop higher accuracy models a number of cross discipline investigations are required, from areas such as photogrammetric image analysis to acquisition of more detailed terrestrial spatial data from alternative sources such as LIDAR.

### 8.2.4 2D Viewpoint Implementation

Based on this theoretical Viewpoint model the methodological approach is defined for implementing this structure in 2D. Given the spatial uncertainty inherent in the retrospective data being used, a process was devised based on the minimum set of knowable Spatial Video survey parameters from a past survey. This involved at least knowing the model or type of video capture device used in the Spatial Video survey. From this the operational parameters are acquired, such as the size of the digital image sensor and lens focal range parameters. Coupling this information with the already available spatial data, and photogrammetric and geodetic formulas, a maximum viewable extend polyhedron could be calculated for a 3D model.

In this experimental implementation it was decided to test this hypothesis in a 2D planar model. This decision simplified the calculations without sacrificing the ability to prove the potential geo-spatial analysis functionalities of the theory. The process was evaluated in two separate experiments where Viewpoints were generated for a sample set of retrospective Spatial Video survey images, but also for a calibrated set of camera images captured in controlled conditions. In the controlled image experiments it was shown that the model results were at best only able to define a fuzzy region boundary of half metre accuracy, on average. In the known Spatial Video experiments it was shown that numerous supervised calibrations of the Viewpoint structure were required to reach a visual parity with the video frame. This was done within the operational parameter ranges of the camera and was only performed on random video stream image samples to, once again, only define a fuzzy boundary.

### 8.2.5 GIS Database Modelling of the Viewpoint Data Structure

Based on these experimental Viewpoint implementation tests a spatial database could then be developed that defines each Spatial Video’s Viewpoint for every frame in the survey file. This was developed as an unsupervised processing system where initial parameters and synchronisations were defined from which all subsequent video frames could be indexed with a spatial database Viewpoint. Some problems were identified in the development of this process with the inaccuracy of the GPS track data from the retrospective systems. Initially this could be accounted for with a measured parameter inclusion, however calculating this parameter in an accurate manner was not defined in detail. As location accuracy errors were common, but not necessarily systematic, a method was not determined to handle these errors over the course of a survey stream. Particularly, any sequences of stationary video footage generated significant problems for the unsupervised Viewpoint calculation algorithm where the overall system accuracy suffered.

Also, the capture frequencies of the video and spatial data had to be reconciled as they generated their data frames at different rates. This required GPS point interpolation to be applied to the track data at a video frame rate. A number of techniques are implemented that included both bespoke and traditional interpolation algorithms; however no single algorithm provided a satisfactory and correctly representative result. The final implementation settled on a combination of an approximating spline and GPS drift averaging to generate a satisfactory survey track representation at a video frame rate.

Based on this software development and GPS error adjustment process a PostGIS spatial database was populated with approximately 75,000 Viewpoints, representing 46 minutes of Spatial Video from four different surveys of Maynooth town and its environs. The PostGIS-based spatial extensions to the PostgreSQL database were chosen for its extensive spatial support of OGC standards, but also for its open-source paradigm. Thus, the objectives could be maintained by defining a novel Spatial Video GIS modelling approach while constraining this model to standardised GIS data structures and analysis operations.

### 8.2.6 Spatial Video GIS Analysis Queries

Following from the development of the Viewpoints database the feasibility was demonstrated of querying this data source in a GIS framework for both spatial and video-orientated reasons. This work has extended the knowledge that can be acquired from a range of different geo-spatial analysis techniques over and above that which is currently possible in existing systems. Both the point and polygon elements of the Viewpoint data structure are used in an implicit restriction relationship where interaction with other spatial data sources, in a number of operational scenarios, highlighted the unique nature of the Spatial Video index representation. An increased range of geo-spatial analysis functionalities was also looked at from both the video and spatial perspective. These functionalities could determine individual video frames or complete sequences appropriate to the query. Alternatively, a spatial perspective could determine, in a generalised way, the volumes of space captured in video and the amount of video captured by spatial content. This process was based on an overview of sample operational functionality as many other possible query scenarios could be generated and performed on the Viewpoint data structures.

### 8.2.7 Viewpoint refinement based on non-video Spatial Data Queries

To complete this objective a bespoke vector based Viewshed algorithm was developed to calculate the viewable spatial difference of a Viewpoints extent region and any obstacles that intersect it, determined from an alternative spatial data source. In this case a buildings footprint vector data set of the Maynooth town centre buildings was used. This operation highlighted the relative flexibility in the Viewpoints design, where its accuracy can be improved based on a number of operational sources, without resorting to the original video footage content in an image analysis approach. As has already been mentioned, image analysis should probably play a role in this process; however it is certainly plausible to develop this spatial approach. Almost any spatial data set could be used in this intersection approach; alternatively, higher accuracy spatial data sources such as different fixed inventory models and sonar or LIDAR scans could be used to develop bespoke Viewpoint accuracy algorithms to calculate even better Viewpoint representations of the geographical space.

### 8.2.8 Dynamic Spatial Video Player

Finally, a location-aware video player highlighted the work performed in developing the various video processing and access tools implemented during the project. While most video players have functionalities that allow dynamic timeline access, they usually do not define this for dynamic file access as well. This Spatial Video player has implemented a novel approach where the video is played based on a dynamically acquired spatial control. As the player's location information changes it will query the Viewpoints database and alter the video playback to account for different video stream access frames or even different file sources. This system is a demonstration of how the Viewpoints data structure can be used to simulate a popular location based service system such as an in car satellite navigation device. Unfortunately the approach taken here was modelled on existing navigation systems were the map display updates on a change in spatial location. However, in the case of video playback a further video playback control would need to be implemented as the velocity captured in the stored video stream is not guaranteed to be the same as that of the player's requirements. In this implementation the video playback was either being advanced or rewound on dynamic recalculation of the player's location and the access point in the video stream.

### 8.3 Future Research Directions

Based on the work described in this thesis a large number of future research directions could be considered. In general they all relate to improving the accuracy of the various components described in this thesis. To begin with the ideal future work approach would be to define, develop and test a fully calibrated system for the capture of Spatial Video. This would require incorporating the spatial data to known and adjusted accuracies, controlling signal propagation delays that are accounted for in integrated field systems, adding various spatial data sensors to provide enhanced accuracy and standardising methods of computation, storage and use of the resultant data, all in a GIS framework. However, this option would still not answer the question of how can retrospective Spatial Video be used in the same GIS context?

While the answer to this question has been achieved in this thesis, by developing the Viewpoint model, it has only been done at an inaccurate level. This has resulted in a subtle distinction between what sort of operational approach would determine more accurate results. Based on the Viewpoints implementation, accuracy of the results is probably better when determining what does not define a successful query. For example, a point-of-interest search of the Viewpoints database cannot guarantee that the results will visually contain the object of interest. However, it can determine to a higher degree of accuracy what Viewpoints do not contain the query point-of-interest, although this is only a consequence of the models inaccurate implementation. Thus, future work would largely consist of defining more accurate techniques to model Viewpoints.

Other future research work would include extending this model into a full 3D context. The challenges here would be non-trivial as 3D supports in GIS are still only developing in both research and commercial fields. Thus, capture, storage, display and analysis areas would all comprise significant areas of research in themselves. Even in the 2D model more complex approaches that incorporate ellipsoid intersection modelling, image analysis techniques, GPS post-processing accuracy improvement; could all be included to better define the Viewpoints boundary planes, frame relationships and spatial locations.

As has already been mentioned, internal Spatial Video file indexing has a completely undefined future. No emerging standards currently exist for video file format indexing of individual frame or even sequences of frames with various types of spatial data. However this work has to emerge in the future as consumer available spatial video recorders, similar to existing camera systems, will eventually be developed. This will probably drive such standards development, but hopefully not from the current situation where multiple bespoke implementations already exist but from a thoroughly researched requirements definition perspective.

Finally, research from areas like photogrammetric image analysis should play a more significant role as this has not been incorporated in the Viewpoint calculation model. Sample work from models developed by Pollefeys (2004) shows how objects captured in a video sequence can be generated in a virtual reality context. However, the spatial extent of these computed models is not defined in this work. Also, work by Liyuan (2003) shows how foreground detection can be achieved in video sequences which could allow more accurate calculations of near depth-of-field. However this work has very constrained video models as opposed to the dynamic Spatial Video data available here. This work is also missing a spatial context that would enable accurate measurements of the spatial context of the detected foregrounds.

## 8.4 Final Remarks

In general the broader context of spatial data is well understood from a GIS point of view; for example, census data are recorded at very detailed spatial levels and aggregated to represent larger spatial domains, while geographical features like rivers, roads or mountains have well developed GIS spatial data representations. Developing research is pushing these well-established models to newer levels of higher dimensionality and complexity where our understandings of the spatial data and their associated processes are becoming more intuitive and interactive. However, reinventing any of these models and processes would be counter-productive unless a well-defined context was established. Thus, this research approach has been to generate a GIS based data model for a bespoke spatial data source, Spatial Video. This has been done through the Viewpoint construct which is grounded in the standard primitive GIS spatial data types.

Realistically, this project and the content of this thesis have been concerned with a very broad application area. This has led to a level of model inaccuracies that could be significantly improved given a smaller, more targeted, set of objectives. Yet, what has been defined and developed was determined by an approach that is searching for the best methods of integrating a visually enriched GIS data source to a higher degree of spatial use. Perhaps this work can form the basis for future projects where detailed research can develop higher accuracy implementations of the model. Spatial Video in any form should have a more inclusive role in GIS, but the scarcity of research into its uses is significantly lacking to date.

# Bibliography

**A9 Maps,** (2006), Business shop front spatial image location and direction system, Internet Source, http://maps.a9.com/?

**Adams G.,** (2006), Timed Text (TT) Authoring Format 1.0 - Distribution Format Exchange Profile (DFXP), W3C World Wide Web Consortium, Internet Source, http://www.w3.org/TR/ttaf1-dfxp/.

**Albrecht J.,** (1996), *Universal GIS Operations for Environmental Modeling*, Third International Conference/Workshop on Integrating GIS and Environmental Modeling, Santa Fe, New Mexico, USA,

**Albrecht J.,** (1997), *Universal Analytical GIS Operations - a task-oriented systematization of data structure-independent GIS functionality*, Geographic Information Research: Transatlantic perspectives, pp.577-591.

**Ardizzone E.,Cascia M. L.,** (1997), *Automatic Video Database Indexing and Retrieval*, Multimedia Tools and Applications, 4, 1, pp.29-56.

**Auclair Fortier M. F., Ziou D., Armenakis C.,Wang S.,** (2000), Survey of Work on Road Extraction in Aerial and Satellite Images, Departement de mathematiques et d'informatique, Universitede Sherbrooke, Technical Report, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.13.9298&rep=rep1 &type=pdf

**Avid,** (2007), Avid Xpress Pro, Avid Technology Inc., Internet Source, http://www.avid.com/products/xpress-pro/index.asp.

**Benedikt M. L.,** (1979), *To take hold of space: isovists and isovist fields*, Environment and Planning B, 6, pp.47-65.

**Bockaert V.,** (2008), Sensor Sizes, Internet Source, http://www.dpreview.com/learn/?/key=sensor%20sizes.

**Bormans J.,Hill K.,** (2002), MPEG-21 Overview v.5, ORGANISATION INTERNATIONALE DE NORMALISATION, Internet Source, http://www.chiariglione.org/mpeg/standards/mpeg-21/mpeg-21.htm.

**Bosi M.,Wilson S.,** (2003), WAVE PCM soundfile format, Stanford, Internet Source, http://ccrma.stanford.edu/courses/422/projects/WaveFormat/.

**Bray C.,** (2006), The OSi National Network RTK Solution, Ordnance Survey Ireland, Internet Source, http://www.irishscientist.ie/2006/contents.asp?contentxml=06p32.xml&contentxsl=is06pages.xsl.

**Cho P.,** (2007), *3D Organization of 2D Urban Imagery,* Applied Imagery Pattern Recognition Workshop, 2007. AIPR 2007. 36th IEEE, pp.3-8.

**CommLinx Solutions,** (2003), GPS NMEA sentences, Internet Source, http://www.commlinx.com.au/NMEA_sentences.htm.

**Conrad J.,** (2006), Depth of Field in Depth, Technical Report, http://www.largeformatphotography.info/articles/DoFinDepth.pdf

**De Floriani L., Magillo P.,Puppo E.,** (1999), *Applications of Computational Geometry to Geographic Information Systems*, pp.333-388, Elsevier Science,

**Derr L.,** (1906), *Photography for Students of Physics and Chemistry*, pp.76, The Macmillan Company,

**Douglas D. H.,Peuker T. K.,** (1973), *ALGORITHMS FOR THE REDUCTION OF THE NUMBER OF POINTS REQUIRED TO REPRESENT A DIGITIZED LINE OR ITS CARICATURE*, Cartographica: The International Journal for Geographic Information and Geovisualization, 10, 2, pp.112-122.

**Egenhofer M. J.,Franzosa R. D.,** (1991), *Point-Set Topological Spatial Relations*, International Journal for Geographical Information Systems, 5, 2, pp.161-174.

**Egenhofer M. J., Sharma J.,Mark D. M.,** (1993), *A Critical Comparison of the 4-Intersection and 9- Intersection Models for Spatial Relations: Formal*

*Analysis.*, Eleventh International Symposium on Computer-Assisted Cartography, (Auto Carto 11), Minneapolis, Minnesota, USA, pp.1-11.

**ESRI,** (1998), ESRI Shapefile Technical Description, Environmental Systems Research Institute, Inc., Technical Report, www.esri.com/library/whitepapers/pdfs/shapefile.pdf

**ESRI,** (2003), Spatial Data Standards and GIS Interoperability, White Paper,

**ESRI,** (2007), ArcGIS 9.2 Viewshed Analysis, ESRI, Internet Source, http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?TopicName=Performing_a_viewshed_analysis.

**Evens L.,** (2003), Some Thoughts on View Camera Calculations, Northwestern University, Technical Report, http://www.math.northwestern.edu/~len/photos/

**FFmpeg,** (2007), FFmpeg - programming API that can convert, record and stream digital audio and video in numerous formats., Fabrice Bellard, Internet Source, http://ffmpeg.org/.

**Fisher P. F.,** (1999), *Probable and fuzzy models of the viewshed operation*, pp.161-176, Taylor&Francis,

**Foy S., Deegan C., Mulvihill C., Fitzgerald C., Markham C.,McLoughlin S. D.,** (2007), *Inertial Navigation Sensor and GPS integrated for Mobile mapping*, GISRUK 2007, NUIM, Maynooth, Ireland, pp.58-65.

**Gavaghan M.,** (2007), Geodetic Curve Distance (Vincentys) Formula in C#, Mike Gavaghan, Internet Source, http://www.gavaghan.org/blog/2007/08/06/c-gps-receivers-and-geocaching-vincentys-formula/.

**Geddes C.,** (2005), A Brief Introduction to Geographic Data, Institution for Fisheries research, Technical Report, http://www.glfc.org/glgis/support_docs/html/GIS_help/pdf/geographic_data_primer.pdf

**Geo Community,** (2006), GIS Data Formats, Internet Source, http://data.geocomm.com/helpdesk/formats.html.

**Google,** (2007), Google Street View, Google, Internet Source, http://maps.google.com.

**GRASS,** (1982), Geographic resources Analysis Support System, Open Source Software Development, Internet Source, http://grass.itc.it/grass62/manuals/html62_user/r.los.html.

**Hartley R.,Zisserman A.,** ( 2003), *Multiple View Geometry in Computer Vision*, Cambridge,

**Herring J. R.,** (2006), Implementation Specification for Geographic information - Simple feature access - Part 1: Common architecture, Open Geospatial Consortium, Technical Report, http://portal.opengeospatial.org/files/?artifact_id=18241

**Hillier B., Leaman A., Stansall P.,Bedford M.,** (1976), *Space syntax*, Environment and Planning B, 3, pp.147-185.

**Hirose M., Watanabe S.,Endo T.,** (1998), *Generation of wide-range virtual spaces using photographic images*, Virtual Reality Annual International Symposium, 1998. Proceedings IEEE 1998, Atlanta, GA, USA, pp.234-241.

**IEEE Long Island,** (2005), ASCII Code Table, IEEE, Internet Source, http://www.ieee.li/computer/ascii.htm.

**Immersive Media,** (2006), Immersive Video GIS Viewer, Immersive Media, Internet Source, http://www.immersivemedia.com/#102.

**ISO/IEC,** (2006), International Standards Organisation and the International Electrotechnical Commission, Internet Source, http://www.standardsinfo.net/info.

**Ive J.,** (2004), Image formats for HDTV, European Broadcasting Union, Technical Report, http://www.ebu.ch/en/technical/trev/trev_home.html

**Jiang H., Helal A., Elmagarmid A. K.,Joshi A.,** (1998), *Scene change detection techniques for video database systems*, Multimedia Systems, 6, 3, pp.186-195.

**Joung Y., Kim K.,Kang K.,** (2003), *The Development of MPEG-7 Interface over MPEG-4*, Consumer Electronics, 2003. ICCE. 2003 IEEE International Conference, pp.276- 277.

**Jupp D. L. B.,** (1978), *Approximation to Data by Splines with Free Knots*, SIAM Journal on Numerical Analysis, 15, 2, pp.328-343.

**Kawasaki H., Yatabe T., Ikeuchi K.,Sakauchi M.,** (1999), *Automatic modeling of a 3D city map from real-world video*, International Multimedia Conference, Orlando, Florida, United States, Proceedings of the seventh ACM international conference on Multimedia (Part 1), pp.11-18.

**Kerr D. A.,** (2006A), Field of View in Photography, Technical Report, http://doug.kerr.home.att.net/pumpkin/#Optics

**Kerr D. A.,** (2006B), Depth of Field in Film and Digital Cameras, Technical Report, http://doug.kerr.home.att.net/pumpkin

**Kerr D. A.,** (2008), Describing Digital Camera Sensor Sizes, Technical Report, http://doug.kerr.home.att.net/pumpkin/

**Khuan C. T., Abdul Rahman A.,Zlatanova S.,** (2007), *3D Solids and Their Management in DBMS*, pp.279-311, Springer,

**Kiger B.,** (1972), *Videography, What Does It All Mean?*, pp.9.

**Kraak M. J.,Ormeling F.,** (2003), *Definition Of Cartography*, Second Edition, pp.36-38, Pearson Education Limited, Harlow.

**Kyong-Ho K., Sung-Soo K., Sung-Ho L., Jong-Hyun P.,Jong-Hun L.,** (2003A), *The interactive geographic video*, Geoscience and Remote Sensing Symposium, 2003. IGARSS '03. Proceedings. 2003 IEEE International, Toulouse, France, 1, pp.59-61.

**Kyong-Ho K., Sung-Soo K., Sung-Ho L., Kyung-Ok K.,Jong-Hun L.,** (2003B), *GEOVIDEO: The Video Geographic Information System as a first step toward MEDIAGIS*, ASPRS 2003 Annual Conference Proceedings, Anchorage, Alaska,

**Lewis J.,** (2006), Open Geospatial Consortium Geo-Video Web Service, Open Geospatial Consortium, OGC Discussion paper, http://portal.opengeospatial.org/files/?artifact_id=12899

**Lippman A.,** (1980), *Movie-Maps: An Application of the Optical Videodisc to Computer Graphics*, ACM SIGGRAPH Computer Graphics, 14, 3, pp.32-42.

**Livingstone D., Raper J.,McCarthy T.,** (1998), *Integrating aerial videography and digital photography with terrain modelling: an application for coastal geomorphology,* Geomorphology, 29, pp.77-92.

**Liyuan L., Weimin H., Irene Y. H. G.,Tian Q.,** (2003), *Foreground object detection from videos containing complex background*, pp.2-10, ACM, Berkeley, CA, USA.

**Long S.,** (2006), Predator UAV Basic Universal Metadata Set, Motion Imagery Standards Board, Engineering Guideline 0104.5, http://www.gwg.nga.mil/misb/engpubs.html

**Longley P. A., Goodchild M. F., Maguire D. J.,Rhind D. W.,** (2001), *Geographic Query and Analysis: From data to Information,* pp.277-325, Wiley,

**Lyon R. F.,** (2006), Depth of Field Outside the Box, Technical Report, http://www.dicklyon.com/

**Maolin M., Miao W., Lianbi Y.,Chun L.,** (2009), *Automatic Vectorization of GPS Road Track in Navigable Database Production*, LBS and Telecartography, Nottingham,

**Marpe D., Wiegand T.,Sullivan G. J.,** (2006), *The H.264/MPEG4 advanced video coding standard and its applications*, Communications Magazine, IEEE, 44, 8, pp.134-143.

**Mc Loughlin S. D., Deegan C., Fitzgerald C.,Markham C.,** (2005), *Classification of road sign type using mobile stereo vision*, Opto-Ireland 2005: Imaging and Vision, Dublin, Ireland, 5823, pp.133-142.

**Mc Loughlin S. D., Deegan C., Mulvihill C., Fitzgerald C.,Markham C.,** (2008), *Mobile mapping for the automated analysis of road signage and delineation*, Intelligent Transport Systems, IET, 2, 1, pp.61-73.

**McCarthy T.,** (1999), *Integrating aerial video with GIS, PhD. Thesis*, University of London,

**McCarthy T., Farrell R., Curtis A.,Fotheringham A. S.,** (2008), *Integrated remotely sensed datasets for disaster management.*, Remote Sensing for Environmental Monitoring, GIS Applications, and Geology VIII, Cardiff,

**Metadata Working Group,** (2009), Guidelines For Handling Image Metadata, Metadata Working Group, Technical Specifications, http://www.metadataworkinggroup.org

**Microsoft,** (1992), New Multimedia Data Types and Data Techniques, Microsoft, http://netghost.narod.ru/gff/vendspec/micriff/index.htm.

**Microsoft,** (2005), Live Search Maps, Microsoft, http://maps.live.com.

**Microsoft,** (2007), Resource Interchange File Format Services, Microsoft, http://msdn.microsoft.com/en-us/library/ms713231.aspx.

**MISB,** (2008), Unmanned Air System (UAS) Datalink Local Metadata Set, Motion Imagery Standards Board, Standard, http://www.gwg.nga.mil/misb/stdpubs.html

**MISB AMIWG,** (2000), Advanced Motion Imagery Working Group, Motion Imagery Standards Board, http://www.gwg.nga.mil/misb/overview.html.

**Miyagi M.,** (1968), *Frequency-Shift-Keying Phase-Modulation code transmission system*, Nippon Electric Company Ltd,

**MPEG7,** (2006), MPEG-7 Overview, Motion Picture Experts Group, http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm.

**MPEG,** (2006), Moving Picture Experts Group, ISO audio/video standards organisation, http://www.chiariglione.org/mpeg/.

**Nack F.,Lindsay A. T.,** (1999), *Everything you wanted to know about MPEG-7: Part 1*, Multimedia, IEEE, 6, 3, pp.65-77.

**National Geodetic Survey USA,** (2006), Inverse/Forward/Inverse3D/Forward3D Computation Utilities, National Geodetic Survey USA,, Internet Site, http://www.ngs.noaa.gov/TOOLS/Inv_Fwd/Inv_Fwd.html.

**NATO STANAG,** (1995), NATO Standardisation Agreements, North Atlantic Treaty Organisation Standardisation Organisation, http://www.nato.int/docu/standard.htm.

**Navarrete T.,Blat J.,** (2002), *VideoGIS: Segmenting and indexing video based on geographic information*, 5th AGILE Conference on Geographic Information Science,, Palma, Balearic Islands, Spain,

**NavTech,** (1995), Real time GPS encoders and microwave systems development, NavTech Systems, Internet Source, http://www.navtechsystems.co.uk/.

**NavTech CamNav,** (2004), Functional Code for Geo Survey Application - Software Description, Report, http://www.navtechsystems.co.uk/

**NDIIPP,** (2008), National Digital Information Infrastructure and Preservation Program, US Library of Congress, http://www.digitalpreservation.gov/formats/fdd/fdd000183.shtml.

**Neumann U., Pintaric T.,Rizzo A.,** (2000), *Immersive Panoramic Video*, 8th ACM International Conference on Multimedia, Los Angeles, California, USA,

**Nobre E.,Camara A.,** (2001), Spatial Video, Exploring Space Using Multiple Digital Videos, Environmental Systems Analysis Group, New University of Lisbon, http://virtual.inesc.pt/egmm2001/proceedings/

**Ó Riain G.,McCarthy T.,** (2006A), Innovative data capture and presentation techniques in support of the EU Environmental Noise Directive, NCG, NUI Maynooth, Technical Report,

**Ó Riain G., McCarthy T., Kelly K., Reddy S.,O'Malley V.,** (2006B), *Data Capture and Presentation Techniques in support of the EU Evironmental Noise Directive*, EuroNoise 2006, Finland,

**Obermeyer K.,** (2006), VisiLibity.org, Internet Source, http://www.visilibity.org/.

**O'Connor N. E., Duffy T., Ferguson P., Gurrin C., Lee H., Sadlier D. A., Smeaton A. F.,Zhang K.,** (2008), *A content-based retrieval system for UAV-like video and associated metadata*, Airborne Intelligence, Surveillance, Reconnaissance (ISR) Systems and Applications V, 6946, pp.69460I.

**OGC 3D,** (2001), The OpenGIS Abstract Specification, Topic 1 - Feature Geometry (ISO 19107 Spatial Schema), Open Geospatial Consortium, Technical Report, http://www.opengeospatial.org/standards/as

**OGC OWS-3,** (2005), Open Geospatial Consortium, Open Web Services phase 3, Open Geospatial Consortium, Internet Source, www.opengeospatial.org/projects/initiatives/ows-3.

**OGC Simple Features,** (1999), OpenGIS Simple Features Specification For SQL, Open Geospatial Consortium, Technical Specifications, http://www.opengeospatial.org/standards/sfs.

**OOC,** (2008), The Future of Digital Terrestrial Television, Office of Communications, Standards Report, http://www.ofcom.org.uk/consult/condocs/dttfuture/dttfuture.pdf

**Panasonic Corporation,** (2008), Sensor Data Sheets, Technical Report, http://industrial.panasonic.com/www-cgi/jvcr21pz.cgi?E+SC+4+ADC7000+5++WW

**Pollefeys M., Van Gool L., Vergauwen M., Verbiest F., Cornelis K., Tops J.,Koch R.,** (2004), *Visual Modeling with a Hand-Held Camera*, International Journal of Computer Vision, 59, 3, pp.207-232.

**PostGIS,** (2001), PostgreSQL, PostGIS Spatial Extension, Refractions Research, Internet Source, http://postgis.refractions.net/.

**Qiang L., JaeJun Y., Byung-Tae J., KyoungHo C.,Jenq-Neng H.,** (2004), *A scalable VideoGIS system for GPS-guided vehicles*, Signal Processing: Image Communication, 20, 3, pp.205-218.

**Rana S.,** (2001), Isovist Analyst for ArcGIS 3.X, Sanjay Rana, GIS Software, http://www.ucl.ac.uk/~ucessan/software/isovist-analyst.htm.

**Red Hen,** (2005), Spatial Video hardware and software website, Red Hen Systems, Internet Source, http://www.redhensystems.com.

**Reinsch C.,** (1967), *Smoothing by spline functions*, Numerische Mathematik, 10, 3, pp.177-183.

**RouteMapper,** (2007), Spatial Video solutions for transportation sector, Internet Source, www.routemapper.net.

**Rzhanov Y., Linnett L. M.,Forbes R.,** (2000), *Underwater video mosaicing for seabed mapping*, Image Processing, 2000. Proceedings. International Conference on Image Processing, Vancouver, BC, Canada, Volume 1, pp.224-227.

**SDL,** (1998), Simple DirectMedia Layer - cross-platform multimedia programming library, Internet Source, http://www.libsdl.org/.

**SMPTE,** (2007), SMPTE 336m-2007 Data Encoding Protocol Using Key-Length Value, Society of Motion Picture and television Engineers, http://store.smpte.org/product-p/smpte%200336m-2007.htm.

**SMPTE,** (2007), Society of Motion Picture and Television Engineers, http://www.smpte.org/home.

**Sony Corporation,** (2008), Sensor DataSheets, Sony Corporation, Technical Report, http://www.sony.co.jp/~semicon/english/90203.html

**Springmeyer D.,** (2008), DBSGEO Cascade Union, Google Code Project, Internet Source, http://code.google.com/p/dbsgeo/source/browse/trunk/postgis/hydro_model/scripts/unite_cascade_functions.sql?spec=svn7&r=7#.

**STANAG 4609,** (2007), NATO Motion Imagery (MI) STANAG 4609 (Edition 2) Implementation Guide, North Atlantic Treaty Organisation Standardisation Organisation, Standards Implementation Guide, http://www.nato.int/structur/AC/224/standard/4609/4609.htm

**Sun R., Meng X.,Yang L.,** (2009), *Network Real-Time Kinematic GPS Based Positioning for Precise Vehicle Tracking*, LBS and Telecartography, Nottingham,

**Tae-Hyun H., Kyoung-Ho C., In-Hak J.,Jong-Hun L.,** (2003), *MPEG-7 Metadata for Video-Based GIS Applications*, Geoscience and Remote Sensing Symposium, 2003. IGARSS '03. Proceedings. 2003 IEEE International, 6, pp.3641- 3643.

**Tandy C. R. V.,** (1967), *The isovist method of landscape survey*, Symposium:Methods of Landscape Analysis Ed., London, pp.9-10.

**Tao,** (2006), The Tao Framework - C# Library of .Net bindings to popular graphics projects., Internet Source, http://taoframework.com/.

**Tiberius C.,** (2003), Standard Positioning Service - Handheld GPS Receiver Accuracy, GPS World, Technical Report, http://www.gpsworld.com/gpsworld/article/articleDetail.jsp?id=46415&searchString=Handheld%20GPS%20Receiver%20Accuracy%20GPS%20Receiver%20Accuracy

**TopoFusion,** (2002), TopoFusion.com, Morris A., Morris S., Software Processing Application for GPS, http://www.topofusion.com/about.php.

**Turner A.,** (1998), UCL DepthMap, University College London, Internet Source, http://www.vr.ucl.ac.uk/depthmap/.

**Turner A., Doxa M., O'Sullivan D.,Penn A.,** (2001), *From isovists to visibility graphs: a methodology for the analysis of architectural space*, Environment and Planning B: Planning and Design, 28, pp.103-121.

**Victor Company Of Japan (JVC),** (2006), JVC HD Camera Recorder GY-HD111 Instructions, JVC, Technical Report,

**Vincenty T.,** (1975), *Direct and Inverse Solutions of Geodesics on the ellipsoid with application of nested equations*, Survey Review, 23, 176, pp.88-93.

**Viqueira J., Lorentzos N.,Brisaboa N.,** (2005), *Survey on Spatial Data Modelling Approaches*, pp.1-18, Idea Group Inc.,

**Waters G. T.,** (1991), Specification of the European Broadcasting Union Subtitling data exchange format, European Broadcasting Union, Technical Reference Report, http://tech.ebu.ch/Jahia/site/tech/cache/bypass/publications;jsessionid=7EF17AE6A66F1477815C0B34FC3E7CEE.jahia1

**Wheeler R. E.,** (2003), Notes on View Camera Geometry, Technical Report, www.bobwheeler.com/photo/ViewCam.pdf

**Wiegand T., Sullivan G. J., Bjontegaard G.,Luthra A.,** (2003), *Overview of the H.264/AVC video coding standard*, Circuits and Systems for Video Technology, IEEE Transactions on, 13, 7, pp.560-576.

**Wilson S.,** (2003), WAVE PCM soundfile format, Stanford University Center for Computer Research in Music and Acoustics, http://ccrma.stanford.edu/courses/422/projects/WaveFormat/.

**Wolf P. R.,DeWitt B. A.,** (2000), *Elements of Photogrammetry(with Applications in GIS)*, McGraw-Hill Higher Education,

**Wood D.,** (2004), High Definition for Europe, European Broadcasting Union, http://www.ebu.ch/en/technical/trev/trev_home.html

**Worboys M.,Duckham M.,** (2004A), *GIS A Computing Perspective*, Second Edition, CRC Press,

**Worboys M.,Duckham M.,** (2004B), *Uncertain Viewsheds,* Second Edition, pp.355, CRC Press,

**Zlatanova S., Abdul Rahman A.,Pilouk M.,** (2002), *Trends in 3D GIS Development*, Geospatial Engineering, 4, pp.71-80.

# Publications

**Publications from work carried out during this PhD.**

M.O'Brien, A.S.Fotheringham, **P.Lewis** & M.Charlton, *Mapping Psychiatric Patient Records*. In report for Health Research Board, Ireland – Research on Mental Health 2000-2006.

**P.Lewis**, M.O'Brien, A.S.Fotheringham, M.Charlton & A.Winstanley, *Mapping Psychiatric Address Data*. In Proceedings of the 14th Annual GISRUK Conference 2006, Nottingham, England. Pages 131-135, April 2006.

A.S.Fotheringham, **P.Lewis**, M.O'Brien & M.Charlton, *A Spatial Analysis of Psychiatric Patient Records*. In Proceedings of the 36th Annual RSAIBIS Conference 2006, St.Helier, Jersey. Page 33, August 2006.

**Publications from work based on this PhD.**

**P.Lewis**, T.McCarthy, A.Winstanley & A.S.Fotheringham, *Software Algorithm for decoding GPS from spatially encoded video*. In Proceedings of GeoComputation Conference 2007, Maynooth, Ireland. September, 2006.

**P.Lewis**, A.Winstanley & T.McCarthy, *Synchronised Encoding of GPS NMEA Messages onto High Definition Video Streams*. Poster at the International Machine Vision and Image Processing Conference 2007, Maynooth, Ireland. September, 2007.

**P.Lewis** & A.Winstanley, *Enhancing Location Based Services with Spatial Video*. Finalist poster at the Year2008, Young European Arena of Research competition in the Transport Research Arena Conference 2008, Ljubljana, Slovenia. April, 2008.

**P.Lewis**, A.Winstanley & A.S.Fotheringham, *Position Paper: A conceptual model of Spatial Video moving objects using VIEWPOINT data structures*. Paper given at

Moving Objects Workshop as part of the GIScience Conference 2008, Park City, Utah, USA. September, 2008.

**P.Lewis** & A.Winstanley, *Towards Integrating Spatial Video in an LBS Context*. Submitted to LBS and TeleCartography Conference 2009, Nottingham, England. September, 2009.

**P.Lewis**, A.Winstanley & A.S.Fotheringham, *Using ViewCones to Model Terrestrial Spatial Video*. Submitted to GeoInfo3D Conference 2009, Ghent, Belgium. November, 2009.

**P.Lewis**, A.S.Fotheringham & A.Winstanley, *Role of Spatial Video in GIS*. Submitted to GeoComputation Conference 2009, Sydney, Australia. December, 2009.

## Appendix One:

### Geodetic Direct Algorithm converted to C#

```
public void Direct_Calculate()
{
  // Algorithm taken from National Geodetic Survey GeoTools
  // Forward Fortran program
  // http://www.ngs.noaa.gov/TOOLS/Inv_Fwd/Inv_Fwd.html
  // *** SOLUTION OF THE GEODETIC DIRECT PROBLEM AFTER T.VINCENTY
  // *** MODIFIED RAINSFORD'S METHOD WITH HELMERT'S ELLIPTICAL TERMS
  // *** EFFECTIVE IN ANY AZIMUTH AND AT ANY DISTANCE SHORT OF
  // *** ANTIPODAL LATITUDES AND LONGITUDES IN RADIANS POSITIVE NORTH
  // *** AND EAST AZIMUTHS IN RADIANS CLOCKWISE FROM NORTH
  // *** GEODESIC DISTANCE S ASSUMED IN UNITS OF SEMI-MAJOR AXIS A
  // *** (Meters)

  double EPS = .5 * Math.Pow(10, -13);
  double tempR = 1 - EllipFlat;
  double tempTU = tempR*Math.Sin(Sta1Lat) /Math.Cos(Sta1Lat);

  double azimuth = RadianToDegree(Sta1Azimuth)
                   -RadianToDegree(AzimuthDifference);

  if (azimuth < 0)
  {
     azimuth += 360;
  }
  else if (azimuth > 360)
  {
     azimuth -= 360;
  }


  double tempSF = Math.Sin(DegreeToRadian(azimuth));
  double tempCF = Math.Cos(DegreeToRadian(azimuth));
  this.Sta2Azimuth = DegreeToRadian(azimuth);

  double tempBAZ = 0.0;
  if(tempCF != 0.0)
  {
     tempBAZ = Math.Atan2(tempTU, tempCF) * 2;
  }

  double tempCU = 1 / Math.Sqrt(tempTU * tempTU + 1);
  double tempSU = tempTU * tempCU;
  double tempSA = tempCU * tempSF;
  double tempC2A = -tempSA * tempSA + 1;
  double tempX = Math.Sqrt((1 / tempR / tempR - 1)* tempC2A + 1) + 1;
  tempX = (tempX - 2) / tempX;
  double tempC = 1 - tempX;
  tempC = (tempX * tempX / 4 + 1) / tempC;
  double tempD = (tempX * .375 * tempX - 1) * tempX;
  tempTU = this.Distance / tempR / EllipAxis / tempC;
  double tempY = tempTU;

  double tempSY = 0.0, tempCY = 0.0, tempCZ = 0.0, tempE = 0.0;

  while(Math.Abs(tempY - tempC) > EPS)
  {
    tempSY = Math.Sin(tempY);
    tempCY = Math.Cos(tempY);
    tempCZ = Math.Cos(tempBAZ + tempY);
```

```
      tempE = tempCZ * tempCZ + 2 - 1;
      tempC = tempY;
      tempX = tempE * tempCY;
      tempY = tempE + tempE - 1;
      tempY = (((tempSY*tempSY*4-3)*tempY*tempCZ*tempD/6+tempX)
                          *tempD/4-tempCZ)*tempSY*tempD+tempTU;
    }
    tempBAZ = tempCU * tempCY * tempCF - tempSU * tempSY;
    tempC = tempR * Math.Sqrt(tempSA * tempSA + tempBAZ * tempBAZ);
    tempD = tempSU * tempCY + tempCU * tempSY * tempCF;

    if ( Double.IsNaN(Math.Atan2(tempD, tempC)))
    {
      this.Sta2Lat = this.Sta1Lat;
    }
    else
    {
      this.Sta2Lat = Math.Atan2(tempD, tempC);
    }

    tempC = tempCU * tempCY - tempSU * tempSY * tempCF;
    tempX = Math.Atan2(tempSY * tempSF, tempC);
    tempC = ((tempC2A * -3 + 4)*EllipFlat + 4)*tempC2A*EllipFlat / 16;
    tempD = ((tempE*tempCY*tempC+tempCZ)*tempSY*tempC+tempY)*tempSA;

    if(Double.IsNaN(Sta1Long + tempX -(1 - tempC)* tempD * EllipFlat))
    {
       this.Sta2Long = this.Sta1Long;
    }
    else
    {
        this.Sta2Long = this.Sta1Long+tempX-(1-tempC)*tempD*EllipFlat;
    }
  } // END Direct_Calculate()
```

## Appendix Two:

```
//Geodetic Interpolation distance calculation method.
//Original C# Implementation by Mike Gavaghan, www.gavaghan.org
public void Distance_Calculate(int FPS)
{
  // get constants
  double a = EllipAxis;
  double b = EllipAxisMinor;
  double f = EllipFlat;

  // get parameters as radians
  double phi1 = this.Sta1Lat;
  double lambda1 = this.Sta1Long;
  double phi2 = this.StaNextLat;
  double lambda2 = this.StaNextLong;

  // calculations
  double a2 = a * a;
  double b2 = b * b;
  double a2b2b2 = (a2 - b2) / b2;

  double omega = lambda2 - lambda1;

  double tanphi1 = Math.Tan(phi1);
  double tanU1 = (1.0 - f) * tanphi1;
  double U1 = Math.Atan(tanU1);
  double sinU1 = Math.Sin(U1);
  double cosU1 = Math.Cos(U1);

  double tanphi2 = Math.Tan(phi2);
  double tanU2 = (1.0 - f) * tanphi2;
  double U2 = Math.Atan(tanU2);
  double sinU2 = Math.Sin(U2);
  double cosU2 = Math.Cos(U2);

  double sinU1sinU2 = sinU1 * sinU2;
  double cosU1sinU2 = cosU1 * sinU2;
  double sinU1cosU2 = sinU1 * cosU2;
  double cosU1cosU2 = cosU1 * cosU2;

  // eq. 13
  double lambda = omega;

  // intermediates we'll need to compute 's'
  double A = 0.0;
  double B = 0.0;
  double sigma = 0.0;
  double deltasigma = 0.0;
  double lambda0;
  bool converged = false;

  for (int i = 0; i < 10; i++)  {
    lambda0 = lambda;
    double sinlambda = Math.Sin(lambda);
    double coslambda = Math.Cos(lambda);

    // eq. 14
    double sin2sigma = (cosU2 * sinlambda * cosU2 * sinlambda) +
                     (cosU1sinU2 - sinU1cosU2 * coslambda) *
                     (cosU1sinU2 - sinU1cosU2 * coslambda);

      double sinsigma = Math.Sqrt(sin2sigma);
      // eq. 15
      double cossigma = sinU1sinU2 + (cosU1cosU2 * coslambda);

      // eq. 16
      sigma = Math.Atan2(sinsigma, cossigma);

      // eq. 17 Careful! sin2sigma might be almost 0!
      double sinalpha = (sin2sigma == 0) ? 0.0 :
                       cosU1cosU2 * sinlambda / sinsigma;

      double alpha = Math.Asin(sinalpha);
      double cosalpha = Math.Cos(alpha);
      double cos2alpha = cosalpha * cosalpha;

      // eq. 18 Careful! cos2alpha might be almost 0!
      double cos2sigmam = cos2alpha == 0.0 ? 0.0 :
                    cossigma - 2 * sinU1sinU2 / cos2alpha;

      double u2 = cos2alpha * a2b2b2;

      double cos2sigmam2 = cos2sigmam * cos2sigmam;

       // eq. 3
       A = 1.0 + u2 / 16384 * (4096 + u2 *
                    (-768 + u2 * (320 - 175 * u2)));

       // eq. 4
       B = u2 / 1024 * (256 + u2 * (-128 + u2 * (74 - 47 * u2)));

       // eq. 6
       deltasigma = B * sinsigma * (cos2sigmam + B / 4
                    * (cossigma * (-1 + 2 * cos2sigmam2) - B / 6
                    * cos2sigmam * (-3 + 4 * sin2sigma)
                    * (-3 + 4 * cos2sigmam2)));

       // eq. 10
       double C = f / 16 * cos2alpha * (4 + f * (4 - 3 * cos2alpha));

       // eq. 11 (modified)
       lambda = omega + (1 - C) * f * sinalpha
                    * (sigma + C * sinsigma * (cos2sigmam + C
                    * cossigma * (-1 + 2 * cos2sigmam2)));

        // see how much improvement we got
        double change = Math.Abs((lambda - lambda0) / lambda);

        if ((i > 1) && (change < 0.0000000000001)){
          converged = true;
          break;
        }
     }

// eq. 1
double dist = Convert.ToDouble((b * A * (sigma - deltasigma)) / FPS);
if ( Double.IsNaN(dist) ){
  this.Distance = 0.0;
}
else
{
  this.Distance = dist;
 }

        }
```

## Appendix Three:

```
//Geodetic Azimuth calculation method.
//Original C# Implementation by Mike Gavaghan, www.gavaghan.org
private void Calculate_Azimuth()
{
 // get constants
 double a = EllipAxis; //ellipsoid.SemiMajorAxis;
 double b = EllipAxisMinor; //ellipsoid.SemiMinorAxis;
 double f = EllipFlat; //ellipsoid.Flattening;
 double TwoPi = 2.0 * Math.PI;

 // get parameters as radians
 double phi1 = this.Sta1Lat; //start.Latitude.Radians;
 double lambda1 = this.Sta1Long; //start.Longitude.Radians;
 double phi2 = this.StaNextLat; //end.Latitude.Radians;
 double lambda2 = this.StaNextLong; //end.Longitude.Radians;

 // calculations
 double a2 = a * a;
 double b2 = b * b;
 double a2b2b2 = (a2 - b2) / b2;

 double omega = lambda2 - lambda1;

 double tanphi1 = Math.Tan(phi1);
 double tanU1 = (1.0 - f) * tanphi1;
 double U1 = Math.Atan(tanU1);
 double sinU1 = Math.Sin(U1);
 double cosU1 = Math.Cos(U1);

 double tanphi2 = Math.Tan(phi2);
 double tanU2 = (1.0 - f) * tanphi2;
 double U2 = Math.Atan(tanU2);
 double sinU2 = Math.Sin(U2);
 double cosU2 = Math.Cos(U2);

 double sinU1sinU2 = sinU1 * sinU2;
 double cosU1sinU2 = cosU1 * sinU2;
 double sinU1cosU2 = sinU1 * cosU2;
 double cosU1cosU2 = cosU1 * cosU2;

 // eq. 13
 double lambda = omega;

 // intermediates we'll need to compute 's'
 double A = 0.0;
 double B = 0.0;
 double sigma = 0.0;
 double deltasigma = 0.0;
 double lambda0;
 bool converged = false;

 for (int i = 0; i < 20; i++)
 {
   lambda0 = lambda;
   double sinlambda = Math.Sin(lambda);
   double coslambda = Math.Cos(lambda);

   // eq. 14
   double sin2sigma = (cosU2 * sinlambda * cosU2 * sinlambda) +
            Math.Pow(cosU1sinU2 - sinU1cosU2 * coslambda, 2.0);
   double sinsigma = Math.Sqrt(sin2sigma);
     // eq. 15
```

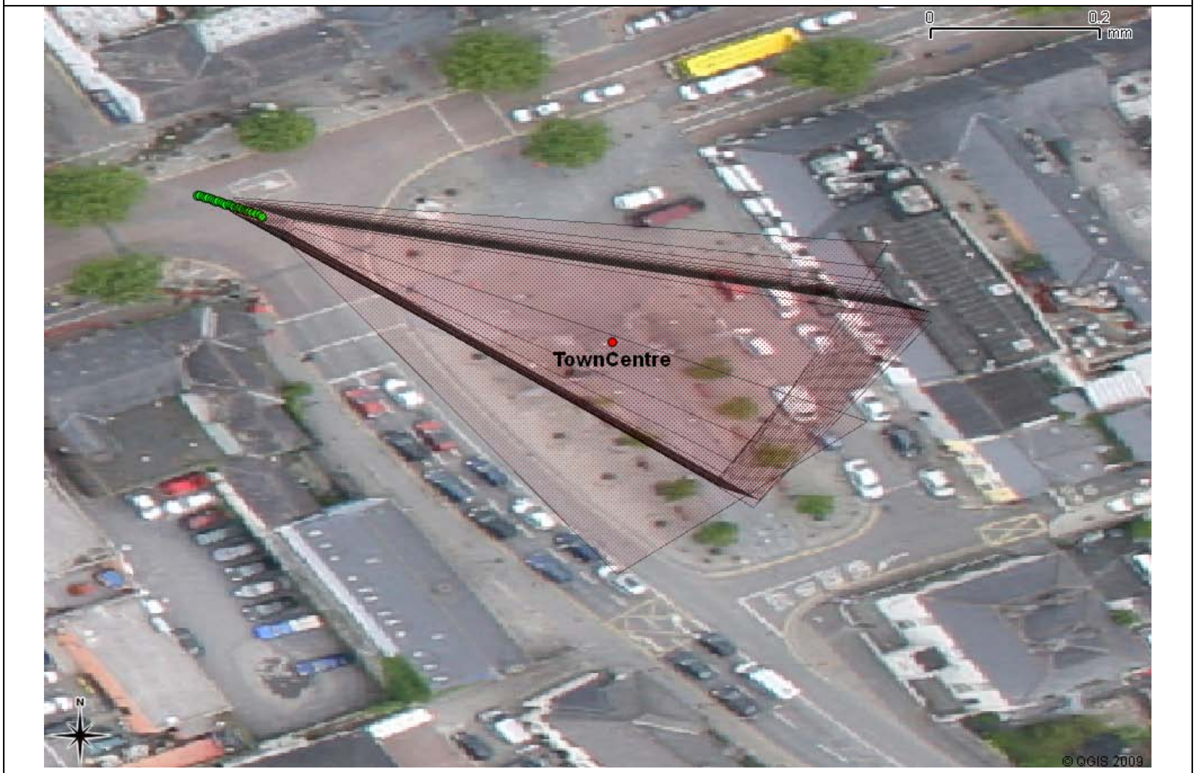
```
  double cossigma = sinU1sinU2 + (cosU1cosU2 * coslambda);

  // eq. 16
  sigma = Math.Atan2(sinsigma, cossigma);

  // eq. 17    Careful!  sin2sigma might be almost 0!
  double sinalpha = (sin2sigma == 0) ? 0.0 : cosU1cosU2 *
                     sinlambda / sinsigma;
  double alpha = Math.Asin(sinalpha);
  double cosalpha = Math.Cos(alpha);
  double cos2alpha = cosalpha * cosalpha;

  // eq. 18    Careful!  cos2alpha might be almost 0!
  double cos2sigmam = cos2alpha == 0.0 ? 0.0 : cossigma - 2 *
                      sinU1sinU2 / cos2alpha;
  double u2 = cos2alpha * a2b2b2;

  double cos2sigmam2 = cos2sigmam * cos2sigmam;

  // eq. 3
  A = 1.0 + u2 / 16384 * (4096 + u2*(-768 + u2*(320 - 175 * u2)));

  // eq. 4
  B = u2 / 1024 * (256 + u2 * (-128 + u2 * (74 - 47 * u2)));

  // eq. 6
  deltasigma = B * sinsigma * (cos2sigmam + B / 4 * (cossigma *
             (-1 + 2 * cos2sigmam2) - B / 6 * cos2sigmam *
             (-3 + 4 * sin2sigma) * (-3 + 4 * cos2sigmam2)));

  // eq. 10
  double C = f / 16 * cos2alpha * (4 + f * (4 - 3 * cos2alpha));

  // eq. 11 (modified)
  lambda = omega + (1 - C) * f * sinalpha * (sigma + C *
           sinsigma *(cos2sigmam + C * cossigma *
           (-1 + 2 * cos2sigmam2)));

   // see how much improvement we got
   double change = Math.Abs((lambda - lambda0) / lambda);

   if ((i > 1) && (change < 0.0000000000001))
   {
      converged = true;
      break;
   }
 }
// eq. 19
double s = b * A * (sigma - deltasigma);
double alpha1 = 0.0;
double alpha2 = 0.0;
// didn't converge?  must be N/S
if (!converged)
{
  if (phi1 > phi2)
  {
     alpha1 = 180;
     alpha2 = 0;
  }
  else if (phi1 < phi2)
  {
     alpha1 = 0;
     alpha2 = 180;
  }
  else
  {
        alpha1 = Double.NaN;
        alpha2 = Double.NaN;
     }
   }
   // else, it converged, so do the math
   else
   {
     double radians;

     // eq. 20
     radians = Math.Atan2(cosU2 * Math.Sin(lambda), (cosU1sinU2 –
               sinU1cosU2 * Math.Cos(lambda)));
     if (radians < 0.0) radians += TwoPi;
         alpha1 = RadianToDegree(radians);

     // eq. 21
     radians = Math.Atan2(cosU1 * Math.Sin(lambda), (-sinU1cosU2 +
               cosU1sinU2 * Math.Cos(lambda))) + Math.PI;
     if (radians < 0.0) radians += TwoPi;
         alpha2 = RadianToDegree(radians);
    }
    if (alpha1 >= 360.0) alpha1 -= 360.0;
    if (alpha2 >= 360.0) alpha2 -= 360.0;
    if ( !Double.IsNaN(alpha1) )
    {
      this.StaNextAzimuth = DegreeToRadian(alpha1);
    }
//END
}
```

## Appendix Four:

```
//Azimuth difference calculation method, used to
//normalise the result.
//Algorithm from Olof Bjarnason
//** www.gameprogrammer.com/archive/html/msg13901.html **
public void diffAzimuth_Calculate(double FPS)
 {
      Calculate_Azimuth();

      double angle1 = RadianToDegree(this.Sta1Azimuth);
      double angle2 = RadianToDegree(this.StaNextAzimuth);

     // Rotate angle1 with angle2 so that the sought after
      // angle is between the resulting angle and the x-axis
      angle1 -= angle2;
      // "Normalize" angle1 to range [-180,180)
      while(angle1 < -180)
        angle1 += 360;
      while(angle1 >= 180)
        angle1 -= 360;
      // angle1 has the signed answer, just "unsign it"

    //Result
      this.AzimuthDifference =
           DegreeToRadian(Convert.ToDouble(angle1 / FPS));
}
```

## Appendix Five:

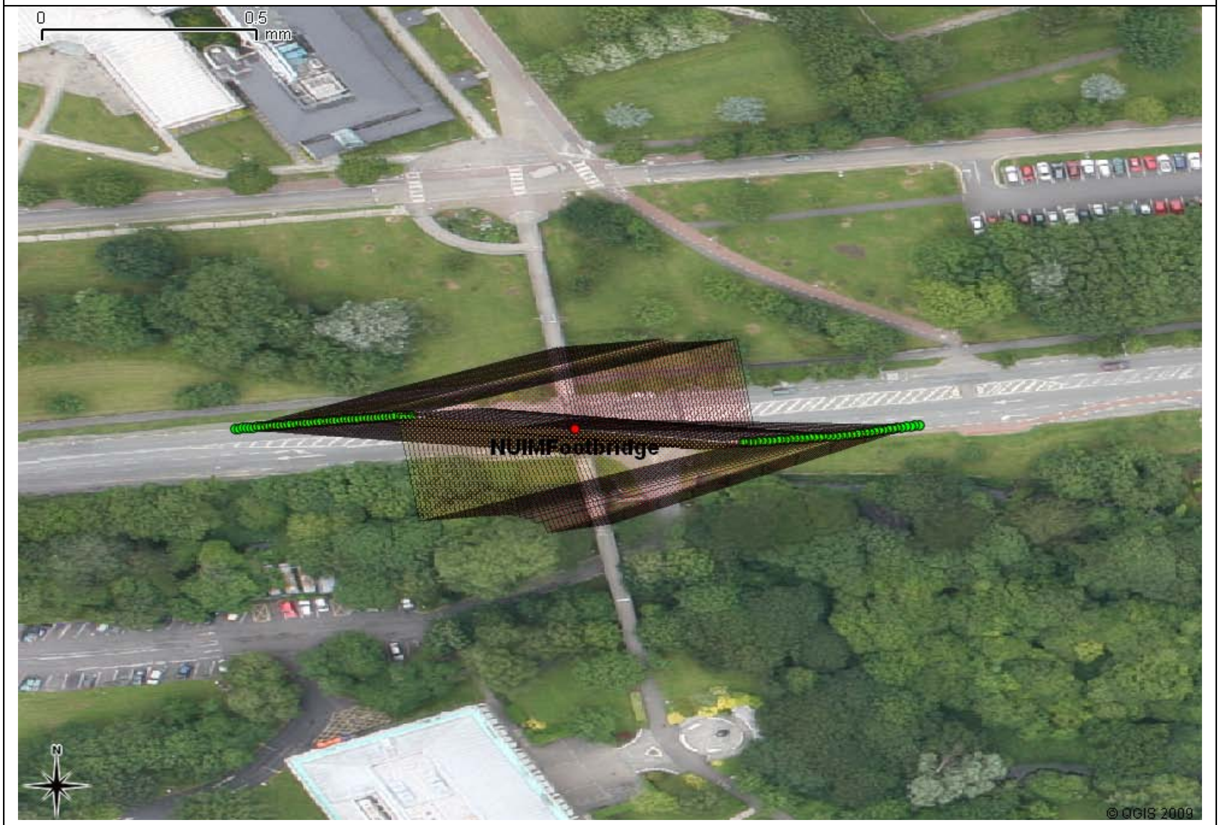
**Town Centre Point-Of-Interest spatial search results.**



In this set of results the query related to a point-of-interest for the old town square in the centre of Maynooth town. A total of 25 frames were returned from the query and all belonged to the Route 1 survey. The start and end frames for each trajectory sequence are shown in the following tables and clearly display the point-of-interest.

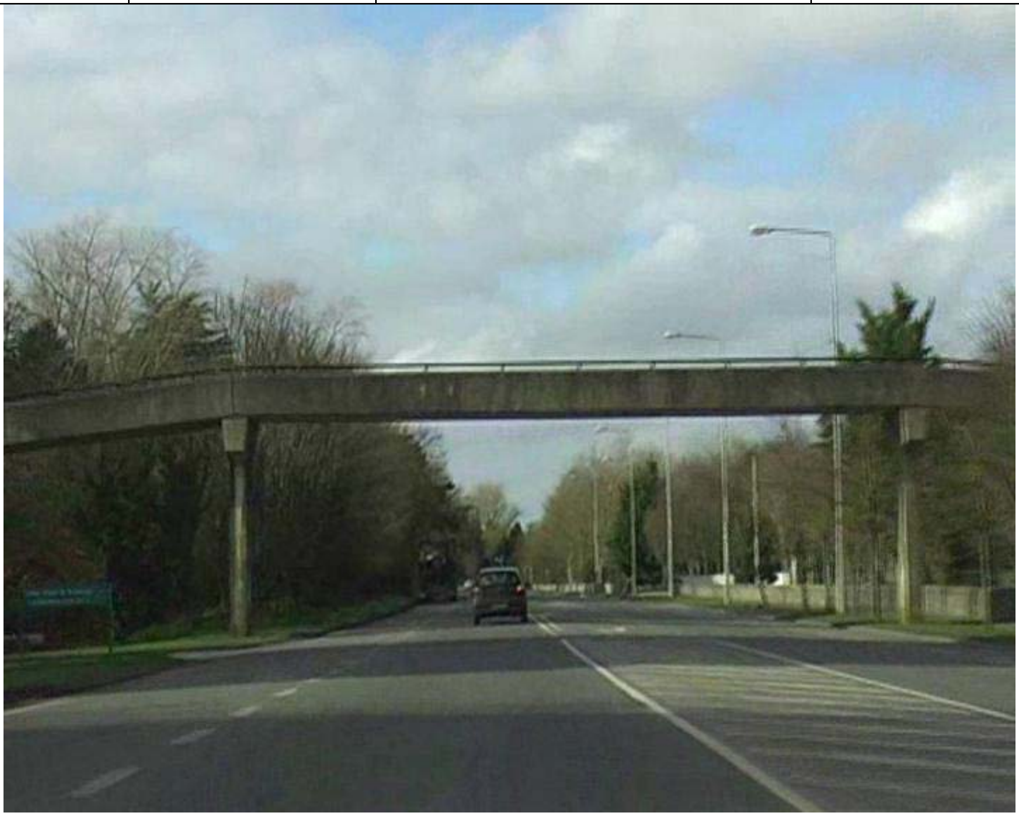
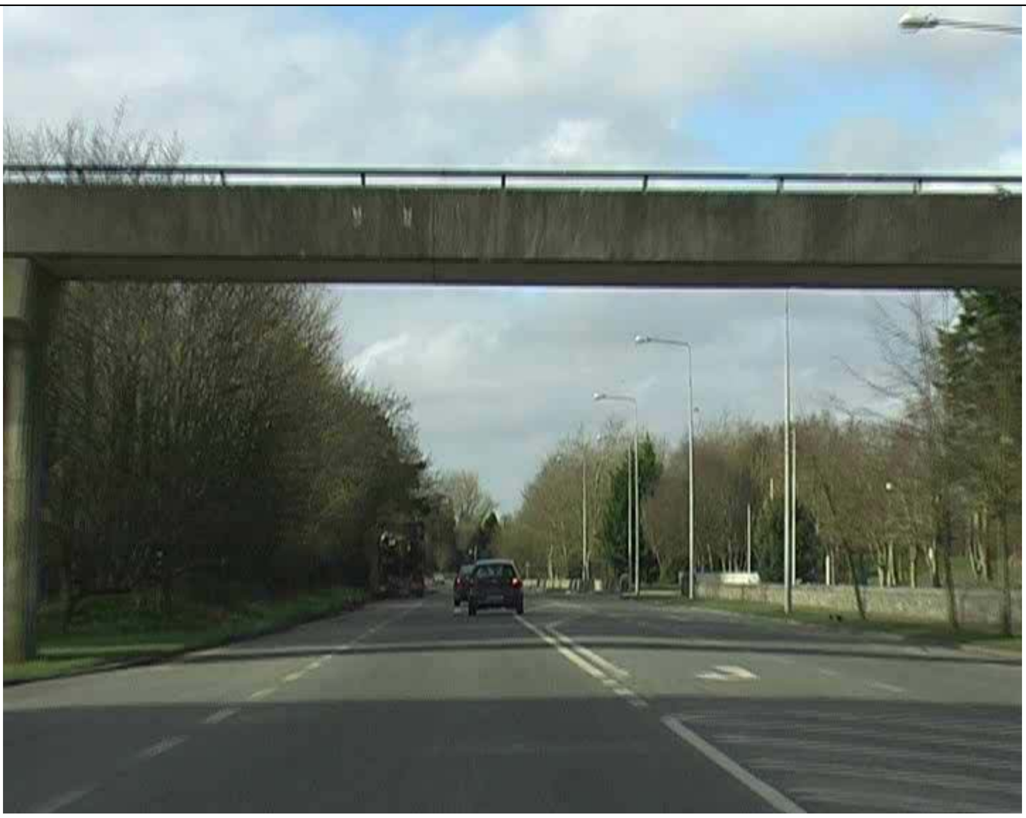
| **File Source** | Route1.wmv | **Video Frame Number Range** | 25 frames:<br>162120 to 163080 |
|---|---|---|---|
| | | | |
| | | | |

**NUIM Footbridge Point-Of-Interest spatial search results.**

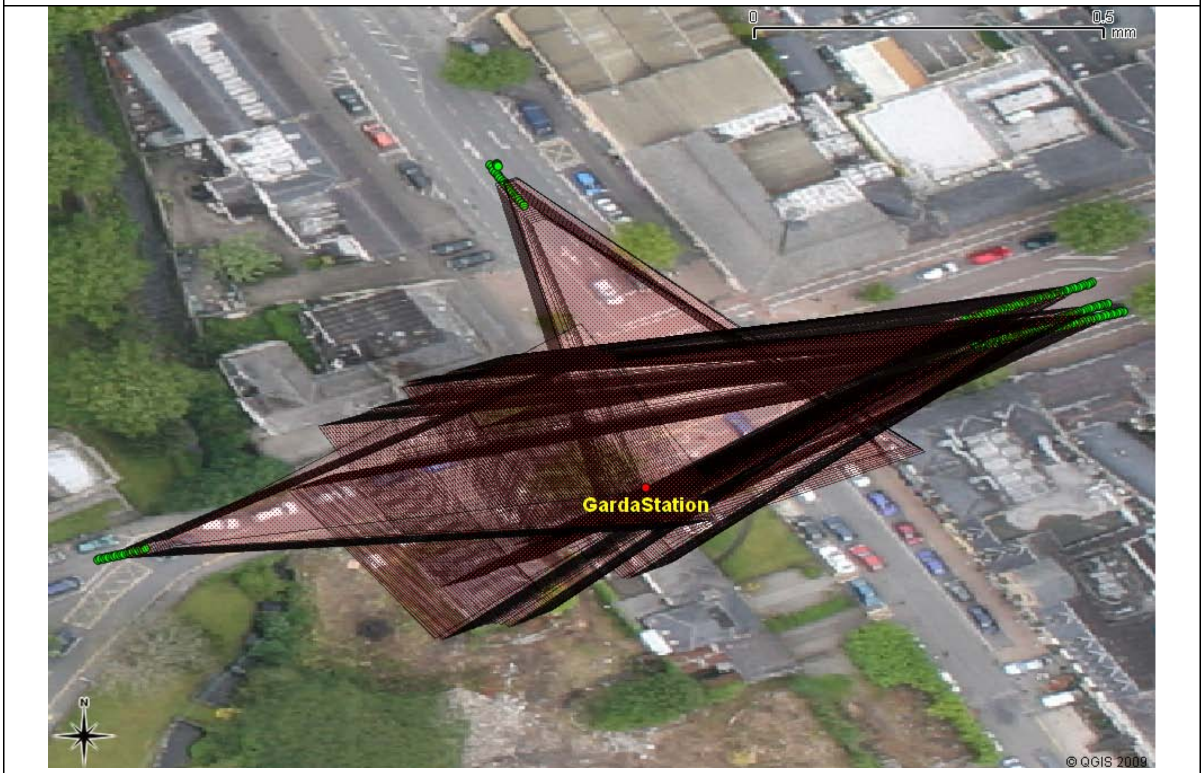


In this set of results the query related to a point-of-interest for the pedestrian footage that links the north and south campus at NUI Maynooth. A total of 86 frames were returned from the query and all belonged to the Route 2 survey; 47 frames for the west to east and 39 for the east to west trajectories. The start and end frames for each trajectory sequence are shown in the following tables and clearly display the point-of-interest.

| **File Source** | Route2.wmv | **Video Frame Number Range** | 47 frames:<br>20499 to 22379 |
|---|---|---|---|
| | | | |
| | | | |

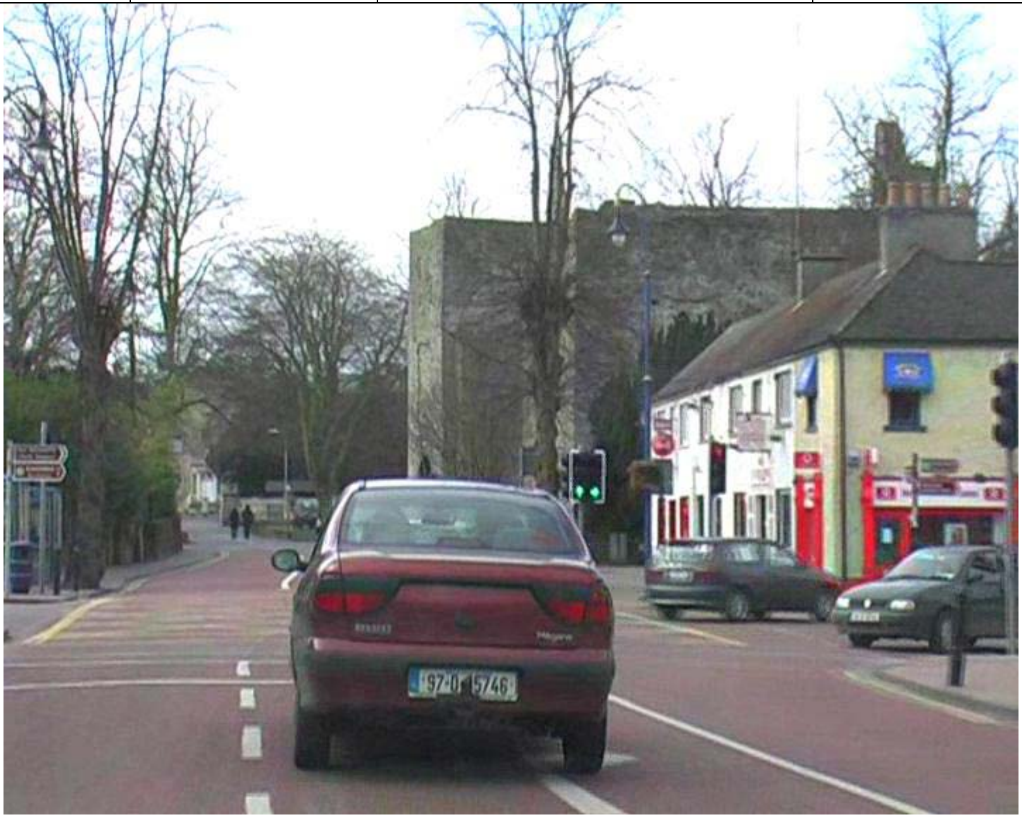
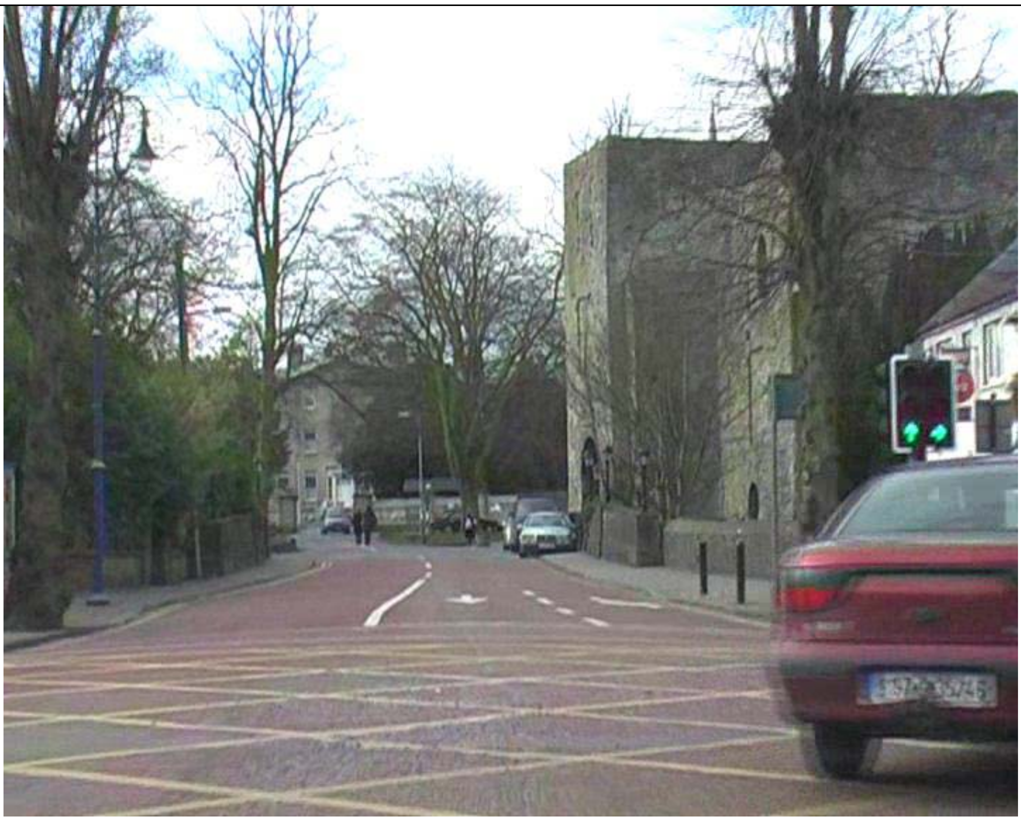
<table>
<tr><td><b>File Source</b></td><td>Route2.wmv</td><td><b>Video Frame Number Range</b></td><td>39 frames:<br>498059 to 499579</td></tr>
<tr><td colspan="4"></td></tr>
<tr><td colspan="4"></td></tr>
</table>

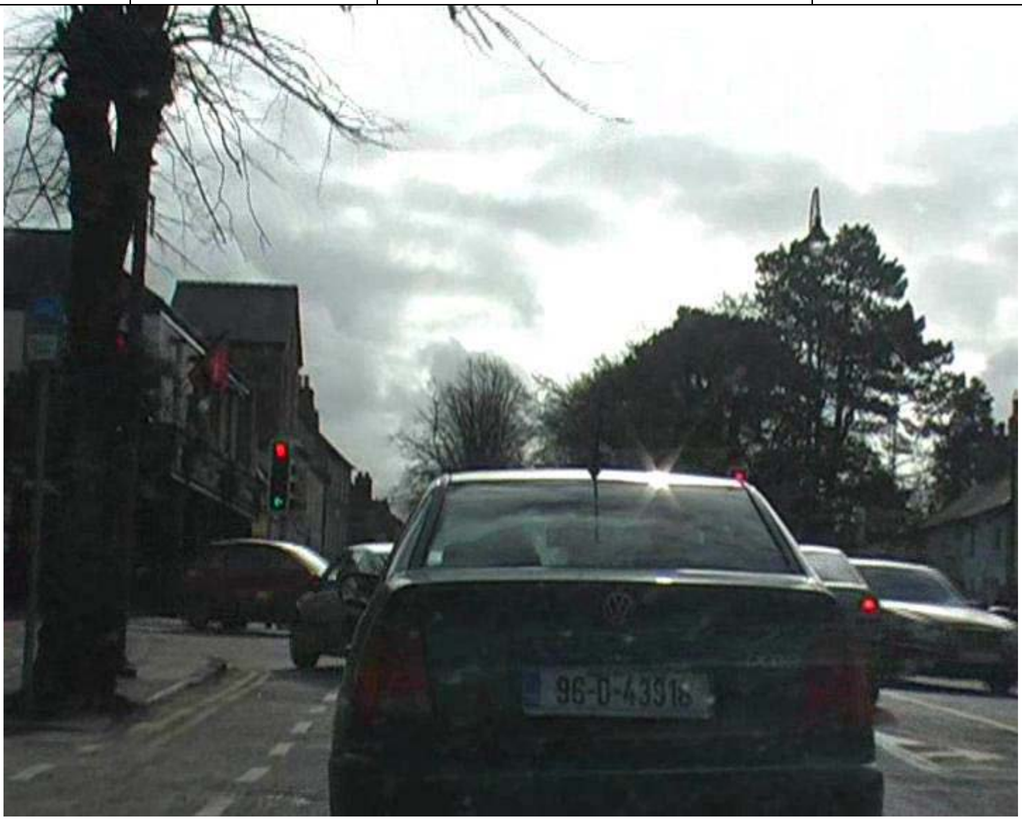
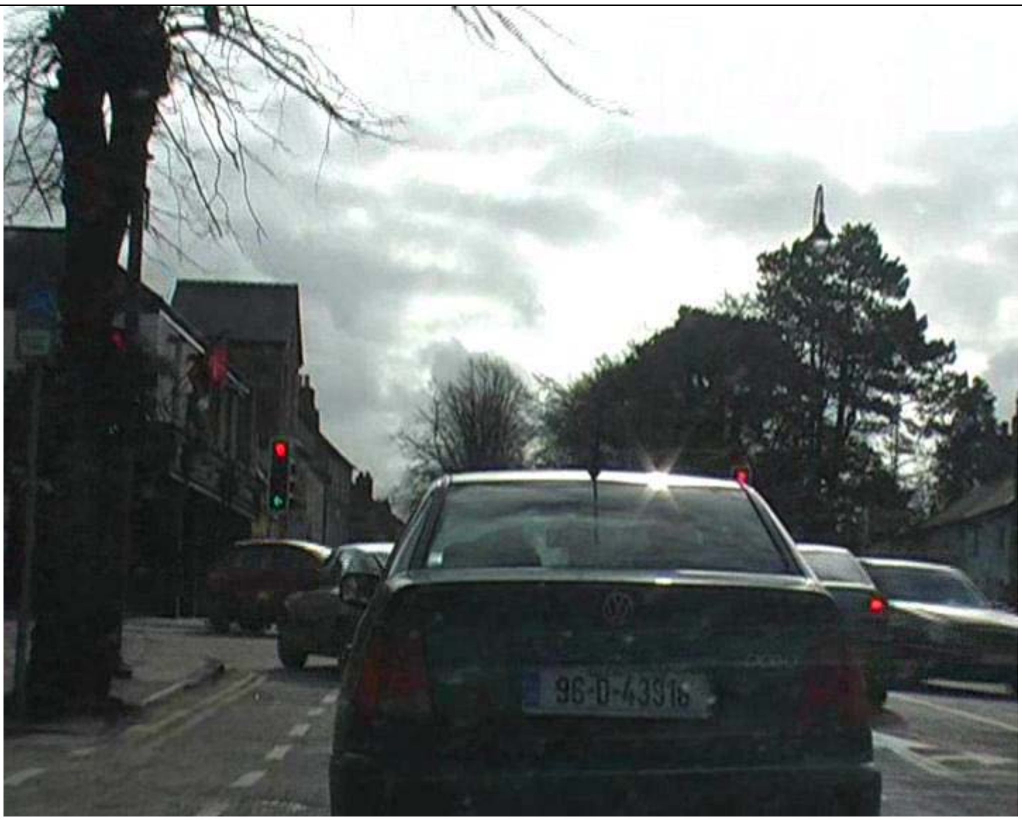
**Garda Station Point-Of-Interest spatial search results.**



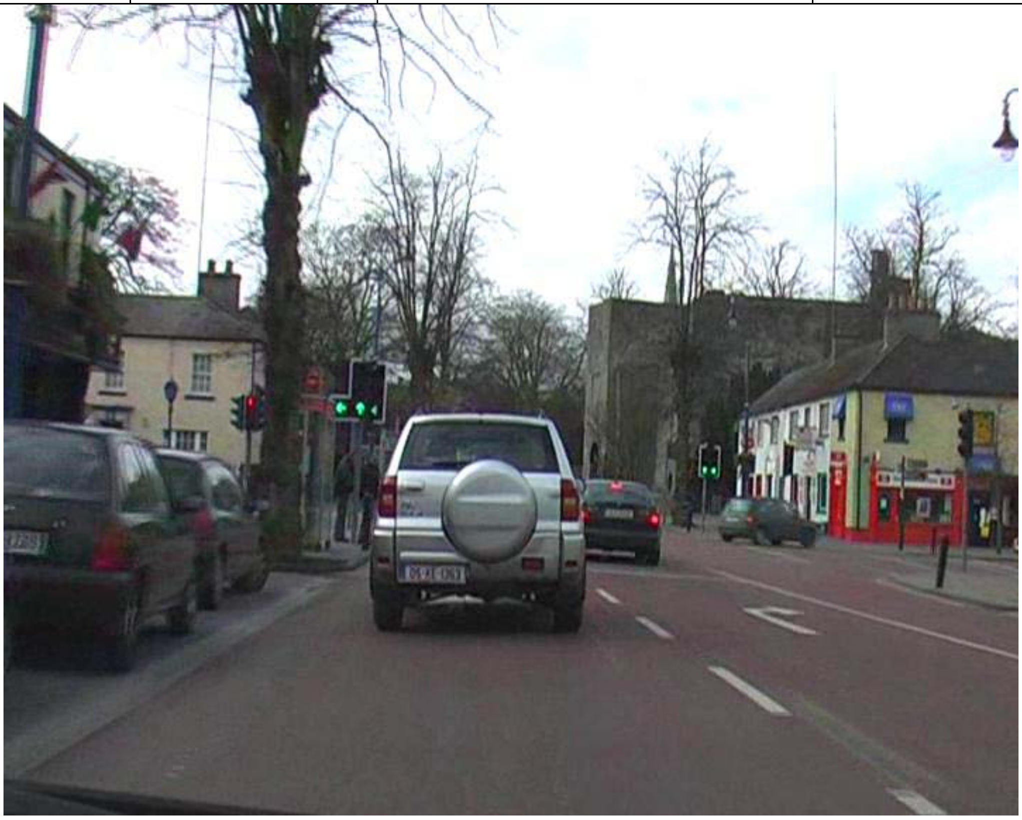
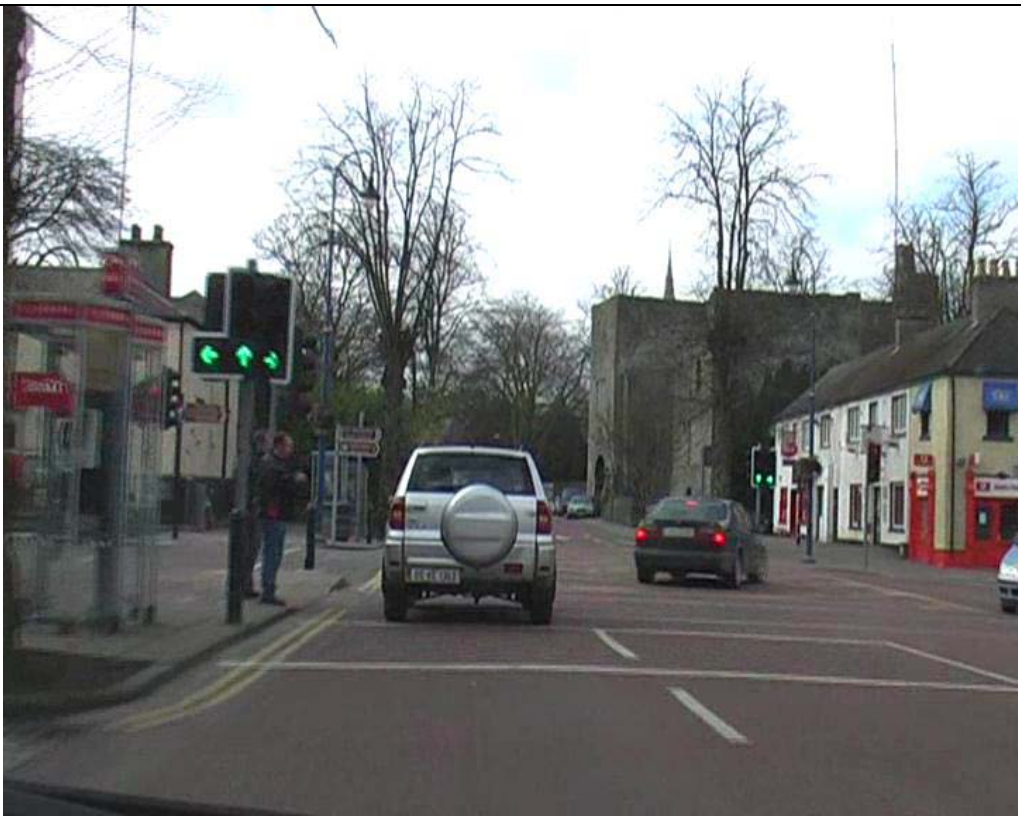
In this set of results the query related to a point-of-interest for Maynooth Garda (Police) Station. A total of 273 frames were returned from the query which belong to three different surveys; Route 1, Route 2 and Route 4. The start and end frames for the sequence are shown in the following tables. Route 4 has a number of frames, 152360 to 153080, where trees and overgrown hedgerow are occluding the view of the point-of-interest. However, based on knowledge of the area’s geography it can be determined that these frames would be correct but for the occlusions, thus the inaccuracy of the basic model is evident.

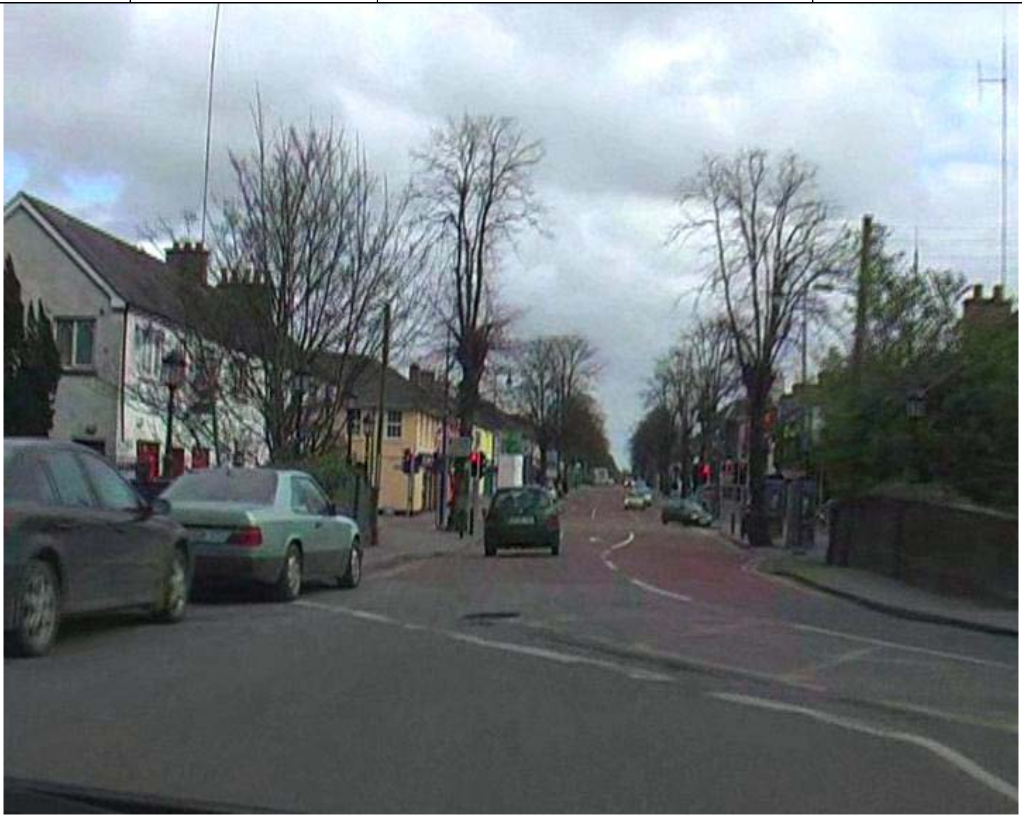
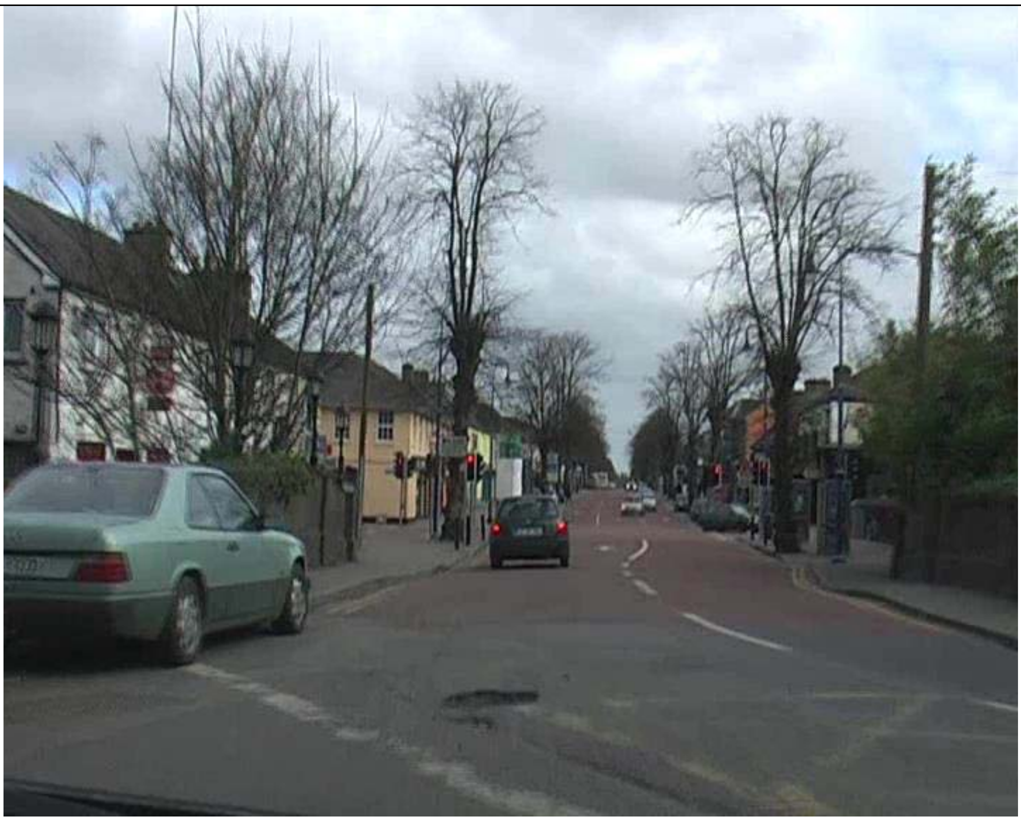
<table>
<tr><td><b>File Source</b></td><td>Route1.wmv</td><td><b>Video Frame Number Range</b></td><td>19 frames:<br>145600 to 146320</td></tr>
<tr><td colspan="4"> </td></tr>
<tr><td colspan="4"> </td></tr>
</table>

| **File Source** | Route1.wmv | **Video Frame Number Range** | 85 frames:<br>509440 to 512800 |
|---|---|---|---|
|  | | | |
|  | | | |

<table>
<tr><td><b>File Source</b></td><td>Route2.wmv</td><td><b>Video Frame Number Range</b></td><td>3 frames:<br>164539 to 164619</td></tr>
<tr><td colspan="4">
</td></tr>
<tr><td colspan="4">
</td></tr>
</table>

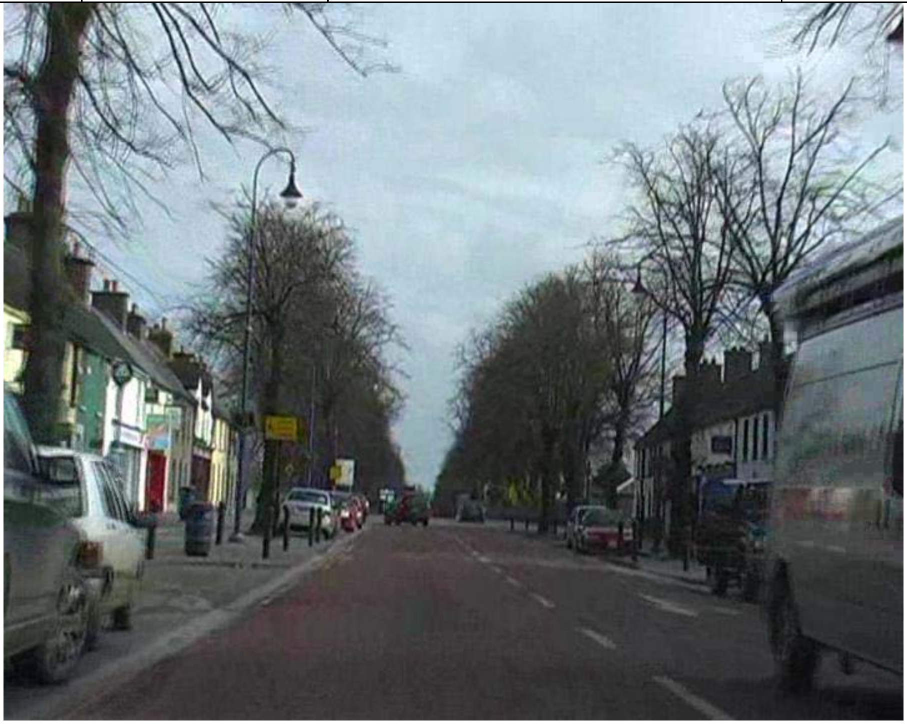
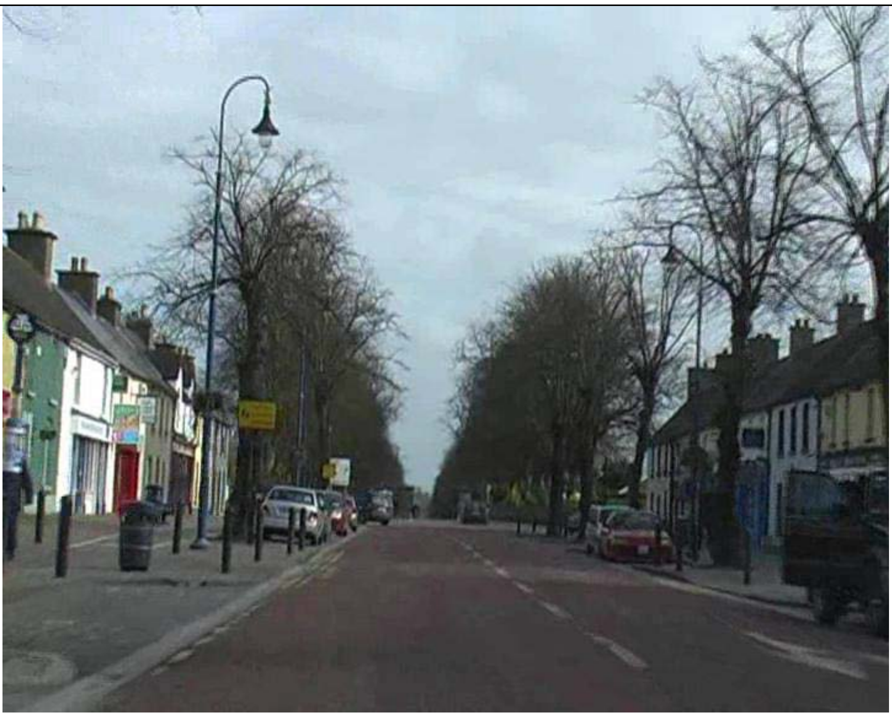
| **File Source** | Route2.wmv | **Video Frame Number Range** | 102 frames:<br>403259 to 407619 |
|---|---|---|---|
| | | | |
| | | | |

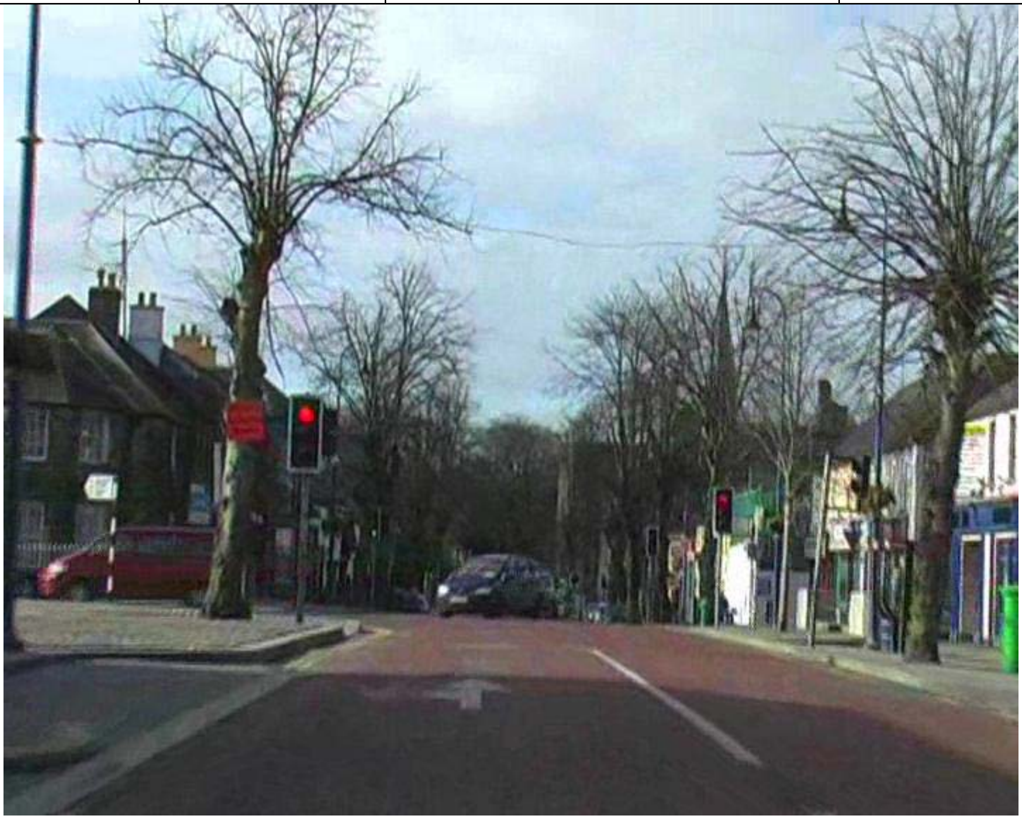
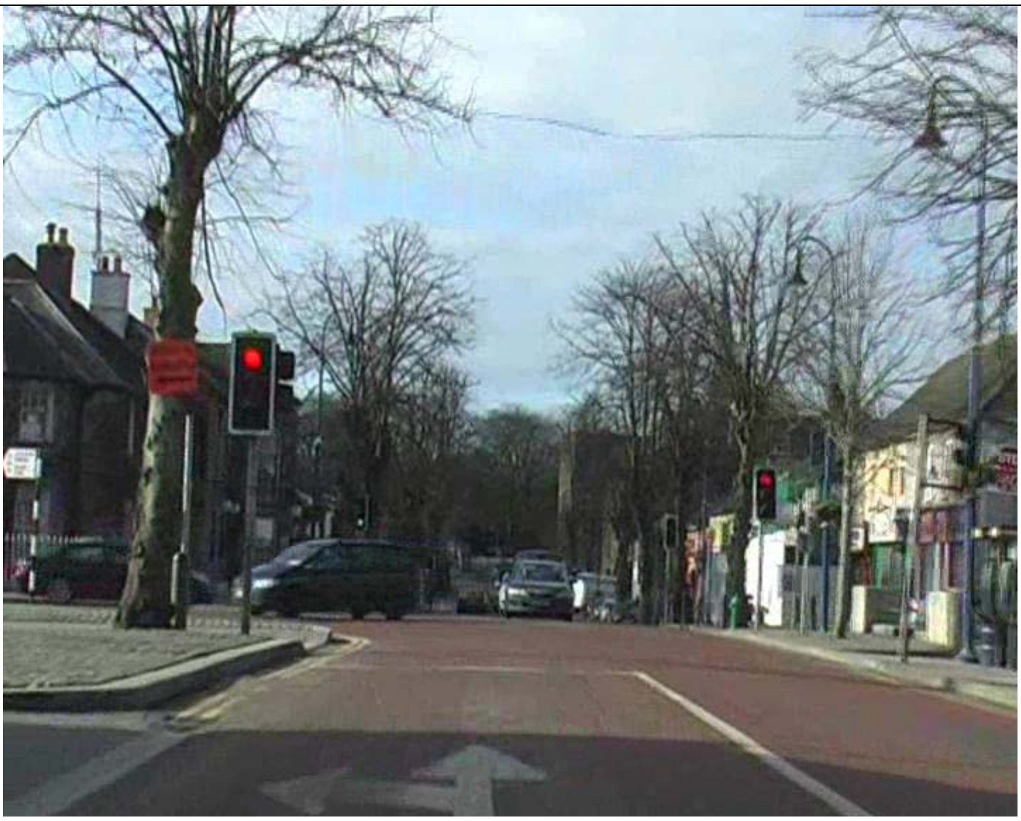
| **File Source** | Route4.wmv | **Video Frame Number Range** | 45 frames:<br>118880 to 120640 |
|---|---|---|---|
|  | | | |
|  | | | |

<table>
<tr><td><b>File Source</b></td><td>Route4.wmv</td><td><b>Video Frame Number Range</b></td><td>19 frames:<br>152360 to 153080</td></tr>
<tr><td colspan="4"></td></tr>
<tr><td colspan="4"></td></tr>
</table>

**Appendix Six:**

Spatial Video frame results for the second point-of-view operation with a five meter buffer control based on SQL query 7.3.

<table>
<tr><th>File Source</th><td>Route2.wmv</td><th>Video Frame Number Range</th><td>22 frames:<br>190539 to 191379</td></tr>
<tr><td colspan="4"></td></tr>
<tr><td colspan="4"></td></tr>
</table>

| **File Source** | Route2.wmv | **Video Frame Number Range** | 28 frames:<br>283419 to 284499 |
|---|---|---|---|
|  | | | |
|  | | | |

| **File Source** | Route4.wmv | **Video Frame Number Range** | 38 frames:<br>101160 to 102640 |
|---|---|---|---|
| | | | |
| | | | |

| **File Source** | Route4.wmv | **Video Frame Number Range** | 18 frames:<br>232960 to 233640 |
|---|---|---|---|
| | | | |
| | | | |

**Appendix Seven:**

Third point-of-view operation results with a five meter buffer and an orientation constraint control. These results are identical for both the 7.4 and 7.5 SQL statements.

| **File Source** | Route2.wmv | **Video Frame Number Range** | 22 frames:<br>190539 to 191379 |
|---|---|---|---|
| | | | |
| | | | |

| **File Source** | Route4.wmv | **Video Frame Number Range** | 15 frames:<br>233080 to 233640 |
|---|---|---|---|
| | | | |
| | | | |

## Appendix Eight:

Here we list the SQL Query script for the Line-Of-View operation where route and orientation constraints are defined to determine a more logical route.

```
/*Scripted SQL: Get one continuous route with orientation and consistency constraint*/
DECLARE @I, @J, @Temp1, @Temp2, @TempTable1, @TempTable2, @LinePoints;
DECALRE @Result, @RouteTable, @Loop, @Loop2, @Lines;
DECLARE @MAX, @MIN, @Record;
SET @I = 0;

/*Get all line points into a record array*/
SET @LinePoints = SELECT ST_PointN(the_geom, generate_series(1, ST_NPoints(the_geom)))
                                                        FROM "LineOfView";
SET @Loop = LINES(@LinePoints);

/*PRINT @LinePoints[@I + 1][0];*/

/*Create temp table to hold all line segments*/
SET @TempTable1 = DROP TABLE IF EXISTS "Temp_Store1";
SET @TempTable1 = CREATE TABLE "Temp_Store1"(id int4, Azimuth numeric DEFAULT 0)
                  with oids;
SET @TempTable1 = ALTER TABLE "Temp_Store1" ALTER COLUMN Azimuth SET NOT
                  NULL;
SET @TempTable1 = SELECT AddGeometryColumn( 'Temp_Store1', 'the_geom', 4326,
                  'LINESTRING', 2 );

/*Loop through all linepoints and insert into temp table the line segments*/
WHILE( (@Loop-1) > @I )
BEGIN
  SET @Temp1 = @LinePoints[@I][0];
  SET @Temp2 = @LinePoints[@I+1][0];
  SET @Lines = INSERT INTO "Temp_Store1"(id, the_geom, Azimuth) VALUES (@I,
                           ST_MakeLine('@Temp1', '@Temp2'), ST_Azimuth('@Temp1',
                                                         '@Temp2')/(2*pi())*360);
  SET @I = @I + 1;
END

/*Get all possible route streams on the line, count and order them in order of most common, use this list
to help define consistency*/
SET @RouteTable = SELECT poly.file_source, COUNT(*) AS how_many FROM svindex AS poly
                   INNER JOIN "LineOfView" AS ln ON
                            ST_DWithin(ST_Transform(ln.the_geom, 29903),
                                    ST_Transform(poly.point_geom, 29903), 5.0)
                   GROUP BY poly.file_source ORDER BY how_many DESC;

/*Create another temp table to hold all SV points and polygons that are close to each line segment and
azimuth orientated*/
SET @TempTable2 = DROP TABLE IF EXISTS "Temp_Store2";
SET @TempTable2 = CREATE TABLE "Temp_Store2"(id int4, sv_id integer, frame text, file text)
                  with oids;
SET @TempTable2 = SELECT AddGeometryColumn( 'Temp_Store2', 'point_geom', 4326, 'POINT',
                   2 );

SET @TempTable2 = SELECT AddGeometryColumn( 'Temp_Store2', 'polygon_geom', 4326,
                    'POLYGON', 2 );
/*Get all line segments for processing control loop*/
SET @Lines = SELECT the_geom FROM "Temp_Store1";
SET @MIN = SELECT ROUND(CAST(MIN(azimuth) AS NUMERIC), 0)-30 FROM
               "Temp_Store1";
SET @MAX = SELECT ROUND(CAST(MAX(azimuth) AS NUMERIC), 0)+40 FROM
               "Temp_Store1";
PRINT @MIN;
SET @MIN = @MIN[0][0];
SET @MAX = @MAX[0][0];
PRINT @MIN;

/*Loop through all lines segments to get all SVpoints within X meters of the line and within an
orientation range for the most common file.*/
SET @I = 0;
WHILE(@Loop > @I)
BEGIN
   SET @J = 0;
   SET @Loop2 = 0;
   WHILE((@Loop2 = 0) AND (LINES(@RouteTable) > @J))
   BEGIN
      SET @Record = '$$' + @RouteTable[@J][0] + '$$';
      SET @Temp2 = SELECT DISTINCT ON (poly.sv_id) poly.sv_id
                      FROM svindex AS poly, "Temp_Store1" AS line
                      WHERE line.id = @I
                      AND ST_DWithin(ST_Transform(line.the_geom, 29903),
                                     ST_Transform(poly.point_geom, 29903), 5.0)
                      AND poly.azimuth BETWEEN @MIN AND @MAX
                      AND poly.file_source = @Record;
      SET @Loop2 = LINES(@Temp2);
      SET @Temp2 = INSERT INTO "Temp_Store2" (id, sv_id, frame, file, point_geom,
                   polygon_geom)
                      SELECT DISTINCT ON (poly.sv_id) @I, poly.sv_id, poly.frame_number,
                                       poly.file_source, poly.point_geom,
                                       poly.polygon_geom
                        FROM svindex AS poly, "Temp_Store1" AS line
                        WHERE line.id = @I
                        AND ST_DWithin(ST_Transform(line.the_geom, 29903),
                                     ST_Transform(poly.point_geom, 29903), 5.0)
                         AND poly.azimuth BETWEEN @MIN AND @MAX
                         AND poly.file_source = @Record;
      PRINT @Loop2;
      SET @J = @J + 1;
   END
   SET @I = @I + 1;
END
PRINT 'DONE';
/*****END of scripting **************/
```

## Appendix Nine:

Listed here is the SQL Query script for the Thematic Coverage Polygon operation where the spatial union of all Viewpoint ViewCones with a 60% or greater coverage of each polygon is constructed. This also calculates the area of coverage in metres squared.

```
/*Scripted SQL Query Start***************/
DECLARE @TempTable1, @TempTable2, @I, @J, @LowValue, @NumValues, @Loop, @Temp;

SET @TempTable1 = DROP TABLE IF EXISTS viewlanduse;
SET @TempTable1 = CREATE TABLE viewlanduse (id int4, area_name text, land_use text, area
                                                            numeric, per_area numeric) WITH oids;
SET @TempTable1 = SELECT AddGeometryColumn ( 'viewlanduse', 'the_geom', 4326,
                                                             'MULTIPOLYGON', 3);

SET @LowValue = SELECT MIN(ogc_fid) FROM "smallareas";
SET @I = 0;
SET @NumValues = SELECT ogc_fid FROM "smallareas";
SET @Loop = LINES(@NumValues);

WHILE(@Loop > @I)
BEGIN
   SET @TempTable2 = DROP TABLE IF EXISTS temp2;
   SET @TempTable2 = CREATE TABLE temp2 (id int4) WITH oids;
   SET @TempTable2 = SELECT AddGeometryColumn('temp2', 'the_geom', 4326,
                                                              'MULTIPOLYGON', 3);
   SET @Temp = @LowValue[0][0] + @I;
   SET @J = @I+1;
   SET @TempTable2 = INSERT INTO temp2 (id, the_geom)
               (SELECT @J, ST_Multi(ST_Intersection(polygon_geom, (
                 SELECT wkb_geometry FROM "smallareas" WHERE ogc_fid = @Temp)))
                 FROM svindex
                 WHERE (ST_Area(ST_Transform(ST_Intersection(polygon_geom,
                      (SELECT wkb_geometry
                       FROM "smallareas"
                       WHERE ogc_fid = @Temp)),29903))
                       >=(SELECT AVG(ST_AREA(ST_Transform(polygon_geom, 29903)))/100*60
                  FROM svindex)));

    SET @TempTable1 = INSERT INTO viewlanduse (id, the_geom)
                         (SELECT @J, ST_Multi(ST_MemUnion(the_geom)) FROM temp2);

    SET @TempTable1 = UPDATE viewlanduse SET
                                area_name = (SELECT sm.area_name FROM "smallareas" AS sm
                                               WHERE ogc_fid = @Temp),
                                land_use = (SELECT land_use_description FROM "smallareas"
                                             WHERE ogc_fid = @Temp),
                                area = round(CAST (ST_Area(ST_Transform((
                                              SELECT the_geom FROM viewlanduse
                                              WHERE id = @J), 29903)) AS numeric), 2),
                                per_area = (100 / (round(CAST (ST_Area(ST_Transform((
                                              SELECT wkb_geometry FROM "smallareas"
                                              WHERE ogc_fid = @Temp), 29903)) AS numeric), 2))
```

```
                                                  )*(round(CAST (ST_Area(ST_Transform((
                                                  SELECT the_geom FROM viewlanduse WHERE id=@J),
                                               29903)) AS numeric), 2)) WHERE id = @J;
  SET @I = @I + 1;
END
/*Scripted SQL Query END***************/
```

## Appendix Ten:

Listed here is the SQL Query script for the Spatial Video Viewpoint Viewshed analysis operation. The PostGIS spatial operations environment does not inherently define a Viewshed algorithm for vector data support, therefore this algorithm was developed to generate the requirements of section 7.4.3.

```
/*Scripted SQL Query Start****************/
DECLARE @I, @J, @K, @LoopCrtl1, @TempVal, @TempVal2, @TempTable1, @TempTable2;
DECLARE @TempTable3, @TempTable4;

/*START: - Create Table to hold all Viewpoints that intersect with the buildings model.*/
SET @TempTable1 = DROP TABLE IF EXISTS "Temp_Store5";
SET @TempTable1 = CREATE TABLE "Temp_Store5" (sv_id integer) WITH oids;
SET @TempTable1 = SELECT AddGeometryColumn( 'Temp_Store5', 'polygon_geom', 4326,
                                                            'POLYGON', 2 );
SET @TempTable1 = INSERT INTO "Temp_Store5" (sv_id, polygon_geom)
                    (SELECT DISTINCT(sv_id), polygon_geom FROM svindex, model WHERE
                     ST_Intersects(svindex.polygon_geom, model.the_geom) ORDER BY sv_id);
SET @TempTable1 = VACUUM "Temp_Store5";
/*END: - Create Table to hold all Viewpoints that intersect with the buildings model.*/

/*Get total number of intersecting Viewpoints that need processing*/
SET @TempTable1 = SELECT sv_id FROM "Temp_Store5";
SET @LoopCtrl1 = LINES(@TempTable1);
Print @LoopCtrl1;

/*START: - Create table to hold all the Viewshed analysed polygons*/
SET @TempTable4 = DROP TABLE IF EXISTS "viewshedpolys";
SET @TempTable4 = CREATE TABLE "viewshedpolys" (sv_id integer) WITH oids;
SET @TempTable4 = SELECT AddGeometryColumn( 'viewshedpolys', 'polygon_geom', 4326,
                                                            'POLYGON', 2 );
/*END: - Create table to hold all the Viewshed analysed polygons*/

/*While Viewpoints are available for Viewshed analyses keep processing*/
SET @I = 0;
WHILE(@I < @LoopCtrl1)
BEGIN
  SET @TempVal = @TempTable1[@I][0];

  /*START: Create temporary table to hold the deconstructed Viewpoint ViewCone*/
  /*          1 holds the base view-line, 2 holds the far field view-line extent,          */
  /*          and 3 holds the dynamically created intersection search line                 */
  SET @TempTable2 = DROP TABLE IF EXISTS "Temp_Store6";
  SET @TempTable2 = CREATE TABLE "Temp_Store6" (id integer) WITH oids;
  SET @TempTable2 = SELECT AddGeometryColumn( 'Temp_Store6', 'line_geom', 4326,
                                                              'LINESTRING', 2 );
  SET @TempTable2 = INSERT INTO "Temp_Store6" (id, line_geom) VALUES
                    (1, (SELECT ST_MakeLine(ST_PointN(ST_ExteriorRing(polygon_geom), 1),
                                   ST_PointN(ST_ExteriorRing(polygon_geom), 2)) FROM
                                   "Temp_Store5" WHERE sv_id = '@TempVal')),
                    (2, (SELECT ST_MakeLine(ST_PointN(ST_ExteriorRing(polygon_geom), 4),
                                   ST_PointN(ST_ExteriorRing(polygon_geom), 3))FROM
                                   "Temp_Store5" WHERE sv_id = '@TempVal')),
                    (3, (SELECT ST_MakeLine(ST_PointN(ST_ExteriorRing(polygon_geom), 1),
                                   ST_PointN(ST_ExteriorRing(polygon_geom), 4))FROM
                                             "Temp_Store5" WHERE sv_id = '@TempVal'));

  SET @TempTable2 = VACUUM "Temp_Store6";
  /*END: - Create temporary table Viewpoint ViewCone deconstruction.*/

  /*START: - Create temporary table to hold query line intersection points*/
  SET @TempTable3 = DROP TABLE IF EXISTS "Temp_Store7";
  SET @TempTable3 = CREATE TABLE "Temp_Store7" (id integer) WITH oids;
  SET @TempTable3 = SELECT AddGeometryColumn( 'Temp_Store7', 'point_geom', 4326,
                    'POINT',
                                                  2 );
  /*END: - Create temporary table to hold query line intersection points*/

  /*While loop to calculate Viewshed*/
  SET @J = 1;
  SET @K = 0.01;
  WHILE(@K <= 1.01)
  BEGIN
    SET @TempTable3 = SELECT ST_PointN(ST_Intersection(model.the_geom,
                                        temp.line_geom),1)
                       FROM "Temp_Store6" AS temp, model WHERE temp.id = 3 AND
                       ST_Crosses(model.the_geom, temp.line_geom);
    SET @TempVal2 = LINES(@TempTable3);

    IF @TempVal2 > 0
    BEGIN
      PRINT 'INTERSECTION';
      SET @TempVal2 = @TempTable3[0][0];
      SET @TempTable3 = INSERT INTO "Temp_Store7" (id, point_geom) VALUES (@J,
                                                        '@TempVal2');
    END
    ELSE
    BEGIN
      PRINT 'NO INTERSECTION';
      SET @TempTable3 = INSERT INTO "Temp_Store7" (id, point_geom) VALUES
                    (@J, (SELECT ST_PointN(line_geom, 2) FROM "Temp_Store6"
                         WHERE id = 3));
    END

    IF @K <= 1
      SET @TempTable2 = UPDATE "Temp_Store6" SET line_geom = (SELECT ST_MakeLine(
                         ST_Line_Interpolate_Point((SELECT line_geom FROM
                         "Temp_Store6" WHERE id = 1), @K),
                         ST_Line_Interpolate_Point((SELECT line_geom FROM
                         "Temp_Store6" WHERE id = 2), @K))) WHERE id = 3;

    SET @J = @J + 1;
    SET @K = @K + 0.01;
  END
  SET @TempTable3 = INSERT INTO "Temp_Store7" (id, point_geom) VALUES
                    (@J, (SELECT ST_PointN(line_geom, 2) FROM "Temp_Store6"
                         WHERE id = 1));



     SET @J = @J + 1;
  SET @TempTable3 = INSERT INTO "Temp_Store7" (id, point_geom) VALUES
```

```
                    (@J, (SELECT ST_PointN(line_geom, 1) FROM "Temp_Store6"
                          WHERE id = 1));
  SET @J = @J + 1;
  SET @TempTable3 = INSERT INTO "Temp_Store7" (id, point_geom) VALUES
                    (@J, (SELECT point_geom FROM "Temp_Store7" WHERE id = 1));
  SET @TempTable3 = VACUUM "Temp_Store7";

  SET @TempTable4 = INSERT INTO "viewshedpolys" (sv_id, polygon_geom) VALUES
                    (@TempVal, (SELECT ST_MakePolygon(ST_MakeLine(point_geom))
                                   FROM "Temp_Store7"));

  SET @I = @I + 1;
END

SET @TempTable4 = VACUUM "viewshedpolys";
PRINT 'ALL DONE';
/*Scripted SQL Query Start***************/
```

**Appendix Eleven:**

Detailed here is the complete set of calculation tables for the results discussed in section 5.5.3. These calculations define the distances of each control point from the text image boundary lines.

| **Viewpoint Two** | **X1** | **Y1** | **X2** | **Y2** | **X** | **Y** | **Y2-Y1** | **X2-X1** | **Y1-Y2** | **X1*Y2** | **X2*Y1** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Tele Left Near : Far | | | | | | | **D** | **C** | **A** | **B** | **E** |
| Near Control | 3787180.71 | -438274.04 | 3787101.72 | -438304.44 | 3787182.39 | -438273.45 | -30.3997 | -78.9880 | 30.3996 | -1.66E+12 | -1.66E+12 |
| Far Control | 3787180.71 | -438274.04 | 3787101.72 | -438304.44 | 3787162.29 | -438279.83 | -30.3997 | -78.9880 | 30.3996 | -1.66E+12 | -1.66E+12 |

| **Distance to Line in Meters** | |
|---|---|
| **abs((A*X)+(C*Y)+(B-E))/sqrt(C*C+D*D)** | |
| 0.049494834 | Outside |
| 1.219050056 | Inside |

| **Viewpoint Two** | **X1** | **Y1** | **X2** | **Y2** | **X** | **Y** | **Y2-Y1** | **X2-X1** | **Y1-Y2** | **X1*Y2** | **X2*Y1** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Tele Right Near : Far | | | | | | | **D** | **C** | **A** | **B** | **E** |
| Near Control | 3787180.55 | -438272.04 | 3787099.12 | -438270.32 | 3787182.33 | -438272.12 | 1.7208 | -81.4389 | -1.7208 | -1.66E+12 | -1.66E+12 |
| Far Control | 3787180.55 | -438272.04 | 3787099.12 | -438270.32 | 3787161.19 | -438270.66 | 1.7208 | -81.4389 | -1.7208 | -1.66E+12 | -1.66E+12 |

| **Distance to Line in Meters** | |
|---|---|
| **abs((A*X)+(C*Y)+(B-E))/sqrt(C*C+D*D)** | |
| 0.03935661 | Inside |
| 0.971870801 | Outside |

| Viewpoint One | X1 | Y1 | X2 | Y2 | X | Y | Y2-Y1 | X2-X1 | Y1-Y2 | X1*Y2 | X2*Y1 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Wide Left Near : Far | | | | | | | **D** | **C** | **A** | **B** | **E** |
| Near Control | 3787184.83 | -438272.80 | 3787108.30 | -438335.32 | 3787184.50 | -438273.03 | -62.5167 | -76.5333 | 65.5167 | -1.66E+12 | -1.66E+12 |
| Far Control | 3787184.83 | -438272.80 | 3787108.30 | -438335.32 | 3787171.06 | -438283.02 | -62.5167 | -76.5333 | 65.5167 | -1.66E+12 | -1.66E+12 |

| **Distance to Line in Meters** | |
|---|---|
| **abs((A*X)+(C*Y)+(B-E))/sqrt(C*C+D*D)** | |
| 0.026349139 | Inside |
| 0.797888471 | Inside |

| Viewpoint One | X1 | Y1 | X2 | Y2 | X | Y | Y2-Y1 | X2-X1 | Y1-Y2 | X1*Y2 | X2*Y1 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Wide Right Near :Far | | | | | | | **D** | **C** | **A** | **B** | **E** |
| Near Control | 3787184.75 | -438271.80 | 3787100.87 | -438237.96 | 3787184.33 | -438271.68 | 33.8395 | -83.8857 | -33.8395 | -1.66E+12 | -1.66E+12 |
| Far Control | 3787184.75 | -438271.80 | 3787100.87 | -438237.96 | 3787165.93 | -438263.05 | 33.8395 | -83.8857 | -33.8395 | -1.66E+12 | -1.66E+12 |

| **Distance to Line in Meters** | |
|---|---|
| **abs((A*X)+(C*Y)+(B-E))/sqrt(C*C+D*D)** | |
| 0.051048604 | Inside |
| 1.066548432 | Outside |

| Telephoto Average Distance | Wide Angle Average Distance |
|---|---|
| 0.569943088 | 0.485458661 |

Conversion tables from WGS84 coordinates to Cartesian X, Y coordinates. This conversion was performed using Grid InQuest, available from Ordnance Survey Ireland website.

| **MAP** | **Longitude** | | **Latitude** | | **X** | **Y** | |
|---|---|---|---|---|---|---|---|
| **POINT** | **Wide Angle** | | | | | | |
| | -6.601191667 | 6° 36' 4.29000" | 53.38460833 | 53° 23' 4.58999" | 3787185.626 | -438272.15 | **Camera Origin** |
| **B** | -6.601187922 | 6° 36' 4.27652" | 53.38461848 | 53° 23' 4.62653" | 3787184.754 | -438271.798 | **Right Near Field** |
| **A** | -6.600827652 | 6° 36' 2.97955" | 53.38559483 | 53° 23' 8.14139" | 3787100.868 | -438237.959 | **Right Far Field** |
| **D** | -6.602268292 | 6° 36' 8.16585" | 53.38538693 | 53° 23' 7.39295" | 3787108.297 | -438335.317 | **Left Far Field** |
| **C** | -6.601202742 | 6° 36' 4.32987" | 53.38461634 | 53° 23' 4.61882" | 3787184.83 | -438272.8 | **Left Near Field** |
| | **Control Points** | | | | | | |
| **W** | -6.601206778 | 6° 36' 4.34440" | 53.38461967 | 53° 23' 4.63081" | 3787184.504 | -438273.032 | **Left Near Field Wide** |
| **Z** | -6.601186917 | 6° 36' 4.27290" | 53.38462336 | 53° 23' 4.64410" | 3787184.328 | -438271.682 | **Right Near Field Wide** |
| **X** | -6.601379028 | 6° 36' 4.96450" | 53.38475628 | 53° 23' 5.12261" | 3787171.064 | -438283.015 | **Left Far Field Wide** |
| **Y** | -6.601089889 | 6° 36' 3.92360" | 53.38483911 | 53° 23' 5.42080" | 3787165.925 | -438263.053 | **Right Far Field Wide** |

| **MAP** | **Longitude** | | **Latitude** | | **X** | **Y** | |
|---|---|---|---|---|---|---|---|
| **POINT** | **Telephoto Angle** | | | | | | |
| | -6.601191667 | 6° 36' 4.29000" | 53.38460833 | 53° 23' 4.58999" | 3787185.626 | -438272.15 | **Camera Origin** |
| **C** | -6.60119883 | 6° 36' 4.31579" | 53.38466487 | 53° 23' 4.79353" | 3787180.554 | -438272.043 | **Right Near Field** |
| **D** | -6.601313841 | 6° 36' 4.72983" | 53.38557268 | 53° 23' 8.06165" | 3787099.115 | -438270.322 | **Right Far Field** |
| **A** | -6.601818743 | 6° 36' 6.54747" | 53.38549982 | 53° 23' 7.79935" | 3787101.718 | -438304.443 | **Left Far Field** |
| **B** | -6.601228432 | 6° 36' 4.42236" | 53.3846606 | 53° 23' 4.77816" | 3787180.706 | -438274.043 | **Left Near Field** |
| | **Control Points** | | | | | | |
| **Z** | -6.601196917 | 6° 36' 4.30890" | 53.38464508 | 53° 23' 4.72229" | 3787182.325 | -438272.12 | **Right Near Field Tele** |
| **W** | -6.601216639 | 6° 36' 4.37990" | 53.38464264 | 53° 23' 4.71350" | 3787182.39 | -438273.448 | **Left Near Field Tele** |
| **Y** | -6.601211667 | 6° 36' 4.36200" | 53.384882 | 53° 23' 5.57520" | 3787161.187 | -438270.662 | **Right Far Field Tele** |
| **X** | -6.601346556 | 6° 36' 4.84760" | 53.38485794 | 53° 23' 5.48858" | 3787162.291 | -438279.825 | **Left Far Field Tele** |