UNIVERSITE DE LIMOGES

ECOLE DOCTORALE SCIENCE - TECHNOLOGIE

FACULTE DES SCIENCES & TECHNIQUES

Laboratoire XLIM

Thèse N°

Thèse

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE LIMOGES

Discipline / Spécialité : Informatique

présentée et soutenue par

Christos YIAKOUMETTIS

le 26 Novembre 2012

Semantic Navigation of Large Scale Geo-Referenced 3D Scenes and Virtual Worlds

Navigation sémantique de grandes scènes 3D georéféréncées et des mondes virtuels

Thèse dirigée par le Professeur Djamchid GHAZANFARPOUR

Co-directeur Professeur Georges MIAOULIS

JURY :

Président :

Yves DUTHEN Professeur, Université Toulouse - IRIT

Rapporteurs :

Yves DUTHEN Professeur, Université Toulouse - IRIT

Elli PETSA Professeur, Institut d'Education Technologique d'Athènes

Examinateurs :

Djamchid GHAZANFARPOUR Professeur, Université de Limoges - XLIM

Georges MIAOULIS Professeur, Institut d'Education Technologique d'Athènes

Olivier TERRAZ Maître de Conférences, Université de Limoges - XLIM

Invités :

Dimitri PLEMENOS Professeur (retraité), Université de Limoges

Nikolaos VASSILAS Professeur, Institut d'Éducation Technologique d'Athènes

I dedicate this work

to everyone

who inspired and supported me

Acknowledgements

This document is my thesis at University of Limoges. This research area is dealing with automatic navigation into large scale 3D geo-referenced scenes. I would like to thank my research committee that supervised this work and everyone who supported this work.

In particular

I would like to thank Dr. Georgios Miaoulis, Professor of the department of Informatics at T.E.I. of Athens, for having supervised and supported my work.

I would like to thank Dr. Djamchid Ghazanfarpour, Professor of the department of Informatics at University of Limoges, for having supervised and supported my work.

I would like to thank Dr. Dimitris Plemenos, Retired Professor of the department of Informatics at University of Limoges.

Moreover I would like to thank the readers Georgios Bardis and Nikolaos Doulamis. Finally I would like to thank everyone that believed and supported with their love, my work.

Table of Contents

List of Tables

List of Figures

Résumé

Les évolutions technologiques actuelles font enter la géo-informatique 3D à son âge numérique, permettant de nouvelles applications potentielles dans le domaine du tourisme virtuel, du loisir, du divertissement et du patrimoine culturel. Des mondes virtuels 3D et des scènes géo-référencées sont utilisés encore plus dans les simulations de catastrophes physiques et de plan d'évacuation ou de scénarios militaires. Il est prouvé que les informations 3D offrent un moyen naturel de navigation. Cependant la personnalisation est un aspect essentiel dans un système de navigation. En dehors des optimisations de distance et du temps l'intégration des préférences des utilisateurs est finalement le point les important d'une navigation. Une architecture efficace de planification d'itinéraire personnalisée est basée tant sur des critères géométriques que sur les préférences de l'utilisateur. Dans ce travail, un cadre multi-couche intégrée est introduit, pour soutenir une navigation efficace centrée sur l'utilisateur, dans des mondes 3D.

Habituellement, les préférences de l'utilisateur sont exprimées sous forme d'un ensemble de poids qui représente le degré d'importance du contenu des métadonnées sémantiques de la scène sur le processus de sélection du tracé. Ces poids, cependant, sont définis par les utilisateurs, en mettant la complexité du côté de l'utilisateur, ce qui rend la personnalisation une tâche ardue. Dans ce travail, une approche alternative est proposée dans laquelle les poids du contenu des métadonnées sont estimés implicitement et de façon transparente pour les utilisateurs, en déplaçant la complexité du côté du système. Ce résultat est obtenu par l'introduction d'une stratégie d'apprentissage en ligne par retour d'expérience qui ajuste automatiquement les poids de variables des métadonnées par exploitation de l'information renvoyée vers le système sur la pertinence des jugements de préférences utilisateur présentés sous une forme de comparaisons par paires. Dans la pratique la mise en œuvre d'un algorithme de retour d'expérience présente la limitation que plusieurs comparaisons par paires (échantillons) sont nécessaires pour converger vers un ensemble de poids de variables des métadonnées fiables. Pour cette raison, dans ce travail une stratégie de rectification de poids est proposée qui permet d'estimer le poids en exploitant les interrelations de variables de métadonnées définies par une ontologie. Dans la suite, un algorithme d'optimisation génétique est proposé pour sélectionner les itinéraires les plus préférés de l'utilisateur basé sur une approche d'optimisation multi-critères. Pour améliorer le degré de personnalisation en navigation 3D, un algorithme efficace est également mis en place pour estimer les trajectoires 3D autour des objets choisis par la fusion de meilleurs vues projetées en 2D qui contiennent des faces qui ont la plus grande préférence des utilisateurs. Des simulations et des comparaisons ont été effectuées avec d'autres approches, soit dans le domaine de l'apprentissage en ligne ou du choix du tracé en utilisant des mesures objectives en termes de précision et de rappel des valeurs.

Abstract

The current technological evolutions introduce 3D geo-informatics to their digital age, enabling new potential applications in the field of virtual tourism, recreation, entertainment and cultural heritage. Virtual 3D worlds and geo-referenced scenes are used even more in simulations of physical disasters or in evacuating and military scenarios. It is argued that 3D information provides the natural way of navigation. However, personalization is a key aspect in a navigation system, since a route that incorporates user preferences is ultimately more suitable than a route which simply provides the shortest distance or travel time. An efficient personalized route planning architecture is based on geometrical criteria and on human factors regarding user's preferences in the 3D itinerary. In this work a multi-layer integrated framework is introduced for efficient, user-centric navigation of 3D worlds.

Usually, user's preferences are expressed as a set of weights that regulate the degree of importance of the scene semantic metadata on the route selection process. These weights, however, are defined by the users, setting the complexity on the user's side, which makes personalization an arduous task. In this work, an alternative approach is proposed in which metadata weights are estimated implicitly and transparently to users, transferring the complexity to the system side. This is achieved by introducing a relevance feedback online learning strategy which automatically adjusts metadata weights by exploiting information fed back to the system about the relevance of user's preferences judgments given in a form of pairwise comparisons. Practically implementing a relevance feedback algorithm presents the limitation that several pairwise comparisons (samples) are required to converge to a set of reliable metadata weights. For this reason, in this work a weight rectification strategy is proposed which improves weight estimation by exploiting metadata interrelations defined through an ontology. In the sequel, a genetic optimization algorithm is incorporated to select the user's most preferred routes based on a multi-criteria optimization approach. To increase the degree of personalization in 3D navigation, an efficient algorithm is also introduced for estimating 3D trajectories around objects of interest by merging best selected 2D projected views that contain faces which are mostly preferred by users. Simulations and comparisons have been conducted with other approaches either in the field of online learning or route selection using objective metrics in terms of precision and recall values.

Introduction

1. Introduction

In this chapter the motivation which triggered this work will be presented. The corresponding research areas will be briefly discussed along with the objectives of this research work.

This thesis is dealing with personalized understanding and semantic navigation in three dimensional large scale geo-referenced scenes or virtual worlds. Initially, the motivation of this work will be listed and a reference to the current state of the GIS platforms will take place. Secondly, the goal and the objectives of this work will be presented. Finally, the implementation and the contributed areas of our implementation will be outlined.

1.1. Introduction

Throughout the last decades a rapid evolution of technology is taking place, continuously changing contemporary life style. We are surrounded daily by a variety of devices and offered comforts, products of the twentieth and twenty first century. Our era is governed by information and computers, so nowadays people are encountered with a huge amount of information. Tasks such as choosing, understanding, data gathering and, in many cases, decisions based on information and data are taking place.

Devices and media have been developed in order to be able to manipulate the requirements of today. At the same time, part of this evolution is computers and telecommunication technology.

The field of computer science has boosted the progress of many scientific fields. The evolution in computer science is on the increase these last decades. Today there are personal computers with multi cores and the speed of each core can reach 4GHz. The main memory has the capacity of many dozens of Giga bytes. In the time line of last 40 years, the computational power of personal computers has increased by more than 1,000 times.

Apart from personal computers, several devices interrelated with computers have been developed. One of them is displays, which are used for visualizing data. Today monitors can display data and content in three dimensions, while three dimensional holograms displays seem to be the next step in display technology and data visualization.

Parallel to the development of computers, data transfer protocols and architectures have been delivered. Quite expectedly, the evolution of technology has pushed telecommunications into new levels, new limits are set and the transfer bandwidth of telecommunications always increases. Data can be digitalized and transferred by a unique unified network. Last but not least, exceptionally high speed broadband internet connections have been established practically everywhere around the world. Apart from the broadband landlines, high speed internet access can be granted from Wi-Fi spots or 3G/4G mobile networks.

Global Positioning System was developed in the early 1960s by NASA. It was designed for military purposes, and it is clear from its definition that it is a platform estimating the position of an object on the surface of the whole planet. It is supported by satellites and it was used for military purposes only until the end of 1990s when a limited edition of it was provided as free service to public. In a short time period it became popular and many applications have since been developed using GPS data.

The evolution in computer science, in displays and in communication technologies are three main features of our time. They define a new contemporary life style. At the same time they create a fruitful ground for new applications and new technologies.

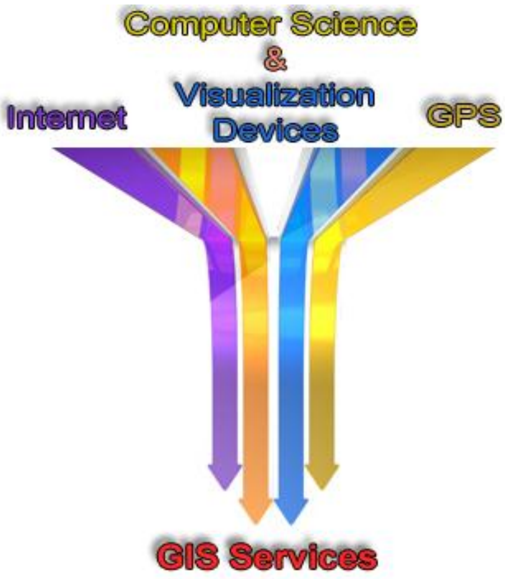


Figure 1: Convergence of technological fields

The convergence of the aforementioned technological fields and the GPS technology, leads to the motivation for new implementations. An example of a modern application that takes advantage of the four technological fields is the Geographic Information System (GIS) services.

GIS services have recently come in the forefront of people’s interest. Existing geographic information systems gained spatial reference on a real or on a virtual world.

It was at the end of 2004 when Google bought a standalone mapping program. The application was transformed into a web service and that was the beginning of a new era for the web based mapping services. Google maps were introduced early in 2005. Since that day several GIS platforms have been presented. These applications are known as “*virtual globes*” and their evolution was rapid.

Satellite raster images [72] and three dimensional models of buildings have been added. Whole cities have been modelled and appeared on GIS platforms. Some of these modelled cities are representations of real cities while some others are virtual ones.

These environments have to handle a large amount of information all calibrated using a common reference spatial system. The global coordinates system is generally used for this purpose. The content that is used is typically text, images and, lately, three dimensional models of objects (buildings).

Navigation and exploration of such environments can take place, by the user, using a special 3D browser. This exploration is controlled manually by the user, resulting into a path in the three dimensional virtual environment. The exploration of a three dimensional geo-referenced scene is practically a path planning process. Path planning is typically not a trivial task. The path estimation has to respect user’s preferences and search criteria. This process could be improved through automatic adaptation or guidance of the exploration with regard to each user’s personal preferences.

An effective path planning system has to consider user’s preferences. Therefore a personalized path planning architecture has been designed. Initially, the virtual world of the GIS platform has to be defined in a language compatible with the proposed research. Each 3D object of the scene is being modelled and specifications that describe it are defined. Moreover, a machine learning algorithm is used, in order to capture the user’s preferences.

Respecting the user’s preferences, the platform estimates the *optimal* path that matches to user’s desires.

On the top of interest, the navigation and the gathering of information in large scale scenes is important. The aforementioned platform can estimate efficiently a virtual tour on GIS environment respecting user’s preferences. Such an architecture can be applied in areas of entertainment. Computer graphics and 3D scenes are part of any video game. The aforementioned implementation can be used to navigate automatically into the scene based on the case of exploration and the user's search criteria. Moreover, it can be used for educational purposes. A virtual tour of an existing historical or cultural museum or memorial is doubtlessly appealing and important while it can support the process of learning through multimedia content thus leading to easier acquisition of knowledge.

Applying the aforementioned architecture in a 3D large scale scene, a virtual tour can be accomplished. Not only the distance but also the cost is eliminated. Moreover it gives the opportunity to a user, for whom it is physically hard if not impossible to visit a place, to enjoy a tour and admire the local sightseeings. Additionally, virtual 3D scenes may present or simulate alternative states or help the user jump along the time line of the location, reviving the environment of historical and cultural events.

1.2. Thesis objectives

For the aforementioned reasons, the creation of an intelligence platform for automatic navigation into 3D large scale scenes is important. This work is dealing with a personalized route planning architecture into a large scale three dimensional scene. The path has to be estimated using both semantic and geometrical features of the scene, along with respect of the user’s preferences and constrains.

1.3. The proposed methodology

The creation of such architecture on the one hand requires knowledge of the three dimensional scene and on the other hand the desires and preferences of the user have to be expressed in a form appropriate for integration to the platform. Moreover the interesting points have to be selected by respecting user’s preferences. Each desired visiting point has to be presented based on user’s criteria.

- ✓ Initially the three dimensional models of the scene have to be constructed. For such a process, 3D re-construction tools were used.
- ✓ Three dimensional models have to be defined efficiently by a set of descriptors. An ontological model was used for this purpose.
- ✓ Subsequently, an active learning algorithm acquires the user's preferences. These preferences will represent the user's potential desire to see/visit an object
- ✓ Path estimation will take place considering user's criteria.
- ✓ Finally, individual 3D object representation will take place respecting user's preferences and 3D model specifications.

An ontological model is used to define user's preferences and 3D model specifications. The learning algorithm is responsible for the acquisition of the user's desires and the elimination of the conceptual "gap" between the user's desires and the set of descriptors.

1.4. Contribution Areas

The current work is introducing a prototype architecture that deals with many scientific areas. Although there are many approaches and methodologies on each individual area, there is not any approach, according to the literature review, that combines these fields in a way that this work does. To the best of my knowledge, semantic navigation in 3D geo-referenced scene is a new scientific field and there is not any implementation or approach that combines semantic and geometric features of a 3D scene in order to estimate efficiently a personalized path in a 3D large scale geo-referenced scene.

Three dimensional re-construction process is the first phase of what will be presented on this work. This initial process provides the required dataset for the experiments and our implementation. 3D reconstruction in computer graphics is the process during which the 3D shape of an object is estimated by the captured images. The reconstruction process can take place in an automated or semi-automated way. In the first case we can see two different approaches the active one and the passive one. The first approach uses sensors both to measure distances and to have interaction with the physical object. In the other case, the 3D reconstruction process is based only on captured images specifications. In the case of semi-automated 3D reconstruction the process of estimating the shape of the 3D object is based on human guidance on a 3D perception of an object.

Machine Learning is a scientific area under Artificial Intelligence. Machine learning is the process of knowledge acquisition or adapting behaviour based on examples or rules through the use of computer algorithms. According to evolutionary artificial intelligence, the machine learning process uses dynamic structures such as artificial neural networks. In general, the aim of machine learning is to take advantage of training examples and capture characteristics of interest. The trainee uses a set of training representative examples. The training set has to contain as many samples as possible. The difficulty lies in the fact that the set of all possible behaviours given all possible inputs is too large to be covered by the set of training examples. Hence the trainee must generalize from the given examples, so as to be able to produce a useful output in new cases. An active inductive learning algorithm has beed adopted and it will be presented on this work using a pairwise comparison method for user's profile estimation in semantic navigation in the 3D scene.

Scene exploration is the area concerning the presentation of a 3D object in a way to respect a set of given criteria. In computer science, scene exploration deals with automatic navigation into a 3D scene. In the field of computer graphics, the scene exploration process deals with the estimation of specific points or viewing angles that have to be considered in the final path. These points are selected based on special criteria or evaluation. Several algorithms have been developed that estimate a set of points that have to be included into the final calculated path. According to literature review several methodologies have been implemented. Some of the approaches use predefined points of interest and some others use measuring criteria in order to evaluate the points or viewing angles. In any case, the final path is calculated by considering the aforementioned selected visiting points. This work will modify an existing best viewing algorithm in order to consider semantic and geometrical features of the scene in a way to estimate a personalized 3D navigation in a large scale geo-referenced scene.

1.5. Thesis organization

This work is organized in three parts. In the first part the theoretical background that is required will be presented in three chapters. In particular, the first chapter of this part and second of this thesis contains a presentation of the chronological overview of the evolutions of GIS platforms and an overview report of their functional structure. The next chapter presents the relevant concepts from the area of machine learning. The last chapter of the first part comprises a literature review of similar state of the art systems and techniques.

The second part of this work contains six chapters. The fifth chapter refers to the contribution of the research project and to a brief overview of the platform. The sixth chapter presents the methodology of 3D model creation of buildings. In chapter seven the active learning methodology will be implemented for acquiring user’s preferences. In chapter eight the selection of the visiting points using a genetic algorithm is discussed. In chapter nine, a best a viewing algorithm is used in order to estimate an optimal individual 3D object presentation corresponding to user’s preferences. A detailed implementation of the platform will be presented in chapter ten.

The last part consists of two chapters. The results of this research project are presented and a detailed discussion of the findings is taking place. In chapter eleven, a general discussion about the results and the performance of the proposed architecture take place. Furthermore, in the last chapter, an overview of this work its contribution to the relevant scientific fields is presented along with proposed topics for potential future work.

Part I: Theoretical Background & State of the Art

2. Large Scale Geo-referenced 3D Scenes

2.1. Large Scale 3D Scenes Using Coordinate System

Computer graphics have greatly evolved during the last years. Graphics processing units have achieved to provide enough computational power to design photorealistic 3D graphics on today's personal computers.

The 3D scenes are designed using either vectors or voxels in a three dimensional space. In both cases the virtual space of the three dimensional environment is defined by values referring to a three dimensional coordinate system. Depending on individual needs the coordinate system may be different in each application. The basic principle is that an object (element) can be placed in the scene using the coordinate system of the environment. The size of the 3D scene may vary. Both small and large scale 3D scenes are using such a coordinate system in order to integrate the objects (graphics) in the scene. The coordinate system is used as a reference system of the location of each individual object of the scene.

2.2. Geographic Information Systems (GIS) and the World Geodetic System

The Geographic Information Systems (GIS) are information systems that are able to manage geographical content and organize it spatially. GIS are using a reference system in order to manage and locate content. The goal of such a system is to be able to map efficiently an area. It represents geographical content and it helps in making hypotheses and decisions. GIS have become popular since the early 1960s. GIS offers a low cost and efficient way to represent and map the real world.

The evolution of computer science boosts the development and management of geographic information systems. Based on the scale and the purpose of the GIS platform, different reference systems can be used. Large scale GIS are using the World Geodetic System as a reference system. World Geodetic System is a system that allows the definition of an object's position that is located on the surface of the earth.

World Geodetic System is used by cartography and geodesy. It provides an efficient coordinate frame of the whole surface of the earth. Each point of the surface of the earth can be defined by the two coordinates, the longitude and the latitude. The same reference coordinates system is used by the Global Positioning System (GPS). GPS is a service, developed by NASA that estimates the position of an object using the world geodetic system.

2.3. Virtual globes

As GPS devices became popular, large scale GIS platforms adapted to the global reference system. The appliance of the computer science on a large part of the geographic information systems lead to the first versions of digitals maps using the world geodetic coordinates system as a reference system. Microsoft tried to boost a digital map, which was part of the Encarta cyclopaedia in 1990s.

In 1999 Chen introduced a three dimensional urban planning system for Los Angeles [14] which was a combination of Virtual Reality (VR) technology and GIS systems. Web services have also been part of with GIS systems for urban planning in [22].

In 2003, a generic framework for dynamic environment of three dimensional modelling was presented using Java [27]. An interactive framework for public participation in urban planning was presented by Wu, He and Gong [68]. Their approach is based on AJAX and XML technology.

An important milestone on the evolution of the GIS platforms was the launch of Google Earth from Google in 2005. It defined a new type of software that it is entitled as *virtual globes*. Virtual globes are a type of software that provides to the user a friendly interface to freely navigate and explore spatial content of a digitalized GIS platform. Mostly they offer a special browser in a two dimensional or three dimensional space.

Google Earth wasn’t the first of its kind, but it was the reason that virtual globes became well known and famous to the public. In addition, NASA had released, a year earlier, a similar virtual globe platform but it did not have the same success as Google Earth.

2.4. Virtual Globe’s Basic Architecture

Google Earth, offers a three dimensional environment that allows user to browse, navigate and explore the 3D model of the scene. It is based on server-client GIS system architecture. A typical web browser is used on the user’s side. The typical scenario is initiated by the client requesting data based on user's position in the virtual scene. In response to the request, the data is placed on the server side in a GIS database and subsequently the retrieved data are visualized on client's side with the aid of a specialized application. Initial releases offered a two dimensional environment. Google Earth started publishing static maps, evolved to static web mapping applications and then to GIS distributed systems [72]. Satellite or aerial photographs have been used to provide the necessary database content [62]. Besides the images, systems of this category contain information about ground elevation, roads and points of interest as well as three dimensional building models [30]. The global positioning system is used to integrate and calibrate the geo-referenced content.

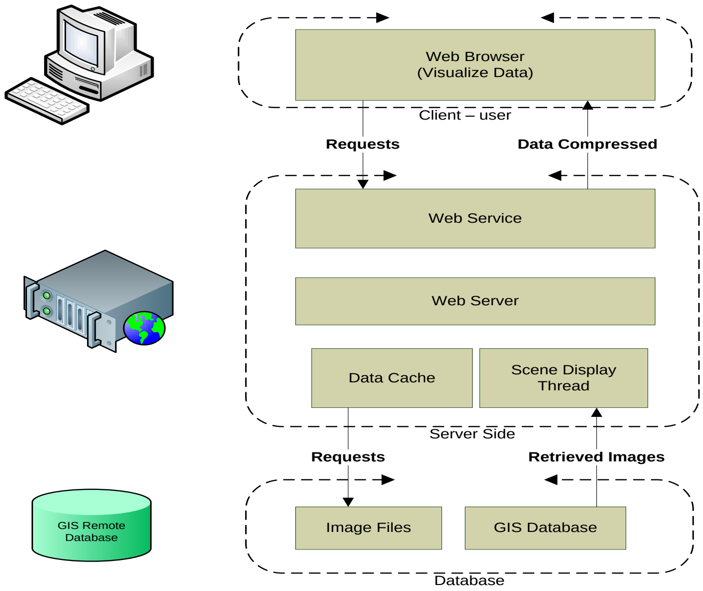


Figure 2: Block Diagram of two dimensional GIS System [69]

The basic functionality of a server-client GIS system is presented in Figure 2. This block diagram presents the functionality of Google maps, a two dimensional GIS platform.

In particular, requests from the users were submitted to the server. Communication between client and server was accomplished by using AJAX and XML technology. The user was requesting data depending on its position (longitude, latitude and altitude). The requested data were mined either from a temporary cache memory of by the geo-referenced database. The retrieved information was sent back to the user's side in compress format and it was visualized on its side. The web server was used for the communication between the client's visualizing application and the geo-referenced database.

Satellite or aerial photographs have been used to provide the necessary database content and the global positioning system has been used to calibrate the images (see Figure 2 and Figure 5).

2.5. Server – Client Functional Model

The virtual globe applications of interest in the context of this work are internet based and implement the server – client model. They are using a client application which is responsible for requesting data according to user's criteria and location. A web server is submitting the required data. The communication between server and client is achieved by web services. Real time data transmission can be accomplished by AJAX and SOAP technology [68]. Many users can request the same data or different ones, but all of them are accessing the same database. The latency between the retrieved models and data and the real-world information mainly depends on the update rate of the database but may also represent different levels of accessibility offered by the data provider where high latency characterizes free access but almost real-time access is only commercially available.

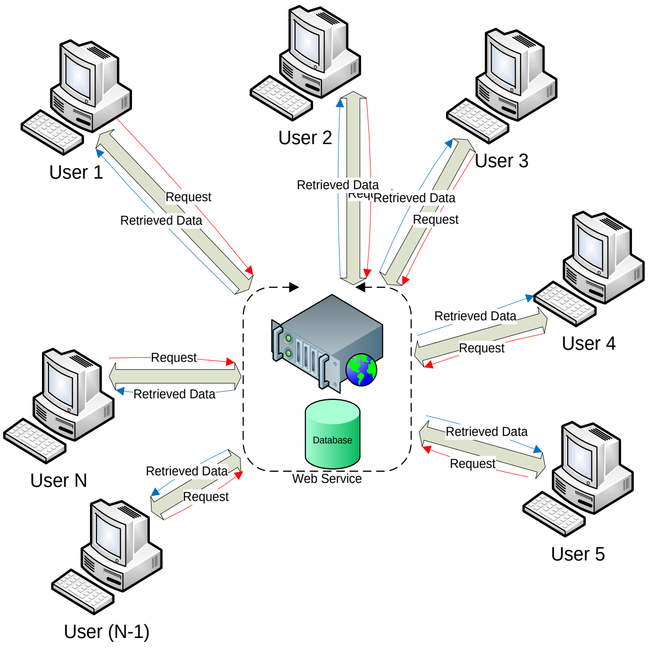


Figure 3: Server – Client model of virtual globe systems [70]

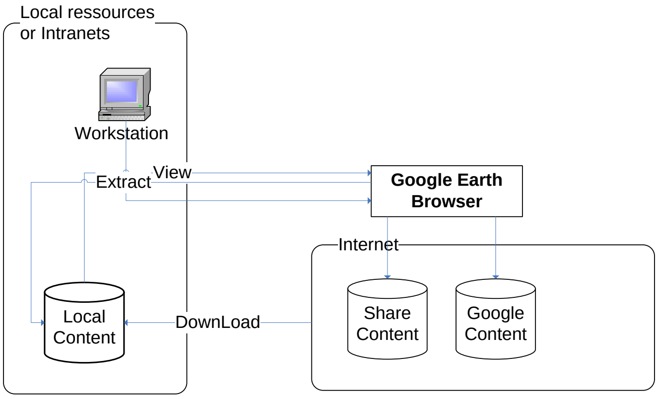


Figure 4: Functional Block of a virtual globe (Google Earth) [70]

According to the above functional block, the path of the information is only from the server to the clients, the latter only submitting requests for data to be retrieved from the server.

The latest versions of user’s side client application have the ability to cache data, or even to load external data, locally stored on user’s system. The local data offer the ability to create and maintain a local database thus enhancing through customization the experience of the virtual globe system. However, this local database can neither be shared nor it can be used to update the public database. Each user’s client can only visualize his/her own data in combination with the publicly available data.

2.6. The Evolution of GIS Architecture

The evolution of the GIS systems has led to the improvement of the services and the amount of available information. In the beginning, additional information was added to the database, mainly based on aerial photographs. Names for the streets and roads network were among the first data to enhance the database content. Afterwards, metadata such as text, external links, hypertext and place marks also found their place the database.

Despite this evolution through additional material, the fundamental functional block diagram has remained practically the same Figure 5. All data is imported to databases and calibrated over the same geographical positioning system.

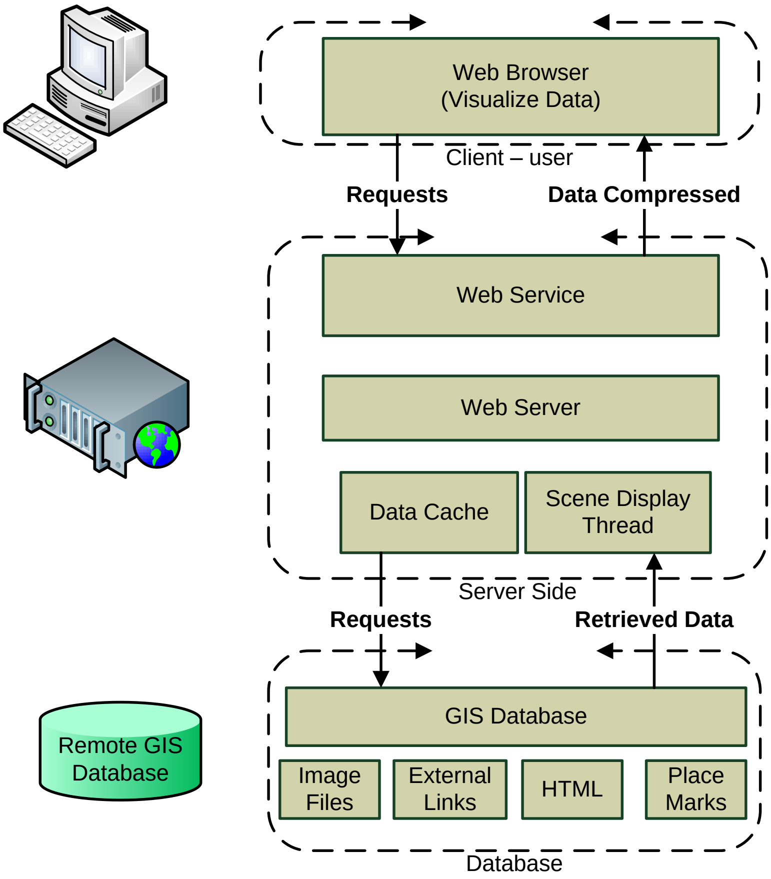


Figure 5: Block Diagram of improved GIS system [69]

In this case the data repository of the system is comprised of four different resources, each of them potentially forming a separate database. Nevertheless, all databases have to use the geographic positioning system. This architecture allows for system flexibility and easy content distribution to the extent that, for example, each database may be hosted by a different provider in physically remote locations from the others. Moreover, there is no theoretical limit on the number of physical databases that may constitute even one of the aforementioned resources. The end-user may choose which information to retrieve from the GIS system through the client application, which is presenting the data organized in intuitive layers, completely transparent with respect to the actual data organization. Each layer typically contains data from one or more different databases. When all the data are being presented, a synthesized image is created from the projection of the superimposed layers on the same geographic area.

Evolution led to three dimensional GIS platforms. Evolved platforms offer browsers with 3D capabilities and free exploration of the scene. Beyond the images and hypertext, new browsers can visualize the morphology of the ground. Moreover, 3D models of buildings are also included in some browsers (see Figure 6). The result is a synthesized three dimensional scene that can represent a three dimensional model of an area or city with realistic or reality-enhancing attributes. Initially, models of buildings were limited and only specific cities were modelled.

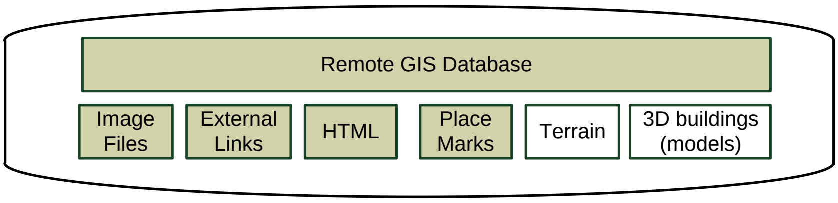


Figure 6: Geographical data base of GIS system

3D models were designed and built in a 3D object description language called COLLAborative Design Activity (COLLADA). It has a language with XML schema and it has been adapted by many software and hardware developing companies.

On the user's, side a standalone application or web browser with visualizing capabilities of COLLADA is required. Components have been developed by distributors for this purpose using OpenGL or DirectX graphics languages.

Today, there are numerous databases that can provide information and data for such a GIS system. The popularity of such systems and the advancements in computing and telecommunications have made possible the creation of powerful three dimensional GIS systems, based on the aforementioned architecture. Information about the altitude and the morphology of the ground has been incorporated to them whereas 3D models of buildings are also being gradually added. The goal is to create a virtual world that closely simulates the real one. Recently, many 3D models of buildings have started to appear on most popular capitals and cities around the world on the virtual globe systems. Each virtual globe application typically features its own independent database and is usually supported by its own developing team. Many cities have been modelled but most of them are still incomplete or not modelled at all.

2.7. Recent GIS Platforms

During the recent years many databases with geo-referenced content have been released. There is no limit on the data that can be added. There are GIS databases that are freely available for public use, whereas some others are not.

Data are requested based on user's position and preferences. Users can filter the data using the GIS platform interface. Data might be organized in dedicated databases. The user's application is presenting the data organized in layers. Each layer contains data from a different database. When all the data are being presented, a synthesized image is been created from the projection of the layers onto the same geographic position system. The result is a synthesized three dimensional scene.

2.8. GML

Most of the popular virtual globes applications are using layers in order to organize and manage the spatial data geographical features. Spatial data can be defined by a special markup language that has been developed for this purpose. This special markup language was developed by the Open Geospatial Consortium (OGC) in order to define spatial content of a GIS platform. This language is well known as Geography Markup Language (GML).

Geography Markup Language (GML) is a simple text based description language. It is based on a XML schema. It has been adapted by almost all the virtual globe applications. It is used to model and format geographical content. The main features of GML are that it provides a generic schema of transferring geographic information over the server – client connection and that it is able to model any possible geographic information.

2.9. KML

Most of the known virtual globes have either adopted GML as it is, or created a new based on it. One of them is the Keyhole Markup Language (KML). This implementation of KML was presented by Google Earth. Later, it became one of the standard and popular distributions of GML. At the moment, it has been adapted by most virtual globes platforms.

KML has the ability to define vertices, bounding boxes and other simple geometrical figures. All of them are defined using a coordinate reference system. Besides the geometrical figures, KML is able to control the viewing angle and the camera position of the virtual world of the GIS platform. In addition, KML has been enriched in order to support Google Maps and Google Earth functionality. It is able to control animation, place marks and all geographical data that are necessary for the aforementioned applications.

One of the main features that it is used from Google Earth is that it can integrate 3D models of obstacles such as building and 3D morphological model of the ground (terrain). Three dimensional models are been defined usually by COLLAborative Design Activity (COLLADA).

2.10. COLLAborative Design Activity (COLLADA)

COLLADA is an acronym of the Collaborative Design Activity. It was established in order to bridge various graphics applications of incompatible graphic types. It is based on an XML schema and it provides an interconnection between different graphics technologies.

Initially it was developed by Sony. The goal of it was to transport data from a digital content to another application. Several companies and distributors have adapted Collada. Many 3D CAD applications are using Collada as a media to interchange 3D models. Moreover, Collada is used in order to transform and render computer graphics at the hardware level from some companies that design computer graphics cards.

Collada architecture is able to host semantic information about the 3D model of the object. Its structure is organized in sections. Each section of the Collada schema has different information. Besides the basic information of 3D model that Collada schema contains, it can also include images, visual effects and environment lights. Moreover, Collada schema is able to model semantic information relative to 3D model. Last but not least it can include information of geometrical structure and the contracted material. This information can be valuable for 3D animation of the 3D model. Later versions of Collada schema are able to model animation too.

Collada schema continues to evolve by the Kronos Group which has madethe Collada schema and its implementation freely available to the public.

3. Relative Artificial Intelligence Techniques

3.1. Machine Learning

Machine learning is one of the fields of artificial intelligence. The scientific area that tries to simulate the aforementioned human's conceptual function of gaining knowledge from the environment, using a machine, is the machine learning. For a human or an animal the process of learning is innate. In case of the machine complex structures and algorithms are used. Machine learning is a research area that constantly evolves.

The goal of machine learning is to make a machine able to gain information in an efficient way from the environment. The gathered information has to lead to understanding of the data. The final goal is to develop machines that are able to make decisions or predict cases based on input data.

The machine learning is dealing with two issues. The first one is the proper representation of the collected data and the second one is the approach that will be used in order to extract the required knowledge of these data. The second process is mostly followed by ordering information or by an evaluation indicator of the gained information.

Data representation and gathering is the most important process of machine learning. Data can be gathered from the environment with an automatic way or inquiring information from human experts or users of the system. The input data can be either alphanumerical or numerical discrete values such as data from a sensor or a database. On the other hand, the input data can be a preference (choice) of an object against others. The nature and the philosophy of the described problem are often suggesting the format of the input data.

3.1.1.Decision Support Systems

In most of the cases the goal of the machine learning process is to create a Decision Support System (DSS). A system that is supported by machine learning features, using some techniques and algorithms in order to achieve its goal. The nature of the problem will define the most appropriate algorithm to be used. There are two types of problems that a DSS is called to solve, the deterministic ones and the non-deterministic ones. In the first case DSS is trying to choose one possible solution (one case) within a well-defined and limited search space. The DSS will provide a solution by choosing one possible solution of the whole set of solutions.

On the other hand, in the case of non-deterministic problems, the DSS is called to predict or to estimate an optimal solution within an infinite search area. Problems of this category are harder to be solved and the estimation solution can be trapped in local maximum. There are problems that can be solved as deterministic and non-deterministic approaches.

3.1.2. Pairwise Comparison

Pairwise method was initially used for scientific purposes by psychometrician Thurstone [63] as a comparison measurement. Saaty has used pairwise comparisons as a part of his Analytic Hierarchy (AHP) [54] and Analytic Network Processes [55].

Pairwise comparison method is used in multiple criteria decision support systems. Pairwise comparison method is based on the principle that a set of factors or criteria are required to lead to a decision.

Therefore, the input vector of data is comprised by multiple factors (indicators). These indicators describe each possible solution. Pairwise comparison is used on deterministic problems. According to this approach, each possible solution is compared and evaluated against all the others. In this way, the most appropriate features (factors) are highly marked as both the features and the objects are ordered based on user’s preferences.

The pairwise comparison method achieves to gather information about the order of the factors and of the objects in an efficient way. This method is friendlier to the user rather than defining the significant of each factor. It provides an easy method of ordering by choosing on object against the others.

The main advantage of the method is that the user is not called to order the features but to choose in a more abstract form, by the image of an object. This process comes to fill the gap between the real object and the representation of it with the features of the DSS. Moreover there are cases that the features that are used by the system are unknown to the user. In addition there are cases that even the user he cannot define the difference between factors of two objects. On the other hand a representation of them by the whole object, in our case by a 3D model, can reduce the error to minimum, by using the pairwise comparison method.

This aforementioned approach has been used on this research. Each 3D model of the scene is defined by a vector that is consisted by a set of features (factors). During the online inductive learning process a random set of ten 3D models appear and it is evaluated by the user. All possible combinations of comparison between 3D models are taking place. The user can interact with the image of the 3D models only and by the support of the pairwise comparison the order of the features and of the 3D models is achieved.

3.2. Genetic Algorithms

Genetic Algorithms are essentially a search mechanism. It is an instance of evolutionary computing. Rechenberg introduced evolutionary computing in 1960s and developed further in 1970s [46]. The main idea of Genetic Algorithms was introduced by John Holland, his students and his colleagues [26]. Based on the principles of Genetic Algorithms, later in 1992, John Koza implemented Genetic Programming (GP). It was a method to evolve a program in order to perform specific task.

Genetic Algorithms are inspired by the Darwin’s theory of evolution. Genetic Algorithms are used to provide an efficient and robust solution. Genetic Algorithms simulate the natural evolution of life.

Genetic Algorithms search for optimal solutions in a well-defined solution space. Each possible solution is represented by an artificial creature with the proper chromosome (gene). The chromosome of artificial creature describes one possible solution. The chromosomes are evaluated and only the fittest creatures can survive. The evaluation of the artificial creatures is taking place by an evaluation process through the fitness function. The artificial creatures are used to create offsprings. The same process goes on with offsprings at the same way that Darwin’s evolution theory implements. At the end, only the optimal chromosomes can survive, thus providing an efficient and robust solution of a declared problem.

3.2.1.Chromosome – gene

Each possible solution of a well-defined problem can be encoded in a string formula. The most appropriate encoding that can be used depends on the problem that has to be solved.

Genetic Algorithms are simulating the natural evolution of life in computer science. Each possible solution of a well-defined problem can be encoded in a string formula. The most appropriate encoding has to be used in order to lead to an efficient solution.

Binary encoding is mostly used, because it is the most suitable and efficient for Genetic Algorithms. In this case each chromosome is represented by a set of zeros and ones (bits). Using the aforementioned encoding each bit or set of bits represents information. In this case all possible combinations of bits have to represent a possible solution.

On the other hand, natural numbers can be used to minimize the gap between the chromosome representation and the real problem. This representation can be applied to model *permutations*. According to this methodology, problems such as ordering or estimation of travelling path can be represented easier. Problems such as travelling salesman's or ordering tasks can be solved faster and with more efficacy by using this type of chromosome representation. For instance, the chromosomes might represent the selected points and the order of them in path planning problem.

3.2.2.Path Estimation Using Genetic Algorithms - Chromosome Encoding

In our research the path is represented by the sequence of the visiting points. Each possible 3D model of building or 3D obstacle of the virtual word has an identification number. Moving forward, the inactive online training process the platform is in position to classify the 3D models of the whole scene. Each one of them might be a possible visiting point. The classified 3D models that meet the user's preferences and desires have to be considered by the Genetic Algorithm. The chromosomes are representing possible visiting points of the path into the 3D virtual scene with the proper order.

The next chromosomes represent two possible solutions (see Table 1). In case of a path planning problem, that each number represents a visiting point of interest and the order of them is defined by the sequence of them in the path. In the first solution the path will contain the visiting points 1, 5, 3, 2, 4, 7, 9, 8. Initially the 3D model with the identification number *one* will be visited and then the 3D model with the identification number *five* will follow. The visiting sequence will continue as it is described in the order of chromosome A. SP is marked as the Starting Point and it can randomly be defined or selected by the user. The SP doesn't contain any building (visiting point).

Chromosomes encoding	
1. Chromosome A	SP 1 5 3 2 4 7 9 8
2. Chromosome B	SP 8 5 6 2 3 1 4

Table 1: Simple path planning chromosomes encoding

On the other hand, the chromosome B describes a solution with less visiting points and different order. Both solutions are valid and might be good under different user preferences. In the second case, there are less 3D models (visiting points - buildings) that meet the user's preferences.

According to our implementation each valid chromosome has to contain the same number of buildings, as it has predefined, otherwise is invalid. The number of buildings of each possible solution is defined by the user before the beginning of the path estimation process.

3.2.3.Fitness Function

Fitness function is the evaluation measure of the genetic algorithm's process. It provides a measure of the quality and efficacy of the solution described by each chromosome. The fitness function is a procedure that is able to produce a score about the quality of the chromosome. Under known circumstances the score of the "best" solution is calculated. This score is used as the optimal upper limit of all the possible solutions. The genetic algorithm continues to look for an optimal solution up to the time that fitness function estimates either the score of the optimal solution or close to that number for a solution described by the chromosomes. By that time the provided chromosomes are considered as the optimal solutions for the provided problem.

For this purpose, an error indicator is typically used. The indicator error is used as termination threshold of the process. It is assumed that the genetic algorithm has researched its goal and has an efficient solution when the square error is minimized and it reaches a value under a minimal threshold. In other words, the genetic algorithm is trying to minimize the square error calculated by the fitness function.

Fitness function defines the evaluation of the chromosome. According to the simple problem of travelling salesman, the fitness function might be represented by the total length of the path of the visiting points sequence. A detailed implementation of the fitness function that has been used for this work can be found on Chapter 8, in paragraph 8.2.

3.2.4. Genetic Algorithms

Genetic Algorithms is an area in Artificial Intelligence. A Genetic Algorithm is an extensive search algorithm. Initially, it starts with a set of random possible solutions. Each set includes a set of chromosomes (solutions), which is called initial population.

Solutions from one population are taken and used to form a new population. This is motivated by a hope, that the new population will be better than the old one. Solutions which are then selected to form new solutions (offspring) are selected according to a fitness function which provides a score. The higher score the better solution. In the following Table 2 a functional overview of the genetic algorithm will be represented.

Basic principle of Genetic Algorithms
1. [Start] Generate random population of n chromosomes (suitable solutions for the problem)
2. [Fitness] Evaluate the fitness f(x) of each chromosome x in the population
3. [New population] Create a new population by repeating following steps until the new population is complete
4. [Selection] Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to be selected)
5. [Crossover] With a crossover probability cross over the parents to form a new offspring (children). If no crossover was performed, offspring is an exact copy of parents.
6. [Mutation] With a mutation probability mutate new offspring at each locus (position in chromosome).
7. [Accepting] Place new offspring in a new population
8. [Replace] Use new generated population for a further run of algorithm
9. [Test] If the end condition is satisfied, stop, and return the best solution in current population
10. [Loop] Go to step 2

Table 2: Functional block of Genetic Algorithms

During the crossover process the chromosomes are reproduced, generating the new population. This process is taking place with special rules based on the nature of the problem and the chromosome encoding. It is inspired by nature.

In general each chromosome can be separated in different parts. During the crossover process each chromosome can be split at least into two parts. Combination of parts from two different parents can lead at least to two new chromosomes (offsprings). In the next figure several different types of splits are being represented.

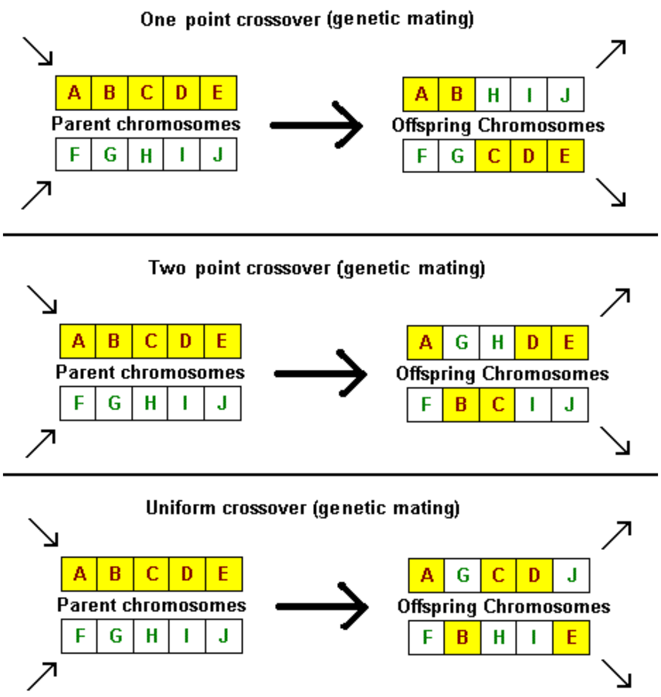


Figure 7: Chromosome crossover

In the first example, the first new chromosome will gain the first yellow part from the first parent and the second part for the second parent. In similar way, the second offspring will be produced by the combination of the first part of the second parent and the second part of the first parent. Furthermore, the crossover process can use three parts of each chromosome or more as it is presented in the examples of Figure 7. Every time, two chromosomes are being used in order to produce two new ones. In this way, the number of each population is always the same in each generation. This process is the core of the genetic algorithm. It has to take place in a way so that each newly produced chromosome represents a valid solution. In different case, chromosomes have to be killed and new ones have to replace the defective ones.

Last but not least, it is the mutation process of the genetic algorithms. The Darwin's theory of evolution is based on this principle. After the crossover process the mutation process is taking place by modifying some of the offsprings, in a random way. After the mutation process the chromosomes might have to be checked if they are still valid. Mutation process is the most important process of the genetic algorithm. Minimizing mutation the solution might be lead to local maximum.

On the other hand, maximizing mutation might be lead to defective solutions and ineffective genetic algorithm. The better estimation of the mutation process, the more efficient and faster genetic algorithm will be produced.

Genetic algorithms can't always provide an efficient solution. Genetic algorithm mechanisms can provide an efficient solution but it is not always the optimal solution. The approach of the *optimal* solution is depending on the size of the population, the mutation indicator and the number of the generations. The higher the size of the population and the generations, the larger is the possibility to reach the *best* possible solution with the genetic algorithm.

4. State of the Art

This research project is dealing with navigation into three dimensional large scale scenes. Two main architectural components are involved in a platform dealing with large scale navigation of 3D geo-referenced content; a) the route and path planning (it is entitled as “macro-path” process, and b) the best viewing algorithms (it is entitled as “micro-path” process). Although several algorithms exist in the literature for solving the aforementioned two individualized aspects (see below), according to my best knowledge there is no work that either integrates these two components into a singular platform or integrates conceptual and geometrical criteria in a singular platform in order to provide an efficient personalized exploration and understanding of a geo-referenced 3D model.

Furthermore, a personalized route planning architecture requires two more components such as the metadata (either textual or visual) used for describing the rich geographical content and the learning algorithms used for estimating the users’ preferences dynamics. Still the integration of the aforementioned four components for an efficient 3D virtual navigation in geo-referenced data is limited. In the following, some methods used for solving the aforementioned individual problems are briefly described.

4.1. Geo-referenced Systems

One well-known approach is using a topic that represents the context of the image in order to classify geo-referenced images. The topic produced by a probabilistic latent semantic analysis (pLSA), a well-known technique in the text-mining community. Recently vocabulary was extended to describe, in a compact way, the contents of an image database [9].The input of the probabilistic latent semantic analysis is a set of images and their location. The analysis classifies the images [17].

In the field of geo-referenced databases, there are many distributed systems that have been developed. Most of the approaches are using a standard database schema with some extra geographical data for each object. On the other hand, the first implementations of spatial query languages were in 1980s. Borrmann Andre and Rank Ernst have developed a 3D spatial query language for building information models. The language is based on directional relationships among the buildings. The relationship between two buildings was calculated by the euclidean distance of the centres of the objects and the relative position of them, such as *northOf, southOf, eastOf, westOf*, above and below [8], [9]. Recently, the vocabulary of the language was extended and topological relationships were added such as within, contain, touch, overlap, disjoint and equal [8], [9].

4.2. Generation of 3D City Models

There are several approaches on 3D city models generation. It can be accomplished either by reconstruction or by generation process based on rules.

According to the first method, an efficient approach has been implemented using photogrammetry. A set of sensors are scanning the building and the scene [33]. The representation of the model is very accurate but experts, special equipment and heavy computational power are required.

Another approach is based on LIDAR (Light Detection And Ranging) [36]. It is an optical remote sensing technology that measures properties of scattered light to find range and/or other information of a distant target. The prevalent method to determine distance to an object or surface is to use laser pulses.

On the other hand the 3D model can be generated by an engine. A declarative engine for generating a 3D large scale scene was introduced by Larive & Gaildrat[37] – (see Figure 8).

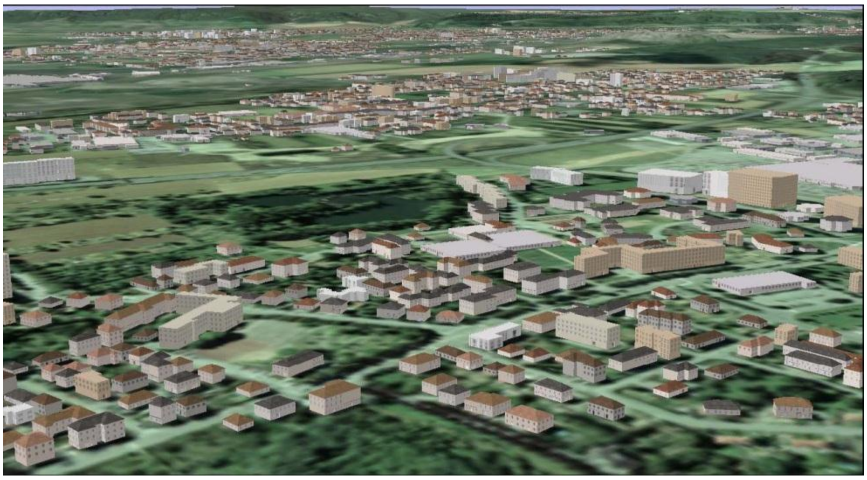


Figure 8: 3D Large Scale Scene of 17.362 buildings [37]

In addition several commercial applications have been presented in 3D reconstruction of geo-referenced scenes. They can generate the model based on rules and on image recognition of satellite and aerial photographs.

4.3. Wiki-mapia Platforms

Besides the three dimensional GIS systems, there are examples of collaborative GIS systems using two dimensional environment. These approaches are known as wiki-mapia platforms. They permit the user to interact with a two dimensional map. Consequently, all the users of the platform are using the same resources, i.e. the same map. The user can add geo-referenced content on the map which is instantly shared to the whole community of the platform. Some examples which support these platforms can be multimedia, place markers, or hypertext content. The wiki-mapia platforms function over web services and typically use AJAX and XML technologies to communicate with the user. The majority of the most known platforms are using the API of known GIS systems for the visualizing process.

4.4. Route Path Planning

Moving forward to the scene exploration some well-known route planning algorithms are Dijkstra's algorithm [19] and the A* algorithm [25]. Improvements of these algorithms have been introduced in literature. Although A* or Dijkstra may be good solutions for a large number of isolated problems, they should not be the starting point of the development of a navigation system [41]. Both A* and Dijkstra increase with the size of the search space [6].

Demyen and Buro implemented two more recent hierarchical triangulation-based path planning approaches, namely Triangulation A* (TA*) and Triangulation Reduction A* (TRA*) [18]. TA* exploits the Delaunay triangulation to build a polygonal representation of the environment in order to reduce the path-finding search effort. On the other hand, TRA* is an extension of TA* that abstracts the triangle mesh into roadmap structures [41].

Rogers and Langley implement a personalized driving route recommendation architecture based on the Dijkstra's shortest path algorithm, operating on personalized metadata that are approximated by a weighted sum of factors (such as driving time, length, number of turns, and number of intersections) [51]. In addition, Mekni and Moulin present a multi-agent implementation for path planning able to be solved in real time, even under constraints of limited memory and CPU resources [41]. In [40], the route is planned based on user's defined points of interest, which are manually set, on 3D virtual workspaces. Then, the route is controlled by a set of parameters and a weight is assigned to each of them in order to regulate their degree of interest. The parameters are either pre-defined by the system or set by the user [39]. Alternatively, the weight values can be defined by constraint satisfaction techniques [5]. Other approaches exploit Genetic Algorithms (GA) in regard to estimation of the best route [1], with the focus, however, being on the Field Programmable Gate Array (FPGA) implementation of the GA in case of real-time controlling Unmanned Aerial Vehicles (UAV's). Another method is the Analytical Hierarchical Process (AHP) that allows a systematic evaluation of multiple criteria, resulting in a hierarchical structure that can help in personalized routing [54].

4.5. Best Viewing Algorithms

The exploration of a virtual environment on behalf of the users can be classified in three distinct categories: (a) navigation, presenting data from the surface of the world, (b) presentation of an object or a path and (c) presentation of an object or item from specific views in order to reveal as much information as possible [63].

In the area of best viewing algorithms of individual objects or scenes, several approaches have been implemented in the literature. An efficient approach for the calculation of the best view of a 3D scene has been presented by Sokolov and Plemenos [58],[59],[60],[61]. It is based on the coverage of the surface of the scene. Similar methodologies have been implemented by Tab and Vazqueq [63],[66]. All these approaches exploit optimization strategies so that the best projected 2D views are selected in a way to cover as much as possible of the whole 3D surface information of the scene. Each view is evaluated individually and then the best views are selected for the trajectory of the camera.

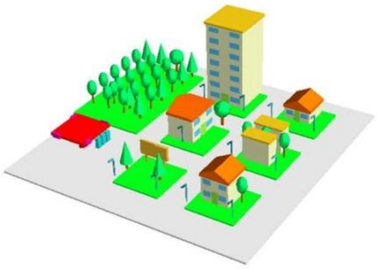


Figure 9: Initial urban scene

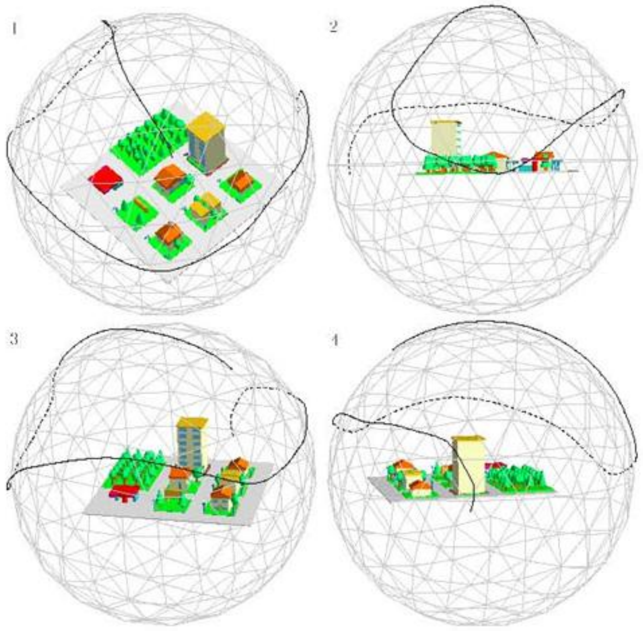


Figure 10: Camera trajectory of an urban scene (Sokolov & Plemenos)

Each projected view is independently evaluated based on its complexity, measured as the total number of faces projected onto the 2D plane. In particular, [58], [72] optimizes the area of the visible surface over the sum of the area of the surface of the whole scene. Instead [66] introduces an entropy metric that measures the projection complexity; views that maximize this information are selected as the best ones. The angle of viewing an object in a three dimensional space requires five variables to be defined. The first three are used to define the position of the camera on an orthogonal 3D coordinate system and the other two are used for the horizontal and vertical orientation of the angle of view. All possible points of views have been marked on the figures, where the points of the net of a virtual sphere are shown [61].

In particular, Vazquez presented in 2003 a methodology for quality measurement of a certain view of a scene. It was based on information that can be extracted from a specific aspect of view of an object [66].

Moreover, Mackinlay introduced another approach of path planning and navigating in a 3D scene. It is based on points of interest [40], [52], [67]. The path is planned considering the user’s desired points of interest. The path has to be calculated based on a set of parameters. Each parameter is defined by a weight or a degree of interest [39]. Measuring the various parameters and their weight, it is possible to optimize the information that can be gained following the designed path. The parameters may be pre-defined by the system or defined and evaluated by the user. The evaluation can be done in several ways. A weight value or a degree of interest can define each point of visit, representing the importance of it, compared to the others [5].

Finally, methodologies of moving the camera on a virtual sphere have been published. In these approaches the object can be watched closer, either by user's intervention or by the system automatically [63].

Each of the aforementioned approaches is using either geometric criteria or semantic. The combination of semantic with geometric criteria is limited on the known approaches according to the literature review.

4.6. Semantic Metadata

In the field of personalization a system must be able to capture user's preferences and then to satisfy his needs in a known content [44], [49]. In this study, an ontology based framework is presented for modelling geographical, user as well as contextual information. Some recent works use ontological models for personalization [28]. Ontologies have been developed using artificial intelligence to facilitate knowledge sharing and reuse and to establish a domain of spatial information [4], [28], [29].

An ontology-based system allows experts to cooperate with the system, resulting in a list of criteria related to the goal of the personalized modelling. From a criteria combination point of view, analytical hierarchical process (AHP) allows systematic evaluation of multiple criteria, resulting in a hierarchical structure that can help in dealing with the characteristics of personalized preferences [29], [47], an approach that has already been successfully applied in a 3D Scene Modelling environment [3].

The work of Nadi and Delavar gives emphasis on the algorithms for creating a personalized routing plan based on a set of multi-criteria derived from user preferences and geographical information [43]. The algorithm was an online weight allocation algorithm, much like the weighted majority algorithm [15], [38]. The last two algorithms are based on "Hedge Algorithm" which was presented by Freund and Schapire in 1997 [23]. It is known that the best way to gain information about the user’s preferences is simply to ask for them [32].

4.7. Image Complexity

Image complexity is relevant to the time that an ‘observer’ requires in order to understand the content of the image. The more complex an image is, the more time is required. Image complexity measure can be defined upon the semantic and the syntax of the image [45].

According to Prkio and Hyvarinen a natural measure of image complexity can be defined by the entropy of the discrete probability distribution of Shannon [45],[57].

A well-known approach of measuring image complexity is based on edge level of image [10],[11]. According to literature review the complexity of an image can be measured with a fuzzy based approach and it can be classified in discrete values such as ‘Little Complex’, ‘More or Less Complex’ and ‘Very Complex’ [10],[11].

4.8. Related Work

The current personalized route planning systems use a set of explicit metadata (usually textual) to describe user’s preferences. Then, they apply an optimization procedure that involves several multi-criteria factors to create the personalized route [43], [44]. However, two main limitations are encountered: (a) textual metadata are not able to describe the rich geographical content [21], [53] while (b) users’ preferences cannot be modelled through a simple set of keywords [53]. Keywords are also inconsistent since they are dependent on the indexer. To address the aforementioned difficulties, the work of Niaraki and Kim introduces an ontological framework to provide a semantic description of the geo-referenced and contextual information, such as tourist attraction, road safety issues (telephone, medical assistance) and road facilities (gas station, services, terminal), etc. [44].

However, the work of Niaraki and Kim focuses only on the ontology representation and thus it fails to provide online learning strategies, which are necessary for a truly personalized route construction [44]. Another approach introduces symbolic representation which is combined with visual properties of the objects, extracted through the application of image processing tools, to provide a cognitive spatial path planning [48]. However, again, this work is focused on a better organization of the metadata (textual or visual) that describes the geographical content, instead of applying personalized strategies able to capture the dynamics of the user’s preferences.

In [43], a generic model is proposed to calculate the most probable user preferred path using historical data, as a prior path planning, and an online strategy of time dependent routing selected using an approach of multiple criteria. The model integrates a pairwise comparison method and quantifier-guided ordered weighted averaging (OWA) aggregation operators to form a personalized route planning. The main differences of the aforementioned research approaches with the one presented are:

1. the use of a 3D navigation that enables users' personalized 3D experiences,
2. the relevance feedback scheme, as the online machine learning algorithm, which automatically adjusts the metadata weights of the 3D objects based on an active inductive approach that exploits information fed back to the system through user’s interaction about the relevance of preferences judgments between pairs of two objects,
3. the ontology driven weight rectification strategy that improves the weight estimation process, especially in cases that a small number of preferences judgments has been fed back to the system by exploiting semantic relationships among the objects and the route metadata,
4. the genetic optimization algorithm used to calculate the most probable user preferred path based on a maximization of a combination of cost functions that describe objects’ properties, user’s preferences and overall route characteristics,
5. the introduction of a personalized entropy metric to estimate “best views” (preferred views) for selected 3D objects in an itinerary by simultaneously combining geometric constraints and user-centric criteria,
6. the integration of the aforementioned innovative aspects in a single platform in order to offer the aforementioned facilities in real-world cases.

Part II: Thesis Contribution to Semantic Navigation of 3D Scenes

5. Thesis Contribution to Semantic Navigation of 3D Scene

5.1. Motivation

The recent advances in information and communication technology boost research towards the generation of personalized Geographical Information Systems (p-GIS) [12],[43],[44],[49] that allow for the users to precisely plan their trips and enjoy their stays in a place fully respecting their preferences [32]. In a p-GIS the "best route" is not defined only based on geometrical criteria, such as the shortest distance or the travel time, but it should mostly exploit user's contentment, resulting in an itinerary that fully respects users' preferences (navigation through buildings or places of user's particular interest) [12],[43]. Most of existing personalized route guidance systems exploits a two dimensional (2D) geometric and/or thematic representation; they estimate proper routes consisting of places/buildings of user's interest (thematic representation) within a near to the current user's position distance (geometric representation). However, currently three-dimensional (3D) navigation systems gain more popularity due to the fact that they provide the natural way for virtual touring [35], [63]. It is true that we are living in a 3D world and we perceive most of the events and activities occurring in it based on depth information. This boosts a series of new applications in the field of tourism, leisure and entertainment, cultural heritage, etc. [47]. It is argued that 3D virtual tours offer something more that the ordinary 2D ones. For instance, they can potentially explain a story or even show up the space [34]. Although 3D navigation imposes higher computational complexity compared to the 2D approaches, the current technological evolutions, both in hardware and software, make feasible the implementation of 3D geo-information systems (see for example Google Earth and Microsoft Virtual Earth). This is the main reason, why we choose to focus on 3D navigation systems in this work.

In a p-GIS, user's preferences can be set manually or automatically [3]. Systems that apply the manual selection are called "adaptable personalized route guidance systems", while systems that use the automatic method are called "adaptive personalized route guidance systems" [73]. For both cases, the first necessary component is to extract a set of features (metadata) able to describe either the overall properties of a route (e.g., a small scenic road in contrast with a high speed freeway) or characteristics of places/buildings (in this work we call them objects) comprising the route . Then, a set of weights are assigned to regulate the degree of importance of the extracted metadata on route selection to provide personalization. For example, in case that a user is interested in an itinerary that includes buildings of modern architectural style, the weights that correspond to metadata of modern style take higher values than the remaining ones.

In an adaptable personalized route guidance system (i.e. the manual case), weight assignment is a quite difficult process; users should appropriately associate their actual preferences with the extracted metadata, used to describe the objects and the scene, forcing the complexity to the user’s side. However, usually, there is no a direct relationship between user’s preferences and metadata. On the contrary, adaptive personalized route guidance systems (i.e. the automatic case) incorporate online learning strategies to discover common patterns in repeated relevant (or irrelevant) previously selected routes. This way, the weights are automatically estimated forcing the complexity to the system side. In this work, we focus on second case.

A Relevance Feedback scheme is used in this work as the online learning strategy [54]. Relevance feedback is a powerful machine learning method, initially applied in traditional text-based information retrieval systems [50], to automatically adjust metadata weights by exploiting information fed back to the system about the relevance of previously selected routes [13], [21]. The relevance feedback component is combined in this work with an ontology driven weight rectification module that improves the weight estimation process, especially in cases that few relevant/irrelevant routes have been selected by the user, considering metadata relationships.

Metadata weights are then used to select personalized itineraries that fully respect user’s preferences. This is achieved by incorporating efficient optimization strategies, able, on the one hand, to suggest the most preferred routes based on maximization of multiple geometric, thematic and user-centric cost criteria (called macro-path optimization layer) and, on the other, to estimate 3D trajectories around objects of interest (called micro-path optimization layer).

Therefore, the research assumptions made in this work are the following;

(a) each 3D object and route is described through a set of metadata (features),

(b) users' preferences are modelled through a set of weights that regulate the degree of importance of the scene metadata on the route selection process,

(c) a relevance feedback machine learning strategy is adopted to automatically adjust metadata weights,

(d) an ontology-driven weight rectification mechanism is introduced to improve the weight estimation process and finally,

(e) optimization strategies are incorporated either to suggest the most preferred routes with respect to user's information needs or to recommend best views and 3D trajectories for the objects of interest.

Based on the above, the main components of our system are the following:

The Relevance Feedback Component: In this work, an inductive machine learning algorithm is used to implement relevance feedback. We have selected this strategy since it offers minimum user's interaction (we do not force the user to get familiar with technical knowledge of the system). In particular, the system depicts to the user a set of pairs of two objects and then the user feeds back to the system information concerning judgments of preferences between the two of them. Then, the inductive process is activated to automatically adjust the metadata weights.

The Ontology driven Weight Rectification Component: One main difficulty in implementing the online learning strategy concerns the number of pairs required to be submitted to get a reliable estimation of user's preferences. To reduce the number of comparisons needed and thus the number of user's interactions, in this work, we introduce a novel weight rectification algorithm that exploits relationships among the extracted objects' metadata.

The Route Planning Component (Macro-Path Layer): The goal of this component is to select the best preferred route for a user by taking into consideration metadata weights as well as several geometric and thematic criteria. In regard to optimization, a genetic algorithm is employed. The genetic scheme provides a solution much closer to the optimal one, faster than other conventional optimization approaches. Additionally, the genetic approach provides scalability in contrast to the traditional shortest path methodologies that always yield the same solution. Instead, in the genetic approach, we can improve route selection performance at the cost of demanding more computational resources.

The Best View Selection Component (Micro-Path Layer): This module selects best 3D trajectories by merging projected 2D views of selected (preferred) 3D objects within an itinerary. A personalized entropy metric is considered that takes into account, apart from geometric objects properties (the volume that is projected from the 3D objects onto the 2D plane), the degree of importance of the projected 3D faces, as expressed through the respective metadata weights, onto the 2D plane.

5.2. Contribution

In this research, a 3D personalized route planning architecture is proposed. The architecture automatically suggests to the user the most suitable itinerary for navigating into 3D worlds that fully respects user's preferences. To accomplish this, we initially extract a set of features (metadata) from 3D objects and the route to describe their architectural, contextual and environmental content. We also conceptually relate these semantic metadata through an ontological schema. For each feature, a weight value is assigned to regulate the degree of importance of the respective feature on route selection. Then, we introduce an online learning strategy to automatically estimate the metadata weights in a way that reflects user's preferences. This is achieved through the application of a relevance feedback machine learning algorithm. Finally, an optimization process is adopted either to estimate the best preferred routes based on a combined maximization of several cost functions that reflect geometric, thematic and user-centric criteria or to recommend a set of 2D views from the selected 3D objects within the itinerary that simultaneously optimize geometric constraints and user's preferences.

The main contributions of this work are summarized in the following:

1. In contrast with the current approaches dealing with personalized route planning, our architecture incorporates an active inductive learning methodology able to fill the gap between computers and humans by allowing high level concepts to be modelled in the

future GIS decision support systems. Therefore, the proposed framework permits modelling of complex user's preferences through the use of metadata extracted from low level features.

2. Preferences are implicitly extracted without imposing users to be aware of any technical details regarding the metadata used in the geo-referenced 3D environment. In particular, we adopted a relevance feedback machine learning strategy for ordering 3D objects based on semantic features of a 3D scene. This method allows an automatic adjustment of metadata weights, used to regulate the degree of importance of the features, by exploiting information fed back to the system in a form of preferences judgments of pairwise comparisons. Then, we apply the pairwise comparisons to get an overall 3D object ordering in the scene.

3. Another contribution concerns the framework under which we use the ontological modelling. In this work, we adopt the ontology not only for providing a human perceptive concept modelling of the properties of the 3D objects as the current approaches do, but we exploit metadata interrelations in order to estimate in a better way user's preferences, through a small number of training samples. This is achieved by introducing a rectification strategy that corrects the weight estimation process by exploiting the knowledge provided through the ontology for semantic navigation in 3D scenes.

4. Optimal route planning is accomplished based on a multi-criteria framework that extends conventional shortest path methodologies enabling involvement of complex cost functions while simultaneously retaining scalability in computational processing.

5. A hierarchical framework is adopted for route planning estimation. The framework comprises macro-path and micro-path layer optimization. In the macro-path layer, we optimally compute the best route from a start to an end point that includes the most preferred objects of a scene according to user's needs constrained with additional geo-referenced criteria. In the micro-path layer, we estimate the best personalized view for an object of interest depicted in user's screen. The latter is achieved by extending the current entropy-based best view selection methodologies via a new personalized entropy metric, that measures not only the geometric properties of a projection but also personal preferences.

5.3. Functionality of the Proposed Intelligent System

Considering the above methodologies and approaches, it is possible to design an intelligent platform that can evaluate a number of parameters, design a path and automatically navigate into a three dimensional environment presenting all the interesting points and important information based on user's preferences.

The proposing GIS platform has to offer the following functionalities:

- Ability to extract knowledge from the scene. The features of 3D models have to be extracted such as location, shape, size, volume, texture, etc.).
- Support of multiple databases
- Automatic navigation into the scene based on user's preferences and criteria

In order to design such a system, focused on a 3D large scale space, it is necessary to be aware of the virtual environment and the objects it contains. The altitude of the ground can be known. The system must be able to calculate the volume and the size of the obstacles (3D Models) in the environment. Their shape and their type could be important. In addition there are cases that properties of models or specifications have to be considered in order to plan a flight. Moreover, unique specifications will help the system to decide and better present an obstacle.

All these data have to be built or retrieved from an existing GIS system. Additionally, information and data that might be used will be stored in a local collaborative database. When the system has all of the above data and knowledge, it will be able to choose the points or objects to present and potentially the ones to avoid. These choices will take place based on the user's criteria, preferences and subsequently, the path will be planed according to them.

In the final processing stage of the system, the solution will be presented using a three dimensional viewer or browser.

5.4. System Overview and Basic Requirements

Figure 1 presents an overview of the proposed personalized route guidance system. As is observed, the proposed architecture comprises five different modules; 1) the 3D object modelling and metadata component, 2) the relevance feedback component, 3) the ontology driven weight rectification component, 4) the optimal macro-path layer: route planning component and 5) the micro-path layer: best view selection component. In Figure 11, the processing steps have been also depicted that take place within each individualized component (as rounded rectangle boxes the number of each indicating the order of the respective processing step). In Figure 11, one may see all the main components of the proposed personalized 3D navigation system along with the order of execution of the respective processing units. All these components are seamlessly inter-woven with each other in order to provide an efficient truly personalized navigation of 3D worlds (virtual, natural or hybrid).

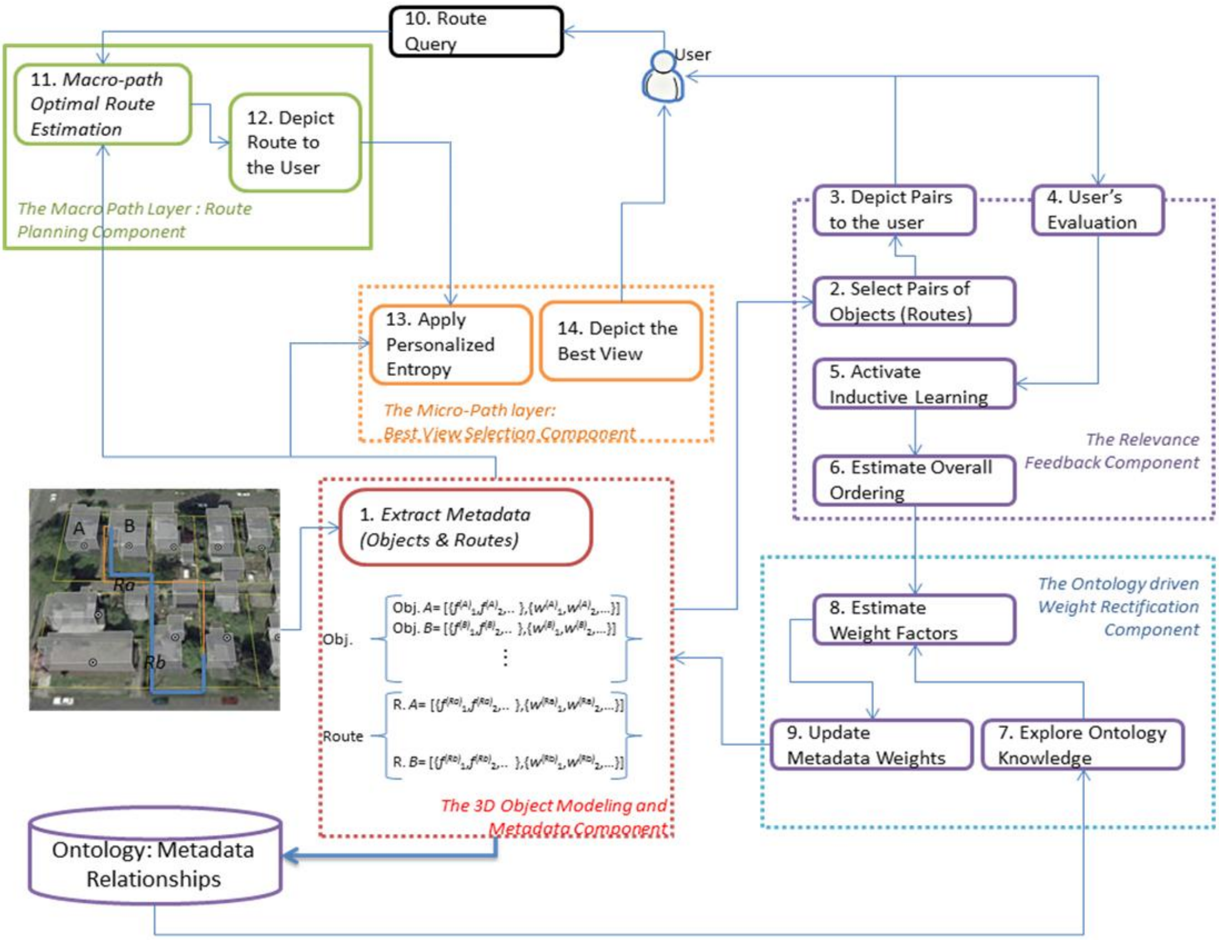

10. Route Query
User
11. Macro-path Optimal Route Estimation
12. Depict Route to the User
The Macro Path Layer : Route Planning Component
3. Depict Pairs to the user
4. User's Evaluation
2. Select Pairs of Objects (Routes)
5. Activate Inductive Learning
6. Estimate Overall Ordering
The Relevance Feedback Component
13. Apply Personalized Entropy
14. Depict the Best View
The Micro-Path layer: Best View Selection Component
A
B
Ra
Rb
1. Extract Metadata (Objects & Routes)
Obj.
Obj. $A=[\{f^{(A)}_1,f^{(A)}_2,\ldots\},\{w^{(A)}_1,w^{(A)}_2,\ldots\}]$
Obj. $B=[\{f^{(B)}_1,f^{(B)}_2,\ldots\},\{w^{(B)}_1,w^{(B)}_2,\ldots\}]$
Route
R. $A=[\{f^{(Ra)}_1,f^{(Ra)}_2,\ldots\},\{w^{(Ra)}_1,w^{(Ra)}_2,\ldots\}]$
R. $B=[\{f^{(Rb)}_1,f^{(Rb)}_2,\ldots\},\{w^{(Rb)}_1,w^{(Rb)}_2,\ldots\}]$
The 3D Object Modeling and Metadata Component
The Ontology driven Weight Rectification Component
8. Estimate Weight Factors
9. Update Metadata Weights
7. Explore Ontology Knowledge
Ontology: Metadata Relationships

Figure 11: An overview of the proposed personalized route planning architecture.

Figure 12 presents the flowchart of the proposed adaptive personalized route guidance system. Two different procedures are described. The first (on the left side) refers to the route and the best view selection process. The user selects a start and an end point and the system responds with the best preferred route. Then, within a selected itinerary and for the selected objects of interest, the system returns the best views composing a personalized 3D trajectory around objects of interest. The second procedure described in the right side of Figure 12 shows the main steps of the estimation of the metadata weights used to approximate user's preferences. In the following, we describe the flows of the basic components of the proposed platform.

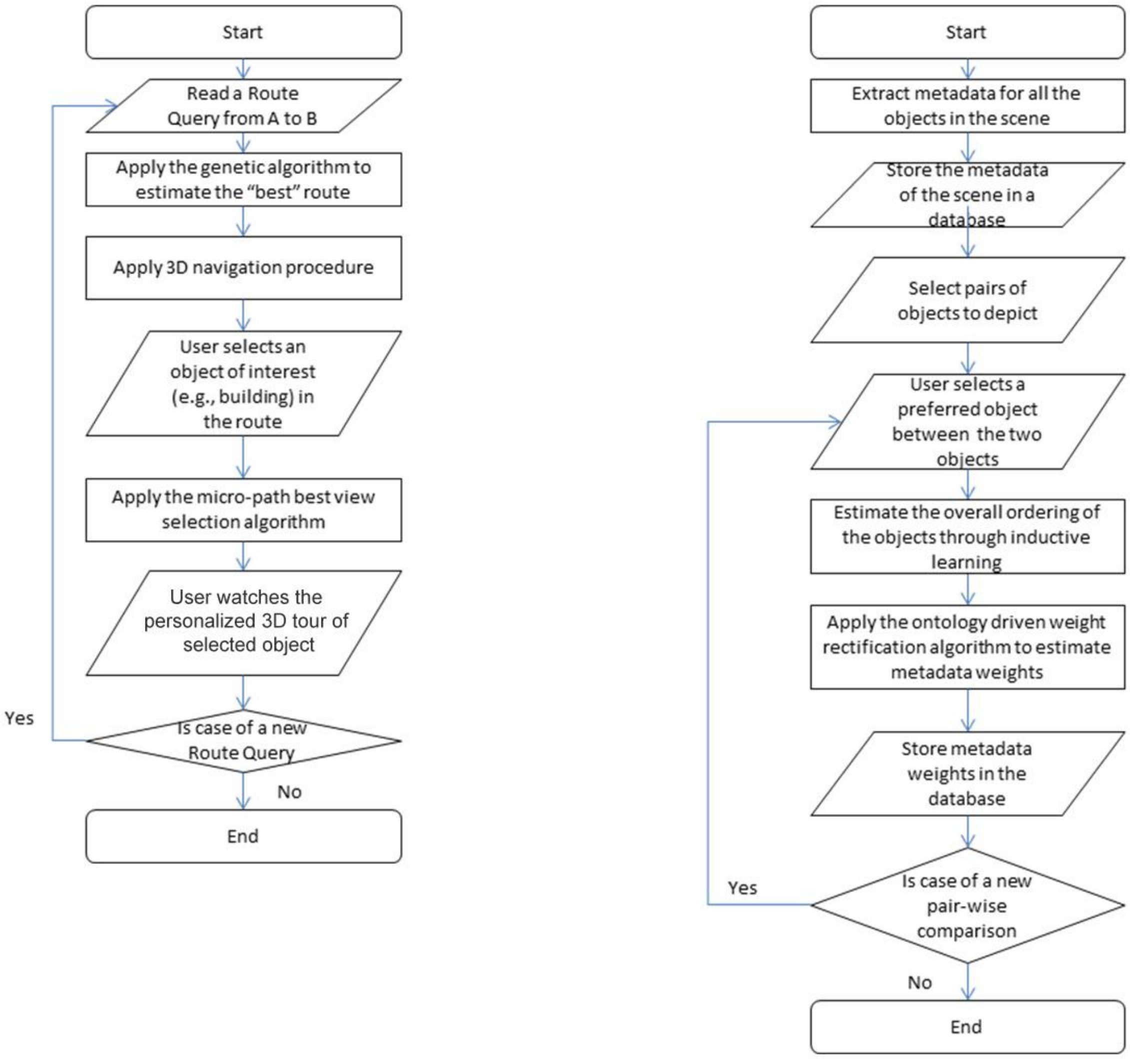

Start
Read a Route Query from A to B
Apply the genetic algorithm to estimate the "best" route
Apply 3D navigation procedure
User selects an object of interest (e.g., building) in the route
Apply the micro-path best view selection algorithm
User watches the personalized 3D tour of selected object
Yes
Is case of a new Route Query
No
End
Start
Extract metadata for all the objects in the scene
Store the metadata of the scene in a database
Select pairs of objects to depict
User selects a preferred object between the two objects
Estimate the overall ordering of the objects through inductive learning
Apply the ontology driven weight rectification algorithm to estimate metadata weights
Store metadata weights in the database
Yes
Is case of a new pair-wise comparison
No
End

Figure 12: The flowchart of the proposed adaptive personalized route guidance system; (a) the flowchart concerning the route and the best view selection procedure (macro-path/micro-path layer optimization), (b) the flowchart concerning estimation of the weight factors used to approximate user's preferences.

Modelling and metadata flows: Modelling aims at extracting metadata to represent either the 3D objects of the scene or the respective routes (e.g., navigation paths). It is clear that if we select more representative metadata (textual or visual), we can derive a better performance regarding the overall personalized route guidance system. For each extracted feature, a weight factor is assigned to regulate the degree of importance of the respective feature with respect to user's information needs. Initially, all the weights have similar importance but as the system proceeds their values are automatically adjusted through the use of relevance feedback and the ontology driven weight rectification strategy. Metadata and the respective weights are stored in a database and then they are used for route planning and best view selection.

Relevance feedback learning flows: In our system, we have proposed an active, inductive learning policy for computing the final object ordering. In particular, the system selects pairs of objects that are depicted to the user. Then, the user evaluates these objects and feeds information back to the system indicating which of the two objects better fits his/her information needs. In the following, the inductive learning algorithm takes place to estimate the overall objects' ordering. Therefore, we propose an implicit methodology as far as user's preferences estimation is concerned. User selects a set of relevant /irrelevant objects or performs comparisons between two objects and then leaves the system to automatically determine the respective user profile by assigning a degree of importance (weight value) to each of the extracted object metadata. As a result, the selection process accomplished is hidden from the users.

Ontology driven weight rectification flows: The main difficulty of the aforementioned approach is that a sufficiently large number of selected pairs is required to get a reliable object ordering. This is more evident for large 3D virtual worlds, consisting of hundreds of objects of quite different characteristics. Large number of samples means that we need several feedback iterations to converge to reliable metadata weights. To address this difficulty, in this work, we propose a weight rectification strategy that exploits metadata relationships as described through an ontology. In other words, the general knowledge of a virtual world, as expressed through the ontology, reduces the number of samples required for a reliable estimation of the metadata weights.

Optimal route planning (macro-path layer) flows: An efficient creation of a personalized 3D geographical path includes, apart from the 3D objects (such as buildings) that satisfy as much as possible user's information needs, additional criteria. For example, the distance of the path, aesthetic factors of a route (e.g., route with parks, or with bars, or shops) and the services offered over the path could serve as criteria. All these multi-criteria factors are included in a common formula for optimization in order to extract “the best”-with respect to user’s preferences- route. The main bottleneck of such multi-criteria optimization is that it is computationally intensive, since all possible combinations should be exhaustively examined. The conventional algorithms, that exploit a shortest path optimization approach by creating a graph the vertex of which are the 3D objects (e.g., buildings) and the edges the multi-criteria distance between two examined objects, usually are trapped to local minima surfaces deteriorating optimization performance [43]. In addition, such approaches lack scalability in computational processing. Different types of client devices, such as mobile phones, PDAs or laptops present quite different computational capabilities and, therefore, require scalable algorithms in the estimation of the personalized 3D paths.

To address these difficulties, we introduce in this work a genetic solution for the selection of the most preferred route. Genetic Algorithm (GA) incorporates quite well with the adopted multi-criteria approach where several factors are involved in the optimization process. GA is an effective methodology for solving computationally intensive problems yielding solutions close to the global optimum within an acceptable time of operations. Additionally, another advantage of the genetic scheme is its scalability; the performance of the optimization is regulated by the number of the algorithm operations in the sense that the more operations we apply the better solution we yield. On the contrary, conventional shortest path optimization algorithms are not able to improve the solution on demanding more computational power.

Best view selection (micro-path layer) flows: The final stage of the proposed system includes the algorithm used to estimate the best view of selected 3D objects within an itinerary. In particular, initially the user selects a set of objects within a route and then the best view selection algorithm calculates the most preferred 3D trajectories around them that satisfy as much as possible user’s preferences as well as geometric constraints.

6. 3D Object Modelling and Semantic Metadata

In this section, a new collaborative approach of 3D city modelling will be presented using existing 3D reconstruction tools. Moreover semantic metadata of the 3D models will be presented. The respective semantic metadata will be used for the personalized 3D navigation in the scene. We also discuss the ontological model used to interrelate the extracted semantic metadata (textural or visual) with each other.

6.1. 3D Building Modelling

6.1.1.Introduction

Most of the well-known virtual globe applications and platforms offer a three dimensional (3D) virtual environment enriched with three dimensional objects. Physical objects have been represented with 3D models. Besides the 3D models, all the spatial data are organized in layers, based on the content type. These data can be provided by the same distributor as the platform, or they can be provided as a service from a third party distributor. Spatial data usually have been developed for the needs of each individual virtual globe application, according to their own needs and requirements. A simple functional block diagram of a virtual globe platform is displayed in Figure 13 (Figure 4 repeated for convenience).

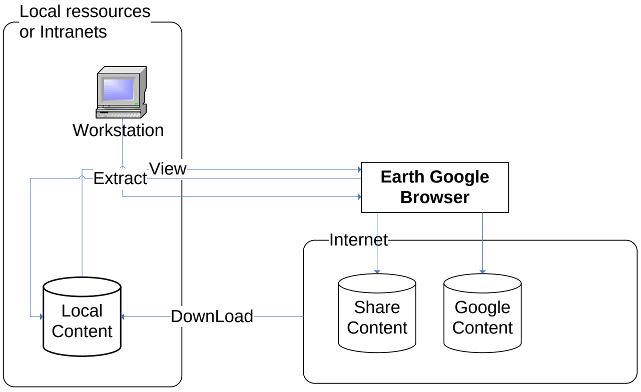


Figure 13: Simple virtual globe functional block diagram [70]

Each individual database, either an integrated one or a third party solution, is of restricted access. The integrated layers are richer in content and the contained material usually is more complex and harder to be prepared. On the other hand, third party services are offering spatial data that can be easily gathered and is usually text, photos, or multimedia content.

Two of the most popular commercial virtual globe distributions have restricted the direct access to the layer of 3D building models. They are offering limited access to some 3D models, those that users of the platform have created and shared into the community of each platform.

Our work is dealing with personalized navigation and exploration into 3D large scale geo-referenced scenes. In order to achieve such a functionality, 3D models of buildings have to be known and well defined. Unfortunately, as it has been mentioned before, it was impossible to retrieve data such as 3D models from the current virtual globe distributions. These data are stored on the integrated layer of each of them. All the known virtual globe applications that have 3D environment do not allow access to the layer of 3D models of buildings. None of them offer the users the ability to interact or download 3D models without any restrictions. The aforementioned spatial data of 3D models of buildings are contained into the integrated database of the virtual globe implementations and cannot be isolated.

Therefore an external database with 3D models of buildings was designed and created for this purpose. All the 3D models and city blocks were stored on this database. Three dimensional models of real buildings were reconstructed using existing 3D reconstruction tools as it is presented in the next paragraphs. Afterwards, each 3D model was defined by a set of features, in order to meet the requirements of the proposed platform.

6.1.2.Reconstruction Process of 3D models

Urban planning and city modelling have gradually shifted, during the last decade, from highly demanding, in terms of required skills and supporting information, to tasks now supported by efficient, widely available applications, thus becoming popular and largely accessible. There are many GIS systems nowadays that offer a freely navigable three dimensional environment. This evolution of GIS systems has, in turn, led to the requirement for and creation of virtual 3D models of the ground and buildings. Online communities have created and distributed over the Internet libraries of geo-referenced 3D models. The public is encouraged to participate in the design of 3D scenes and many companies offer free tools to facilitate the design of 3D models, specialized in buildings. A collaborative approach is used for the construction of a city model and its implementation through a prototype environment, employing freely available design tools. A prototype system which is entitled GIS Scene Explorer (GIS-SE) [69] comprises a collaborative database, supported by a web-based interface. Users are able to create and upload their models to the common database over the web, thus constructing a realistic 3D city model in a given area in a collaborative manner [70].

One of the main components of the 3D model reconstruction process incorporates the Google Building Maker reconstruction tool of Google. During the end of 2009, Google introduced a new tool for reconstructing a 3D model of a building. It was the Google Building Maker, a simple web based tool that allowed the user to reconstruct the model of a building from multiple satellite and aerial images within minutes. In our approach, we adapted Google Building Maker as the initial tool to design and build the 3D model of a building [70].

This part of the current work presents the implementation of a collaborative web-based environment for geo-referenced 3D city modelling. Our approach is using the Google Earth platform. The environment offers the following functionalities:

- Construction of custom 3D building models.
- Refinement of the custom 3D building models to improve realism.
- Placement of custom 3D building models in a geo-referenced scene.
- Publication and sharing of the custom models and scenes with the other users of the environment.

In order to achieve the aforementioned functionalities a series of tasks had to be accomplished. In particular, initially, a custom GIS collaborative database had to be built to store and maintain all custom constructed models. Next, the database was populated by building models comprising 3D scenes. These constructed models were contributed by a team of student volunteers.

A working region was set and each student was assigned the construction of the model of a building within the region. Google Building Maker was used for the initial design of the models and Sketchup was used to correct possible errors with respect to the corresponding building.

Moreover, the texture of 3D models could be improved with higher quality photographs or aerial images from the area.

By completion of each building model, it was uploaded to the collaborative database and all the community was able to access it, visualize it and navigate to it. For the navigation process the Google Earth plug-in component was used.

The communication between the local server and the clients but also between clients and Google's server was achieved by AJAX technology. The Google Earth API was used in order to manage the collaborative database and filter the data. In Figure 14, the steps of the whole process are being presented [70].

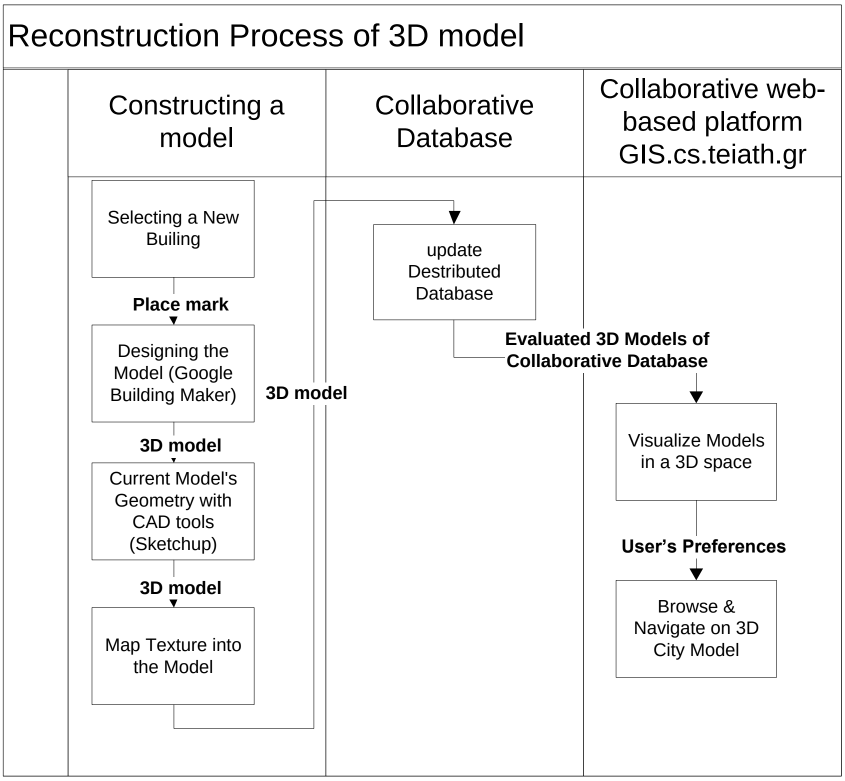

Reconstruction Process of 3D model
Constructing a model
Collaborative Database
Collaborative web-based platform GIS.cs.teiath.gr
Selecting a New Builing
Place mark
Designing the Model (Google Building Maker)
3D model
Current Model's Geometry with CAD tools (Sketchup)
3D model
Map Texture into the Model
3D model
update Destributed Database
Evaluated 3D Models of Collaborative Database
Visualize Models in a 3D space
User's Preferences
Browse & Navigate on 3D City Model

Figure 14: Collaborative 3D city development process

Users were called to design 3D models of buildings in a desired specified area. Our web platform interface was visualizing the region in a 3D environment. All the available models were visible to all users. Users could freely navigate into the scene and select a desired free, i.e. not yet reconstructed, building to reconstruct. The building was reserved by placing a placemark over the corresponding area. The placemark was announcing to the community that the building had been chosen for reconstruction. The model of building was reconstructed using Google Earth Building Maker (see Figure 17). If it were necessary, modifications and improvements would take place in order to improve realism. Afterwards, the 3D model was uploaded on the collaborative web-platform and became visible to all the members of the community.

Our approach has been based on the server – client architecture, using a virtual globe system. On the side of the local server a collaborative database was built. The users could access the collaborative database through a web based interface. A more detailed functional diagram is displayed below. Each module is presented in Figure 15 [69],[70].

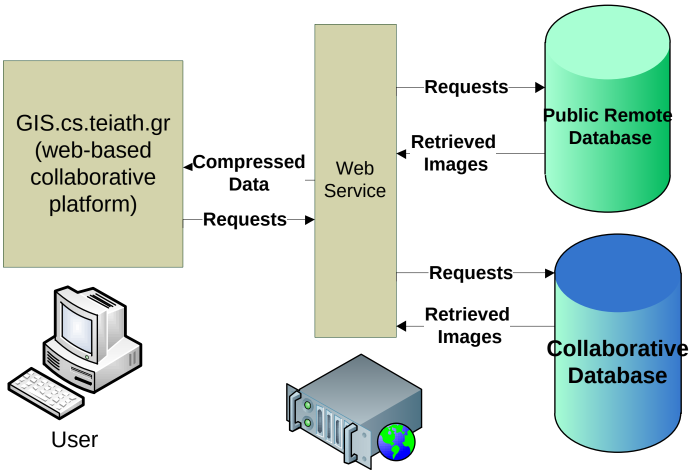


Figure 15: Block Diagram

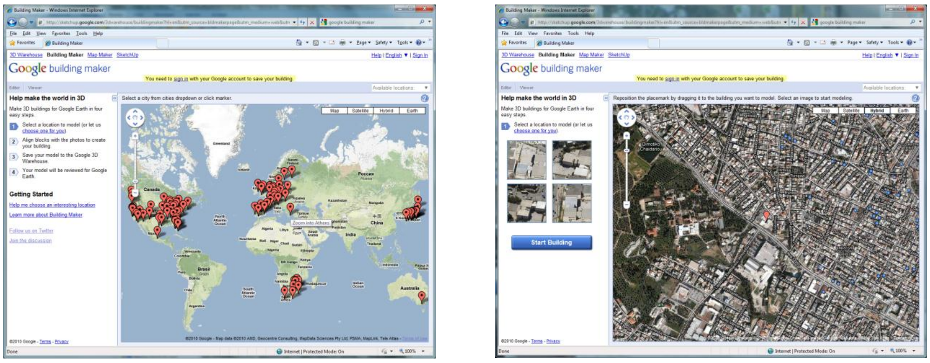
A collaborative local database was used. It was accessible by all the users of the platform. A programming component released by Google was used to browse and visualize the content of the GIS databases. The users were able to see their team’s models and all the content of the public database through a web site. Any item that was added on the database was instantly visible to anyone.

For the reconstruction process of the city we have restricted the working area in a small region, by marking it on the database. The models and any other geo-referenced context had to be built on the user’s side.

6.1.3.Single 3D Model Reconstruction

Comparing all the existing tools for construction of a 3D building model, Google Building Maker was selected due to its user interface and its reconstruction speed. With this tool, a model can be designed within a few minutes. In addition, texture can be added on the sides of the building model from the real satellite or aerial photographs. Other tools, such as Sketchup or CAD tools, present certain advantages but they lack the ability to automatically match the texture on each side of the building model.

Google Building Maker was not available for all regions in the world. In particular, it was available in cities with incomplete models or in ones having only a few 3D models. For our purpose we chose Athens [see Figure 16(a)] and initially a particular region nearby the Technological Educational Institute of Athens. Afterwards, the reconstruction area was extended to the entire section of the greater metropolitan area of Athens where Google Building Maker is allowed to be used. The time that this research is running the area of Athens available for reconstruction is limited and it is the only city with even partial availability in Greece.


Building Maker - Windows Internet Explorer
File Edit View Favorites Tools Help
Favorites
Building Maker
Page Safety Tools
3D Warehouse Building Maker Map Maker SketchUp
Help | English | Sign In
Google building maker
Editor Viewer
Available locations:
Map Satellite Hybrid Earth
Zoom into Athens
Start Building
Done
Internet | Protected Mode: On
100%

You need to sign in with your Google account to save your building.

Help make the world in 3D

Make 3D buildings for Google Earth in four easy steps.

1 Select a location to model (or let us choose one for you)

2 Align blocks with the photos to create your building

3 Save your model to the Google 3D Warehouse.

4 Your model will be reviewed for Google Earth.

Getting Started

Help me choose an interesting location

Learn more about Building Maker

Follow us on Twitter

Join the discussion

Select a city from cities dropdown or click marker.

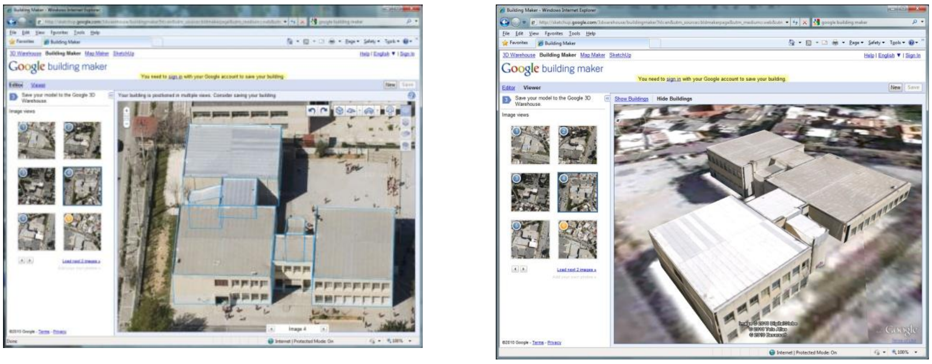
Reposition the placemark by dragging it to the building you want to model. Select an image to start modeling.

©2010 Google - Terms - Privacy

(a) (b)

Figure 16: (a) Choosing a region (b) Selecting a building for reconstruction

Users could choose a building to construct by placing a red place mark on it. With a blue circle existing models of public Google’s Earth database buildings were marked [see Figure 16(b)].


Building Maker - Windows Internet Explorer
File Edit View Favorites Tools Help
Favorites
Building Maker
Page Safety Tools
3D Warehouse Building Maker Map Maker SketchUp
Help | English | Sign In
Google building maker
Editor Viewer
New Save
Image views
Load next 2 images »
©2010 Google - Terms - Privacy
Image 4
Show Buildings Hide Buildings
Internet | Protected Mode: On
100%

You need to sign in with your Google account to save your building.

Save your model to the Google 3D Warehouse.

Your building is positioned in multiple views. Consider saving your building

(a)

(b)

Figure 17: (a) Constructing a model with "Google Building Maker" (b) 3D reconstructed model of building ("Building Maker" process)

The process of designing a model was based on bounding geometrical objects. Bounding boxes and bounding pyramids were used to construct the model. The bounding boxes had to cover and shape the building in the best possible way. The GIS platform was providing the necessary photographs for a wide variety of points of view of the desired building. An arbitrary number of photos could be used for the design. An increased number of photos led to increased accuracy. The building was designed from the image of the first point of view and corrections were made by the next views. The 3D reconstruction process was done by the GIS system and a three dimensional geometrical model was created. Texture images were applied automatically on the sides of the model, from the satellite or aerial photographs. Each model that was created by the above process was placed on the public web library of Google Earth and was accessible by anyone.

6.2. Semantic and Geometric Features

The goal of this research project is to implement a new methodology for personalized navigation and exploration into 3D large scale geo-referenced scenes. In order to achieve this goal, the whole 3D scene has to be defined and specified clearly. The specifications have to provide efficient data to the decision support system (macro and micro path layer) to plan the optimum path based on the user's preferences. It is acceptable that one of the first features of the 3D models will be their referenced location and their geometrical specifications. These features are enough to service the simplest queries of finding the shortest path among the 3D models on a geo-referenced environment.

Besides the spatial properties of each individual 3D model, additional features have to be defined. These features will help us to answer questions such as, what type of buildings they are, or, what they are used for, or what the purposes are for their existence or for visiting them.

A set of descriptors have been defined in order to answer questions such as the above. The descriptors were selected in a way to be able to provide an efficient description of the main content of the buildings and what a visitor is expected to interact with or see. Moreover, descriptors that describe the outside features of the buildings were added [71]. An overview of the features of the buildings is presented in the following Figure 18.

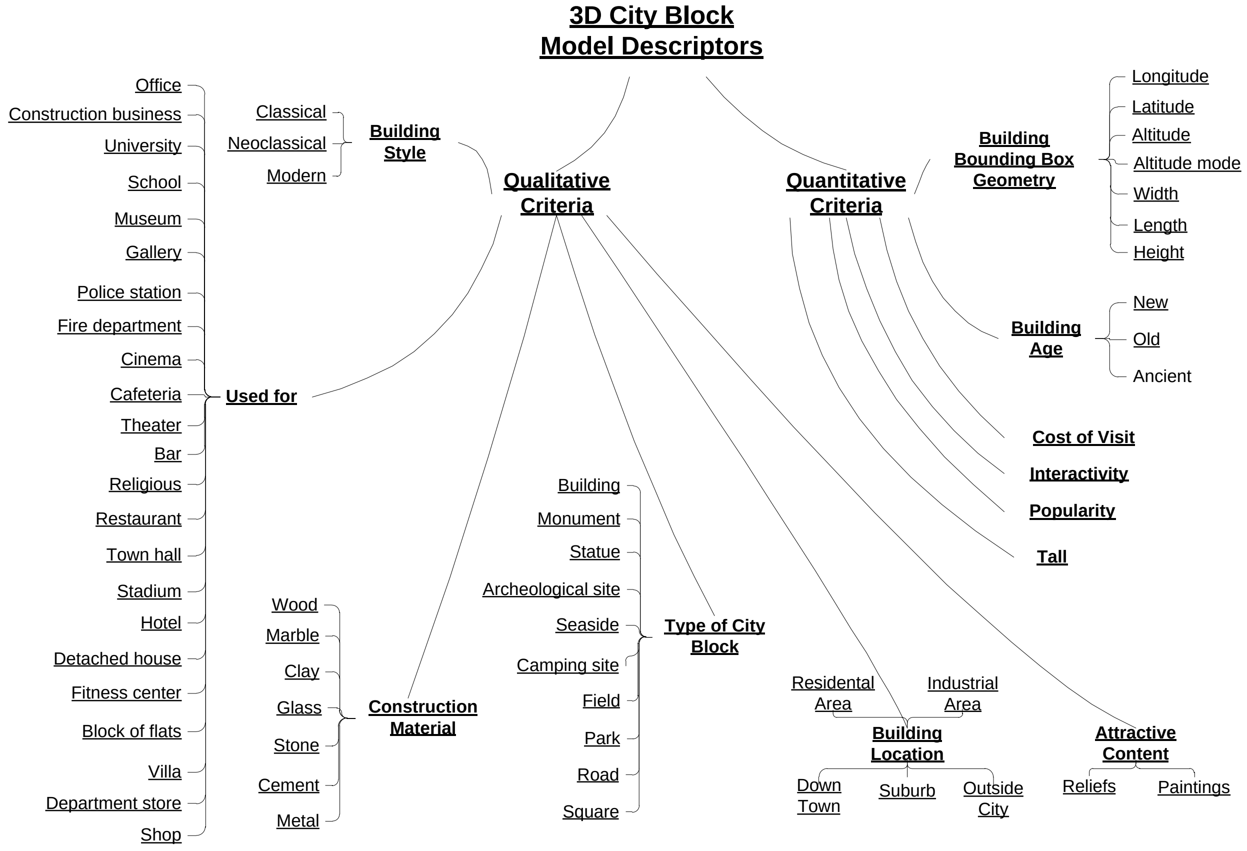


Figure 18: The set of the metadata used to describe the geo-referenced 3D content [71]

6.3. Semantic Metadata

Scene metadata are classified into those ones associated with 3D objects – involved in the construction of the 3D world – and the ones related with the route. As far as objects' metadata are concerned, two different types are taken into account; the qualitative and quantitative ones. Qualitative metadata include the type of building use, building style, construction material, building location, attractive content and the type of city block. Qualitative metadata are expressed Boolean form. On the other hand some of the quantitative metadata, such as cost of visiting, interactivity and popularity are expressed in numeric form through a set of discrete values ranging in our case from zero to three {0,1,2,3} (0-none, 1-low/few, 2-medium/several, 3-high/many). Each of the above discrete values are normalized in the space of [0,1] due to learning algorithm requirements. Besides the above features building age and building geometry is included in the quantitative metadata. The first one is represented by three different states of Boolean values and the buildings geometry is defined by a set of numeric variables [71]. A detailed description of the metadata used to describe the 3D object in a scene is presented in Figure 18.

Route metadata express the description of a path starting from an object *u* and ending to an object *v*. Examples of these metadata are the level of safety of a route, the number of services offered by this route (e.g., gas stations), or emotional properties of the route (i.e. a path for sightseeing, or one of shops or cafes). However, it should be mentioned that the proposed system is open to include additional metadata for describing objects properties and route characteristics, like, for instance, speed limits, traffic conditions, number of junctions, and number of turns. It should be mentioned that the main focus of this work is on the automatic adjustment of the semantic metadata weights which model user's preferences to yield a personalized 3D navigation experience rather than on metadata themselves. After all, metadata are application dependent, meaning that different descriptions are required for different application domains. So, future improvements can be obtained by providing more precise semantic metadata concerning either the description of the 3D objects or of the route; however, this is highly depended on the application scenario used.

An overall N-dimensional feature vector $\mathbf{f}_u$ is used to describe the extracted semantic metadata concerning the u-th 3D object in the scene. The qualitative metadata are included in the feature vector $\mathbf{f}_u$ as Boolean values indicating the existence or not of the specific property. This is essential in order to operate the learning component on the extracted metadata and relates them with the high level user's preferences. Besides feature vector $\mathbf{f}_u$ which correspond to a particular 3D object, there exists features that describe a route that connects object A with

B. We denote as $\mathbf{f}_{u,v}$ the feature vector that includes the metadata of the route connecting the object u with object v.

6.4. Semantic Metadata Interrelations through Ontologies

All the aforementioned extracted semantic metadata (qualitative and quantitative) are semantically interrelated with each other through the use of an ontological model, schematically depicted in Figure 19. As we have stated above, the ontology has a dual purpose. On the one hand, it provides a framework for a more efficient users' preferences estimation since it allows via a few number of clicks (users' interaction) to estimate the degree of importance of several (many) metadata (features and/or keywords). This is achieved by propagating the weight values of some metadata (assigned automatically and transparently to the users through the learning strategy) to other interrelated ones through the conceptual ontology [71]

The second purpose of the ontology is to model more complex concepts paving the way for a human-perceptive navigation. For example, one can model, apart from the use of a building (through the "building used for" metadata, see Figure 18 and Figure 19), the scope of visiting such a building (e.g., educational scope) and what is expected for the visitor to gain from such a visit. Considering a museum as a building of interest, the scope of a visit regards history, culture or science.

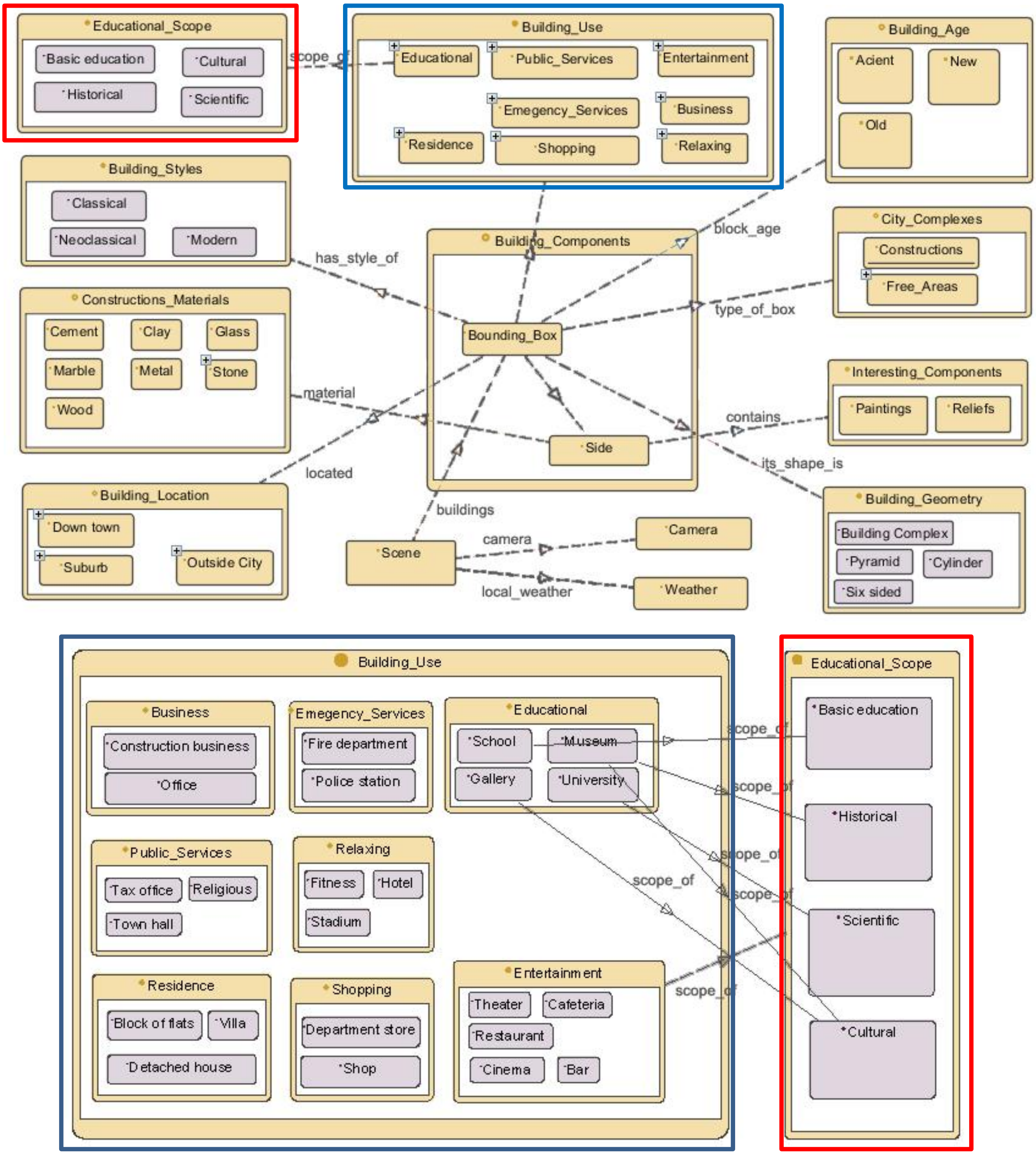


Figure 19: The ontological model used in the proposed architecture

6.5. Definition of Semantic 3D Model Specifications

Afterward the reconstruction process, each 3D model of a building was characterized by a set of descriptors. Due to the complexity of the 3D model of the whole city, it was assumed that each 3D model, no matter the individual shape and size, could be considered as a rectangular cube. In this way each 3D model is contained in a bounding box (see Figure 20).

Considering more complex geometrical figures was out of the purpose of this research. Almost all the 3D models that were reconstructed had simple geometry of a single cuboid figure. The bounding box was estimated in a semi-automatic way. The coordinates of the four points with zero altitude was the face of the bounding box that was touching the ground. Then the bounding box was extended as much as it was required to cover the higher point of the 3D model. Afterwards the estimated bounding box was stored in the database. In case of more complex 3D models of building, corrections of the bounding box were taking place by human. As a result, the bounding box always touches the ground and the size and orientation of it is estimated in a way to minimize the area of it and hug the 3D model of the building, in the most efficient way.

In addition, it was assumed that each 3D model was representing one building. Moreover, it was assumed that each building has one use and it is used only for one purpose. Therefore, the features broke down into two categories. There are features that can be applied to the whole building and some others that can be applied in each individual bounding box side. Each 3D model has five bounding sides, four of them are around the building and the last of them is on the roof of it.

The features that are applied to each side of the bounding box are mentioned in the Figure 19.

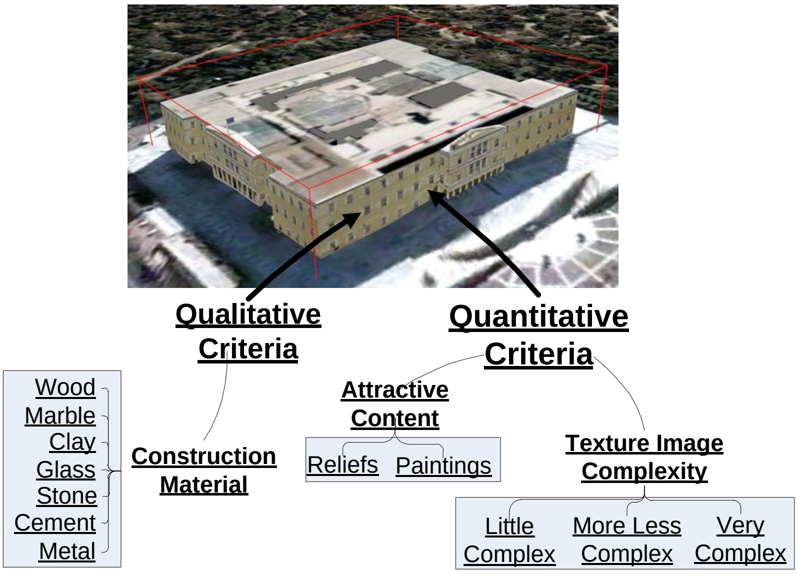


Figure 20: Bounding side features

The features that are applied to the whole bounding box are hierarchically inherited to the sides. In the Figure 21 the features of the whole bounding box are presented.

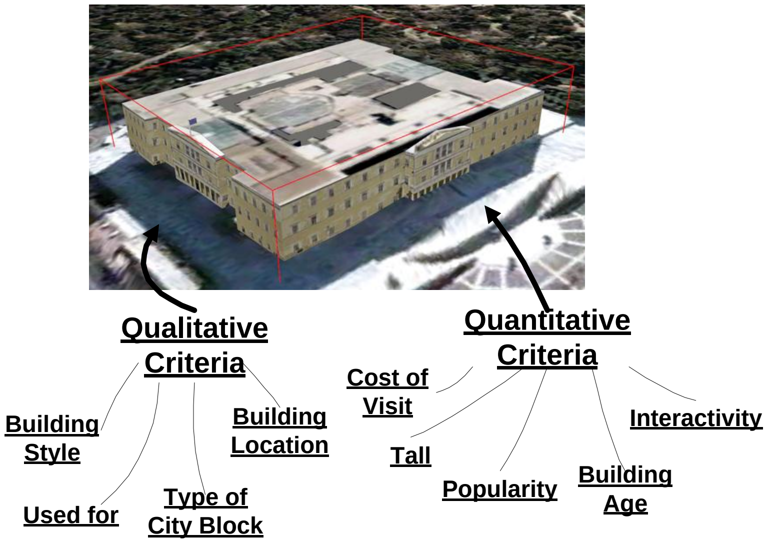


Figure 21: Bounding box features

There was an assumption that all the 3D models were *touching* the ground with one of their faces. A bounding box with 6 faces was used to represent the geometry of the 3D model. The bounding box was including the 3D model of the object. Therefore each 3D model once it has been 3D reconstructed was analysed by the prototype GIS-SE standalone application and the optimal bounding box was calculated. Hierarchically each bounding box was analysed in 5 faces; one for the roof and 4 faces for the sides (the ground face was not considered). Each face was defined by three points (9 coordinates) on the 3D space and their coordinates were stored on the database of the platform. These coordinates were used by the platform in order to identify the faces and define features for each (see Figure 22).

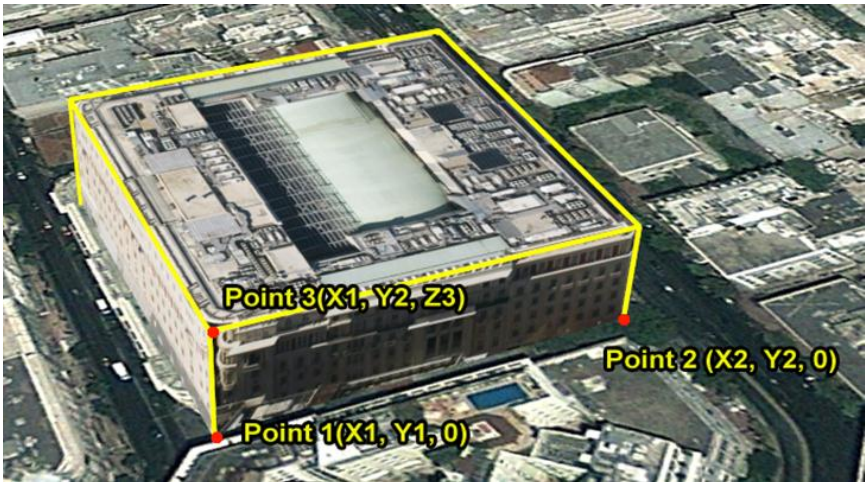


Figure 22: Points of a bounding box

On each face, features were attached characterised by a set of descriptors as they are presented in the Figure 21. The database schema that hosts all the above data is presented in Figure 23.

The assumption of the bounding box was made in order to reduce the complexity of the whole 3D scene.

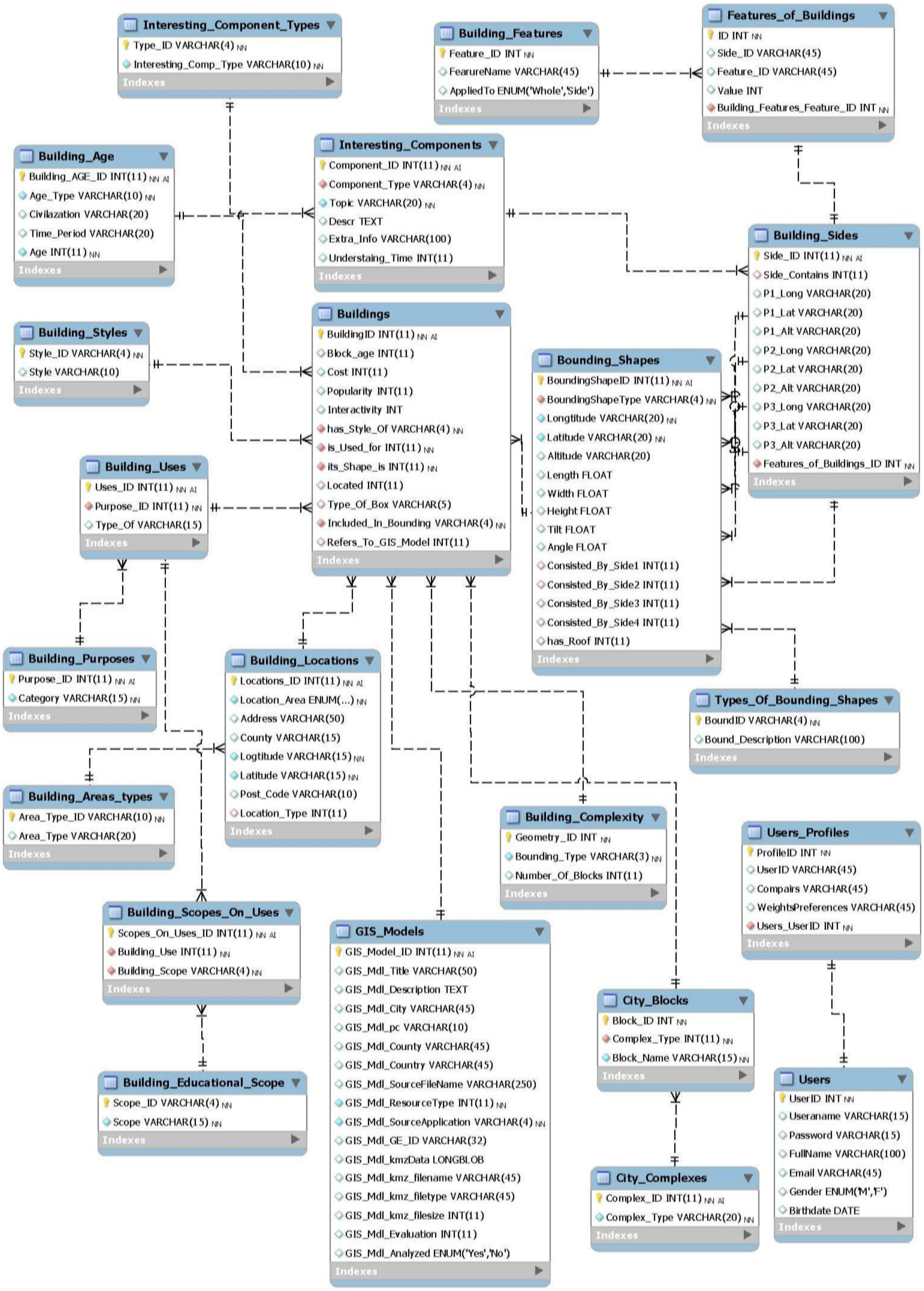


Figure 23: GIS-SE database schema

6.6. City Modelling

Exploiting the possibilities of the aforementioned 3D building reconstruction tool, a collaborative environment was created, based on the relevant scene database, in order to gather many models of buildings and create a model of a localized scene or a whole city [70]. The collaborative GIS system that was designed and used is shown and analysed in processes in the diagram of Figure 24.

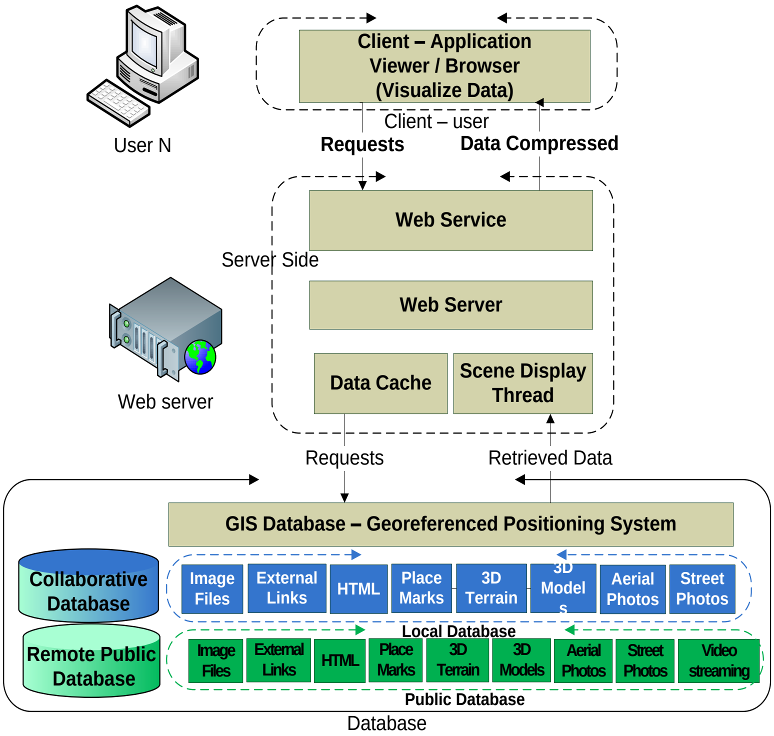


Figure 24: “gis.cs.teiath.gr” Detailed Block Diagram

The remote access of the database and the distribution of data were achieved through HTTP web service. In particular, a web site was built and the database was hosted in an SQL Server. A Google Earth plug-in component was used to visualize the data and create a 3D dimensional world with OpenGL-DirectX. Google has made public an API for Google Earth use. Using AJAX technology and the Google Earth API, the public database was filtered and the shared database was loaded and visualized on Google Earth window [70].

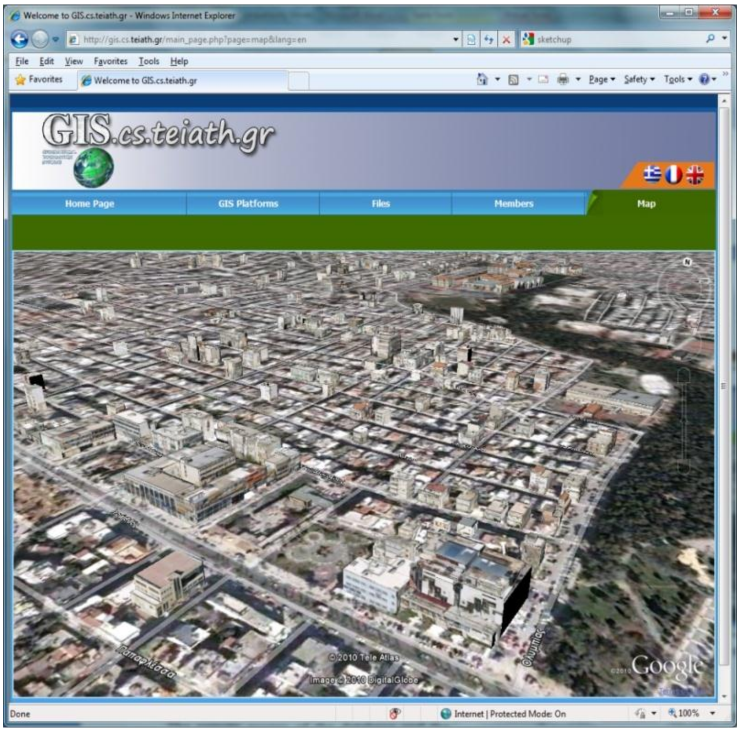

Welcome to GIS.cs.teiath.gr - Windows Internet Explorer
http://gis.cs.teiath.gr/main_page.php?page=map&lang=en
sketchup
File Edit View Favorites Tools Help
Favorites
Welcome to GIS.cs.teiath.gr
Page Safety Tools
GIS.cs.teiath.gr
Home Page
GIS Platforms
Files
Members
Map
©2010 Tele Atlas
Image © 2010 DigitalGlobe
Google
Done
Internet | Protected Mode: On
100%

Figure 25: "gis.cs.teiath.gr" collaborative environment – 3D models of building

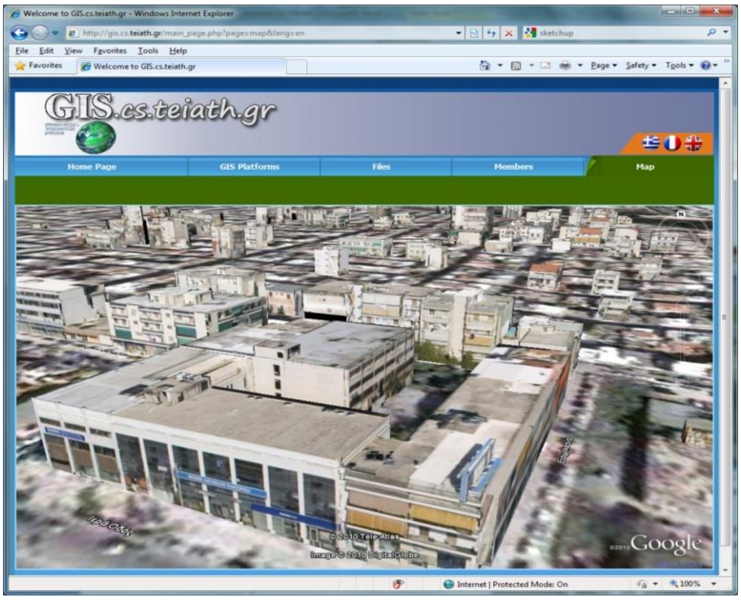

Welcome to GIS.cs.teiath.gr - Windows Internet Explorer
http://gis.cs.teiath.gr/main_page.php?page=map&lang=en
sketchup
File Edit View Favorites Tools Help
Favorites
Welcome to GIS.cs.teiath.gr
Page Safety Tools
GIS.cs.teiath.gr
Home Page
GIS Platforms
Files
Members
Map
©2010 Tele Atlas
Image © 2010 DigitalGlobe
Google
Internet | Protected Mode: On
100%

Figure 26: Experimental set of 3D models of buildings

Each user could access other members' buildings. All members of the team were sharing the same map, working on the same region. They could navigate in a three dimensional space, by zooming in or out and by flying around the buildings. A mark could be placed on a building that was under construction, by the user, announcing that the specific building was in design process. Each model was constructed using the Google Building

Maker. All members of the team collaborated in order to create the 3D model of a scene like the one in Figure 25 and Figure 26 [70].

In a small amount of time a geo-referenced 3D scene reconstructed and semantic metadata features were added, suitable for semantic navigation in the scene. These metadata are used by the learning algorithm and the path planning components of this work as described in the next chapters.

7. Active Inductive Learning Process

In this section, we discuss a new online learning strategy, used in our proposed architecture to efficiently estimate users' preferences. This strategy is also able to *model high-level concepts of users' profiles taking into account minimum users' interactions* as being reflected via the use of feedback mechanisms provided by the user about the preferred judgment between two objects; i.e., which of these two objects better fit users' information needs and preferences, or in other words, which of the two should be ranked as more preferred compared to the other. An active inductive learning algorithm adopt and used in semantic navigation in 3D large scale scenes. A pairwise method has been used for this purpose. The adopted *active inductive learning* strategy comprises the following basic steps; i) Data ordering, ii) estimation of feature weights and iii) weight rectification driven by the ontology model [71].

Step 1 – Data Ordering: We initially construct a small set of representative objects (e.g., buildings). Then, two of them are depicted to the user for a pairwise comparison. In this way, we are able to construct directed graphs, the nodes of which represent the objects of the training set, while the edges signify the pairwise preference distance between the two objects. It can be proven that this pairwise comparison is adequate to order the preferences for all objects in the training set (see Paragraph 7.2).

Step 2 – Estimation of the Metadata Weights: In this step, we exploit the object ordering for all samples in the training set (see Step 1) to automatically define the metadata weights for all objects of the scene, either within or outside the training set (see Paragraph 7.3).

Step 3 – Ontology Driven Weight Rectification: In the last step, we improve the initially estimated metadata weights (see Step 2) by considering metadata relationships defined through the ontology model (see Paragraph 7.4). This step allows reliable weight estimation even in the case that a few training samples are used, minimizing user's interaction. The following Table 3 summarizes the main steps of the active inductive learning algorithm.

Active Inductive Learning Algorithm:

1. Construct a small set of representative 3D objects
2. For every new selected Object do
    a. Update the training set
    b. Apply algorithm 1 as described in Table 4
    c. Apply the algorithm 2 as described in Table 5
    d. Activate the algorithm 3 (see Table 6)

Table 3: Main steps of the proposed active inductive learning algorithm.

7.1. Pairwise Comparisons

Let us denote as S a training set that contains representative 3D object (e.g., buildings) instances. Let us also denote as $g(\cdot)$ a user preference function defined as

$$g(u,v):SxS->[0\ \ 1] \quad (1)$$

where $u,v \in S$ are two selected objects. Function $g(u,v)$ defines the order between the object pair (*u*,*v*) with respect to user's preferences. More specifically, $g(u,v) \to 1$ (or respectively $g(u,v) \to 0$) is interpreted as a strong recommendation of being *u* more preferred (or less preferred) than *v*. On the other hand, a value around ½ indicates an abstention of making any recommendation.

As we have stated above, we put the user in the process loop by allowing him/her to judge which of the two depicted 3D objects is more preferred and feeding this information back to the system. Function $g(\cdot)$ is sufficient to create an overall ordering for all 3D objects (e.g., buildings) within the training set *S*. However, if we had a social group, we would need to perform a weighted average policy for all values of $g(\cdot)$ to estimate the group's mean preference function [23], [38].

To do this, we need to construct a graph, the vertices of which are the objects within the training set *S*, while the edges represent the value of the ordering function $g(u,v)$ for the object pair (*u*,*v*). Figure 27 presents an example of such a graph assuming a training set of four objects.

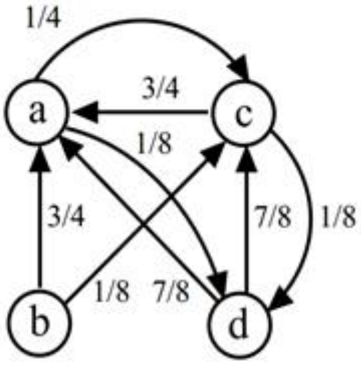


Figure 27: An example of a graph that represents the preference function for four objects (source [15]).

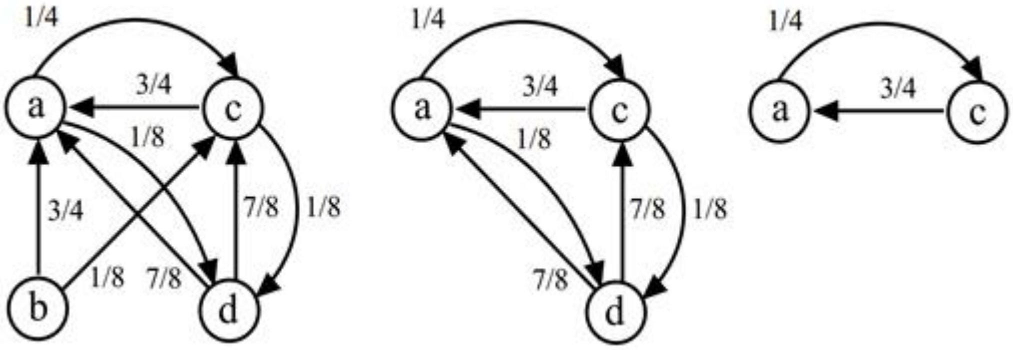


Figure 28: Graphical representation of the greedy approach used to estimate the overall ordering of all objects in training set S. The leftmost figure represents the graph of Figure 27. From this graph, object b will assign the maxima value and will be deleted, resulting to the middle graph. Then, node d will assign the maximum value of $\beta(u)$ and thus it will be deleted, leading to the rightmost graph. Finally, node c will be ranked ahead the node a. Therefore, the overall ordering will be b>d>c>a (source [13]).

7.2. Estimation of the Overall Ordering

As mentioned in [15], the problem to derive the overall order for all samples within the training set *S*, from the graph of the preference functions between two objects is NP-hard. To overcome this problem, we present an approximative algorithm. In particular, for each vertex say *u*, we assign a value that indicates the difference between the outgoing and ingoing edges. We denote this value as $\beta(u)$ in the following (see the example of Figure 28 for more details). Large value for the outgoing metric means that object *u* is highly preferred by the user while the opposite is valid for the ingoing metric (large values refer to objects of low significance to the user). In other words, for each vertex of the graph, that is, for each object *u*, we have to estimate the following metric

$$\beta(u) = \sum_{v} g(u,v) - \sum_{v} g(v,u) \qquad (2)$$

It is clear that metric $\beta(u)$ indicates a potential value of user's preference regarding object *u*. Then, by exploiting the values of $\beta(u)$, we can estimate a greedy algorithm that approximates the overall ordering of all objects in training set *S*.

In particular, the greedy algorithm selects from the graph as the most suitable object *u* the one that maximizes the value of $\beta(u)$ for all objects within the training set *S*. In other words, the selected node is set to be ahead from all the remaining nodes of the graph. In next step, we remove this node (that is, the respective object) from the graph and then we update the potential values of all the remaining objects (vertices) within the graph. This process is iteratively repeated until no nodes are included in the graph. Figure 28 presents a graphical representation of the method used for the estimation of the overall ordering of the objects in the training set *S* for the example of Figure 27. Table 4 presents a summary of the proposed pairwise comparisons and overall data ordering.

Algorithm 1 - Data Ordering: *Returns for all Objects in the Training Set (* $u \in S$ *) the Preference Value* $\beta(u)$

```
1. For every object $u \in S$ in the training set S do
    a. Select an object pair, say $u,v \in S$
    b. Depict this pair to the user and get an estimate of the function $g(u,v): SxS -> [0\ \ 1]$
2. Construct a graph $G^{(0)}\{v,e\}$ the vertex of which are the objects in the training set and the edges the values of $g(u,v): SxS -> [0\ \ 1]$
3. Estimate for every vertex of the graph the metric $\beta^{(0)}(u)$ [see Eq. (2)]
4. Set $n \leftarrow 0$;
5. While (graph $G^{(n)}\{v,e\}$ not empty) do
    a. Selects the vertex $\hat{v}$ of $G^{(n)}\{v,e\}$ that corresponds to the maximum value of $\beta^{(n)}(u)$, $\hat{v} = \max_{u \in G} \beta^{(n)}(u)$
    b. Assign to the object $\hat{v}$ of the training set S, the value of $\beta^{(n)}(\hat{v})$
    c. Update Graph $G^{(n+1)}\{v,e\} = Graph\_Update\ (G^{(n)}\{v,e\}, \hat{v})$
    d. // Function $G^{(n+1)}\{v,e\} = Graph\_Update\,(G^{(n)}\{v,e\}, \hat{v})$ removes vertex $\hat{v}$ from the graph and update the graph.
    e. Update metric $\beta^{(n+1)}(u) = Update\_Metric\,(\,G^{(n+1)}\{v,e\})$
    f. // Function $Update\_Metric\,(\,G^{(n+1)}\{v,e\})$ updates values of $\beta^{(n)}(u)$ of the graph $G^{(n+1)}\{v,e\}$
    g. Increase n, $n \leftarrow n+1$
    h. go to step 5.
6. Set $\beta(u) \leftarrow \beta^{(n)}(u)$
```

Table 4: The main steps of the algorithm used for pairwise comparisons and data ordering.

The main advantage of the proposed algorithm is that it can provide an estimate of the overall ordering of the objects in the training set *S*, by exploiting only pairwise comparisons. This way, we are able to better implement the interaction interface of the proposed personalized route guidance system. That is, the user evaluates pairs of objects as judgments of preferences instead of providing an overall ordering for all objects which is in fact a very arduous process.

7.3. Estimation of the Semantic Metadata Weights

The main limitation of this approach is that the overall object ordering is approximated only from the objects within the training set *S*, that is, from objects that have been depicted to the user for evaluation. However, as stated above, *S* is a small representative set. Thus, it is quite probable that many other interesting objects exist that are not included in *S* and therefore they are not ranked. In the following, we represent an algorithm that overcomes this limitation by estimating the weights of the metadata for all objects in the scene by taking into consideration the ordering information in the training set *S*.

In particular, we exploit the metadata weights to define a quantitative measure of how preferred are the objects of the scene with respect to user's preferences.

$$S(u) = \sum w_u^j f_u^j \qquad (3)$$

In equation (3), f_u^j denotes the *j*-th feature element for object *u*, while w_u^j the respective weight. Equation (3) means that the most important feature elements with respect to current user's information needs are characterized by greater weight values in contrast to feature elements of low significance to which low weight values are assigned. Therefore, in (3) the most important feature elements contributes more to the ordering metric $S(u)$. One naïve way to estimate the weights w_u^j are through direct user assignment. However, such a manual process implies that the user has a detailed knowledge about the extracted features which is practically impossible for an average user. An alternative way is to automatically estimate the weight factors w_u^j under an implicit and transparent, to the user, framework, by taking into consideration the pairwise object comparisons during the training phase.

In particular, let us rank objects $u \in S$ in descending order according to the values of $\beta(u)$. Let us denote as $u_{i_1}, u_{i_2}, u_{i_3},...$ the first, second, third … ranked object of *S* respectively. It is held that $\beta(u_{i_1}) \geq \beta(u_{i_2}) \geq \beta(u_{i_3}) \geq ...$. Therefore, if we define the energy for the first *K* ranked objects over all objects of *S* as

$$0 < E(K) = \frac{\sum_{m=1}^{K} \beta(u_{i_m})}{\sum_{m=1}^{card(S)} \beta(u_{i_m})} \leq 1 \tag{4}$$

then we have that

$$E(K) \leq E(K+1) \tag{5}$$

In equation (4) function $E(\cdot)$ has the form of a cumulative density function; monotonically increasing and bounded by unity. Thus, we can estimate the minimum number of objects $\hat{K}$ required so that the cumulative energy $E(\hat{K})$ is greater or equal than a pre-defined threshold value *a*.

$$\hat{K} = \arg\min_{K} E(K) \geq a \tag{6}$$

Having estimated the most $\hat{K}$ relevant objects according to the user's preferences, as provided by the learning algorithm over the training samples of *S*, we can, then, compute the weight factors w_u^j as the inverse ratio of the standard deviation of the respective feature element over all the $\hat{K}$ selected objects.

$$w^j = \frac{1}{std(f_{u_{i_1}}^j, f_{u_{i_2}}^j, \dots, f_{u_{i_{\hat{K}}}}^j) + \delta} \qquad (7)$$

where the *std*(·) denotes the standard deviation operator, while the $f_{u_{i_m}}^j$,*m*=1,2,…, $\hat{K}$ the *j*-th element of feature vector $\mathbf{f}_{u_{i_m}}$. Equation (7) means that feature elements that share similar values among all the $\hat{K}$ objects lead to large weight values (small standard deviation) since in this case the respective feature element seems to be consistent with respect to user's preferences. On the other hand, large deviation values indicate no significance for the respective feature element. To avoid instability issues occurred by a zero division, we have added a small constant value *δ* to the denominator of (7). Table 5 presents a summary of the proposed feature weight estimation algorithm.

Algorithm 2 - Feature Weight Estimation: *Returns Initial Feature Weights*

```
1. Using the Results of the Algorithm 1: (Algorithm 1 is applied for all objects in set S
   and it gets preference values of $\beta(u)$)
2. Rank the objects in the training set S, according to the values of $\beta(u)$
3. Set $a \leftarrow User\ Defined$ and $K \leftarrow 1$
4. Estimate $E(K)$ – see equations (4,5)
5. While $E(K) < a$ do
     a. $K \leftarrow K + 1$
     b. Update $E(K)$, $E(K) = Update\_E(E(K-1), K)$
6. Set $\hat{K} \leftarrow K$
7. Select the feature vectors of all the $\hat{K}$ ranked objects, $\{\mathbf{f}_{u1}, \mathbf{f}_{u2}, \dots, \mathbf{f}_{u\hat{K}}\}$
8. Estimate weight factors for all elements of the feature vectors according to the
   equation (7)
```

Table 5: The main steps of the algorithm used for the estimation of the feature weights.

7.4. Weight Rectification Driven by the Ontology Model

In this section, we improve the weight estimation process by considering metadata interrelations as defined by the ontology. The weight rectification allows a more precise estimation of the weights in cases that few training samples are considered (few user's interactions). This is a very important practical aspect for the proposed personalized route planning architecture, since it permits a very efficient mapping between the extracted low-level features and the high level concepts used by humans to perceive the content, though a small number of training samples are exploited [71].

Let us denote as **A** a matrix that contains the interrelations among all feature elements. Matrix **A** stems from the ontology model described in Chapter 6. Let as also denote as $\mathbf{w}(0)$ a vector that contains all the initial weight factors w^j as estimated by (7). Then, at the *n*-th iteration stage, the weights $\mathbf{w}(n)$ are updated as

$$\mathbf{A}\cdot\mathbf{w}(n-1)=\lambda\cdot\mathbf{w}(n) \tag{8}$$

where λ is scalar factor. Equation (8) can be solved at the steady state through the eigenvector decomposition of matrix **A**. In particular, let us denote as $\mathbf{u}_i$ the *i*-th eigenvector of matrix **A** and as **U** a matrix concatenating all eigenvectors $\mathbf{u}_i$, that is $\mathbf{U}=[\cdots\mathbf{u}_i\cdots]$. Then, we have that the new rectified weights are given by

$$\mathbf{w}^r=\mathbf{U}^T\cdot\mathbf{w}(0) \tag{9}$$

In equation (9), vector $\mathbf{w}^r$ contains all the new rectified weight (see Table 6 for the algorithmic form).

Algorithm 3 - Weight Rectification: *Updates the Weights Using the Ontology Model*

1. Form the matrix **A** from the ontology
2. Set a weight vector $\mathbf{w}(0)$ equals to that provided by the Algorithm 2: Feature Weight Estimation (see 0)
3. Estimate eigenvectors of matrix **A**, say **U**
4. Update the weight factors as equation (9)

Table 6: The main steps of the weight rectification algorithm driven by the ontology model.

8. Route Path Planning – Macro Path Layer

Multi-criteria are involved in this work to estimate the optimal route path that satisfies apart from user's preference additional constraints such as affective the factors of the route, the services offered by the route as well as the route distance, etc. In our approach, a genetic algorithm (GA) is adopted to optimize the overall cost function composing a series of factors. We introduce a path planning implementation with GA. The genotype encoding and the fitness function that respects user's preferences are presented. Moreover hard constrains of the path planning such as distance among the visiting points or the overall distance are considered by the fitness function.

Theoretically, a shortest path algorithm can be used to estimate the optimal route, like for instance Dijkstra [19]. Selecting the shortest path is not a hard problem, but the shortest path is not always what the user wants; what the user really wants to have is the most comfortable route for him/her in a multi-criteria-based framework. However, we can modify the data structure of the graph used in Dijkstra algorithm so as to incorporate additional criteria that affect the selection for an optimal route.

Such modification is valid only if one can approximate the fitness function of a combination through the sum of the cost functions of their pairwise element combination. This case, however, is not true in real-life GIS platforms. Thus, more complicated schemes should be applied like the genetic one. In addition, the shortest path approach lacks scalability; the computational complexity cannot be dynamically adapted to the capabilities of a terminal device (e.g., PDA's, i-phones, etc) something which is more evident for large scale virtual worlds. Instead, a GA algorithm, which is actually a quasi-optimal method, allows a different number of iterations (epochs) to be run during the optimization procedure, allowing a scalable implementation.

8.1. Genotype encoding

Each possible solution has been encoded in a genotype. Each genotype is containing a sequence of the visiting points including the initial Starting Point (SP). A detailed description of the formula of the implemented genotype has been made on the Chapter 3, paragraph 3.2.2.

The initial generation is estimated randomly. Every new generation is produced by the crossover process of two genotypes. The crossover is taking in place by slitting the chromosomes in half (see Figure 7). If the initial number of the containing visiting points is odd, then the second part of each chromosome is bigger by 1 visiting point than the first half of it. Moving forward randomly selected chromosomes are mutated by changing one visiting point with another one that doesn't exist on the aforementioned solution.

There are cases that crossover process or mutation will lead to invalid chromosomes. The length of the genotype is defined by the user before the generation process starts. All the chromosomes have to contain the same number of visiting points otherwise are invalid and they have to be replaced with a valid one. Moreover a chromosome that doesn't respect the hard constrains of total path distance and maximum distance among the visiting points is defined as invalid too. An invalid chromosome is killed and it is replaced by a valid random one.

8.2. Fitness Function

In this paragraph we introduce a fitness function that respects user's preferences and hard topological constrains of the path planning. Let us define as *r* a route. Route *r* starts from a point (SP) and ends to a point (object) Finishing Point (FP). We assume that the route includes *M* out of the *N* available objects in the scene. For instance, assuming that we have a scene of *N*=10 objects and that we would like to construct a route consisting of the *M*=4 most representative objects, starting from the object SP and ending to the object FP then, route *r* can be any of the combinations *Comb*(4,10), say, for example, the SP,1,3,7, 2 (2 is FP). Let us also assume that for each of these *N* objects, features (metadata) have been extracted and metadata weights have been estimated using the learning methodology of Chapter 7. Then, a preference cost function *S*(*u*) is computed for each object *u* of the scene, using equation (3), that measures how preferred is the *u* object for a particular user. It is clear that different users correspond different values of *S*(*u*) for the same object *u*.

Finding the optimal route using as fitness function only the scores *S*(*u*) is a straightforward problem since the only thing we need to do is to select the objects with the highest *S*(*u*) values. This selection, however, is not adequate due to the fact that additional criteria should be included in the optimization process (see Paragraph 5.4). This leads to a maximizing of an aggregate cost function consisting of multiple criteria. These multi-criteria cost functions are discriminated into two main categories; the ones that refer to the overall properties of the route and the ones that refer to objects' properties. Example of the fitness criteria regarding a route are the route distance or aesthetic factors of a route such as a route with many parks or a route passing nearby other points of interests.

The fitness function is considering the score of each individual selected visiting point and emotional map of the path throughout the selected points. In additional the overall distance of the path is considered. The lower the overall distance is, the higher the score of the fitness function is. The emotional criteria of the path are expressed as $E_{n-1,n}$, where indicate the score of the nearby buildings of the path between the visiting point N-1 and N. The fitness function is given by the equation (10).

$$Fitness = (E_{0,1} \cdot S_1 + E_{1,2} \cdot S_2 + \cdots + E_{n-1,n} \cdot S_n) \cdot DFactor_{path}(d_{overall}) \tag{10}$$

$$DFactor_{Path}(d_{overall}) = \begin{cases} 0 & d > OverallMaxDistance \\ OverallMaxDistance - d & d < OverallMaxDistance \end{cases} \tag{11}$$

where $d_{overall}$ is overall distance of the path, $DFactor_{path}$ is a factor that expresses the score of the path based on the distance of it. The OverallMaxDistance is a predefined variable that indicates the maximum possible distance of the path. In case that the distance of the estimated path is more than the OverMaxDistnce threshold, the return score is zero and the chromosome is characterized as invalid and it is replaced by a valid one.

Then, the optimal route is retrieved by maximizing equation (10), i.e.,

$$r = \underset{all\ possiblepath}{\arg\max}\ Fitness \tag{12}$$

It is clear that equation (12) is a combinatorial problem due to the fact that all possible routes should be examined. Therefore, maximization of (12) is not a straightforward task, especially in real-world cases where a large number of N (objects in the scene) and M (objects in the route) are encountered.

A numerical threshold d_n is the perpendicular distance of a neighbour objects to the partial path between the objects n-1 and n. As neighbour objects, all the objects of the 3D scene are considered that haven't been included into the estimated path. In addition, as

Neighbour 3D model, a model can be considered only if its specifications are matching with the user's preferences.

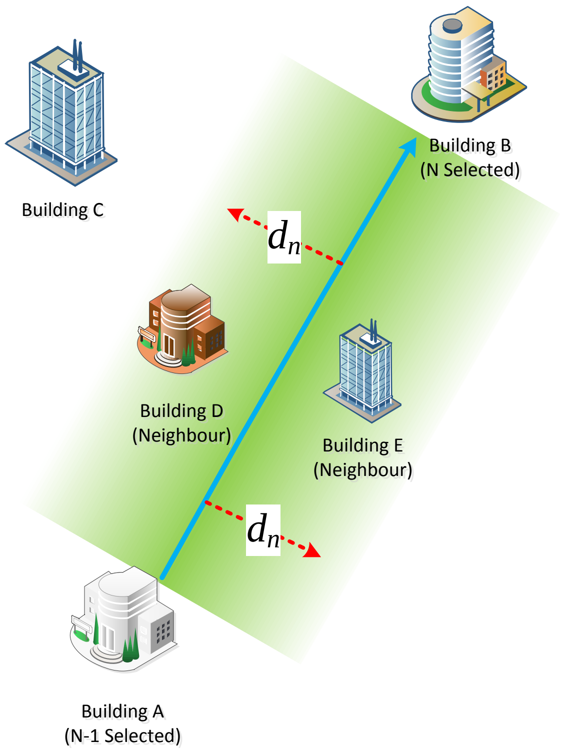


Figure 29: perpendicular distance of a neighbour objects to the partial path between the objects n-1(Building A) and n (Building B)

In the Figure 29 the *Building A* and the *Building B* have been included in the examined solution. The distance of the *Building D* is less than the threshold of d_n and the *Building D* is considered in the cost function as one object of equation (13). On the other hand, the *Building C* is far away from the estimated path. Furthermore, several buildings, such as *Building E,* even if they are close enough to be considered as Neighbours, they might match with the user's desires, according to equation (13).

$$E_{n-1,n} = \sum_{obj} \frac{\sum_i \; w_{(i,obj)} \cdot f_{(i,obj)}}{NeightbourDistance_{(obj,r_p)}} \cdot DFactor_{partial}(dp_{n-1,n}) \qquad (13)$$

where $obj \in R(r_p)$ and $\mathrm{r_p}$ is current point of the path.

$$DFactor_{partial}(dp_{n-1,n}) =$$

$$\begin{cases} 0 & dp_{n-1,n} > MaxPartialDistance \\ MaxPartialDistance \; - dp_{n-1,n} & dp_{n-1,n} < MaxPartialDistance \end{cases} \qquad (14)$$

where $dp_{n-1,n}$ is the distance between the object N-1 and N. The $DFactor_{partial}$ is a factor that expresses the score of the distance between the aforementioned buildings. The MaxPartialDistance is a predefined variable that indicates the maximum possible distance between two selected visiting points.

In case that the distance is more than the MaxPartialDistance threshold, the return score of equation (13) is zero and the aforementioned visiting point S_n is ignored. In that case the chromosome is characterized as invalid and it is replaced by a valid one too.

One approach for solving the aforementioned multi-criteria maximization is to model it as a travelling salesman problem [2]. In the travelling salesman the goal is, given a list of cities (objects in our case), to find the shortest possible route that visits each city exactly once. In our problem formulation, the shortest possible route refers the maximization of a multi-criteria cost function as the one presented in (10). However, again the complexity of this problem is NP-hard, meaning that it is difficult to compute the optimal solution by estimating all possible combinations. Therefore, approximative heuristic solutions have been proposed in literature [2]. Examples include the *nearest neighbor algorithm*, which is a greedy approach, that lets the salesman choose the nearest unvisited city as next move. Although, this algorithm quickly yields a short route solution, the performance is not so satisfactory especially in our case, where several multi-criteria cost functions are involved (see Paragraph 10.4). Additionally, such an approach lacks scalability. The algorithm produces always the same solution even in case that we can tolerate more computational effort to improve the optimization performance. There are also several other approximative heuristic algorithms, such as the *k*-opt or *v*-opt heuristic with similar limitations such as the *nearest neighbor algorithm*. For this reason, in this work, we use a genetic algorithm to solve equation (12). Genetic algorithm is a promising solution in case of NP-hard optimization problems. It provides a framework for a guided and scalable search, reducing the number of combinations needed to obtain the optimal solution.

8.3. Genetic Algorithm

Let us consider as $\mathbf{r}(n)$ a vector that contains the ordered indices of the objects belonging to the selected route *r*. Index *n* indicates the iteration of the Genetic Algorithm. Initially, a population $P(0) = \{\mathbf{r}(0), \ldots, \mathbf{r}_L(0)\}$ of *L* chromosomes (possible routes) is created as a possible solution for the fitness function *C*. Although, in theory, the initial population can be randomly selected, fast convergence is achieved in case the genetic material of the initial chromosomes is of somehow “good quality”. For this reason, in our case, the L initial chromosomes (routes) are selected as the ones having the highest $S(\boldsymbol{u})$ scores among the 3D objects [see equation (3)].

Then, appropriate "*parents"* are selected so that a fitter chromosome (i.e., a route that further decreases the aggregate fitness function *Fitness*) is generated at next iteration. This is achieved by assigning to each chromosome a probability equal to $C(r_i(n))/\sum_{i=1}^{L} C(r_i(n))$ and then selecting the next L chromosomes as candidate parents based on their assigned probabilities through the *roulette wheel selection* procedure [42].

A set of new chromosomes (offspring) is then produced by mating the genetic material of the parents using a *crossover* operator, which defines how the genes should be exchanged to produce next generations. The next step of the algorithm is to apply *mutation* to the newly created chromosomes, introducing random gene variations that are useful for restoring lost genetic material, or for producing new material that corresponds to new search areas. In our case, the genetic material of a chromosome is randomly mutated with a probability of p_m. After that, the next population $P(n+1)$ is created by inserting the new chromosomes and deleting the older ones. Several GA cycles take place by repeating the procedures of fitness evaluation, parent selection, crossover and mutation, until the population converges to an optimal solution. Table 7 shows the algorithmic form of the genetic-based route optimization method.

Algorithm 4 - Macro-Path Estimation

```
1. Define the Start and Finish Point, $SP \leftarrow User\ Defined$, $FP \leftarrow User\ Defined$
2. Set the number of objects in the route $M \leftarrow User\ Defined$ and set $n \leftarrow 0$

   //Produce k different route combinations,
3. For $i$=1 to $k$ do    // $k$ is the number of chromosomes
       a. $r[n][i] = Set\_Route(SP, Rand(M, N), FP)$
          //The function Rand($M$, $N$) returns $M$ randomly selected numbers between 1 and $N$.
4. Set $Best\_Cost \leftarrow Maximum\_Value$ and $Iteration \leftarrow 0$
5. For i=1 to $k$ do      //$i$ indexes the number of routes
```

```
    //Estimate the Cost of the Objects in the Scene
        a. Set $Total\_Cost\_of\_Objects \leftarrow 0$
        b. Set $Total\_Cost\_of\_Route \leftarrow 0$

        c. For j=1 to $M$+2 do  //$j$ indexes the number of objects
            i. u=Retrieve_Object( $r[n][i], j$) //The function Retrieve_Object( $r[n][i], j$) returns the $j$-th element from the combination $r[n][i]$
            ii. Set $S(j) \leftarrow Equation(6)$ //Estimate cost function of the u object using equation (6)
            iii. $Total\_Cost\_of\_Object \leftarrow Total\_Cost\_of\_Object + S(j)$
            iv. Estimate the cost function of the route $r[n][i]$,
                $E(m-1,m) \leftarrow Equation(13)$ Emotional cost of the partial path between the object m-1 and m

                //Estimate the Overall Cost
            v. $Total\_Cost \leftarrow Total\_Cost + E(m-1,m) \cdot Total\_Cost\_Object$

        d. $Total\_Cost \leftarrow Total\_Cost \cdot OverallDistance$

        //Estimate the Best Route
        e. If ($Total\_Cost > Best\_Cost$) do
            i. $Best\_Cost = Total\_Cost$
            ii. $Best\_Route = r[n][i]$

6.  // Crossover the $k$ routes to generate $k$ new routes
7.  For $i$=1 to $k$ do
        a. $r[n][i] \leftarrow mix(r[n][1],...,r[n][2])$ // Crossover the elements of $r[n][i]$ to perform new routes

    // Apply mutation to the q new generated objects
8.  For $i$=1 to q do
        a. $r[n][i] \leftarrow mute(r[n][i])$// Randomly change the elements of $r[n][i]$

    //Check elements validation
9.  For $i$=1 to $k$ do
        a. If (notvalid( $r[n][i]$) do
                    i.  $r[n][i] \leftarrow create()$ // Kill and replace invalid element with a random one

  10. If ($Iteration < Max\_Value$) Go to Step 5;  Else Stop.
```

Table 7: The main steps of the Macro Path Estimation Algorithm

9. Estimation of Object Best Viewing-Micro Layer

In this section, we introduce a new best viewing algorithm, used in our proposed architecture. An existing algorithm has been adopted and modified in a way to consider semantic metadata of 3D objects for navigation in 3D scene. Moreover the texture image complexity of the 3D objects is considered in order to estimate efficiently the camera speed of the 3D trajectory. This approach is able to present a 3D object in a unique way, based on user's preferences and on special properties of the 3D object.

The Micro-path layer includes the personalized viewing algorithm. This implies that we need to implement a methodology for selecting the best view of a building. An entropy-based oriented scheme has been adopted in this work to estimate the best view selection. The '*Micro Path*' layer can be divided in parts (see Figure 30).

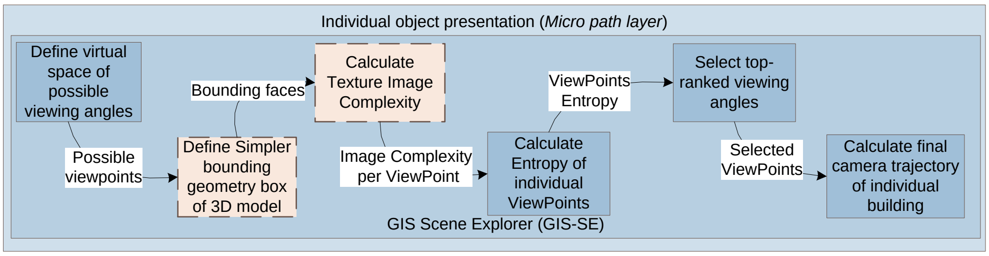

Individual object presentation (*Micro path layer*)
Define virtual space of possible viewing angles
Possible viewpoints
Define Simpler bounding geometry box of 3D model
Bounding faces
Calculate Texture Image Complexity
Image Complexity per ViewPoint
Calculate Entropy of individual ViewPoints
ViewPoints Entropy
Select top-ranked viewing angles
Selected ViewPoints
Calculate final camera trajectory of individual building
GIS Scene Explorer (GIS-SE)

Figure 30: An overview of the proposed personalized route planning architecture

Best Viewing for Micro Path Creation: The final stage and the focus of this part of this work is on the creation of the micro-path, which is a methodology for best viewing of the selected 3D objects of interest, included in the macro-path route. Our approach consists of six individual processes. A virtual space is defined of all possible viewing angles of the 3D model. Afterwards a simpler 3D representative of the object is created.

Then the complexity of the texture of 3D model is considered. Based on the complexity of the texture an indicator is defined. The complexity indicator represents the time that requires a 'visitor' to understand the conceptual content of the texture. The complexity of the image measure is independent of the human perception criteria [10],[11]. An indicator that represents the image complexity is used. As the image complexity measure increases, more time is required by a 'visitor' to understand the conceptual content of the object. Then, as the time of understanding increases, the speed of the camera's trajectory decreases.

According to our assumption, the second and the third step have been accomplished during the reconstruction process of the 3D model. We assume that the real 3D model and the bounding box are the same. Moreover, the complexity of the texture of the image is calculated once during the 3D object modelling process. Generalizing the process, the sequence of the processes has to be as presented in Figure 30.

Afterwards, an entropy based optimization methodology is adopted so that the angle that maximizes the projection entropy is selected for the navigation around an object.

9.1. Definition of virtual space of possible viewing angles

Each 3D model mostly represents a building in a 3D GIS platform. Therefore the geometrical figure of semi sphere was used in order to define the virtual space of all possible viewing points, as it is presented in Figure 31. The virtual semi sphere is adjusted every time, according to 3D model geometrical requirements. For simplifying our experiments we assume that every 3D model is far enough of each other, in a way not to exclude points of the virtual semi sphere.

9.2. Estimating geometrical features of 3D model

Most of the 3D models that are met in a 3D GIS platform are having a simple 3D geometrical model, i.e. they lack geometrical details. As a result, we assume that every 3D model of building or object has a simpler geometry than in reality and that it can be defined by a bounding box that includes the model. All the figures of the model, as they were described in the metadata section, have been assigned to the 5 faces (ground face has been excluded) of this simpler geometrical model. Moreover, an indicator of image texture complexity of each face has been assigned to each face of the bounding box, based on real textures of the 3D building model. The real faces are projected on the bounding box’s ones. According to our experiments we assume that all the 3D models are geometrically described by an orthogonal prism and the real geometry is the same as the bounding box. Due to this fact, the estimation of the bounding box and the calculation of the texture image complexity of the projected faces are taking place on the reconstruction process of the 3D model. In case of generalization in future work, these two steps have to be accomplished according to the functional block of Figure 30.

The estimation of the image texture complexity is calculated based on classification of image complexity introduced by Chanon and Alma [10],[11].

9.3. Entropy-Based View Selection

According to Shannon's theory [57], the entropy of a discrete random variable X with values in the set $\{a_1, a_2, \ldots a_n\}$ is given by

$$H(X) = -\sum_{i=1}^{n} p_i \log_2 p_i \tag{14}$$

where $p_i = \Pr(X = a_i\}$ is the probability of the random variable X to take the value of a_i and the $\log_2 p_i$ the 2-base logarithm of p_i. For continuity reasons the $p_i \log_2 p_1 = 0$ in case of p_i=0. Similarly, if we assume a set of k faces for a particular object u, we can use as probability distribution the relative area of the projected faces over the sphere of directions centred in the viewpoint p, that is [20],[65]

$$H(\xi, u) = -\sum_{i=0}^{k} \frac{A_i}{4\pi} \log \frac{A_i}{4\pi} \tag{15}$$

where A_i is the projected area of face i, while ξ denotes the view (2D plane) over which the 3D object is projected onto. Equation (15) implies a spherical coordinate system and thus $A_i / 4\pi$ indicates the visibility of the i-th face with respect to the point p.

The main difficulty of equation (15) is that the projected areas A_i cannot be directly computed. For this reason, in this work, we adopt an approximate calculation, based on number of pixels of the i-th face projected onto a 2D plane over the total number of object pixels. In this sense, equation (15) can be written as

$$H(\xi, u) = -\sum_{i=0}^{k} \frac{NP_i}{NP_{total}} \log \frac{NP_i}{NP_{total}} \tag{16}$$

The problem with equation (16) is that the entropy at a particular view ξ is not related with user's preferences. For this reason, we extend equation (16), in this work so as to include user's preferences in the best object view selection process.

In particular, let us recall that user's preferences are modelled through the weights $\mathbf{w}^r$ [see Paragraph 7.3 and equation (9)]. In the following, we omit superscript (r) for simplification purposes, since we assume that all the weight values have been properly rectified by the ontology. The weight element w_u^j expresses the degree of importance for the j-th feature element of the u-th 3D object. Since the j-th feature element refers to one of the multiple faces that 3D object is decomposed to, we can derive a personalized entropy using the values of w_u^j. In particular, let us denote as $\psi(u,f)$ an operator that returns the weight indices which are related with those metadata that describe the f-th face of the u-th object. For example, let us assume for a 3D object that the material of one face consists of gypsum and the material of the other face of marble. Then, operator $\psi(\cdot)$ returns the weight indices referring to the marble feature element only in case that the specific face has this property. Therefore, the personalized entropy for a particular object u is defined as by the equation (17) [71].

$$H(\xi,u) = -\sum_{i=0}^{k} \frac{NP_i}{NP_{total}} \cdot \frac{\sum\limits_{for\,\mathrm{all}\; n\in\psi(u,i)} w_u^n}{\sum\limits_j w_u^j} \log\left(\frac{NP_i}{NP_{total}} \cdot \frac{\sum\limits_{for\,\mathrm{all}\; n\in\psi(u,i)} w_u^n}{\sum\limits_j w_u^j} \right) \tag{17}$$

Using equation (17), we can estimate the best object view that maximizes the entropy $H(\xi,u)$. The main problem for performing the aforementioned maximization is that the number of projections ξ is infinity. To overcome this difficulty, we evaluate $H(\xi,u)$ over discrete (limited) viewpoints on the spherical coordinate system that encloses the 3D object. The spherical coordinate system and the discrete points used to estimate best object views is depicted in Figure 31 around a 3D object of a church.

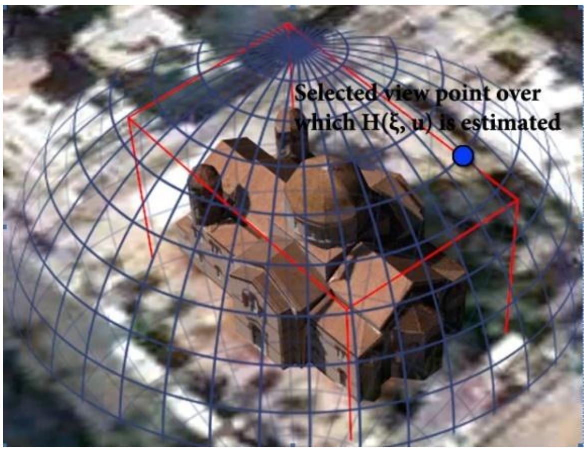


Figure 31: The spherical coordination system used to estimate the best micro-path view through the maximization of the entropy $H(\xi,u)$

Table 8 shows the main algorithm of the viewpoint entropy estimation.

Algorithm 5 - ViewPoints Entropy Estimation

```
1. Set azimuth angle $Phi \leftarrow 0$ and inclination angle $Theta \leftarrow 0$
2. Set $Best\_View \leftarrow Mimimum\_Value$
3. For Theta=0 to 90, Theta+=Step do
    a. For Phi=0 to 360, Phi+=Step do
        i. $Entropy \leftarrow Equation(4)$
        ii. If ($Entropy > Best\_View$)
            1. $Best\_View \leftarrow Entropy$
            2. $Best\_Phi \leftarrow Phi$, $Best\_Theta \leftarrow Theta$
```

Table 8: The main steps of the Micro Path Estimation Algorithm (ViewPoint Entropy)

9.4. Estimation of the Camera Speed

Besides the entropy we introduce an indicator for texture image complexity. Each face of a bounding box has been characterized by a discrete value ‘Little Complex’, ‘Less or More Complex’ or ‘Very Complex’, represented by numerical values from zero to two. It is clear that the more complex a scene is the lower the camera speed is; thus complexity of the scene and camera speed is inverse proportional. Equation (18) expresses the measure regarding image texture complexity, as a function of entropy. The feature of image complexity I_i is calculated during the creation or reconstruction of the 3D model, for each face of it. It is calculated based on

$$C(\xi,u) = -\sum_{i=0}^{k} \frac{I_i}{4\pi} \log \frac{I_i}{4\pi} \tag{18}$$

where I_i is the image complexity of face i, while ξ denotes the view (2D plane) over which the 3D object is projected onto. Table 9 shows the main algorithm of the camera speed estimation based on the texture image complexity.

Algorithm 6 - Camera Speed Estimation based on Texture Image Complexity

```
1. $Camera\_Speed \leftarrow Normal\_Speed$
2. For each Best_Phi as Visiting_Point
    a. $Image\_Complexity \leftarrow Equation(5)$
    b. Switch (Image_Complexity)
        i. Case >=Very_Complex
3. $Camera\_Speed \leftarrow Extremly\_Low\_Speed$
    a. Case >=More_Less_Complex
        i. $Camera\_Speed \leftarrow Low\_Speed$
    b. Case >=Little_Complex
        i. $Camera\_Speed \leftarrow Normal\_Speed$
```

Table 9: The main steps of Camera Speed estimation

9.5. Calculation of Final Camera Trajectory

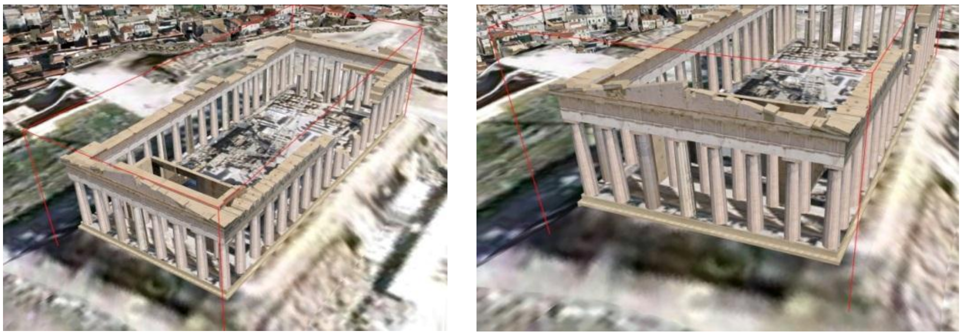
This is the final step of the Micro path component. All the data are gathered and the camera trajectory is calculated from the top ranked viewpoints. The final path respects the requirements of the “Macro Path” by defining the starting and ending point of the camera trajectory in order to create a smooth overall path of all visiting points (scenes) on the whole 3D GIS platform.

In order to understand the results of the personalized scene understanding algorithm, several use case scenarios will be presented. In our implementation, each projected view of the 3D model is calculated on discrete points around the semi spherical coordination system (see Figure 31 for more details). Then, for the given set of selected discrete view points, the ones that maximize the overall personalized entropy metric for all the 5 faces are selected as the most appropriate for the user. In other words, the personalized entropy metric includes not only geometric properties of the projected view but, in addition, the importance of a face with respect to user's preferences.

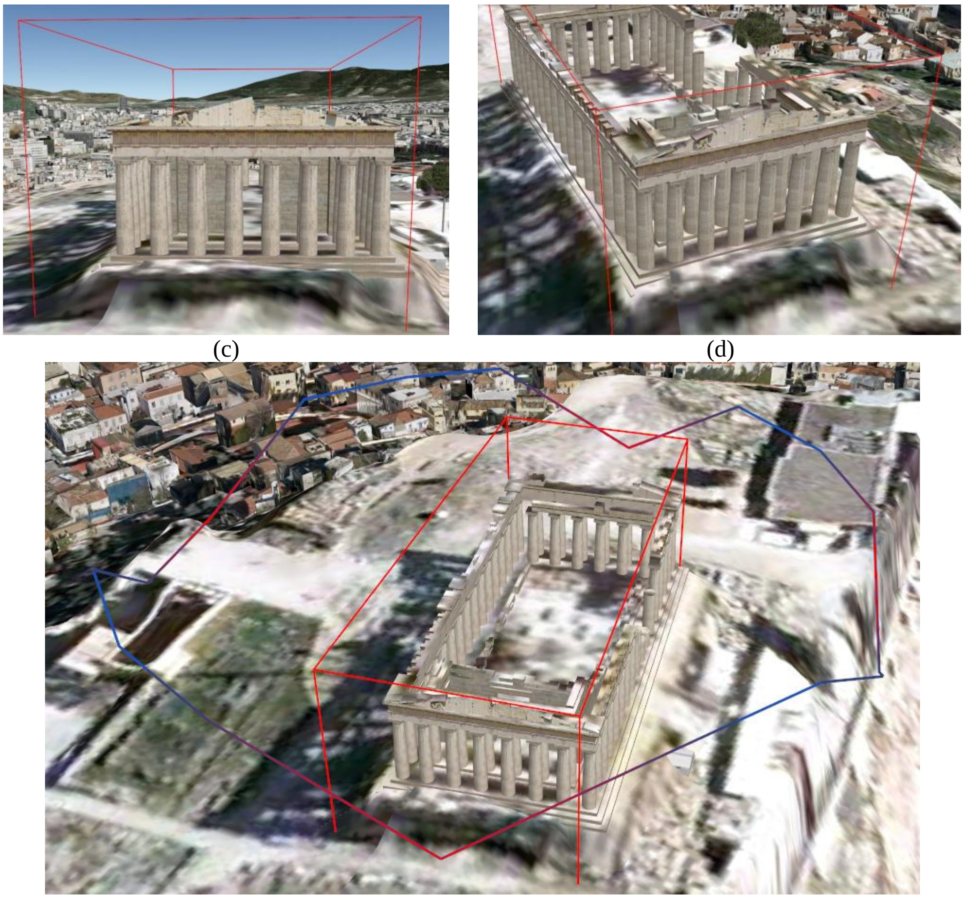
We assume that the user desires to visit Athens for sightseeing. Moreover, it is assumed that a user has already created a profile using the active learning component of GIS-SE. Macro path component provide the list of selected visiting points (sightseeing).

Among the selected buildings, Parthenon was chosen. As it is observed, Figure 32(a) indicates the projection that maximizes the total number of projected points of the view. Instead, Figure 32(b), (c) and (d) indicate the selected view where additional criteria (apart from the total number of projected points on the 3D image plane) that capture user's preferences are incorporated in the best view selection.

Parthenon is included in a bounding box, each of the four surrounded faces of the bounding box have the same vector values ("*Made from Marble*", "*Provides Historical Education*", "*Provides Cultural Education*", "*Is_Accient*", "*Popularity*"), but the upper face of the bounding box doesn't have "*Made from Marble*" (there is not any roof). Besides the specifications of the 3D model, the two faces that contain the entrance of Parthenon has been defined as "*very complex*" image texture complexity and the rest rounded faces as "*little complex*". According to user's profile, "*Made from Marble*" vector is preferred by the user. The aforementioned case leads the best viewing "*Micro Path*" algorithm to rank higher viewing angles such as Figure 32(b), (c) and (d) rather Figure 32(a). Moreover the camera speed with be slower among viewing points of Figure 32(b) to Figure 32(c) and then to Figure 32 (d) rather than any other viewing point of the 3D model.



(a) (b)



(c) (d)

(e)

Figure 32: (a) Best view selected using only geometrical properties. (b)(c)(d) Best view selected using geometrical and human factors. (e) Indicating camera trajectory and the speed of it for the selected viewpoints

In Figure 32(e) top ranked viewpoints have been marked. The camera trajectory is presented by red-purple-blue line around the 3D model of Parthenon. The entropy is maximized when the camera is placed straight ahead of the bounding faces (of red bounding box). There are no selected viewpoints from the top of Parthenon because this it is not made from marble. The camera speed change has been marked by the change of colour from blue to red .Blue indicated that the camera speed is “*Normal*”, purple means that the camera speed has to be “*Low speed*” and red means that the camera speed set to be “*Extremely Low Speed*”.

10. Platform Implementation

In this section, we provide implementation details for the proposed personalized route planning architecture and we also apply simulations to assess the performance of each individual component as well as of the overall system platform.

10.1. Implementation Details

The platform has been developed using the Google Earth Building maker tools. The buildings are reconstructed as 3D models, each face of which is described using a set of metadata. We assume that all the 3D objects have five faces (the faces that point to the ground are not considered), no matter of their 3D complexity. In particular, we represent all 3D objects as a "*bounding-box*" of five faces around the object as indicated in Figure 33 where the 3D model of the national Gallery/Library of Athens is presented. The platform exploits the Google Earth plug-in and the Google Earth virtual environment to visualize the 3D geographical data. We have developed a front end interface to handle user's interactions with the system and a back end module, called GIS Scene Explorer (GIS-SE), which contains the learning component, the Macro-path and the Micro-path processes as described in the aforementioned sections. Since Google Earth does not allow retrieval and management of 3D geographical content, a separate geo-referenced database was used for this purpose in this work. The morphology and the texture of the ground were retrieved by the Google Earth's database to increase the overall scene visualization.

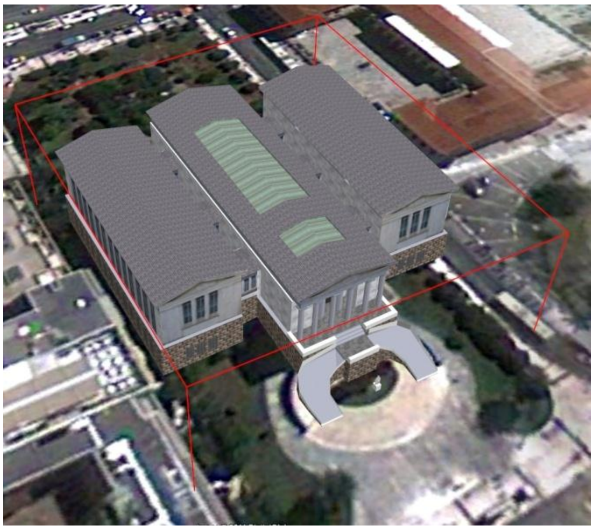


Figure 33: The 3D model representation of National Gallery/Library of Athens.

In our simulations, the geographic area of the city of Athens in Greece was selected. To verify the generality of our results, we have formed a large dataset of 700 3D objects of different thematic categories (e.g., historical buildings, parks, contemporary houses, old factories, etc). In addition, the results have been derived using 15 different experiments of routes of different start and end points and the average precision-recall values have been depicted.

10.2. Evaluation Metrics

The most common objective criterion used for evaluating the performance of a personalized route planning system is through the *precision-recall curve*. We denote as *Precision* Pr, the ratio of the number of relevant retrieved objects, say *R*(*M*), over the total number of objects, say *M*, including in a selected route path [56]. That is,

$$\Pr(M) = \frac{R(M)}{M} \tag{19}$$

On the contrary, recall Re is defined as the ratio of the number of relevant retrieved objects *R*(*M*) over the total number of relevant objects including in the whole scene of the 3D world, say *G* [56]

$$\mathrm{Re}(M) = \frac{R(M)}{G} \tag{20}$$

For a "perfect" system, both Pr(*M*) and Re(*M*) should be high [56] (ideally equal one). However, in real situations, as the number *M* increases, precision decreases, while recall increases. This means that as the route complexity increases, it is more probable irrelevant objects, with respect to user's preferences, to be selected thus decreasing the precision ratio. On the other hand, complex paths impose higher probability to select more relevant objects among all possible relevant ones resulting in an increase of the recall values. Because of this, instead of using a single value of Pr(*M*) or Re(*M*), the Precision-Recall curve is usually adopted to characterize the performance of a system.

In equations (19) and (20), precision and recall have been estimated for a specific path. However, to evaluate the overall performance of the system, many paths should be created and then, for each path the precision-recall curve needs to be computed. The average values over all routes yield the total precision-recall metrics. Let us assume that Q paths have been examined, then, the average precision/recall, APr, ARe are defined as

$$\mathrm{APr}(M) = \sum_{q=1}^{Q} \mathrm{Pr}(M) \qquad \mathrm{ARe}(M) = \sum_{q=1}^{Q} \mathrm{Re}(M) \tag{21}$$

10.3. Incremental Learning

The incremental learning process is suitable for determining user's preferences based on simple user's feedback given in a form of preferences judgments, i.e., statements that indicate that one instance should be ranked ahead of another. In particular, each time the system requires the user to evaluate pairwise comparisons between two 3D models. Then, the learning module recursively updates user's preferences by estimating the most dominant feature elements, used for describing the rich 3D content, with respect to the current user's information needs. In particular, each time a new pairwise comparison is submitted to the system, new weight values concerning the feature elements are estimated using the algorithm of Paragraph 7.3. Feature element interrelationships, described through the ontology, are exploited in order to refine the initially estimated weights. In this way, we conclude to efficient weight estimation strategies that exploit only few training samples.

In order to understand the learning process, we present a use case scenario in which a user desires to visit Athens for sightseeing, since he/she interested in ancient Greek civilization. Additionally, the user is interested in souvenirs regarding ancient Greek culture. In Figure 34, we depict 29 weight vectors. Table 10 indicates the semantic meaning of these 29 weights depicted in Figure 34. In Figure 34, the weights values have been normalized in a range of [0 1]. Weight values close to zero (0) indicate feature elements that do not coincide with user's preferences. On the contrary, weight values around unity (1) represent feature elements of high importance.

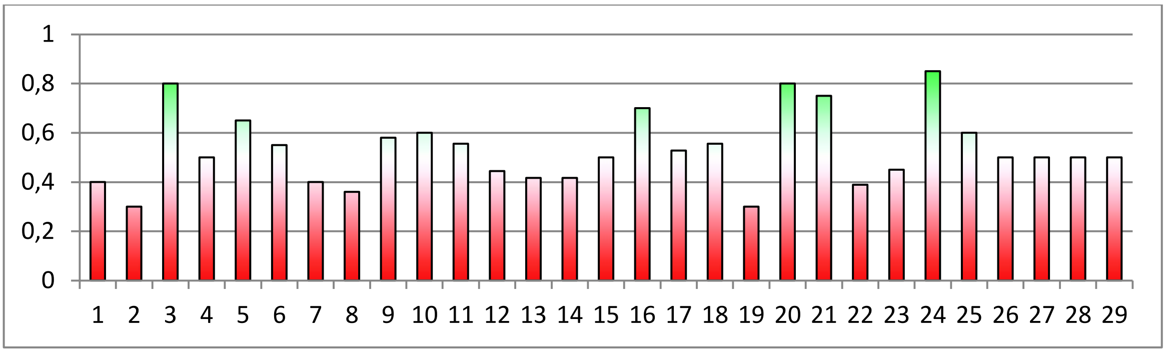


Figure 34: Indicative weight values for a scenario where a user selects the ancient Greek civilization and respective shops of souvenirs are the most interesting places according to his/her information needs.

Features	No	Features	No
New	1	For Shopping	16
Old	2	For Public use	17
Ancient	3	Provides Basic Education	18
Visit Cost	4	Provides Scientific Education	19
Popularity	5	Provides Cultural Education	20
Interactivity	6	Provides Historical Education	21
Is Classical	7	Made from Cement	22
Is Modern	8	Made from Clay	23
Is Neoclassical	9	Made from Marble	24
For Business	10	Made from Stone	25
For Education	11	Made from Wood	26
For Emergency	12	Made from Metal	27
For Entertainment	13	Made from Glass	28
For Relaxing	14	Is Tall	29
For Residence	15		

Table 10: Explanation of the 29 weights depicted in Figure 34

As is observed, feature elements that correspond to “Popularity”, “Ancient”, “Neoclassical”, “Made from Stone”, “Made from Marble”, “Provides Cultural Education”, “Provides Historical Education”, “For Shopping” and “For business” are those selected as most dominant feature descriptors. This is quite straightforward since the user selects Ancient Greek civilization and shops for souvenirs as far as his/her information needs are concerned.

Figure 35 presents the weight values of two selected feature elements versus the iteration of the recursive learning algorithm. As is observed, for consistent user's selections, the weight values converge. It is clear, however, that in case that a user changes his/her preferences, the weight values also change.

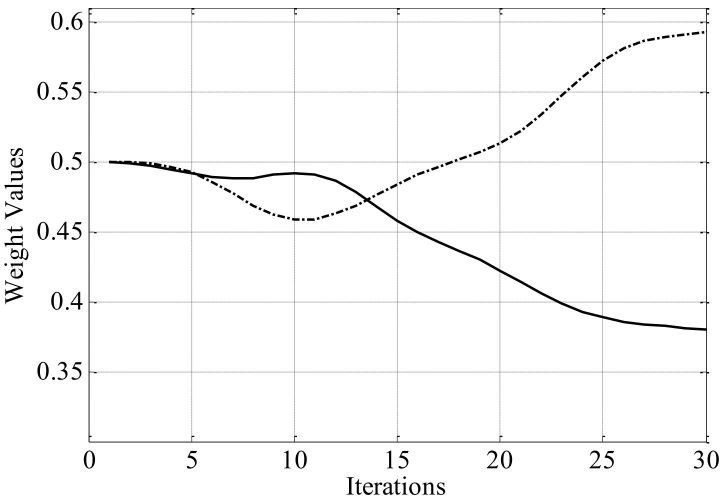
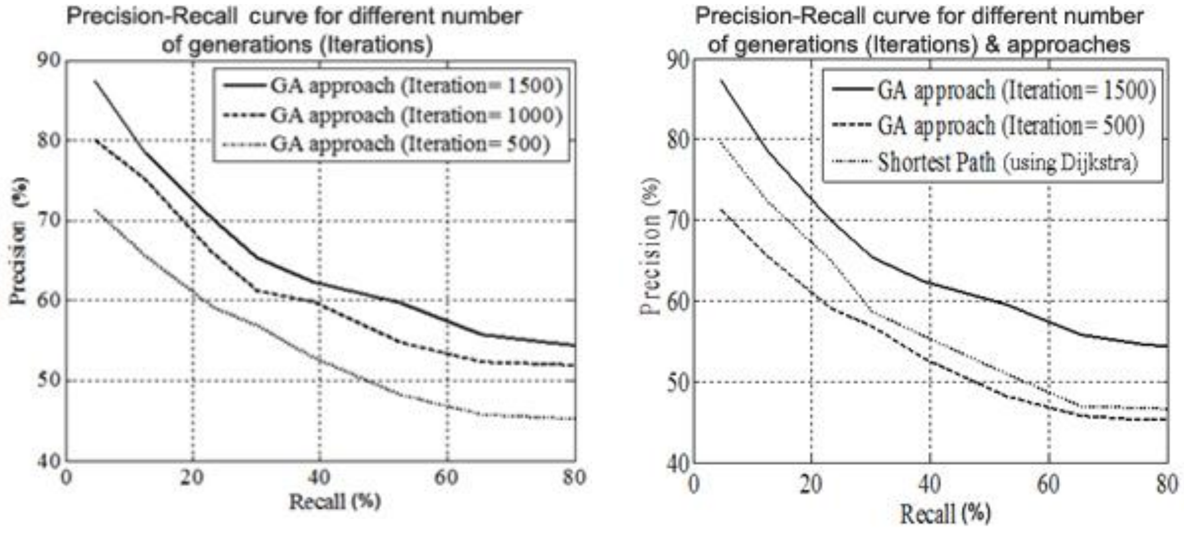


Figure 35: Convergence of the weight values in case that users are consistent in their selections; a two case weight value scenario

10.4. Route Path Estimation-Macro-Path Layer

In this section, we present the results regarding the Marco path estimation. In particular, using the learning component (see the previous Paragraph), the platform is able to estimate user's preferences and express them is a vector of weights, of the preferred features of the 3D models. The extracted features are used by a Genetic Algorithm (GA) in the "Marco-path" process in order to select the most "appropriate" buildings that match user's preferences. The selected set of 3D objects (buildings) was evaluated by precision and recall measurements, in order to objectively evaluate the performance of the proposed method and compare it with other current approaches. Figure 36(a) presents the precision-recall curve in a case where the GA algorithm has been used for the estimation of the optimal Macro path. In this figure, we have presented the precision-recall curve for various numbers of iterations of the GA. It is clear that, as the number of iterations of the genetic algorithm increases, the precision accuracy increases for the given recall value. It is also expected that as recall increases, the precision accuracy decreases since more objects are included in the selected route path, meaning that we have more complex routes. In Figure 36(b), we have presented the performance of the proposed GA approach compared with the current methods that use a shortest path algorithm for the route selection. As is observed the GA outperforms the shortest path algorithm. It should be mentioned that comparison has been performed using the multi-criteria cost function described in chapter of the path planning. In case of the implementation of the shortest path algorithm, we have assumed that the cost function can be approximated as the sum of the cost functions of their pairwise element combination. The aforementioned precision-recall curves have been estimated as an average value over several experiments regarding route path estimation of different numbers of 3D objects participating in the route [71].



(a) (b)

Figure 36: (a) Precision-Recall curve for different iterations of the Genetic Algorithm, (b) Comparison of the Genetic Algorithm with the shortest path algorithm

10.5. Best View Estimation – Micro Path Layer

The final step of the proposed architecture regards the micro-path layer; individualized representation of a selected 3D object according to the maximization of the personalized entropy metric (see Chapter 9). For simplification purposes, the best view has been estimated by exploiting a bounding box around the 3D model (see above for more details). Therefore, each 3D building is modelled using a 5-faces object (the last face corresponding to the ground is not taken into consideration in our algorithm). Figure 37 presents the bounding box of a 3D building (in this example a church) as well as the spherical coordination system over which the best view is calculated.

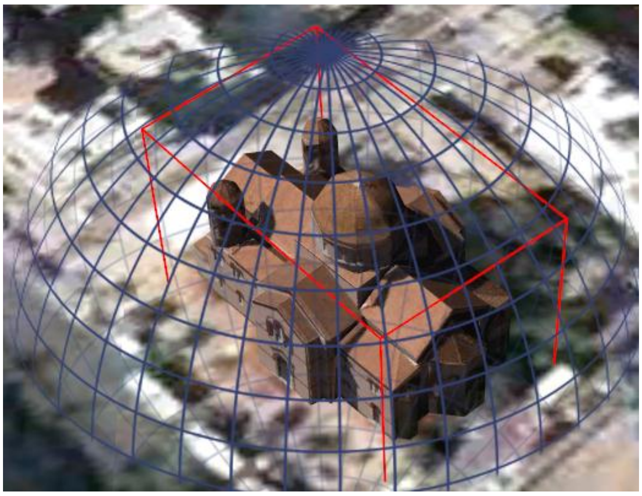
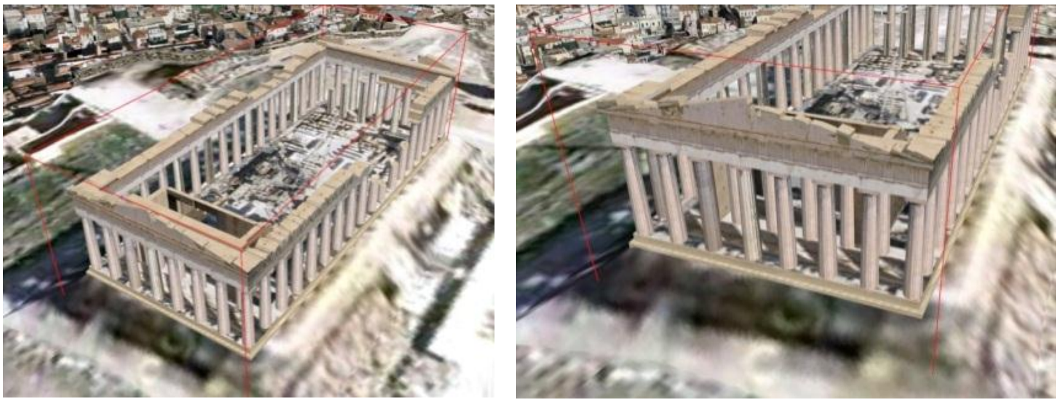


Figure 37: The semi-spherical coordination system used to project 3D objects.

Figure 38 presents the results regarding a best view selection in case that a geometric criterion is used as well as in case that the proposed personalized entropy based metric is adopted. As is observed, Figure 38(a) indicates the projection that maximizes the total number of the projected points of view. Instead, Figure 38(b) indicates the selected view where additional criteria (apart from the total number of projected points on the 3D image plane) that capture user's preferences are incorporated in the best view selection. These criteria are provided through the weight values estimated through the personalization algorithm of Chapter 7.

The best view selection ("micro-path" layer) process was tested for many buildings in the examined 3D environment. In the following, we evaluate the complexity required for estimating the best view of a 3D building. It is clear that the complexity is straightforwardly proportional to the number of examined discrete points on the semi-sphere bounds the 3D. The more the examined points on the semi-sphere coordination system are, the more the complexity is.



(a) (b)

Figure 38: (a) Best view selected using only geometrical properties. (b) Best view selected using geometrical and human factors

10.6. Path Generation

The goal of this research is to estimate the optimal paths. One of the first processes activated after the calculation of the points of the path, using GIS-SE platform, is to visualize it. The trajectory of the camera is calculated by estimating several points of the macro and micro path. Sequentially, connecting these points the camera trajectory is created. The camera trajectory is created in KML format and can be viewed using Google Earth, the same way a path is created using path creation tool of Google Earth.

A path in Google Earth is a KML file which contains an array of points. Each point is defined by three coordinates in the three dimensional space of Google Earth. The coordinates for longitude and latitude are expressed in the global coordinate system. The altitude can be expressed as an absolute value relevant to the sea level or to the ground level. According to our implementation requirements, the altitude is set to be always relevant to ground level.

On the contrary, the 3D models are using a relative coordinate system for the points that comprise them. There is only one absolute reference to the global coordinate system and the rest of the points are relatively defined to that initial point.

Considering all the aforementioned data and the selected relevant optimal view, the absolute global coordinates of the selected points can be calculated. The GIS-SE platform generates a KML file that contains a path just like the one that Google Earth "*Create Path*" tool can generate. In particular, the path generated by Google Earth is an array of coordinates in three dimensional space.

10.7. Generating Tour

Moving forward, our goal is to generate the camera trajectory. In a similar way as the path generation process, the camera trajectory is calculated. In order to achieve this process the best views have to be calculated first. Considering the micro path results about the complexity of the projected view and the estimated camera speed, the final tour is generated. The tour is generated in KML format too. It is composed by a list of points which is called "*FlyingList*". Each point is refered as "*FlyingPoint*" and it contains the coordinates of the camera in the 3D space of Google Earth. Moreover, it includes the horizontal and vertical orientation of the viewing angle. Finally it encloses the duration of the "*flight*" expressed in milliseconds to the next "*FlyingPoint*". For the smooth creation of the tour the flying mode of the whole tour is set to "*smooth*". The orientation of the camera is expressed in degrees, 360 degrees for the horizontal orientation and from 90 to -90 for the vertical orientation.

The exported KML file, according to the aforementioned methodology and calculations is containing the real tour into the three dimensional geo-referenced scene. This file can be viewed and watched by the Google Earth application.

Part III: Evaluation - Discussion & Conclusions

11. Discussion – Conclusions

11.1. 3D building and city modelling evaluation

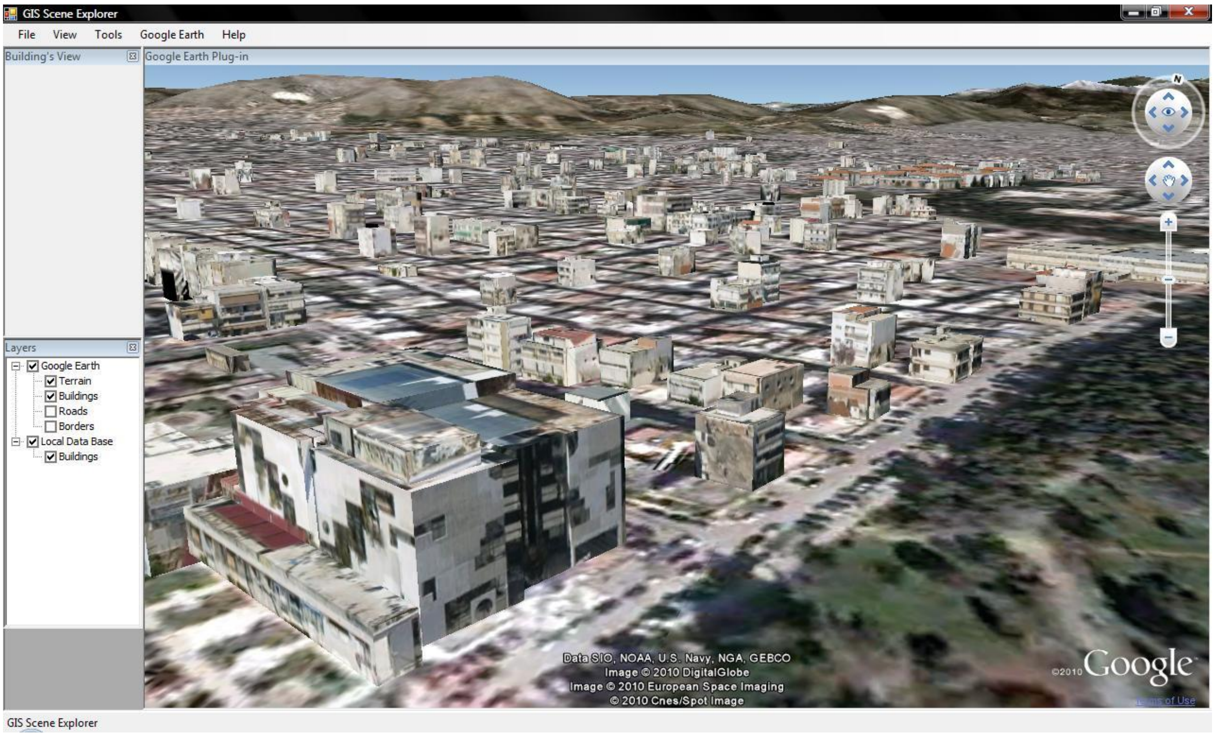
All 3D models of buildings that were required for this research project were created using existing tools that Google provides. These are the Google Building Maker and the Google Sketchup. These tools provide and efficient and easy way to 3D reconstruct a 3D geo-referenced model. In the initial step of this research project, all 3D models were reconstructed using Google Building Maker. Then, corrections of the models took place using Google Sketchup. The choice for the combined use of the aforementioned tools on the geo-referenced 3D reconstruction area is also confirmed by the global commercial success of these tools.

By 2005, and for a long period of time, there were many regions around the world without any 3D building models or with only a few ones. Once the Google Building Maker was introduced by Google in the midst of October 2009, hundreds of three dimensional models of buildings appeared on the GIS system. Many of them have been evaluated by the team of Google and they appear on Google Earth too.

On the other hand all these data that exist on the Google Earth and the 3D content cannot be retrieved. Therefore a collaborative database of a GIS system made possible the modelling of a geo-referenced scene or a city. The participants of the experiment could choose a place to build, or add material in the form of hypertext, data or image on the map. The result of the collaboration was to build a three dimensional scene with about 700 buildings in a very short period of time. Each model was completed within minutes. A screenshot of part of the 3D scene is presented in the Figure 39. The whole city could be modelled in this way distributing the work to all the participants of the team.

As soon as a building model had been constructed, it was placed on the map using the global positioning system. Some building models were good copies of the original buildings whereas some others were poor or incomplete. Each building that was designed had to be evaluated before it is integrated into the 3D scene.

In our approach, each building could be evaluated in a scale from 0 to 10. The evaluation was taking place by a human expert, considering the satellite images of different point of views and 3D model of the buildings. In future versions of our platform, users will be able to evaluate the buildings of others. The buildings featuring correct outline and accurate placement were allowed to be visible in the Google Earth component [69], [70].



- ☑ Google Earth
  - ☑ Terrain
  - ☑ Buildings
  - □ Roads
  - □ Borders
- ☑ Local Data Base
  - ☑ Buildings

Figure 39: Some of the reconstructed 3D buildings of the city of Athens

11.1.1.3D Building and Obstacles Overlapping

Building a model with the aforementioned tools, although a relatively trivial task, does not always have the desired result. The accuracy of the model and the detail of the texture are not always the expected ones. There are cases that the satellite or aerial photographs are not clean, or there are obstacles in front of the building. The obstacles can be anything, trees, big cars, etc., the most common problem being the partial overlap of the buildings. This problem is very common in cities, where one building is close to the other. In the Figure 40 (Figure 26 repeated for convenience) there is a green part at the bottom of the texture image of some buildings. This occurred due to tall trees in front of the building.

If there is no spare space between the buildings, one solution is to create the 3D model of the whole complex of buildings. Hence, the whole building complex has to be modelled up to the point where a clear area is reached. Likewise, if there is a small area between the buildings, then the texture among the nearby buildings will be erroneously applied on the sides of them.

Problems with low quality of images or texture images with noise (obstacles mapped on them) can only corrected by replacing the image with a real photograph from the proper aspect of view. All the models are implemented in COLLADA graphics language. They can be changed either manually or using a CAD tool compatible with COLLADA. Google Sketchup is one famous tool for that purpose. Moreover, a faulty model may occur if the bounding boxes or pyramids do not fit correctly on the satellite or aerial photographs. In that case, the produced 3D model is poor in quality and typically rejected during the evaluation process.

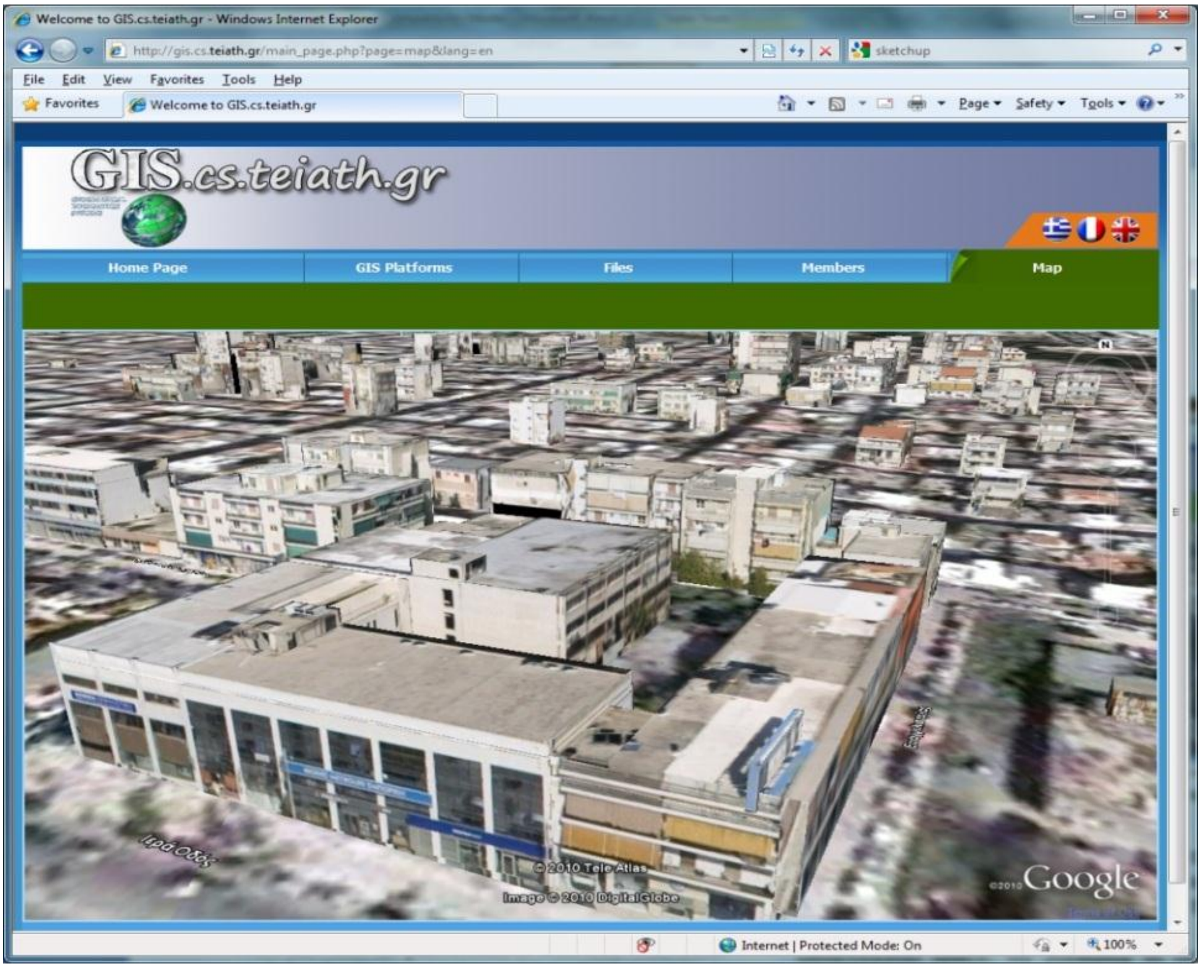

Welcome to GIS.cs.teiath.gr - Windows Internet Explorer
http://gis.cs.teiath.gr/main_page.php?page=map&lang=en
sketchup
File Edit View Favorites Tools Help
Favorites
Welcome to GIS.cs.teiath.gr
Page ▾ Safety ▾ Tools ▾
GIS.cs.teiath.gr
Home Page
GIS Platforms
Files
Members
Map
© 2010 Tele Atlas
Image © 2010 DigitalGlobe
©2010 Google
Internet | Protected Mode: On
100%

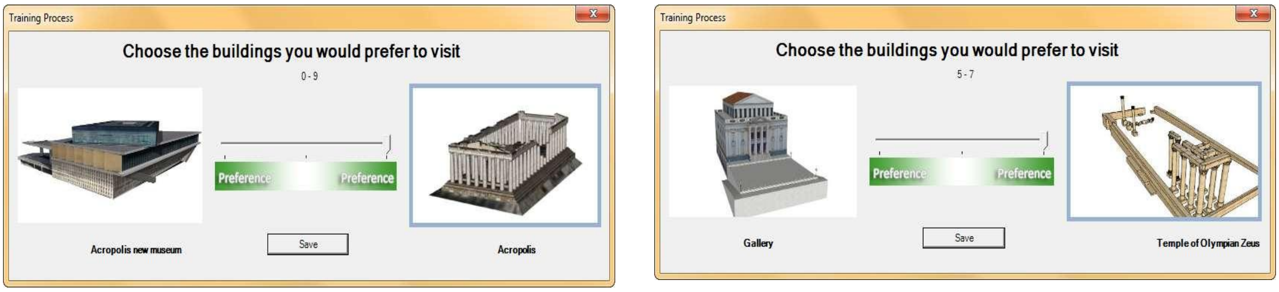
Figure 40: Experimental set of 3D models of building

11.2. Incremental Learning

Initially a standard set of ten predefined objects is used to get an approximate estimate of user’s preferences. In particular, all possible combinations of pairwise comparisons of the ten models are considered (forty five combinations) to create the initial user profile.

Afterwards, the learning module recursively updates user’s preferences by estimating the most dominant feature elements, to fit the current user’s information needs. More specifically, each time a new pairwise comparison is submitted to the system, new weight values are estimated using the algorithm of Paragraph 7.3. Feature element interrelationships, described through the ontology, are also exploited in order to refine the initially estimated weights. In this way, we conclude to a reliable weight estimation even in cases that few samples are submitted to the system.

In order to understand the approach adopted in this work, we present in the following a use case scenario that clarifies system operation. In this scenario a user wishes to have a virtual tour in the city of Athens and therefore he/she uses the proposed adaptive personalized route guidance platform. The ultimate goal of the platform is to select a 3D navigation route that satisfies his/her preferences. In order to achieve this, initially a set of pairwise comparisons is depicted to the user for evaluation purposes, as preference judgments, that indicate which of the two presented objects is preferred against the other. Figure 41 presents four characteristic screenshots, which clarify the interactive training process. In this particular example, the user selects the Acropolis archaeological place against the Acropolis Museum [see Figure 41(a)] while in Figure 41(b) he/she selects the Temple of Olympian Zeus against the National Gallery/Library of Athens. Finally, Figure 41(c) and (d) indicate that the user better prefers buildings of Neoclassical architecture[1] such as the Greek parliament and the National library of Athens against buildings of general purpose and public use (e.g., hotels and town hall buildings).


Training Process

Choose the buildings you would prefer to visit

0 - 9

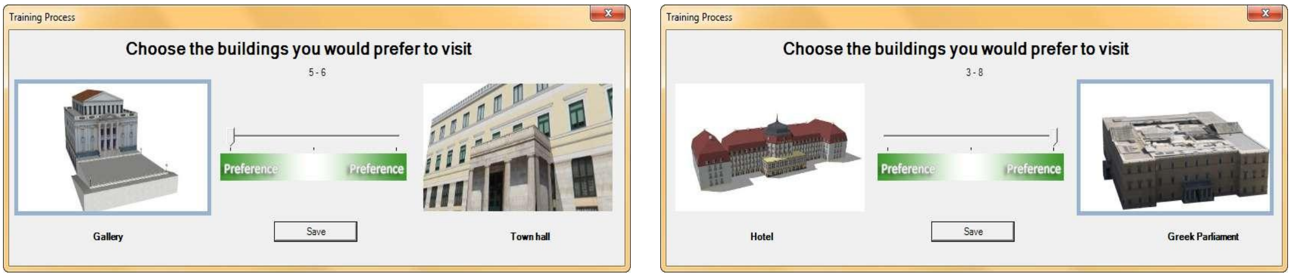
Training Process

Choose the buildings you would prefer to visit

5 - 7

(a) (b)

[1] Neoclassical architecture is an architectural movement that began in the mid-18th century as a style principally derived from the architecture of Classical Greece and Rome combined with recent architectural movements of 18th century



Training Process

Choose the buildings you would prefer to visit

5 - 6

Training Process

Choose the buildings you would prefer to visit

3 - 8

(c) (d)

Figure 41: We present four screenshots as an explanation of our approach used to estimate user’s preferences through the interaction process given in the form of preferences judgments (i.e., statements that indicate that one object should be preferred against of another) that are then fed back to the system. a) The New Acropolis Museum is presented against the Acropolis Archaeological site and the user selects the Acropolis site. (b) The temple of Olympian Zeus is selected against the National Gallery/Library. (c) The National Gallery/Library of Athens is selected against the Town Hall. (d) The Greek Parliament is chosen against a hotel to indicate preferences in architectural style.

Then, the proposed online learning algorithm takes place to automatically adjust metadata weights and thus to model user’s preferences. Within a few number of comparisons, the algorithm converges to a reliable set of user’s preferences expressing through the weights of objects’ metadata. The ontology driven weight rectification strategy supports the reliable convergence of metadata weights [71].

In this experiment, we have used 29 metadata features to describe the 3D object, each of which is associated with a relative weight. Figure 42 visualizes the convergence rate of the weights for this particular experiment. In this figure, we have depicted only 18 weights out of the 29 in total, just for readability purposes. As is observed, the weights that are associated with the “Ancient”, “Made from Stone” and “Made from Marble”, “Historical Education” and “Neoclassical” metadata receive high relative values, since these metadata describe better the archaeological places and the neoclassical architectural style which are preferred by the user. The respective weight values indicate that the platform selects a 3D personalized route that primarily composes, as much as possible, of archaeological places followed by Neoclassical buildings. Instead, in systems where the weights are set manually, it is quite difficult for simple users to exactly quantify their preferences by accurately defining the respective weights for all objects’ metadata.

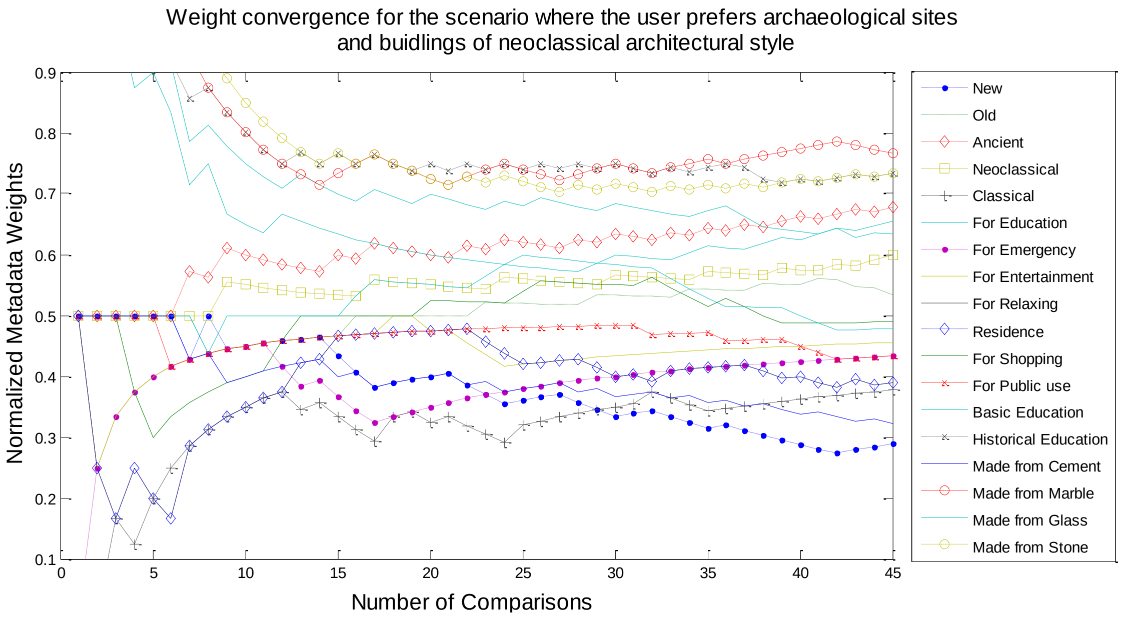
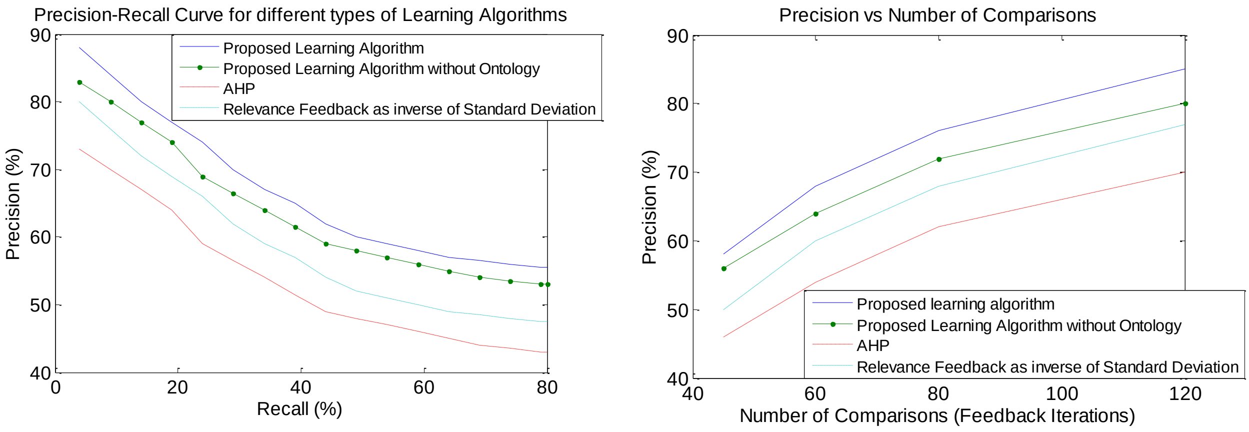


Figure 42: Weight convergence versus the number of pairwise comparisons for a scenario where the user selects archaeological sites and buildings of neoclassical architectural style.

The advantages of the proposed architecture, compared with existing personalized route guidance systems, is the incorporation of the relevance feedback learning component, which has the capability of automatically adjusting the objects' metadata weights to capture user's preferences. Instead, in the existing approaches, user's preferences are manually defined by allowing users to adjust the weight values according to their information needs. However, such an approach presents the drawback that manually setting all the weights by the users or performing an overall ranking of all objects within the scene is an arduous task which is practically impossible to be implemented, especially in large scale virtual scenes, where thousands of 3D objects are involved.

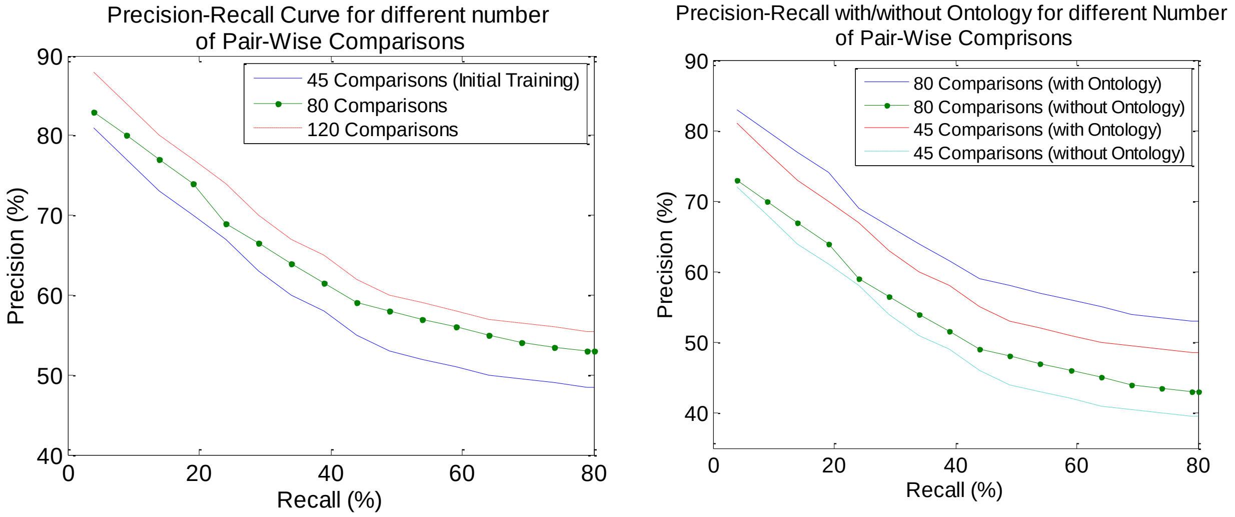
In the following, we present experimental results that indicate performance of the proposed relevance feedback component in contrast with other learning approaches. Quantitative comparisons have been performed using the objective criteria described in Paragraph 10.2.



(a) (b)

Figure 43: Comparisons of the proposed relevance feedback learning component with other online learning strategies and experimental documentation of the ontology driven rectification mechanism. (a) The precision-recall curve for various learning algorithms versus the proposed one. (b) The precision versus the number of feedback iterations to illustrate the effect of number of pairwise comparisons on the overall system performance. The results have been obtained at recall 50%.

Figure 43 (a) presents comparison results of the proposed relevance feedback learning strategy against other learning methods, such as the method of [53] (relevance feedback as inverse of standard deviation), in which the weights are updated inversely proportionally to the standard deviation of the metadata over the relevant selected 3D objects, being retrieved on previous iterations and the Analytic Hierarchical Process (AHP) algorithm [54] in which the weights are ranked with respect to the outcome of a pairwise comparison. Additionally, in this figure, we have presented the effect of the ontology-driven weight rectification component on precision-recall curve. As we observe, the proposed algorithm outperforms the compared ones in terms of precision at a given recall value. Figure 43(b) illustrates the effect of precision versus the number of feedback iterations which is directly related with the number of pairwise comparisons executed in the proposed adaptive personalized route guidance system. All results have been obtained at recall 50%. In particular, we assume that in each feedback iteration, a user interacts with the system selecting preferred (relevant) objects between two depicted ones. As is observed, precision increases as feedback iterations increase. However, the improvement ratio is saturated meaning that beyond a certain limit of iterations no significant improvement is encountered. In this figure, we also notice that the ontology-driven weight rectification strategy significantly improves precision due to the exploitation of the metadata relations involved. This means, in other words, that the ontology scheme forces the algorithm to localize "representative" samples able to cover the whole variations of the feature space. Figure 44 shows the number of pairwise comparisons on the system performance as regards the precision-recall curve. Again, we observe that the ontology-driven weight rectification strategy improves precision for all recall values even though a small number of samples are selected.

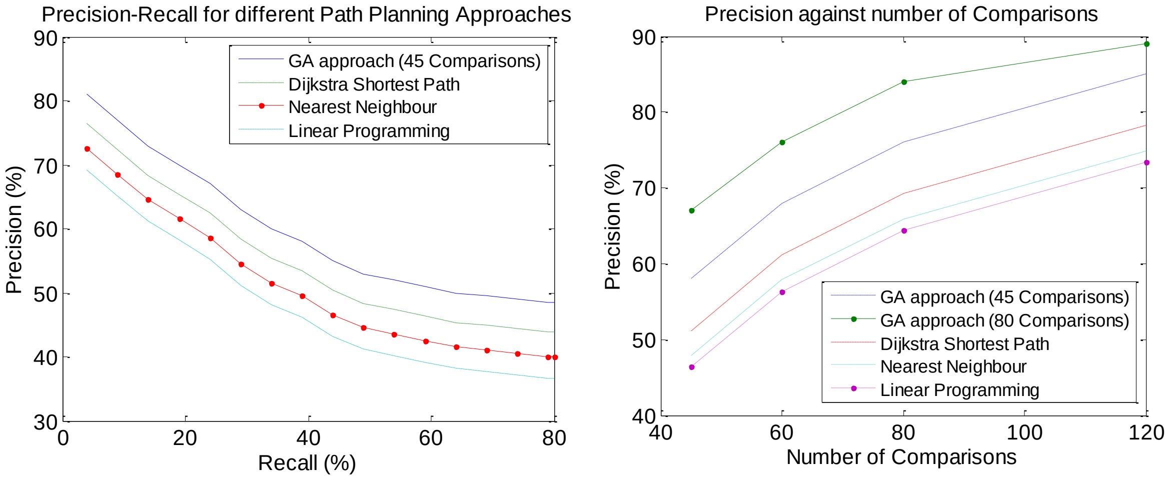


(a) (b)

Figure 44: The effect of the number of selected samples on the precision-recall curve. (a) Precision recall for different numbers of comparisons (feedback iterations) as regards the proposed learning strategy. (b) The effect of ontology on precision-recall for different numbers of comparisons.

11.3. Route Path Estimation – Macro-Path Layer

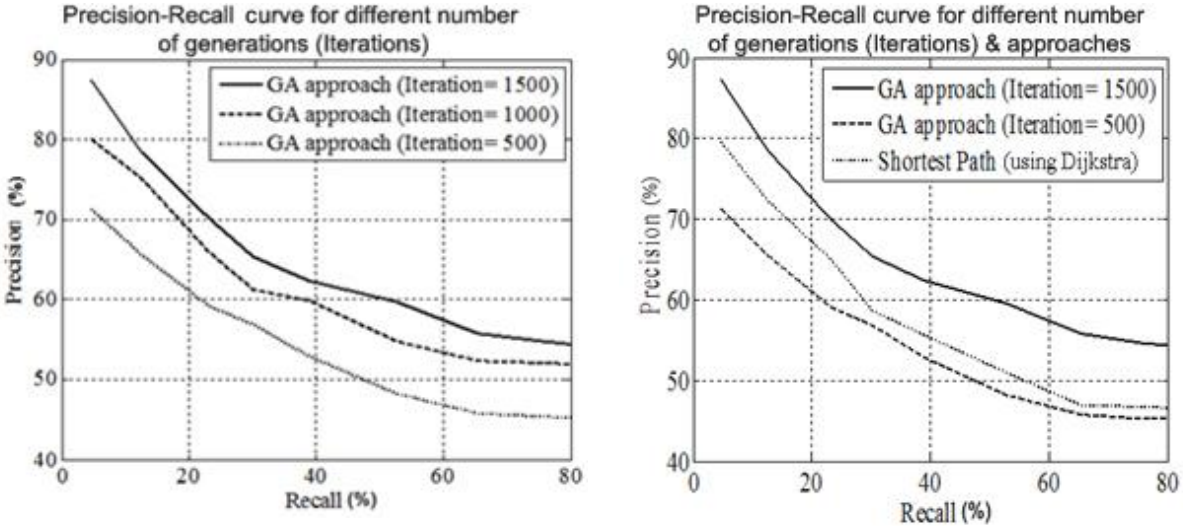
Simulations are performed using three different methodologies; (a) the Dijkstra shortest path method [19], (b) the nearest neighbor algorithm which provides a heuristic approximate of the travelling salesman problem [24] and (c) a linear programming optimization policy [2]. Figure 44 compares the proposed scheme against the three aforementioned route planning algorithms in terms of precision-recall. In all experiments, the proposed genetic optimization (GA scheme) has been implemented at 1,500 iteration cycles. As is observed, the proposed GA scheme outperforms the compared ones.



(a) (b)

Figure 45: Comparisons of the genetic based route planning algorithm with other existing optimization strategies. (a) The precision-recall curve. (b) Precision versus feedback iterations.

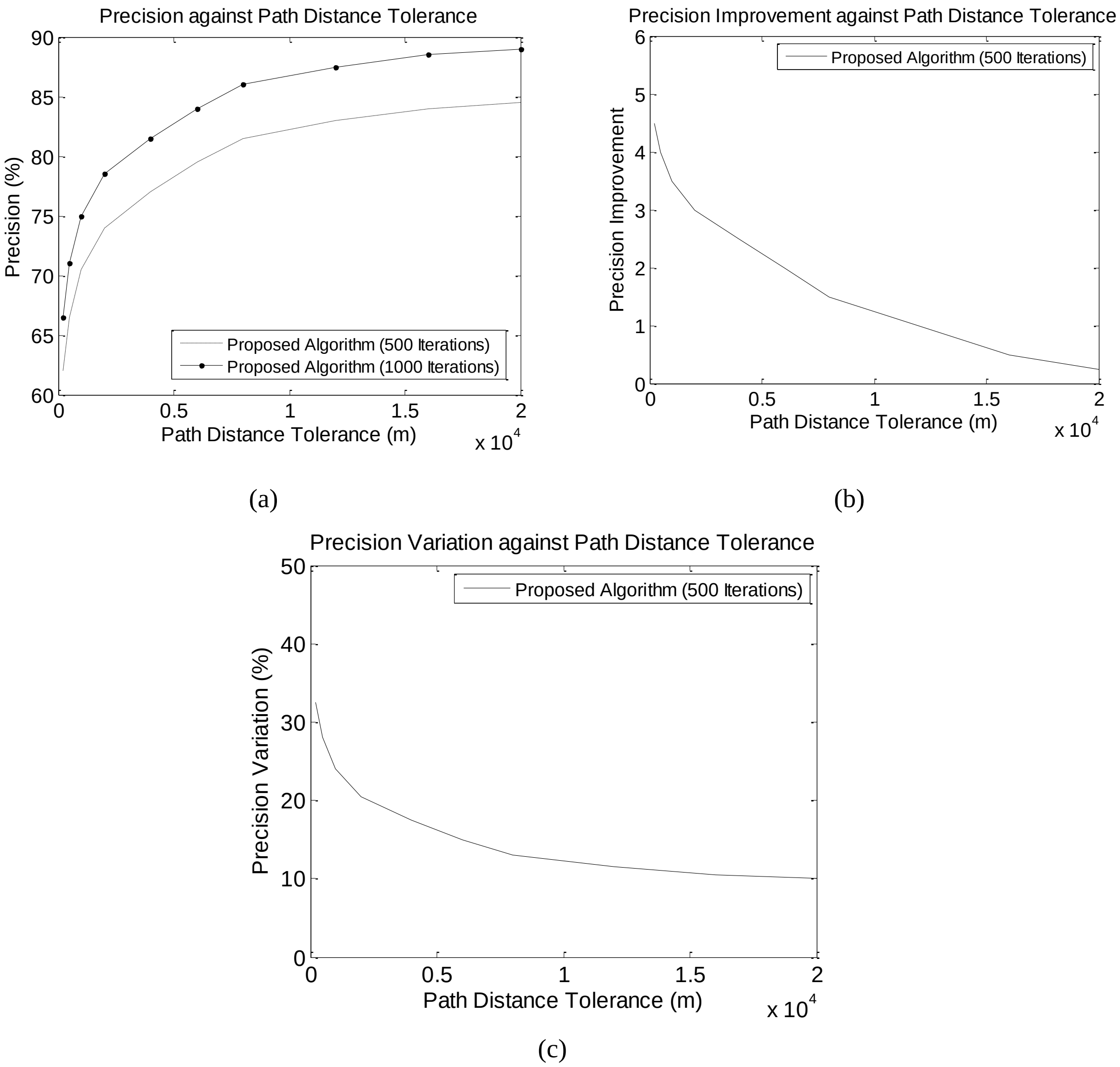
Another advantage of the proposed genetic-based optimization approach is its scalability. As we increase the number of iterations of the genetic algorithm, we are able to also improve the optimization performance, at the cost of computational complexity. However, in application scenarios where the computational cost is critical or in cases that the capabilities of the terminal devices are not so powerful, we are able to terminate the genetic optimization strategy earlier to save complexity at the cost of performance. Figure 46 presents experimental results as regards the effect of iteration cycles on the performance of the genetic algorithm. As is observed, the more iteration cycles we have, the better the performance of the route selection we achieve. We also observe that less iterations are required in order for the genetic optimization approach to retain a similar performance with the Dijkstra shortest path route selection algorithm (see Figure 46). It is clear that, as the number of iterations of the genetic algorithm increases, the precision, as defined through equation (21), increases, for a given recall value. It is also expected that as recall increases, precision decreases, since more objects are included in the selected route path, meaning that we have more complex routes.



(a) (b)

Figure 46: Precision-recall curve for different iterations of the genetic algorithm. (b) Comparison of the genetic algorithm with the shortest path algorithm.

A factor that affects the precision of the route selection algorithm is the route distance that a user can tolerate. In particular, if a user does not care about route distance, the genetic-based optimization can find a solution that mostly satisfy his/her needs by including in the itinerary most of the preferred objects. Instead, in case that the user considers routes with the shortest distance from a start to an end point, the optimization algorithms fails to include preferred objects in the itinerary reducing the precision performance of the system. The regulating factors γ_i of equation (10) adjust the importance of route (path) distance to the aggregate cost function. Figure 47(a) depicts the precision against the distance that a user can tolerate for a route. In this figure, the performance has been shown for two different iteration cycles of the genetic optimization algorithm. The results have been obtained as average values over 15 different experiments where user selects different start and end points and he/she sets a fixed route distance that can tolerate with. As is observed, as path distance tolerance increases, the precision also increases since more preferred objects are included in the itinerary. However, the performance rate decreases indicating saturation of the precision. This is verified in Figure 47(b) which presents the precision improvement versus the distance that the user can tolerate. Precision improvement is derived as the derivative of the precision curve of Figure 47(a) indicating at which amount we can increase the precision by increasing path distance. Another aspect that determines the performance of the proposed system is the variation of the precision achieved over the 15 different experiments. As is expected, different precision values are encountered across the 15 different executed experiments. Figure 47(c) shows the precision variation versus the path distance that a user may to tolerate. As is observed, higher variations are encountered for short path distances. This is expected since, for short distances, precisions highly depend on the selection of the start and end points. The opposite is valid for high path distance tolerance, where the effect of the selection of the start and end point is minimized.



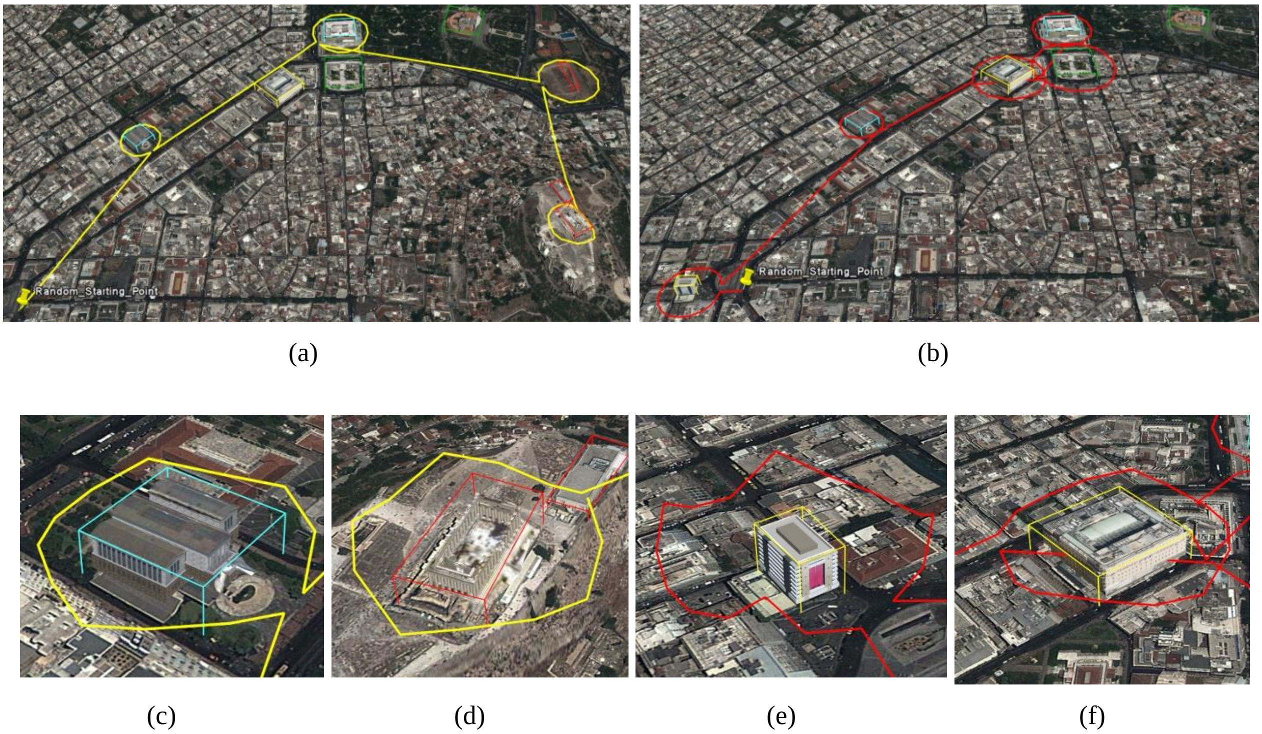
(a)

(b)

(c)

Figure 47: (a) Precision against path distance tolerance for different iterations of the genetic algorithm. (b) Precision improvement against path distance tolerance in case of a genetic optimization algorithm of 500 iterations. (c) Precision variation against path distance tolerance in case of a genetic optimization algorithm of 500 iterations.

Figure 48 presents a subjective (qualitative) evaluation of the proposed genetic optimization strategy. Simulation results have been performed using the scenario described in Paragraph 11.2. We recall that in this particular case, the user prefers archaeological places and buildings of neoclassical architectural style in his/her 3D navigation. In this figure, we also present the route provided by the shortest path Dijkstra's algorithm for comparison purposes. In particular, the shortest path algorithm yields the shortest navigation route that connects SP and FP points. On the contrary, the proposed GA approach includes in the itinerary objects that are most preferred by the user at a cost of increasing the overall route distance. However, the user is tolerant to such a distance increase. Figure 48(c,d) zooms in two of the preferred objects of the itinerary extracted by the genetic algorithm, while Figure 48(e,f) shows two of the retrieved objects in case of the shortest path algorithm. As is observed, in the first case, the National Gallery/Library of Athens and the Acropolis archaeological site have been selected in the itinerary, which are consistent with user's preferences. Instead, the buildings, retrieved using the shortest path algorithm, do not satisfy the actual user's preferences.



(a) (b)

(c) (d) (e) (f)

Figure 48: (a) Generated path using the proposed GA approach, (b) Generated path using Dijkstra shortest path algorithm. (c,d) Zoom in two preferred 3D objects of the itinerary retrieved by the proposed genetic algorithm approach. (e,f) Zoom in two preferred 3D objects of the itinerary retrieved by the shortest path distance algorithm.

All the aforementioned figures demonstrate the outperformance of the proposed scheme in terms of precision-recall against the compared ones. However, computational complexity is still an issue. The proposed genetic-based scheme requires higher computational cost than the other methodologies, especially in cases that high iteration cycles are required. Nevertheless, this drawback can be also seen as an alternative advantage due to the scalable nature of the genetic scheme. In other words, we can stop further executing the genetic algorithm when the computational cost exceeds a maximum acceptable limit as defined by our application case scenario. Instead, the aforementioned three optimization strategies lack of scalability; they always require the same computational time. Table 11 shows the computational complexity of the proposed genetic algorithm at different iterations compared with the aforementioned conventional route planning algorithms, i.e., the Dijkstra shortest path method, the nearest neighbor and the linear programming policy. The complexity is depicted in Table 11 as a normalized ratio with respect to Dijkstra which provides the minimum cost. In this table, we also depict the precision achieved at the given cost assuming a recall of 49%. As is observed, the proposed genetic algorithm provides almost similar performance for the same computational cost.

Algorithm	Precision	Cost
Proposed method at 500 iteration cycles	49.7	111%
Proposed method at 1,500 iteration cycles	59.8	332%
Dijkstra	48.2	100%
Nearest Neighbor	44.4	104%
Linear Programming	41.0	146%

Table 11: Performance and computational complexity of the proposed genetic based algorithm along with other optimization strategies.

11.4. Best View Estimation – Micro-Path Layer

In our implementation, each projected view of the 3D model is calculated on discrete points around the spherical coordinate system (see Figure 31 for more details). Then, for the given set of selected discrete view points, the ones that maximize the overall personalized entropy metric for all the 5 faces are selected as the most appropriate for the user. In order to understand the proposed algorithm, we discuss in the following, a use case scenario, where the user selects the Greek Parliament, as an interest object to navigate around. Figure 17(a) presents the best projected view obtained by maximizing the number of projected pixels of the object faces over the total number of the object pixels, i.e., maximization of the entropy metric of equation (16). Instead, in our approach, the best view is obtained by incorporating additional criteria that capture user’s preferences, resulting in a maximization of the personalized entropy metric [see equation (17)]. The best three top ranked views are depicted in Figure 49(b,c,d). The Greek parliament is represented by a bounding box. Each of the four surrounded faces have almost the same metadata (“*Made from Marble*”, “*Historical Education*”, “*Neoclassical*”), but the upper face of the bounding box has “*Made from Cement*” instead of “*Made from Marble*”. Since the user is more interested with marbles instead of cement, the proposed personalized best view selection algorithm estimates views that are more suitable to user’s preferences instead of the conventional methods where only geometric constraints are taken into consideration.

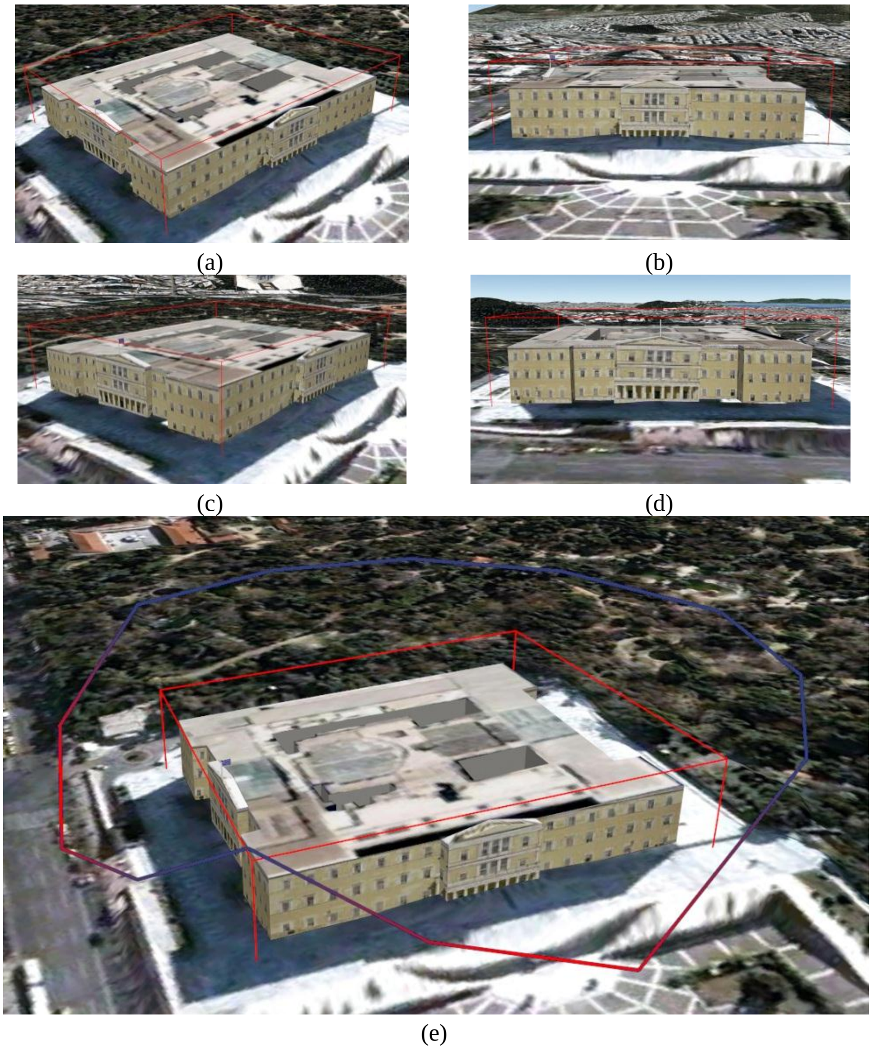
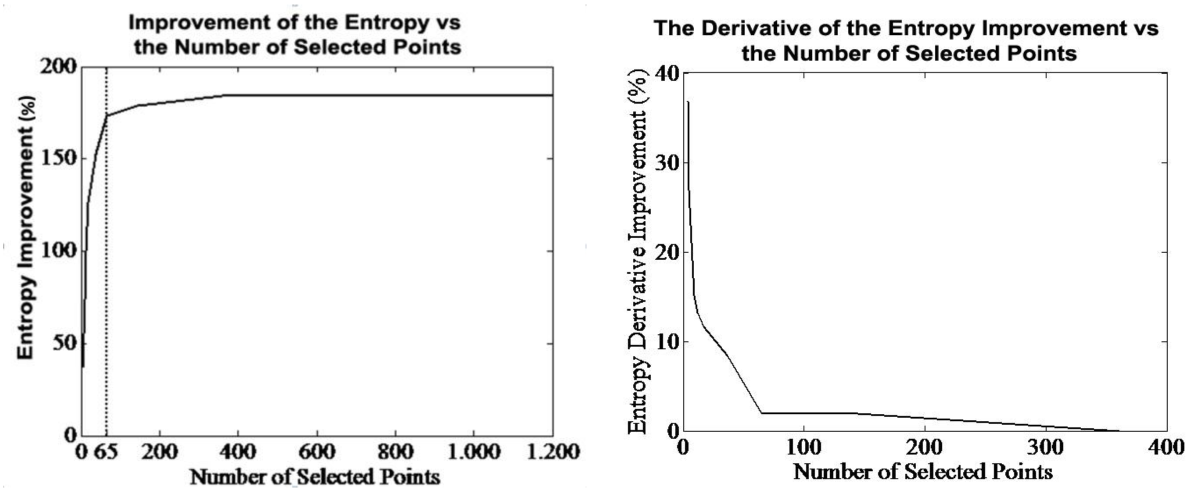


Figure 49: (a) Best view selected using only geometrical properties. (b,c,d) Best view selected using geometrical and human factors. (e) Indicating camera and speed trajectory of the selected viewpoints

In Figure 49(e), we have illustrated the 3D trajectory around the Greek parliament, which is useful to support a personalized 3D navigation experience. The trajectory is formed by merging several top ranked views as obtained either by the proposed best view selection algorithm or the conventional geometric approach. Blue regions indicate views that maximize the classical entropy criterion, that is, maximization of the projected area [see equation (16)]. Instead, red regions correspond to views that maximize the personalized entropy, that is, geometrical and human factors [see equation (17)]. Similar results are also depicted in Figure 48(c,f) in which the personalized best views are also illustrated. We recall that Figure 48(c,d) depicts two preferred objects as they have been retrieved using the genetic scheme while Figure 48(e,f) two objects retrieved using the shortest path algorithm.

In the following, we evaluate the complexity of the best view selection algorithm. It is clear that the complexity is straightforwardly proportional to the number of examined discrete points on the semi-spherical bounding boxes. The more the examined points on the semi-sphere coordinate system are, the more the complexity is. It is also clear that as the complexity increases, the performance is also improved. This is verified in Figure 50(a), which presents the entropy improvement versus the number of selected points used. However, the ratio of the improvement is decreased, meaning that beyond a certain number of points a slight improvement is achieved though complexity continues to increase. This is depicted in Figure 50(b), where we plot the improvement ratio versus the computational complexity. These results have been obtained as the average over 50 experiments of different 3D objects. Based on these figures, we can approximate an average number of computations required to achieve a projected view of a 3D building which is *94.6*% close to the optimal solution. For example, in case that we use 65 different points on the semi-spherical coordinate system, we can have a very good approximation of the personalized entropy metric as is verified in Figure 50(a). It is also clear that we can relax approximation accuracy for buildings of low user's interest.



(a)

(b)

Figure 50: (a) Improvement of the entropy versus the number of selected points. (b) The derivative of the entropy improvement versus the number of selected points.

12. Conclusions & Future work

12.1. General Conclusions

3D geo-informatics has entered the digital age enabling new potential applications in the field of virtual tourism, leisure, entertainment and cultural heritage. It is argued that 3D virtual tools provide the natural way of navigation since they can explain a story or even show a part of the world. Personalization is a key aspect for an efficient 3D navigation, since routes are selected not only using geometrical criteria, but mostly adhering to user's preferences. Personalization is often modelled through a set of weights that regulate the degree of importance of the scene metadata on the route selection process. In existing approaches, metadata weights are left to the user's side, the latter being responsible for defining his/her profile. This, however, implies that the user is aware of technical details as regards system implementation and, in addition, that there is a direct relation among user's preferences and the metadata of the scene, which is not the usual case.

In this research, we address the aforementioned difficulties by introducing an online learning strategy in order to automatically adjust metadata weights according to information fed back to the system about the relevance of user's preference judgments given in a form of pairwise comparisons. This way, we transfer the complexity in defining user's preferences from the user to the system side. However, the main bottleneck of such approach is that a large number of pairwise comparisons is required to converge to a set of reliable and stable metadata weights that properly model user's preferences, especially in case that large number of objects are encountered in the scene. For this reason, in this work, we have introduced a weight rectification strategy that improves the estimation of metadata weights by exploiting knowledge and metadata interrelationships described through an ontology. Metadata weights that define user's preferences are involved in a multi-criteria optimization process with the aim to calculate the user's most preferred route. Multi-criteria route optimization is a combinatorial problem and thus a genetic algorithm is used to estimate an approximate solution. Another important aspect in a personalized 3D navigation is to determine the way that the preferred 3D objects should be projected onto 2D screens. In this work, in contrast to previous approaches, we introduce a personalized entropy metric to estimate the best projected 2D views taking into account not only the geometric complexity of the projection but also the degree of importance of the projected faces as defined through the metadata weights.

All these components have been integrated in a single platform and comparisons have been conducted to demonstrate the outperformance of the proposed route guidance system. In particular, as far as the online learning strategy is concerned, we have compared it with a relevance feedback scheme that estimates metadata weights inversely proportional to the standard deviation of the feature elements over all relevant selected objects [53] and with the Analytic Hierarchy Process (AHP) algorithm [54]. We observe that the proposed learning scheme yields, on average, a higher precision value of 8.1% (or equivalently an improvement of 13.93% of the proposed scheme in relation with the method of [53]) over all recall values than the second best compared learning strategy which is the method of [53]. In addition, we have performed simulations to illustrate the effect of the weight rectification strategy through the use of the ontology. We conclude that, in this case, we can reduce the number of samples (pairwise comparisons) required of 76% to achieve the same precision.

Regarding route optimization, we have compared our genetic algorithm (GA) with three methodologies, the Dijkstra shortest path method, the nearest neighbor algorithm and a linear programming optimization policy. In all cases, the proposed GA method yields, on average, higher precision value of 4.76% (or equivalently an improvement of 8.75% of the proposed scheme in relation with Dijkstra) over all recall values than the second best one (i.e., the Dijkstra shortest path), however, at a cost of an increase of the overall path distance. In particular, in case that a user does not care about route distance, the genetic optimization can find a solution that mostly satisfies his/her needs by including in the itinerary most of the preferred objects. Instead, in case that the user considers routes with the shortest distance from a start to an end point, the optimization algorithms fails to include preferred objects in the itinerary and the performance of GA is similar to the other compared methods. The limitation of the GA is that it requires high computational cost compared to other methodologies, especially in cases where high numbers of iteration cycles are included. However, this limitation can be also seen as a potential advantage of the GA since it allows a scalable implementation; we can improve the optimization performance at the cost of demanding more computational effort, i.e., more iteration cycles of the GA. At the end, our simulations have demonstrated that the proposed GA optimization algorithm provides better performance than the compared techniques for the same computational cost. Finally, experimental results as regards the best view selection algorithm indicate that we can increase the degree of personalization for 3D navigation. In particular, we are able to estimate personalized 3D trajectories around the preferred objects by merging projected 2D views which have been estimated not only using geometric constraints but also the importance of the projected faces as defined through the metadata weights.

12.2. Future work

Of particular interest for future work is to improve the route selection algorithm by the use of modern optimization techniques, such as particle swarm optimization, ant colony or the artificial bee colony algorithm, which are promising in solving combinatorial problems [16],[31]. Another aspect for future research is to incorporate 3D geometric constraints in the construction of 3D trajectories which increases the complexity of the optimization.

Furthermore, there are plenty of parameters that can be integrated into the macro-path algorithm in order to estimate a more realistic path in case of simulation of a human’s walk or a car. The street layer of the roads might be added. Information about the street junctions and speed limit can be valuable on the path planning. In addition, emotional maps or crime rate data will lead to a more reliable and realistic physical path estimation. In addition, further semantic metadata can be added, in a way to provide the ability to the user to execute queries on the selected path. Finally, it will be useful if the generated virtual tour was subject to user evaluation, in order to improve the estimation of the user’s profile.

Applications for a city 3D modelling system can be found in a variety of fields. Virtual tours can be desirable in places that someone either hasn’t visited or can’t visit. In the field of tourism, a 3D city helps to develop and present a region to the rest of the world. Furthermore, simulations can be based on 3D city models, with respect to physical phenomena, evacuating processes, etc.

13. References

[1] F. Allaire, M. Tabouchi, G. Labonte and G. Fusina “FPGA Implementation of Genetic Algorithm for UAV Real-Time Path Planning” Journal of Intelligent & Robotic Systems Vol. 54, Numbers 1-3, pp.495-510, DOI: 10.1007/s10846-008-9276-8, 2008.

[2] D. L. Applegate, R. M. Bixby, V. Chvátal, and W. J. Cook, “The Traveling Salesman Problem: A computational study,” Princpton Series in Applied Mathematics, 2006.

[3] G. Bardis, “Intelligent Personalization in a Scene Modeling Environment, Miaoulis G., Plemenos D. (Eds.) Intelligent Scene Modelling Information Systems,” Studies in Computational Intelligence, Vol.181, pp. 89-119, ISBN 978-3-540-92901-7, Springer, 2009.

[4] G. Bardis, G. Miaoulis, D. Plemenos, “User Profiling from Imbalanced Data in a Declarative Scene Modeling Environment, Plemenos D., Miaoulis G. (Eds.) Artificial Intelligence Techniques for Computer Graphics,” Series: Studies in Computational Intelligence, Vol. 159, pp. 123-140, ISBN 978-3-540-85127-1, Springer, 2008.

[5] P. F. Bonnefoi and D. Plemenos “Constraint satisfaction techniques for declarative scene modeling by hierarchical decomposition”, 31A'2000 International Conference, 2002.

[6] A. Botea, M. Müller, and J. Schaeffer "Near optimal hierarchical path-finding," Journal of Game Development, Vol. 1, pp. 7-28, 2004.

[7] A. Borrmann and E. Rank “Specification and implementation of directional operators in a 3D spatial query language for building information models,” Advanced Engineering Informatics Vol. 23, pp.32–44 2009.

[8] A. Borrmann and E. Rank “Topological analysis of 3D building models using a spatial query language,” Advanced Engineering Informatics, Vol. 23(4), pp. 370–385, 2009.

[9] A. Bosch, A. Zisserman and X. Muoz, “Scene classification via plsa,“ European Conference on Computer Vision, pp.517-530, 2006.

[10] M. Chacon, D. Alma and R. Rodriguez “Image Complexity Measure: a Human Criterion Free Approach,” NAFIPS 2005, pp. 241-246, 2005.

[11] M. Chacon, D. Alma and R. Sandoval “A Fuzzy Approach on Image Complexity Measure,” pp.268-284, 2007.

[12] C. Chalvantzis, and M. Virvou, “Fuzzy logic decisions and web services for a personalized geographical information system,” Studies in Computational Intelligence 142, pp. 439-450, 2008.

[13] S. Chatzis, A. Doulamis, and T. Varvarigou, “A Content-based Image Retrieval Scheme allowing for Robust Automatic Personalization,” ACM International Conference on Image and Video Retrieval, pp. 1-6, July 2007

[14] C. Chen “Information visualization and virtual environments. London,” Springer-Verlag, 1999.

[15] W. Cohen, R. Schapire, Y. Singer, “Learning to Order Things,” Journal of Artificial Intelligence Research, Vol. 10, pp. 243-270, 1999.

[16] A. Colorni, M. Dorigo et V. Maniezzo, Distributed Optimization by Ant Colonies, actes de la première conférence européenne sur la vie artificielle, Paris, France, Elsevier Publishing, 134-142, 1991.

[17] M. Cristani, A. Perina, U. Castellani, and V. Murino “Content Visualization and Management of Geo-located Image Databases,” In CHI '08 extended abstracts on Human factors in computing systems, ACM, NY, USA, pp. 2823-2828, 2008.

[18] D. Demyen and M. Buro. "Efficient triangulation-based pathfinding",18th International Conference on Innovative Applications of Artificial Intelligence Conference (AAAI’06), pp. 942-947, ISBN 978-1-57735-281-5, 2006.

[19] E.W. Dijkstra, “A note on two problems in connexion with graph,” Numer. Math. 1 Vol. 1, pp. 269–271, 1959.

[20] N. Doulamis, E. Chronis, G. Miaoulis and D. Plemenos "Personalized View Selection of 3D Molecular Proteins," Studies in Computational Intelligence, Vol. 321, pp. 211-227, Springer Press, 2010.

[21] A. Doulamis, "Event-driven video adaptation: A powerful tool for industrial video supervision" Multimedia Tools and Applications doi:10.1007/s11042-012-0992-5 (to appear)

[22] W. J. Drummond and S.P.F. French "The future of GIS in planning: Converging technologies and diverging interests,". Journal of the American Planning Association, Vol. 74(2), pp.161–174, 2008.

[23] Y. Freund and R. Schapire R."A decision-theoretic generalization of on-line learning and an application to boosting", Journal of Computer and System Sciences, Vol. 55, No. 1, pp. 119-139, 1997.

[24] G. Gutin, A. Yeo and A. Zverovich, Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the TSP. Discrete Applied Mathematics 117, pp. 81-86, 2002.[53]

[25] P.E. Hart N.J. Nilsson B. Raphael "A formal basis for the heuristic determination of minimum cost paths," IEEE T. Syst. Man Cyb. Vol. 4, pp. 100–107, 1968.

[26] J. H. Holland "Adaptation in natural and artificial system, Ann Arbor," The University of Michigan Press, 1975.

[27] B. Huang "Web-based dynamic and interactive environmental visualization," Computers, Environment and Urban Systems," Vol 27(6), pp623–636, 2003.

[28] S. Gauch, J. Cha , A. Pretschne r"Ontology-based person-alized search and browsing", Journal of Web Intelligence and Agent Systems, Vol. 1, pp. 219–234, 2003.

[29] N. Guarino "Formal ontology in information systems", In proceedings of the 1st international conference on formal ontologies in information systems FOIS, Trento, Italy: IOS Press, pp. 3-15, 1998.

[30] T.M. Jones "Google's Geospatial Organizing Principle," IEEE Computer Graphics, Vol 27 (4), pp8-13, 2004.

[31] J. Kennedy, R. Eberhart, "Particle Swarm Optimization". Proceedings of IEEE International Conference on Neural Networks, Vol. 5, pp. 1942–1948, 1995..

[32] A. Kobsa, J. Koenemann and W. Pohl "Personalized hypermedia presentation techniques for improving customer relationships," The Knowledge Engineering Review. Vol. 16, No. 2, pp.. 111–155, 2001.

[33] A. Koutsoudis, F. Arnaoutoglou, Ch. Chamzas "On 3D reconstruction of the old city of Xanthi. A minimum budget approach to virtual touring based on photogrammetry," Journal of Cultural Heritage 8, p26-31. 2007.

[34] N. Kraljic, "Interactive Video Virtual Tours", 12th Central European Seminar on Computer Graphics, Split, Croatia 2008.

[35] K. Laakso, O. Gjesdal and Jan Rasmus Sulebak, "Tourist information and navigation support by using 3D maps displayed on mobile devices", Workshop "HCI in mobile Guides", Udine (Italy), 8 September 2003

[36] H. Lan, D. C. Martin, C. Zhou, C.H. Lim "Rockfall hazard analysis using LiDAR and spatial modeling," Geomorphology, Vol. 118(1-2), pp. 213–223, 2010.

[37] M. Larive, V. Gaildrat "Wall Grammar for Building Generation," Graphite '06, Kuala lumpur, Malaysia, September, 2006.

[38] N. Littlestone and M. Warmuth "The weighted majority algorithm. Information and Computation," pp. 212-26, 1994.

[39] M. Lucas and E. Desmontils "Declarative modelers". Revue Intern. de CGAo et d' Infogr., pp. 559-585, 2007.

[40] J. Mackinlay, S. Card and G. Robertson "Rapid Controlled Movement through a Virtual 3D Workspace", Computer Graphics, Vol. 24, Number 4, pp. 171-176, August 1990.

[41] M. Mekni and B. Moulin, "Hierarchical Path Planning for Multi-agent Systems Situated in Informed Virtual Geographic Environments," Second International Conference on Information, Process, and Knowledge Management, Saint Maarten, pp. 48-55, ISBN 978-1-4244-5688-8, 2010.

[42] M. Mitchell “An Introduction to Genetic Algorithms (Complex Adaptive Systems*)*,” MIT press, Cambridge 1996.

[43] S. Nadi, M.R. Delavar, “Multi-citeria, personalized route planning using quantifier-guided ordered weighted averaging operators,” International journal of Applied Earth Observation and Geoinformation, pp.322-3358, 2011.

[44] S.A Niaraki. and Kim K., “Ontology based personalized route planning system using a multi-criteria deision making approach,” Expert Systems with Applications, Science Direct, Vol Experts systems with Applications 36(2), pp. 2250-2259, 2009.

[45] J. Prkio and A. Hyvarinen “Modelling Image Complexity by Independent Component Analysis, with Application to Content-Based Image Retrieval,” ICANN 2009, part II, Springer, pp.704-714, 2009.

[46] I. Rechenberg “Evolutionsstrategie – Optimierung technischer Systeme nach Prinzipien der biologischen Evolution,” PhD thesis, 1971.

[47] A. Reuter and A. Zipf “GIScience – where next”, In S. Fotheringham & J. P. Wilson (Eds.), Handbook to GIS. Blackwell, 2005.

[48] D. Reitter and C. Lebiere, “A cognitive model of spatial path-planning,” Computational & Mathematical Organization Theory, Vol. 16, No 3, pp. 220-245, 2010.

[49] D. Riecken “Personalized views of personalization” Communications of the ACM, 43(8), pp. 27-28, 2000.

[50] J. Rocchio, Relevance Feedback in Information Retrieval: The SMART Retrieval System: Experiments in Automatic Document Processing. Prentice Hall, 1971.

[51] S. Rogers, P. Langley “Personalized driving route recommendations. In: Proceedings of the AAAI Workshop on Recommender Systems,” July 26 , Madison, WI, USA, pp. 96–100, 1998.

[52] M. Ruger, B. Preim and A. Ritter, “Zoom Navigation Exploring Large Information and Application Spaces,” 1996.

[53] Y. Rui, T.S. Huang, M. Ortega, S. Mehrotra, “Relevance feedback: A power tool for interactive content-based image retrieval” IEEE Trans. on Circuits and Systems for Video Technology, Vol. 8, No. 5, pp. 644-655. 2008.

[54] T. L. Saaty “Decision Making for Leaders: The Analytic Hierarchy Process for Decisions in a Complex World,” RWS Publications ed. Pittsburgh PA, 1990, (ii) 2000.

[55] T. L. Saaty “The Analytic Network Process,” RWS Publications, 2006.

[56] G. Salton and M. J. McGill, “Introduction to Modern Information Retrieval,” New York: McGraw-Hill Book Company, pp. ISBN 0070544840, 1982.

[57] C. E. Shannon “A Mathematical Theory of Communication,” Bell System Technical Journal, Vol. 27, pp. 379–423, 623–656, 1948.

[58] D. Sokolov and D. Plemenos “Viewpoint quality and scene understanding,” VAST, Eurographics Symposium Proceedings, Pisa (Italy), pp. 67-73, 2005.

[59] D. Sokolov and D. Plemenos “High level methods for scene exploration,” Journal of Virtual Reality and Breadcasting, Vol 3(12), 2006.

[60] D. Sokolov and D. Plemenos “Intelligent scene display and exploration,” International Conference Graphicon 2006, Novosibirsk Akademgorodok, Russia, 2006.

[61] D. Sokolov and D. Plemenos “Virtual world explorations by using topological and semantic knowledge” The Visual Computer Journal, Vol.23, pp.173-185, 2007.

[62] W. Sriphaisal. and P.K. Arun “A Comparative Assessment of GIServices Architecture,” In Proceeding of the International Conference Map Asia, Bangkok, Thailand, 29 Aug - 1 Sep 2006.

[63] D. Tab, G. Robertson and M.Gzewinski, “Exploring 3D Navigation: Combining Speed-coupled Flying with Orbiting,” Proc. of the SIGCHI conference on Human factors in computing systems, Vol. 3, pp. 418-425, 2001.

[64] L.L. Thurstone "A Low of Comparative Judgment," Psychological Review Vol. 34(4), pp 273-286, 1927.

[65] P. P Vazquez, M. Feixas, M. Sbert and A. Llobet "Realtime automatic selection of good molecular views," Computers & Graphics Vol. 30, pp. 98–110, 2006.

[66] P. P Vazquez "On the selection of good views and its application to computer graphics," Ph.D thesis, Technical University of Catalonia, Barcelona, Spain, 2003.

[67] C. Wolf "Content based Image Retrieval using Interest Points and Texture Features," Technical Report, PRIP-TR-61, 2000.

[68] H. Wu, Z. He and J Gong "A virtual globe-based 3D visualization and interactive framework for public participation in urban planning processes," Computers, Environment and Urban Systems Vol. 34(4) pp. 291–298, 2010.

[69] C. Yiakoumettis, G. Bardis, G. Miaoulis, D. Plemenos, and D. Ghazanfarpour "A GIS platform for automatic navigation into Georeferenced Scenes using GIS Scene Explorer (GIS-SE)," Intelligent Computer Graphics 2010, springer, Vol. 321, pp. 105-222, 2010.

[70] C. Yiakoumettis, G. Bardis, G. Miaoulis, D. Plemenos, and D. Ghazanfarpour "Collaborative Framework for Virtual Globe Based 3D City Modeling," Intelligent Computer Graphics 2010, springer, Vol. 321, pp. 165-184, 2010.

[71] C. Yiakoumettis, N. Doulamis, G. Miaoulis, and D. Ghazanfarpour "Active Learning of User's Preferences Estimation Towards a Personalized 3D Navigation of Geo-referenced Scenes," Research report No 20 of the research lab LASKII, 2012.

[72] J. Zhang , J. Gong , H. Lin, G. Wang, J. Huang, J. Zhu, B. Xu, J. Teng "Design and development of Distributed Virtual Geographic Enviroment system based on web services," Information Sciences Vol. 177(19), pp. 3968-3980, 2007.

[73] A. Zipf, M. Jost "Implementing adaptive mobile GI services based on ontologies: examples from pedestrian navigation support," Comput. Environ. Urban Syst. 30 (6), 784–798, 2006.