ΠΛΗΡΟΦΟΡΙΑΚΟ ΣΥΣΤΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΦΩΤΟΓΡΑΦΙΩΝ ΚΑΙ ΧΩΡΙΚΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΜΕ ΧΡΗΣΗ ΑΙΣΘΗΤΗΡΩΝ, ΟΝΤΟΛΟΓΙΩΝ ΚΑΙ ΣΗΜΑΣΙΟΛΟΓΙΚΩΝ ΧΑΡΤΩΝ

Υπό

Μιχαήλ Φουκαράκης

Πολυτεχνείο Κρήτης

Τμήμα Ηλεκτρονικών Μηχανικών & Μηχανικών Υπολογιστών

Μια εργασία που υπεβλήθη για τη μερική εκπλήρωση των απαιτήσεων απόκτησης Μεταπτυχιακού Διπλώματος Ειδίκευσης

Εργαστήριο Διανεμημένων Πληροφοριακών Συστημάτων και Εφαρμογών Πολυμέσων

Χανιά, 2009

ΠΕΡΙΛΗΨΗ

Με τις εξελίξεις στον τομέα της ψηφιακής φωτογραφίας, η δημιουργία, η αποθήκευση και ο διαμοιρασμός φωτογραφικού περιεχομένου έχουν απλοποιηθεί και η χρήση ψηφιακών συσκευών για αυτό το σκοπό είναι ευρύτατα διαδεδομένη. Αυτό οδήγησε στην εντυπωσιακή αύξηση του μεγέθους των προσωπικών συλλογών των χρηστών, αυξάνοντας και την ανάγκη για καλύτερη οργάνωση, εμπλουτισμό, ταξινόμηση και αναζήτηση των φωτογραφιών. Επιπλέον, οι τεχνολογικές εξελίξεις στην περιοχή των συσκευών που παράγουν ψηφιακές εικόνες, έχουν επιτρέψει σε αυτές να επικοινωνούν με έναν αυξανόμενο αριθμό από εξωτερικούς αισθητήρες και τους έχουν δώσει παραπάνω δυνατότητες για να συλλαμβάνουν και να κατανοούν το περιβάλλον φωτογράφισης. Αυτό που λείπει είναι μια προσπάθεια εκμετάλλευσης αυτών των νέων δυνατοτήτων και κατασκευής ενός συστήματος για την αποτελεσματική διαχείριση του συνεχώς αυξανόμενου ψηφιακού περιεχομένου.

Αυτή η διατριβή προτείνει ένα πλαίσιο για σημασιολογική επεξεργασία χωρικής πληροφορίας. Το πλαίσιο εκμεταλλεύεται τις δυνατότητες των σύγχρονων φωτογραφικών μηχανών στην καταγραφή χωροχρονικών παραμέτρων με χρήση εξελιγμένων αισθητήρων, πληροφοριών από ειδικά εμπλουτισμένους σημασιολογικούς χάρτες και βιομηχανικών προτύπων. Ο στόχος είναι η αποτελεσματική διαχείριση και συσχετισμός της πληροφορίας και των σημασιολογικών αντικειμένων που βρίσκονται στους σημασιολογικούς χάρτες και στις φωτογραφίες. Επιπροσθέτως, για τον αυτόματο εμπλουτισμό των φωτογραφιών, το πλαίσιο χρησιμοποιεί αλγορίθμους για επεξεργασία εικόνας και άλλες εργασίες.

Το πλαίσιο είναι αρκετά γενικό ώστε να επιτρέπει διαφορετικές υλοποιήσεις και ένα σύστημα για την επίδειξη των ιδεών του υλοποιήθηκε για τους σκοπούς της παρούσας διατριβής. Η υλοποίηση του συστήματος παρέχει μια βάση δεδομένων εικόνων που επιτρέπει στους χρήστες να αποθηκεύουν και να βλέπουν τις φωτογραφίες τους. Εκτός από τις φωτογραφίες, οι χρήστες μπορούν να δουν εξατομικευμένους σημασιολογικούς χάρτες εμπλουτισμένους με σημασιολογικά αντικείμενα που περιγράφονται χρησιμοποιώντας οντολογίες. Οι χάρτες αυτοί παρέχονται από έναν απομακρυσμένο εξυπηρετητή. Οι φωτογραφίες που περιλαμβάνουν πληροφορίες θέσης η πληροφορίες θέσης και κατεύθυνσης μπορούν να απεικονιστούν πάνω στους χάρτες και να συσχετιστούν με σημασιολογικά αντικείμενα. Τέλος, αρκετοί αλγόριθμοι που επιτρέπουν την αλληλεπιδραστική εξερεύνηση των περιεχομένων των φωτογραφιών έχουν υλοποιηθεί.

Πειράματα με μια σύγχρονη ψηφιακή φωτογραφική μηχανή και εξωτερικούς αισθητήρες εκτελέστηκαν για να δοκιμαστεί η χρησιμότητα του συστήματος στον τομέα του τουρισμού και για να δείξουν την αποτελεσματικότητα των υλοποιημένων αλγορίθμων.

INFORMATIONAL SYSTEM FOR MANAGING PHOTOS AND SPATIAL INFORMATION USING SENSORS, ONTOLOGIES AND SEMANTIC MAPS

By

Michail Foukarakis

Technical University of Crete

Electronic & Computer Engineering Department

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Electronic and Computer Engineering

Laboratory of Distributed Multimedia Information Systems and Applications

Chania, 2009

ABSTRACT

With the current developments in the digital photography domain, the task of capturing, storing and sharing photographic content has been simplified and the use of digital devices for this purpose is now widespread. This has led to the outstanding increase of the size of user personal photo collections, increasing the need to better organize, annotate, index and browse the photos. Moreover, the technological advancements related to the digital image capturing devices have allowed them to interoperate with an increasing number of internal or external sensors, giving them additional capabilities for capturing and understanding the image taking context. What is missing is an attempt to take advantage of these new capabilities and build a system for efficient management of the ever-increasing media content.

This thesis proposes a framework for semantic spatial information processing. The framework exploits the modern digital cameras' potential for capturing contextual parameters through the use of sophisticated sensor devices, information found in specially annotated semantic maps and industrial standards. The objective is to effectively manage and associate the information and semantic objects contained in both the semantic maps and the pictures. In addition, the framework employs the use of image processing and other algorithms to enable the automatic annotation of the photos.

The framework is generic enough to allow different system implementations and a system for demonstrating its ideas was created for the purposes of this thesis. The system implementation provides a picture database that allows users to store and view their photos. Along with the photos, the users can view personalized semantic maps, annotated with semantic objects described using ontologies. These maps are supplied from a remote server. Photos with position only information or both position and direction information can be visualized on top of the maps and be associated with semantic objects. Finally, several algorithms that allow interactive exploration of the picture contents have been implemented.

Experimentation with a modern digital camera and external sensors has been conducted to test the usefulness of the system in the tourism domain and to show the effectiveness of the implemented algorithms.

DEDICATION

In loving memory of my father.

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis.

I authorize the Technical University of Crete to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the Technical University of Crete to reproduce this thesis by photocopying or by other means, in total or part, at the request of other institutions or individuals for the purpose of scholarly research.

ACKNOWLEDGEMENTS

I would like to express my gratitude to my supervisor Prof. Stavros Christodoulakis for his dedication and for making me realize the importance of doing research and pursuing your goals and for teaching me useful things in research and life in general.

My thanks go to all the people and staff of the MUSIC laboratory, Nektarios Moumoutzis, Nektarios Gioldasis, Giorgos Anestis, Fotis Kazasis, Nikos Pappas, Petros Kapoulas and the graduate students. Many thanks to Stratos Georgoulakis for inviting me to the laboratory, being a good friend and helping me start the foundation of my thesis.

I would also like to thank my friends who always wondered why I left my home city and lost me for several years. They will have to put up with me again soon…

Finally, my eternal gratitude goes to my parents who supported me for as long as they could and believed in me even when I did not. I will never forget your dedication and support.

PUBLICATIONS

1. Stavros Christodoulakis, Michalis Foukarakis, Lemonia Ragia, Hiroaki Uchiyama, Takuya Imai: *Semantic maps and mobile context capturing for picture content visualization and management of picture databases*. MUM 2008: 130-136
2. Stavros Christodoulakis, Michalis Foukarakis, Lemonia Ragia: *Spatial Information Retrieval from Images Using Ontologies and Semantic Maps*. WSKS 2008: 549-556 (Best Paper Award)
3. Stavros Christodoulakis, Michalis Foukarakis, Lemonia Ragia: *Personalised spatial knowledge management for pictures using ontologies and semantic maps*. In International Journal of Digital Culture and Electronic Tourism, Volume 1, Issue 4, 2009.
4. Stavros Christodoulakis, Michalis Foukarakis, Lemonia Ragia, Hiroaki Uchiyama, Takuya Imai: *Mobile Picture Context Capturing for Semantic Pictorial Database Content Access, Browsing, and Interaction*. Submitted for review for IEEE Multimedia Special Issue on Mobile and Ubiquitous Multimedia, April-June 2010.

Table of Contents

List of Figures

List of Tables

1. Introduction

For the past years, cameras have been used as a means to acquire and store images. Unfortunately, all context information associated with the image taking used to be lost and the image files were just treated as bit streams. The ever-increasing size of users' personal photo collections reveals the need to efficiently share, search and view the stored images.

To organize their collections, users would create different folders in their personal computers to differentiate and provide more information about their photos. For example, naming a folder "Trip to Crete 2008" implies that the photos stored inside that folder are about a trip relevant to the user. Meaningful image file names can also help in order to classify the contents of each photo. Finally, using specific software, the images can manually be annotated in more detail. However, all these methods require substantial effort from the user who might not have the interest, the time, the knowledge or memory of the image contents to annotate it properly. Recent advances in the photography world are able to assist in reducing the size of the mentioned problems.

The capabilities of modern digital cameras are no longer restricted to just capturing digital media. They are equipped with many features, such as the ability to interoperate with an increasing number of sensors and integrate the additional information within the digital media. Some of these sensors can be integrated within the devices themselves, allowing for easier context capturing. The use of international standards can also facilitate this procedure and ensure interoperability. The goal is to achieve semi-automatic or automatic (if possible) media (image/video) annotation. This work focuses on digital images[1] only.

Since cameras are now equipped with wireless technology (Bluetooth™, Wi-Fi), the possibility of communicating with different sources of information such as GIS systems and applications or other devices (PDAs etc.) is an attractive option for enabling new applications and addressing the problems stated earlier. To utilize the plethora of information available for image enrichment and annotation, appropriate systems should be built that can efficiently manage and associate the information with the images, taking also into account the importance of personalization. Of significant importance are systems that associate images with geographic information and maps, allowing for greater visualization and indexing.

[1] To represent the concept of still images, the words 'picture' and 'photo' will also be used extensively throughout this thesis.

Many websites such as Flickr [17] enable photo sharing, labeling and geo-coding (associating a photo with position information). Then the users can search for images of interest and view them on top of maps. However, automatic image annotation and association with map features has not been attempted in a consistent and user-centric manner.

1.1. A Framework for Semantic Spatial Information Processing

This thesis proposes a framework for semantic spatial information processing, abbreviated SPIM (SPatial Image Management). This framework is based on the following requirements (in no particular order):

- Semantic information processing
- Semantic interoperability with other applications in a service oriented infrastructure
- Use of industrial standards
- Use of community accepted ontologies
- Automatic image annotation
- Personalization
- Simplicity and automation in the information capturing

The objective is to create the infrastructure for integrated transparent management of semantic spatial multimedia information that includes maps, semantic objects in maps, pictures and semantic objects of the spatial environment captured in pictures. The framework takes into account multi-sensor digital camera capabilities and semantic spatial information encoded in semantic maps to automatically associate digital image contents with semantic spatial information and allow powerful functionality and visualizations. The framework's main content categories (analyzed in chapter 4) are:

- Contextual parameters
- Description of semantic entities of the real world
- Semantic maps
- Image content descriptions
- Events and persons

1.2. Implementing the ideas of the framework

For this thesis, a system implementing the main ideas of the framework has been developed. The system utilizes a modern digital camera integrated with a sensor capturing position and direction parameters and processes the information embedded in the pictures in order to associate image contents with semantic information located in semantic maps.

The system provides a simple interactive map interface and the ability to visualize objects of interest and photos on top of the map. The user can select and view information about semantic objects contained in pictures and also view detailed picture information located either in the image metadata or provided by the system's automatic annotation capability. Viewable information about a semantic object includes:

- Name
- Semantic type (for example "St. Nikolaos Church" can be of type Church)
- Domain (or ontology - for example "Knossos Ruins" belongs to the Archaeology domain)
- Description
- Map representation (as a geometric shape)
- List of photos depicting the semantic object
- Photo representation (colorful layer depicting the semantic object's shape superimposed on a photo)

Viewable information about a picture includes:

- The image itself
- Metadata (camera model/make, date, focal length, comments etc.)
- Semantic objects contained in the image
    - List of semantic object names
    - Colorful shapes superimposed on the image
- Map representation (using a circle to represent position and a conic shape to represent direction and angle of view)
- Event and persons associated with the picture

To provide the above functionalities and visualizations, various algorithms have been implemented. Probably the most important functionality of the system is that it associates map information with the geospatial parameters recorded in the pictures, transforming them into interactive windows to the outside world. It accomplishes that by calculating the 2D spatial view from the position and direction of picture taking and then with a defined procedure that includes image segmentation, region recognition and image registration, it allows the visualization of (interactive) semantic objects and their location in the image. A more detailed description of the most important algorithms can be found in chapter 7.

The system implementation stores the pictures and the semantic maps in a relational database and provides retrieval functionality for both. The maps are acquired from a semantic map server and can be personalized according to user interests. For example, if the user is interested in the archaeological or cultural sites of Crete but not in its geographic features, the server can then provide a version of the map of Crete without the geographic semantic objects. Detailed descriptions of the system functionalities and interfaces are in chapter 8.

1.3. Contributions

The main contributions of this thesis are listed below:

- The proposed framework provides the means to achieve automatic spatial information retrieval for images in a consistent manner. It is generic enough, enabling different implementations and suggests the use of industrial standards to enforce interoperability and flexibility for the implementations.
- Interoperability of digital cameras with various sensors has been largely ignored by the research community. Since modern technology will allow their widespread integration within the devices in the not so distant future, it is time for the appropriate background research and applications to emerge.
- Innovative algorithms for transforming images into interactive windows to the outside world have been implemented. The result is personal images that allow interactive exploration of their contents.
- The prototype system implemented can be useful for cultural, educational and tourism applications. It allows persons to collect information before going on a trip, take pictures while on the trip and automatically organize, annotate, retrieve and view them as (interactive) images and on top of a map.

1.4. Thesis Structure

The rest of the thesis is organized as follows. Chapter 2 lists the related work on the subjects addressed here and states why this work is different. Chapter 3 is about technologies related to this work and how they were used to implement the prototype system. The framework itself and its content categories are described in detail in chapter 4. Information about the contextual parameters captured and processed by the system is provided in chapter 5. Since semantic maps are a big part of this work, chapter 6 deals with this concept. Chapter 7 contains the algorithms used for transforming photos to semantic maps. Chapter 8 shows the evaluation results of an experiment testing the capabilities of the algorithms developed. Finally, chapter 9 is a presentation of the prototype system implementation of both the client and the server, while chapter 10 concludes the thesis.

2. Related Research

This thesis expands upon various fields of research which will be described in this chapter. In particular, section 2.1 provides a short background on image annotation categories, section 2.2 deals with general purpose image annotation as well as annotations for personal photography and section 2.3 presents some map-based applications which are also relevant to this thesis.

2.1. Image Annotation Categories

The first major topic for this thesis is image annotation. Image annotation and metadata creation are essential for managing the increasing amount of personal collections of images and facilitating image retrieval. Before listing the related work, the main methods of image annotation will be presented, starting from low-level metadata and finishing with label (tag) and ontology-based annotations.

2.1.1. Using Low-level Metadata

Low-level metadata are produced during the image taking procedure and include information about the camera (model, manufacturer), its status at the time of capturing the image (flash fired, focal length), the time and date, comments and more, including possibly the position and direction of the shooting.

The widely used standard describing these metadata is Exif [15]. Most modern digital cameras automatically store the image taking parameters in the Exif header of the digital image. The majority of the parameters provide just technical information, which is not directly useful to the users. Subsequently, processing the metadata in relation with other sources is essential in order to produce richer image annotations. More information about Exif and its parameters (also known as Exif tags) is presented in chapters 3 and 5.

Risto Sarvas [62] addresses the usefulness problem of image capturing metadata and introduces the concept of social metadata suited for personal use. However, as seen from this thesis and other similar works, even raw image metadata proves to be rather important for personal photography use.

2.1.2. Using Tags

A tag is a non-hierarchical keyword or term assigned to a piece of information (such as an internet bookmark, digital image, or computer file). This kind of metadata helps describe an item and allows it to be found again by browsing or searching. Tags are chosen informally and personally by the item's creator or by its viewer, depending on the system.

Assigning tags (also known as tagging) a picture is a very simple way of describing the picture's subject, location, persons and general content. Apart from simplicity, another advantage of using tags is its mainstream use in Web 2.0 and the usefulness of tag clouds in various websites such as Flickr [17]. Flickr has been widely used as a source of (possibly geo-coded) images for different kinds of research using tags and image metadata [29] [58] [39] [33] [59] [78]. The related research will be presented later in this chapter.

While using tags is flexible and easy, tagging is not without its drawbacks. Typically there is no information about the meaning or semantics of a tag. For example, the tag "apple" may refer to the fruit, Apple Inc., the Beatles' music label, or Gwyneth Paltrow's baby. This lack of semantic distinction in tags can lead to inappropriate connections between items. Additionally, selection of "tag terms" is highly individualistic. Different people use drastically different terms to describe the same concept: for example, items related to a version of Apple Computer's operating system might be tagged both "Mac OS X" and "Leopard", and possibly many other terms. Users of tagging systems must make judgments, based on the number of connections and the choices of "tag terms", whether possible connections between items are valid for their interests.

2.1.3. Using Ontologies

In computer science and information science, an ontology is a formal representation of a set of concepts within a domain and the relationships between those concepts. It is used to reason about the properties of that domain, and may be used to define the domain.

The aim of using ontologies to describe multimedia documents is to provide ways to define well structured concepts and their relations that may ease the tasks of annotation and retrieval. As noted in Hare et al [23], ontologies make it possible to improve both automatic annotation and retrieval in presence of imperfect annotation; in fact, an automatic annotator that uses concepts instead of labels maintains consistency of annotations in two ways:

- Avoiding unlikely combinations of annotations.
- Using generic concepts (instead of the more specific) in case of uncertainty. Retrieval is improved using the ontology to drive the query rewriting, e.g. to select more general concepts, or concepts that are somehow related to the query.

Many different languages can be used to express ontologies. This includes open standard languages such as OWL, RDF(S), XML, or even a proprietary language. Image annotation approaches using ontologies are discussed later in this chapter.

2.2. The Semantic Gap

A common problem concerning image retrieval is the so-called semantic gap. Smeulders et al. [67] describes the semantic gap as "the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation". As presented in Hare et al [23], there are two main methods of "attacking" the gap:

- Using the raw metadata and processing it to achieve at least identification of the objects in an image (bottom-up approach), usually in an automatic manner.
- Using ontologies as structured knowledge representations (top-down approach).

The bottom-up approach is closely related to automatic image annotation using raw metadata and image characteristics. The techniques for automatic image annotation include:

- Segmenting an image into regions and trying to annotate each relevant segment (spatial context).
- Utilizing global information for scene classification.
- Taking into account the image capture condition context to derive richer metadata and annotations. The contextual parameters can come from different sources (sensors) or can be derived from other parameters.
- Taking advantage of the information contained in neighboring images (temporal context).
- Exploring and exploiting the plethora of information contained in large collections of labeled (tagged) images.
- Using training sets of images to classify images based on similarity detection.
- A combination of the above.

The following two subsections deal with the bottom-up approach, followed by a subsection describing top-down approaches using ontologies.

2.2.1. General Purpose Automatic Image Annotation

One of the first attempts on automatic image annotation was by Mori et al [45]. In this work, a co-occurrence model was applied to keywords and low-level features of rectangular image regions.

The following researchers focused primarily on image segmentation methods for automatic annotation. They are similar to a small part of this work that includes image segmentation for determining specific geographic and other features of the images. The main differences are that the annotation proposed by these authors is usually just keyword-based and that most approaches are general and not focused on specific image categories or domains. Duygulu et al. [14] used a model inspired by machine translation to map between keyword annotations and a discrete vocabulary of clustered regions. The proposed data-set (Corel) has been widely used in annotation systems in the literature. Jeon et al. [32] demonstrated that probabilistic annotation provides better ranking in the results in contrast to hard annotations thus improving on the results of Duygulu et al. [14] by using a different relevance model. This, however, was outerperformed by the relevance model proposed by Lavrenko et al. [37] which built continuous probability density functions for describing the generating region features process. Hare et al. [24] also avoided hard annotations and proposed an approach where annotation is performed implicitly in a soft manner. An inference network approach to connect regions and their annotations has been used by Metzler and Manmatha [42]. The resulting model allows rich queries with structured operators and term weights to be evaluated for combinations of terms and images. Monay and Gatica-Perez [44] applied and compared two commonly used in text analysis and simple latent space models for image annotation, namely Latent Semantic Analysis (LSA) and Probabilistic LSA (PLSA) and discuss annotation strategies for each one. Feng et al. [16] proposed a relevance model based on a multiple Bernoulli distribution and used rectangular image regions. Jeon et al. [31] also used rectangular image regions; however their approach used Maximum Entropy, a statistical technique which allows one to predict the probability of a label given test data. Blei and Jordan [7] provided a clean probabilistic model based on Latent Dirichlet Allocation [6] which could generate keywords for images, image regions and facilitate text-based image retrieval. Barnard et al. [3] extended this work to multi-modal data.

Turning to scene-oriented approaches, Oliva and Torralba [53] [54] used a selection of low-level global filters to apply basic scene annotations such as 'buildings' or 'streets' with success. They also showed [55] how the presence or absence of objects in a scene can be inferred by using simple image statistics. Using even simpler global features and the technique of kernel smoothing, Yavlinsky et al. [77] proposed a modeling framework based on nonparametric density estimation. The results shown by this research were comparable with the inference network [42] and the model by Lavrenko et al. [37]. Important work has also been done in the identification of specific parts of an image, such as the blue sky, by Galagher et al. [19] and Platt [57]. Real-time automatic annotation has been recently attempted by Li and Wang [39], using a training set of tagged images and a probabilistic method to assign tags to unseen images according to their similarity to the training set. Finally, Vailaya et al. [75] exploited domain semantics for the classification of pictures to broad categories like indoor and outdoor pictures and their subcategories. In contrast with these scene-oriented approaches, the work proposed in this thesis can accurately classify pictures containing mountainous landscape, bodies of water and other geographic features as long as the pictures contain various parameters such as position and direction.

2.2.2. Automatic Image Annotation for Personal Photography

In this subsection, research more closely related to personal photography will be presented. Recently there has been strong interest in semantically annotating and sharing personal photographic collections. Combining the context in which a photograph was captured with information from other external resources and services for the purpose of image annotation has been the subject of some projects. Most of them take advantage of location and time information in order to derive richer information. Other approaches, including this work, take advantage of more parameters such as the camera direction. There is also related research examining the context around the user, using sources such as calendars and social networks.

The research by O'Hare et al. [52] combined context-based information and content-based analysis to facilitate image classification and retrieval. In his thesis, Matthew Boutell [8] categorized the context related to images into spatial (e.g. sky, buildings etc. and their relations), temporal (relation between images in a collection – e.g. time elapsed between them) and image capture condition context (captured metadata). He proposed a graphical model for each category for the purpose of scene classification. The part of his work that is closer to this thesis is presented in Boutell et al. [9]. In it, the captured metadata are processed and used as classifiers for indoor/outdoor scene discrimination and sunset or manmade/natural scene detection. This work also takes advantage of the captured metadata (which are more rich thanks to the additional sensors) and uses them for image annotation.

Another interesting dissertation is by Mor Naaman [47]. Naaman did many studies using GPS, weather, and user-created tags for automatically organizing geo-referenced photo collections and other tasks. Like this work, Naaman also emphasized the use of time and location metadata. The fact that position and temporal information can be used for inferring the picture contents is another similarity between the two efforts. The developed system (PhotoCompas) provided a textual interface for browsing and retrieval that categorized photo collections using information such as location, weather conditions, light status (day/night) and events. The description for each picture included an event/location pair. Events in particular are one of the major content categories of the proposed framework of this thesis. A major difference is that Naaman tried to avoid the use of geographic maps and focused only on textual descriptions of location information. He also did not take into account camera direction. Finally, the descriptions offered by PhotoCompas were textual, interactive exploration of the image contents was not available and ontology based annotations were not considered. However, the proposed system was solid and provided useful insight on the organization of photo collections into events using context-based information.

Sarvas et al. [63] proposed a metadata creation system for mobile images. This approach used a camera system connected to external sources (a remote server) and in that regard it is similar to this work. Another similarity is the context capturing process. The location of the mobile phone is provided by the GSM network characteristics (instead of GPS). The captured images were uploaded to the remote server which located similar photos taken at the same location and calculated the appropriate metadata for image annotation. The remote server compared the uploaded picture with other pictures that depicted the same object and were located in close proximity. It then created relevant metadata and sent them back to the mobile device. A significant downside was that the phone model was difficult to use and the connection to the remote server proved to be slow and frustrating. Another difference from this work is that the process was semi-automatic – it required user input on the validity of the proposed metadata. The next attempt to combine image capturing context with external sensors was by Volgin et al. [76], which correlated XML-based metadata with the images. Volgin et al. provided a more consistent sensor model for communication with smart phones. They used external environmental sensors known as motes [46] to capture context metadata and provide them to the mobile devices at the moment of image capture or at a later time. The sensor concept does not differ very much from the proposed model, however they did not use map information to annotate the image contents and camera direction was also not taken into account.

An interesting sensor that can be integrated into modern digital cameras and most researchers have ignored so far is the digital compass. An early attempt was by Smith et al. [68], where a digital camera was augmented with both position and direction sensors and the metadata captured was used to retrieve historical images from a database. There are many similarities between this attempt and the current work. Smith et al. mainly focused on educational applications of the camera and database system, by encouraging students to take pictures of their local community. The naïve system acquired the position and direction parameters of the pictures taken by the camera and retrieved historical photos of the same places as the ones depicted in the pictures. The students could also manually write ontological descriptions for each picture according to their observations and the system was able to provide additional retrieval capabilities based on these annotations. The system had a simple map application for visualizing the local area and some places of interest. The algorithm for determining the contents of the picture was primitive and did not take into account the angle of view of the camera and the elevation of the area. This thesis expands upon the concept of using orientation information in addition to position and provides more focused visualizations and retrieval capabilities using semantic maps.

The metadata that users are mostly interested in revolve around places, people and events as emphasized by Risto Sarvas [62]. Research directed at this kind of "social" metadata has been performed by many researchers. Cao et al. [10] tried to annotate collections of images rather than single images, using hierarchical event and scene models, while Gallagher et al. [20] utilized user calendars as a source for annotations. Using a trained set of images containing events, a freely available GIS database and tags, Joshi et al. [33] proposed a method for inferring generic activities and events. Rattenbury et al. [59] attempted to automatically extract event and place semantics from a large collection of tagged images, based on tag usage distribution. Finally, Yu and Luo [78] experimented with probabilistic season and location context models to prove that even when the contextual data is not precise, accurate scene understanding can be inferred. Joshi et al., Rattenbury et al. and Yu and Luo all took advantage of the large image database and social photography site Flickr [17], which also includes geo-referenced images.

2.2.3. Ontology-based Photo Annotation

In their identically named article, Schreiber et al [65] used ontologies as a tool for annotating and retrieving images more intelligently. The annotations were performed manually and the evaluation was based on retrieval accuracy. The authors in Hollink et al. [25] extended the research using more popular ontologies in RDF Schema and also discussed the links between the ontologies. Further work by Hollink et al. [26] investigated semi-automatic annotation of images with spatial information and compared their tool's semi-automatic annotations to manual ones. The authors in Mezaris et al. [43] combined an unsupervised segmentation algorithm, a simple ontology defining intermediate-level descriptors and a relevance feedback mechanism based on support vector machines for retrieval in generic image collections. Jane Hunter [27] and Tsinaraki et al. [74] [73] have described efforts to achieve interoperability between MPEG-7 [41] descriptions and OWL/RDF ontologies. In the work by Athanasiadis et al. [2], segmentation results are used to produce MPEG-7 descriptors which are then linked to high-level, domain specific concepts, expressed in RDF Schema. Other works related to ontologies are by Kuo et al. [36], where a temporal and spatial ontology for personal photographs is proposed and by Lim et al. [40] where the authors focus on designing an event ontology.

mSpace [64] is a semantic web interface framework that demonstrates the importance of well-structured information in advanced browsing and visualization facilities in comparison with traditional query-based systems. The system used in the MIAKT project [13] enabled image annotation with ontologically controlled terms. The system also had the capability to automatically annotate the images using content-based descriptors.

2.3. Map Applications and Photos

The second major topic for this thesis is (semantic) maps and visualizations on top of maps. One of the very early relevant applications in culture and tourism was by Christodoulakis et al. [12]. In the system proposed, a laptop/GPS combination allowed the user to find their position on top of a map and according to that, useful information about nearby points of interest was shown. Christodoulakis et al. [11] also developed tools that provide geographic information functionalities and utilize map-based information and combine the location and topological relations of spatial objects. The present thesis was based on many of the principles of these applications and tools. The second most relevant to this thesis work (mentioned before) was by Smith et al. [68], due to the integration of both GPS and digital compass in their camera system. The additional information provided by the digital compass greatly enhances the ability to determine the contents of a picture with assistance from map data.

The map system by Diomidis Spinellis [69] utilized the GPS track log and pictures and created trip log based map visualizations in a web page. The points where the pictures were taken were visible on a non-interactive map. A disadvantage of the system, other than the non-interactive maps, is that it does not support hierarchical navigation by event or location.

The Campiello project [35] aimed at using innovative communication and information technology for developing new relationships between local communities and visitors of historical cities of culture and art. The project's objectives were to connect local inhabitants of historical places better, to make them active participants in the construction of cultural information and to support new and improved connections with cultural managers and tourists. The system included a recommender module, a search module, and a shared data space. The similarity with this work is that they both focus on personalization of the provided information. In Campiello, the users can view and submit comments on objects of cultural heritage. The feedback collected by the system enables it to provide personalized information to different users, according to criteria or proactively. In the system implemented in this work, personalization is achieved by selecting the desirable content (which can be from any domain, not just cultural heritage) and maps containing only objects of interest will be returned by the system.

Perhaps the most important contribution in organizing geo-referenced photos is attributed to Toyama et al. [71] and the World Wide Media eXchange (WWMX) project. Photos were indexed by the WWMX database according to time and location and the WWMX browser visualized them using a map interface and provided retrieval functionality. Different kinds of acquiring location tags, browsing images and visualizing them on a map were presented. The ability to put geo-referenced photos on top of a map is a common characteristic of WWMX and this system. Toyama et al. [71] put a lot of effort in trying to collect a large amount of photos, finding the best methods for their visualization on top of maps and dealing with performance issues for large queries. This work focuses more on providing better annotations for the images using ontologies and takes advantage of more than location and time information to achieve this purpose. The ontological approach also allows for richer retrieval queries.

2.4. Summary

This chapter presented most of the work related to this thesis. The first research category related to this research is image annotation and a great number of works concerning that part has been presented. The other relevant research category is map applications and visualizations. This work is different from the previous attempts because it integrates important research conducted from both of these categories and not only from the first or the second individually. In that sense, it can benefit from the knowledge and conclusions of both worlds. The other reason making this research important is its durability in time. It follows closely the technological improvements in the digital photography domain and is able to adapt and utilize the new technologies for its advantage.

3. Related Technologies and Standards

This chapter will briefly present the most significant technologies and standards used in, or useful to the research concerning the thesis. Section 3.1 presents the dominant features of modern digital cameras, section 3.2 describes the communication protocol used for camera and GPS receiver interoperability, while section 3.3 deals with Exif, the dominant image metadata standard. Section 3.4 summarizes the rest of the technologies used.

3.1. Modern Digital Cameras and their Capabilities

Since the transition from film-based cameras to digital cameras, the domain of photography has evolved substantially in the following categories:

Storage

Most digital cameras utilize some form of removable storage to store image data. While the vast majority of the media types are some form of memory card using flash memory (CompactFlash, SD, etc.) there have been storage methods that used other technologies such as Microdrives (very small hard disk drives), CD single (185 MB), and 3.5" floppy disks. Flash memory cards at the moment of the writing of this thesis have a capacity of up to 32GB with a theoretical maximum of 2TB for the current storage formats.

This advancement, along with the fact that the users can freely transfer the photos to their computers and take new pictures, has led to the tremendous increase in the users' personal collections, thus enhancing the need for research on image annotation, indexing and retrieval.

Resolution

The latest advancements in technology have enabled very high resolutions in the digital images and the resolution values follow Moore's Law[2]. The common aspect ratios for the produced images are 4:3 and 3:2, and this reduces the number of useful image sensor sizes. The camera's maximum resolution is measured in megapixels, which are computed by calculating the pixels comprising the produced image (for example an image with resolution 1,600 x 1,200 = 1,920,000 is produced by a "2 megapixel" camera).

Image File Format

[2] Moore's law describes a long-term trend in the history of computing hardware. It states that the capabilities of electronic devices are improving at an exponential speed. This includes processing speed, memory capacity and, in our case, image resolutions.

The three dominant image file formats for digital cameras are RAW, TIFF and JPEG. A raw image is the unprocessed set of pixel data directly from the camera's sensor, while TIFF and JPEG are compression formats (lossless and lossy respectively). The latter is the most commonly used by modern digital cameras. It contains various metadata at its header. The metadata standard used for describing the image capture and other parameters is Exif [15]. More information about Exif and its metadata tags is presented in section 3.3.

Wireless Connectivity
Since the majority of portable devices have already begun integrating wireless connection technology, digital cameras are also following this trend. Bluetooth™ enables easy connectivity with other sensors such as GPS and also allows sending the photos taken to a nearby device. Some digital cameras also feature a Wi-Fi radio for the same purposes. It the future it is expected that these technologies will allow the cameras to communicate with other external resources and acquire essential information about the image taking context, before and after the image capture and without the help of other computers or PDAs.

Sensor Integration
Digital cameras can now interoperate with external sensors using wireless technology. Since the Exif metadata standard contains location tags, pairing a camera with a GPS receiver results in recording automatically the location information in the Exif header of the image. There are a few such receivers that also integrate a digital compass, allowing for direction information to be recorded additionally. Finally, laser rangefinders can be used to calculate and store the subject distance for the photos. For the experiments conducted for this thesis, a camera system utilizing an external compass-enabled GPS receiver with a compatible digital camera has been used. Since then, these sensors have already been integrated internally in some cameras and in some mobile phones.

3.2. GPS and NMEA-0183

NMEA 0183 (or NMEA for short) is a combined electrical and data specification for communication between marine electronic devices such as echo sounder, sonars, anemometer (wind speed and direction), gyrocompass, autopilot, GPS receivers and many other types of instruments. It has been defined by, and is controlled by, the U.S.-based National Marine Electronics Association [50]. Its communication protocol is text-based.

A GPS receiver transmits the satellite data in forms of NMEA sentences, short text strings that contain information such as position, speed, etc. The most important NMEA sentences include the GGA (Global Positioning System Fix Data) which provides the current satellite fix data, the RMC (Recommend Minimum Specific

GPS/TRANSIT Data) which provides the minimum GPS sentences information, and the GSA (GPS Dilution of Precision and Active Satellites) which provides the Satellite status data. GPS receivers with integrated digital compass also output the HCHDG sentence, containing heading/direction information.

For the image capturing context, NMEA sentences provide the following useful information:

- *GGA sentence*: latitude, longitude, satellite fix quality, number of satellites being tracked, altitude
- *GSA sentence*: satellite information, satellite fix type (no fix, 2D, 3D), dilution of precision
- *RMC sentence*: time of the fix, latitude, longitude, speed over the ground in knots, track angle in degrees, date, magnetic variation
- *HCHDG sentence*: magnetic heading, magnetic variation

While the RMC sentence provides the most relevant position parameters, altitude can only be obtained from the GGA sentence. When using a digital camera integrated with a GPS receiver, almost all of the information contained in the transmitted sentences is automatically embedded within the image.

3.3. Exchangeable Image File Format (Exif)

Exif is a specification for the image file format used by digital cameras. The Exif metadata standard's use by camera manufacturers is universal; however the specification is not currently maintained by any industry or standards organization. It defines a number of metadata tags that are embedded in the image's header during image capture and they cover a broad spectrum:

- Date and time information. Cameras keep time and date information and record it in the metadata.
- Camera and image capture settings and parameters. Camera model and make are parameters that do not change for the same camera, while others concern the process of image taking and include focal length, aperture, metering mode, orientation and other dynamic information.
- A thumbnail for previewing the picture on the camera's LCD screen, in file managers, or in photo manipulation software.
- Descriptions and copyright information.
- Position and direction values.

As will be discussed in chapter 5, there are many useful tags that can be used for calculating other contextual parameters and assist in associating the images with spatial and other information. In the system developed, the relevant Exif tags are extracted from the images and the parameters are used for automatic image annotation and association with map features and semantic objects.

3.4. Other Relevant Technologies and Standards

The rest of the technologies used in the proposed framework or in the implemented system are the following:

Elevation Data

Any map of an area can be augmented with elevation data, so that each point on the map is or can be associated with an elevation value. Elevation data can be obtained from Digital Elevation Models (DEMs) which themselves are produced using various techniques, such as remote sensing and land survey.

The maps used in this work have been augmented by a decent quality DEM from the National Aeronautics and Space Administration [49] agency's Shuttle Radar Topography Mission (SRTM) [66]. This global DEM contains elevation values for approximately every 30m (in the United States) and 90m (in the rest of the world). Even though the research for this thesis has been conducted in Europe, thus having a lower quality DEM available, the results were satisfying and only close range elevation values had a negative impact overall.

Ontology Languages

Ontologies can be expressed in various languages, with RDF Schema and OWL varieties being the most popular. They are based on XML and provide the appropriate mechanisms for domain and semantic object descriptions required. They also provide sophisticated query mechanisms. The proposed framework is using ontologies as a means to consistently describe the semantic objects that the users are interested in. These can be expressed by RDF Schema and OWL, but any simple ontological model can also be used.

Map Representations of Semantic Objects

The maps defined in this thesis contain semantic objects on top of them and a standardized way of representing them can be used. The two dominant industrial standards for representing geographical or other features and maintained by the Open Geospatial Consortium [56] are GML (Geography Markup Language [21]) and KML (Keyhole Markup Language [34]).

GML and KML are both XML-based and among other things provide mechanisms for describing map objects. GML has almost unlimited flexibility and is perfectly suited for machine maintained data, but is very complex and too verbose to write and manage without tools. KML on the other hand is focused on geographic visualization, including annotation of maps and images and is complementary to GML. In the future it is expected that both languages are going to use similar geometry representations and as a result, they will both be suitable for semantic object representation using simple geometric shapes (polylines, polygons) or more complex ones.

System Implementation Programming Language

The programming language used for the system implementation of the proposed framework was Java [30]. Java was chosen because it is open-source, widely supported and offered many tools and libraries (for example an image metadata reading library) that were required for this work. Java's Swing was the graphical user interface framework used for building the prototype system's visual interfaces and Java2D was used for the 2D geometric representations required.

4. The SPIM Framework

In this chapter, the proposed framework will be described in more detail. Section 4.1 provides an introduction to the main ideas of the framework and sections 4.2 through 4.5 discuss the framework's content categories. Section 4.6 shows the framework's general architecture and section 4.7 summarizes the chapter.

4.1. Introduction to the Framework

The SPIM (SPatial Image Management) framework is a semantic spatial information processing framework and its purpose is to achieve semantic interoperability between different applications and services that utilize spatial knowledge, international standards and ontologies. The framework takes advantage of the capabilities of modern digital cameras and an increasing set of external or camera-integrated sensors to obtain contextual spatial information. The result of feeding this information to the applications and services is a rich knowledge environment that enables semantic applications, smooth integration of new spatial information and knowledge and reduction of the manual semantic indexing effort. The tasks of archiving, retrieval and visualization are also facilitated. The use of the various spatial services at a given spatial location together with the sensors provides a personal assistant that augments the capabilities of the user to understand and learn about the near and remote environment.

The framework considers the use of maps, along with the semantic objects present in them and combines all the different kinds of information with images and the semantic objects of the spatial environment depicted in the images. This allows for powerful functionalities for the systems built according to the framework and also enhances the corresponding visualizations.

Personalization is achieved either through personalized versions of maps of areas the user is interested in or through the automatic assignment of events to the photographs. The people taking part in these events or being present in the photographs can also be assigned to the pictures to enable querying and facilitate retrieval.

4.2. Description of Semantic Entities of the Real World

The ontology model used in the framework assumes that the objects of interest to the user are described by domain ontologies. The choice of ontologies for the enrichment of semantic objects has many advantages. Firstly, ontologies provide a concrete and well-accepted way of representing and describing semantic concepts. Secondly, semantic interoperability between the user and the system or among users with common interests is ensured. Finally, image indexing and retrieval can consistently be facilitated by an ontology query tool. For an example of the usefulness of this approach, consider a user interested in archaeology concepts. It is highly probable that this user will be taking pictures of objects that correspond to the archaeological domain. With this in mind, a well-implemented system should be able to provide relevant archaeological semantic information about the pictures taken.

Objects of the real world that are described by concepts of ontologies are called individuals. For instance, temples may be described in general by an archaeological ontology, however a specific temple of Zeus in a given location is an individual. Individuals should be uniquely defined by the framework, so specific IDs are given to each one in order to differentiate them and recognize when the same individual appears in more than one image.

Individuals can also be composed of other individuals to indicate "has-a" relationships, for example the fact that certain temples have columns or a sea area contains islands. Properties or attributes define certain values of the individuals' characteristics, for example the height of a building or the population of a village. The framework is responsible for maintaining and processing this kind of information consistently.

A special case of ontology is one that describes geographic concepts, such as lakes, mountains, islands, roads etc. A geographic ontology can be particularly useful, not just for semantically enriching images of geographic features but also for detecting them in the image and allowing greater annotations and visualizations.

4.3. Semantic Maps

Semantic maps are interactive maps in a digital form that visualize location related information. They can also visualize representations of semantic objects (individuals) of the real world using geometric shapes. A semantic map represents the objects, their information and the domain knowledge in it using international standards. This results in the use of ontologies for the domain knowledge and KML or GML profiles (subsets) for the geometric representation of semantic objects. Semantic maps are interactive; the semantic objects on top of a map respond to user input and all the associated information and semantics about them can be shown. The same applies to geo-referenced photos – they can be visualized on top of the map and respond to user input.

It is assumed by the framework that the ontologies used by the semantic maps are the same (or mapped) to the users' ontologies to achieve interoperability. Since the user is interested in specific semantic entities that belong to specific ontologies and domains, it is possible to filter the visible individuals on top of the map accordingly, making it personalized. This gives the user the capability to extract from the map only the information that is relevant to him/her.

Correlating the spatial information available in a semantic map with the spatial view calculated from a photo with both location and direction information, in combination with image segmentation, object detection and registration algorithms, allows for determining the semantic objects and their position in the image. The pictures then can be considered as windows to the real world and functionality for interactive exploration of their content becomes available, essentially making them semantic maps themselves.

More details about semantic maps, their method of creation, types of objects that appear within and visualizations are presented in chapter 6. Chapter 7 presents the algorithms for transforming photos to semantic maps.

4.4. Image Content Descriptions

Image content descriptions concern how an image or regions of an image are identified by the framework. Image annotation is not a onetime procedure; many of the annotations or parameters assigned to an image can be retrieved or become more detailed (for example by getting a more accurate result from a sensor) at a later stage. Being able to uniquely identify an image by the framework also allows for the same individual descriptions to be assigned to different images and gives the ability to link the images in a collection to provide guided tours.

Segmenting an image into regions or segments can be very helpful in assigning individuals to specific regions and showing where they are located in an image. In order to segment an image into regions, various types of algorithms have been developed in the literature, each using specific methods (clustering, histograms, edge detection, region growing etc.). There is also software performing segmentation functions, either open source or commercial.

After individuals depicted in an image have been assigned to image regions (following a segmentation process) or their position within the image has been specified otherwise, the user can apply queries and interact with the annotated image. For example, assuming that the user has taken several photographs from the inside of a stadium, a powerful segmentation algorithm should be able to identify parts of the picture such as the play area, the seats and the field lights. The whole stadium as an entity is an individual composed of other individuals like the previously described parts. The user could perform queries like the following: "In which images in my collection is the play area visible?", "Show me all associated information about the stadium", "What are the properties of the play area (dimensions, color, etc.)?", "What semantic objects are visible in this photo?" and more. When the exact position of the semantic individuals within the picture is known, the user should be able to view them overlaid on the image and interact with them using a mouse or other input device. This would reveal all the associated information about the selected individual from the domain knowledge, including information about the images in the user's collection that depict that individual.

4.5. Events and Persons

Another way the framework supports personalization is through events and persons. It is a known fact that users usually remember specific events rather than specific photos. Events enclose user activities and are defined by a name, a place, the persons participating in them and other information. The framework should be able to automatically assign and organize photos to events using either image metadata or image processing techniques. Persons that appear in photos can be identified using advanced face recognition and correlated with the appropriate events.

The event structure is hierarchical; events contain other events. For example a user can specify an event called "Summer in Crete" that includes events of smaller duration, such as "Trip to the White Mountains of Crete". Pictures, according to the time they were taken, should be assigned to the appropriate event, and all information about that event and the relevant events including it should be available. Like image regions, events can be effectively described using industrial standard metadata specifications such as MPEG-7 [41].

Persons can both be assigned to photographs and events. Photographs of a particular event should contain persons that also participate in that event and the appropriate mappings should be made. Queries based on events and persons participating in events and appearing in photos could then be constructed.

4.6. General Architecture

Since the SPIM framework is generic, there is no strict architecture defining it. Each implementation however should follow the general architecture presented in this section.

Figure 1 shows the entities of the architecture and their relations. The architecture can be split into two parts, the image taking and metadata association part and the part that describes the ontologies and semantic maps that take part in the framework.

In the first part, the user has a digital camera (or any compatible image capturing device) in his possession and uses it to take photographs of objects and places that fit his or her interest. The camera has the capability to connect to a number of external or integrated sensors that each provides more metadata about the photograph to be captured. The metadata is either recorded along with the photograph's metadata or kept by the implemented system for later association with the photograph. Persons participate in events and appear in photographs. In addition, photographs are usually related to an event and the persons that appear in them should also be assigned to the appropriate event.

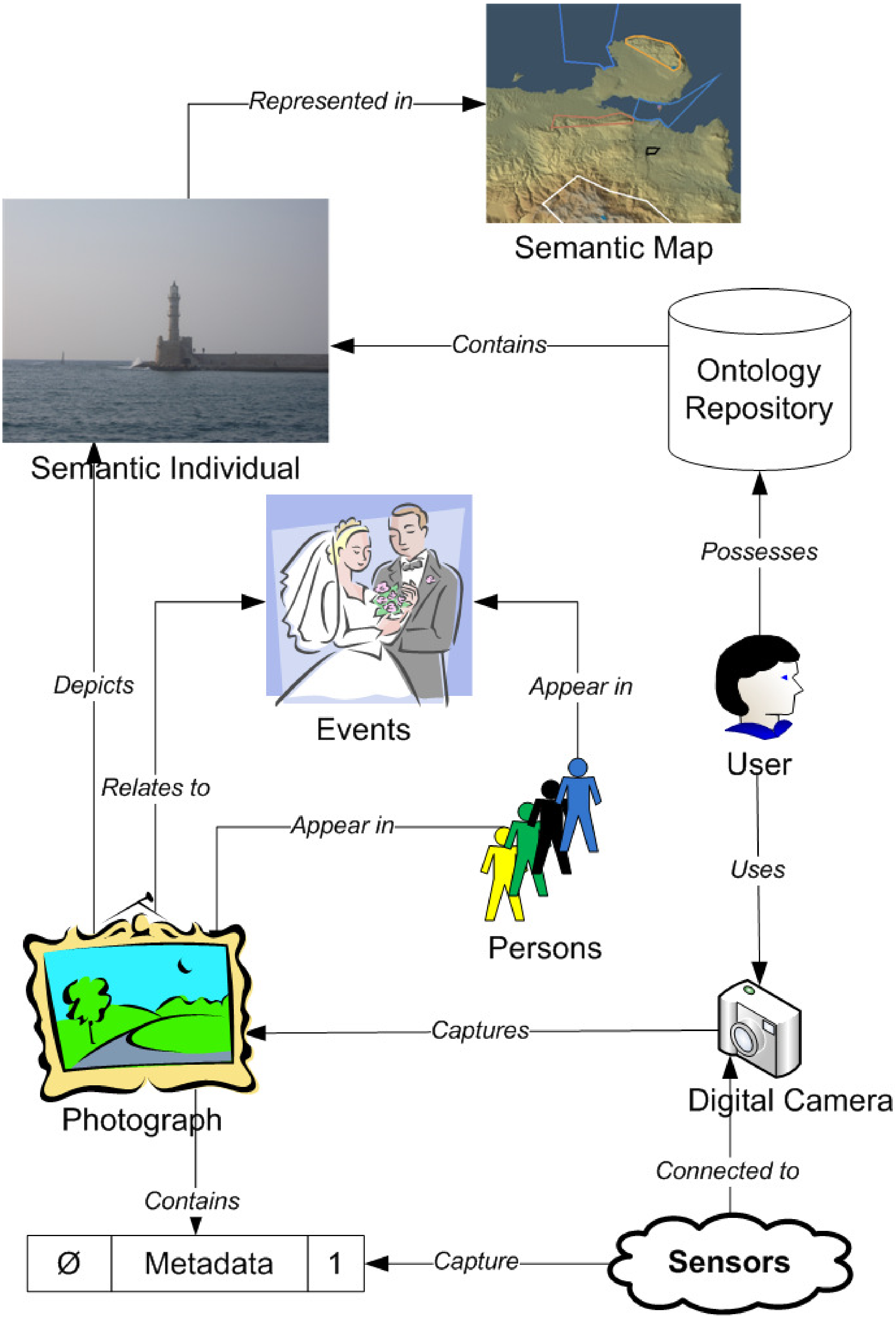


FIGURE 1: GENERAL ARCHITECTURE OF THE SPIM FRAMEWORK.

The second part is connected to the first in two ways. Firstly, the ontologies that are of interest to the user are assumed to be the same or mapped to the ones used by the implemented system to ensure maximum interoperability. Secondly, semantic individuals that belong to these ontologies and are depicted in the photographs are the same that have representations on the system's semantic maps. In this context, algorithms for the association of the semantic individuals to the photographs can be implemented.

4.7. Summary

In this chapter, the SPIM framework and its content categories have been presented. SPIM has been described as a generic framework for semantic spatial information processing that manages spatial multimedia information including pictures, objects and persons that appear in pictures, semantic maps and objects that they contain. The general architecture that includes the relations between ontologies, maps, pictures, sensors and the rest of the entities described in this chapter was revealed at the end.

5. Capturing and Calculating Contextual Parameters

This chapter describes the contextual parameters captured by some of the sensors considered by the SPIM framework, the parameters that can be derived from them and how they are used by the framework to help associate image content with semantic objects of the real world. Section 5.1 describes the parameters concerning the current camera setup and the image taking conditions and section 5.2 deals with position and direction parameters captured with special sensors. Section 5.3 shows how these parameters are combined and more contextual parameters are derived from them.

5.1. Camera and Image Taking Parameters

Camera and image taking parameters are the first step towards automatically inferring information about the contents of an image. These, along with information extracted from sensors (GPS receivers, digital compasses, tilt/rotation sensors, distance measuring devices etc.) or external information sources (for example spatial information contained in semantic maps) and the addition of (semi-)automatic image and graphic processing algorithms, all take part in the creation of richer annotations describing the content. It should be considered that the sensors may not all be available at the time of image taking and have variable accuracy, so the best result measurement should be selected and processed whenever possible. This gives the ability to improve the association accuracy with later processing of the captured information.

Since the dominant metadata standard for describing camera and image taking parameters is Exif, a short description of its relevant tags captured when taking a picture follows.

- **Make**: The name of the digital camera's manufacturer.
- **Model**: The camera's model.
- **Orientation**: Defines the general orientation of the image. Eight values are allowed: horizontal, rotate 180° rotate 90° clockwise, rotate 270° clockwise and their mirrored versions.
- **Exposure time**: Displays the image's exposure time given in seconds.
- **Focal length**: A camera lens' property, focal length determines the magnification at which it images distant objects.
- **Aperture value**: The aperture stop of a photographic lens can be adjusted to control the amount of light reaching the film or image sensor. The values

range from 0 to 10 with greater values reducing the camera lens' exposure to light.

- **F-Number**: Expresses the diameter of the entrance pupil in terms of the focal length of the lens. Effectively, its value can be calculated by dividing the focal length with the aperture diameter.
- **Date/Time**: The date and time of image creation in "YYYY:MM:DD HH:MM:SS" format.
- **Brightness value**: A value representing the brightness of the image which can be transformed to cd/m^2 (the SI unit for luminance).
- **Flash**: Indicates if the camera flash was on/off its mode and if it fired or not.
- **White balance**: Indication if the camera's white balance mode was used.
- **Saturation/Sharpness/Contrast**: These tags indicate the direction of saturation/sharpness/contrast processing applied by the camera when the image was shot.
- **User comment**: Provides space for comments. There are digital cameras that enable the input of such comments.
- **Color space**: Usually sRGB [70] or undefined otherwise.
- **Image width/height**: The image dimensions and resolution expressed in pixels.
- **Shutter speed**: The effective length of time a shutter is open. Directly related to exposure time.
- **Subject distance**: The distance to the subject, given in meters. This parameter is rarely filled in by most digital cameras without external help. Only a few can interoperate with distance measuring devices in order to record the subject distance during image capture and not at a later stage.

Section 5.3 shows how some of these metadata can be used to derive other useful contextual parameters and information.

5.2. Position and Direction Parameters

Determining the position and direction of the camera is very important and essential when interpreting the spatial picture content. In order to capture these parameters, appropriate sensors can be used. These sensors are usually small and cheap and sometimes they are integrated within the digital cameras.

Position information is provided by GPS[3] receivers, while direction information from digital compasses. It is possible to integrate a digital compass within a GPS receiver and it will transmit both types of parameters.

[3] There are two other global positioning systems other than GPS (GLONASS and Galileo); however in this thesis the generic abbreviation GPS will sometimes be used to represent all of them.

Exif provides metadata tags that accommodate this kind of information. These are the following:

- **GPS version ID**: If this field has the default value 2.2.0.0, it indicates that GPS data are available for this image.
- **GPS latitude reference**: Indicates whether the latitude is north (N) or south (S) latitude.
- **GPS latitude**: The latitude expressed in either degrees-minutes-seconds or degrees-minutes format.
- **GPS longitude reference**: Indicates whether the longitude is east (E) or west (W) longitude.
- **GPS longitude**: The longitude expressed in either degrees-minutes-seconds or degrees-minutes format.
- **GPS altitude reference**: Indicates the altitude used as the reference altitude. If the reference is sea level and the altitude is above sea level, 0 is given. If the altitude is below sea level, a value of 1 is given and the altitude is indicated as an absolute value in the GPS altitude tag. The reference unit is meters.
- **GPS altitude**: Indicates the altitude in meters based on the reference in the GPS altitude reference tag.
- **GPS timestamp**: Indicates the time as UTC (Coordinated Universal Time).
- **GPS satellites**: Indicates the GPS satellites used for measurements. This tag can be used to describe the number of satellites, their ID number, angle of elevation, azimuth and other information in ASCII notation.
- **GPS measure mode**: Shows the GPS measurement mode (2-dimensional or 3-dimensional).
- **GPS DOP**: Measures the dilution of precision for the current measurement. Describes the geometric strength of satellite configuration on GPS accuracy. A smaller value means the measurement is more precise.
- **GPS speed reference**: Indicates the unit used to express the GPS receiver speed of movement (kilometers per hour, miles per hour or knots).
- **GPS speed**: Indicates the speed of GPS receiver movement.
- **GPS track reference**: Indicates the reference for giving the direction of GPS receiver movement (true direction or magnetic direction).
- **GPS track**: Indicates the direction of GPS receiver movement. The range of values (in degrees) is from 0.00 to 359.99. This tag can be used in lieu of the actual image direction, when a digital compass is not available. The disadvantage is that the direction the user is heading usually differs from the direction he or she is taking pictures.

- **GPS image direction reference**: Indicates the reference for giving the direction of the image when it is captured (true direction or magnetic direction).
- **GPS image direction**: Indicates the direction of the image when it was captured. The range of values (in degrees) is from 0.00 to 359.99.
- **GPS map datum**: Indicates the geodetic survey data used by the GPS receiver (usually WGS-84).
- **GPS date stamp**: A character string recording date and time information relative to UTC (Coordinated Universal Time).
- **GPS differential**: Indicates whether differential correction is applied to the GPS receiver.

The disadvantage of the Exif specification in recording position and direction parameters is that it does not contain tags for storing the camera tilt and precise rotation, so they cannot be recorded within the image automatically.

5.3. Derived Parameters

This section lists some of the metadata that can be calculated or derived from the captured contextual parameters along with their use for semantically enriching the images. The following calculations have been used in the framework's system implementation.

Image Sensor Size

The size of the image sensor for capturing images varies in value for every manufacturer and model. When the dimension is not specified in the sensor size value, horizontal image sensor size is assumed (instead of vertical or diagonal). It is a critical parameter needed for calculating the angle of view. The implemented system needs to maintain a table of digital camera manufacturers and models that hold the sensor size values for each such combination, since this information is not embedded in the Exif header.

Angle of view

Angle of view describes the angular extent of a given scene that is imaged by a camera. In simpler words, when zooming in with a digital camera, the angle narrows. As a value by itself, angle of view can only be used to probabilistically infer whether the photo is focused on a subject or trying to capture a wider view of the surroundings. However, the knowledge of both the angle of view and the direction of the picture allows for greater understanding of the photographer's intentions and offers great visualizations on top of maps.

Calculating the angle of view (α) is achieved using the following equation:

$$\alpha = 2\tan^{-1}\frac{d}{2f}$$

d represents the sensor size. If the value refers to the horizontal sensor size, then the angle measured is the horizontal angle. The same applies to vertical and diagonal sensor dimensions. f is the effective focal length of the camera lens. Sometimes it is slightly different from the lens' stated focal length (F) found in the Exif header of the image due to the effects of magnification on macro photography. Using a magnification factor(m), the calculated focal length is as follows:

$$f = F \cdot (1 + m)$$

Some camera manufacturers provide sample angle of view/focal length ratios for their camera models.

Subject Distance

When a photographer focuses on an object and takes pictures of it, a very useful contextual parameter is the distance from the camera to the object. The goal is to (semi-)automatically infer which objects are in focus, using help from semantic spatial information contained in semantic maps. This parameter is important for a variety of reasons and its usefulness depends or the amount of already known contextual parameters:

- Subject distance + position: Without the knowledge of the image direction, all objects that are positioned near the user in all directions and at approximately the measured distance are candidates for being focused objects.
- Subject distance + position + direction: This configuration narrows the candidate objects that are focused. Angle of view also helps the process.
- Subject distance + position + direction + hyperfocal distance: This combination of contextual parameters enables the calculation of minimum and maximum focusing distance, enabling clearer interpretation of the image contents and the photographer's intentions. The detailed calculation of the distances is described later in this section.

Obtaining a value for subject distance can be accomplished in several ways. They are the following, sorted by increasing accuracy.

1. Manually: The user can fill in the value manually, either using the camera's interface or later using software.
2. An approximation of the subject distance can be computed with two or more images focused on the same object. The difficulty is not in the calculation but in determining which images focus on the same semantic object. This method

assumes that position and direction information is not only available but also very accurate.

3. There are cameras with integrated rangefinders but usually these do not provide accurate measurements.
4. Some laser rangefinders can interoperate with digital cameras and record the subject distance measured in the corresponding Exif tag.

Circle of Confusion

The circle of confusion diameter limit is sometimes defined as the largest blur circle that will still be perceived by the human eye as a point when viewed at a distance of 25 cm. The diameter limit value is used for the calculation of the hyperfocal distance and it depends on the digital camera model. Subsequently, along with the sensor size for each digital camera model and manufacturer, the circle of confusion needs also to be kept.

Hyperfocal Distance

The hyperfocal distance is the distance beyond which all objects are acceptably sharp, for a lens focused at infinity. Like the angle of view, hyperfocal distance (H) can be approximated with acceptable precision using the following formula:

$$H \approx \frac{f^2}{Nc}$$

f denotes focal length, N is the F number (extracted from the Exif data) and c is the circle of confusion limit. For digital cameras, focal length must be expressed in 35mm equivalent focal length. This value is also supplied by the camera model and manufacturer.

Minimum/Maximum Focus Distance

Minimum focus distance refers to the minimum distance from the camera after which objects are in focus, while maximum focus distance refers to the distance before which objects are in focus. This means that objects in focus appear between these distances. Calculating them can be achieved using hyperfocal distance and subject distance and this formula:

$$D_{min} \approx \frac{Hs}{H+s}$$

$$D_{max} \approx \frac{Hs}{H-s}, for\ s < H$$

H is the hyperfocal distance and s is the subject distance. These, along with position and direction parameters enable sophisticated visualizations on top of a semantic map, as will be seen in chapter 6.

5.4. Summary

In this chapter the majority of the parameters concerning an image were presented. It was also shown how more contextual parameters can be calculated in order to determine the (focused or not focused) objects in a picture and allow visualizations on top of semantic maps. The implemented prototype system takes some of these calculations into consideration and offers the desired functionalities.

6. The Semantic Map

This thesis emphasizes the concept of the semantic map. This chapter shows additional details about semantic maps, including the way they can be created, information about the semantic objects that appear on them and how they can be visualized.

6.1. Introduction

Traditional digital maps usually contain geographical information or just names of populated places, have a defined scale and show a rectangular area as seen vertically from an elevated position. Semantic maps on the other hand have many additional capabilities.

The first important characteristic of semantic maps is that they have the functionality to associate each pixel on the map with geographic coordinates. This enables them to represent any kind of spatial information taken from any angle and not just from above with the top side of the map pointing north. It also allows for semantic objects to be placed onto them. Section 6.2 analyzes this association.

The second important characteristic is that they enable the integration and the visualization of semantic objects overlaid on top of them. The semantic objects have geographic representations and they can be visualized either by transforming them to pixel coordinates or by enforcing a different pixel based (and probably more accurate) representation in advance. The semantic objects are not only shown on top of the map, they can be interacted with and all the information that has been associated with them becomes visible. Semantic maps can additionally visualize geo-referenced photos (with location as well as direction information), which are also interactive.

When certain parameters of the system are available, it is possible to apply certain algorithms to convert an image into a semantic map. The minimum parameters required are semantic map elevation data, image position and image direction. With these parameters, a 2D representation of the spatial view of the picture is calculated along with the position of the encountered semantic objects. The objective is to successfully register this 2D representation with the image, therefore revealing the position of semantic objects and all the information associated with them on top of the image. The image then is considered to be a semantic map (although without the knowledge of the geographic coordinates of each pixel). This is the third major characteristic of semantic maps, discussed in detail in chapter 7.

Finally, semantic maps emphasize personalization. A semantic map service has the ability to provide personalized versions of semantic maps that only include objects of selected domains and types. Section 6.3 provides details about the semantic objects and how they are associated with semantic maps, while section 6.4 shows the related visualizations.

6.2. Creating a Semantic Map

In practice, the difficulty of creating a semantic map can vary from easy to very difficult, according to the manual work and annotations that need to be performed. The following subsections describe the procedure.

6.2.1. GPS to Pixel/Pixel to GPS Transformation

In order to have a mapping from pixel coordinates to GPS coordinates or vice versa, a transformation matrix or function needs to be computed first. The maps built with this method are overhead views of locations with no restriction on where the true north is located. In general, map creators can choose any image and turn it into a semantic map – assigning geospatial coordinates to each pixel – however this would usually require a lot of manual effort according to the smoothness of the landscape and the angle it is viewed from.

The simplest case for the transformation is to compute a transformation matrix [72]. The transformation matrix is defined as the matrix with which a pixel point (vector) must be multiplied with to calculate the corresponding geographic point (vector). Exactly 3 different pixel-to-GPS mappings are needed to create the matrix and after it is created, a simple matrix multiplication performs the one way transformation. For the inverse GPS-to-pixel transformation matrix, the same method can be used, considering the pixel points as the result of the multiplication and the GPS points multiplied by the matrix. This method is accurate for simple overhead maps.

When 6 or more point mappings are available, a more sophisticated method can be applied for the transformation. It includes calculating second-degree bivariate polynomial coefficients and least squares fitting techniques for fitting the mappings [38]. The result is a transformation function that performs the transforming calculations. This method should also work for more complex semantic maps (not just overhead views), as long as the accuracy of the initial point mappings is acceptable.

There are many ways to acquire the initial point mappings. One way is to mark on the digital map some well-known landmarks and associate their coordinates on top of the map with their known geographic coordinates. There is also GIS software that provides geospatial coordinates for global maps, for example Google Earth [22]. Finally, digital elevation model visualization tools provide easily transformable maps, since the elevation data are distributed with location information and the tools visualize them as images where the location of each pixel is established.

6.2.2. Incorporating Semantic Information

Semantically annotating the maps is a difficult process. One needs to choose what domain knowledge to incorporate in each map, locate the appropriate ontologies and search for any relevant semantic objects that are located within the bounds of the map. After this choice, all the semantic objects along with their attributes need to be placed on top of the map. They contain information about their shape in real world coordinates and as a result transforming these coordinates into map coordinates is trivial. It is also possible to use specific geometric representations for the current map only (in pixel coordinates) to increase the shape's accuracy. The system that supports semantic maps should be able to interpret the geometric representations of the objects whether they are in a standard language (GML/KML) or made up of simple points connected as polygons or polylines.

Other standard attributes of semantic objects are height, which can be used for discriminating overlapping objects and visualizing them in 3D and color, which defines their appearance in visualizations. Of course, objects should have a description about them along with any other useful information about them encoded in ontological languages.

6.3. Ontologies, Semantic Types, Semantic Individuals

This section covers the current system implementation's version of semantic objects. The model describing the ontologies is rather simplistic and it is possible to translate it to a more standardized model using sophisticated ontological languages such as OWL.

Ontologies cover different knowledge domains and each one has a name. This domain is the conceptual base for a number of abstract semantic types, each with its own attributes. For example, the domain ontology "Tourism" could have a semantic type "Hotel". Sample attribute values of the type "Hotel" can be "Stars" (1/2/3/4/5), "Swimming Pool" (true/false) etc. The relation between semantic types is hierarchical. For example the type "Hotel" could have the subtype "Luxurious Hotel" and even more attributes common to luxurious hotels only. All of these concepts are abstract.

The entities that represent concrete semantic objects of the real world are called semantic individuals. Following the previous examples, a semantic individual instance of the type "Luxurious Hotel" could be "Europa" and would contain values for all the attributes defined under "Luxurious Hotel" as well as "Hotel". All individuals have values defining their general height and color and also a short description.

Geometric representations of individuals in this thesis include polygons and polylines only, but in the general case any geometric shape defined by geographic standards such as GML or KML should be supported. The important fact here is that the geographic coordinates of each individual do not change, they remain the same. The thing that changes is their specific representation on top of each map. This distinction allows the individuals to be independent of the semantic maps.

6.4. Semantic Map Visualizations

In this section, most of the implemented system's visualizations that have to do with semantic maps will be shown.

Semantic Map Image

The image of the map itself, as mentioned earlier, can come from virtual globe software such as Google Earth, digital elevation model visualization tools and also from any other available digital image that can later be semantically annotated with individuals.

Figure 2 shows a semantic map image taken from Google Earth and showing the west part of the island of Crete, while Figure 3 shows the same area as visualized by digital elevation model (DEM) visualization software [1]. The visualization software takes as input the elevation data provided by NASA and provides appropriately colored shapes according to the elevation of each point. No individuals have been placed on these maps yet.

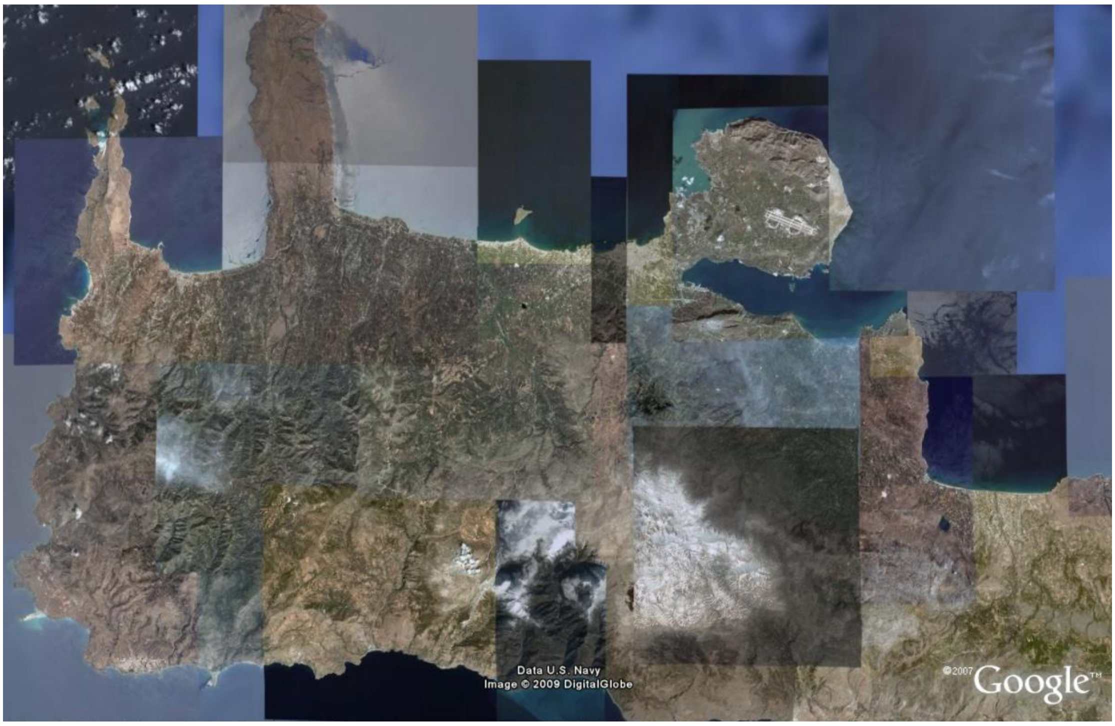


FIGURE 2: THE PREFECTURE OF CHANIA – DIGITAL MAP IMAGE TAKEN FROM GOOGLE EARTH.

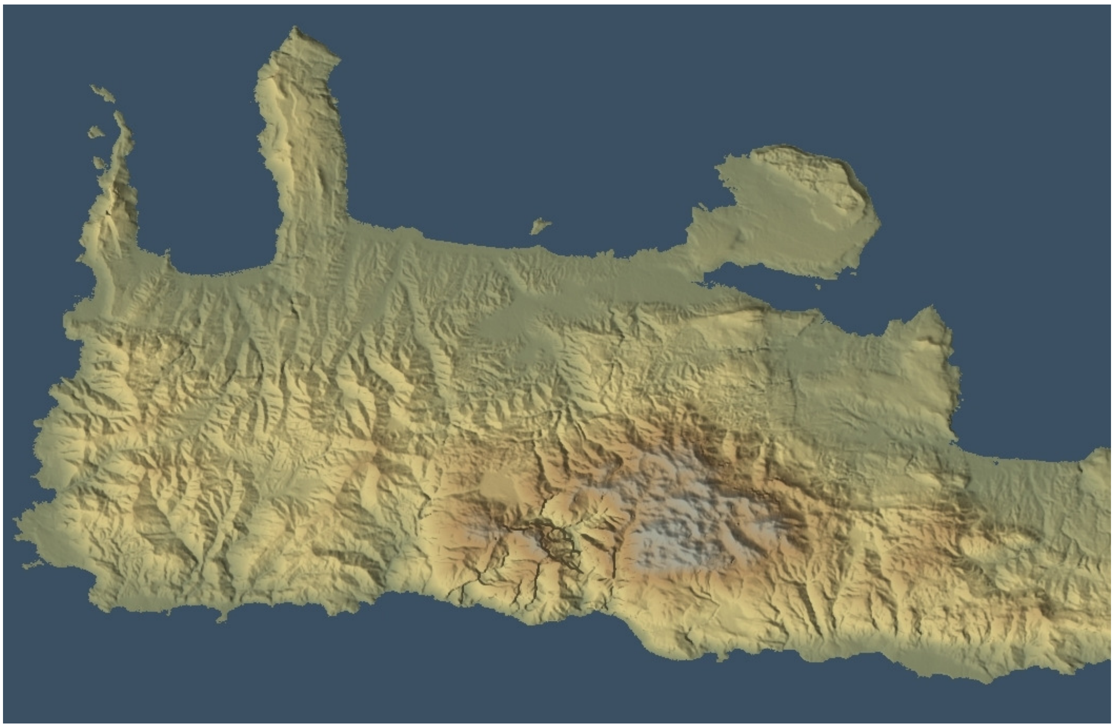


FIGURE 3: THE PREFECTURE OF CHANIA – DIGITAL MAP IMAGE CREATED BY DIGITAL ELEVATION MODEL VISUALIZATION SOFTWARE.

Semantic Map with Individuals

Individuals on top of semantic maps are visualized as colored polygons (polylines are also supported) and are called individual footprints. Polygon representations are displayed in Figure 4 and Figure 5, using the same maps shown in the previous subsection. It is notable that the semantic individuals involved in both maps are the same. They contain the geospatial coordinates describing their shape. The only thing left to do in order to visualize them is to use the transformation process available to find which pixel each (GPS) polygon edge is mapped to. The fact that semantic individuals are independent of their map representations is very important. It allows them to be placed on top of any map depicting areas in their proximity, as seen from the figures below. They are considered interactive objects and can be selected with an input device to view all the relevant information about them.

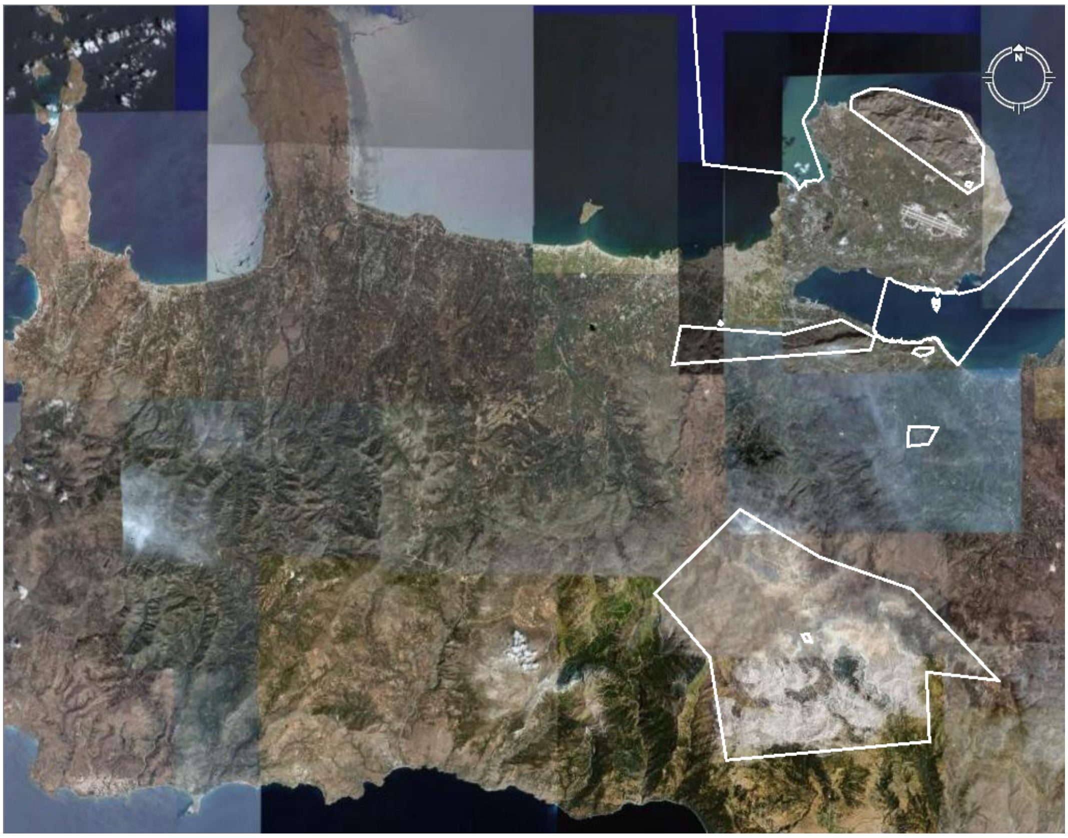


FIGURE 4: THE GOOGLE EARTH MAP IMAGE OF FIGURE 2 ANNOTATED WITH SEMANTIC INDIVIDUAL SHAPES.

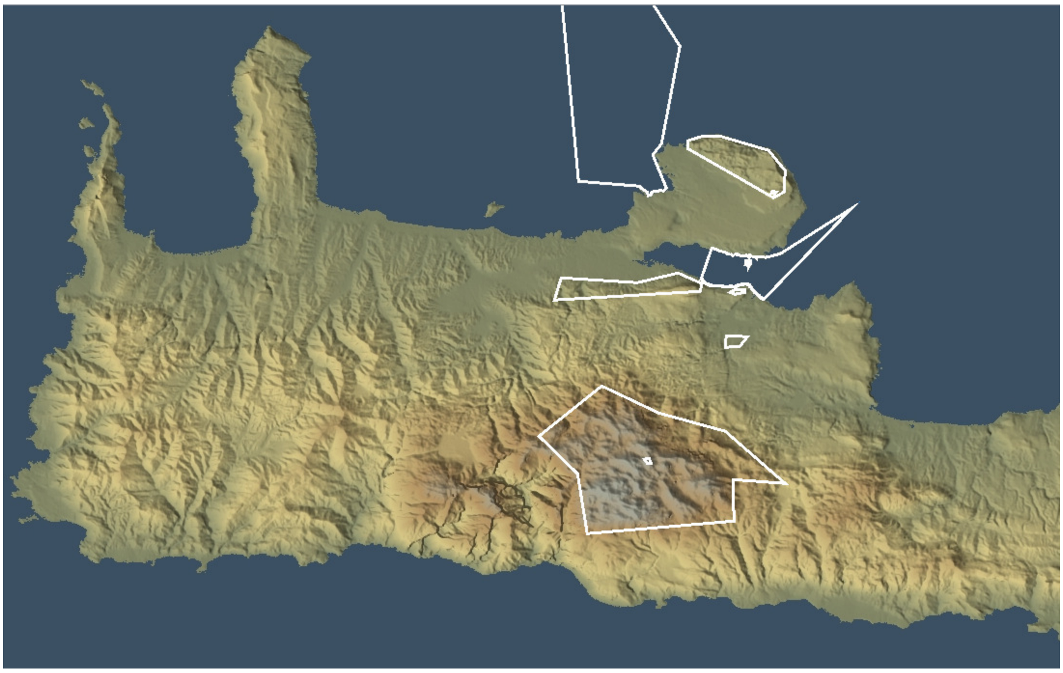


FIGURE 5: THE DEM MAP IMAGE OF FIGURE 3 ANNOTATED WITH SEMANTIC INDIVIDUAL SHAPES. THE INDIVIDUALS IN BOTH THIS FIGURE AND THE PREVIOUS ONE ARE THE SAME AND ARE VISUALIZED ACCORDING TO THE SCALE AND ORIENTATION OF THEIR RESPECTIVE SEMANTIC MAPS.

Photograph Footprints

Except from semantic individuals, photos can also have geographic representations. For example, photos captured using a GPS receiver along with the camera have already embedded location information. If direction information is available, it can be visualized too. The shapes that represent photos and their direction information on top of a map are called photo footprints.

Photo footprints are visualized in the implemented system as semitransparent blue circles. Interacting with one, for example clicking on it with the mouse cursor or selecting it from a list, selects it and turns it into a cyan circle. If direction information is available, then a semitransparent white cone with its tip located at the center of the circle and its base facing the direction of the image appears. The exact heading is shown with a black line. The size of the cone's base depends upon the angle of view for the selected photo footprint. If the angle is wide, the corresponding conic footprint will be larger than when the angle is narrow.

Figure 6 illustrates the described footprints. This is a map of a part of the city of Chania with some individual footprints depicting important parts of the city, colored red. It is assumed that the user has taken some photos with a GPS-enabled camera and has inputted them into the system. The system takes into account each photo's recorded location and places the appropriate photo footprints on top of the map (left part of the figure). When one of them is selected, it changes color and since it also contains compass direction information, the cone footprint of the photo appears, showing that the selected photo was taken with a wide angle (right part of the figure).

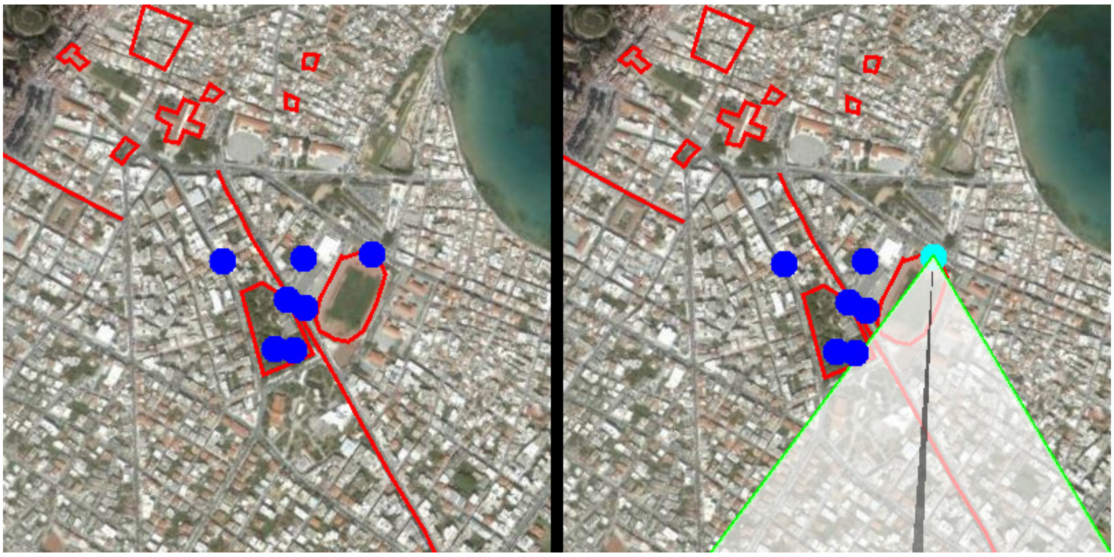


FIGURE 6: VISUALIZING PHOTOS ON TOP OF A MAP. THE SMALL CIRCLES REPRESENT THE LOCATION OF THE PHOTOS. ON THE RIGHT PART OF THE FIGURE, ONE OF THE PHOTOS HAS BEEN SELECTED AND ITS VIEW ANGLE IS VISUALIZED.

The system allows the user to select both a photo and an individual footprint. Individuals are selected in a similar way and change color to green to indicate their status. An additional functionality provided by the system at this point is the ability to determine which photos currently on the map depict the selected individual. Whenever an individual footprint is selected, all the footprints of photos that the system has found to depict the selected individual turn to orange. Figure 7 highlights the action of selecting an individual. The change of color in the appropriate photo footprints is also visible.

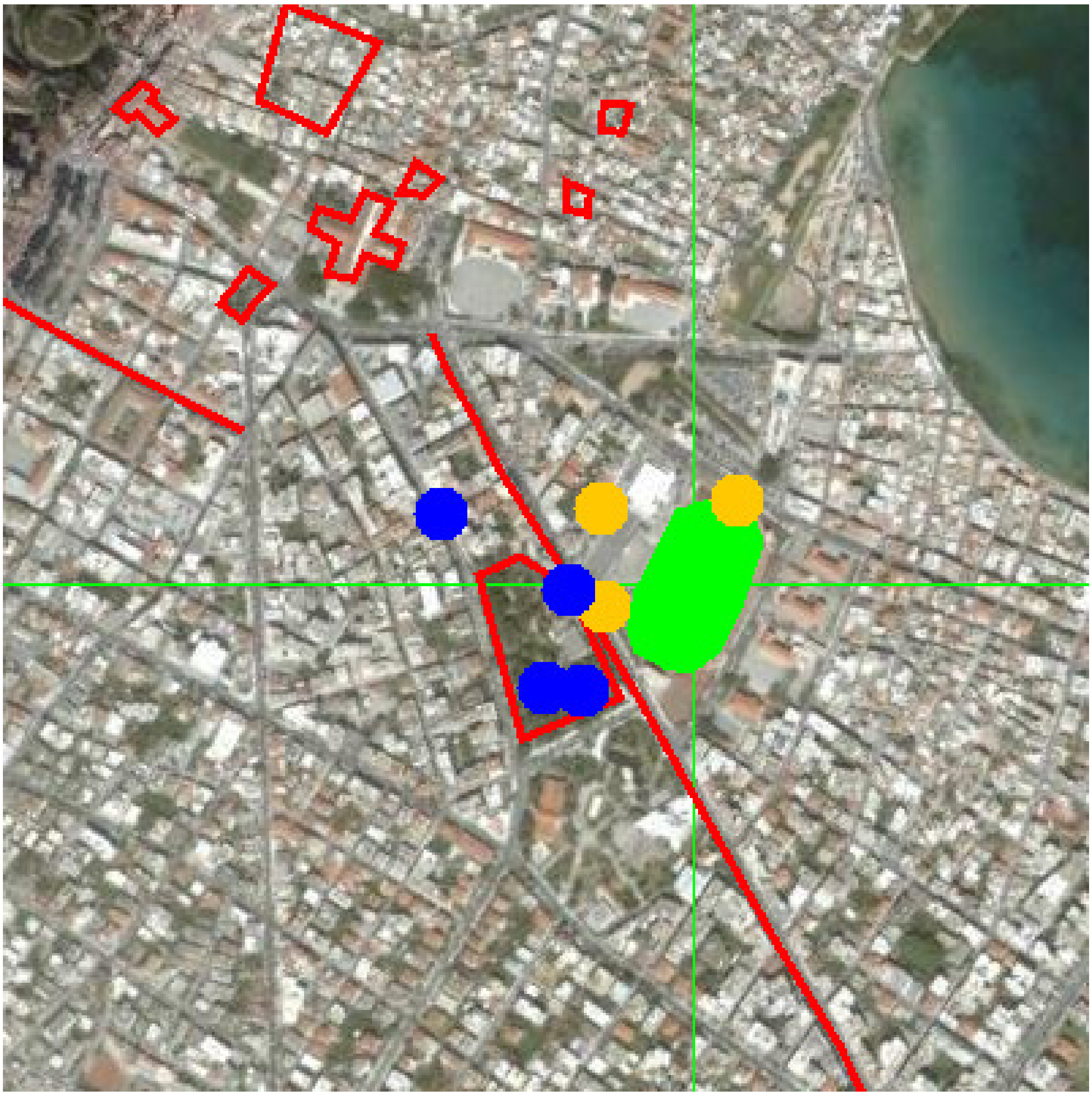


FIGURE 7: VISUALIZING THE PHOTOS OF A SELECTED INDIVIDUAL. THE PHOTO FOOTPRINTS (CIRCLES) OF THE PHOTOS THAT DEPICT THE SELECTED INDIVIDUAL HAVE A BRIGHTER COLOR.

Focus Distance

In chapter 5 it was shown how the minimum and maximum focusing distance for a photo can be calculated. They can be visualized using the cone representations for the photo footprints and the corresponding calculations. Figure 8 shows how these measurements are painted on top of a semantic map. The user has taken two images of the same semantic object from two different positions and the subject distance has been calculated. From this the minimum/maximum focus distance is derived. The area in dark yellow is where objects are too close to be in focus and the area in white is where objects are too far away to be in focus. That leaves the yellow area for the objects in focus. The two photo direction rays intersect at the focused object and a red circle has been placed there for emphasis. The circle is inside the yellow area of both photo footprints, therefore both images have the object in focus.

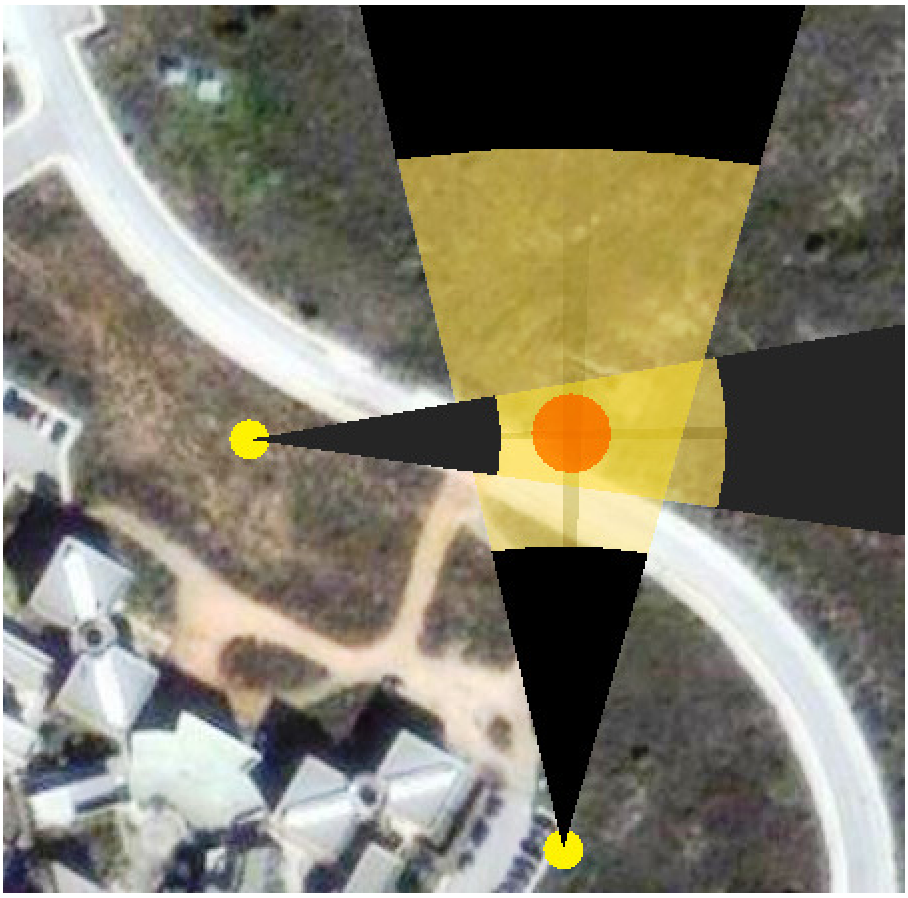


FIGURE 8: MINIMUM/MAXIMUM FOCUS DISTANCE VISUALIZATION. THE TWO PHOTOS FOCUS ON THE SAME OBJECT WHICH IS LOCATED AT THE CENTER OF THE FIGURE. ONLY OBJECTS INSIDE THE NONBLACK AREAS OF EACH PHOTO'S ANGLE OF VIEW ARE CONSIDERED IN FOCUS.

Visible Horizon

When a digital elevation model is available for a semantic map, it enables a variety of powerful algorithms. They will be presented in the next chapter. One of them is the calculation of the visible horizon from the direction of the camera. When taking an outdoors photograph, not everything between the user and the direction of image taking is visible. There are always objects blocking the view – in this case mountainous terrain. The algorithm calculates the visible areas and records their position on top of the map. Figure 9 illustrates the result of the algorithm. The visible areas on top of the map are painted blue. Near the user everything seems to be visible, then visibility is obstructed by mountains; however there are higher mountains in the distance, so they are also visible.

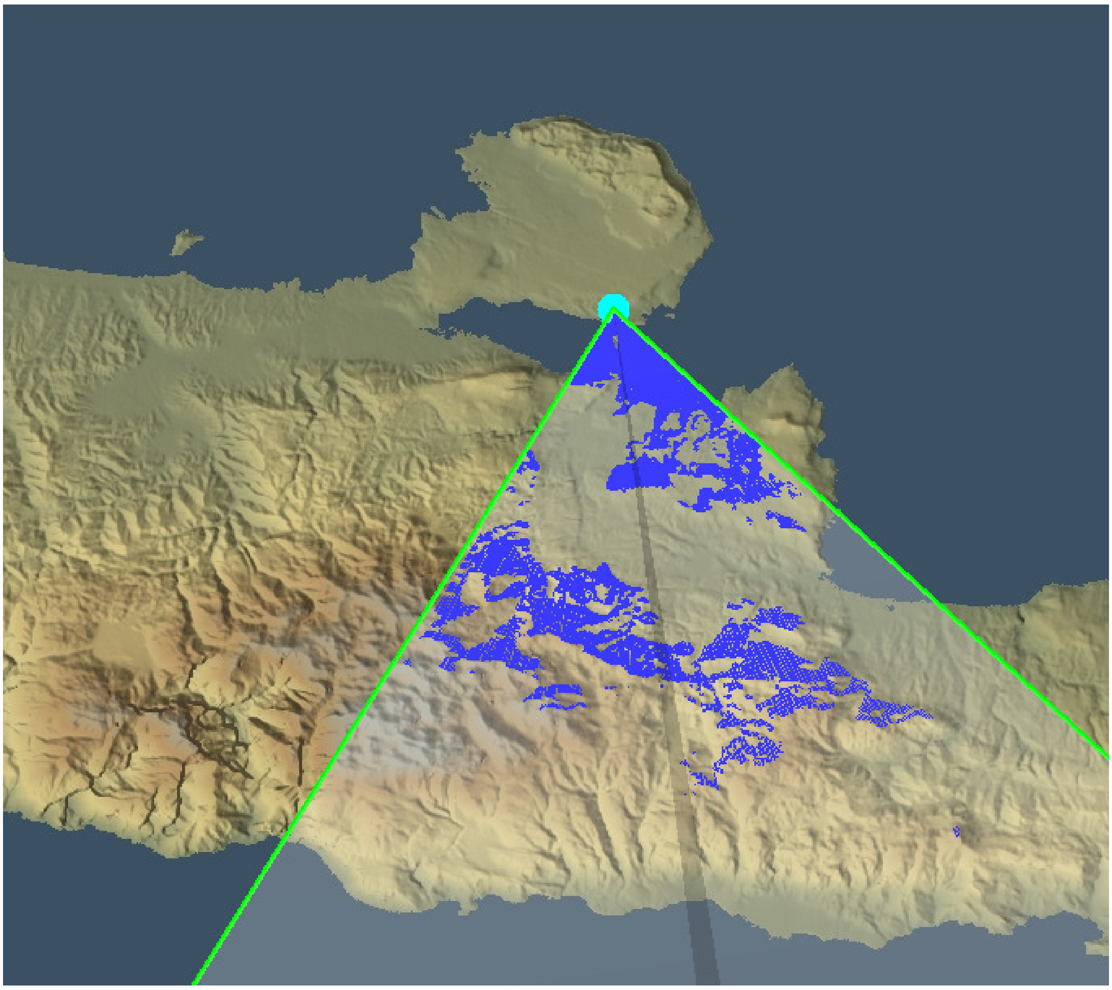


FIGURE 9: VISIBLE HORIZON VISUALIZATION. THE DARK AREAS INSIDE THE ANGLE OF VIEW REPRESENT THE VISIBLE LANDSCAPE AREAS ALONG THE DIRECTION OF THE CAMERA.

6.5. Summary

This chapter analyzed the concept of the semantic map, providing information about how they are created and what entities can be correlated with and visualized on them. The ontological model used in this work was described with examples. Finally, a number of visualizations offered by the implemented system were shown.

7. Photo to Semantic Map Transformation

This chapter analyzes the procedure for converting an image into a semantic map, containing interactive semantic individuals along with their attributes. Section 7.1 describes the general methodology for the procedure and sections 7.2 to 7.4 analyze the steps of the procedure. In section 7.5, the results of the transformation are illustrated. Finally, section 7.6 summarizes the chapter.

7.1. Methodology

This section describes the general methodology for the photo to semantic map transformation. This method works for outdoor images where at least a big part of the skyline is distinguishable and the sky does not contain major obstructions or noise. The procedure is divided into three parts:

- Viewshed calculation
- Image segmentation
- Image registration

The first part utilizes the elevation data and the knowledge of a photo's position and direction parameters in order to determine the visible areas and semantic objects from the spatial view of the image. Without the use of tilt and rotation sensors, it is assumed that the camera is held at a horizontal position without any rotation or tilt. The result of the algorithm is a 2D representation (model) of the landscape and the individuals from the point of view and the direction of the image. The skyline and the boundary for any individuals representing the sea are easily extracted from the model, along with any distinguishable mountain peaks on the skyline.

The second part uses a general segmentation algorithm and segments the image into regions. Then, additional heuristic algorithms are employed in order to locate the regions and boundaries that correspond to specific geographic individuals and features. The implemented system detects the blue sky, the sea and the location of specific individuals such as mountain peaks. Boundaries for the sea and the skyline are also computed.

The final part of the method combines the results from the first and second part. The algorithm divides the points representing the skyline and boundaries from both the image and the 2D model into line segments. Then the algorithm finds the optimal match of the line segments from the model and the image and derives the transformation with which the model must be transformed to successfully register with the image. In the implemented system, the location of mountain peaks in both the model and the image also helps the calculations. Overlaying the transformed model on top of the image places the model's individuals at their corresponding position in the image. The user can select them and view the associated information, just like interacting with a geographic semantic map.

7.2. The Viewshed

The viewshed is defined as an area of land, water, or other environmental element that is visible to the human eye from a fixed vantage point. In this case the human eye is the camera lens and the visible area is only calculated for a specific direction and angle. The following subsections show how the viewshed is computed and how it is visualized for the purposes of this thesis.

7.2.1. Viewshed Calculation

The algorithm presented in this subsection utilizes the elevation data provided by the SRTM [66] to determine the visible horizon. It is a modified version of the direct method for visible area computation presented in Franklin et al. [18]. The major differences are that the algorithm also takes into account the location and height of semantic objects encountered and that the viewshed is computed for the angle of view of the camera and not for 360°.

The algorithm traces rays that begin from the user, follow the direction of the camera and are within the angle of view. At regular intervals, it is computed if the current GPS position is visible from the point of view and if a semantic object is visible at this position. Whenever visibility becomes broken following a visible area, a new visible peak point (not to be confused with mountain peaks) is found and recorded. The visibility information is stored in appropriate data structures and can be used later for making the 2D model of the camera view.

The algorithm is described below using Java-like pseudo code. It takes as input the headings corresponding to the start and end of the angle of view, the elevation and GPS position of the point where the user is standing, the number of rays to be cast and the maximum distance for each ray to search for visible areas. The output is a vector containing all the information about each ray cast.

Each rays starts from the user's position and follows the direction defined by 'CurrentAngle'. The distance covered with each step is variable. Near the user, more detail about the visible areas and individuals is required and the step is small. When the ray has moved farther away, the steps are larger. For example, the step starts at 2 meter intervals, which is then increased gradually, up to 20 meters when the distance becomes more than 20 kilometers. The value of the step is given by the **getKMStep** function. Then, a great circle calculator (**getNextGPSPoint**) provides the next GPS position according to the distance and direction from the point of view. The elevation, the individuals located and the height of this point are computed (**getElevation**, **getFootprintsAtPoint**). The tangent of the angle defined by the horizontal line parallel to the level of the user and the line that connects the user and the current position is calculated. This is essentially the view angle and its tangent is later used for the construction of the 2D view model. Figure 10 shows an example of ray tracing for two different points *p1* and *p2* viewed by an observer *O*. The point *p1* is *d1* meters away from the observer, has an elevation of *h1* meters above the observer's elevation and is viewed from an angle *a1*. Respectively, the same applies to point *p2* with the exception that it is not visible to the observer. For these two points, the tangent is calculated as follows:

$$\tan(a1) = \frac{h1}{d1}, \qquad \tan(a2) = \frac{h2}{d2}$$

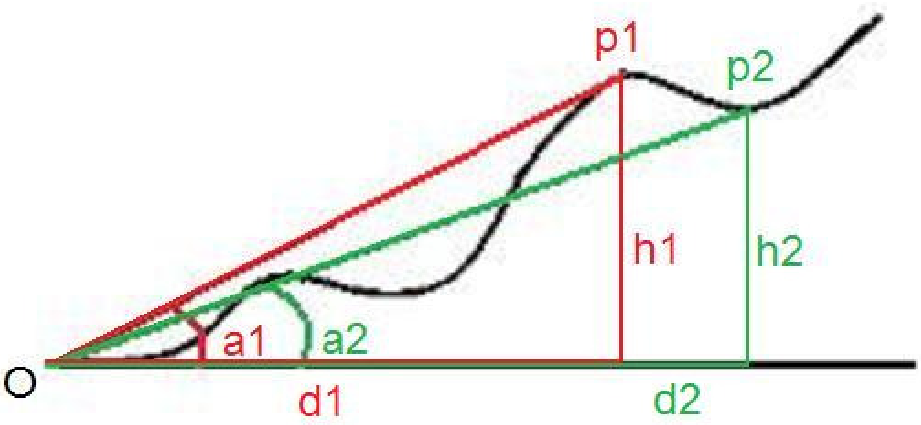


FIGURE 10: VISIBLE POINTS FROM AN OBSERVER O. POINT P1 IS LOCATED D1 METERS AWAY FROM THE OBSERVER, IS H1 METERS ABOVE THE OBSERVER'S ELEVATION AND IS VIEWED FROM AN ANGLE A1. THE SAME APPLIES TO POINT P2 RESPECTIVELY; HOWEVER P2 IS NOT VISIBLE TO THE OBSERVER.

To determine if a point is visible to the observer or not, the *Line of Sight (LOS)* function is used, the return value of which is then compared to the point's height. The details of the derivation of the *LOS* function are in [28]. The formula is:

$$LOS = \max(PointHeight, UserHeight + \frac{Distance}{PrevDistance} \cdot (PrevLOS - UserHeight)$$

Right after computing the *LOS* (**getLineOfSight**), the algorithm compares its value to the current point's height. If *LOS* is greater or equal to the current point's height, then the point is considered visible. In this case, information about the visible individuals at this point and the view tangent calculated beforehand are stored to the current ray's data structures for later use. The visibility algorithm keeps track of whether the ray is ascending or descending, for example at point *p1* in Figure 10, the ray has been going up (ascending) and is about to descend, therefore a visible peak (*p1*) has been found. This is discovered by the algorithm and the view tangent of the visible peak point is stored to the current ray's data structures. After this step, the ray is added to the vector of rays that is the algorithm's output. When all the rays have been cast, the algorithm ends.

Algorithm 7.1. – Viewshed Calculation

```
Input:
Integer StartAngle, EndAngle, UserHeight, NumOfRays, KMLimit
GPSPoint OriginPoint

Output:
Vector<Ray> Rays

Begin
//Local variable declarations
GPSPoint CurrentPoint = null;
Integer CurrentAngle = StartAngle, CurrentHeight = -1, CurrentHeading =
StartHeading, CurrentIteration = 0;
Ray CurrentRay = new Ray();
Boolean Descending = False;
Double CurrentDistance = 0.0, PreviousDistance = 0.0, CurrentTan = 0.0, LOS = 0.0,
AngleStep = (StartAngle - EndAngle) / NumOfRays;
Vector Footprints = new Vector();

//for each ray
while(CurrentIteration < NumOfRays){
  CurrentIteration++;
  //for each GPS point located "CurrentDistance" kilometers away from
  //the camera up to KMLimit kilometers away
  while(CurrentDistance < KMLimit){
    //calculates the distance to the next position in the ray
    CurrentDistance += getKMStep(CurrentDistance);
    //moves to the next GPS position following the ray, according to the current
    //angle of the ray and the distance from the camera
    CurrentPoint = getNextGPSPoint(OriginPoint, CurrentAngle, CurrentDistance);
    //finds the elevation at the current position and current distance, taking also
    //into account the curvature of the Earth
    CurrentHeight = getElevation(CurrentPoint, CurrentDistance);
    //finds which individual footprints are located in the current position
    Footprints = getFootprintsAtPoint(CurrentPoint);
    //returns the height of the highest footprint in the current position to compute
    //the height of the current position
    CurrentHeight += getHighestFootprintHeight(Footprints);
    //calculates the tangent of the angle the camera views the current position
    //(CurrentDistance is converted to meters)
    CurrentTan = (CurrentHeight - UserHeight) / (CurrentDistance*1000);
    //calculates the line of sight function
    LOS = getLineOfSight(LOS, UserHeight, CurrentDistance, PreviousDistance,
CurrentHeight);
    if(LOS ≤ CurrentHeight){ //the current point is visible
      //finds which footprints are visible and adds their information to the ray
      CurrentRay.findAndAddVisibleFootprints(Footprints);
      if(Descending == True){ //The ray was descending, but now it is ascending
        Descending = False;
      } //end if
    else{ //(LOS > CurrentHeight) – the current point is not visible
      //if the ray was ascending but now isn't, a visible peak has been found
      if(Descending == False){
        //stores the tangent of the current view (used for creating the 2D model)
        CurrentRay.addVisiblePeak(CurrentTan);
        Descending = True;
      } //end if
    } //end else
    //stores the previous distance for use in the next step
    PreviousDistance = CurrentDistance;
  } //end of inner while
  //proceeds to the next angle for the next ray
  CurrentAngle += AngleStep;
  CurrentDistance = PreviousDistance = 0.0;
  Descending = False;
  Rays.add(CurrentRay);
}
End
```

7.2.2. The 2D View Model of the Picture

Having completed the viewshed calculation, the implemented algorithm has collected all the needed information to create the 2D model of the picture. Apart from painting the model with the appropriate colors for the visible peaks and individuals, the algorithm also collects information about the individuals, extracts the skyline and the sea boundary (if present) and detects any distinguishable mountain peaks on the skyline. The whole procedure is divided into three steps.

The first step is detecting the skyline and any mountain peaks modeled as individuals. The skyline is defined as the set of visible points with the highest view tangent for all the rays. Naturally, this is susceptible to noise in the calculations. Points that break the smoothness of the skyline are discarded at a later step. Mountain peaks are detected by following skyline points belonging to an individual that represents a mountain peak and determining if it has the characteristics (sharpness) of a mountain top.

The second step is painting the visible peak points. The "canvas" is a white image with width equal to the number of rays cast for the viewshed calculation and height proportional to the vertical angle of view of the image. For each visible peak found in a ray, the view tangent has been calculated (see Figure 10 for clarification – the corresponding tangents are *a1* and *a2*, while point *p1* is considered a visible peak point). With the lack of tilt and rotation sensors, it is assumed that the camera is held at a horizontal position without any rotation or tilt. Therefore, the camera direction falls upon the central pixel of the "canvas" (if the resolution of the model image is (x, y), then the central pixel is located in (x/2, y/2)) and subsequently, all objects with positive/negative view tangent are located above/below that pixel. Since the width of the "canvas" is equal to the number of rays cast, each column of pixels corresponds to one ray. Figure 11 shows an image and its 2D model created with the algorithm, without the individuals in the image. The part of the model corresponding to the sea is flat, except at the point where the two small islands are located. The algorithm successfully detected that there were parts of the sea obstructed by the islands and painted them as visible "peaks".

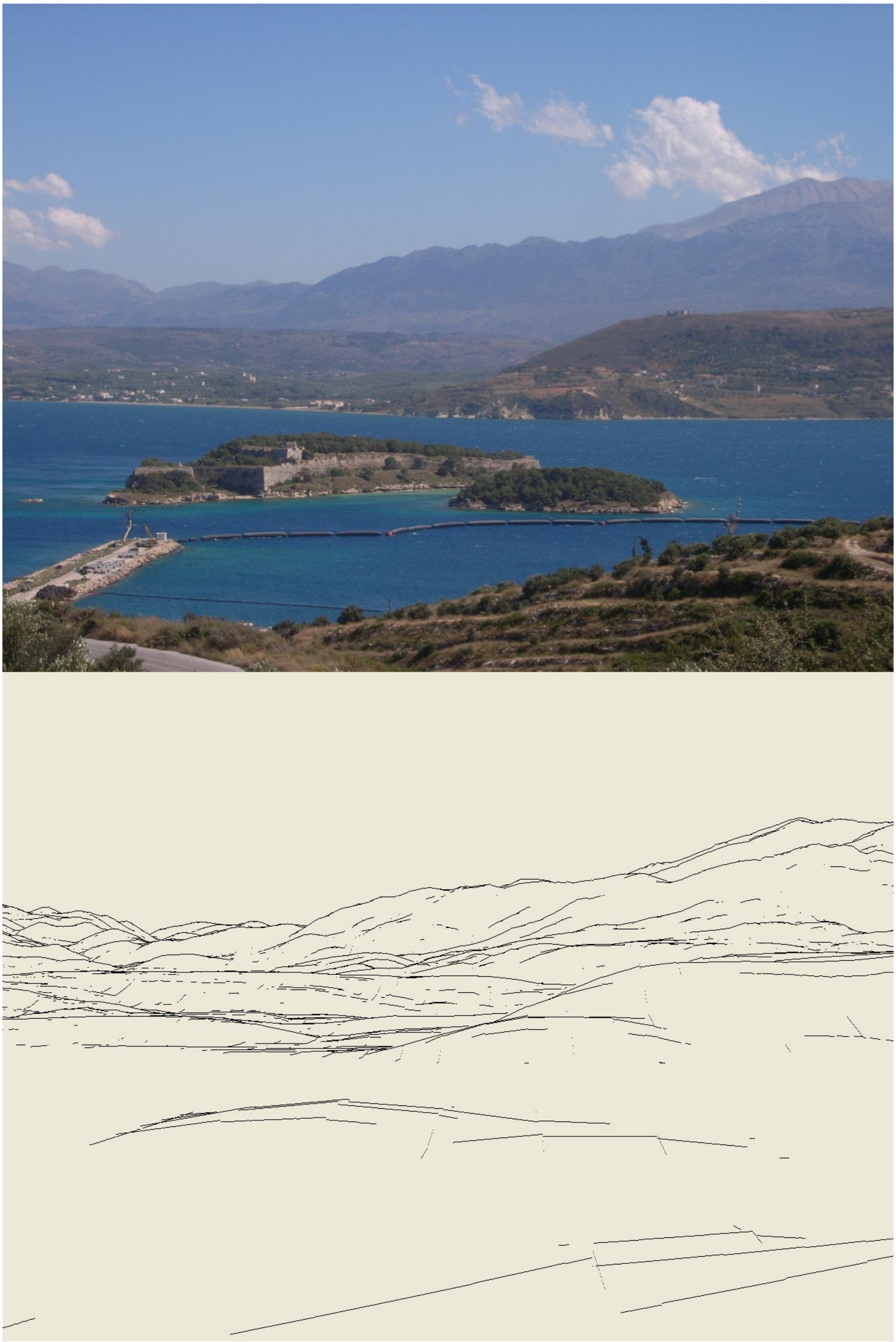


FIGURE 11: AN IMAGE AND ITS 2D VIEW MODEL. THE MODEL CONTAINS JUST THE LAND DISCONTINUITIES CALCULATED FROM THE ELEVATION DATA.

The third and final step is painting the semantic individuals and detecting the sea boundary. The view tangent for the individuals has been calculated and they are painted with their corresponding color on the model. If an individual representing sea is found, its topmost boundary is stored for later matching with the corresponding sea boundary of the image. The complete 2D model of the image in Figure 11 is shown in Figure 12. The sky has been manually painted to provide contrast for this illustration. This model contains five individuals: the sea (blue), two islands (brown and purple), a large mountainous area (white) and a site of archaeological interest (gray). The model's resolution is the same as that of the image (1280 x 960).

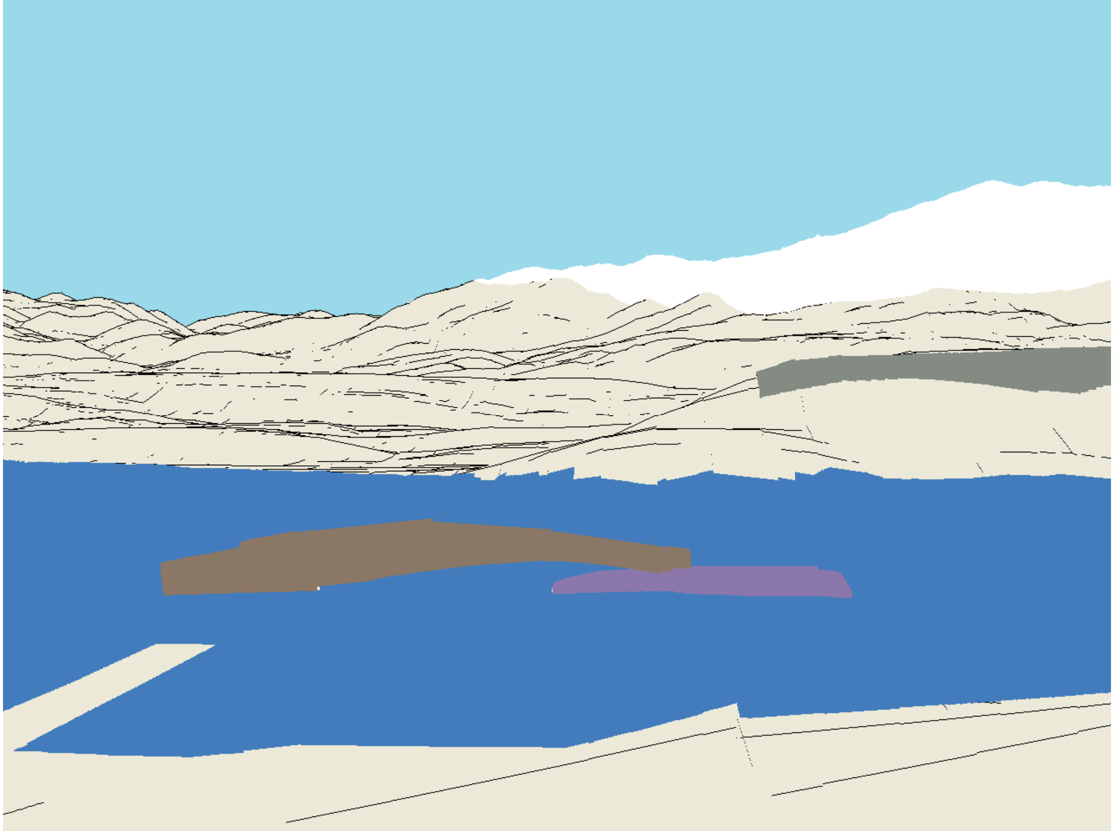


FIGURE 12: THE COMPLETE 2D MODEL OF THE PHOTO IN FIGURE 11. IT CONTAINS BOTH LAND DISCONTINUITIES AND THE LOCATIONS/SHAPES OF THE SEMANTIC OBJECTS THAT ARE VISIBLE ALONG THE DIRECTION OF THE CAMERA.

7.3. Image Segmentation

Image segmentation forms the second part of the photo to semantic map transformation. Its purpose is to facilitate the detection of geographic features on the picture to be used in the registration algorithm. The image is segmented into regions, using the statistical region merging method by Nock and Nielsen [51]. The authors provide a simple but effective region growing method that can be parameterized to define the limits of how easily regions are merged. Figure 13 shows an example of image segmentation using the method on the image from Figure 11. The segmentation is performed on a scaled version of the original image to make the computation faster and avoid many artifacts caused by a high resolution segmented image.

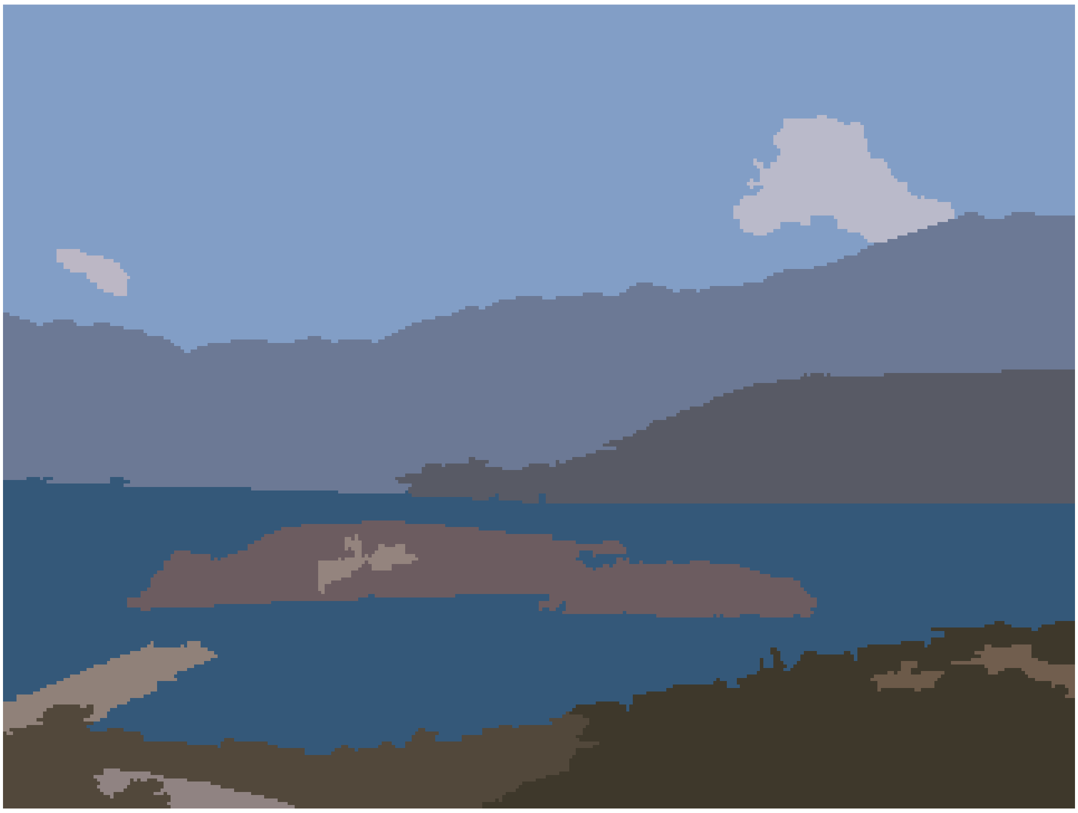


FIGURE 13: LOW RESOLUTION IMAGE SEGMENTATION EXAMPLE. THE PHOTO FROM FIGURE 11 HAS BEEN SEGMENTED INTO REGIONS. THE SKY AND SEA REGIONS CAN BE DISTINGUISHED. THE CURRENTLY DETECTED SKY REGION NEEDS ADDITIONAL PROCESSING TO REMOVE THE OBSTACLES (CLOUDS) CONTAINED IN IT.

After the segmentation, various algorithms are applied in order to successfully detect the skyline (section 7.3.1) and sea boundaries (section 7.3.2) present in the picture.

7.3.1. Skyline Detection

The system has already got the points that comprise the skyline from the 2D view model and the next step is to detect the corresponding skyline points from the segmented image. The objective is to collect the most distinguishable characteristics from the image and the model so that the registration between model and image is as accurate as possible. The skyline is one such characteristic. Even though in the general case it is assumed that the sky should just be the topmost region of the image, it is not always so. Sometimes there are two or more regions that belong to the sky or there are small obstacles (for example clouds) in the way or other major obstructions. The algorithm presented here (called *SkyDetector* from now on) utilizes heuristics and focuses on the first two problems. In order to eliminate the problem of multiple sky regions and obstacles, *SkyDetector* collects statistics about the problematic regions.

Initially, the first sky region is discovered and established by checking the topmost row of pixels and detecting the best candidate sky region. *SkyDetector* then scans the segmented image from the top, one column of pixels at a time, and detects regions where the sky region is interrupted during the scan and then found again below them in the segmented image. These regions have a higher probability of being obstacles such as clouds and are merged with the sky region if they interrupt the scanning too much. For determining which regions should be merged with the sky region, some statistics are collected. These include the smoothness[4] of the candidate obstacle region's top side, the number of columns (of pixels) where it interrupts the sky region and the number of pixels that belong to the sky region and are below the obstacle region. Whenever a new region is merged with the sky region, the algorithm scans the image from the beginning. It stops only when there are no candidate obstacle regions detected during a complete scan of the image. For an example on how *SkyDetector* works, see Figure 13 above. In it, there are two regions representing clouds. The left cloud is completely inside the sky region, while the right one is partially inside. The algorithm scans each column of pixels from top to bottom and finds out that when these clouds are encountered, the sky region is then found again below them. The left cloud is smooth but is completely within the sky region, so it is merged automatically. The right cloud is not very smooth and has a high number of sky pixels below it (spanning for many scanned pixel columns), so it is also detected as an obstacle region.

Extracting the skyline points from the resulting sky region is as simple as taking the lowest pixel of each pixel column that belongs to the sky region. As with the 2D model skyline case, points that break the smoothness of the skyline are discarded at a later step. Figure 14 and Figure 15 show the effect of the sky region detecting algorithm. In the first figure, the many variable colors of the sky are divided into several regions. *SkyDetector* determines that these regions all belong to the sky region and merges them, resulting in the segmentation result of the second figure. The images are in higher resolution for enhancing the effect; in practice lower resolutions are used for the segmentation.

[4] The smoothness of a region is calculated by computing the discrete derivatives of the topmost pixels of the pixel columns that belong to the region. Smooth regions have very low derivative values.

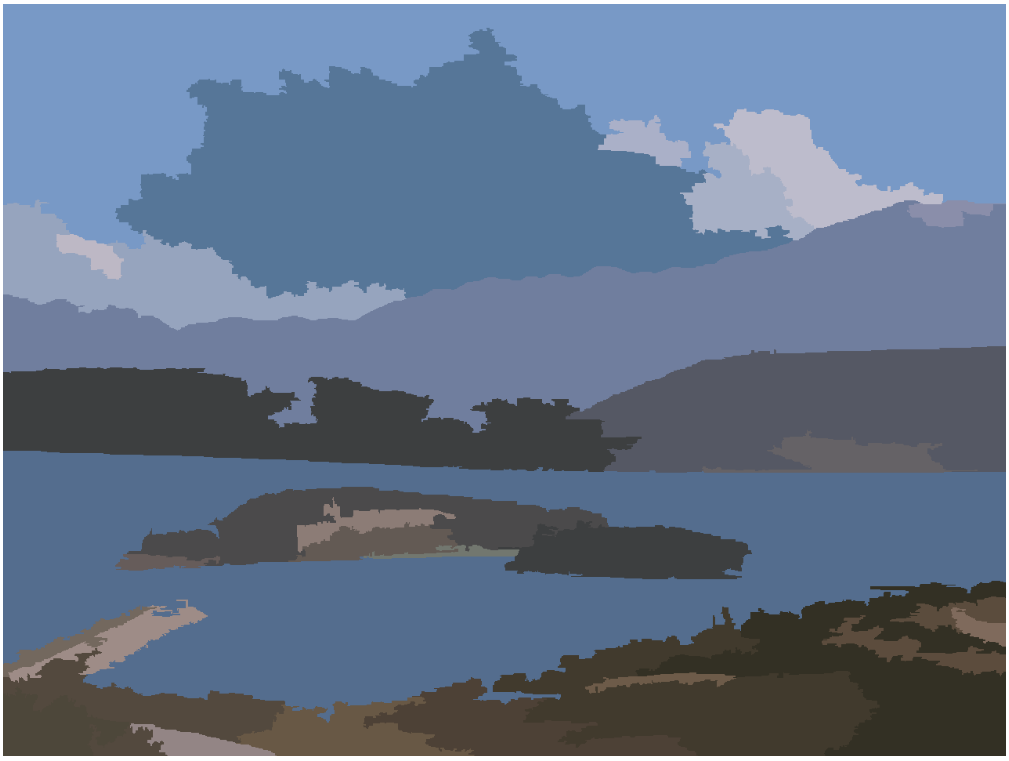


FIGURE 14: HIGH RESOLUTION IMAGE SEGMENTATION WITH MULTIPLE SKY REGIONS. INITIALLY, THE SEGMENTATION ALGORITHM DETECTS MANY REGIONS THAT BELONG TO THE SKY. THEY NEED TO BE MERGED TO PRODUCE A SINGLE SKY REGION.

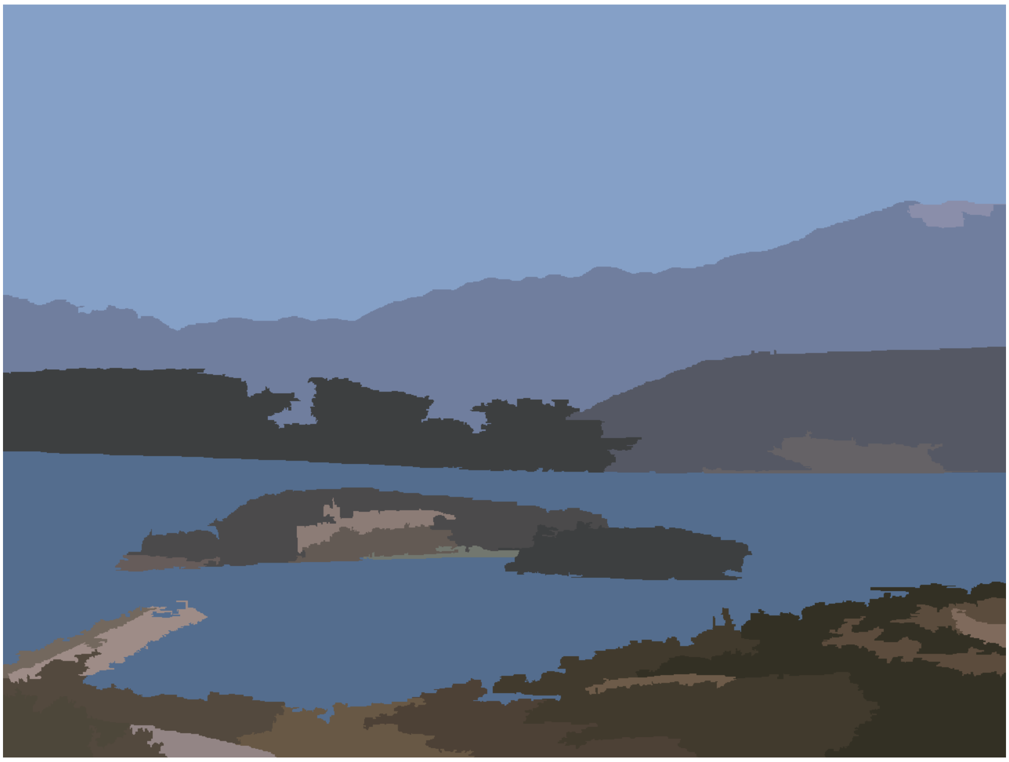


FIGURE 15: HIGH RESOLUTION IMAGE SEGMENTATION WITH THE SINGLE DETECTED SKY REGION. THE ALGORITHM HAS MERGED THE REGIONS THAT WERE NOT SMOOTH, CREATING A SINGLE REGION FOR THE SKY. THE SKYLINE CAN THEN BE EASILY EXTRACTED.

7.3.2. Geographic Individual Detection

This subsection describes the last step of the segmentation algorithm – the detection of specific geographic individuals. The 2D model has already provided the sea boundary points and detected mountain peak individuals. The same entities should also be present and detected in the segmentation image.

The heuristic algorithm for extracting the sea region (called *SeaDetector*) scans all the regions and filters them according to two criteria, *BluePrevalence* and *RedGreenDifference*. *BluePrevalence* filters out all regions in which the blue component[5] of the color model prevails over both the red and green components over a determined threshold value. This procedure eliminates the regions that obviously don't have the common sea color. However, there are also regions that have a strong blue component and are not sea regions. The job of the *RedGreenDifference* criterion is to filter out these regions. It has been found experimentally that the common color of the sea has the following characteristic: the green component is always stronger than the red one and usually their difference is great. *SeaDetector* takes that information into account and removes the regions that do not have a strong green component over the red one, leaving the regions that truly represent the sea. They are then merged accordingly.

The next algorithm (called *PeakDetector*) detects mountain peaks by scanning the skyline points (using their pixel coordinates) found by *SkyDetector* from left to right and searching for parts that quickly ascend up to a point and then start to quickly descend. The respective slopes are calculated and if they are large enough, the peak point is located. Figure 16 shows an example of how this works. By looking at the segmentation image, there are two distinguishable areas candidate for being mountain peaks. They have been (manually) highlighted with white rectangles. *PeakDetector* computes the ascending slope for the first area and finds it satisfactory; however after ascending it does not descent quickly enough to create a mountain peak. In the second area there is no such problem and *PeakDetector* will successfully determine that there is a mountain peak at this area of the skyline. The exact peak (pixel) point is found and marked. It may be used for providing a better registration result later.

[5] The RGB color model is used, in which every color can be represented by a mixture of (R)ed, (G)reen and (B)lue values.

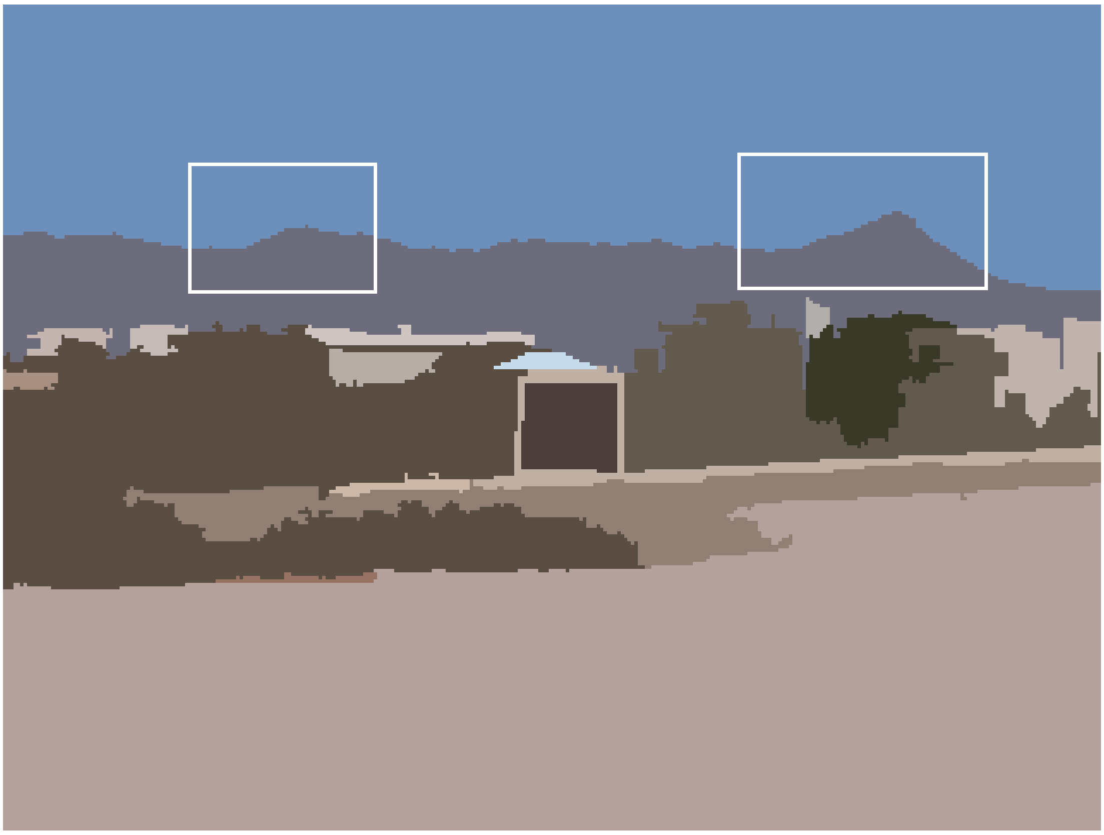


FIGURE 16: MOUNTAIN PEAK DETECTION EXAMPLE. THE TWO RECTANGLES INDICATE POSSIBLE MOUNTAIN PEAK AREAS. THE LEFT ONE IS NOT "POINTY" ENOUGH TO BE CONSIDERED A PEAK. THE RIGHT ONE IS SUCCESSFULLY DETECTED AS A POSSIBLE MOUNTAIN PEAK.

7.4. Image Registration

The third and final part of the photo to semantic map transformation consists of the matching algorithm for image registration and its image direction correction capability. The matching algorithm (called *LineSegmentMatcher*) approximates the extracted boundaries (skyline and sea) from both the photo and the 2D model with line segments. Then it tries to find the optimal match between the approximated line segments. The optimal match provides the best fit transformation for transforming the 2D model to fit the photo. *LineSegmentMatcher* is called several times with slightly different parameters and the best result is kept. The matching method is "Steepest Descent Local Search", as described in Beveridge and Riseman [5]. Their method can be used for matching any 2D object but has also been successfully used specifically for detecting skylines before [4] and that's the reason it was chosen. Subsection 7.4.1 deals with the details of *LineSegmentMatcher*, while subsection 7.4.2 describes how it is used for getting the best match.

7.4.1. The Line Segment Matching Algorithm

The 2D view model construction algorithm and the segmentation algorithms provide skyline and sea boundary points as sets of pixel points. The matching algorithm functions using line segments, therefore the point sets must be approximated as line segments. From now on, the line segments corresponding to the point sets of the 2D model will be called model line segments, while the line segments corresponding to the point sets from the segmentation algorithm will be called data line segments. The method used for approximating point sets as line segments is based on the method proposed by Mohamed Ali Said [61]. It works by computing the upper or lower convex hull of the point set as an initial approximation and computes an error function that compares the perpendicular distance of the points from the line segments that approximate them. If the distance is greater than the specified tolerance, the alternate convex hull is used for further approximation of the points. Figure 17 shows an example of how the algorithm works. First, the upper convex hull is computed between the first and last point, as shown in **B** and provides a rough approximation. Since most of the points are distant from the approximated line segments, the alternate convex hull needs to be computed for the points between these line segments. The result is shown in **C**. The algorithm continues calculating the alternate convex hulls until all the points are close enough to their corresponding approximated line segments, according to the error tolerance value. The upper convex hull for the first three points is computed in **D** for a more detailed approximation. This is only an example; in reality the approximations should produce line segments that are long enough for the registration algorithm to work efficiently.

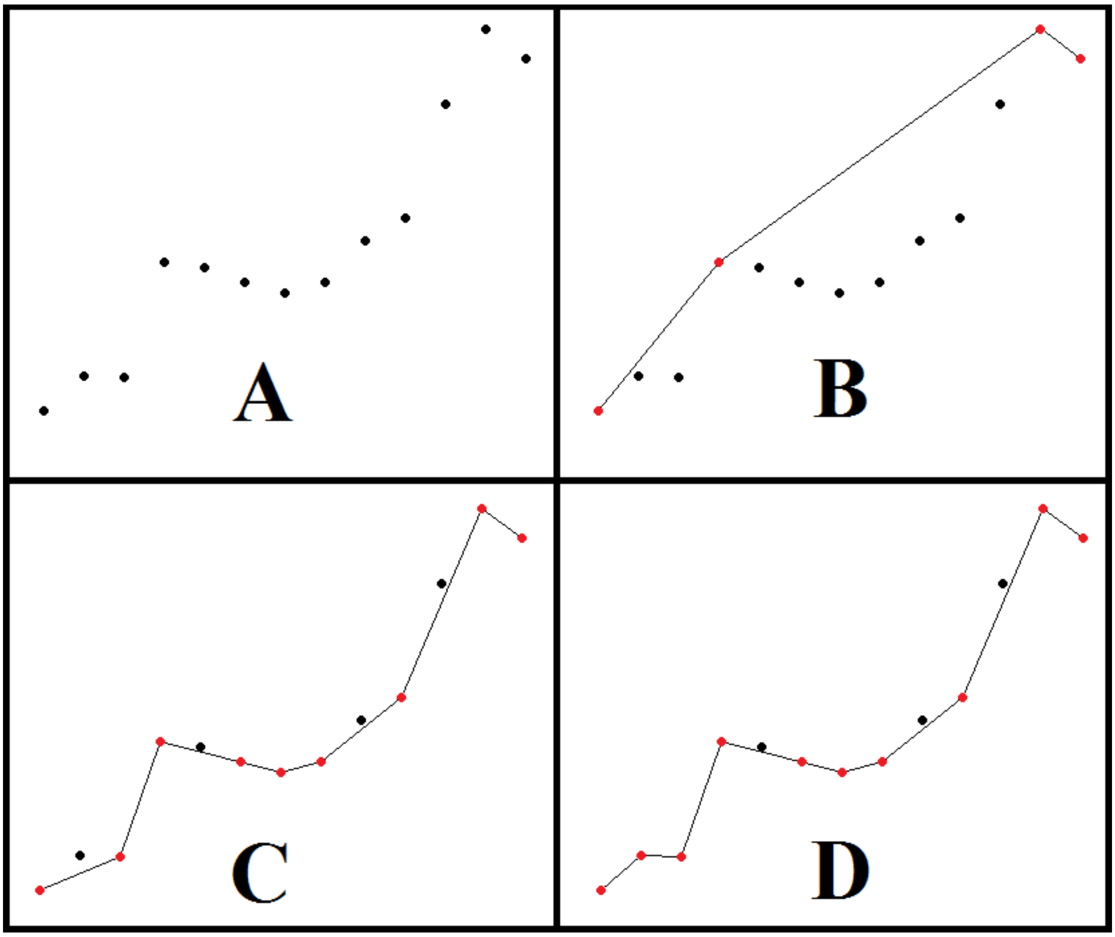


FIGURE 17: LINE SEGMENT APPROXIMATION EXAMPLE. A)POINT SET. B)APPROXIMATION BASED ON COMPUTING THE UPPER CONVEX HULL. C)FURTHER APPROXIMATION AFTER COMPUTING THE LOWER CONVEX HULL. D) FINAL APPROXIMATION USING THE UPPER CONVEX HULL AGAIN FOR THE 3 FIRST POINTS.

After the approximation, an initial estimation of the matches between the model line segments and the data line segments along with the total error for this estimate is computed. This estimation is composed of a collection (set) of matched line segment pairs with each pair containing one model line segment and one data line segment. *LineSegmentMatcher* is now ready to begin the iterations. In each iteration, every possible line segment pair is either added to (if not in the current estimation) or removed from (if it's in the estimation already) the matched set temporarily and one at a time and the total error is again calculated for each case independently. The addition or removal of the line segment pair that produces the least total error is permanently chosen for the best match for this iteration and is added or removed accordingly. The algorithm then moves to the next iteration and performs the same steps using the new set of line segment pair matches, until a constant number of iterations has passed or the algorithm converges, which means that the total error cannot be decreased further.

The total error is composed of the following:

- **Fit error**: Measures how well the model line segments fit the data line segments. This error metric (*E_P*) measures and adds the perpendicular distance between each chosen line segment pair. The greater the distance, the greater the fit error becomes. Its value is the following:

$$E_F = \left(\frac{1}{L_D}\right)\sum_{i=1}^{h}\frac{\ell_i}{2}(\upsilon_{i1}^2 + \upsilon_{i2}^2)$$

  L_D is the cumulative length over all matched data line segments, ℓ_i is the length of the *i*-th data line segment and *h* is the number of matched pairs. $\upsilon_{i1}/\upsilon_{i2}$ are the perpendicular distances from the endpoints 1 and 2 of a data line segment *i* to the corresponding infinitely extended[6] model line segment *i*. The derivation and details of this error metric are provided in Beveridge and Riseman [5]. The authors alternatively used the integrated squared perpendicular distance, which is not just endpoint based but takes into account the whole line segment. This distance was tried by *LineSegmentMatcher* with no success and the endpoint based distance was used instead. The authors also defined the omission error which penalizes matches where few model line segments participate in the final match. The omission error was not used for *LineSegmentMatcher* because the model line segments are not necessarily all correct and in some cases some of them need to be excluded from the final match without penalty.
- **Scale error**: This error discourages transformations that have a very large or very small scale factor *s*, since the 2D model has similar scale to the photo.

$$E_S = \begin{cases} \frac{1}{s} - r, & s < 1/r \\ 0.0, & 1/r \le s \le r \\ s - r, & s > r \end{cases}$$

  The error function allows for a parameter *r* that allows a specific range of scale values to not produce any error at all. If the scale change grows or shrinks beyond *r*, then the error grows in proportion to the relative change in scale.
- **Rotation error**: This error discourages unlikely rotations for the match. The higher the rotation, the greater the error. This metric also allows a small range of rotation values to produce small error values:

$$E_R = \begin{cases} 0.1 * \vartheta, & |\vartheta| < r \\ 10 * \vartheta, & |\vartheta| \ge r \end{cases}$$

[6] This extension is necessary in order to calculate the perpendicular distance.

The parameter *r* and the angle ϑ are in degrees. The scale error is defined as the transformation error in Beveridge and Riseman [5], while the rotation error has been introduced here.

Steepest descent local search iterates having as guides the above error metrics. One disadvantage of this method is that it can converge in local optima instead of the best fit. The fit error defines how the search finds the best fit, while scale and rotation errors help the algorithm avoid falling into local optima. When the iterations complete, a final fit transformation has been computed. It provides values for translating, rotating and scaling the 2D model to fit correctly on the photo.

7.4.2. Image Direction Correction

To improve the results of *LineSegmentMatcher*, the system follows a number of steps based on the facts below:

- The compass direction measurements are not very accurate and are usually off by several degrees, resulting in 2D model calculations that need to be translated considerably.
- The detected mountain peaks from both the 2D model and the photo can guide the registration process towards a better match.

To take advantage of the first fact, the system "trims" a portion of the 2D model point set. For example only the central 75% of points are kept for the matching (12.5% from the left and 12.5% from the right are discarded); therefore bad matches that would be produced if the compass direction was off by a few degrees are eliminated. To facilitate the matching even further, *LineSegmentMatcher* is called many times for different parts of the photo point sets. Continuing from the previous example, *LineSegmentMatcher* would be called 26 times on the 75% of the photo point set. The points in the photo point set are discarded differently each time. The first time it would "trim" 25% of the points from the right side and 0% from the left. The second time, it would "trim" 1% of the points from the left side and 24% of the points from the right side and so on.

LineSegmentMatcher is called both with and without using the sea boundaries. The match error for these calls is calculated and the lowest value is kept. The skyline and sea boundary (if used in the match) of the 2D model are transformed using the fit transformation returned by *LineSegmentMatcher* and the average vertical pixel distance between the corresponding skylines and boundaries (if used) is computed. This is used as a measure for determining the best call of *LineSegmentMatcher* and consequently the best fit transformation.

If the match error is too high, it means that either the skyline and sea boundary has many errors and cannot be matched with precision or the compass measurement is way off. The system makes a second try, this time using a smaller subset of the points in case the errors in the point sets are located in the discarded subset or the compass deviation does not affect the remaining subset. For example, the 2D model point set is reduced to only the central 50% of points, while only 50% of the photo point set is used for each call of *LineSegmentMatcher*. This attempt may produce a better match, but it has an elevated risk of failure because the point sets are too small. In any case, if the match is better this time, it is kept.

The detected mountain peaks from the 2D model and the segmentation further facilitate the process of choosing the best fit transformation. After the locations of the mountain peaks in the point sets have been found, the point sets are "trimmed" around these peak points. For example, only the 75% of points around a peak point (in both the 2D model and the photo) is kept as input to the *LineSegmentMatcher*. *LineSegmentMatcher* is called once for these point sets and the match error is computed. If the match error is better than the error found from the previous calls, the fit transformation is the final one.

The translation required for the 2D model for it to fit the photo, roughly determines the deviation in the compass measurement and can be used for correcting it.

7.5. Visualization of the New Semantic Map

The fit transformation calculated from the algorithm of the previous section is applied to all the pixels of the 2D model image. Then, the individuals are extracted from the image one by one by finding their bounding box, saving it as an image and storing their coordinates. They are then superimposed on the photo. Figure 18 presents the visualization of the newly created photo/semantic map using the familiar photo from a previous section. The individuals present in the 2D model are now overlaid on top of the photo in their correct location. A careful reader should notice that the individual representing the sea in the photo below is different from the corresponding individual in the 2D model. This happens because the sea region in the photo has been successfully detected and has replaced the old individual footprint with a more accurate representation.

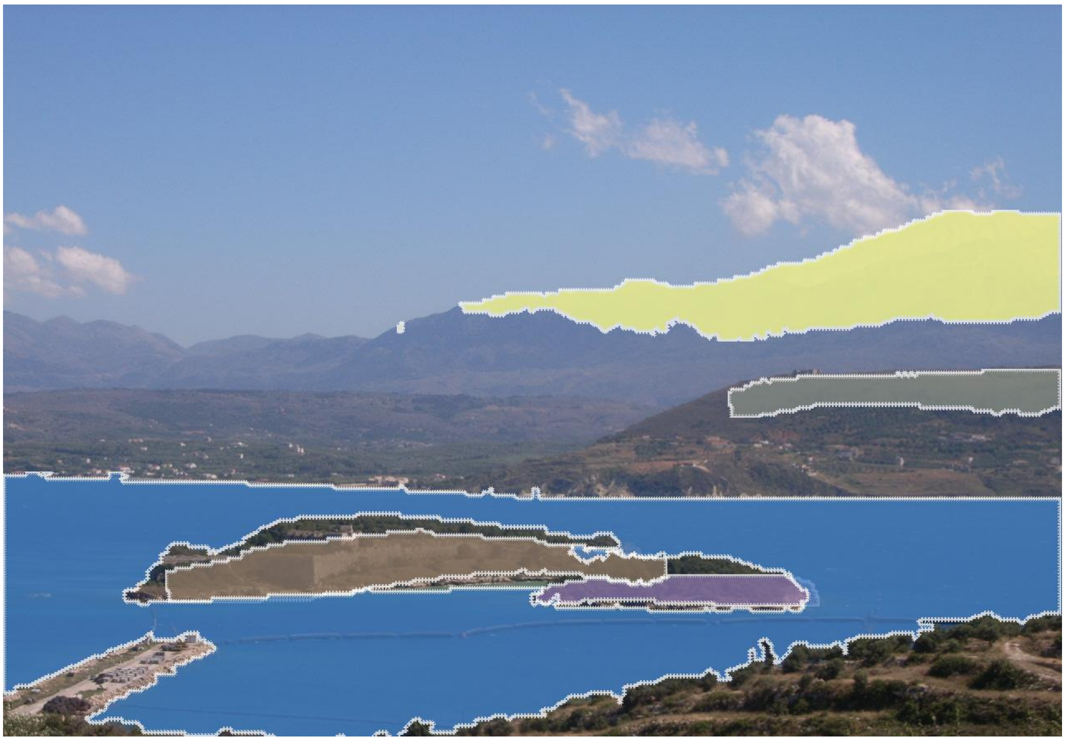


FIGURE 18: A PHOTO VIEWED AS A SEMANTIC MAP. THE VARIOUS INDIVIDUALS ARE SUPERIMPOSED ON THE IMAGE AND HAVE DIFFERENT COLORS.

7.6. Summary

In this chapter, it was shown how a photo can be transformed into a semantic map using the captured metadata and information found in semantic maps. The procedure includes finding a 2D model of the view from the camera's position towards the camera's direction and segmenting the photo into regions. Then, the skylines and sea boundaries (if present) are extracted from them along with information about the location of mountain peaks. The collected information is processed by a matching algorithm, in order to provide a fit transformation to apply to the 2D model and overlay the individuals present in the 2D view on top of the photo, creating individual footprints. The new semantic map is then ready.

8. Evaluation of the System Algorithms

This chapter describes the evaluation of the algorithms presented in chapter 7. For this purpose, a camera system was utilized and photos were captured and given as input to the system to perform the transformations. The evaluation procedure tested transformation precision, direction correction and quality of the semantic maps produced from the pictures. Section 8.1 provides details about the test configuration (camera, sensors, algorithm parameters etc.) for the evaluation and 8.2 gives a short description of the dataset of photos used. The results are discussed in section 8.3.

8.1. The Test Configuration

This section describes all the different components and parameters of the evaluation.

8.1.1. Camera and Sensor

For this research, the following equipment was used:

- Ricoh Caplio 500SE digital camera
- Wintec WBT-100 GPS receiver with integrated digital compass

The 500SE is a state of the art digital camera by Ricoh [60] that has the capability to interoperate with various external sensors through Bluetooth wireless communication. It also has Wi-Fi connection capabilities, allowing it to easily send the captured photos to a remote device. The camera's body is resistant to dust and water, making it robust for many environments. Another important characteristic unique to this camera at the time of writing this thesis is the ability to input text memos for each photograph. Interoperability with a laser rangefinder is supported; however these sensors are typically very expensive.

The WBT-100 is a very small sized (60 x 38 x 16 mm) GPS receiver with an integrated digital compass. It provides all the functionalities of modern GPS receivers without a screen, including NMEA connectivity, GPS data logging and long battery life. The integrated digital compass captures direction information that is transmitted along with the position parameters from the GPS sensor. The direction is given in integer values from 0 to 359, with 0 indicating north, 90 indicating east and so on. Illustrations of the two devices can be seen in Figure 19 (the devices have different scale in the figure).

FIGURE 19: CAMERA AND GPS RECEIVER/DIGITAL COMPASS.

The procedure for the camera/GPS connection is as simple as turning on the two devices and selecting from the "Bluetooth" devices menu of the camera the name of the GPS receiver. When the receiver triangulates the GPS signal, it starts transmitting NMEA sentences to the camera, providing various information including latitude, longitude, satellite fix quality and type, number of satellites being tracked, altitude, speed, date/time, magnetic heading and more. These parameters are all captured in the Exif header of the produced image.

In general, during the experiments, the accuracy of the GPS receiver was satisfactory. When the satellite fix was good, the measurements were accurate with only up to a few meters deviation. The digital compass however proved to be extremely sensitive and it was greatly affected by the sensor's orientation. To produce more accurate results, a contraption was used that kept the sensor as horizontal as possible to avoid large errors in the compass measurements. Even though this helped reduce the huge deviations from the true headings, minor to medium errors in the calculations were common.

It should be noted that the latest version of the camera used now has both GPS and compass sensors embedded and thus does not require an external sensor to record position and direction information. Having the digital compass embedded in the camera's body allows for greater accuracy in the direction measurements. Moreover, the compass transmits direction values in decimal numbers and not integers, increasing the accuracy even further.

8.1.2. System Parameters

A semantic map was constructed for the purposes of this research. The research was conducted in various places on the island of Crete and the appropriate free digital elevation model data was downloaded from the NASA SRTM [66] website [48] and visualized using a freeware DEM visualization tool [1] to produce the semantic map image of the island.

The semantic map was annotated with various individuals represented as colored polygons belonging to domains such as geography, archaeology and culture. Some of the semantic types involved included mountain areas, sea areas, islands, mountain peaks, ruins, villages, towns and buildings. The geographical coordinates for each individual footprint polygon were manually obtained using Google Earth place marks to ensure the accuracy of the geometric representations. The GPS coordinates for each polygon edge were stored in the system database and their map pixel representation was calculated using the GPS to pixel transformation function of the semantic map. The visualization of the semantic map with the individual footprints is similar to the one in Figure 5.

The semantic map was also annotated with the elevation data of the SRTM [66] DEM corresponding to the island of Crete. Unfortunately, NASA hadn't released high resolution elevation data for non-USA locations at the time this work was conducted. High quality processed elevation data from other sources is usually expensive (and not available for Crete). The elevation data used contain elevation values for every three arc seconds, practically providing elevation values for every 90m of distance. This proved not accurate enough for calculations near the user, but sufficient for long distance calculations, for example the skyline formed by distant mountains. Another problem of the SRTM data used was that they contained gaps, which are areas with no elevation values. This problem was somehow averted by using a gap-filling program, at least preventing large errors in the calculations.

8.1.3. Algorithm Parameters

This subsection provides details about the parameters used for each of the algorithms of the photo to semantic map transformation. The choices of the parameter values were determined after careful observation and experimentation with different values. Comments and some examples for the selection of each one are provided.

Viewshed Calculation Parameters

This algorithm has three relevant parameters:

- **Max KM distance**: The maximum distance from the user position in which the algorithm searches for visible areas. In other words it represents the length of each ray cast.
- **KM step**: Defines the distance between two consequent points in the same ray. This parameter is not constant and changes according to the distance of the current point in the ray.

- **Number of rays**: The number of rays cast, which also determines the resolution (width) of the produced 2D view model.

Values:

Max KM distance: 40km. This distance value proved adequate for calculating the skyline from the mountains in the distance. Its value should depend on the type of semantic map and the weather conditions at the time of picture taking. For example in areas where the probability of visible mountains in long distances is higher, this parameter should be increased accordingly.

KM step: To avoid unnecessary checks for visibility in far distances from the user, this parameter was defined as a function of the current distance.

$$KMstep = \begin{cases} 2m, & current\ distance < 2.5km \\ 5m, 2.5km \leq current\ distance < 5km \\ 10m, 5km \leq current\ distance < 10km \\ 20m, & current\ distance \geq 10km \end{cases}$$

This provided the needed accuracy for short distances and reduced the running time of the algorithm by relaxing the interval between points in the ray for distant points.

Number of rays: 320 for fast computation, 1280 for high resolution 2D view models. The value of 320 was chosen because it produces 2D view models quickly and the skyline and individuals extracted are easily detected. When high resolution is chosen, for each 4 rays, one is used for creating the low resolution model (since 1280/4=320) concurrently. The low resolution model was used just for the skyline/individual extraction, while the high resolution model was the one to be transformed using the fit transformation to produce the semantic map. This gives the ability to the user to choose if he/she wants to sacrifice computation time for better visual results.

Image Segmentation Parameters

The image segmentation algorithm has the following parameters:

- **Q**: The statistical region merging method has a parameter Q that defines the statistical complexity, in practice roughly defining the number of regions that should be formed. The larger the value of Q, the more regions the image is segmented into.
- **Resolution**: The resolution of the image to be segmented. Modern digital cameras produce very high resolution images that take too much time and are too complex to be segmented in that resolution. This parameter defines the value that the images should be scaled to, to ensure the segmentation procedure is both fast and produces accurate results.

Values:

Q: 100. This value was chosen because it produces segmentations that generally do not segment the sky into many pieces and the sea areas can be adequately distinguished from the other regions.

Resolution: 320x240. This resolution allows for quick segmentation and the skyline is not complex enough to produce anomalies. It is also compatible with the width of the 2D model, to keep the scale of the detected skyline/individuals equal to the one of the 2D model.

Image Registration Parameters

The parameters of image registration are divided into those that concern the line segment approximation, those that have to do with the matching algorithm and those that govern how the matching algorithm is used.

- **Line segment approximator pixel tolerance**: Used by the line segment approximation algorithm in [61]. It is the value (measured in pixels) that the maximum perpendicular distance between the set of points and the approximation line of that segment must not exceed. The accuracy of the approximation is inversely proportional to the tolerance value. Figure 20 **B** illustrates the distances that need to be computed.

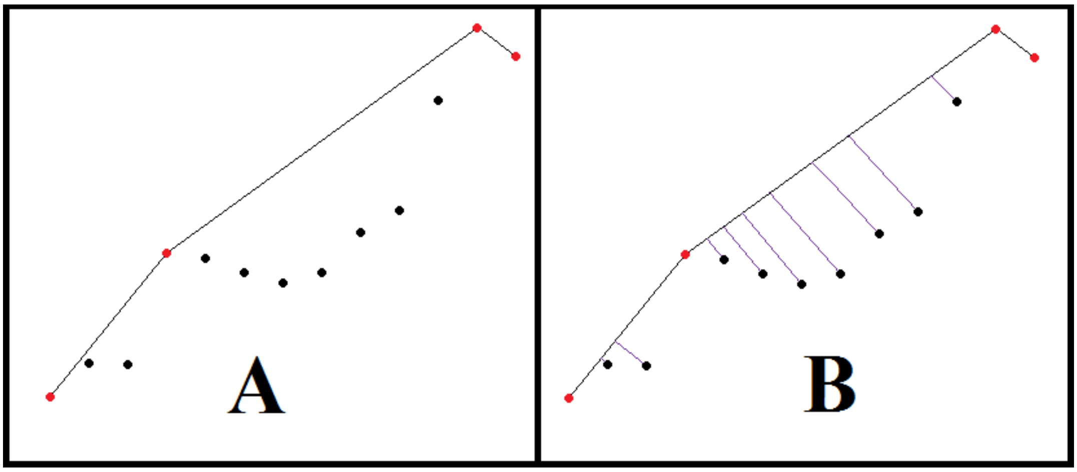


FIGURE 20: APPROXIMATION ERROR TOLERANCE EXAMPLE. A) A SET OF POINTS APPROXIMATED BY 3 LINE SEGMENTS BY COMPUTING THE UPPER CONVEX HULL. B) THE PERPENDICULAR DISTANCES FROM THE POINTS TO THEIR RESPECTIVE APPROXIMATIONS ARE CALCULATED AND COMPARED TO THE ERROR TOLERANCE TO DETERMINE IF FURTHER APPROXIMATION IS REQUIRED.

- **Line segment approximator minimum segment length**: The approximated line segments must have length (in pixels) equal or greater than the value of this parameter.

- **Line segment matcher maximum iterations**: The number of times the matching algorithm iterates without having converged.
- **Line segment matcher minimum percentage of mandatory line segment pairs matched**: The line segment matching algorithm works by calculating the total error of the matches between line segment pairs. Bad pairs are eliminated in each iteration to provide a better match. This parameter defines the minimum percentage of matched pairs to ensure that the match has adequate number of pairs to define a concrete transformation. Without this limit, the algorithm would eliminate too many matched pairs, leaving only several that match perfectly but result in a failed overall match. In other words, it does not allow the few to decide for the many.
- **Line segment matcher maximum midpoint distance between matched segments**: This parameter blocks the addition of a line segment pair to the set of matched pairs during the iterations, if the model and line segment midpoints are too distant from each other, effectively protecting the algorithm from even considering matching highly unlikely line segment pairs. The distance is measured in pixels. Figure 21 shows what kind of matches this threshold forbids. For the leftmost data line segment d1, the algorithm may choose to match it with the one shown with the arrow that has a tick (m1). Even though the model line segment shown with the arrow that has an X (m2) is more similar to d1 (they have the same slope), they are too distant to be considered useful in the match.

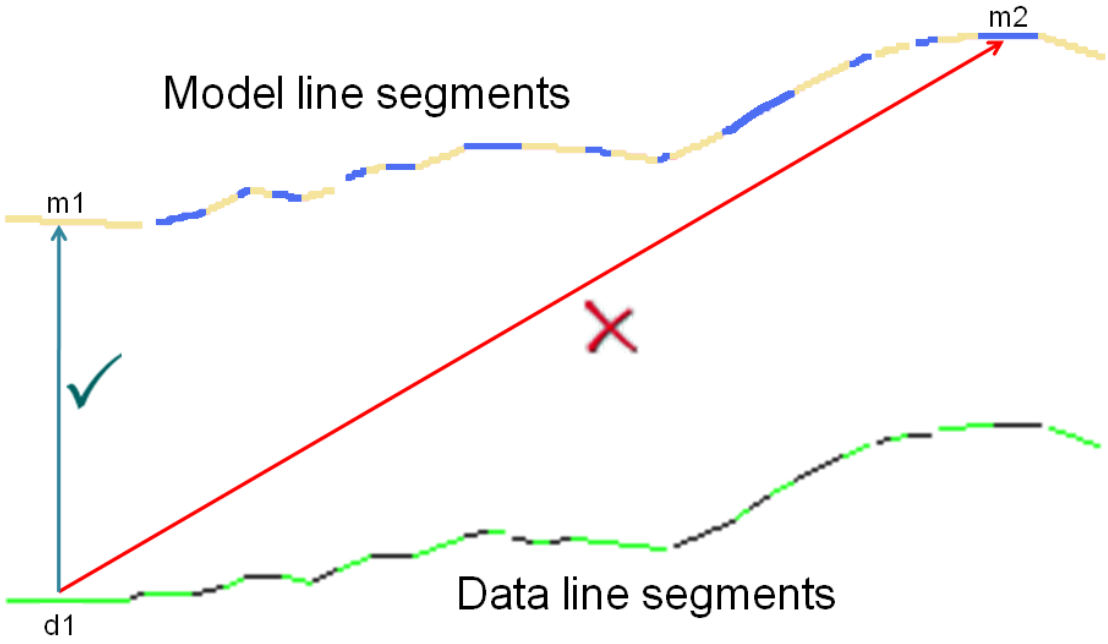


FIGURE 21: MAXIMUM MIDPOINT DISTANCE THRESHOLD. THIS PARAMETER FORBIDS THE MATCHING ALGORITHM TO CHOOSE DISTANT LINE SEGMENTS TO MATCH.

- **Line segment matcher scale error tolerance**: Defines the scale values that are acceptable by the matching algorithm before increasing the total error. If the scale parameter of the fit transformation calculated exceeds the acceptable range, the scale error is linearly increased. It is essentially the parameter *r* as defined in the scale error metric in section 7.4.1.
- **Line segment matcher rotation error tolerance**: Defines the minimum rotation in degrees that is acceptable before producing a large rotation error. Small rotations produce small amounts of rotation error values. When the rotation calculated exceeds the tolerance, a larger error value is produced that increases linearly with the rotation. It is essentially the parameter *r* as defined in the rotation error metric in section 7.4.1.
- **2D model points discard percentage**: Represents the percentage of points to be discarded from each side of the 2D model point sets. The matching algorithm does not use all the points from the 2D model point sets for the matching. As described in the corresponding section about image direction correction (7.4.2), only a percentage of the central points in each set are kept and the rest are discarded. This is useful when the heading measurement of the compass is several degrees off and as a result a part of either the left or the right side of the 2D model does not correspond to the photo.
- **Average distance threshold for initiating second try with fewer points**: After each call of the line segment matching algorithm, the average vertical pixel distance between the corresponding skylines and boundaries is computed and used as a measure to define the quality of the transformation. When all the calls are completed and the minimum average distance error is found to be larger than this threshold, a second set of tries is initiated, this time discarding more points (double in number) from the 2D point sets in an attempt to correct large deviations in the compass measurements. Of course, when this attempt does not produce better results, the previous transformation is considered as the best result.

Values:

Line segment approximator pixel tolerance: 2 pixels. This value was selected because 1 pixel was too small and produced a lot of line segments, while a value of 3 or more could not produce accurate enough approximations.

Line segment approximator minimum segment length: 4 pixels. Line segments less than 4 pixels long were too small to be considered reliable for the matching.

Line segment matcher maximum iterations: 100 iterations. 100 was proven to be a solid number for stopping the number of iterations if the algorithm had not yet converged.

Line segment matcher minimum percentage of mandatory line segment pairs matched: 80% of the number of model line segments. For example, if the model line segments are 200, it means that the set of matched pairs must have at least 160 pairs for it to be reliable enough.

Line segment matcher maximum midpoint distance between matched segments: 1 pixel. 1 pixel is a very strict limitation on the midpoint distance but it also eliminated many undesirable matches.

Line segment matcher scale error tolerance: 5%. Usually a good match has a scale close to 1, because the 2D model is built to be about the same scale as the photo. 5% tolerance allowed for small deviations (which were acceptable).

Line segment matcher rotation error tolerance: 5 degrees. This allowed for small rotations in the transformation, which would always occur.

2D model points discard percentage: 12.5% of points from each side. For the first try of the matching, 25% of the 2D model points were discarded and the central 75% were kept. For the second try, only the central 50% of points were used.

Average distance threshold for initiating second try with fewer points: 2.5 pixels. Taking the skylines as an example, the average vertical pixel distance between the skyline detected from the segmentation and the transformed skyline from the 2D model was computed and if it was more than 2.5, the second try was initiated, discarding more points and running the matching algorithm again as described in section (7.4.2).

8.2. The Dataset

The camera and GPS receiver were used to capture 69 images of various places on the island of Crete. The photos depicted villages and towns, mountains, sea areas, islands, mountain peaks and archaeological or other places. A sample subset of the photos is shown in Figure 22.

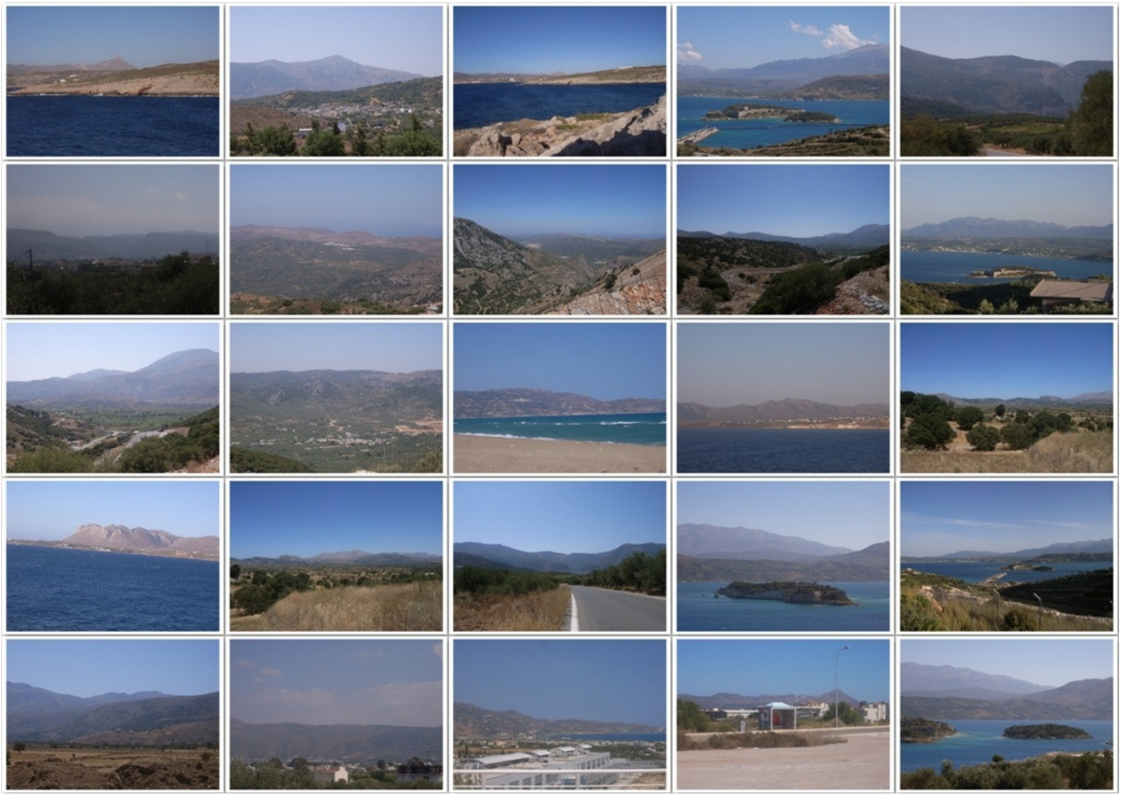


FIGURE 22: SAMPLE SUBSET OF THE PHOTOS USED FOR THE SYSTEM EVALUATION.

From the 69 photos, 33 contained sea areas, 16 contained distinguishable mountain peaks and 14 contained islands. The minimum focal length of the camera lens was 5.8mm and the maximum (for most of the photos) was 17.4mm. The compass measurements using the external digital compass were highly inaccurate due to the sensitivity of the small sensor. The true headings for each photo were experimentally computed and it was found that the average deviation of the compass measurements was about 5.4 degrees. The manual handling of the compass using human perception for determining the camera direction and pointing with the compass was the main reason for the erroneous measurements. These factors should be eliminated when the digital compass is integrated with the digital camera and in this case the errors should not occur.

After the images were captured, they were downloaded to a PC running the SPIM prototype and using the semantic map of Crete described in subsection 8.1.2. The pictures were stored in the database and the photo to semantic map transformation was applied to each one.

The evaluation procedure used the implemented algorithms on the dataset three times. In the first test run, the images were processed using their true direction, while in the second and third test runs, the directions were modified on purpose (-4 degrees and +7 degrees respectively) to show the algorithms' ability to correct small errors in the direction measurements. Details about the results are in the next section.

8.3. Results and Discussion

This section shows the results of the implemented system's algorithms on the images that comprise the dataset. Results are below, followed by comments and discussion about their values.

Skyline Detection and Segmentation Quality

Detecting the skyline from the 2D view models was a trivial task; the difficulty was in segmenting the image correctly to ensure the skyline can be extracted without significant errors. The segmentation algorithm tried to detect and merge the multiple regions representing the sky and also remove obstacles such as clouds.

Applying the segmentation algorithm to the 69 images of the dataset yielded the following statistics:

- 1 complete failure in the segmentation, caused by the sky detection algorithm due to low visibility of the distant mountains.
- 1 failure where the sky detecting algorithm merged too many regions thinking they should be included with the sky region.
- 6 minor anomalies in the skyline detection without affecting the other results.
- 61 satisfactory skyline detections

Overall, the sky detection technique worked for the majority (88.5%) of the images.

Sea Detection

Once again, sea detection from the 2D view models was done without complications. The results of the sea detection for the 33 images containing sea regions are below:

- 1 complete failure in the segmentation (same as above).
- 4 images where the sea was not detected at all. Interestingly, all the images were captured at a wide angle. Maybe having too many objects out of focus prevented the algorithm to find a distinguishable sea region.
- 1 image had two sea regions and only one was detected.
- 5 images had minor inaccuracies in the detected sea region, due to errors in the segmentation and not the detection technique.
- 22 images had satisfactory sea region detection.

Overall, the sea detection technique detected the correct region for a decent percentage (81.8%) of the photos. Due to segmentation errors, a smaller percentage (66.6%) is considered completely successful.

Of the 33 images containing sea bodies, an average of 30% proved helpful for acquiring a better registration result. The rest used just the skylines and the mountain peaks for the transformation.

Mountain Peak Detection

The inclusion of a mountain peak in a photo was based on whether there is a mountain peak individual in the photo's 2D view model when the direction measurement is correct. With that in mind, the results for the 16 images with mountain peaks were:

- 2 complete failures in the segmentation.
- 1 inability to detect a peak due to small segmentation errors (even without the errors it would not help to detect that peak in that specific photo).
- 3 images with wide angle in which the mountain peaks could not be distinguished from the skyline.
- 10 successful detections.

Overall, mountain peak detection did the best it could to determine where the mountain peaks were located. The detection percentage was not high by itself (62.5%) but the problem was not in the method, as already two of the images in the dataset were difficult to segment and always counted as negatives in all the results. In addition, images taken with a wide angle were not expected to have very distinguishable mountain peaks. There were some cases in the results where the peak detection helped produce the best result even when the match error was high.

Resulting Semantic Map Quality

The most important test for the algorithms was how well they cooperated to produce high quality semantic maps from the photos. There isn't a good measure for determining the quality of the semantic maps other than visual perception. With that in mind, the resulting semantic maps were divided by a user into 7 categories, each corresponding to an adjective describing the quality of the semantic map. From best to worst they were: *Perfect*, *Good*, *Acceptable*, *Average*, *Bad*, *Awful* and *Failed*. The first four were considered "pass" grades, while the other three "fail" grades. Figure 23 shows nine semantic maps belonging to various quality classifications. Starting from the top row and following each row from left to right, the classifications for each picture are: *Good*, *Good*, *Average*, *Perfect*, *Bad*, *Awful*, *Failed*, *Acceptable*, and *Acceptable*.

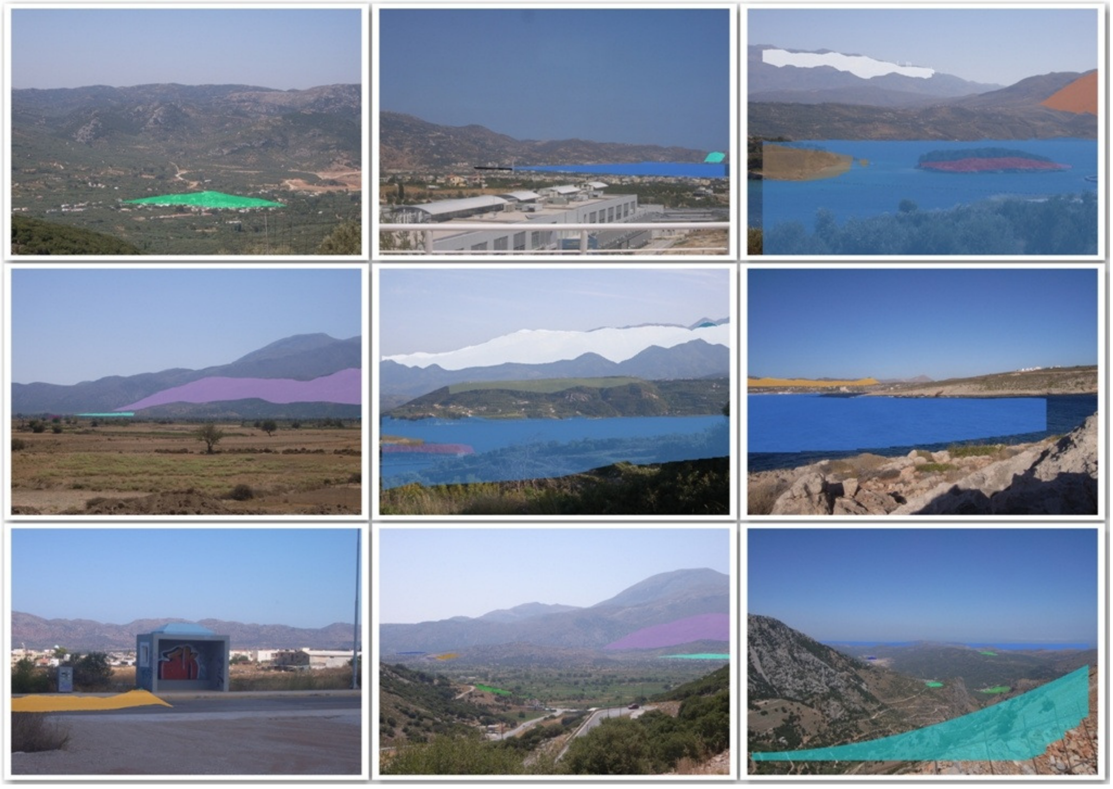


FIGURE 23: VARIOUS PHOTOS VIEWED AS SEMANTIC MAPS.

Each resulting semantic map was inspected thoroughly in terms of how well the individual footprints were matched to the picture contents. The whole process was done in an unbiased way. Table 1 summarizes the results of the evaluation for each test run of the algorithms.

Distinction	True Heading	-4 degrees	+7 degrees
Perfect	7	10	3
Good	26	16	13
Acceptable	19	18	12
Average	11	12	14
Bad	3	7	11
Awful	1	4	10
Failed	2	2	6
Pass	63	56	42
Fail	6	13	27
Total	69	69	69
Pass Percentage	91.3%	81.2%	60.9%
Fail Percentage	8.7%	18.8%	39.1%

TABLE 1: EVALUATION OF THE CREATED SEMANTIC MAPS.

At first sight, the resulting semantic maps when the images have their true heading were satisfactory. The same applies to the small heading deviation (-4 degrees), showing that the system algorithms were capable of correcting small errors in the compass measurements. When the deviation was greater (+7 degrees), the 2D models needed to be translated a great deal in order to register correctly on the photo because they contained imperfect data (skylines etc.). The percentage of decent transformations in that case was more than 50%, however most transformations were considered average to acceptable. Table 2 shows the effect that the second try of the matching algorithm (with more discarded 2D model points) had on the transformations with match error greater than the defined threshold (2.5 pixels).

Statistic	True Heading	-4 degrees	+7 degrees
Average Vertical Distance Match Error – First Try	1.22 pixels (69/69)	1.46 pixels (69/69)	3.17 pixels (69/69)
Average Vertical Distance Match Error when Match Error > 2.5 – First Try	6.27 pixels (1/69)	3.32 pixels (7/69)	3.73 pixels (35/69)
Average Vertical Distance Match Error when Match Error > 2.5 – Second Try	5.73 pixels (1/69)	1.88 pixels (7/69)	1.69 pixels (35/69)

TABLE 2: AVERAGE VERTICAL DISTANCE MATCH ERRORS IN DIFFERENT CASES.

The average match errors for the true headings and -4 degree deviation were below 2.5 pixels, indicating (but not proving) that the average match was decent. In the cases that the match error exceeded the threshold of 2.5 pixels, the second try was initiated, always lowering the match error. In the case of the +7 degree deviation, more than half of the transformations needed the second try to lower the (usually) large match error, with moderate success.

Discussion

Overall, the algorithms seemed to work well even with small deviations of the compass heading in comparison with the true heading. It can be safely assumed that the whole procedure had an accuracy of up to 4-5 degrees before the quality of the resulting semantic maps deteriorated.

It is important to focus on why the rest of the results were unsatisfactory. First of all, 2 of the segmentations were totally wrong due to bad or wrong sky detection and making the algorithm not work at all. They also negatively affected all of the statistics and average calculations for all the photos. The sea region detection was correct most of the time, on the other hand the sea boundary extraction was not. This was caused by obstacles (usually the islands) that got in the way. For example, in the 2D model of one of the pictures it was observed that there was an island in the sea but in the photo the island was high enough to obscure the sea boundary far away, resulting in different sea boundary detection in the segmentation and confusing the matching algorithm.

Another problem for the drop in semantic map quality was the lack of detailed elevation data. In some semantic maps, hills in the photo got in the way but in the 2D view model they were not visible due to the inaccuracies. That way, some individual footprints that should not have been in the photo because they were obscured by the landscape, were incorrectly shown to be visible. Moreover, the lack of detail in the DEM data forbade the detection and matching of nearby mountain peaks, making the algorithm accurate only for skylines produced by far away mountain ranges, where elevation details were not required. The filled-gaps present in the elevation data also affected the results, although not greatly.

8.4. Summary

This chapter presented the evaluation part of the research. To use the implemented algorithms, a camera system including a state of the art digital camera and a small GPS receiver with digital compass was assembled and used for capturing images. The images then were given as input to the system's algorithms which used all the available data from a semantic map of Crete to detect the semantic individuals present in the photos. Then, after segmenting the images, the registration procedure took into account features of the segmented images (skylines, sea boundaries, mountain peaks) and their 2D view models and computed a transformation for overlaying the semantic individuals from the 2D view models on top of the photos. Applying that transformation resulted in transforming the photos into interactive semantic maps. Evaluation of the resulting semantic maps showed that when the compass measurements were accurate or deviated up to 4 degrees, the transformation proved to be adequate.

9. The SPIM Prototype

To implement the basic ideas of the proposed framework, a prototype system has been implemented, complete with a client program and a server providing semantic maps for demonstrating the distributed aspect of the framework. The descriptions of the server and the client are the subjects of section 9.1 and section 9.2 respectively.

9.1. The SPIM Server

The server maintains a repository containing semantic maps and ontologies. Map representations of the individuals of each ontology have been added to the semantic maps. The server creates and distributes personalized versions of the semantic maps, containing only individuals with the names, semantic types or belonging to ontologies of the user's choice.

9.1.1. Server Architecture

Figure 24 shows the database design for the SPIM server's architecture. There is a purely conceptual distinction between the "ontology database" and the "semantic map database", emphasizing the independence between them.

The Semantic Map Base is the entity that contains both the Semantic Maps and the Semantic Map Individuals.

The Semantic Map:

- Includes pairs of GPS/pixel points (GPS Mapped Point) used for calculating the transformation matrix or function as described in section 6.2.
- Includes details about interpreting the transformation matrix or function (Semantic Map Transformation)
- Stores attributes such as information about the map's name, creator, the creation date, where north is on the map and how far should photo footprints with direction reach in the visualizations.
- Contains one or more Semantic Map Individual Representations or footprints, which are geometric (polygon/polyline) representations of the Semantic Map Individuals. The polygon/polyline edges are stored as ordered sets of pixel points (Footprint Pixel Point).

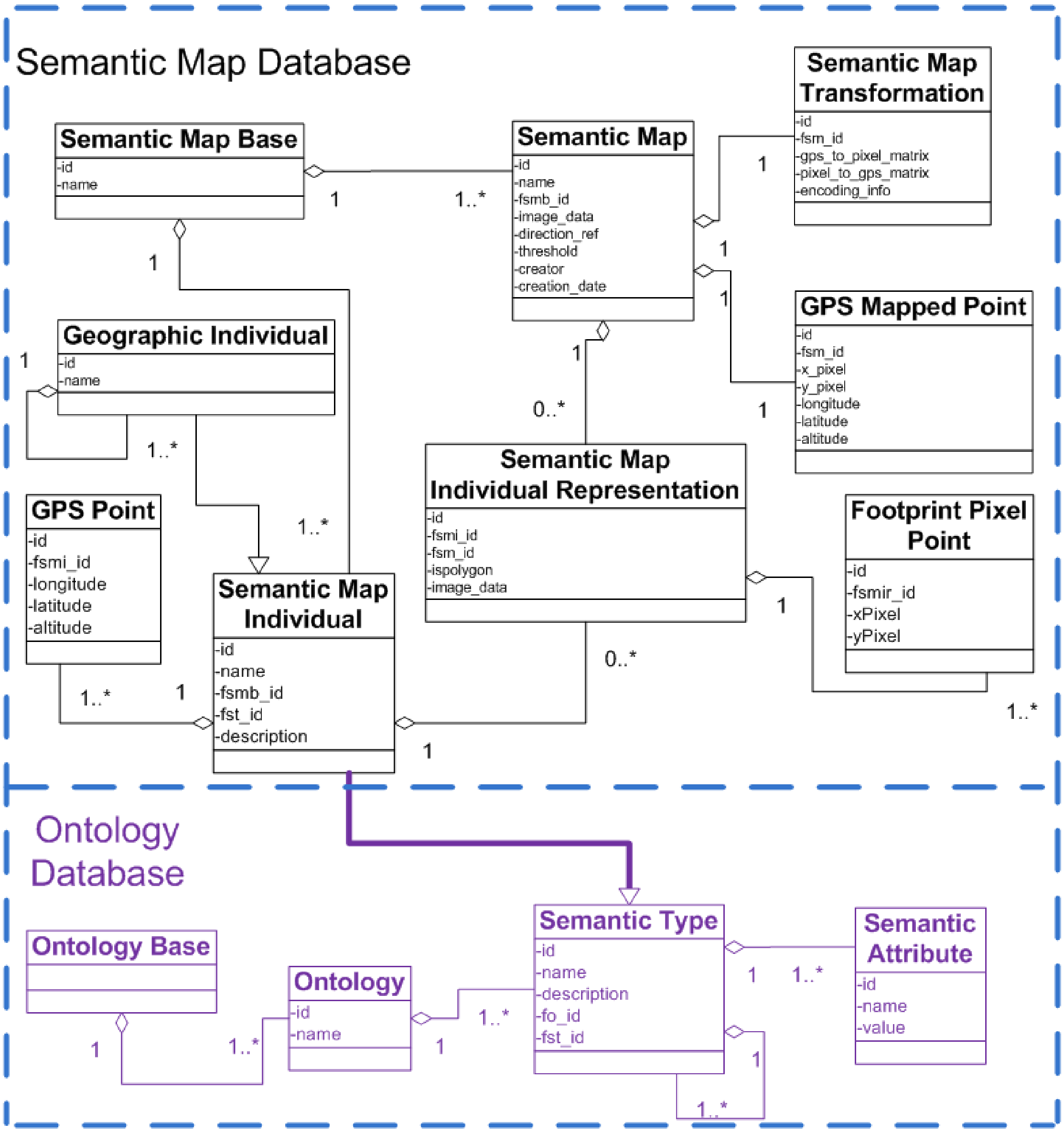


FIGURE 24: SPIM SERVER ARCHITECTURE.

The Semantic Map Individuals have a name and a description and can be distinguished as Geographic Individuals. They also have a polygon or polyline GPS footprint modeled as an ordered set of edges with GPS coordinates (GPS Point).

Moving to the "ontology database" part of the architecture, there is an Ontology Base containing all the relevant Ontologies to be used with the Semantic Maps. Each Ontology has a name and contains a number of Semantic Types, each with a name, a description and a set of Semantic Attributes. The Semantic Types have a hierarchy, meaning that there can be several layers of abstraction, with Semantic Attributes of concrete types inherited from the more abstract types. The connection of the ontology part of the database with the semantic map part is done by assigning Semantic Types to the Semantic Map Individuals.

9.1.2. Server Functionality

The SPIM server's main responsibility is to provide semantic maps on demand. The maps stored at the server's repository are richly annotated and contain loads of information. Usually, the users are only interested in a subset of the individuals present in the maps, the ones that fit their needs and interests. Therefore, the ability to provide personalized versions of the maps is a necessity. This also has the advantage that it reduces clutter on the semantic maps.

Personalization is achieved by offering several options to the user. The semantic entities of a semantic map are represented by a hierarchical tree-like structure. The root of the tree is the semantic map itself and its children are the ontologies involved with that map. The children, grandchildren etc. of each ontology that are not leafs are the semantic types belonging to that ontology. Finally, all the leaf nodes represent semantic individuals of their parent node's type. The user has the option to include or exclude from the semantic map whole ontology domains and semantic types. He/she can even choose to include or exclude specific individuals. After the selection process, the server creates a personalized instance of the chosen semantic map which the user can then download.

9.2. The SPIM Client

The SPIM client's responsibilities are numerous. It has the ability to visualize semantic maps and their individuals and allows interaction with them using the mouse. The semantic maps are provided by the SPIM server and after a map is retrieved, it is stored in the client's database. The client also allows the visualization of photo footprints on top of the map. It contains several panels that provide information about the individuals (ontology and semantic type they belong to, description, user photos that depict them etc.) and the photos (metadata, individuals in them etc.). Viewing the images is supported, with the added functionality of overlaying interactive individual footprints on top of the images. These individuals can be selected just like the individuals on the semantic maps and the relevant information is shown. User interfaces for all these functionalities, including the photo to semantic map transformation described in chapter 7 have been implemented.

9.2.1. Client Architecture

The database design for the SPIM client architecture is shown in the UML-based diagram of Figure 25. Apart from the conceptual distinction between the "ontology database" and the "semantic map database" as described for the SPIM server, there is also support for a "photo database", connecting the user photos with the semantic maps and their individuals.

Information stored in the client's database about a Photo includes the file name, the image data for the photograph, the date and time taken, some comments and a thumbnail for quick viewing in the client. More technical Photo Metadata is stored separately. The metadata includes the camera's model and manufacturer, the vertical and horizontal angle of views, the circle of confusion, the hyperfocal distance, longitude, latitude, altitude, heading, focal length and F number.

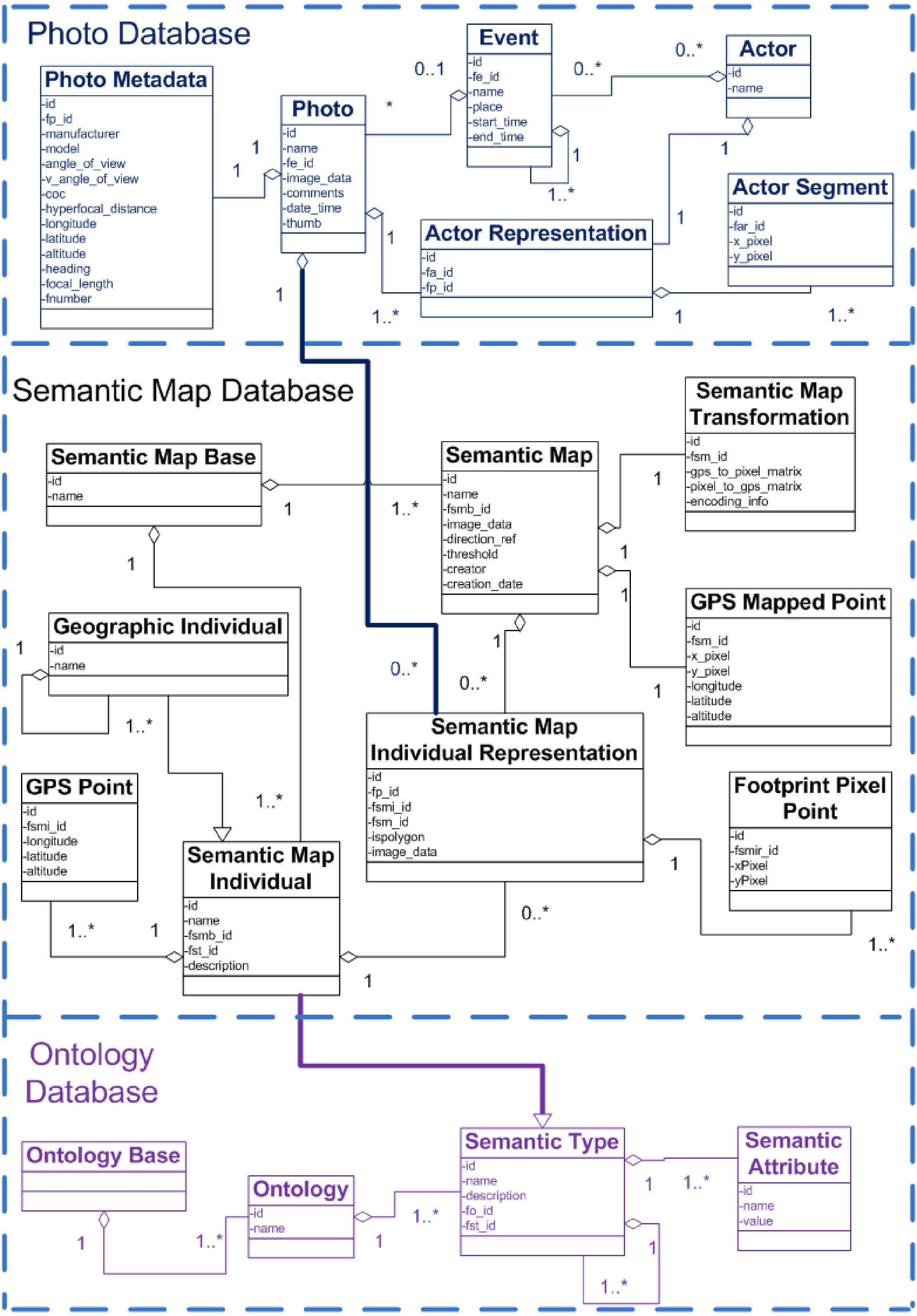


FIGURE 25: SPIM CLIENT ARCHITECTURE.

Photos are associated with Events and Persons. A Photo may belong to one Event, while an Event may contain more than one Photo. Events have a name, a place and duration defined by the time it started and the time it ended. An Event may be composed of other Events and has a number of people (Actors) participating. An Actor has a name and may participate in Events. Actor Representations are defined as ordered sets of Actor Segments (pixel points on the photo image) and represent faces or silhouettes of persons depicted in the Photos [7]. Finally, Actor Representations are associated with the Photos to define which persons appear in which photographs.

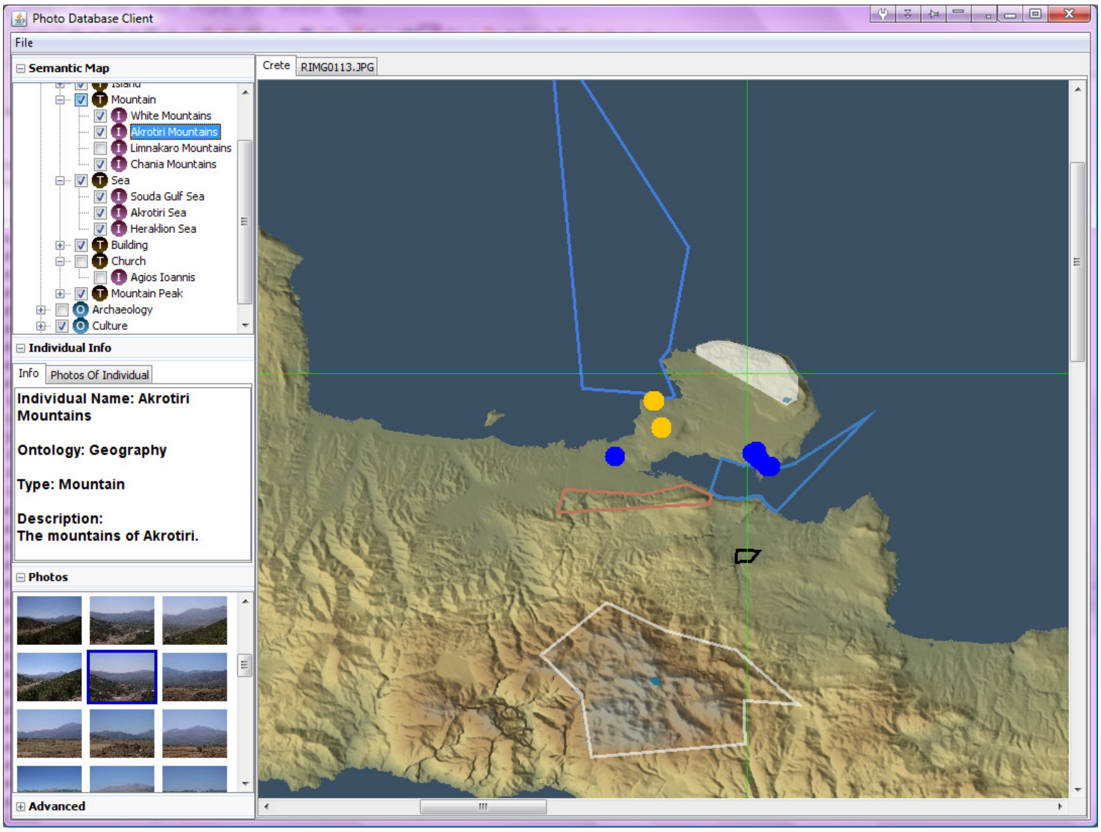
Semantic Map Individual Representations and their Footprint Pixel Points have an additional interpretation in the SPIM client. In the context of the Semantic Map they provide instructions on how the Semantic Map Individuals can be visualized on top of the map. Since the photos can be transformed to Semantic Maps themselves, using the transformation algorithms presented in chapter 7, the Semantic Map Individual Representations and their corresponding Footprint Pixel Points also provide visualization instructions for visualizing the individuals on top of the Photos.

9.2.2. Overview

The Graphical User Interface (GUI) of the SPIM client has many similarities with map and GIS applications. It is composed of several panels. Figure 26 shows an overview of the main application panel. It is divided into two areas. The area on the right is the Semantic Map Panel and contains the semantic map image, along with individual and photo footprints. The semantic map and footprint visualizations have already been presented in chapter 6. The same area is also suitable for viewing the photos, before and after they have been transformed to semantic maps. The area on the left contains informative panels. Each one of them can be collapsed to make space for showing more information from the others. From top to bottom they are: Ontology Tree, Semantic Individual Information Panel and Photo Information Panel. The ontology tree is directly associated to the semantic individuals on top of the current semantic map and offers functionalities concerning their selection and also visibility. The semantic information panel shows information about the selected individuals (from the map), while the photo information panel provides information about the photos and also the capability to transform the photos into semantic maps.

A more detailed description of the functionalities offered by each panel is presented in the following sections.

[7] This functionality requires advanced image recognition techniques (e.g. face recognition) which are out of the scope of this thesis and has not been implemented. It is shown here to emphasize how image content descriptions can be used in the framework.



Individual Name: Akrotiri Mountains

Ontology: Geography

Type: Mountain

Description:
The mountains of Akrotiri.

FIGURE 26: AN OVERVIEW OF THE SPIM CLIENT GUI.

9.2.3. Semantic Map Panel

The semantic map panel offers an interactive experience for browsing a semantic map and obtaining information about the entities that it contains. It has been explained many times in this thesis that the individual footprints and the photo footprints can be selected with the mouse button and information about them is shown in the corresponding panels. The GUI highlights appropriately the selected entities and makes sure their location is visible when selected. Figure 27 shows in more detail a state where an individual footprint has been selected (green) and the photos that the system has determined that depict that individual have been highlighted (in orange instead of blue). A photo has also been selected and its footprint is shown.

Another feature of the map panel is that the map can be dragged by pressing the right mouse button and dragging it to move the map.

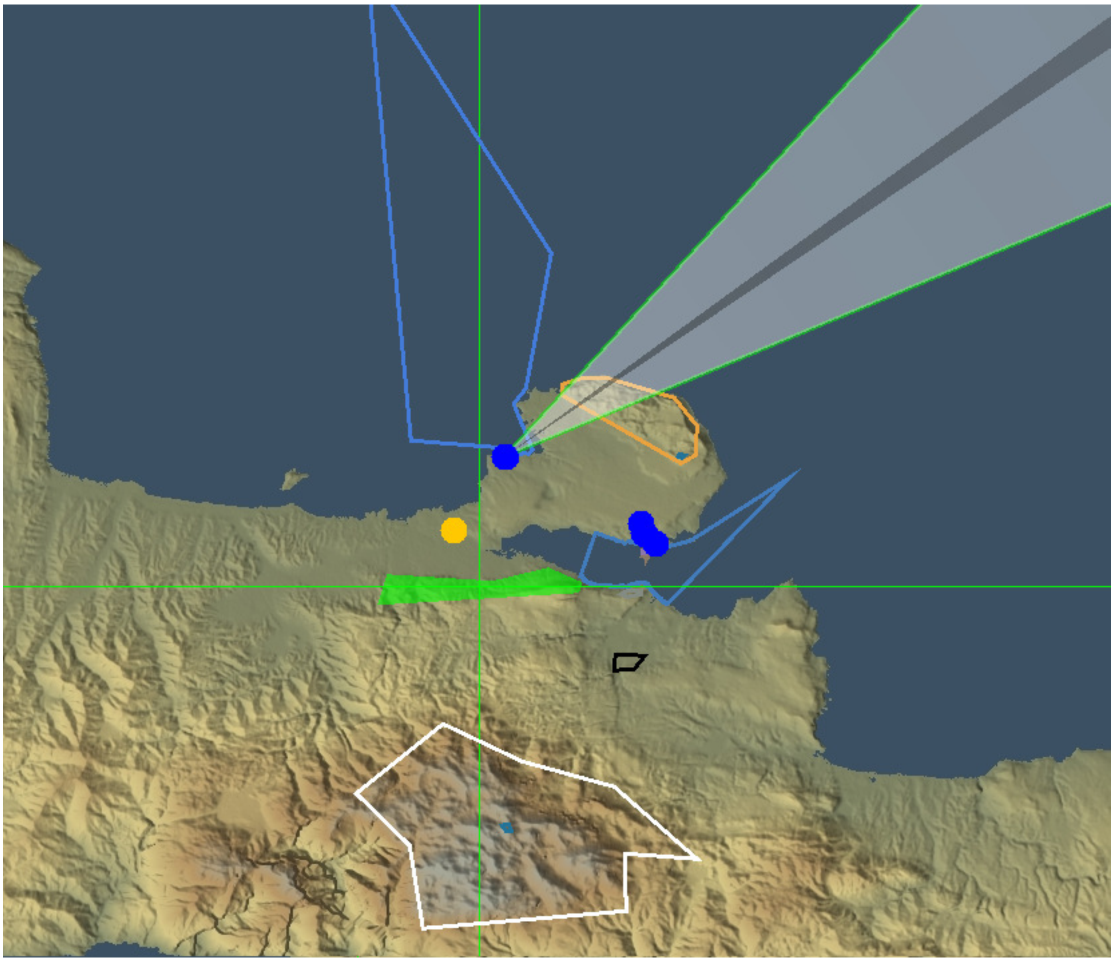
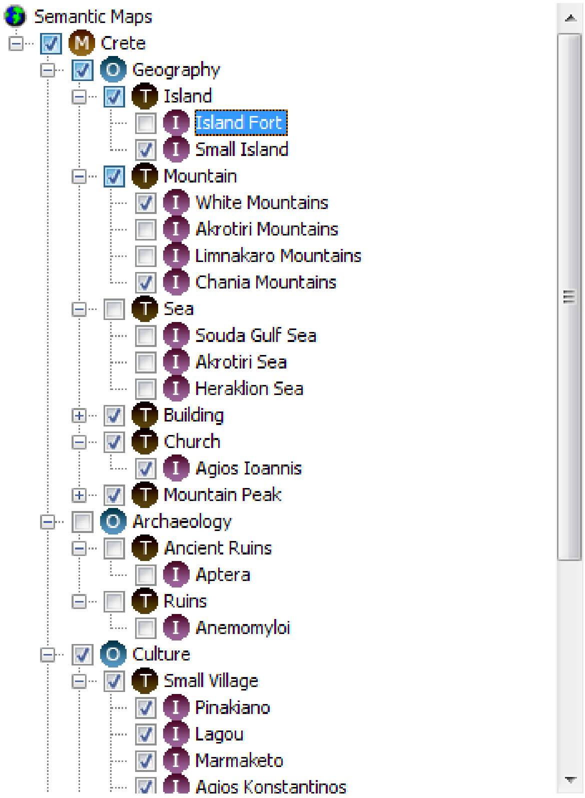


FIGURE 27: THE SEMANTIC MAP PANEL.

9.2.4. Ontology Tree

The ontology view panel is located at the upper left part of the window and contains the ontology hierarchy for the individuals of the currently visible semantic map. It is a tree-like structure that has a checkbox and an icon next to each ontology, semantic type or semantic individual name, as shown in Figure 28.



- Semantic Maps
  - ☑ M Crete
    - ☑ O Geography
      - ☑ T Island
        - □ I Island Fort
        - ☑ I Small Island
      - ☑ T Mountain
        - ☑ I White Mountains
        - □ I Akrotiri Mountains
        - □ I Limnakaro Mountains
        - ☑ I Chania Mountains
      - □ T Sea
        - □ I Souda Gulf Sea
        - □ I Akrotiri Sea
        - □ I Heraklion Sea
      - ☑ T Building
      - ☑ T Church
        - ☑ I Agios Ioannis
      - ☑ T Mountain Peak
    - □ O Archaeology
      - □ T Ancient Ruins
        - □ I Aptera
      - □ T Ruins
        - □ I Anemomyloi
    - ☑ O Culture
      - ☑ T Small Village
        - ☑ I Pinakiano
        - ☑ I Lagou
        - ☑ I Marmaketo
        - ☑ I Agios Konstantinos

FIGURE 28: THE ONTOLOGY TREE. THE LETTERS M, O, T AND I CORRESPOND TO MAP, ONTOLOGY, TYPE AND INDIVIDUAL RESPECTIVELY.

The icons reveal what kind of entity has the name next to the icon. There are four icons: M for semantic map names, O for ontology names, T for semantic type names and I for semantic individual names. For example, in Figure 28 the name of the semantic map is "Crete" and it contains at least the ontologies "Geography", "Archaeology" and "Culture". Under each ontology there are semantic types that belong to it. In the previous example, the semantic type "Mountain" has four individuals corresponding to differently marked mountains on top of the semantic map.

The ontology tree has two other important functionalities other than showing the semantic entities of the semantic maps. The first one is that the semantic individuals (the leafs of the tree) can be selected. When an individual is selected in this way, the semantic map panel centers the view on the selected individual and highlights it just like when clicking on its footprint on the map itself.

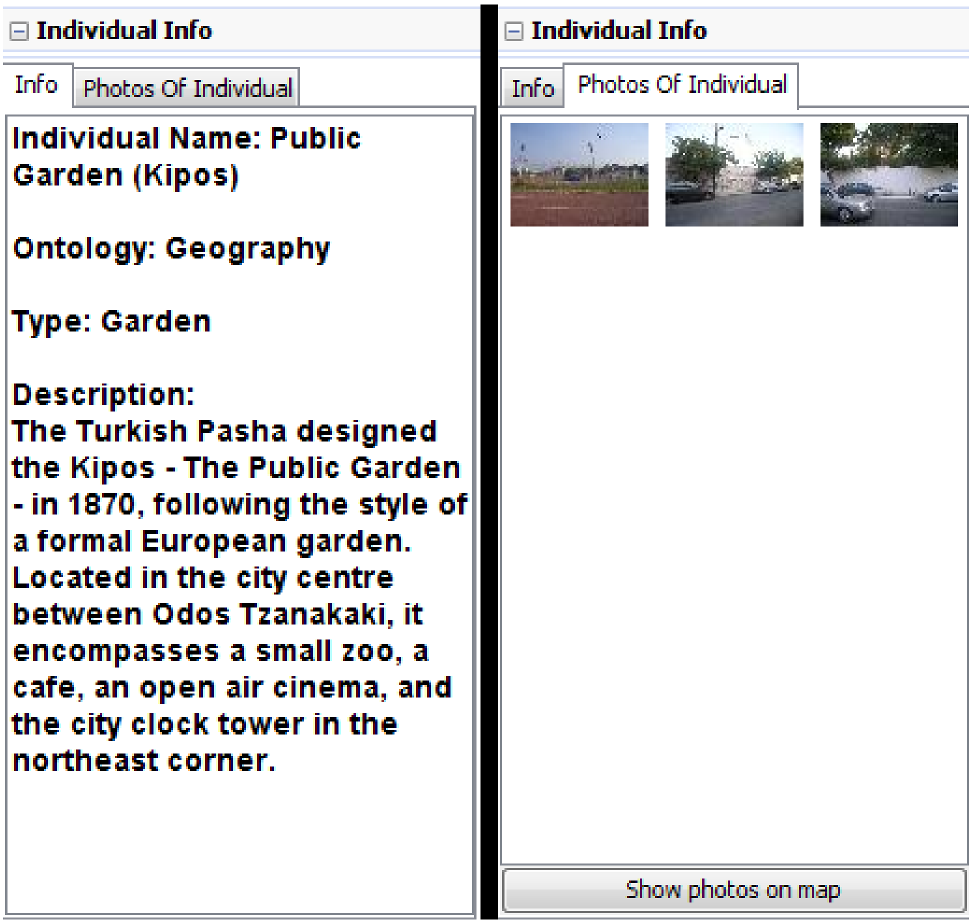
The second functionality is that it allows the user to toggle the visibility of the semantic individuals, using the corresponding checkboxes. Individuals that have a checked checkbox are visible on the map. For facilitating the user, the ontology tree is made in a special way and has the following properties:

1. When a node's checkbox is selected, all its descendants should also get checked and vice versa.
2. When all descendants of a node are not checked but some of them are checked (i.e. partial selection) then the node should be checked with a differently colored tick.
3. When all children of a node are unselected, the node should also be unselected.

The second and third properties define how the tree is visualized when there is partial or no selection. The tree in Figure 28 shows many different combinations of checked, unchecked and partially checked nodes. The first property enables the user to set the visibility of all the individuals that have a specific semantic type or belong to a specific ontology rather than one at a time. This can save a lot of time. The checkbox next to the semantic map name controls the visibility of all the individuals on the current map and can also be toggled.

9.2.5. Semantic Individual Information Panel

The semantic individual information panel is located just below the ontology tree and its purpose is to show information about the selected semantic individuals. Whenever an individual is selected from the semantic map, from the ontology tree or otherwise, this panel shows two kinds of information for the selected item, divided into two tabs. The first tab is a text area that prints relevant information about the individual: name, semantic type, ontology and description. It is shown on the left side of Figure 29. The right side of this figure shows the second tab, which contains thumbnails for all the photos determined by the system to depict the individual. These thumbnails are retrieved from the database and are not shown on the current semantic map. To do that, the user can click the "Show photos on map" button and then all the information about the photos will be available for the current semantic map. This was done for performance purposes; otherwise it would be necessary to load all the associated images and their information each time an individual was selected, which is computationally expensive.



Individual Name: Public Garden (Kipos)

Ontology: Geography

Type: Garden

Description:
The Turkish Pasha designed the Kipos - The Public Garden - in 1870, following the style of a formal European garden. Located in the city centre between Odos Tzanakaki, it encompasses a small zoo, a cafe, an open air cinema, and the city clock tower in the northeast corner.

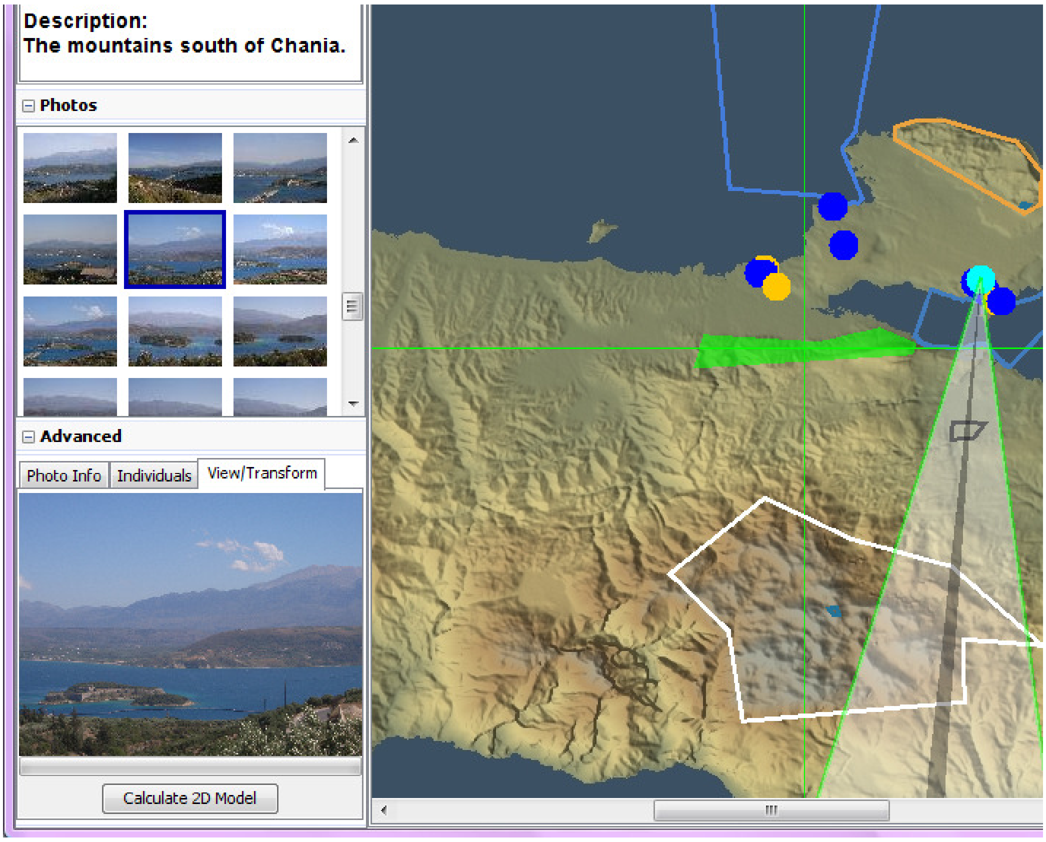
FIGURE 29: THE TWO FUNCTIONALITIES OF THE INDIVIDUAL INFO PANEL. THE LEFT TAB SHOWS INFORMATION ABOUT THE SELECTED INDIVIDUAL, WHILE THE RIGHT TAB SHOWS SELECTABLE THUMBNAILS OF PHOTOS THAT SHOW THE INDIVIDUAL.

9.2.6. Photo Information Panel

The photo information panel is located below the semantic individual info panel and is divided into two subpanels, the photo list and the advanced functions panel. The photo list contains the set of currently active photos. The list can be populated in three ways:

- Choosing the photos from the file system of the user's computer. This option is usually selected when the user has just downloaded the photos from a digital camera to the computer and desires to store them in the database provided by the SPIM client.
- Selecting from the already present photos in the database those that depict individuals of specific type, name and/or ontology. The query is formed using an ontology tree structure with checkboxes similar to the one in the GUI.
- Pushing the "Show photos on map" button from the semantic individual information panel.

Selecting a photo from the list selects and centers its photo footprint (if any) on the semantic map and information about the photo is shown on the subpanel below the list. Both subpanels are illustrated in Figure 30. The photos list contains thumbnails for the photos, while the advanced functions subpanel shows information and functionalities for the currently selected photo.



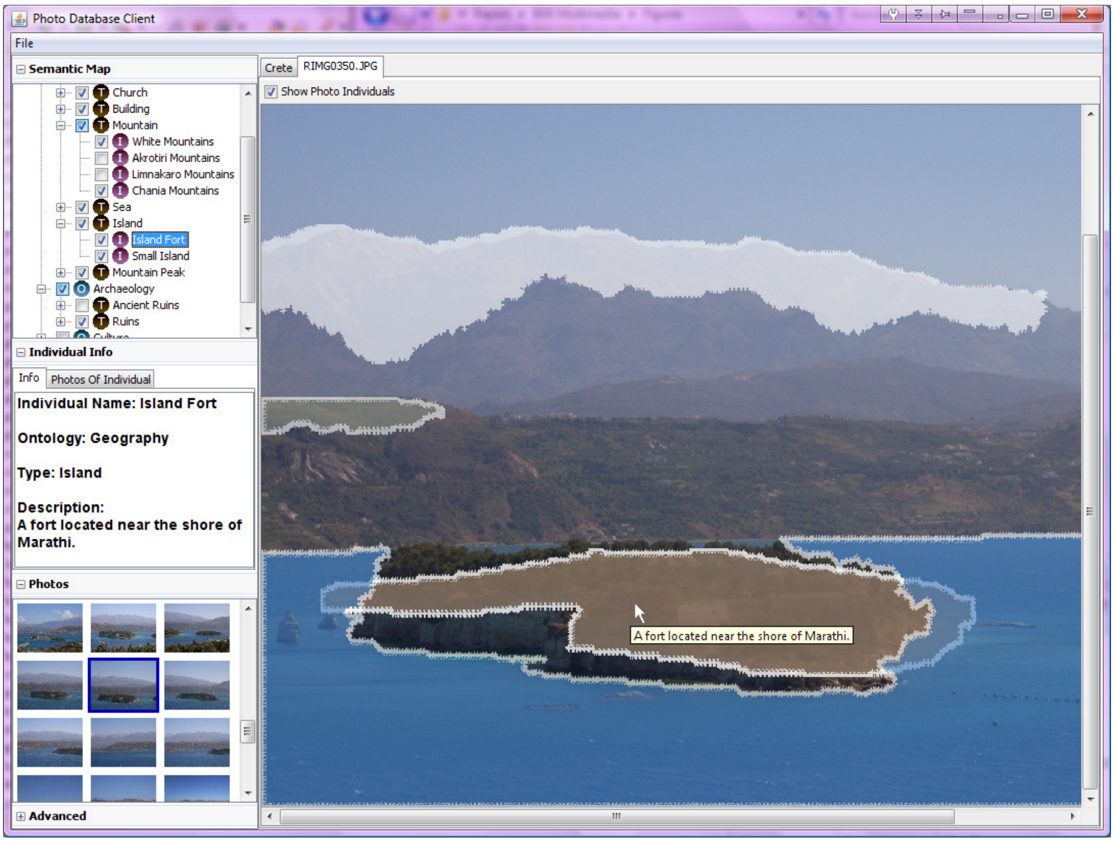
Description:
The mountains south of Chania.

FIGURE 30: PHOTO INFORMATION PANELS. THE "PHOTOS" PANEL SHOWS THUMBNAILS OF THE CURRENTLY OPENED PHOTOS, WHILE THE "ADVANCED" PANEL SHOWS INFORMATION ABOUT THE SELECTED PHOTO AND ENABLES THE PHOTO TO SEMANTIC MAP ALGORITHM TO RUN.

The advanced functions subpanel is divided into three tabs. The first tab ("Info") shows the metadata for the selected photo (both Exif and calculated parameters such as angle of view), the events and the persons associated with the photo. "Individuals" is a list of individuals that have been found to be in the photo image. The individual names can be selected from the list (and their information will be shown in the corresponding panel) and if the current semantic map contains them, they will also be highlighted on the map. The "View/Transform" tab allows at first a chance to view a larger thumbnail of the selected image. Pressing the "Calculate 2D Model" button begins the algorithm for the transformation of the photo to a semantic map (if the required parameters are in place). This procedure takes a while and is run in the background. There is also a progress bar to show the percentage of the algorithm's progress. The result of the transformation can be shown in the interactive photo-semantic map panel described below.

9.2.7. Interactive Photo-Semantic Map Panel

Clicking on the name of the selected photo on the top of the screen shows the image on the area previously occupied by the semantic map image. If the image has been transformed into a semantic map, the individual footprints overlaid on top of the image are also shown, as in Figure 31.



Individual Name: Island Fort

Ontology: Geography

Type: Island

Description:
A fort located near the shore of Marathi.

A fort located near the shore of Marathi.

FIGURE 31: INTERACTIVE EXPLORATION OF THE IMAGE CONTENTS. THE PANEL ON THE RIGHT SIDE ENABLES THE USER TO SELECT THE INDIVIDUAL FOOTPRINTS THAT ARE SUPERIMPOSED ON THE PHOTO AND VIEW INFORMATION ABOUT THEM.

The footprints are selectable and their information is shown in the semantic individual information panel on the left, just like when selecting them from any semantic map. When the mouse cursor hovers on top of a footprint, tooltip text with the individual description is revealed. If the user does not desire to view/select the footprint, he/she can uncheck the "Show Individual Footprints" checkbox and the photo will be shown in its original form without the footprints on top of it.

9.2.8. Other Client Functionalities

The rest of the functionalities of the SPIM client can are summarized below.

- Manual creation and management of user events and persons that participate in them.
- Automatic assignment of user events according to the date extracted from the Exif metadata and the begin/end date of the most suitable event.
- Connection with the SPIM server and retrieval of personalized semantic maps or other info concerning ontologies and semantic individuals.
- Automatic assignment of semantic individuals to photos according to their presence on the photo footprints and the 2D view model calculated.

9.3. Summary

This chapter described the architectures and functionalities of a prototype client/server system implementing the main ideas of the SPIM framework. Description of the client user interfaces was presented in detail. The programs were developed for demonstration purposes; however they supply enough functionality to be considered decent applications for the tourism domain, enabling the user to review the maps of places he/she wants to visit and download the maps to his/her computer before going on a trip. After the trip, the geo-referenced photos the user took will be visible on top of the map, ready for annotation and other processing.

10.Conclusions and Future Work

10.1. Conclusions

The generic framework presented in this thesis provides a means for integrated transparent management of semantic spatial multimedia information. The basic thesis of this work is that integration of inexpensive sensors with the camera is now feasible and it can result in semantic management of image contents and in interactive semantic exploration and visualization. The main advantages of the framework are its adoption of industrial standards and commonly used ontologies for the purpose of image annotation and the association of semantic map spatial information with the photos. Moreover, the use of external sensors, applications and services opens the path for more efficient understanding of the image context.

To capture the essence and ideas of the framework, a concrete implementation of a system that integrates a specific digital camera with digital GPS and compass direction sensors and a knowledge base system implementing semantic maps has been implemented and presented. The system uses automatically captured parameters by GPS and compass as well as contextual knowledge from 3D maps and geographic ontologies to produce good 2D representations of the scene visible in the direction of the camera. This gives a better chance for good registration of the 2D representation with the picture. After the registration, all the visible semantic objects can be used for the annotation of the picture automatically. The captured 2D information which contains the representations of the semantic objects (geographic, cultural, etc) has been used as a layer for interaction with the picture after the retrieval, effectively turning the picture into a semantic map.

Details about the algorithms for transforming the images into semantic maps have been presented and experiments have been conducted to prove their effectiveness, with satisfying results and visualizations.

In conclusion, the system provides a rich, transparent, and integrated functionality for managing a personal database of digital pictures and digital maps, in a semantic spatial information processing environment. The pictures are associated with semantic objects present in semantic maps, events defined by the users and persons participating in these events. The contents of the database can then be seen as an interactive living memory of the trips or activities performed by the users, even years after the completion of these events.

10.2. Future Work

As future work for this research, the following are recommended:

- Once modern technology allows it in a wider scale, experiments with an increased number of sensors should be conducted, allowing for more powerful 2D view models and giving better contextual parameters for the purpose of image annotation.
- One could use well-defined ontological languages such as OWL for the system's domain knowledge instead of a custom model. The same applies to the semantic map individual geometric representations in the sense that a well-structured language such as GML or KML should be used for their description.
- The system could take into account folksonomies, which are systems that use tags as a method for annotation and are a characteristic of Web 2.0 services. Even though tag-based annotations are always worse than ontology-based annotations, the huge amount of tagged and even geo-referenced images of services such as Flickr should not be ignored.
- The ability to represent persons (actors) within the images has not been exploited by the system. Portrait pictures and pictures with humans can be identified with current technologies and techniques such as face recognition. They could be used for the automatic or semiautomatic detection of specific persons and their silhouettes or faces.
- The distributed aspect of the framework can be further exploited by moving the computationally expensive algorithm workload to specialized servers and even provide remote services to the users without the use of a full heavyweight graphical user interface.

References

[1]. *3DEM Visualization Tool.* [Online] http://www.visualizationsoftware.com/3dem.html.

[2]. **Athanasiadis, T., et al.** *Using a Multimedia Ontology Infrastructure for Semantic Annotation of Multimedia Content.* Galway : 5th International Workshop on Knowledge Markup and Semantic Annotation at the 4th International Semantic Web Conference, 2005.

[3]. **Barnard, K., et al.** *Matching words and pictures.* Cambridge : Special issue on Machine learning methods for text and images, 2003. pp. 1107 - 1135. ISSN:1533-7928.

[4]. **Beveridge J. R., Balasubramaniam K., Whitley D.** Matching horizon features using a messy genetic algorithm. *Computer Methods in Applied Mechanics and Engineering.* June 9, 2000, Vol. 186, 2-4, pp. 499-516.

[5]. **Beveridge, J. R. and Riseman, E. M.** How Easy is Matching 2D Line Models Using Local Search? *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 1997, Vol. 19, 6, pp. 564-579.

[6]. **Blei, D. M., Ng, A. Y. and Jordan, M. I.** *Latent dirichlet allocation.* Cambridge : The Journal of Machine Learning Research, 2003. pp. 993 - 1022. ISSN:1533-7928.

[7]. **Blei, D.M. and Jordan, M.I.** *Modeling annotated data.* New York : 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, 2003. pp. 127–134.

[8]. **Boutell, M.** *Exploiting Context for Semantic Scene Classification.* University of Rochester. 2005. Doctoral Thesis. ISBN:0-542-05608-9.

[9]. **Boutell, M. and Luo, J.** Beyond Pixels: Exploiting Camera Metadata for Photo Classification. *Pattern Recognition.* Jul 2005, Vol. 38, 6, pp. 935-946.

[10]. **Cao, L., et al.** *Annotating Collections of Photos Using Hierarchical Event and Scene Models.* Anchorage : IEEE Conference on Computer Vision and Pattern Recognition, 2008. pp. 1-8. ISSN:1063-6919.

[11]. **Christodoulakis, S, et al.** *A modular approach to support GIS functionality in tourism applications.* Istanbul : Enter 98, 1998.

[12]. **Christodoulakis, S., et al.** *MINOTAURUS: A Distributed Multimedia Tourism Information System.* Edinburgh : Information and Communication Technologies in Tourism, 1997. pp. 295-306.

[13]. **Dupplaw, D., et al.** *Multimedia Distributed Knowledge Management in MIAKT.* Hiroshima : Knowledge Markup and Semantic Annotation, 3rd International Semantic Web Conference, 2004. pp. 81-90.

[14]. **Duygulu, P., et al.** *Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary.* London : 7th European Conference on Computer Vision, 2002. pp. 97-112.

[15]. *Exif Metadata Standard.* [Online] http://www.exif.org/specifications.html.

[16]. **Feng, S.L., Manmatha, R. and Lavrenko, V.** *Multiple bernoulli relevance models for image and video annotation.* Washington, DC : IEEE Computer Society Conference On Computer Vision And Pattern Recognition, 2004. pp. 1002-1009. ISSN:1063-6919.

[17]. *Flickr.* [Online] http://flickr.com.

[18]. **Franklin, W.R., Ray, C.K. and Mehta, S.** *Geometric Algorithms for Siting of Air Defense Missile Batteries.* Columbus : Technical Report DAAL03-86-D-0001, 1994.

[19]. **Gallagher, A. C., Luo, J. and Hao, W.** *Improved blue sky detection using polynomial model fit.* Singapore : International Conference on Image Processing, 2004. pp. 2367- 2370. ISSN: 1522-4880.

[20]. **Gallagher, A. C., Neustaedter, C. G., Cao, L. and Luo, J., Chen, T.** *Image annotation using personal calendars as context.* Vancouver : 16th ACM international conference on Multimedia, 2008. pp. 681-684. ISBN:978-1-60558-303-7.

[21]. *Geography Markup Language.* [Online] http://www.opengeospatial.org/standards/gml.

[22]. *Google Earth Virtual Globe.* [Online] http://earth.google.com/.

[23]. **Hare, J. S., et al.** *Bridging the semantic gap in multimedia information retrieval: top-down and bottom-up approaches.* Budva : 3rd European Semantic Web Conference, June 2006.

[24]. **Hare, J.S.** *Saliency for Image Description and Retrieval.* University of Southampton. 2005. Phd Thesis.

[25]. **Hollink, L., et al.** *Adding Spatial Semantics to Image Annotations.* Hiroshima : 4th International Workshop on Knowledge Markup and Semantic Annotation at ISWC'04, 2004. pp. 31-40.

[26]. **Hollink, L., et al.** *Semantic Annotation of Image Collections.* Sanibel : Workshop on Knowledge Markup and Semantic Annotation, KCAP'03, 2003. pp. 0-3.

[27]. **Hunter, J.** *Adding multimedia to the semantic web: Building an mpeg-7 ontology.* Stanford : International Semantic Web Working Symposium, 2001. pp. 261-281.

[28]. **Izraelevitz, D.** A Fast Algorithm for Approximate Viewshed Computation. *Photogrammetric Engineering and Remote Sensing.* 2003, Vol. 69, 7, pp. 767-774.

[29]. **Jaffe, A., et al.** *Generating summaries and visualization for large collections of geo-referenced photographs.* Santa Barbara : International Multimedia Conference, 2006. pp. 89-98. ISBN:1-59593-495-2.

[30]. *Java Programming Language.* [Online] http://java.sun.com/.

[31]. **Jeon, J. and Manmatha, R.** Using maximum entropy for automatic image annotation. [ed.] P.G.B. Enser, et al. 2004, Vol. 3115 of Lecture Notes in Computer Science, pp. 2040-2041.

[32]. **Jeon, J., Lavrenko, V. and Manmatha, R.** *Automatic image annotation and retrieval using cross-media relevance models.* New York : 26th annual international ACM SIGIR conference on Research and development in information retrieval, 2003. pp. 119-126.

[33]. **Joshi, D. and Luo, J.** *Inferring Generic Activities and Events from Image Content and Bags of Geo-tags.* Niagara Falls : Conference On Image And Video Retrieval, 2008. pp. 37-46. ISBN:978-1-60558-070-8.

[34]. *Keyhole Markup Language.* [Online] http://www.opengeospatial.org/standards/kml/.

[35]. **Koch, M.** Knowledge management and knowledge agents in Campiello. [ed.] B., Muller, H., Branki, C. Lees. *Proceedings of the Workshop on Intelligent Agents in CSCW.* 1998.

[36]. **Kuo, P., Aoki, T. and Yasuda, H.** *Building personal digital photograph libraries: An approach with ontology-based mpeg-7 dozen dimensional digital content architecture.* Washington, DC : Computer Graphics International, 2004. pp. 482-489.

[37]. **Lavrenko, V., Manmatha, R. and Jeon, J.** *A model for learning the semantics of pictures.* [ed.] S. Thrun, L. Saul and B. Scholkopf. Cambridge : MIT Press, 2004.

[38]. *Least Squares Fitting - Wikipedia.* [Online] http://en.wikipedia.org/wiki/Least_squares.

[39]. **Li, J. and Wang, J.** *Real-time computerized annotation of pictures.* Santa Barbara : 14th annual ACM international conference on Multimedia, 2006. pp. 911 - 920. ISBN:1-59593-447-2.

[40]. **Lim, J., Mulhem, P. and Tian, Q.** Home photo content modeling for personalized event-based retrieval. *IEEE Multimedia.* 2003, Vol. 10, 4, pp. 28-37.

[41]. *MPEG-7 - Multimedia content description standard.* [Online] http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm.

[42]. **Metzler, D. and Manmatha, R.** *An inference network approach to image retrieval.* Dublin : 3rd International Conference on Image and Video Retrieval, 2004. pp. 42-50.

[43]. **Mezaris, V., Kompatsiaris, I. and Strintzis, M.** Region-based image retrieval using an object ontology and relevance feedback. *EURASIP Journal on Applied Signal Processing.* 2004, Vol. 2004, pp. 886-901.

[44]. **Monay, F. and Gatica-Perez, D.** *On image auto-annotation with latent space models.* Berkeley : 11th ACM International Conference on Multimedia, 2003. pp. 275-278.

[45]. **Mori, Y. and Takahashi, H.** *Image-to-word transformation based on dividing and vector quantizing images with words.* Orlando : First International Workshop on Multimedia Intelligent Storage and Retrieval Management, 1999.

[46]. *Mote external sensor - Wikipedia.* [Online] http://en.wikipedia.org/wiki/Motes.

[47]. **Naaman, M.** *Leveraging Geo-Referenced Digital Photographs.* Stanford University. 2005. Doctoral Thesis. ISBN:0-542-29657-8.

[48]. *NASA SRTM Elevation data repository.* [Online] http://dds.cr.usgs.gov/srtm/.

[49]. *National Aeronautics and Space Administration.* [Online] http://www.nasa.gov.

[50]. *National Marine Electronics Association.* [Online] http://www.nmea.org.

[51]. **Nock, R. and Nielsen, F.** Statistical Region Merging. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 2004, Vol. 26, 11, pp. 1452-1458.

[52]. **O'Hare, N., et al.** *Combination Of Content Analysis And Context Features For Digital Photograph Retrieval.* London : 2nd European Workshop on the Integration of Knowledge, Semantic and Digital Media Technologies, 2005. pp. 323- 328. ISSN:0537-9989.

[53]. **Oliva, A. and Torralba, A. 2001.** Modeling the shape of the scene: A holistic representation of the spatial envelope. 2001, Vol. 42, 3, pp. 145-175.

[54]. **Oliva, A. and Torralba, A. 2002.** *Scene-centered description from spatial envelope properties.* London : Second International Workshop on Biologically Motivated Computer Vision, 2002. pp. 263–272. ISBN:3-540-00174-3.

[55]. **Oliva, A. and Torralba, A. 2003.** Statistics of natural image categories. *Network: Computation in Neural Systems.* 2003, Vol. 14, 3, pp. 391-412.

[56]. *Open Geospatial Consortium - Standards organization for geospatial information systems.* [Online] http://www.opengeospatial.org/.

[57]. **Platt, J.** *AutoAlbum: Clustering digital photographs using probabilistic model merging.* Fort Collins : IEEE Workshop on Content-based Access of Image and Video Libraries, 2000.

[58]. **Rattenbury, T. and Naaman, M.** Methods for Extracting Place Semantics from Flickr Tags. *ACM Transactions on the Web (TWEB).* January 2009, Vol. 3, 1.

[59]. **Rattenbury, T., Good, N. and Naaman, M.** *Towards Automatic Extraction of Event and Place Semantics from Flickr Tags.* Amsterdam : Annual ACM Conference on Research and Development in Information Retrieval, 2007. pp. 103-110. ISBN:978-1-59593-597-7.

[60]. *Ricoh Company, Ltd.* [Online] http://www.ricoh.com/.

[61]. **Said, M. A.** Polyline Approximation Of Single-Valued Digital Curves Using Alternating Convex Hulls. *Computer Graphics & Geometry Internet Journal.* 2002, Vol. 4, 2, pp. 75-99.

[62]. **Sarvas, R.** *Designing User-centric Metadata for Digital Snapshot Photography.* Helsinky University of Technology. December 2006. Doctoral dissertation.

[63]. **Sarvas, R., et al.** *Metadata creation system for mobile images.* Boston : 2nd International Conference on Mobile Systems, Applications and Services, 2004. pp. 36-48. ISBN:1-58113-793-1.

[64]. **schraefel, m., Karam, M. and Zhao, S.** *mSpace: interaction design for user-determined, adaptable domain exploration in hypermedia.* Nottingham : Workshop on Adaptive Hypermedia and Adaptive Web Based Systems, 2003. pp. 217-235.

[65]. **Schreiber, A.T., et al.** Ontology-based photo annotation. *IEEE Intelligent Systems.* 2001, Vol. 16, 3, pp. 66-74.

[66]. *Shuttle Radar Topography Mission.* [Online] http://www2.jpl.nasa.gov/srtm/.

[67]. **Smeulders, A.W.M., et al.** Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Anaysis and Machine Intelligence.* 2000, Vol. 22, 12, pp. 1349-1380.

[68]. **Smith, B. K., et al.** *Inquiry with Imagery: Historical Archive Retrieval with Digital Cameras.* Orlando : International Multimedia Conference, 1999. pp. 405-408. ISBN:1-58113-151-8.

[69]. **Spinellis, D.** Position-annotated Photographs: The Geotemporal Web. *IEEE Pervasive Computing.* 2003, Vol. 2, 2, pp. 72-79.

[70]. *sRGB - A Standard Default Color Space for the Internet.* [Online] http://www.w3.org/Graphics/Color/sRGB.

[71]. **Toyama, K., Logan, R. and Roseway, A.** *Geographic Location Tags on Digital Images.* Berkeley : 11th International Conference on Multimedia, 2003. pp. 156-166.

[72]. *Transformation Matrix - Wikipedia.* [Online] http://en.wikipedia.org/wiki/Transformation_matrix.

[73]. **Tsinaraki, C., Polydoros, P. and Christodoulakis, C.** Interoperability support between MPEG-7/21 and OWL in DS-MIRF. *IEEE Transactions on Knowledge and Data Engineering.* 2007, Vol. 19, 2, pp. 219-232.

[74]. **Tsinaraki, C., et al.** *Coupling owl with mpeg-7 and tv-anytime for domain-specific multimedia information integration and retrieval.* Avignon : Proceedings of RIAO, 2004. pp. 45-61.

[75]. **Vailaya, A., et al.** Image classification for content-based indexing. *IEEE Transactions on Image Processing.* Jan 2001, Vol. 10, 1, pp. 117 - 130.

[76]. **Volgin, O., Hung, W., Vakili, C., Flinn, J., Shin, K.G.** *Context-Aware Metadata Creation in a Heterogeneous Mobile Environment.* Stevenson, Washington : International Workshop on Network and Operating System Support for Digital Audio and Video, 2005. pp. 75-80. ISBN:1-58113-987-X.

[77]. **Yavlinsky, A., Schofield, E. and Ruger, S.** Automated Image Annotation Using Global Features and Robust Nonparametric Density Estimation. [ed.] W. K. Leow, et al. *Image and Video Retrieval.* 2005, Vol. 3568 of LNCS, pp. 507–517.

[78]. **Yu, J. and Luo, J.** *Leveraging Probabilistic Season and Location Context Models for Scene Understanding.* Niagara Falls : Conference On Image And Video Retrieval, 2008. pp. 169-178. ISBN:978-1-60558-070-8.