Geoinformatica (2014) 18:27–62
DOI 10.1007/s10707-013-0176-0

Active learning of user's preferences estimation towards a personalized 3D navigation of geo-referenced scenes

Christos Yiakoumettis · Nikolaos Doulamis · Georgios Miaoulis · Djamchid Ghazanfarpour

Received: 15 February 2012 / Revised: 24 December 2012 /
Accepted: 18 March 2013 / Published online: 12 April 2013
© Springer Science+Business Media New York 2013

Abstract The current technological evolutions enter 3D geo-informatics into their digital age, enabling new potential applications in the field of virtual tourism, pleasure, entertainment and cultural heritage. It is argued that 3D information provides the natural way of navigation. However, personalization is a key aspect in a navigation system, since a route that incorporates user preferences is ultimately more suitable than the route with the shortest distance or travel time. Usually, user's preferences are expressed as a set of weights that regulate the degree of importance of the scene metadata on the route selection process. These weights, however, are defined by the users, setting the complexity to the user's side, which makes personalization an arduous task. In this paper, we propose an alternative approach in which metadata weights are estimated implicitly and transparently to the users, transferring the complexity to the system side. This is achieved by introducing a relevance feedback online learning strategy which automatically adjusts metadata weights by exploiting information fed back to the system about the relevance of user's preferences judgments given in a form of pair-wise comparisons. Practically implementing a relevance feedback algorithm

C. Yiakoumettis · N. Doulamis (✉) · G. Miaoulis
Department of Informatics, Technological Education Institute of Athens, Ag. Spyridonos St., 122 10 Egaleo, Greece
e-mail: ndoulam@cs.ntua.gr

C. Yiakoumettis
e-mail: chgiak@teiath.gr

G. Miaoulis
e-mail: gmiaoul@hol.gr

N. Doulamis
National Technical University of Athens, 9 Heroon Polytechniou Str., 15773 Zografou, Athens, Greece

C. Yiakoumettis · G. Miaoulis · D. Ghazanfarpour
Department of Mathematics and Computer Sciences, University of Limoges, Xlim, 123, avenue Albert Thomas, 87060 Limoges Cedex, France

D. Ghazanfarpour
e-mail: ghazanfarpour@unilim.fr

presents the limitation that several pair-wise comparisons (samples) are required to converge to a set of reliable metadata weights. For this reason, we propose in this paper a weight rectification strategy that improves weight estimation by exploiting metadata interrelations defined through an ontology. In the sequel, a genetic optimization algorithm is incorporated to select the most user preferred routes based on a multi-criteria minimization approach. To increase the degree of personalization in 3D navigation, we have also introduced an efficient algorithm for estimating 3D trajectories around objects of interest by merging best selected 2D projected views that contain faces which are mostly preferred by the users. We have conducted simulations and comparisons with other approaches either in the field of on-line learning or route selection using objective metrics in terms of precision and recall values. The results indicate that our system yields on average a 13.76 % improvement of precision as regards the learning strategy and an improvement of 8.75 % regarding route selection. In addition, we conclude that the ontology driven weight rectification strategy can reduce the number of samples (pair-wise comparisons) required of 76 % to achieve the same precision. Qualitative comparisons have been also performed using a use case route scenario in the city of Athens.

Keywords Personalized path planning · Active learning · 3D scene exploration · GIS systems · Virtual tour · Scene understanding

1 Introduction

The recent advances in information and communication technology boost research towards the generation of personalized Geographical Information Systems (p-GIS) [7, 29, 30, 34] that allow for the users to precisely plan their trips and enjoy their stays in a place fully respecting their preferences [21]. In a p-GIS the "best route" is not defined only on geometrical criteria, such as the shortest distance or the travel time, but it should mostly exploit user's contentment, resulting in an itinerary that fully respects users' preferences (navigation through buildings / places of user's interest) [7, 29]. Most of existing personalized route guidance systems exploits a two-dimensional geometric and/or thematic representation; they estimate routes consisting of places/buildings of user's interest (thematic representation) within a near to the current user's position distance (geometric representation). However, currently three-dimensional navigation systems gain more popularity due to the fact that they provide the natural way for virtual touring [23, 43]. 3D representation boosts a series of new applications in the field of tourism, leisure and entertainment and cultural heritage [33]. It is argued that 3D virtual tours offer something more that the ordinary 2D ones. For instance, they can potentially explain a story or even show up the space [22]. Although 3D navigation imposes higher computational complexity compared to the 2D approaches, the current technological evolutions make feasible the implementation of 3D geo-information systems (see for example Google Earth and Microsoft Virtual Earth). This is the main reason, why we choose to focus on 3D navigation systems in this article.

In a p-GIS, user's preferences can be set manually or automatically [3]. Systems that apply the manual selection are called "adaptable personalized route guidance systems", while systems that use the automatic method are called "adaptive personalized route guidance systems" [46]. For both cases, the first necessary component is to extract a set of features (metadata) able to describe either the overall properties of a route (e.g., a small scenic road in contrast with a high speed freeway) or characteristics of places/buildings (we call them objects) that the route consists of. Then, a set of weights are assigned to regulate the degree of importance of the extracted metadata on route selection to provide personalization. For example, in case that a user is interested in an itinerary that includes buildings of modern architectural style, the weights that correspond to metadata of modern style take higher values than the remaining ones.

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

A Relevance Feedback scheme is used in this paper as the on-line learning strategy [37]. Relevance feedback is a powerful machine learning method, initially applied in traditional text-based information retrieval systems [35], to automatically adjust metadata weights by exploiting information fed back to the system about the relevance of previously selected routes [8, 14]. The relevance feedback component is combined in this paper with an *ontology driven weight rectification* module that improves the weight estimation process, especially in cases that few relevant/irrelevant routes have been selected by the user, considering metadata relationships.

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

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

(a) each 3D object and route is described through a set of metadata (features),
(b) users' preferences are modeled through a set of weights that regulate the degree of importance of the scene metadata on the route selection process,
(c) a relevance feedback machine learning strategy is adopted to automatically adjust metadata weights,
(d) an ontology-driven weight rectification mechanism is introduced to improve the weight estimation process and
(e) optimization strategies are incorporated either to suggest the most preferred routes with respect to user's information needs or to recommend best views and 3D trajectories for the objects of interest.

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

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

The ontology driven weight rectification component One main difficulty in implementing the on-line learning strategy concerns the number of pairs required to be submitted to get a reliable estimation of user's preferences. To reduce the number of comparisons needed and thus the number of user's interactions, in this paper, we introduce a novel weight rectification algorithm that exploits relationships among the extracted objects' metadata.

The macro-path layer: route planning component The goal of this component is to select the best preferred route for a user by taking into consideration metadata weights as well as several geometric and thematic criteria. As regards optimization, a genetic algorithm (GA) is considered. The genetic scheme provides a solution much closer to the optimal one, faster than other conventional optimization schemes. Additionally, GA provides scalability in contrast to the traditional shortest path methodologies that they always yield the same solution. Instead, in the genetic approach, we can improve route selection performance at the cost of demanding more computational resources.

The micro-path layer: best view selection component This module selects best 3D trajectories by merging projected 2D views of preferred 3D objects within an itinerary. A personalized entropy metric is considered that takes into account, apart from geometric objects properties (the volume that is projected from the 3D onto the 2D plane), the degree of importance of the projected 3D faces, as expressed through the metadata weights, onto the 2D plane.

1.1 Related work

The current personalized route planning systems use a set of explicit metadata (usually textual) to describe user's preferences. Then, they apply an optimization procedure that involves several multi-criteria factors to create the personalized route [29, 30]. However, two main limitations are encountered; (a) textual metadata are not able to describe the rich geographical content [14, 37] while (b) users' preferences cannot be modeled through a simple set of keywords [37]. Keywords are also inconsistent since they are dependent from the indexer. To address the aforementioned difficulties, the work of [30] introduces an ontological framework to provide a semantic description of the geo-referenced and contextual information, such as tourist attraction, road safety issues (telephone, medical assistance) and road facilities (gas station, services, terminal). Ontologies have been developed using artificial intelligence to facilitate knowledge sharing and reuse and to establish a domain of spatial information [4, 16, 17].

However, the work of [30] focuses only on the ontology representation and thus it fails to provide on-line learning strategies, which are necessary for a truly personalized route construction. Another approach introduces symbolic representation which is combined with visual properties of the objects, extracted through the application of image processing tools, to provide a cognitive spatial path planning [32]. However, again, this work is focused on a better organization of the metadata (textual or visual) that describes the geographical content, instead of applying personalized strategies able to capture the dynamics of the user's preferences.

Recently, the work of [31] has proposed a modified version of the Invasive Weed Optimization (IWO) algorithm for personalized route planning. Although the presented work provides an efficient framework for solving the personalized urban multi-criteria quasi optimum path problem compared to other conventional optimization strategies (such as the Genetic Algorithm), the approach of [31] does not incorporate on-line learning strategies in the optimization process, which are required for estimating the dynamics of user's preferences. More specifically, [31] focuses on the optimization algorithm as regards personalized route planning in contrast with the presented research where the focus is given on the Relevance Feedback component and the ontology driven weight rectification strategy to dynamically update the fitness cost function according to user's interaction. Additionally, the presented work introduces a hierarchical framework for route planning estimation; the macro path and the micro path layer optimization. The macro path is responsible for computing the best route in a way that exploits the user's preferences dynamics as provided by the relevance feedback component, while the micro path layer for selecting the best personalized views using on an entropy-based optimization. Furthermore, the approach of [31] has been designed for path estimation between two points, whereas our research focuses on the creation of a virtual tour of a city composing of multiple points of interest.

In [29], a generic model is proposed to calculate the most probable user preferred path using historical data, as a prior path planning, and an on-line strategy of time dependent routing selected using an approach of multiple criteria. The model integrates a pair-wise comparison method and quantifier-guided ordered weighted averaging (OWA) aggregation operators to form a personalized route planning. The main differences of the aforementioned research approaches with the one presented in this paper are:

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

In the following we briefly describe for completeness state of the art methodologies as regards individualized components involved in the proposed route guidance architecture.

Previous works in route planning A* and Dijkstra algorithms are probably some of the most popular route planning techniques [12, 19]. Since the complexity of these approaches increases with the increase of the size of the search space, near optimal solutions for path-finding have been proposed [6]. Demyen and Buro implement two modifications of A* by introducing an efficient triangulation-based path-finding scheme, namely Triangulation A* (TA*) and Triangulation Reduction A* (TRA*) [11]. TA* exploits the Delaunay triangulation to build a polygonal representation of the environment in order to reduce the path-finding search effort. On the other hand, TRA* is an extension of TA* that abstracts the triangle mesh into roadmap structures. On the other hand, Rogers and Langley implement a personalized driving route recommendation architecture based on the Dijkstra's shortest path algorithm, operating on personalized metadata that are approximated by a weighted sum of factors (such as driving time, length, number of turns, and number of intersections) [36]. In addition, Mekni and Moulin present a multi-agent implementation for path planning able to be solved in real time, even under constraints of limited memory and CPU resources [27]. In [26], the route is planned based on user's defined points of interest, which are manually set, on 3D virtual workspaces. Then, the route is controlled on a set of parameters and for each of them a weight is assigned to in order to regulate their degree of interest. The parameters are either pre-defined by the system or are set by the user [25]. Alternatively, the weight values can be defined by constraint satisfaction techniques [5]. Other approaches exploit Genetic Algorithms (GA) as regards estimation of the best route [1], with the focus, however, be on the Field Programmable Gate Array (FPGA) implementation of the GA in case of real-time controlling Unmanned Aerial Vehicles (UAV's). Another method is the Analytical Hierarchical Process (AHP) that allows a systematic evaluation of multiple criteria, resulting in a hierarchical structure that can help in personalized routing [38].

Previous works in best viewing algorithms Efficient approaches for calculating the best view of 3D scenes have been presented in the works of [41, 42, 44]. All these approaches exploit optimization strategies so that the best projected 2D views are selected in a way to cover as much as possible the whole 3D surface information of the scene. Each projected view is independently evaluated based on its complexity, measured as the total number of faces projected onto the 2D plane. In particular, [41, 42] optimizes the area of the visible surface over the sum of the area of the surface of the whole scene. Instead [44] introduces an entropy metric that measures the projection complexity; views that maximize this information are selected as the best ones. However, all the aforementioned approaches consider only geometric constraints preventing from a personalized implementation.

1.2 Contribution

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

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

1. In contrast with the current approaches dealing with personalized route planning, our architecture incorporates an active inductive-based learning methodology able to fill the gap between computers and humans by allowing high level concepts to be modeled in the future GIS decision support systems. Therefore, the proposed framework permits modeling of complex user's preferences through the use of metadata extracted from low level features.

2. Preferences are implicitly extracted without imposing users to be aware of any technical details regarding the metadata used in the geo-referred 3D environment. In particular, we introduce a relevance feedback machine learning strategy that allows an automatic adjustment of metadata weights, used to regulate the degree of importance of the features, by exploiting information fed back to the system in a form of preferences judgments of pair-wise comparisons. Then, we apply the pair-wise comparisons to get an overall object ordering in the scene.
3. Another contribution concerns the framework under which we use the ontological modeling. In this paper, we adopt the ontology not only for providing a human perceptive concept modeling of the properties of the 3D objects as the current approaches do, but we exploit metadata interrelations in order to estimate in a better way user's preferences though a few number of training samples are exploited. This is achieved by introducing a rectification strategy that corrects the weight estimation process by exploiting the knowledge provided through the ontology.
4. Optimal route planning is accomplished based on a multi-criteria framework that extends conventional shortest path methodologies enabling involvement of complex cost functions while simultaneously retaining scalability in computational processing. The latter permits direct implementation of the algorithm to terminal devices and communication networks of heterogeneous capabilities.
5. A hierarchical framework is adopted for route planning estimation. The framework comprises macro-path and micro-path layer optimization. In the macro-path layer, we optimally compute the best route from a start to an end point that includes the most preferred objects of a scene according to user's needs constrained with additional geo-referred criteria. In the micro-path layer, we estimate the best personalized view for an object of interest depicted in user's screen. The latter is achieved by extending the current entropy-based best view selection methodologies via a new personalized entropy metric, that measures not only the geometric properties of a projection but also personal preferences.

This paper is organized as follows: Section 2 presents an overview of the proposed architecture and the basic requirements of our system. Section 3 describes the metadata (quantitative and qualitative) used to represent scene properties and the ontology. Section 4 reports on machine learning algorithms adopted to order 3D objects according to user's preferences and then to estimate the significance of the metadata. In Section 5, we describe a multi-criteria route planning approach based on a GA optimization scheme, while Section 6 discusses the personalized entropy metric for selecting the best view and the personalized 3D trajectories around selected objects. Finally, implementation issues and simulation results are presented in Section 7, while Section 8 concludes the paper.

2 System overview and the main flows of the algorithms

Figure 1 presents an overview of the proposed personalized route guidance system. As is observed, the proposed architecture comprises of five different modules; 1) *the 3D object modeling and metadata component, 2) the relevance feedback component*, 3) *the ontology driven weight rectification component, 4) the optimal macro-path layer: route planning component and 5) the micro-path layer: best view selection component.* In Fig. 1, we have also depicted the processing steps that take place within each individualized component (as rounded rectangle boxes the number of which indicates the order of the respective processing step). In Fig. 1, could see all the main components of the proposed personalized 3D navigation system along with the order of execution of the respective processing units.

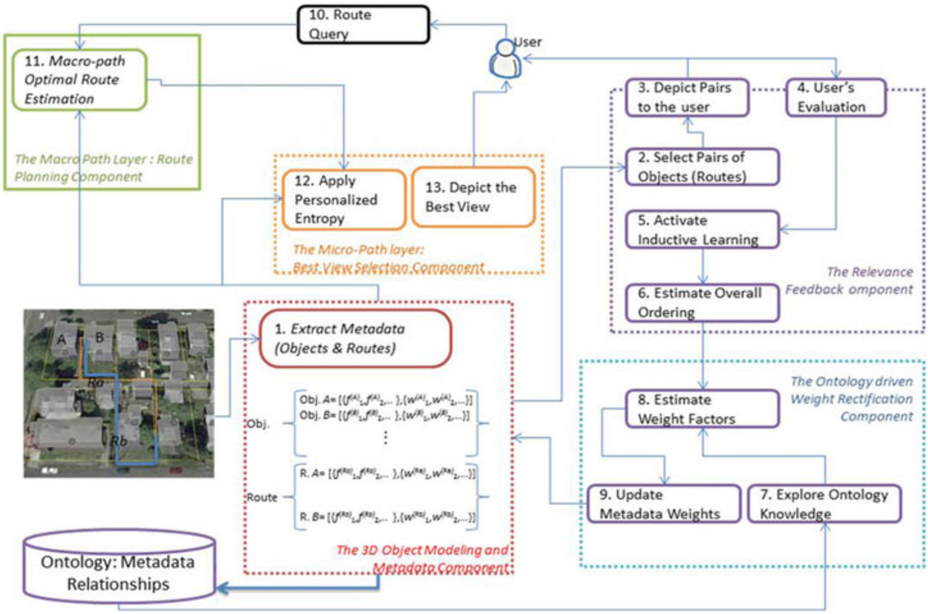


Fig. 1 An overview of the proposed personalized route planning architecture

Figure 2 presents the flowchart of the proposed adaptive personalized route guidance system. Two different procedures are described. The first (on the left side) refers to the route

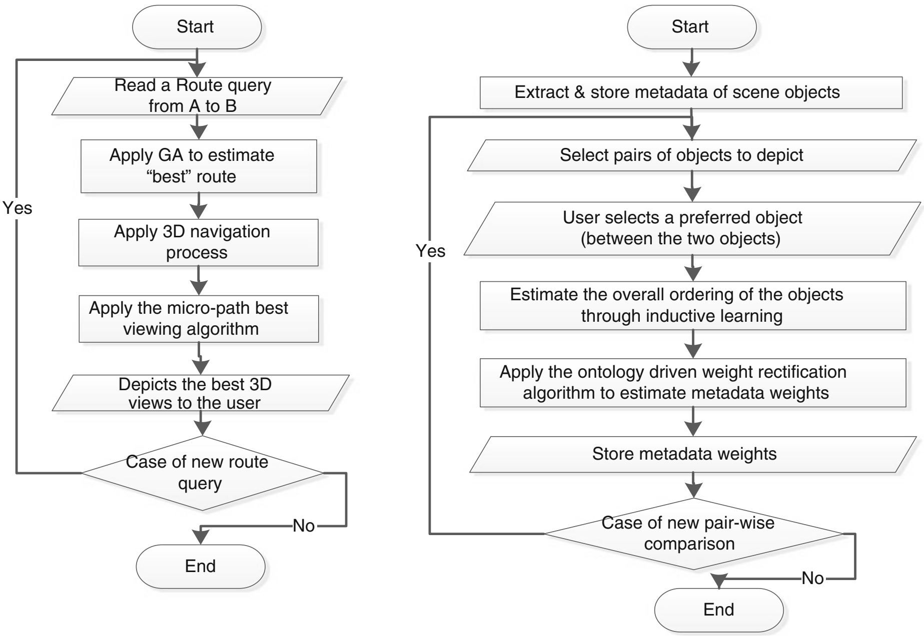


Fig. 2 The flowchart of the proposed adaptive personalized route guidance system; **a** the flowchart concerning the route and the best view selection procedure (macro-path/micro-path layer optimization), **b** the flowchart concerning estimation of the weight factors used to approximate user's preferences

and the best view selection process. The user selects a start and an end point and the system responds with the best preferred route. Then, within a selected itinerary and for the selected objects of interest, the system returns the best views composing a personalized 3D trajectory around objects of interest. The second procedure described in the right side of Fig. 2 shows the main steps of the estimation of the metadata weights used to approximate user's preferences. In the following, we describe the flows of the basic components of the proposed platform.

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

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

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

Optimal route planning (macro-path layer) flows An efficient creation of a personalized 3D geographical path includes, apart from the 3D objects (such as buildings) that satisfy as much as possible user's information needs, additional criteria, like for example, the distance of the path, aesthetic factors of a route (e.g., route with parks, or with bars, or shops) and the services offered over the path. All these multi-criteria factors are included in a common formula for optimization in order to extract "*the best*"-with respect to user's preferences- route. The main bottleneck of such multi-criteria optimization is that it is computationally intensive, since all possible combinations should be exhaustively examined [29]. In addition, such approaches lack scalability in computational processing. Different types of client devices, such as mobile phones, PDAs or laptops present quite different computational capabilities and therefore they require scalable algorithms in the estimation of the personalized 3D paths.

To address these difficulties, we introduce in this paper a genetic solution for the selection of the most preferred route. GA incorporates quite well with the adopted multi-criteria approach, yielding solutions close to the global optimum within an acceptable time of operations. Another advantage is its scalability; the performance is regulated by the number of operations in the sense that the more operations we apply the better solution we yield in contrast with conventional algorithms the solution cannot be improved by demanding more computational power.

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

3 Scene modeling and metadata

3.1 The scene metadata

Scene metadata are classified into those ones associated with 3D objects -involved in the construction of the 3D world- and the ones related with the route. As far as objects' metadata are concerned, two different types are taken into account; the qualitative and quantitative ones. Qualitative metadata include the type of building use, building style, construction material, building location, attractive content and the type of city block. Qualitative metadata are expressed in numeric form through a set of discrete values arranging in our case from 0 to three [0,3] (0-none, 1-low/few, 2-medium/several, 3-high/many). On the other hand, examples of quantitative metadata include the building cost of visit, interactivity, popularity, age and geometry. For instance, the "interactivity" metadata are defined as the number of clicks that a user has been made on a 3D object by selecting it within an itinerary in order to enjoy a 3D navigation around it. On the other hand, 'popularity' descriptor indicates the fame of a building as being important on a specific area and is defined as the average clicks over all users.

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

An N-dimensional feature vector $\mathbf{f}_u$ is used to gather all the metadata of the u-th 3D object in a scene. The qualitative metadata are also included in the feature vector $\mathbf{f}_u$ as discrete values in a way that we have discussed previously. Besides feature vector $\mathbf{f}_u$, we denote as $\mathbf{f}_{u,v}$ the feature vector that includes the metadata of a route that starts from object u and ends to object v.

3.2 Metadata interrelations through ontology

All the aforementioned extracted metadata (qualitative and quantitative) are semantically interrelated with each other through the use of an ontology model. As we have stated above, ontology has a dual purpose. On the one hand, it provides a framework for a more efficient users' preferences estimation since it improves the weight estimation though a small number of preferences judgments (given in the form of pair wise comparisons) are fed back to the system.

The second purpose of the ontology is to model more complex concepts paving the way for a human-perceptive navigation. For example, one can model, apart from the use of a building (through the "building used for" metadata, see Fig. 3), the scope of visiting in a building (e.g., educational scope) and what is expected for the visitor to gain from such a visit. An example of creating more complex concepts through metadata interrelations is presented in Fig. 4, where we define semantic properties of a 3D object enclosed around a bounding box.

4 Active inductive-based learning process

The adopted *active inductive-based learning* strategy comprises of the following basic steps; i) data ordering, ii) estimation of metadata weights and finally iii) weight rectification driven by the ontology model.

> *Step 1 – Data Ordering*: We initially construct a small set of representative objects (e.g., buildings). Then, two of them are depicted to the user for a pair-wise comparison. In this way, we are able to construct directed (non-symmetric) graphs, the nodes of which represent the objects of the training set, while the edges the pair-wise preference

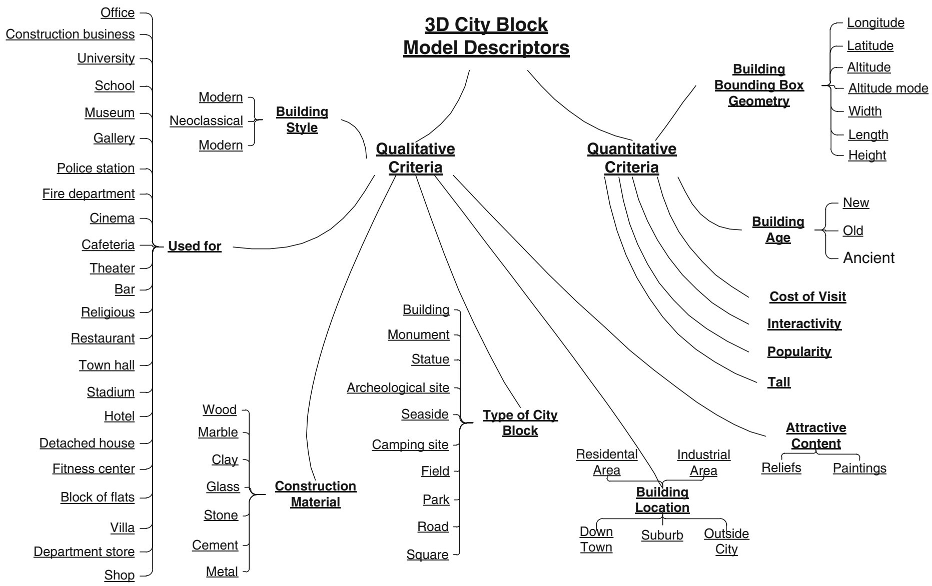


Fig. 3 The objects metadata used in the proposed route guidance system

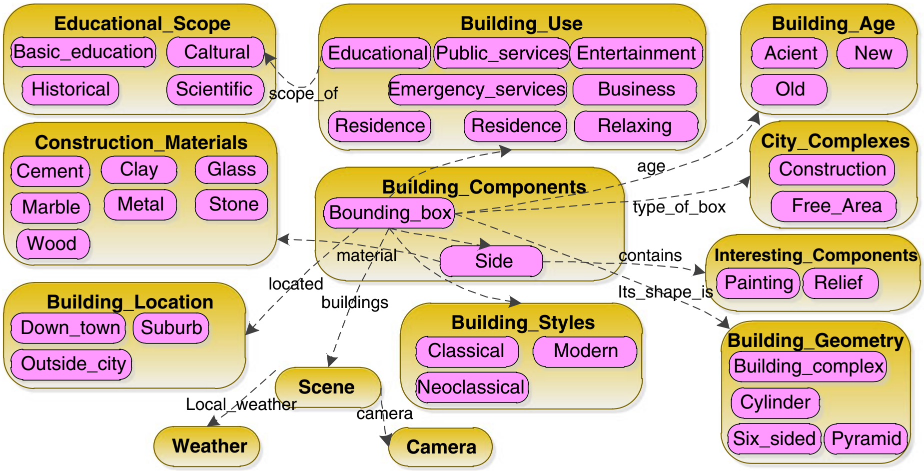


Fig. 4 The ontological model used in the proposed architecture

distance between the two objects. It can be proven that this pair-wise comparison is adequate to order the preferences for all objects in the training set (see Section 4.2).
Step 2 – Estimation of the Metadata Weights: In this step, we exploit the object ordering for all samples in the training set (see Step 1) to automatically define the metadata weights for all objects of the scene, either within or outside the training set (see Section 4.3).
Step 3 – Ontology driven weight Rectification: In the last step, we improve the initially estimated metadata weights (see Step 2) by considering metadata relationships defined through the ontology model (see Section 4.4). This step allows reliable weight estimation even in case that few training samples are used, minimizing user's interaction. The following Table 1 summarizes the main steps of the active inductive-based learning algorithm.

4.1 Pair-wise comparisons

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

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

where $u, v \in S$ are two selected objects. Function $g(u, v)$ defines the order between the object pair (u,v) with respect to user's preferences. More specifically, $g(u, v) \rightarrow 1$ (or respectively

Table 1 Main steps of the proposed active inductive-based learning algorithm

Active Inductive-Based Learning Algorithm:
1. Construct a small set of representative 3D objects
2. For every new selected Object do
a. Update the training set
b. Apply algorithm 1 as described in Table 2
c. Apply the algorithm 2 as described in Table 3
d. Activate the algorithm 3 (see Table 4)

$g(u, v) \rightarrow 0$) is interpreted as a strong recommendation of being u more preferred (or less preferred) than v. On the other hand, a value around ½ indicates an abstention of making any recommendation.

As we have stated above, we put the user in the process loop by allowing him/her to judge which of the two depicted 3D objects is more preferred and feeding this information back to the system. Function $g(\cdot)$ is sufficient to create an overall ordering for all 3D objects (e.g., buildings) within the training set S. However, if we had a social group, we would need to perform a weighted average policy for all values of $g(\cdot)$ to estimate the group's mean preference function [15, 24].

To do this, we need to construct a graph, the vertices of which are the objects within the training set S, while the edges represent the value of the ordering function $g(u, v)$ for the object pair (u,v). Figure 5 presents an example of such a graph assuming a training set of four objects as well as steps of the greedy overall ordering algorithm.

4.2 Estimation of the overall ordering

As mentioned in [9], NP-hard is the problem to derive the overall order for all samples within the training set S, from the graph of the preference functions between two objects. In computational complexity, NP hard refers to a non-deterministic polynomial-time problem (combinatorial optimization), which requires exponential time to be solved, that is, estimation of the optimal ordering among all 3D objects of S, is practically unaffordable to be found.

To overcome this problem, we present an approximate algorithm. For each vertex say u, we assign a value that indicates the difference between the outgoing and ingoing edges. We denote this value as $\beta(u)$ in the following (see the example of Fig. 5 for more details). Large value for the outgoing metric means that object u is highly preferred by the user while the opposite is valid for the ingoing metric (large values refer to objects of low significance to the user). In other words, for each vertex of the graph, that is, for each object u, we have to estimate the following metric

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

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

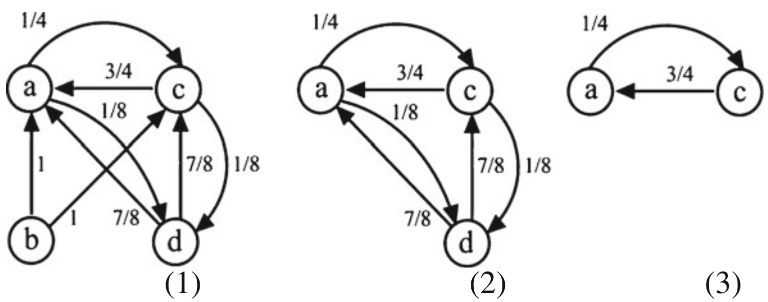


Fig. 5 (1) An example of preference function graph for objects **a**, **b**, **c** and **d**, (2) and (3) illustrate the greedy overall ordering of objects **a**, **b**, **c** and **d** given their preference functions (source [8])

selects from the graph as the most suitable object u the one that maximizes $\beta(u)$ for all objects within the training set S. In other words, the selected node is set to be ahead from all the remaining nodes of the graph. In next step, we remove this node (that is, the respective object) from the graph and then we update the potential values of all the remaining objects (vertices) within the graph. This process is repeated until no nodes are left in the graph as illustrated in Fig. 5. Table 2 outlines the pair-wise comparison and overall data ordering algorithm.

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

4.3 Estimation of the metadata weights

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

Table 2 The main steps of the algorithm used for pair-wise comparisons and data ordering

Algorithm 1: Data ordering: *Returns for all objects in the training Set ($u \in S$) the preference value* $\beta(u)$

1. For every object $u \in S$ in the training set S do
   a. Select an object pair, say $u, v \in S$
   b. Depict this pair to the user and get an estimate of the function $g(u, v){:}S\times S -> [01]$
2. Construct a graph $G^{(0)}$v,$e\}$ the vertex of which are the objects in the training set and the edges the values of $g(u, v){:}SxS -> [01]$
3. Estimate for every vertex of the graph the metric $\beta^{(0)}(u)$ [see Eq. (2)]
4. Set $n \leftarrow 0$;
5. While (graph $G^{(n)}$v,$e\}$ not empty) do
   a. Selects the vertex $\widehat{v}$ of $G^{(n)}$v,$e\}$that corresponds to the maximum value of $\beta^{(n)}(u)$, $\widehat{v} = \max_{u \in G} \beta^{(n)}(u)$
   b. Assign to the object $\widehat{v}$ of the training set S, the value of $\beta^{(n)}(\widehat{v})$
   c. Update Graph $G^{(n+1)}\{v, e\} = Graph\_Update\ \left(G^{(n)}\{v, e\},\ \widehat{v}\right)$
   // Function $G^{(n+1)}\{v, e\} = Graph\_Update\ \left(G^{(n)}\{v, e\},\ \widehat{v}\right)$ removes vertex $\widehat{v}$ from the graph and update the graph.
   d. Update metric $\beta^{(n+1)}(u) = Update_Metric(G^{(n+1)}$v, $e\})$
   // Function $Update_Metric(G^{(n+1)}$v, $e\})$ updates values of $\beta^{(n)}(u)$ of the graph $G^{(n+1)}$v, $e\}$
   e. Increase n, $n \leftarrow n+1$
   f. go to step 5.
6. Set $\beta(u) \leftarrow \beta^{(n)}(u)$

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

$$S(u) = \sum w_u^j f_u^j \tag{3}$$

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

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

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

then we have that

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

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

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

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

$$w^j = \frac{1}{std\left(f_{u_{i_1}}^j, f_{u_{i_2}}^j, \ldots, f_{u_{i_{\widehat{K}}}}^j\right) + \delta} \tag{7}$$

where the $std(\cdot)$ denotes the standard deviation operator, while the $f^j_{u_{im}}$,m=1,2,…,$\widehat{K}$ the j-th element of feature vector $\mathbf{f}_{\mathbf{u_{im}}}$. Equation (7) means that feature elements that share similar values among all the $\widehat{K}$ objects lead to large weight values (small standard deviation) since in this case the respective feature element seems to be consistent with respect to user's preferences. On the other hand, large deviation values indicate no significance for the respective feature element. To avoid instability issues occurred by a zero division, we have added a small constant value δ to the denominator of (7). Table 3 presents a summary of the proposed feature weight estimation algorithm.

4.4 Weight rectification driven by the ontology model

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

Let us denote as $\mathbf{A}$ a matrix that contains the interrelations values among all feature elements. Matrix $\mathbf{A}$ stems from the ontology model described in Section 3. Let as also denote as $\mathbf{w}(0)$ a vector that contains all the initial weight factors w^j as estimated by (7). Then, at the n-th iteration stage, the weights $\mathbf{w}(n)$ are updated as

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

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

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

Table 3 The main steps of the algorithm used for the estimation of the feature weights

Algorithm 2: Feature weight estimation: *Returns initial feature weights*

1. Using the Results of the Algorithm 1: (Algorithm 1 is applied for all objects in set S and it gets preference values of $\beta(u)$)
2. Rank the objects in the training set S, according to the values of $\beta(u)$
3. Set $a \leftarrow$ *UserDefined* and $K \leftarrow 1$
4. Estimate $E(K)$ – see Eqs. (4) and (5)
5. While $E(K) < a$ do
   a. $K \leftarrow K + 1$
   b. Update $E(K)$, $E(K) = Update\_E(E(K-1), K)$
6. Set $\widehat{K} \leftarrow K$
7. Select the feature vectors of all the $\widehat{K}$ ranked objects, $\left\{\mathbf{f}_{u_1}, \mathbf{f}_{u2}, \ldots, \mathbf{f}_{u\widehat{K}}\right\}$
8. Estimate weight factors for all elements of the feature vectors according to the Eq. (7)

In Eq. (9), vector $\mathbf{w}^r$ contains all the new rectified weight (see Table 4 for the algorithmic form).

5 The macro path layer-route planning

5.1 Cost function

Let us define r as a route. Route r starts from a point (object) SP and ends to a point (object) FP. We assume that the route includes M out of the N available objects in the scene, apart from SP and FP. For instance, assuming that we have a scene of $N=10$ objects and that we would like to construct a route consisting of the $M=4$ most representative objects, starting from the object SP and ending to the object FP then, route r can be any of the combinations $Comb(4,10)$, say, for example, the SP,1,3,7,9, FP. Let us also assume that for each of these N objects, features (metadata) have been extracted and metadata weights have been estimated using the learning methodology of Section 4. Then, a preference cost function $S(u)$ is computed for each object u of the scene, using Eq. (3), that measures how preferred is the u object for a particular user. It is clear that different users correspond to different values of $S(u)$ for the same object u.

Finding the optimal route using as cost function only the scores $S(u)$ is a straightforward problem since the only thing we need to do is to select the objects with the highest $S(u)$ values. This selection, however, is not adequate due to the fact that additional criteria should be included in the optimization process (see Section 2). This leads to a minimization of an aggregate cost function consisting of multiple criteria. These multi-criteria cost functions are discriminated into two main categories; the ones that refer to the overall properties of the route and the ones that refer to objects properties. Examples of cost criteria regarding a route are the route distance or aesthetic factors of a route such as a route with many parks or a route with shops. The first criterion can be extracted from distance maps while for the second we exploit emotional maps of a city. We denote in the following as $C_k(r)$ the k-th criterion that describe a route. On the other hand, cost criteria that describe objects properties are defined through the preferences cost function $S(u)$. For each route/object criterion, a weight factor is assigned, say γ_i, to determine the importance of this criterion to the overall aggregate cost function. The fitness function is given by the following equation.

$$C = \gamma_1 \cdot C_1(r) + \gamma_2 \cdot C_2(r) + \ldots + \gamma_m \cdot S(\vartheta(r,1)) + \gamma_{m+1} \cdot S(\vartheta(r,2)) + \ldots \tag{10}$$

where $\vartheta(r, m)$ is an operator that returns the m-th object of route r.

Table 4 The main steps of the weight rectification algorithm driven by the ontology model

Algorithm 3: Weight rectification: *Updates the weights using the ontology model*
1. Form the matrix **A** from the ontology
2. Set a weight vector **w**(0) equals to that provided by the Algorithm 2: Feature Weight Estimation (see Table 3)
3. Estimate eigenvectors of matrix **A**, say **U**
4. Update the weight factors as Eq. (9)

Then, the optimal route is retrieved by minimizing Eq. (10), i.e.,

$$r = \underset{all\ possible\ r}{\arg\min}\ C \tag{11}$$

It is clear that Eq. (11) is a combinatorial problem due to the fact that all possible routes should be examined. Therefore, minimization of (11) is not a straightforward task, especially in real-world cases where a large number of N (objects in the scene) and M (objects in the route) are encountered.

One approach for solving the aforementioned multi-criteria minimization is to model it as a travelling salesman problem [1]. In the travelling salesman the goal is, given a list of cities (objects in our case), to find the shortest possible route that one visits each city exactly once. In our problem formulation, the shortest possible route refers the minimization of the multi-criteria cost function as the one presented in (10). However, again the complexity of this problem is NP-hard, meaning that it is difficult to compute the optimal solution by estimating all possible combinations. Therefore, approximate heuristic solutions have been proposed in the literature [1]. Examples include the nearest neighbor algorithm, which is a greedy approach, that lets the salesman choose the nearest unvisited city as next move. Although, this algorithm quickly yields a short route solution, the performance is not so satisfactory especially in our case, where several multi-criteria cost functions are involved (see Section 7.4). Additionally, such an approach lacks scalability. The algorithm produces always the same solution even in case that we can tolerate more computational effort to improve the optimization performance. There are also several other approximated heuristic algorithms, such as the k-opt or v-opt heuristic with similar limitations such as the nearest neighbor algorithm. For this reason, in this paper, we use a genetic algorithm to solve Eq. (11). Genetic algorithm is a promising solution in case of NP-hard optimization problems. It provides a framework for a guided and scalable search, reducing the number of combinations needed to obtain the optimal solution.

Recently, a modification of the invasive weed optimization algorithm has been proposed as an efficient tool for solving the personalized multi-criteria path planning problem. The modified invasive weed optimization algorithm seems to present advantages compared to traditional optimization strategies and thus it can be considered as an alternative methodology to the genetic algorithm regarding the best route selection. However, it should be mentioned that the presented work does not focus on introducing a new efficient optimization algorithm for best route selection. Instead, the main research is focused on the relevance feedback algorithm, which automatically adjusts metadata weights by exploiting information fed back to the system about the relevance of user's preferences judgments given in a form of pair-wise comparisons, and the ontology driven rectification strategy that improves the estimation of the user's preference dynamics by exploiting metadata interrelations defined through an ontology.

5.2 The genetic algorithm

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

Then, appropriate "*parents*" are selected so that a fitter chromosome (i.e., a route that further decreases the aggregate cost function *C*) is generated at next iteration. This achieved by assigning to each chromosome a probability equal to $C(r_i(n)) \Big/ \sum_{i=1}^{L} C(r_i(n))$ and then we select the next *L* chromosomes as candidate parents based on their assigned probabilities through the roulette *wheel selection* procedure [28].

A set of new chromosomes (offspring) is then produced by mating the genetic material of the parents using a *crossover* operator, which defines how the genes should be exchanged to produce next generations. The next step of the algorithm is to apply *mutation* to the newly created chromosomes, introducing random gene variations that are useful for restoring lost genetic material, or for producing new material that corresponds to new search areas. In our case, the genetic material of a chromosome is randomly mutated with a probability of p_m. After that, the next population $P(n+1)$ is created by inserting the new chromosomes and deleting the older ones. Several GA cycles take place by repeating the procedures of fitness evaluation, parent selection, crossover and mutation, until the population converges to an optimal solution. Table 5 shows the algorithmic form of the genetic-based route optimization method.

6 The micro path layer- best view selection

The aforementioned algorithm estimates a route that it is more comfortable by a specific user according to his/her preferences and additional multi-criteria factors. Once a route is created (macro-path layer optimization-Section 5), the next question is which is the best view of a 3D building projected onto the 2D monitor screen to allow users to perform a personalized 3D navigation around objects of interest. An entropy-based algorithm has been adopted in this paper to estimate the best object views.

6.1 Entropy-based best view selection

According to Shannon's theory [40], the entropy of a discrete random variable *X* with values in the set $\{a_1, a_2, \ldots, a_n\}$ is given by

$$H(X) = -\sum_{i=1}^{n} p_i \log_2 p_i \tag{12}$$

where $p_i = \Pr(X = a_i)$ is the probability of the random variable *X* to take the value of a_i and the $\log_2 p_i$ the 2-base logarithm of p_i. For continuity reasons the $p_i \log_2 p_1 = 0$ in case of p_i= 0. Similarly, if we assume a set of *k* faces for a particular object *u*, we can use as probability distribution the relative area of the projected faces over the sphere of directions centered in the viewpoint *p*, that is [13, 45]

$$H(\xi, u) = -\sum_{i=0}^{k} \frac{A_i}{4\pi} \log \frac{A_i}{4\pi} \tag{13}$$

where A_i is the projected area of face *i,* while ξ denotes the view (2D plane) over which the 3D object is projected onto. Equation (13) implies a spherical coordinate system and thus $A_i/4\pi$ indicates the visibility of the *i*-th face with respect to the point *p*.

The main difficulty of Eq. (13) is that the projected areas A_i cannot be straightforwardly computed. For this reason, in this paper, we adopt an approximate calculation based on

Table 5 The main steps of the genetic algorithm for route planning

Algorithm 4: The genetic algorithm for route planning
1. Define the Start and Finish Point, *SP* ← *UserDefined*, *FP* ← *UserDefined*
2. Set the number of objects in the route *M* ← *UserDefined* and set *n* ← 0
//Produce *L* different route combinations,
3. For *i*=1 to *L* do // *L* is the number of chromosomes
a. *r*[*n*][*i*] = *Set_Route*(*SP*, *Rand*(*M*, *N*), *FP*)
//The function Rand(*M*, *N*) returns *M* randomly selected numbers between 1 and *N*.
4. Set *Best_Cost* ← *Maximum_Value* and *Iteration* ← 0
5. For i=1 to *L* do //*i* indexes the number of routes
//Estimate the Cost of the Objects in the Scene
a. Set *Total_Cost_of_Objects* ← 0
b. For *j*=1 to *M*+2 do //*j* indexes the number of objects
i. u = Retrieve_Object(*r*[*n*][*i*], *j*) //The function Retrieve_Object(*r*[*n*][*i*], *j*) returns the *j*-th element from the combination *r*[*n*][*i*]
ii. Set *S*(*j*) ← *Eq.* (3) //Estimate cost function of the *u* object using Eq. (3)
iii. *Total_Cost_of_Objects* ← *Total_Cost_of_Objects* + *S*(*j*)
//Estimate the Cost of the Route
c. Set *Total_Cost_of_Route* ← 0
d. For *m*=1 to Number_of_Cost_Functions do
i. Estimate the cost function of the route *r*[*n*][*i*], *C*[*m*] = *Cost_Function*(*r*[*n*][*m*],*m*); *Cost_Function* returns the *m*-th cost for the route *r*[*n*][*m*].
ii. *Total_Cost_of_Route* ← *Total_Cost_Route* + *C*[*m*]
//Estimate the Overall Cost
e. *Total_Cost* ← *Total_Cost_Route* + *Total_Cost_Objects*
//Estimate the Best Route
f. If (*Total_Cost* < *Best_Cost*) do
i. *Best_Cost* = *Total_Cost*
ii. *Best_Route* = *r*[*n*][*i*]
// Crossover the *L* routes to generate *L* new routes
6. For *i*=1 to *L* do
a. *r*[*n*][*i*] ← *mix*(*r*[*n*][1], …, *r*[*n*][2]) // Crossover the elements of *r*[*n*][*i*] to perform new routes
// Apply mutation to the *L* new generated objects
7. For *i*=1 to *L* do
a. *r*[*n*][*i*] ← *mutate*(*r*[*n*][*i*])// Randomly change the elements of *r*[*n*][*i*]
8. If (*Iteration* < *Max_Value*) Go to Step 5.
9. Else Stop and set *r* ← *Best_Route*

number of pixels of the i-th face projected onto a 2D plane over the total number of object pixels. In this sense, Eq. (13) can be written as

$$H(\xi, u) = -\sum_{i=0}^{k} \frac{NP_i}{NP_{total}} \log \frac{NP_i}{NP_{total}} \tag{14}$$

The problem with Eq. (14) is that the entropy at a particular view ξ is not related with user's preferences. For this reason, we extend Eq. (14), in this paper so as to include user's preferences in the best object view selection process.

In particular, let us recall that user's preferences are modeled through the weights $\mathbf{w}^r$ [see Section 4 and Eq. (9)]. In the following, we omit superscript (r) for simplification purposes, since we assume that all the weight values have been properly rectified by the ontology. The weight element w_u^j expresses the degree of importance for the j-th feature element of the u-th 3D object. Since the j-th feature element refers to one of the multiple faces that 3D object is decomposed to, we can derive a personalized entropy using the values of w_u^j. In particular, let us denote as $\psi(u,f)$ an operator that returns the weight indices which are related with those metadata that describe the f-th *face* of the u-th object. For example, let us assume a 3D object the material of one face of which consists of gypsum and the material of the other face of marble. Then, operator $\psi(\cdot)$ returns the weight indices referring to the marble feature element only in case that the specific face has this property. Therefore, the personalized entropy for a particular object u is defined as follows

$$H(\xi,u) = -\sum_{i=0}^{k} \frac{NP_i}{NP_{total}} \cdot \frac{\sum\limits_{for\ \text{all}\ n\in\psi(u,i)} w_u^n}{\sum\limits_j w_u^j} \log\left(\frac{NP_i}{NP_{total}} \cdot \frac{\sum\limits_{for\ \text{all}\ n\in\psi(u,i)} w_u^n}{\sum\limits_j w_u^j}\right) \tag{15}$$

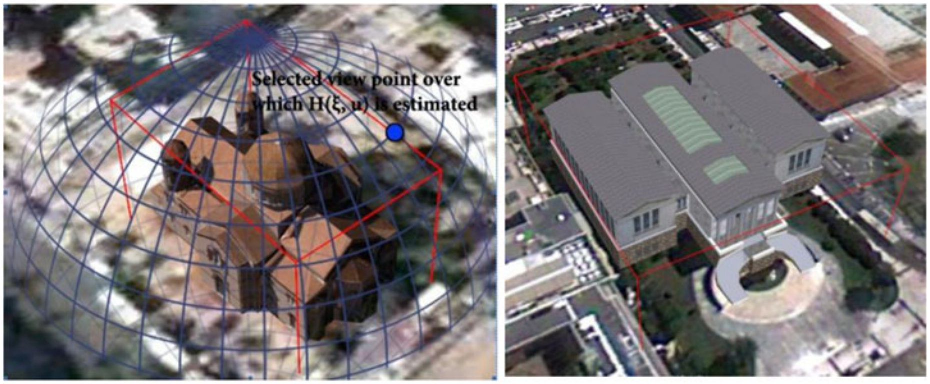
Using Eq. (15), we can estimate the best object view that maximizes the entropy $H(\xi, u)$. The main problem for performing the aforementioned maximization is that the number of projections ξ is infinity. To overcome this difficulty, we evaluate $H(\xi, u)$ over discrete (limited) viewpoints on the spherical coordinate system that encloses the 3D object. The spherical coordinate system and the discrete points used to estimate best object views is depicted in Fig. 6 around a 3D object of a church.

Table 6 shows the main steps of the proposed personalized best view selection procedure.

7 Platform implementation

7.1 Implementation details

The platform has been developed using the Google Earth Building maker tools. The buildings are reconstructed as 3D models, each face of which is described using a set of metadata. We assume that all the 3D objects have five faces (the face that points to the ground is not considered), no matter of their 3D complexity. In particular, we represent all 3D objects as a "*bounding-box*" of five faces around the object as indicated in Figure 6, where the 3D model of the national Gallery/Library of Athens is presented. The platform exploits the Google Earth plug-in and the Google Earth virtual environment to visualize the 3D geographical data. We have developed a front end interface to handle user's interactions with the system and a back end module, called GIS Scene Explorer (GIS-SE), that contains the learning, the Macro-path and the Micro-path processes as described in the aforementioned sections. Since Google Earth does not allow retrieval and management of 3D geographical content, a separate geo-referenced database was used for this purpose in this paper. The morphology and the texture of the ground were retrieved by the Google Earth's database to increase the overall scene visualization.



a b

Fig. 6 **a** The spherical coordinate system and the discrete points used to estimate best object views, **b** The 3D model representation of National Gallery/Library of Athens

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

7.2 Evaluation metrics

The most common objective criterion used for evaluating the performance of a personalized route planning system is through the *precision-recall curve*. We denote as *Precision* Pr, the ratio of the number of relevant retrieved objects, say $R(M)$, over the total number of objects, say M, including in a selected route path [39]. That is,

$$\Pr(M) = \frac{R(M)}{M} \tag{16}$$

Table 6 The main steps of the Macro Path Estimation Algorithm

Algorithm 5: The best object view selection algorithm
1. Set azimuth angle *Phi* ← 0 and inclination angle *Theta* ← 0
2. Set *Best_View* ← *Mimimum_Value*
3. For *Theta*=0 to 90, *Theta* + =*Step* do
a. For *Phi*=0 to 360, *Phi* + =*Step* do
i. *Entropy* ← *Eq.* (15)
ii. If (*Entropy* > *Best_View*)
1. *Best_View* ← *Entropy*
2. *Best_Phi* ← *Phi*, *Best_Theta* ← *Theta*

On the contrary, recall Re is defined as the ratio of the number of relevant retrieved objects $R(M)$ over the total number of relevant objects including in the whole scene of the 3D world, say G [39]

$$Re(M) = \frac{R(M)}{G} \tag{17}$$

For a "perfect" system, both Pr(M) and Re(M) should be high [39] (ideally equal one). However, in real situations, as the number M increases, precision decreases, while recall increases. This means that as the route complexity increases, it is more probable that irrelevant objects, with respect to user's preferences to be selected, decreasing the precision ratio. On the other hand, complex paths impose higher probability to select more relevant objects among all possible relevant ones resulting in an increase of the recall values. Because of this, instead of using a single value of Pr(M) or Re(M), the Precision-Recall curve is usually adopted to characterize the performance of a system.

In Eqs. (16) and (17), precision and recall have been estimated for a specific path. However, to evaluate the overall performance of the system, many paths should be created and then, for each path the precision-recall curve needs to be computed. The average values over all routes yield the total precision-recall metrics. Let us assume that Q paths have been examined, then, the average precision/recall, APr, ARe are defined as

$$\mathrm{APr}(M) = \sum_{q=1}^{Q} \mathrm{Pr}(M)/Q \qquad \mathrm{ARe}(M) = \sum_{q=1}^{Q} \mathrm{Re}(M)/Q \tag{18}$$

7.3 Simulations regarding the learning component

Initially a standard set of ten predefined objects is used to get an approximate estimate of user's preferences. In particular, all possible combinations of pair-wise comparisons of the ten models are considered (forty five combinations) to create the initial user profile. Afterwards, the learning module recursively updates user's preferences by estimating the most dominant feature elements, to fit the current user's information needs. More specifically, each time a new pair-wise comparison is submitted to the system, new weight values are estimated using the algorithm of Section 4.3. Feature element interrelationships, described through the ontology, are also exploited in order to refine the initially estimated weights. In this way, we conclude to a reliable weight estimation even in cases that few samples are submitted to the system.

In order to understand the approach adopted in this paper, we present in the following a use case scenario that clarifies system operation. In this scenario a user wishes to have a virtual tour in the city of Athens and therefore he/she uses the proposed adaptive personalized route guidance platform. The ultimate goal of the platform is to select a 3D navigation route that satisfies his/her preferences. In order to achieve this, initially a set of pair-wise comparisons is depicted to the user for evaluation purposes, as preference judgments, that indicate which of the two presented objects is preferred against of the other. Figure 7 presents four characteristics screenshots, which clarifies the interactive training process. In this particular example, the user selects the Acropolis archaeological place against the Acropolis Museum [see Fig. 7(a)], while in Fig. 7(b) he/she selects the Temple of Olympian

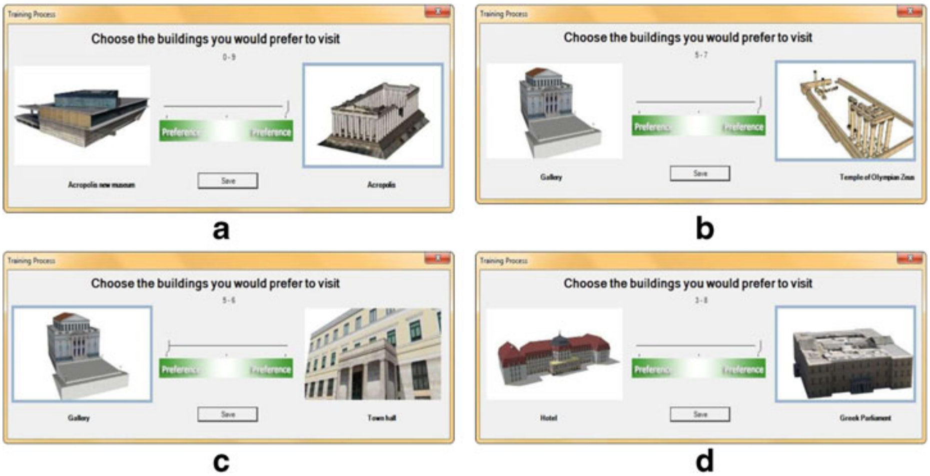


Fig. 7 We present four screenshots as an explanation of our approach used to estimate user's preferences through the interaction process given in the form of preferences judgments (i.e., statements that indicate that one object should be preferred against of another) that are then fed back to the system. **a** The New Acropolis Museum is presented against the Acropolis Archaeological site and the user selects the Acropolis site. **b** The temple of Olympian Zeus is selected against the National Gallery/Library. **c** The National Gallery/Library of Athens is selected against the Town Hall. **d** The Greek Parliament is chosen against a hotel to indicate preferences in architectural style

Zeus against the National Gallery/Library of Athens. Finally, Fig. 7(c),(d) indicate that the user better prefers buildings of Neoclassical architecture[1] such as the Greek parliament and the National library of Athens against buildings of general purposes and public use (e.g., hotels and town hall buildings).

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

In this experiment, we have used 29 metadata to describe the 3D object, each of which is associated with a relative weight. Figure 8 visualizes the convergence rate of the weights for this particular experiment. In this figure, we have depicted only 18 weights out of the 29 in total, just for clarification purposes. As is observed, the weights that are associated with the "Ancient", "Made from Stone" and "Made from Marble", "Historical Education" and "Neoclassical" metadata receive high relative values, since these metadata describe better the archaeological places and the neoclassical architectural style which are preferred by the user. The respective weight values indicates that the platform selects a 3D personalized route that primarily composes, as much as possible, of archaeological places followed by Neoclassical buildings. Instead, in systems where the weights are set manually, it is quite difficult for simple users to exactly quantify their preferences by accurately defining the respective weights for all objects' metadata.

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

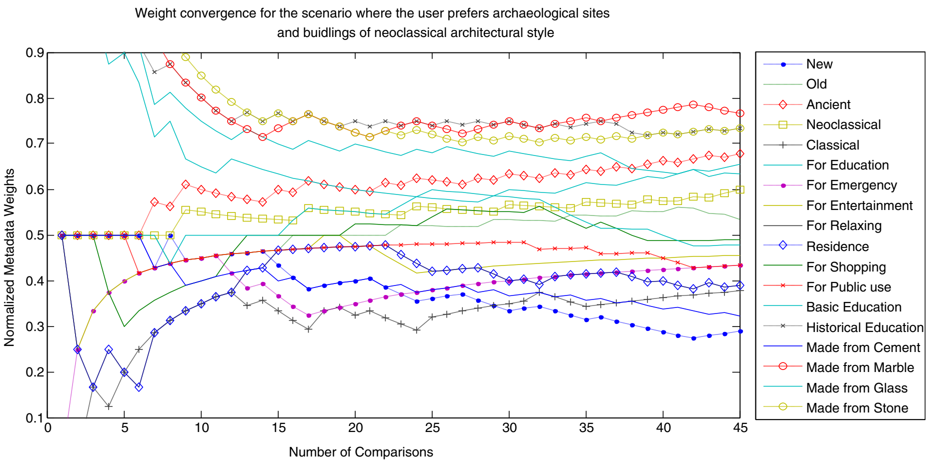


Fig. 8 Weight convergence versus the number of pair-wise comparisons for a scenario where the user selects archaeological sites and buildings of neoclassical architectural style

The advantages of the proposed architecture compared with existing personalized route guidance systems is the incorporation of the relevance feedback learning component, which has the capability of automatically adjusting the objects' metadata weights to capture user's preferences. Instead, in the existing approaches, user's preferences are manually defined by allowing users to adjust the weight values according to their information needs. However, such an approach presents the drawback that manually setting all the weights by the users or performing an overall ranking of all objects within the scene is an arduous task which is practically impossible to be implemented, especially in large scale virtual scenes, where thousands of 3D objects are involved. In the following, we present experimental results that indicate performance of the proposed relevance feedback component in contrast with other learning approaches. Comparisons have been performed quantitative, using the objective criteria described in Section 7.2.

Figure 9(a) presents comparison results of the proposed relevance feedback learning strategy against other learning methods, such as the method of [37] (relevance feedback as inverse of standard deviation), in which the weights are updated inversely proportionally to the standard deviation of the metadata over the relevant selected 3D objects, being retrieved on previous iterations and the Analytic Hierarchical Process (AHP) algorithm of [38] in which the weights are ranked with respect to the outcome of a pair-wise comparison. Additionally, in this figure, we have presented the effect of the ontology-driven weight rectification component on precision-recall curve. As we observe, the proposed algorithm outperforms the compared ones in terms of precision at a given recall value. Figure 9(b) illustrates the effect of precision versus the number of feedback iterations which is directly related with the number of pair-wise comparisons executed in the proposed adaptive personalized route guidance system. All the results have been obtained at recall 50 %. In particular, we assume that in each feedback iteration, a user interacts with the system selecting preferred (relevant) objects between two depicted ones. As is observed, precision increases as feedback iterations increase. However, the improvement ratio is saturated meaning that beyond a certain limit of iterations no significant improvement is encountered. In this figure, we also notice that the ontology-driven weight rectification strategy significantly

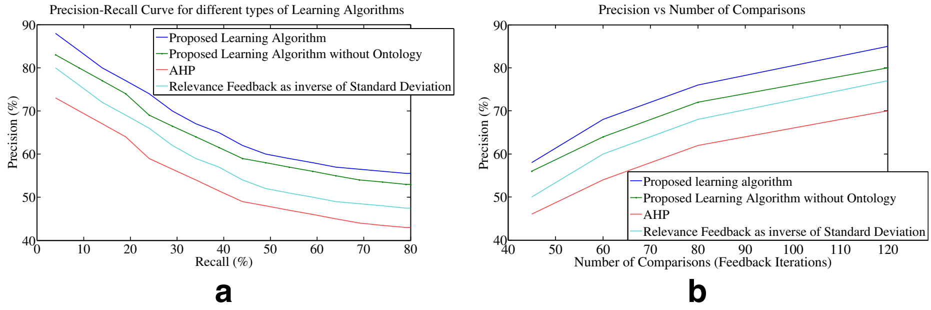


Fig. 9 Comparisons of the proposed relevance feedback learning component with other on-line learning strategies and experimental documentation of the ontology driven rectification mechanism. a The precision-recall curve for various learning algorithms versus the proposed one. b The precision versus the number of feedback iterations to illustrate the effect of number of pair-wise comparisons on the overall system performance. The results have been obtained at recall 50 %

improves precision due to the exploitation of the metadata relations involved. This means, in other words, that the ontology scheme forces the algorithm to localize "representative" samples able to cover the whole variations of the feature space. Figure 10 shows the number of pair-wise comparisons on the system performance as regards the precision-recall curve. Again, we observe that the ontology-driven weight rectification strategy improves precision for all recall values even though a small number of samples is selected.

7.4 Simulations regarding the macro-path layer component

Simulations are performed using three different methodologies; (a) the Dijkstra shortest path method [12], (b) the nearest neighbor algorithm which provides a heuristic approximate of the travelling salesman problem [18] and (c) a linear programming optimization policy [1]. Figure 11 compares the proposed scheme against the three aforementioned route planning algorithms in terms of precision-recall. In all experiments, the proposed genetic optimization (GA scheme) has been implemented at 1,500 iterations cycles. As is observed, the proposed GA scheme outperforms the compared ones. 4

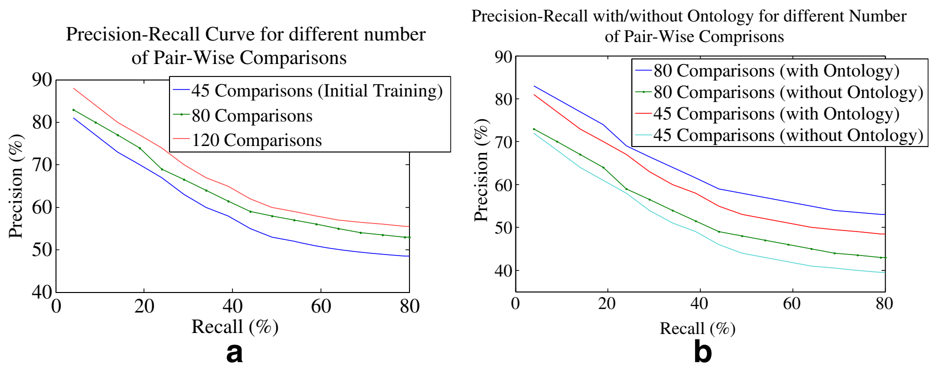


Fig. 10 The effect of the number of selected samples on the precision-recall curve. **a** Precision recall for different number of comparisons (feedback iterations) as regards the proposed learning strategy. **b** The effect of ontology on precision-recall for different number of comparisons

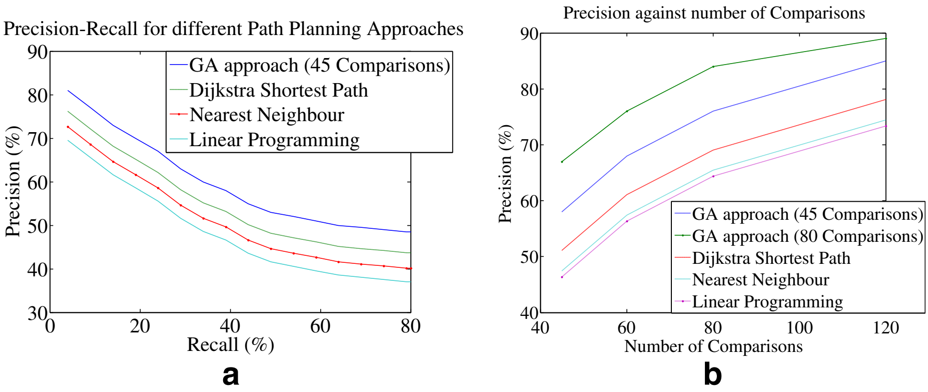


Fig. 11 Comparisons of the genetic based route planning algorithm with other existing optimization strategies. **a** The precision-recall curve. **b** Precision versus feedback iterations

Another advantage of the proposed genetic-based optimization approach is its scalability. As we increase the number of iterations of the genetic algorithm, we are able to also improve the optimization performance, at the cost of computational complexity. However, in application scenarios where the computational cost is critical or in cases that the capabilities of the terminal devices are not so powerful, we are able to terminate the genetic optimization strategy earlier to save complexity at the cost of performance. Figure 12 presents experimental results as regards the effect of iteration cycles on the performance of the genetic algorithm. As is observed, the more the iteration cycles we have, the better the performance of the route selection we achieve. We also observe that few iterations are required in order the genetic optimization approach to retain a similar performance with the Dijkstra shortest path route selection algorithm (see Fig. 12). It is clear that as the number of iterations of the genetic algorithm increases, the precision, as defined through Eq. (18), increases for a

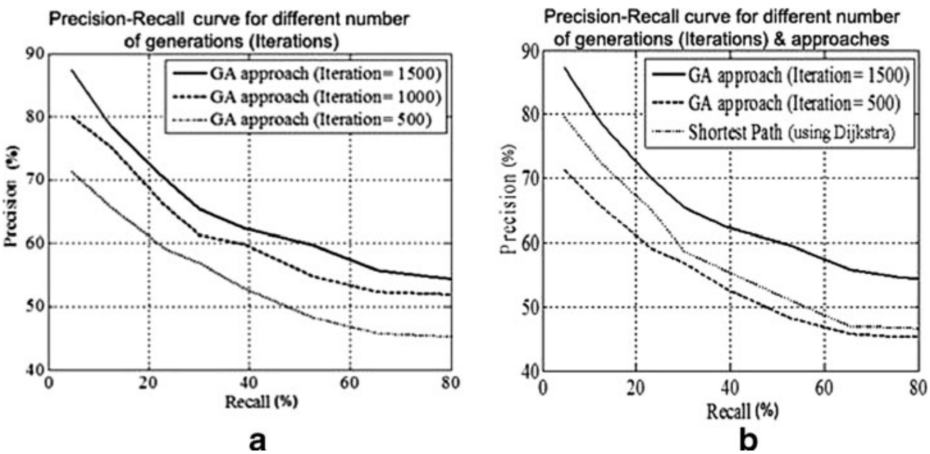


Fig. 12 **a** Precision-recall curve for different iterations of the genetic algorithm. **b** Comparison of the genetic algorithm with the shortest path algorithm

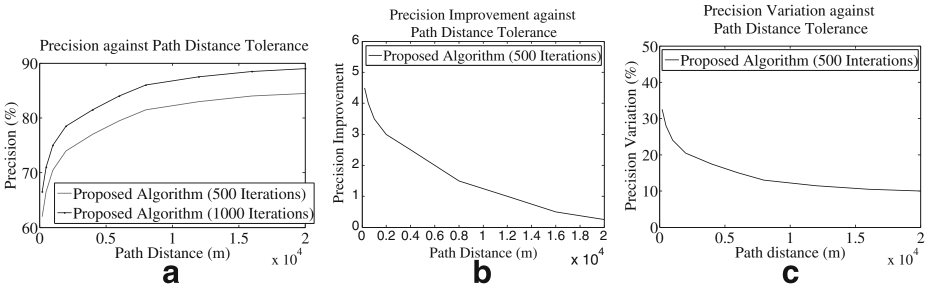


Fig. 13 **a** Precision against path distance tolerance for different iterations of the genetic algorithm. **b** Precision improvement against path distance tolerance in case of a genetic optimization algorithm of 500 iterations. **c** Precision variation against path distance tolerance in case of a genetic optimization algorithm of 500 iterations

given recall value. It is also expected that as recall increases, precision decreases since more objects are included in the selected route path, meaning that we have more complex routes.

A factor that affects the precision of the route selection algorithm is the route distance that a user can tolerate. In particular, if a user does not care about route distance, the genetic-based optimization can find a solution that mostly satisfy his/her needs by including in the itinerary most of the preferred objects. Instead, in case that the user considers routes with the shortest distance from a start to an end point, the optimization algorithms fails to include preferred objects in the itinerary reducing the precision performance of the system. The regulating factors γ_i of Eq. (10) adjust the importance of route (path) distance to the aggregate cost function. Figure 13(a) depicts the precision against the distance that a user can tolerate for a route. In this figure, the performance has been shown for two different iteration cycles of the genetic optimization algorithm. The results have been obtained as average values over 15 different experiments where user selects different start and end points and he/she sets a fixed route distance that can tolerate with. As is observed, as path distance tolerance increases, the precision also increases since more preferred objects are included in the itinerary. However, the performance rate decreases indicating saturation of the precision. This is verified in Fig. 13(b) which presents the precision improvement versus the distance that the user can tolerate. Precision improvement is derived as the derivative of the precision curve of Fig. 13(a) indicating at which amount we can increase the precision by increasing path distance. Another aspect that determines the performance of the proposed system is the variation of the precision achieved over the 15 different experiments. As is expected different precision values are encountered across the 15 different executed experiments. Figure 13(c) shows the precision variation versus the path distance that a user can tolerate with. As is observed, higher variations are encountered for short path distances. This is expected since for short distances, precisions highly depend on the selection of the start and end points. The opposite is valid for high path distance tolerance, where the effect of the selection of the start and end point is minimized.

Figure 14 presents a subjective (qualitative) evaluation of the proposed genetic optimization strategy. Simulation results have been performed using the scenario described in Section 7.3. We recall that in this particular case, the user prefers archaeological places and buildings of neoclassical architectural style in his/her 3D navigation. In this figure, we also present the route provided by the shortest path Dijkstra's algorithm for comparisons purposes. In particular, the shortest path algorithm yields the shortest navigation route that connects SP and FP points. On the contrary, the proposed GA approach includes in the

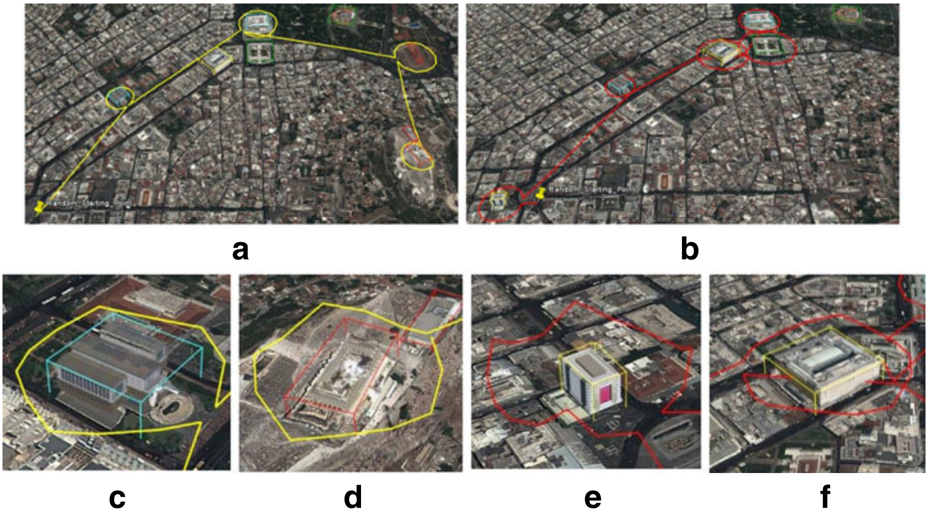


Fig. 14 **a** Generated path using the proposed GA approach, **b** Generated path using Dijkstra shortest path algorithm. **c**, **d** Zoom in two preferred 3D objects of the itinerary retrieved by the proposed genetic algorithm approach. **e**, **f** Zoom in two preferred 3D objects of the itinerary retrieved by the shortest path distance algorithm

itinerary objects that are most preferred by the user at a cost of increasing the overall route distance. However, the user is tolerant with such a distance increase. Figure 14(c, d) zooms in two of the preferred objects of the itinerary extracted by the genetic algorithm, while Fig. 14(e, f) shows two of the retrieved objects in case of the shortest path algorithm. As is observed, in the first case, the National Gallery/Library of Athens and the Acropolis archaeological site have been selected in the itinerary, which are consistent with user's preferences. Instead, the buildings, retrieved using the shortest path algorithm, do not satisfy the actual user's preferences.

7.5 Simulations regarding the micro path layer

In our implementation, each projected view of the 3D model is calculated on discrete points around the spherical coordinate system (see Fig. 6(a)) for more details. Then, for the given set of selected discrete view points, the ones that maximize the overall personalized entropy metric for all the five faces are selected as the most appropriate for the user. In order to understand the proposed algorithm, we discuss in the following, a use case scenario, where the user selects the Greek Parliament, as an interest object to navigate around. Figure 15(a) presents the best projected view obtained by maximizing the number of projected pixels of the object faces over the total number of the object pixels, i.e., maximization of the entropy metric of Eq. (14). Instead, in our approach, the best view is obtained by incorporating additional criteria that capture user's preferences, resulting in a maximization of the personalized entropy metric [see Eq. (15)]. The best three top ranked views are depicted in Fig. 15(b, c, d). The Greek parliament is represented by a bounding box. Each of the four surrounded faces have almost the same metadata ("*Made from Marble*", "*Historical Education*", "*Neoclassical*"), but the upper face of the bounding box has "*Made from Cement*" instead of "*Made from Marble*". Since the user is more interested with marbles instead of cement, the proposed personalized best view selection algorithm estimates views that are

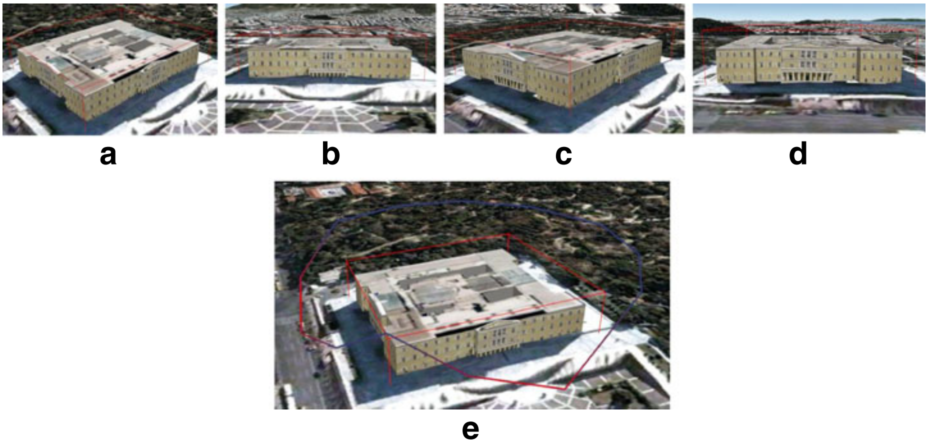


Fig. 15 **a** Best view selected using only geometrical properties. **b**, **c**, **d** Best view selected using geometrical and human factors. **e** Indicating camera and speed trajectory of the selected viewpoints

more suitable to user's preferences instead of the conventional methods where only geometric constraints are taken into consideration.

All the aforementioned figures demonstrate the outperformance of the proposed scheme in terms of precision-recall against the compared ones. However, computational complexity is still an issue. The proposed genetic-based scheme requires high computational cost than the other methodologies, especially in cases that high iteration cycles are required. Nevertheless, this drawback can be also seen as an alternative advantage due to the scalable nature of the genetic scheme. In other words, we can stop further executing the genetic algorithm when the computational cost exceeds a maximum acceptable limit as defined by our application case scenario. Instead, the aforementioned three optimization strategies lack scalability; they always require the same computational time. Table 7 shows the computational complexity of the proposed genetic algorithm at different iterations compared with the aforementioned conventional route planning algorithms, i.e., the Dijkstra shortest path method, the nearest neighbor and the linear programming policy. The complexity is depicted in Table 7 as a normalized ratio with respect to Dijkstra which provides the minimum cost. In this table, we also depict the precision achieved at the given cost assuming a recall of 49 %. As is observed, the proposed genetic algorithm provides almost similar performance for the same computational cost.

In Fig. 15(e), we have illustrated the 3D trajectory around the Greek parliament, which is useful to support a personalized 3D navigation experience. The trajectory is formed by

Table 7 Performance and computational complexity of the proposed genetic based algorithm along with other optimization strategies

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

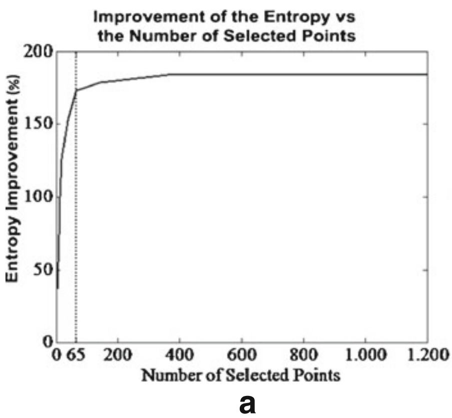
merging several top ranked views as obtained either by the proposed best view selection algorithm or the conventional geometric approach. Blue regions indicate views that maximize the classical entropy criterion, that is, maximization of the projected area [see Eq. (14)]. Instead, red regions correspond to views that maximize the personalized entropy, that is, geometrical and human factors [see Eq. (15)]. Similar results are also depicted in Fig. 14(c, f) in which the personalized best views are also illustrated. We recall that Fig. 14(c, d) depict two preferred objects as they have been retrieved using the genetic scheme while Fig. 14(e, f) two objects being retrieved using the shortest path algorithm.

In the following, we evaluate the complexity of the best view selection algorithm. It is clear that the complexity is straightforwardly proportional to the number of examined discrete points on the semi-spherical bounding boxes. The more the examined points on the semi-sphere coordinate system are, the more the complexity is. It is also clear that as the complexity increases, the performance is also improved. This is verified in Fig. 16(a), which presents the entropy improvement versus the number of selected points used. However, the ratio of the improvement is decreased, meaning that beyond a certain number of points a slight improvement is achieved though complexity continues to increase.

This is depicted in Fig. 16(b), where we plot the improvement ratio versus the computational complexity. These results have been obtained as the average over 50 experiments of different 3D objects. Based on these figures, we can approximate an average number of computations required to achieve a projected view of a 3D building close to the optimal solution (94.6 % approximation). For example, in case that we use 65 different points on the semi-spherical coordinate system, we can have a very good approximation of the personalized entropy metric as is verified in Fig. 16(a). It is also clear that we can relax approximation accuracy for building of low user's interest.

8 Conclusions and discussions

3D geo-informatics has entered the digital age enabling new potential applications in the field of virtual tourism, pleasure, entertainment and cultural heritage. It is argued that 3D



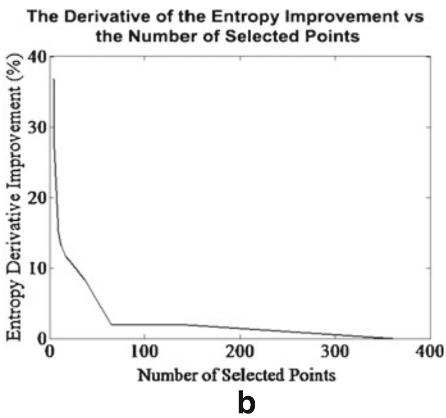


Fig. 16 **a** Improvement of the entropy versus the number of selected points. **b** The derivative of the entropy improvement versus the number of selected points

virtual tools provide the natural way of navigation since they can explain a story or even show up the space of a world. Personalization is a key aspect for an efficient 3D navigation since routes are selected not only using geometrical criteria, but mostly adhere to user's preferences. Personalization is often modeled through a set of weights that regulate the degree of importance of the scene metadata on the route selection process. In existing approaches, metadata weights are left to the user's side who is responsible for defining his/her profile. This, however, implies that the user is aware of technical details as regards system implementation.

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

All these components have been integrated in a single platform and comparisons have been conducted to demonstrate the outperformance of the proposed route guidance system. In particular, as far as the on-line learning strategy is concerned, we have compared it with a relevance feedback scheme that estimates metadata weights inversely proportional to the standard deviation of the feature elements over all relevant selected objects [37] and with the Analytic Hierarchical Process (AHP) algorithm [38]. We observe that the proposed learning scheme yields, on average, a higher precision value of 8.1 % (or equivalently an improvement of 13.93 % of the proposed scheme in relation with the method of [37]) over all recall values than the second best compared learning strategy which is the method of [37]. In addition, we have performed simulations to illustrate the effect of the weight rectification strategy through the use of the ontology. We conclude that, in this case, we can reduce the number of samples (pair-wise comparisons) required of 76 % to achieve the same precision.

Regarding route optimization, we have compared our genetic algorithm (GA) with three methodologies, the Dijkstra shortest path method, the nearest neighbor algorithm and a linear programming optimization policy. In all cases, the proposed GA method yields, on average, higher precision value of 4.76 % (or equivalently an improvement of 8.75 % of the proposed scheme in relation with Dijkstra) over all recall values than the second best one (i.e., the Dijkstra shortest path), however, at a cost of an increase of the overall path distance. In particular, in case that a user does not care about route distance, the genetic optimization can find a solution that mostly satisfy his/her needs by including in the itinerary most of the preferred objects. Instead, in case that the user considers routes with the shortest distance from a start to an end point, the optimization algorithms fails to include preferred objects in the itinerary and the performance of

GA is similar to the other compared methods. The limitation of the GA is that it requires high computational cost compared to other methodologies, especially in cases where high iteration cycles are included. However, this limitation can be also seen as a potential advantage of the GA since it allows a scalable implementation; we can improve the optimization performance at the cost of demanding more computational effort. At the end, our simulations have been demonstrated that the proposed GA optimization algorithm provides better performance than the compared techniques for the same computational cost. Finally, experimental results as regards the best view selection algorithm indicate that we can increase the degree of personalization for 3D navigation. In particular, we are able to estimate personalized 3D trajectories around the preferred objects by merging projected 2D views which have been estimated not only using geometric constraints but also the importance of the projected faces as defined through the metadata weights.

Of particular interest for future work is to improve the route selection algorithm by the use of modern optimization techniques, such as particle swarm optimization, ant colony or the artificial bee colony algorithm, which are promising in solving combinatorial problems [10, 20]. Moving forward the modified IWO algorithm for personalized multi-criteria quasi-optimum path problem (PUMMQPP) presented in [31] can be also used as an alternative optimization process for route planning of the *'Macro Path'* instead of our GA approach. Another aspect for future research is to incorporate 3D geometric constraints in the construction of 3D trajectories which increases the complexity of the optimization.

References

1. Applegate DL, Bixby RM, Chvátal V, Cook WJ (2006) The traveling salesman problem: a computational study, princpton series in applied mathematics
2. Allaire F, Tabouchi M, Labonte G, Fusina G (2008) FPGA implementation of genetic algorithm for UAV real-time path planning. J Intell Robot Sys 54(1–3):495–510
3. Bardis G (2009) "Intelligent personalization in a scene modeling environment," In: Miaoulis G., Plemenos D. (Eds), Studies in computational intelligence, Vol.181, pp. 89–119, ISBN 978-3-540-92901-7, Springer
4. Bardis G, Miaoulis G, Plemenos D () "User profiling from imbalanced data in a declarative scene modeling environment. In: Plemenos D, Miaoulis G (Eds.), Artificial intelligence techniques for computer graphics," Series: Studies in computational intelligence, Vol. 159, pp. 123–140, ISBN 978-3-540-85127-1, Springer, 2008
5. Bonnefoi PF, Plemenos D (2002) "Constraint satisfaction techniques for declarative scene modeling by hierarchical decomposition", 31A′2000 international conference
6. Botea A, Müller M, Schaeffer J (2004) Near optimal hierarchical path-finding. J Game Dev 1:7–28
7. Chalvantzis C, Virvou M (2008) Fuzzy logic decisions and web services for a personalized geographical information system. Stud Comput Intell 142:439–450
8. Chatzis S, Doulamis A, Varvarigou T (2007) "A content-based image retrieval scheme allowing for robust automatic personalization," ACM international conference on image and video retrieval, pp. 1–6
9. Cohen W, Schapire R, Singer Y (1999) Learning to order things. J Artif Intell Res 10:243–270
10. Colorni A, Dorigo et M, Maniezzo V (1991) Distributed optimization by Ant Colonies, actes de la première conférence européenne sur la vie artificielle. Elsevier Publishing, Paris France, pp 134–142
11. Demyen D, Buro M (2006) "Efficient triangulation-based pathfinding", 18th international conference on innovative applications of artificial intelligence conference (AAAI'06), pp. 942–947, ISBN 978-1-57735-281-5
12. Dijkstra EW (1959) "A note on two problems in connexion with graph". Numer Math 1(1):269–271
13. Doulamis N, Chronis E, Miaoulis G, Plemenos D (2010) Personalized view selection of 3D molecular proteins. Stud Comput Intell 321:211–227, Springer Press
14. Doulamis A () "Event-driven video adaptation: A powerful tool for industrial video supervision" Multimedia Tools and Applications doi:10.1007/s11042-012-0992-5 (to appear)

15. Freund Y, Schapire R (1997) A decision-theoretic generalization of on-line learning and an application to boosting. J Comput Syst Sci 55(1):119–139
16. Gauch S, Cha J, Pretschner A (2003) Ontology-based personalized search and browsing. J Web Intell Agent Sys 1:219–234
17. Guarino N (1998) "Formal ontology in information systems", In proceedings of the 1st international conference on formal ontologies in information systems FOIS. Ios Press, Trento, pp 3–15
18. Gutin G, Yeo A, Zverovich A (2002) Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the TSP. Discret Appl Math 117:81–86
19. Hart PE, Nilsson NJ, Raphael B (1968) A formal basis for the heuristic determination of minimum cost paths. IEEE T Syst Man Cyb 4:100–107
20. Kennedy J, Eberhart R (1995) Particle swarm optimization. Proc IEEE Int Conf Neural Netw 5:1942–1948
21. Kobsa A, Koenemann J, Pohl W (2001) Personalized hypermedia presentation techniques for improving customer relationships. Knowl Eng Rev 16(2):111–155
22. Kraljic N (2008) "Interactive video virtual tours", 12th central european seminar on computer graphics. Split, Croatia
23. Laakso K, Gjesdal O, Jan Rasmus Sulebak (2003) "Tourist information and navigation support by using 3D maps displayed on mobile devices", Workshop "HCI in mobile Guides", Udine (Italy)
24. Littlestone N, Warmuth M (1994) "The weighted majority algorithm. Information and Computation," pp. 212–26
25. Lucas M, Desmontils E (2007) "Declarative modelers". Revue Intern. de CGAo et d' Infogr., pp. 559–585
26. Mackinlay J, Card S, Robertson G (1990) Rapid controlled movement through a virtual 3D workspace. Comput Graphics 24(4):171–176
27. Mekni M, Moulin B (2010) "Hierarchical path planning for multi-agent systems situated in informed virtual geographic environments," Second international conference on information, process, and knowledge management, Saint Maarten, pp. 48–55, ISBN 978-1-4244-5688-8
28. Mitchell M (1996) An introduction to genetic algorithms (Complex Adaptive Systems*)*. MIT Press, Cambridge
29. Nadi S, Delavar MR (2011) "Multi-citeria, personalized route planning using quantifier-guided ordered weighted averaging operators". International Journal of Applied Earth Observation and Geoinformation, pp. 322-3358
30. Niaraki SA, Kim K (2009) Ontology based personalized route planning system using a multi-criteria decision making approach. Expert Sys App Sci Direct 36(2):2250–2259
31. Pahlavania P, Delavar MR, Frank AU (2012) Using a modified invasive weed optimization algorithm for a personalized urban multi-criteria path optimization problem. Int J Applied Earth Observ Geoinformation 18:313–328
32. Reitter D, Lebiere C (2010) A cognitive model of spatial path-planning. Comput Math Org Theory 16(3):220–245
33. Reuter A, Zipf A (2005) "GIScience—where next". In: Fotheringham S, Wilson J (Eds.), Handbook to GIS. Blackwell
34. Riecken D (2000) Personalized views of personalization. Commun ACM 43(8):27–28
35. Rocchio J (1971) Relevance feedback in information retrieval: The SMART retrieval system: Experiments in automatic document processing. Prentice Hall
36. Rogers S, Langley P (1998) "Personalized driving route recommendations. In: Proceedings of the AAAI Workshop on Recommender Systems," July 26, Madison, WI, USA, pp. 96–100
37. Rui Y, Huang TS, Ortega M, Mehrotra S (2008) Relevance feedback: a power tool for interactive content-based image retrieval. IEEE Trans Circ Sys Video Technol 8(5):644–655
38. Saaty TL (1990) "Decision making for leaders: the analytic hierarchy process for decisions in a complex world". RWS Publications ed
39. Salton G, McGill MJ (1982) "Introduction to modern information retrieval," New York: McGraw-Hill Book Company, ISBN 0070544840
40. Shannon CE (1948) A mathematical theory of communication. Bell Syst Tech J 27:379–423, 623–656
41. Sokolov D, Plemenos D (2005) "Viewpoint quality and scene understanding," VAST, Eurographics Symposium Proceedings, Pisa (Italy), pp. 67–73
42. Sokolov D, Plemenos D (2007) Virtual world explorations by using topological and semantic knowledge. Vis Comput J 23:173–185
43. Tab D, Robertson G, Gzewinski M (2001) Exploring 3D navigation: combining speed-coupled flying with orbiting. Proc SIGCHI Conf Human Factors Comput Syst 3:418–425

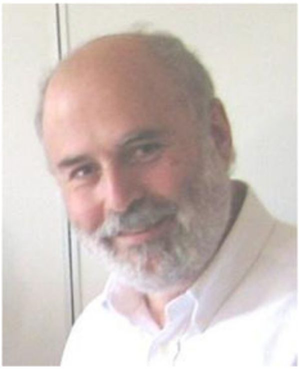
44. Vazquez PP (2003) On the selection of good views and its application to computer graphics. Ph.D thesis, Technical University of Catalonia, Barcelona
45. Vazquez PP, Feixas M, Sbert M, Llobet A (2006) Realtime automatic selection of good molecular views. Comput Graph 30:98–110
46. Zipf A, Jost M (2006) Implementing adaptive mobile GI services based on ontologies: examples from pedestrian navigation support. Comput Environ Urban Syst 30(6):784–798

Christos Yiakoumettis is a PhD Students in University of Limoges. He holds an MRes degree in Computer Science & Artificial Intelligence from the University of Sussex, Brighton, United Kingdom (2005) and a Diploma in Informatics from the Technological Educational Institute (T.E.I) of Athens, Greece (2004).

He had worked as a web developer and programmer in a few companies. Moreover he has participated in several research and developing projects.

He currently works as a part-time instructor at T.E.I. of Athens. Results of his research activities can be found published in conferences and in Springer's volumes.

Nikolaos D. Doulamis (S'96, M'00) received a Diploma in Electrical and Computer Engineering from the National Technical University of Athens (NTUA) in 1995 with the highest honor and the PhD degree from the same department in 2000. He is now Assistant Professor at the National Technical University of Athens. Dr. Doulamis received the Best Greek Student award in the field of engineering at national level by the Technical chamber of Greece in 1995. He is a recipient of the Best Graduate Thesis Award in the area of electrical engineering, and of NTUA's Best Young Engineer Medal. He has served as Chairman or member of the program committee of several international conferences. He is author of more than 40 journals papers in the field of video transmission, computer vision, multimedia, content-based image retrieval, and grid computing, and of more than 120 conference papers. 14 of the journals papers have been published in the IEEE press. He has more than 1500 citations. In 2007, he has been selected among the best 500 young European scientists.



Georgios Miaoulis is a Professor at the Informatics Department, Technological educational institute of Athens, Greece and head of a joint Greek-French Master Program concerning multimedia, internet technology, and digital image.

He received an Engineering degree from the *Polytechnic School of Aristotle University* in Thessaloniki, Greece in 1977, a MS in organization & informatics from *CNAM*, and a MS from *EHESS* Paris-France, a PhD in Medical Informatics from *University of Crete*, Greece, a PhD and a Research Supervision Diploma in informatics from *University of Limoges*, France.

His main research interests include intelligent information systems and image-based systems in various application fields

Djamchid Ghazanfarpour is a Professor at University of Limoges, France, since 1993 and head of *Realistic Image Synthesis* research team of XLIM Laboratory (UMR CNRS 6172).

Professor GHAZANFARPOUR received his BS from *Tehran Polytechnic*, Iran, in 1980, his MS from *INPG* in Grenoble, France, in 1982, his Doctor-Engineering degree from *ENSMSE* in Saint-Etienne, France, in 1985, his PhD from *ENSMSE* and *Saint-Etienne University*, France, in 1990, and his Research Supervision Diploma from *Strasbourg University*, France, in 1992.

His main research interests include texture synthesis, surface details modeling, natural phenomenon synthesis and animation, antialiasing and real time rendering.