Article

Object Identification Based on the Automated Extraction of Spatial Semantics from Web3D Scenes

Konstantinos Kontakis[1], Athanasios G. Malamos[1], Malvina Steiakaki[1], Spyros Panagiotakis[1] and J. Andrew Ware[2,*]

[1]Department of Informatics Engineering, Technological Educational Institution of Crete, Greece
kostas.kod@gmail.com, amalamos@ie.teicrete.gr, malvinast@ie.teicrete.gr, spanag@ie.teicrete.gr
[2]Faculty of Computing, Engineering and Science, University of South Wales, UK
andrew.ware@southwales.ac.uk
*Correspondence: andrew.ware@southwales.ac.uk

Abstract: We present a web-based methodology for the extraction of semantic information and object identification in poorly annotated Web3D scenes. Our approach is based on a set of rules that mimic human spatial cognition, backed by an indexing mechanism for the automated spatial correlation between objects in terms of linguistic predicates. The proposed methodology is comprised of real-time techniques with broad application areas, ranging from CAD and 3D room planning tools based on architectural designs, to medical and e-learning environments where their objects' relative position is closely associated with identification attributes.

Keywords: Spatial semantics; object identification; 3D model indexing; Web3D

1. Introduction

The unremitting evolution of Web3D technologies over the last decade has led many developers of high-interest application areas to replace traditional 2D presentation formats with 3D graphics content. This continuous progress is reflected on a daily basis by the increasing engagement rates of virtual worlds with science, business and education domains [27]. Typical examples are found in numerous Internet applications that promote such things as automobiles, electrical appliances and industrial furniture, while others provide complete interior design solutions for imprinting a real-world room space or creating a new virtual space from scratch. Moreover, Web3D movement will spread even further thanks to the latest changes in VR technology, which aim at turning an initially expensive technology into a series of affordable commercial products for the masses. Even though these applications come from different domains, they tend to share common practices in the design of their 3D models, where area experts sketch such models using appropriate software tools. These tools ultimately output the necessary resources which will be integrated into the corresponding application or webpage. At the same time, the prevalence of search engines combined with the structure of current Web, instruct users to employ (mainly) textural descriptions as search criteria.

However, the latter constitutes a limiting factor in the problem which confronts a user with regard to information retrieval, making the visual information included in image and video elements of a webpage to be practically unreachable. This searching restriction remains an open problem which must be addressed in a different way, since nowadays a world without web search engines would be irrational.

The adoption of Web3D technologies goes hand in hand with the native support of real-time rendering offered by most modern browsers, allowing an on-the-fly conversion of countless images and video that ship with X3DOM, OBJ and other textual formats of 3D graphics [24]. So, the question that quickly arises is whether it is feasible to facilitate the searching and retrieval of the information enclosed inside such files. At first sight, it is obvious that several parts of these files comply with a technical standard, specification sheet or API, allowing web users to search for specific features like shapes, colours, URLs, animations, etc. [25] [26] Yet, the majority of users are interested in the non-technical aspects of them, where there is available a vast semantic content about the 3D scene (e.g. what this scene represents, where is an object located in relation to its surroundings). However, this requirement comes in contradiction with today's relatively poor annotated scenes, due to the fact that their objects do not possess the appropriate metadata to support such annotations. Additionally, even namespaces and objects definitions do not have an agreed terminology or adhere to set of consistency rules. Thus, a research branch in 3D graphics deals with the problem of object identification, which is mainly based on techniques that take advantage of objects' shape and volume data [22] [23]. Meanwhile, designers and producers provide some primitive annotation in their scenes, which is usually limited to a textual description or object definition of a 3D model. On the other hand, most advanced solutions may apply several identification rules that can be perceived as semantic attributes based on human ranking data, but such methodologies have proven to be time-consuming and error-prone [28].

In this paper, we present an object identification algorithm which relies on the spatial characteristics of a scene and the relative position of its objects in the world. A 3D scene does not simply represent the underlying environment, but in most cases, contains an abstract information layer with additional semantics about the environment being described and its objects. However, even though this kind of semantic conceptualization has been successfully addressed with the usage of various technologies and information management systems [8], an efficient methodology for the automated extraction of spatial relationships is still absent. Typical use cases of such relationships can attribute more realistic prospects to the semantic representation of scenes; since they are based on human perception and cognition (e.g. A is in front of B, below C, and on the left side of D). In this way, a formal representation of natural language's concepts not only enhances the spatial awareness of the user, but also improves their ability to understand the spatial relations between objects. A first attempt to extensively annotate such scenes and move from a CAD-oriented to a human-oriented spatial environment took place with an artificial intelligence framework [12] which did not support a spontaneous mechanism for spatial relations. Taking it a step further, this work deals with the challenge of providing a web-based and platform-independent algorithm for the automated extraction of spatial semantics from 3D scenes, which could be used to answer regional questions such as "What model is in front of, behind, on the left of, on the right of, under, over, another model" or "Where am I in the virtual world in relation to a model".

The rest of this paper is structured as follows. In Section 2 we address relevant research works on spatial semantics and those that contributed to the implementation of the extraction mechanism. Section 3 presents the available spatial relations and describes step-by-step their application in the underlying computational model, while Section 4 makes use of this model to retrieve spatial information from various use cases. Finally, Section 5 presents our conclusions.

2. Motivation

Today, numerous applications come with an integrated set of topological operators for the deduction of spatial relationships between their datasets. The functionality of such operators complies with the principles set by various topological models, such as the DE-9IM, RCC or OGC

spatial standards. Among them, DE-9IM remains the most widely used standard for building information models and spatially indexing DBMS [9] by defining three different but interconnected geometric regions (interior, exterior and boundary) for each pair of objects which are validated against a set of spatial predicates. These predicates are based on human spatial cognition [10] and have been proved quite practical and flexible in 1D and 2D applications. In a first attempt to process topological relations in R-tree variants [11], all possible relations between two-dimensional MBRs were extracted and tested by complex queries, but today, octrees implementations [13] [14] are supernumerary compared to R-trees in aspects of spatial relations and boundary representation methodologies. Premised on such findings, our previous work [20] aimed to contribute in this area by presenting a uniform three-dimensional topological classification mechanism for R-trees in 3D space.

Walking on the same path, but adopting a different set of linguistic predicates, the implemented mechanism takes also into consideration directional relations. Early works [15] in this area pointed out the fuzzy state of human perception and artificial intelligence systems in selecting the appropriate directional relations between two objects. This inability was not only proceeded from the chosen orientation plane and the objective technique used, but it was also amplified by the fact that neither a qualitative nor a quantitative classification of a space was sufficient to capture similarities between spatial representation and spatial language [16]. However, a set of directional predicates could adequately reflect linguistic semantics in terms of spatial applicability and accuracy. These commonly referenced predicates were mapped onto the *left*, *right*, *above*, *below*, *over*, *front* and *behind* directional relationships, which describe the position of each 3D object according to the global coordinate system of the corresponding application. The latest technological advances allowed the automated generation of such relations for real-world indoor scenes with the utilization of a robotic mechanism [17], while software-based approaches [18] integrated the four cardinal directions in the form of R-tree nodes, to hasten k-NN queries performance at the expense of space utilization. In our work, however, a standalone computational model steps on a R-tree instance for the automated implication of directional relations between the spatially indexed content. Such instances are based on a valid R-tree variant [20] that fully complies with the set of properties addressed in the original version of this data structure.

The implemented R-tree consists of a multi-level hierarchy of axis-aligned bounding boxes, which come in the form of rectangular parallelepipeds due to their application in 3D space. Each box encases a finite number of entries which can either point to tree nodes or spatially indexed records. These records correspond to the actual X3D objects of an X3DOM scene; while the lowest and highest coordinate values among these records define the boundaries of the minimum bounding boxes (MBB) that constitute any R-tree instance. Nodes' splitting methodology makes use of the Quadratic algorithm [7], since it provides a satisfying ratio of time complexity and space utilization. This methodology is accompanied by the appropriate Insertion operation and together forms the boundaries of each MBB. When the desired 3D objects are successfully spatially indexed, then the coordinates of these boundaries are finalized and the following spatial reasoning process is ready to start.
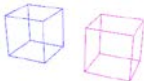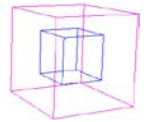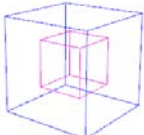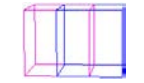
3. Spatial Reasoning

The implemented 3D R-tree data structure has been enriched with a novel computational model for the automated implication of spatial relations. This model makes use of successive topological and directional predicates between each pair of indexed objects, eliminating pairwise entries for improved performance. Such objects are associated with the Cartesian coordinate system, a three-dimensional distance metric which specifies their position with a signed triplet of numerical coordinates. The minimum and maximum values of the latter denote each object's boundaries that ultimately take the form of a rectangular parallelepiped. The vertices deriving from this type of geometry supply the computational model with the necessary semantics for the attribution of the appropriate spatial relations. In this way, the spatial reasoning process requires only each object's

position to constitute a network of relationships based on the topological and directional relations being described below.

3.1 Topological Relations

Topological relationships take advantage of the eight topological predicates that are defined in DE-9IM for 3D space environments [2]. Each one corresponds to a different boundary-based intersection pattern, which classifies an object to a specific topological relation according to another reference object. These relations come with a carefully designed taxonomy to prevent incorrect or nonessential implication of spatial predicates, since only one of the former can hold at a given space and time. Thus, the provided topological relations are serially tested one by one against every MBB pair of a 3D R-tree instance, until a topological criterion is successfully met. This first valid occurrence is the one pointing to the best fitting topological predicate for the current MBB pair.

Table 1. Topological criteria for 3D R-tree structure

	Disjoint	(RectA.xMax <= RectB.xMin || RectA.xMin >= RectB.xMax) || (RectA.yMax <= RectB.yMin || RectA.yMin >= RectB.yMax) || (RectA.zMax <= RectB.zMin || RectA.zMin >= RectB.zMax)
	Touch	
	Equal	(RectA.xMin == RectB.xMin && RectA.xMax == RectB.xMax) && (RectA.yMin == RectB.yMin && RectA.yMax == RectB.yMax) && (RectA.zMin == RectB.zMin && RectA.zMax == RectB.zMax)
	Within	(RectA.xMin >= RectB.xMin && RectA.xMax <= RectB.xMax) && (RectA.yMin >= RectB.yMin && RectA.yMax <= RectB.yMax) && (RectA.zMin >= RectB.zMin && RectA.zMax <= RectB.zMax)
		Complementary relation: CoveredBy
		Opposite MBB relations: Contains, Covers
	Contains	(RectA.xMin <= RectB.xMin && RectA.xMax >= RectB.xMax) && (RectA.yMin <= RectB.yMin && RectA.yMax >= RectB.yMax) && (RectA.zMin <= RectB.zMin && RectA.zMax >= RectB.zMax)
		Complementary relation: Covers
		Opposite MBB relations: Within, CoveredBy
	Overlap	(SearchRect.xMin < Rect.xMax && SearchRect.xMax > Rect.xMin) && (SearchRect.yMin < Rect.yMax && SearchRect.yMax > Rect.yMin) && (SearchRect.zMin < Rect.zMax && SearchRect.zMax > Rect.zMin)

First, the possibility of *Disjointness* between two MBBs is tested for, implying that neither their boundaries nor their interior regions are in contact. The same formula has been also included the *Touch* predicate, since the intersection of MBBs' boundaries alone does not affect the result-set of the upcoming directional relations. If these two topological relations are not satisfied, then the next closest spatial relation is initiated which is none other than the *Equal*. In this case, the engaged rectangular parallelepipeds must have in common not only their relative boundaries, but also their entire interior region. However, this kind of relation is rarely met in real-world applications due to its strict constraints, leading the majority of 3D objects to the next formula of our computational model. That is the *Within* spatial predicate which signifies that the first MBB totally encloses the boundary and interior regions of the second, taking also into account the automated attribution of the diametrically opposed predicates *Contains* to the second one. On the other hand, there is always a chance that this pair of MBBs satisfies the inverse operators of these criteria, implying this time a

Contains relation for the first parallelepiped. This kind of relation is also complementary to the encasement of intersecting boundaries, unifying in this way also the *Covers* spatial predicate for the same parallelepiped. Finally, there is one last spatial predicate left for those MBB pairs that do not satisfy any of the aforementioned topological relations. This predicate refers to the *Overlap* relation, denoting that the interior regions of these MBBs are partially intersecting in 3D space. The aforementioned series of relationships along with their relative allocation formulas for rectangular parallelepipeds *RectA (blue)* and *RectB (red)* are displayed in Table 1.

At this point, all indexed objects of a 3D R-tree instance will have been attributed with a set of topological relationships, terminating the reasoning process for evenly arranged or overlapping records. However, an additional level of spatial reasoning is applied for disjoint MBBs, since their spatial representation can be further enhanced with the usage of the directional relations described below.

3.2 Directional Relations

Directional relations refer to another major category of spatial analysis, where linguistic predicates incorporate directional constraints on the tested subjects. In contrast to topological relations which are independent of user's current viewpoint, directional relations are applied to each spatially indexed object according to the default position and orientation of the right-hand coordinate system used by the X3D standard [1]. Each directional relation is bestowed with a dedicated intersection-based formula for the approximation of the best fitting relation in each case. Such formulas act as a uniform classification scheme for the directional annotation of those MBB pairs, which have been derived to be Disjoint during the stage of topological reasoning. These exclusively accessed spatial extensions deduct the appropriate directional relation for the first parallelepiped of a MBB pair, while at the same time, the second parallelepiped is attributed with the inverse relations of the former skipping unnecessary rounds of spatial reasoning.

Table 2. Directional criteria for 3D R-tree structure

	Left	RectA.xMax <= RectB.xMin
		Opposite MBB relation: Right
	Right	RectA.xMin >= RectB.xMax
		Opposite MBB relation: Left
	Above	RectA.yMin > RectB.yMax
		Opposite MBB relation: Below
	Below	RectA.yMax < RectB.yMin
		Opposite MBB relation: Above
	Over	(RectA.yMin == RectB.yMax || RectA.yMax == RectB.yMin) && (!((RectA.xMax <= RectB.xMin) || (RectA.xMin >= RectB.xMax)) && (!((RectA.zMax <= RectB.zMin) || (RectA.zMin >= RectB.zMax)))
		Opposite MBB relation: Below
	Front	RectA.zMin >= RectB.zMax
		Opposite MBB relation: Behind
	Behind	RectA.zMax <= RectB.zMin
		Opposite MBB relation: Front

Initially, it is checked whether the rightmost boundary of the current MBB has lower value than - or equal value to - the leftmost boundary region of the reference MBB. On this occasion, the first object is located on the *left* side of the second object and the opposite relation is attributed to the second parallelepiped. Otherwise, if the leftmost boundary region of the tested MBB has a value higher than, or equal to, the rightmost boundary region of the reference MBB, then the first object is located on the *right* side of the second object, with left relation corresponding to the second parallelepiped. Following the same pattern, if the lowest boundary region of the tested MBB has a higher value than the highest possible boundary region of the reference MBB, then the first object is located above the second object. However, in case these two boundaries are intersecting (even at a single point) in 3D space, then the first object is instead found to be *over* the second object. Regarding their highest boundary region check against the lowest one of their reference MBB, the first object is always located *below* the second object on both occasions. Finally, it is checked whether the most posterior boundary region of the first MBB has a value higher -or equal to the frontmost boundary region of the reference MBB. In case it does, then this object is in *front* of the second object. On the other hand, if the frontmost boundary region has a value lower or equal to the most posterior boundary region of the reference MBB, then the first object is located *behind* the second object. All formulas used for the deduction of each directional relation are displayed in Table 2.

Once more, boundary representation methodologies were established in order to deduct the appropriate directional relations for each indexed object. However, a 3D R-tree instance with thousands of objects generates an exponential number of MBB relationships, resulting in a small but considerable increase in the overall runtime. For this reason, the presented computational model takes advantage of Web Worker scripts [3] due to its unhindered operation in the background, following the trends of parallel computing on the Web [4]. Doing so, ensures that the X3DOM's presentation layer remains independent of the underlying 3D R-tree structure, allowing end-users to freely interact with the rest of the provided platform. Last but not least, such scripts are also capable of exploiting multicore machines in a more efficient way than the classic JavaScript programming methodologies, guaranteeing their hardware scalability .

4. Case Studies

As has been stated before, 3D R-tree complies with VR environments authored in X3D language and accessed through the X3DOM framework. However, the generic structure of any R-tree instance remains independent of these two data formats. In this way, the result-set of a spatial reasoning operation can be further manipulated by other platforms and semantic services, or presented with various graphical approaches. In this section, a collection of demos is presented and described in order to denote possible application areas of 3D R-tree.

4.1 Semantic Mapping of Spatial Relations

Several research works in the recent past developed innovative mechanisms for the spatial annotation of 3D content, using either a hybrid metadata acquisition framework [5] for the management of such annotations in online scenes, or an ontological framework [6] composed of various OWL-DL properties for mapping spatial correlations between 3D objects of indoor environments. However, both approaches lacked the ability to provide an automated spatial reasoning mechanism, since their primary target was the sufficient quantitative and qualitative annotation of a 3D space. In this use case, we deal with this inefficiency by granting to DEC-O's object properties the spatial relations estimated by our computational model. This supplement is done automatically with the assistance of Apache Jena[1] without any input or further action from the end-user of the application. For the presentation purposes of this scenario, we have created and populated with various 3D objects the realistically decorated indoor scene shown in Fig. 1.

---

[1] https://jena.apache.org/

In order to avoid the generation of numerous spatial relations, a small only subset - namely a *dining table*, a *colorful orb* and a *shelf* - of its 3D objects was selected to demonstrate R-tree's semantical capabilities. These three objects were indexed into a 3D R-tree instance, where our computational model was applied to extract the spatial relations shown in the same figure. At first, each object is found to satisfy the *disjoint* topological relation, initiating in this way the implication of their directional relations. The dining table is found to be on the *right* side of the colorful orb and the shelf, while at the same time, it is placed in *front* of and *below* both of them. The inverse directional relationships, *left*, *behind* and *above*, are automatically applied with respect to the colourful orb and shelf without proceeding to extra reasoning procedures. On the other hand, the colourful orb and shelf share a common boundary region, indicating a *touch* topological relation which also denotes that the colorful orb is *over* the shelf. Finally, the latter object is found to be *below* the former and the spatial reasoning terminates since no other directional relation pattern is satisfied. All these automatically implicated relations were then forwarded through DEC-O to the OWL ontology representing this interior space, filling in its corresponding axioms for the semantic representation of 3D space with linguistic predicates. In Fig. 1 above, these axioms are displayed in the form of object properties via the open-source ontology editor Protégé 3.4.4.

Figure 1. Automated implication and semantic matching of spatial relations

4.2 Blending Bio-medical Models

The implemented computational model can be also applied in domains of medical evaluation for the provision of a different perspective to their relative VR environments. For the purposes of this demo, we took advantage of human body polygon mesh 3D data that correspond to full-body MRI images extracted from patients. These data are publicly available from BodyParts3D [21], a dictionary-type database for anatomy in which anatomical concepts are represented by segments of a 3D whole-body model of an adult human male. It is worthwhile to mention that this library encompasses with the morphological and geometrical knowledge in anatomy, complying at the same time with the FMA (Foundational Model of Anatomy) ontological representation model.

Our set of organs constitutes the X3DOM scene shown in Fig 2, which is comprised of the heart, lungs, liver, kidneys, pancreas, spleen, stomach, and ureters organs. Each one of them was passed by the mesh simplification procedure provided through MeshLab[2] software system to reduce their excessive number of polygons, avoiding however any undesired distortion and keeping their quality levels high. The presented bio-medical environment demonstrates how internal organs can be successfully identified inside a patient's body, even though each organ's shape, size and relative position is highly patient-specific and unique for every human being. The automated extraction of spatial information and its correlation with linguistic semantics is instantaneously done each time users click on an organ. The overall procedure is based on the artificial selection of only the appropriate relations, instead of deducting the entire set of possible relations between R-tree's records. Moreover, a 10 percent offset has been also applied to each spatial relation formula to

---

[2] http://meshlab.sourceforge.net/

minimize boundaries' error rate and, at the same time, maximize the spatial accuracy of the result-set. In this way, records which have their bounding boxes overlapping each other by only a small percentage amount, are treated as disjoint objects and are granted the necessary directional relations. All the aforementioned features allow users to interact with an intangible information space to not only derive human organs without having a physical contact with them, but to also enhance their spatial reasoning ability with the visualization of the corresponding 3D R-tree instance.
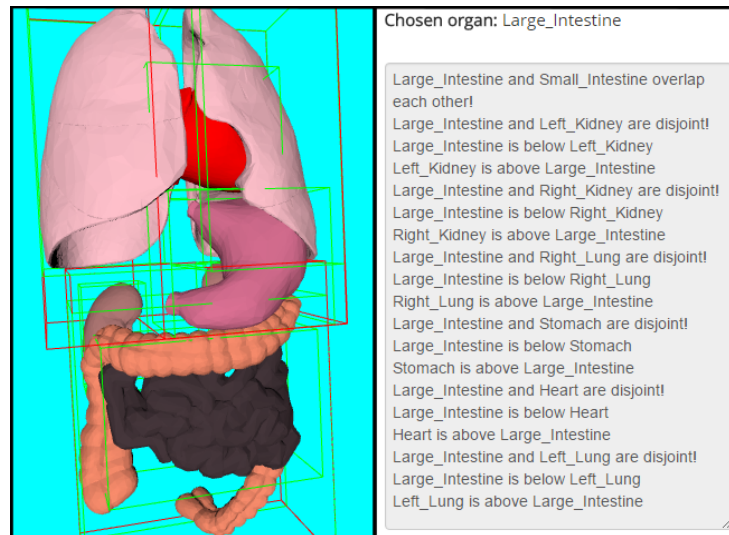

Figure 2. The selective spatial correlation of a human organ.

6. Conclusions

This paper introduces an interoperable and practical computational model that combines a Quadratic R-tree data structure with a set of the DE-9IM spatial relations, both implemented and optimized for application in Web3D space. The model takes advantage of the MBBs formed from the spatially indexed content of X3DOM scenes to extract the relative topological and directional relations between each record. Adopting such boundary-intersection methodologies, possible complex and heavyweight calculations are thus avoided, significantly decreasing the overall spatial reasoning process. However, despite the fact that R-tree's construction and spatial semantics extraction costs were trivial for all tested case studies, the Quadratic algorithm increases the search space redundancy during the last step of its operation. This is because the remaining entries are placed in a single node without any geometrical checks, increasing in this way the area coverage and/or the overlapping MBBs. Even though that such a thing takes place on rare only occasions, alternative splitting solutions [19] may be deployed and replace the current splitting methodology for those domains that require optimal construction and computation costs. Finally, the presented use cases demonstrated the semantic capabilities provided by the combination of a spatial reasoning model and a flexible 3D R-tree data structure, along with their utilization potentialities in diverse areas of interest.

References

[1]   Brutzman, D., & Daly, L. (2010). X3D: extensible 3D graphics for Web authors. Morgan Kaufmann.

[2]   Strobl, C. (2008). Dimensionally Extended Nine-Intersection Model (DE-9IM). In Encyclopedia of GIS (pp. 240-245). Springer US.

[3]   HTML Living Standard – Web Workers. (2015, September 25). Available: https://html.spec.whatwg.org/multipage/workers.html

[4]   Herhut, S., Hudson, R. L., Shpeisman, T., & Sreeram, J. (2012, June). Parallel programming for the web. In Proceedings of the 4th USENIX conference on Hot Topics in Parallelism, HotPar (Vol. 12, p. 1).

[5]   Tuovinen, L., Kahelin, J., & Roning, J. (2015, February). A conceptual framework for middle-up-down semantic annotation of online 3D scenes. In Semantic Computing (ICSC), 2015 IEEE International Conference on (pp. 464-469). IEEE.

[6]   Kontakis, K., Steiakaki, M., Kapetanakis, K., & Malamos, A. G. (2014, August). DEC-O: an ontology framework and interactive 3D interface for interior decoration applications in the web. In Proceedings of the Nineteenth International ACM Conference on 3D Web Technologies (pp. 63-70). ACM.

[7]   Guttman, A. (1984). R-trees: a dynamic index structure for spatial searching (Vol. 14, No. 2, pp. 47-57). ACM.

[8]   Pittarello, F., & De Faveri, A. (2006, April). Semantic description of 3D environments: a proposal based on web standards. In Proceedings of the eleventh international conference on 3D web technology (pp. 85-95). ACM.

[9]   Piórkowski, A. (2011). Mysql spatial and postgis–implementations of spatial data standards. Electronic Journal of Polish Agricultural Universities, 14(1), 1-8.

[10]   Mark, D. M., & Egenhofer, M. J. (1994). Modeling spatial relations between lines and regions: combining formal mathematical models and human subjects testing. Cartography and geographic information systems, 21(4), 195-212.

[11]   Papadias, D., Sellis, T., Theodoridis, Y., & Egenhofer, M. J. (1995). Topological relations in the world of minimum bounding rectangles: a study with R-trees (Vol. 24, No. 2, pp. 92-103). ACM.

[12]   Kontakis, K., Steiakaki, M., Kalochrsitianakis, M., Kapetanakis, K., & Malamos, A. G. (2015). Applying Aesthetic Rules in Virtual Environments by Means of Semantic Web Technologies. In Augmented and Virtual Reality (pp. 344-354). Springer International Publishing.

[13]   Borrmann, A., & Rank, E. (2009). Topological analysis of 3D building models using a spatial query language. Advanced Engineering Informatics, 23(4), 370-385.

[14]   Daum, S., & Borrmann, A. (2014). Processing of Topological BIM Queries using Boundary Representation Based Methods. Advanced Engineering Informatics, 28(4), 272-286.

[15]   Freeman, J. (1975). The modelling of spatial relations. Computer graphics and image processing, 4(2), 156-171.

[16]   Hayward, W. G., & Tarr, M. J. (1995). Spatial language and spatial representation. Cognition, 55(1), 39-84.

[17]   Johannsen, K., Swadzba, A., Ziegler, L., Wachsmuth, S., & De Ruiter, J. P. (2013). A Computational Model for Reference Object Selection in Spatial Relations. In Spatial Information Theory (pp. 358-376). Springer International Publishing.

[18]   Li, G., & Tang, J. (2010, July). A new DR-tree K-nearest neighbor query algorithm based on direction relationship. In Environmental Science and Information Application Technology (ESIAT), 2010 International Conference on (Vol. 2, pp. 246-250). IEEE.

[19]   Al-Badarneh, A. F., Yaseen, Q., & Hmeidi, I. (2010). A new enhancement to the R-tree node splitting. Journal of Information Science, 36(1), 3-18.

[20]   Konstantinos Kontakis, Athanasios G. Malamos, Malvina Steiakaki and Spyros Panagiotakis, "Spatial Indexing of Complex Virtual Reality Scenes in the Web", International Journal of Image and Graphics Vol. 17, No. 2 (2017) 00523 (18 pages), DOI: 10.1142/S0219467817005235

[21]   Mitsuhashi, N., Fujieda, K., Tamura, T., Kawamoto, S., Takagi, T., & Okubo, K. (2009). BodyParts3D: 3D structure database for anatomical concepts. Nucleic acids research, 37(suppl 1), D782-D785.

[22]   Funkhouser, T., Min, P., Kazhdan, M., Chen, J., Halderman, A., Dobkin, D., & Jacobs, D. (2003). A search engine for 3D models. ACM Transactions on Graphics (TOG), 22(1), 83-105.

[23]   Karpathy, A., Miller, S., & Fei-Fei, L. (2013, May). Object discovery in 3d scenes via shape analysis. In Robotics and Automation (ICRA), 2013 IEEE International Conference on (pp. 2088-2095). IEEE.

[24]   Evans, A., Romeo, M., Bahrehmand, A., Agenjo, J., & Blat, J. (2014). 3D graphics on the web: A survey. Computers & Graphics, 41, 43-61.

[25] Behr, J., Eschler, P., Jung, Y., & Zöllner, M. (2009, June). X3DOM: a DOM-based HTML5/X3D integration model. In Proceedings of the 14th international conference on 3D web technology (pp. 127-135). ACM.

[26] Spala, P., Malamos, A. G., Doulamis, A., & Mamakis, G. (2012). Extending MPEG-7 for efficient annotation of complex web 3D scenes. Multimedia Tools and Applications, 59(2), 463-504.

[27] Bochicchio, M. A., Longo, A., & Vaira, L. (2011, September). Extending Web applications with 3D features. In Web Systems Evolution (WSE), 2011 13th IEEE International Symposium on (pp. 93-96). IEEE.

[28] Tejada, S., Knoblock, C. A., & Minton, S. (2001). Learning object identification rules for information integration. Information Systems, 26(8), 607-633.