Topology of the Prism Model for 3D Indoor Spatial Objects

Joon-Seok Kim, Hye-Young Kang, Tae-Hoon Lee, Ki-Joune Li
Department of Computer Engineering
Pusan National University
joonseok@pnu.edu, hyezero@pnu.edu, expyrex@gmail.com, lik@pnu.edu

Abstract

Topological relationships between spatial objects are an essential property of spatial objects. They are used for spatial analysis and query processing. In this paper, we first introduce an alternative 3D geometric model, called the prism model*. The prism model is based on an extrusion method from 2D footprints of 3D objects. Then we study the correspondence between the 2D topology of footprints and the topologies of 3D objects using the prism model. We propose a method of topological analysis to implement topological operators for the prism model based on this study. This method is simpler and more efficient than the methods of the 3D spatial model. The topological operators are easy to implement since we can extend the 2D Simple Feature Geometry of OGC provided by most spatial DMBS.*

1. Introduction

The 3D topology between spatial objects is an essential component of indoor spatial information used to analyze spatial properties in indoor space. The topology of 3D spatial objects is more difficult to analyze than 2D topology due to the complexity of 3D geometry. Therefore, the implementation of topological operators in 3D spatial DBMS is more difficult and inefficient than spatial DBMS for 2D spatial objects due to the geometric complexity. However, most spatial objects in indoor space are not so complex that we need a full 3D geometry model, such as the model of ISO 19107 [4].

In this paper, we introduce a simple but efficient geometry model, called the *prism model* to describe 3D geometry of indoor building space. Using the prism model, we define a 3D spatial object with upper and lower bound geometries that share the same 2D footprint. We first observe the correspondence between the topology of 2D footprints and the 3D topology of the spatial objects extended from 2D footprints to develop a method of topological analysis for the prism model. We propose a method of topological analysis for the spatial objects of the prism model based on this observation. This method is applied to implement topological relationship operators using the existing 2D spatial DBMS that provides 2D Simple Feature Geometry.

The remainder of this paper is organized as follows. In the next section, we introduce the prism model and review the research related to 3D topological operations. In section 3, we examine topological relations with the prism model. We propose an efficient algorithm to implement 3D topological operators in section 4. We conclude this paper in section 5.

2. Related Work on 3D Geometric Model

With the recent increasing demand for 3D spatial information, we need a robust data model of 3D spatial objects to meet the requirements of diverse applications. Several studies [4, 14, 6, 12] have proposed different representation techniques proposed to depict 3D solids in diverse application areas, such as computer aided design, computer vision and geographic information systems. The data models of ISO 19107 spatial schema [4] and KML are the most important schema defined as international standards by ISO and OGC respectively. The data model specified by ISO 19107 provides a strong expressive power of 3D spatial information. GML [9] is based on the spatial data model of ISO 19107. However, this data model has serious drawbacks. First, the size of data in ISO 19107 is large due to the multifaceted structure of its 3D model. Second, the complex model of ISO 19107 leads to performance degradation of geometric and topological operations. These drawbacks result in difficult and heavy implementation of ISO 19107.

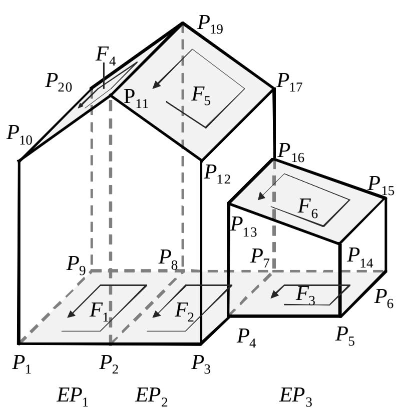
Conversely, the data model employed by KML, developed for the use of Google Earth, includes a simple 3D spatial data model. The main purpose of this data model is the visualization of 2D and 2.5D spatial objects. The dataset size in KML is smaller, and the systems supporting KML are less complex that for GML. However, the expressive power of KML is limited due to its simple spatial data model. Therefore COLLADA [1] must be used with KML to describe complex 3D spatial objects.

Several spatial DBMS for 3D objects provide operations for topological relationships [13]. For example, Oracle 11g represents 3D spatial objects using the B-Rep[6] model and processes any 3D topological relationship query based on $point - in - solid$ operation and Jordan curve theorem in 3D. In [7], a 3D geo-DBMS is introduced. It provides 3D topological operations, based on 4-intersection model including $overlap$, $meet$, $disjoint$, $inside$, $covers$, $covered\_by$, $contain$ and $equal$.

Our study is motivated by the observation that most spatial objects in indoor space do not need the complexity of ISO 19107. We need stronger expressive power than provided by KML to describe the objects of interiors. The objectives of our study are to provide sufficiently descriptive but simple 3D spatial model and develop a method for topological operators for indoor spatial DBMS.

3. Prism Model

The prism model is an alternative 3D spatial data model proposed [8] to provide sufficient expressive power and whilst achieving reasonable performance. The prism refers to polygonal prisms, not just triangular prisms. This model is based on the extrusion technique to represent 3D objects from 2D footprints, like the model in KML. We generalized KML to handle more diverse shapes with upper and lower geometries, as shown in Figure 1. In this figure, a building object is described by a set of three prism objects (extrusive polygon), which are bounded by upper and lower geometries.



$EPS_1 = \{EP_1, EP_2, EP_3\}$
$EP_1 = (F_1, F_4)$
$F_1 = (P_1, P_2, P_8, P_9)$
$F_4 = (P_{10}, P_{11}, P_{19}, P_{20})$
$EP_2 = (F_2, F_5)$
$F_2 = (P_2, P_3, P_7, P_8)$
$F_5 = (P_{11}, P_{12}, P_{17}, P_{19})$
$EP_3 = (F_3, F_6)$
$F_3 = (P_4, P_5, P_6, P_7)$
$F_6 = (P_{13}, P_{14}, P_{15}, P_{16})$

Figure 1. Examples of prism model

The geometry schema of the prism UML class is based on the SFG [5] model of OGC. Figure 2 shows the diagram of the basic prism model. There are three types of extrusive geometry; extrusive point, extrusive curve, and extrusive surface. These are a subclass of curve, surface, and solid respectively. Each extrusive geometry is bounded by lower and upper geometries that have the same 2D footprint. The vertices of an upper geometry shapes have higher z-values than lower geometry shapes.

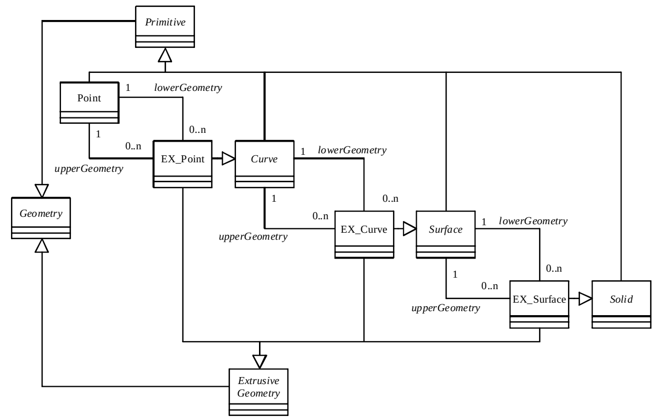


Figure 2. Diagram of the prism model

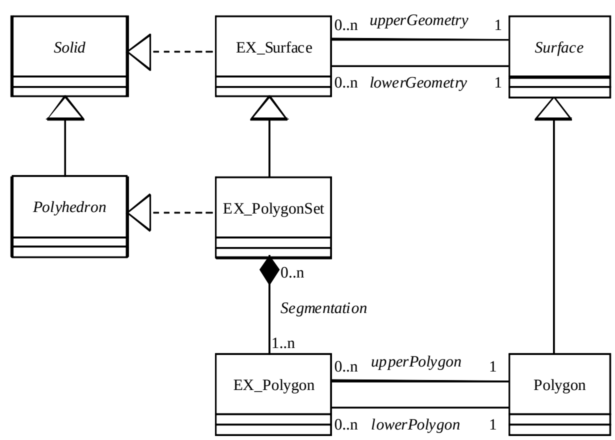
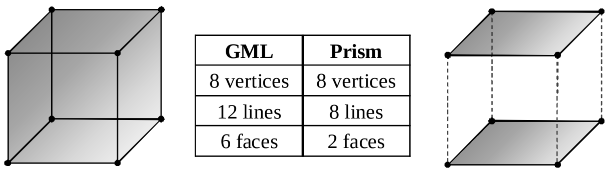


Figure 3. Diagram of extrusive polygon

A solid is represented by a set of extrusive surfaces called prisms. For example, a complicated polyhedron (Figure 3) is decomposed into a set of prisms, so that each prism has flat lower and upper surfaces, as in Figure 1.

The prism model has several advantages. First, the prism model is simpler and contains fewer geometric components than the ISO 19107 and consequently the dataset size in the prism model is less than GML, as shown Figure 4. Storage, data transfer, and geometric computation are less expensive than ISO 19107, due to the smaller dataset size.

Second, we improve the query processing performance by a 2D filtering technique to process 3D spatial queries, like the filtering and refinement policy of 2D spatial query processing [3]. For example, given a 3D containment query, we first filter only the spatial objects whose 2D footprints



GML	Prism
8 vertices	8 vertices
12 lines	8 lines
6 faces	2 faces

Figure 4. Comparative dataset size of GML and prism model

are contained by the 2D footprint of the query region and then refine the small number of candidates obtained from the filtering step to check if they are really in the given 3D range. This is a practical approach for indoor applications since the shapes of 3D objects, such as buildings and rooms approximate prisms.

Third, 3D geometric computation by the prism model is simpler and easier to implement than full 3D geometric computation [2]. For example, for the point-in-polyhedron computation, we only need to check if the point is in one of the prisms of the polyhedron represented as a set of prisms. The point in the prism is very simple to implement.

Fourth, most 3D geometries can be described by the prism model. A curved surface is and exception. We can define the lower and upper surfaces as curved surfaces, although we simplify them to flat surfaces in this paper. Figure 5 shows an intuitive comparison between the expressive powers of the 3D spatial data models. The model of ISO 19107 or GML contains most 3D geometries, while the TEN (TEtrahedronized irregular Network) model [10, 11, 12] includes only polyhedrons. The expressive power of the prism model is almost equivalent to that of the TEN model, except that the primitive shape is not a tetrahedron, but a prism. This implies that the prism model is more appropriate for indoor spatial objects than the TEN model. We need a more rigorous study on the expressive power of those models.

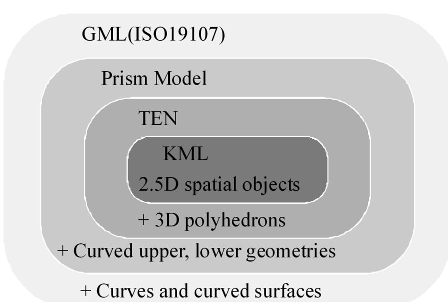


Figure 5. Expressive power of 3D spatial models

4. Topological Relationship of Prism Model

In this paper, the topological relationship is based on the eight topological relationship model between 3D spatial objects: $disjoint$, $covers$, $covered\_by$, $contains$, $inside$, $meet$, $overlap$ and $equal$.

We discuss the topology of 3D objects from two different viewpoints to describe the topology between two 3D prism objects

- topology of 2D footprints and
- topology of elevations

4.1. Topology of 2-D footprints

We first observe the correspondence between the topology for 3D prisms and the topology for their 2D footprints to determine 3D topological relationship using prism model. For example, if the topology between two prisms is equal, then the topology between their 2D footprints must also be equal. Using similarities, we can derive a set of possible 3D topologies from a 2D topology of footprints. For two 3D prisms objects, we call the topology between their 2D footprints *TP-F*, where the topology between the original 3D prism objects is termed *TP-3D*.

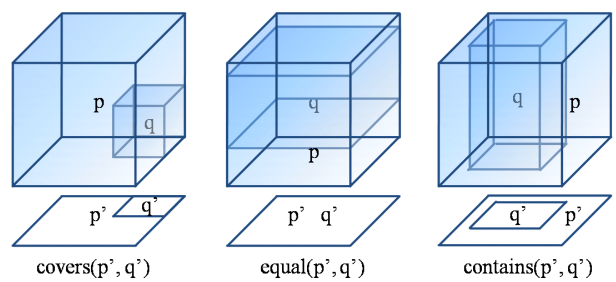


Figure 6. Examples of TP-F

Figure 6 shows examples that have the same TP-3D ($meet$) but different TP-F. We find a correspondence between TP-3D and TP-F, as depicted in Figure 7.

When TP-F $overlap$s, TP-3D should be one of $overlap$, $meet$ and $disjoint$ (see Figure 8) according to the elevation of object q. This means that we should compare the elevations of lower and upper geometries to determine the exact topological relationship.

4.2. Topology of Elevation

We also need to evaluate the topology of elevations in addition to the topology of footprints to determine 3D topology between prism objects. Given two prism objects bounded by lower and upper geometries, ten primitive topological relationships are possible: $upper\_disjoint$,

	TP-F							
TP-3D	e	i	ct	cv	cb	o	m	d
e	1	0	0	0	0	0	0	0
i	0	1	0	0	0	0	0	0
ct	0	0	1	0	0	0	0	0
cv	1	0	1	1	0	0	0	0
cb	1	1	0	0	1	0	0	0
o	1	1	1	1	1	1	0	0
m	1	1	1	1	1	1	1	0
d	1	1	1	1	1	1	1	1

e : eqaul
i : inside
ct : contains
cv: covers
cb: covered by
o : overlap
m : meet
d : disjoint

0 : impossible
1 : possible

Figure 7. Possible relationships between TP-3Ds and TP-Fs

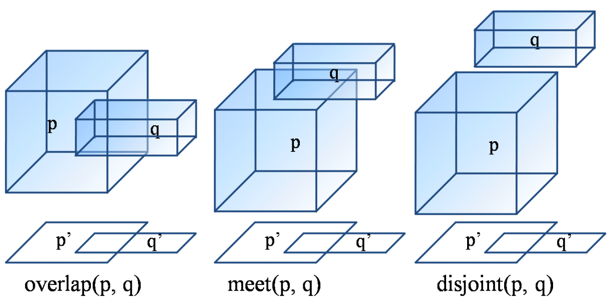


Figure 8. Examples of different elevations

$upper\_meet$, $equal$, $overlap$, $contains$, $inside$, $covers$, $covered\_by$, $lower\_disjoint$, $lower\_meet$ (Figure 10). The topology of elevation is defined between two prism objects sharing the same footprint. If the footprints are different, we clip the footprints into only the overlapping region (Figure 9). We call this topology of elevation between two prism objects sharing the same footprint, $TP-ER$ (Topology of Elevation on a Region).

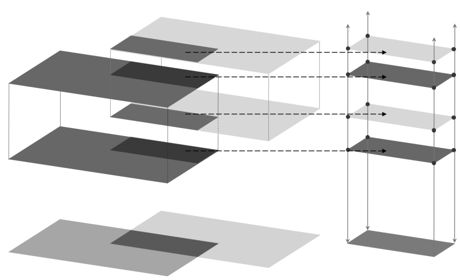


Figure 9. Clipping the footprint for TP-ER

The topology of elevation between two prism objects can be derived from the elevation topology at vertices of the footprint polygon, since the upper and lower geometries are planar. We call the topology of elevation at vertices $TP-EP$ (Topology of Elevation at Point). As shown in Figure 11, the TP-EP at point v_1 and v_2 are $overlap$ and $contains$ respectively. Then the topology of elevation between two prism objects A and B is determined as the $overlap$ by the topology decision directed acyclic graph.

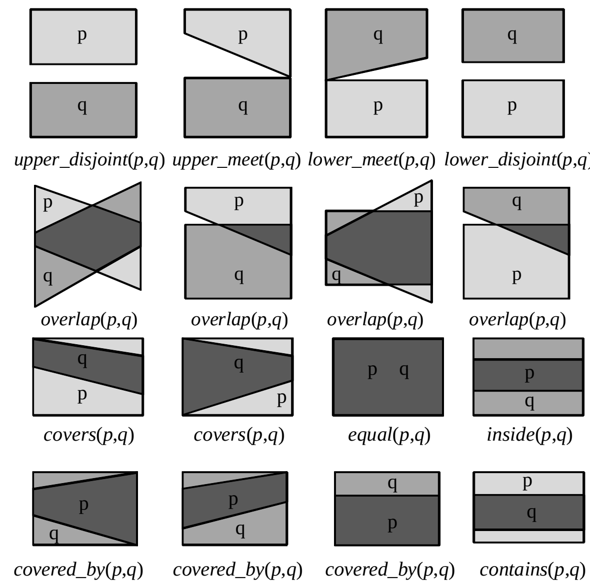


Figure 10. Topologies of elevation

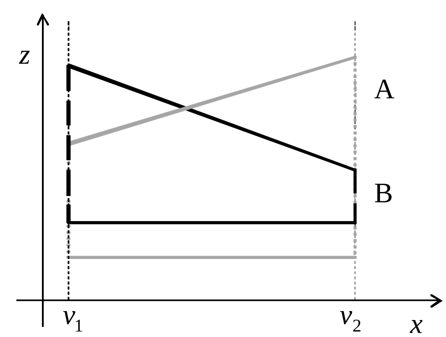


Figure 11. Example of TP-EP and TP-ER

The decision directed acyclic graph for TP-ER in Figure 12 allows derivation of the topology of elevation between prism objects (TP-ER) from the topology of elevation at points (TP-EP). Starting from the node of the decision DAG for TP-EP at each vertex of the footprint polygon, we ascend the DAG until the lowest common ancestor node is found. For example, when the TP-EP at v_1 and v_2 are $inside$ and $equal$, then the TP-ER between the prism objects is determined as $covered\_by$ using the decision DAG.

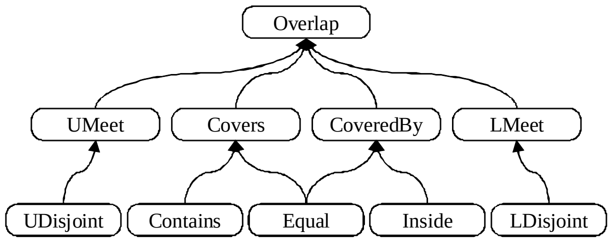


Figure 12. Topology decision DAG for elevation

4.3. Topology of 3D

The TP-3D decision DAG of Figure 13 is used to combine the TP-F and TP-ER into TP-3D in a similar way to TP-ER decision DAG. The $upper\_meet$ and $lower\_meet$ of TP-ER are considered as $meet$, and $upper\_disjoint$ and $lower\_disjoint$ of TP-E are considered as $disjoint$ in the TP-3D decision DAG. For example, given two 3D spatial objects in the prism model, if the TP-F of the objects is $overlap$ and the TP-ER of the objects is $upper\_meet$, then TP-3D is $meet$.

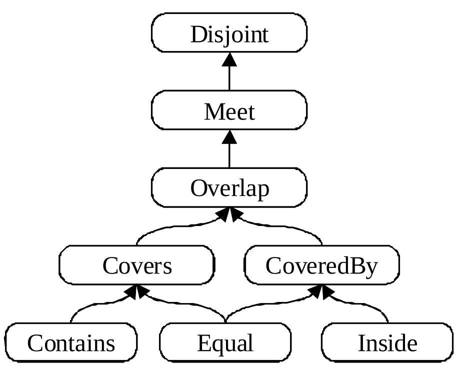


Figure 13. TP-3D decision DAG

5. Implementation of 3D Topological Operator

In this section, we propose a topological query processing method based on the topology observation for the prism model in the previous sections. For a given topological relation query, algorithm 1 returns the query result as Boolean.

Algorithm 1: 3D Topological relation query processing

```
input  : 3D topological relation TP-3D, geometries G_A and G_B
output: Boolean value for the topological relations between G_A and G_B
TP-F ← ComputeTP-F(G_A, G_B);
if TP-F is one of possible relations of TP-3D then
    polygon ← Intersection2D(G_A, G_B);
    TP-EPs ← {};
    foreach vertex p in polygon do
        TP-EP ← ComputeTP-EP(G_A, G_B, vertex p);
        Append TP-EP to TP-EPs;
    TP-ER ← MergeTP-ER(TP-EPs);
    Result ← DeriveTP-3D(TP-ER, TP-F);
    if Result = TP-3D then
        return true;
    else
        return false;
else
    return false;
```

Since this algorithm checks if the given topological relation is true, we first check if the topological relationship of footprints is possible. If it is not a possible topology, then we terminate the algorithm by returning False. Otherwise, we compute the 3D topology, combining TP-ER and TP-F, as explained in the previous section.

We can use the function provided by the 2D spatial DBMS to implement the `ComputeTP-R` and `Intersection2D` functions. The time complexity of this algorithm is determined by the number of vertices of clipped footprint and the cost of the two functions provided by 2D spatial DBMS. Even though the algorithm evaluates a given topological relationship, it can be modified to determine the topology of two 3D prism objects.

6. Conclusion

We need a 3D geometric and topological model to provide a spatial DBMS for indoor spatial objects. In this paper, we analyzed the topological relationships between 3D spatial objects with the prism model, an alternative 3D spatial data model. The prism model is based on the extrusion technique from the 2D footprint. The 3D geometric objects are represented by a pair of upper and lower geometries in the prism model. As the 2D footprints of upper and lower geometries are the same, the geometry preserves 2D spatial properties. We derived topological relations between 3D geometry from a combination of the 2D footprint and elevation in 3D space.

The advantages of the proposed methods are summarized as follows;

- Since the geometric and topological models are simple, the processing cost is relatively inexpensive.
- We can reuse the 2D spatial DBMS to implement geometric and topological operations for 3D objects.

The prism model is appropriate for indoor spatial objects that are vertically aligned, such as rooms and walls. For this reason, we applied the prism model to indoor spatial objects. We expect the prism also supports 3-D building objects; this issue may be included by the future work. This paper only deals with the topology between two prism objects. In future work, we will extend the topology between two 3D objects that are described as sets of prism objects. The performance of the proposed methods will be studied to compare it to previous methods.

7. Acknowledgements

This research was supported in part by the Brain Korean 21 Project and a grant (08KLSGC04) from Cutting-edge Urban Development - Korean Land Spatialization Research Project funded by Ministry of Land, Transport and Maritime Affairs.

References

[1] M. Banes and E. L. Finch. *COLLADA-Digital Asset Schema Release 1.5.0 specification*. Khronos Group, Sony Computer Entertainment Inc., April 2008.

[2] H. D. Eberly. *3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2006.

[3] A. Guttman. R-trees: a dynamic index structure for spatial searching. In *SIGMOD '84: Proceedings of the 1984 ACM SIGMOD international conference on Management of data*, pages 47–57, New York, NY, USA, 1984. ACM.

[4] J. R. Herring. *The OpenGIS Abstract Specification, Topic 1: Feature Geometry (ISO 19107 Spatial Schema)*. Open Geospatial Consortium, Inc., 2001. Version 5, OGC Doc. No. 01-101.

[5] J. R. Herring. *OpenGIS Implementation Specification for Geographic information - Simple feature access - Part 1: Common architecture*. Open Geospatial Consortium, Inc., 2006. Version 1.2.0, OGC Doc. No. 06-103r3.

[6] C. M. Hoffmann. *Geometric and solid modeling: an introduction*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1989.

[7] C. T. Khuan and A. Abdul-Rahman. Implementing 3d geometrical data type and spatial operations for postgis. In *The Proceedings of the 3rd International Workshop on 3D Geo-information*, pages 17–35, Seoul, South Korea, 2008.

[8] J.-S. Kim, T.-H. Lee, and K.-J. Li. Prism geometry: Simple and efficient 3-d spatial model. In *The Proceedings of the 3rd International Workshop on 3D Geo-information*, pages 139–145, Seoul, South Korea, 2008.

[9] Open Geospatial Consortium, Inc. *OpenGIS Geography Markup Language (GML) Encoding Standard*, 2007. Version 3.2.1, OGC Doc. No.07-036.

[10] F. Penninga, P. v. Oosterom, and B. M. Kazar. A tetrahedronized irregular network based dbms approach for 3d topographic data modeling. In A. Riedl, W. Kainz, and G. A. Elmes, editors, *Progress in Spatial Data Handling*, pages 581–598. Springer Berlin Heidelberg, 2006.

[11] F. Penninga, P. v. Oosterom, and B. M. Kazar. First implementation results and open issues on the poincaré-ten data structure. In P. v. Oosterom, S. Zlatanova, F. Penninga, and E. Fendel, editors, *Advances in 3D Geoinformation System*, pages 581–598. Springer, 2007.

[12] M. Pilouk. *Integrated modelling for 3D GIS*. PhD thesis, ITC Enschede, Netherlands, 1996.

[13] S. Ravada, B. M. Kazar, and R. Kothuri. Query processing in 3d spatial databases: Experiences with oracle spatial 11g. In J. Lee and S. Zlatanova, editors, *3D Geo-Information Sciences*, pages 153–173. Springer, 2008.

[14] T. Wilson. *OGC KML*. Open Geospatial Consortium, Inc., 2008. Version 2.2.0, OGC Doc. No.07-147r2.