# A Spatial Indexing Approach for Protein Structure Modeling

Wendy Osborn
Department of Mathematics and Computer Science
University of Lethbridge
Lethbridge, Alberta
T1K 3M4 Canada
wendy.osborn@uleth.ca

## Abstract

*This paper explores the use of spatial indices for the modeling and retrieval of protein structures. With two existing spatial indices, a preliminary framework for protein structure modeling that uses a spatial index is proposed. It provides a novel technique for modeling. In addition, it provides additional flexibility with respect to modeling granularity and structure manipulation. It is expected that this modeling approach will lead to new ways of analyzing protein structures.*

## 1. Introduction

Protein structure analysis is an exciting and challenging research area in the area of bioinformatics [1]. Many repositories exist today that maintain three-dimensional protein structures and provide tools for search and retrieval. The first such repository is the Protein Data Bank (PDB) [4]. It remains a very popular source for information. The PDB currently maintains over 20,000 protein structures. The current representation of the three-dimensional structure in PDB is very inflexible and archaic [1]. Therefore, it is important to explore how three-dimensional protein structures are modeled for future analysis.

One promising approach is in the area of spatial databases [11]. Spatial indexing [5, 11] provides a technique for modeling and retrieval of data based on its location in multidimensional space. It also provides a framework that can be used in conjunction with existing protein analysis strategies.

The application of spatial indexing to protein structure modeling has not received significant attention. Srinivasa and Kumar [12] propose a platform for modeling three-dimensional structures in a database. They also present strategies for search and retrieval. One limitation of their work is that their use of spatial indexing is limited to a strategy for point data only. Yan, Yu and Han [13] propose a strategy for indexing a graph representation of a biomolecular structure. Although their approach is promising, it has a limitation of only being applicable to substructures.

The application of the approximation spatial index for the modeling protein structures is investigated. Given the properties of existing approximation strategies, a preliminary framework for protein structure modeling is proposed. This approach works across all descriptions of a protein structure, and also with different granularities of a model. For example, one can model at the atomic level, at the substructure level, or anywhere in between.

The remainder of this paper proceeds as follows. Section 2 presents some required background information on protein structure modeling and spatial indexing. Section 3 presents an application of existing spatial indexing structures for the modeling and retrieval of protein structures. Section 4 concludes with future research directions that need to be considered for this modeling strategy.

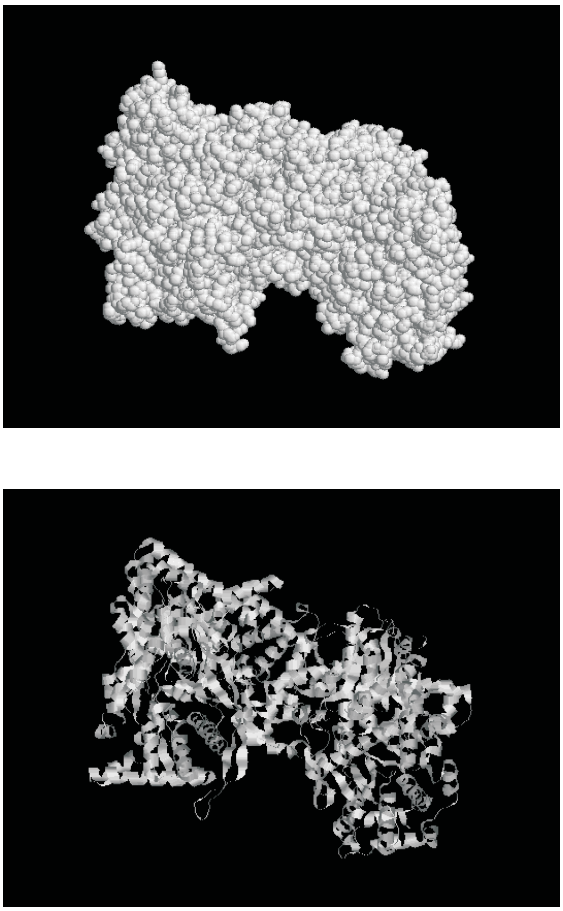
## 2. Preliminaries

This section presents some required background information that is required for this investigation. First, the hierarchical structure description, which is the common technique for describing proteins at multiple levels, is summarized with an example. Second, spatial indexing is summarized, with a focus on two existing strategies.

### 2.1. Hierarchical Structure Description

A protein structure can be described using a hierarchical method [1]. This layered approached is composed of four levels: primary, secondary, tertiary and quaternary. The primary level consists of the sequence of amino acids that make up the protein. The secondary level consists of the

21st International Conference on
Advanced Information Networking and Applications Workshops (AINAW'07)
0-7695-2847-3/07 $20.00 © 2007 IEEE



MSQPIFNDKQFQEALSRQWQRYGLNSAAEMTPRQWWLAVSEALAEMLRAQPFAKPV
ANQRHVNYISMEFLIGRLTGNNLLNLGWYQDVQDSLKAYDINLTDLLEEEIDPALGA
GGLGRLAACFLDSMATVGQSATGYGLNYQYGLFRQSFVDGKQVEAPDDWHRSNYP
WFRHNEALDVQVGIGGAVTKDGRWEPEFTITGQAWDLPVVGYRNGVAQPLRLWQAT
HAHPFDLTKFNDGDFLRAEQQGINAEKLTKVLYPNDNHTAGKKLRLMQQYFQCACS
VADILRRHHLAGRELHELADYEVIQLNDTHPTIAIPELLRVLIDEHQMSWDDAWAITS
KTFAYTNHTLMPEALERWDVKLVKGLLPRHMQIINEINTRFKTLVEKTWPGDEKVWA
.....

**Figure 1. Hierarchical Description for 1AHP**

substructures that the protein is composed of. Two common protein substructures are the helix ($\alpha$) and the beta sheet ($\beta$-sheet). The tertiary level consists of the final protein structure, which is composed of all substructures from the secondary level. Finally, the quaternary contains a structure of multiple proteins, each from a different tertiary level.

Figure 1 depicts an example of a hierarchical description at the primary, secondary and tertiary levels for the protein 1AHP, which is retrieved from the Protein Data Bank [4] and viewed using RasMol [2]. The primary level contains a portion of the amino acid sequence for 1AHP. The secondary level contains the secondary structures, each of which is a helix or a $\beta$-sheet. The tertiary level contains the protein structure in its entirety.

## 2.2. Spatial Indexing

A spatial index (or spatial access method) [5, 11] provides access to data based on its location in multidimensional space. Data that is indexed based on location usually consists of points, lines and objects of arbitrary shape. In addition, data with non-spatial information can be stored along with spatial data. For example, a map consists of towns (points), roads (lines) and cities (arbitrarily-shaped objects). Each datum on a map can have non-spatial data associated with it, such as a city name or a road number.

A spatial index supports many types of searches [5]. One common search type is the region search. Given an object that represents a region of space (usually a rectangle), the goal is to find all data that overlap the region.

Many spatial indices are proposed in the literature. A comprehensive survey is available in [5]. One particular class of spatial indices that is of interest for this proposed framework are approximation strategies. A general overview of an approximation spatial index is given next. Following this, two different approaches that adopt the approximation strategy - the R-tree [7] and the 2DR-tree [10, 9] are summarized.

An approximation spatial index maintains a hierarchy of approximations for objects and the subregions containing one or more objects. Most approximation strategies are based on the B+-tree [3], so they are height balanced (or, every path from the root node to leaf node is the same length).

An approximation for both objects and subregions is usually represented in the form of a *minimum bounding rectangle* ($MBR$). A minimum bounding rectangle defines the extent of an object along each dimension in space.

The hierarchy is maintained using nodes. Each node can contain a minimum of $m$ records and a maximum of $M$ records, where $M$ is the total number of records allowed in the node, and $m$ is a user defined value. Usually, $m = M/2$. The exception is the root note, which can contain at minimum of two records. Each record maintains:

$$(MBR_{(i,j)}, ptr_{(i,j)})$$

where $MBR_{(i,j)}$ is an approximation and $ptr_{(i,j)}$ is a pointer. In a leaf node, $MBR_{(i,j)}$ approximates an object and $ptr_{(i,j)}$ references the actual object on secondary storage. In a non-leaf node, $MBR_{(i,j)}$ encloses all approximations in the subtree referenced by $ptr_{(i,j)}$.

## 2.3. R-tree

The R-tree [7] is the first approximation spatial index proposed in the literature. As with the B+-tree, the R-tree uses linear nodes to organize approximations into a hierarchy. An example is described first, followed by a brief de-

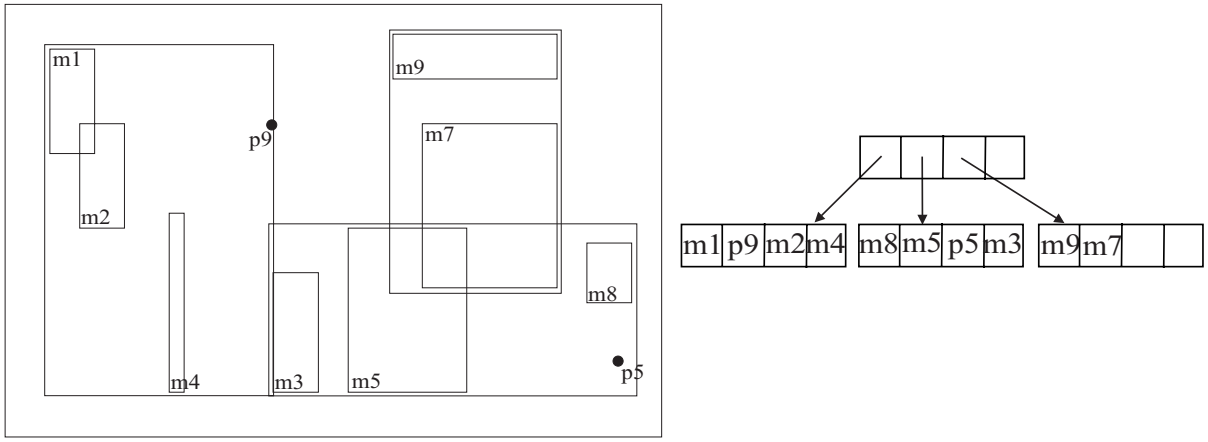



**Figure 2. The R-tree**

scription of the region search, insertion and deletion strategies.

Figure 2 depicts an example of an R-tree that is indexing a spatial data set (the data set comes from [5]). The approximations for actual objects are represented by points p5 and p9, and rectangles m1 to m5 and m7 to m9. In addition, there are three approximations that represent subregions - one that contains m1, m2, m4, and p9; one that contains m3, m5, m8 and p5; and one that contains m7 and m9. The leaf level contains the object approximations, while the non-leaf level (also the root level) contains the subregion approximations.

The R-tree region search begins at the root by examining all approximations to find the ones that overlap the query region. For all qualifying approximations, the search continues in the corresponding subtrees until it reaches zero or more leaf nodes. The approximations for all qualifying leaf nodes are retrieved and tested for overlap with the query region.

The insertion algorithm first identifies the appropriate leaf node for storing the new entry. The insertion path contains minimum bounding rectangles that require the least enlargement to include the new object. After inserting the new approximation into the chosen leaf node, the approximations along the insertion path are updated.

If overflow of the leaf node occurs, it is handled by splitting the node into two new nodes. The node is split so that the new nodes cover the smallest amount of area and both nodes are not checked for the same query. Three splitting strategies are proposed: exhaustive, quadratic, and linear. The exhaustive approach finds all candidate splits and chooses the best one. The quadratic and linear approaches identify the two objects that are the farthest apart and cluster the remaining objects into two nodes. The quadratic and linear approaches differ in how the two objects are identified. If a split causes an overflow in the parent node, the split is propagated up the tree as far as necessary. If it propagates to the root, a new root node is created.

The R-tree deletion strategy removes the approximation of the object to be deleted and adjusts the minimum bounding rectangles along the deletion path. Underflow is handled using a forced reinsertion strategy.

### 2.4. 2DR-tree

The 2DR-tree [10, 9] extends the one-dimensional structure of the R-tree [7] to two dimensions. Its structure and organization is summarized first, followed by an example and a description of the search, insertion and deletion strategies.

The nodes used in the 2DR-tree to organize approximations are two-dimensional in nature. The $M$ locations in a node are organized in the following manner. For each node $N$, $X$ is the number of locations along the $x$-axis of the node, and $Y$ is the number of locations along the $y$-axis.

An approximation is stored in an appropriate location with respect to all other approximation in the node. Using two-dimensional nodes allows spatial relationships between objects to be preserved. The spatial relationships that can be maintained in the 2DR-tree are north, northeast, east, southeast, south, southwest, west and northwest. A spatial relationship is defined between two objects using the centroids of their approximations. For example, approximation MBR 1 is northeast of MBR 2 if the centroid of MBR 1 is northeast of the centroid of MBR 2.

Organization of approximations is dictated by the following validity rules [10]. For each approximation $MBR$:

- All approximations located directly north of $MBR$ in the node have a centroid that is north, northwest, or west of the centroid for $MBR$ in space,
- All approximations located northeast of $MBR$ in the node have a centroid that is northeast of the centroid

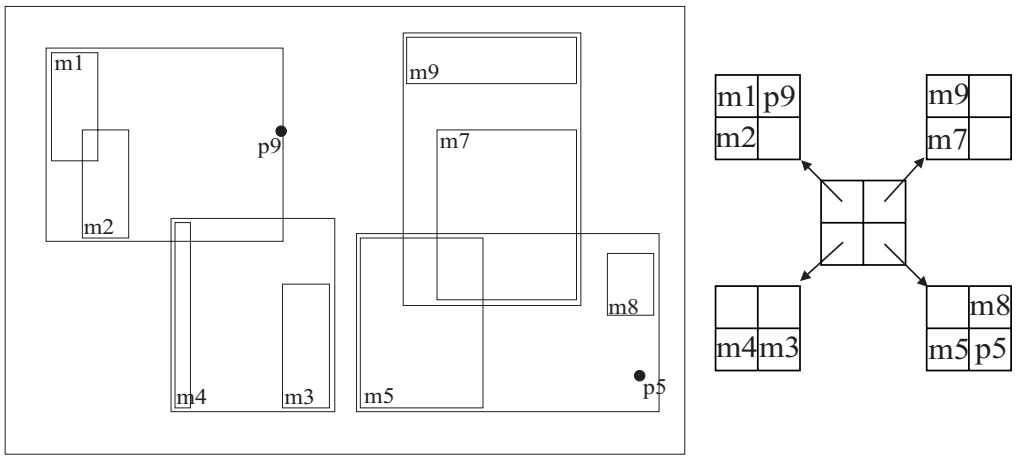



**Figure 3. The 2DR-tree**

for $MBR$ in space, and

- All approximations located directly east of $MBR$ in the node have a centroid that is east, southeast or south of the centroid for $MBR$ in space.

Figure 3 shows an example of a 2DR-tree that is indexing the same spatial data set from [5]. As with Figure 2, the leaf level contains the approximations for all points and objects, while the subregions are maintained in the non-leaf level. The difference here is how the spatial relationships are maintained. If we look at the subregion enclosing m4 and m3, we observe that the centroid of m3 is southeast of the centroid for m4, so both they are placed in appropriate locations with respect to each other in the leaf node. Similarly, the spatial relationships are also maintained between subregions in the non-leaf level.

Since approximations are organized, a binary search strategy [9] can be applied in the following manner. The search region is applied recursively to half of the objects in the node until all overlapping approximations are found.

The insertion strategy employs a greedy search for locating an insertion path contains a minimal (if not the absolute minimal) area increase required to insert a new approximation. At the leaf level, a location is found for the approximation that obeys all validity rules. Any overflow that occurs is handled using one of the splitting strategies proposed in [9, 10]. Deletion simply requires the removal of the objects and the updating of the approximations along the insertion path.

## 3. Structure Modeling Using Spatial Indices

This section presents the proposed framework for protein structure modeling. The framework can utilize either the R-tree or the 2DR-tree for modeling a protein structure. The hierarchical model described earlier consists of four levels - primary, secondary, tertiary, and quaternary. We discuss how a spatial index can be applied to each of the secondary, tertiary and quaternary levels. In addition, we discuss how a spatial index can be applied in different ways within the same level using different amounts of detail.

### 3.1. Secondary Level Modeling

Protein secondary structures can be modeled in different ways. One approach is to group helices and $\beta$-sheets that are adjacent to each other into substructures. Each substructure can be represented with an approximation. This approximation in turn can be placed in a spatial index structure. These substructures can be formed by inserting helices and $\beta$-sheets by location into the spatial index structure. This can potentially lead to the discovery of new substructures that serve an important purpose in the function of a protein. An entire substructure can be discovered independently, and can be inserted as a single unit into the spatial index structure.

Figure 4 depicts an example of substructure modeling using the R-tree, while Figure 5 depicts an example of substructure modeling using the 2DR-tree. In both cases, each substructure is enclosed with an approximation that can be accessed from the index structure. The difference between the R-tree and 2DR-tree approaches is the following. Although the R-tree is a one-dimensional structure, it can be applied to data residing in any dimension. However, in its current form, the 2DR-tree represents a two-dimensional topological modeling of the protein structure, since it is designed to work in two-dimensional space.

A protein structure can also be indexed at the atomic level, where atoms and bonds are grouped into leaf nodes. If a 2DR-tree is applied at the atomic level, the bonds between atoms can be implicitly represented by maintaining

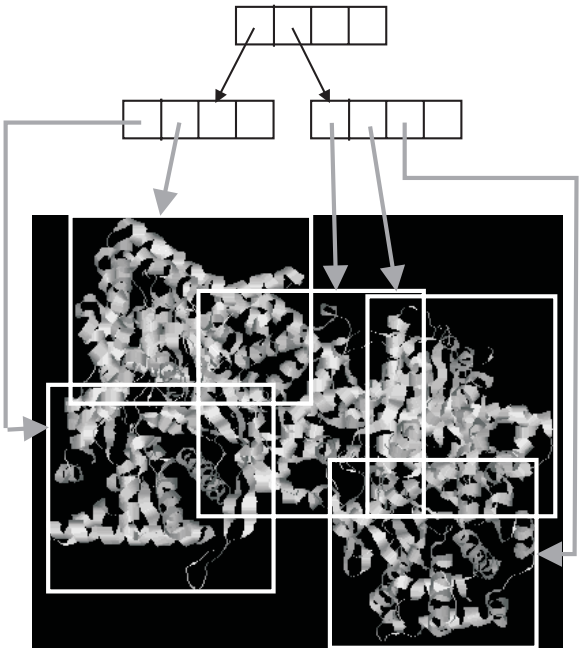


**Figure 4. R-tree Substructure Modeling**

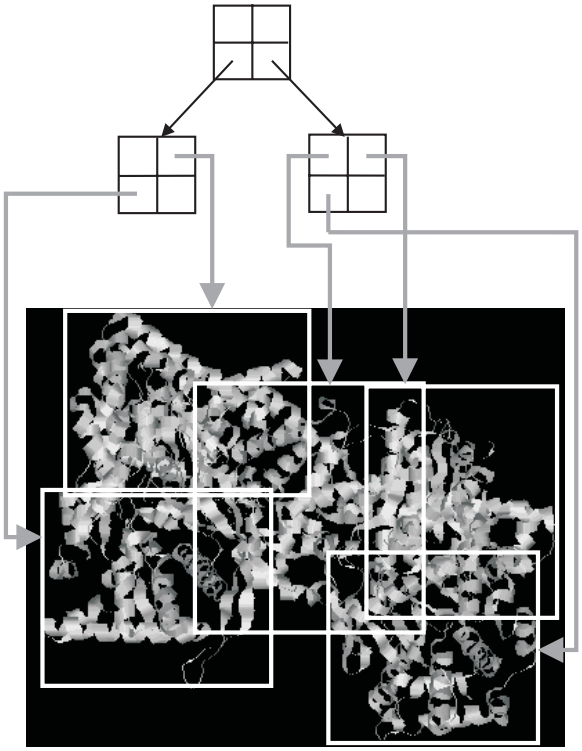


**Figure 5. 2DR-tree Substructure Modeling**

the spatial relationships between the atoms, therefore eliminating the need to store them. A protein structure can also be indexed at the helix/$\beta$-sheet level. In this configuration, leaf nodes can contain helices, $\beta$-sheets, and other known and unknown secondary structures.

In addition, multiple protein secondary structures can be modeled within the same spatial index. This allows for protein structure homology comparisons to be carried out with ease, since potentially related substructures are stored together. Alternatively, secondary structures can be modeled with different spatial indices (but of the same index type) but still compared for structural homology.

### 3.2. Tertiary Modeling

Similarly to secondary modeling, the tertiary representation of a protein structure can be represented at the substructure level and the atomic level. Also, it is possible to apply the same spatial index to multiple tertiary structures for the purposes of structure homology comparison.

### 3.3. Quaternary Modeling

One more option is added to the existing levels of modeling resolution given above. The quaternary description of multiple tertiary structures can be successfully modeled with one spatial index structure. This will provide support for modeling across proteins. Also, if one is interested, it is possible to model multiple quaternary descriptions with one index to provide further opportunities for the analysis of protein structures.

### 3.4. Searching and Analysis Strategies

One can use existing spatial searching strategies, such as the region search, or the binary or greedy search techniques of the 2DR-tree [9, 10], on protein structures. In addition, one can apply other search and analysis techniques as the basis for data retrieval. For example, strategies for structure homology comparision such as VAST [8, 6] can be incorporated into a spatial index model. Structure homology comparisons can be applied not only between existing structures, but also for the prediction of a newly-discovered protein structure based on existing structures.

### 3.5. Inclusion of Non-Spatial Data

Descriptive non-spatial data can be stored alongside an approximation. This comes in handy for storing data from

the primary level of the description hierarchy. A corresponding amino acid subsequence can be stored with the approximation representing its substructure or secondary structure. Information on the number of secondary structures and the $\alpha/\beta$ ratio can be stored with a substructure or an entire protein. Also, one can simply provide a reference the appropriate lower level of the hierarchy.

## 4. Conclusions

This paper investigates the feasibility of applying a spatial index to the problem of modeling protein structures. A preliminary model is proposed, which has the advantage of flexibility in representation and manipulation of structures. Also, additional functionality to facilitate analysis can easily be incorporated. It is expected that this approach will lead to new approaches to the study of protein structures.

Research will continue in the following directions. First, the 2DR-tree can be extended to three dimensions (i.e. 3DR-tree). The 2DR-tree has specific properties, such as its support for maintaining spatial relationships between objects, which make it a desirable option for protein structure modeling. However, in its current form, the 2DR-tree only indexed a topological view of a structure. A similar three-dimensional structure will alleviate this problem.

Second, the co-ordination between a spatial index and strategies for protein structure homology comparison, such as VAST [8, 6] will be investigated further. This will lead to new strategies for comparisions and prediction of new structures based on existing structures.

Third, other strategies for searching, insertion and deletion will be explored. Although most work with protein structures require access only [1], it is believed that by providing a framework to manipulate proteins by adding and removing features such as secondary structure elements, atoms and substructures, future work will lead to exploration of new comparison and prediction techniques.

Finally, strategies for visualizing a protein structure that is modeled using a spatial index will be explored. The images presented in this paper display the spatial index. Currently, the spatial index provides a modeling technique that resides in the lower levels of a protein structure repository, and therefore is not displayed when viewing a protein model. However, visualization of the spatial index structure with the protein structure, including a rotatable view, is also an important direction of future research.

## Acknowledgments

The author wishes to thank the reviewers for their helpful comments and suggestions, in particular the suggestion concerning model visualization.

## References

[1] A. Baxevanis and B. Ouellette. *Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins*. Wiley & Sons, Hoboken, New Jersey, 2005.

[2] H. J. Bernstein. Openrasmol: Molecular graphics visualisation tool. Website, last visited January 2007. `http://www.openrasmol.org/`.

[3] D. Comer. The ubiquitous B-tree. *ACM Computing Surveys*, 11(2):121–137, 1979.

[4] R. C. for Structural Bioinformatics. Protein data bank. Website, last visited January 2007. `http://www.rscb.org/pdf`.

[5] V. Gaede and O. Guenther. Multidimensional access methods. *ACM Computing Surveys*, 30(2):170–231, 1998.

[6] J. Gibrat, T. Madej, and S. Bryant. Surprising similarities in structure comparison. *Current Opinion in Structural Biology*, 6(3):377–85, 1996.

[7] A. Guttman. R-trees: a dynamic index structure for spatial searching. In *Proceedings of SIGMOD'84*, pages 47–57, 1984.

[8] T. Madej, J. Gibrat, and S. Bryant. Threading a database of protein cores. *Proteins*, 23(3):356–69, 1995.

[9] W. Osborn and K. Barker. Searching through spatial relationships using the 2dr-tree. In *Proceedings of the 10th International Conference on Internet and Multimedia Systems and Applications (IMSA 2006)*, pages 71–76, 2006.

[10] W. K. Osborn. *The 2DR-tree: a Two-dimensional Spatial Access Method*. PhD thesis, University of Calgary, June 2005.

[11] S. Shekhar and S. Chawla. *Spatial Databases: A Tour*. Prentice-Hall, New Jersey, 2003.

[12] S. Srinivasa and S. Kumar. A platform based on the multidimensional data model for analysis of bio-molecular structures. In *Proceedings of the 29th International Conference on Very Large Data Bases*, 2003.

[13] X. Yan, P. Yu, and J. Han. Graph indexing based on discriminative frequent structure analysis. *ACM Transactions on Database Systems*, 30(4):960–993, 2005.