Review of Spatiotemporal Data Indexing Structures

K.Venkateswara Rao[1], A.Govardhan[2] and K.V.Chalapati Rao[3]
[1,3]Department of CSE, CVR College of Engineering, Ibrahimpatnam, Hyderabad, India
[1]kvenkat.cse@gmail.com
[3]chalapatiraokv@gmail.com
[2]School of Information Technology, JNTUH Hyderabad, India

govardhan_cse@yahoo.co.in

Abstract: **Spatiotemporal data are those objects that change their position and/or extent over time. There are many applications like geographic information systems, intelligent transportation systems, environmental information systems and mobile communications data management systems which deal with spatiotemporal data. Efficient processing of the spatiotemporal data needs sophisticated indexing schemes. Many indexing structures are available in the literature for accessing such data sets. This paper provides a comprehensive survey, classification and brief description of spatiotemporal data indexing structures for efficient accessing of the data by various types of spatiotemporal queries.**

Index terms:* *Database index, spatiotemporal index, aggregate data index

I. Introduction

A spatiotemporal object can be represented as (NScT, ScT, T) [29] where NScT is the time-stamped non-spatial conventional component, ScT is the time-stamped spatial component and T is the temporal component indicating its valid time. The spatial and non-spatial conventional components can take many values along time line. ScT is a set of records where each record contains a spatial component value at i^{th} time point, Ti. Similarly NScT is a set of records where each record contains non-spatial component value at time point, Ti. All Ti's are contained in T and they do not intersect.

The mathematical model defining spatiotemporal data structure is given below.

SDS = (THC, SC, TC) where
SDS = Spatiotemporal Data Structure
THC = Thematic Component
SC = Spatial Component
TC = Temporal Component
THC = { (A,T) /A = Thematic Attribute Л T = Temporal Attribute }
SC = { (S,T) /S = Spatial Attribute Л T = Temporal Attribute }
TC = { T / T = Temporal Attribute }

Thematic Attributes are non spatial properties of spatial objects whose values may change with time. Spatial Attributes are geometry and location of the spatial objects whose values also may change with time. Temporal Attributes are temporal point and temporal interval.

The aim of indexing spatiotemporal data is to facilitate its retrieval for efficient processing of spatiotemporal queries such as time slice queries e.g., " Find all objects that cross a certain area at time t", window queries or range queries e.g., " Find all objects that cross a certain area in the time interval [t1,t2]", n-nearest neighbor queries e.g., " Find n hospitals that are closest to a given moving point", trajectory queries e.g., " Find trajectory of a given point for the past one hour". More research is done in developing spatiotemporal data indexing and access methods to support spatiotemporal queries. The spatiotemporal access methods [1,2] with underlying spatial and temporal structures are grouped into different categories as follows.

- Indexing the past data
- Indexing the current data
- Indexing the future data
- Indexing data at all points of time
- Indexing aggregate historical spatiotemporal data
- Miscellaneous indexing structures

This paper is organized as follows. Section 1 presents definition of spatiotemporal object and classification of spatiotemporal data indexing structures. Review of the literature relevant to Indexing the past data is given in Section 2. Indexing the current spatiotemporal data related work are reviewed in Section 3. Section 4 deals with indexing structures for the future spatiotemporal data. Section 5 describes an Indexing Past, Present and Future Positions of the spatiotemporal data. Indexing structures for aggregate historical spatiotemporal data are reviewed in Section 6. Other miscellaneous indexing structures for spatiotemporal data are provides in section 7. Finally section 8 provides conclusion

II. Indexing The Past

The spatiotemporal indexing methods for historical data are grouped into three categories.

- First one is dealing with temporal dimension.
- Second one is dealing with overlapping and Multi-version structures
- Third category is dealing with Trajectory oriented access methods.

2.1 Indexing Structures dealing with Temporal Dimension

In this category, the main focus is to deal with the spatial domain. The temporal dimension is handled by augmenting the temporal aspect into existing spatial access methods. RT-tree, 3DR-tree, STR-tree, MTSB-tree, FNR-tree and MON-tree fall into this category. RT-tree [1] combines R-tree [3] as spatial access method and Time-Split B-tree (TSB-tree) [1] as a method for temporal access. It indexes entire spatiotemporal data in a single R-tree [3]. It is very difficult to manage RT-tree when more number of objects change and interval queries span entire tree. 3DR-tree [1,4,5,6] handles interval queries efficiently by treating time as third dimension but its performance is poor on timestamp queries. STR-tree [1,7] is a variant of R-tree. R-tree is extended with a different algorithm for inserting and splitting. The Multiple TSB (MTSB) tree [2,8] uses a TSB-tree for supporting spatial range and historical queries on a database of moving objects. Objects moving in a set of connected and fixed line segments in a two-dimensional space are indexed by Fixed Network R-tree (FNR-tree) [2,9]. FNR-tree is extended by MON-tree [2,10]. The constrained network is modeled in MON-tree as a set of junctions and routes that are non-intersecting poly-lines.

2.2 Overlapping and Multi-version Structures

In this category, the temporal aspects are separated from spatial aspects so that entire spatial data that is valid at one time instant is kept together and managed in one structure. Excessive storage is required by this approach. Consecutive instances are combined into a single structure using overlapping in access methods to avoid storage of identical sub-structures. MR-tree, HR-tree, HR+-tree, MV3R-tree, PA-tree and GS-tree belong to this category. Both MR-tree [1] and Historical R-tree (HR-tree) [4,5,11] create a separate R-tree to every timestamp. This allows sharing the branches of consecutive trees to avoid storage overhead. Both these structures are efficient for timestamp queries but their performance on interval queries is poor. HR+-tree [1,11] is developed to avoid the replication problem in the HR-tree. Multi-version 3DR-tree (MV3R-tree) [1,4] mainly depends on multi-version B-tree (MVB-tree) [4]. It combines a multi-version R-tree (MVR-tree) [4] and a 3DR-tree so that timestamp queries are processed using MVR-tree [4,12] and interval queries are processed using 3DR-tree [4,5,6]. Partially persistent R-tree (PPR-tree) [1,11,13] structure is designed for bi-temporal databases. It maintains R-tree evolution so that any historical query at time point t is able to use R-tree structure corresponding to time t. The trajectories of moving objects are indexed using Parametric tree (PA-tree) [2,14], that indexes by splitting time duration of the moving object into n disjoint time intervals and also dividing trajectory of the object into n line segments corresponding to the respective time intervals. Graph Strip tree (GS-tree) [2,15] indexes historical position data of moving objects using a constrained graph that is defined by vertices and edges as in MON-tree.

2.3 Trajectory-Oriented Access Methods

This category of access techniques focuses on trajectory-oriented queries on historical spatiotemporal data. TB-tree, SETI, SEB-tree, CSE-tree, Polar-tree and RTR-tree fall into this category. Trajectory-bundle tree (TB-tree) [1,16] structure is similar to R-tree and is an extension of STR-tree [7]. A leaf node of a TB-tree stores line segments of same trajectory. The drawback is that line segments belonging to the trajectories of spatially co-located moving objects are maintained in different nodes of TB-tree. Scalable and Efficient Trajectory Index (SETI) [1] creates static, non-overlapping partitions for spatial dimension. The trajectory segments within each partition are indexed using R-tree. Start/End timestamp B-tree (SEB-tree) [1] divides the space into zones. It indexes each zone. The zones may overlap. A hash function that is based on both timestamps of a moving object is used to hash the object into its zone. Compressive Start-End Tree (CSE-tree) [2,17] divides the space into disjoint cells and maintains a time for each spatial cell. It is used in GPS data sharing applications. Polar-tree [2] is used to index orientations of moving objects to a given focal point to detect the objects which get closer or move away from the focal point. RTR-tree[2,18] is based on R-tree. It is used to index trajectories of objects which are moving in symbolic indoor space. The RFID readers are used to gather position data of the moving objects. HBSTR-Tree [19] is a hybrid index structure comprising spatio-temporal R-tree2, B-tree and hash table. It is used for trajectory databases.

III. INDEXING THE CURRENT POSITIONS

The idea of the "current" positions [1] is challenging in database systems. Spatiotemporal index methods that support queries involving current positions data are 2+3 R-tree, 2-3 TR-tree, LUR-tree, LUGrid, RUM-tree and IMORSS. The 2+3 R-tree [1] maintains two R-trees. First one is used to index two dimensional current positions of the moving object. Second one is used to index the trajectories in spatial and temporal dimensions to maintain the history of the moving objects. The trajectories are built for updated current objects in three dimensions. Then the objects are inserted into the second R-tree and deleted from the first R-tree. The 2-3 TR-tree [1] is similar to the 2+3 R-tree except that its three dimensional R-tree does not maintain the trajectories. It maintains only multi-dimensional points. It uses the underlying structure of TB-tree to answer trajectory-oriented queries. The Lazy Update R-tree (LUR-tree) [1] is only concerned with the current locations of the objects. It stores no historical data. The Lazy-Update Grid-based (LUGrid) [2,20] index uses grid file for lazy insertion and deletion. The positions of moving objects are updated frequently through the grid file. The current object identifiers are tracked in LUGrid using in-memory data structures. The R-tree with Update Memo (RUM-tree) [2,21] manages frequent updates to locations of moving objects by inserting the object in the new position without touching its old location. The obsolete entries in the RUM-tree are removed by lazy garbage cleaner. Indexing Moving Objects on Road Sectors (IMORS) [2,22] is used to index current locations of moving objects on a fixed route network. R*-tree [3] is used in IMORS to index road sectors.

IV. Indexing The Future Positions

It is required to have extra data like destination and speed of a moving object to predict its future locations in a variety of ways such as the original space-time continuum, transformation methods and parametric spatial access methods.

4.1 The Original Space-time Continuum

The moving object in one dimensional space can be modeled using linear equation $p_t = vt + c$ where v is the constant velocity, c is a constant for the starting position of a moving object and p_t is the predicted location at time t. The predicted locations of moving objects can be computed and represented in the original spatiotemporal space. PMR-quadtree [23] and MOVIES [24] follow this approach. PMR-quadtree [23] for moving objects indexing destroys whole index structure and rebuilds it again when an update to moving objects occurs. In this way, it only maintains current PMR-quadtree. Moving objects indexing using frequent snapshots (MOVIES) [24] uses linearized kd-trees [3] to index locations of moving objects for supporting predictive queries on the moving objects.

4.2 Transformation Methods

These methods transform original time-space domain into another space representation to ease the representation and querying of the data in future. Duality transformation [1,25], SV-Model, PSI, STRIPES, Bx-tree, By-tree, ST2B-tree, Bdual-tree and BdH-tree are transformation methods based indexing techniques [2,1]. Duality transformation [1,25] represents the equation $p_t = mt + n$ as a point (m, n) in a dual two dimensional space in which horizontal dimension is the velocity m and vertical dimension is the reference location n. KD-tree [3] based spatial index is used instead of an R-tree in the dual space due to highly skewed nature of the distribution. SV-model [1] represents a moving object using four parameters - starting position, starting timestamp, initial velocity and destination. The horizontal dimension in the dual space representation is starting time and the vertical dimension is the destination. It assumes starting position and Velocity as constants. The dual space is indexed by SS-tree. PSI [1,26] approach uses an R-tree to index three dimensional space for modeling trajectories. The three dimensions correspond to reference location, velocity and time. STRIPES [2] models positions of a moving object using a linear function in n-dimensional space and predicted locations of the moving object in n-dimensional space are transformed into points in a two dimensional dual space. It maintains two different indexes using PR-quadtree. These two indexes correspond to first and second half of time intervals of the moving object. Bx-tree [2] is an extension of the B+-tree for indexing moving objects. By-tree [2] is an extension of the Bx-tree. It uses different update frequencies for each moving object. ST2B-tree [2] indexes the future positions of moving objects in a manner similar to Bx-tree. It improves Bx-tree by handling data skewness in both spatial and temporal dimensions. Bdual-tree [2] represents both d-dimensional positions and velocities in a dual two dimensional space. It is built on Bx-tree. BdH-tree [2] builds on Blink-tree instead on Bx-tree. The internal nodes of Blink-tree at the same level are linked.

The transformation methods have some drawbacks. The entire data in the primal space may not be captured in the dual space. It is not guaranteed that objects which are near in original space continue to be near in dual space.

4.3 Parametric *Spatial Access Methods*

Parametric bounding rectangles are used in these methods to index the original time-space. These rectangles are represented as functions of time in such a way that the enclosed moving object in the rectangle remains to be in the same rectangle. TPR-tree, PR-tree, VCI R-tree, STAR-tree, R^{EXP}-tree, TPR*-tree, STP-tree and ANR-tree fall into this category. Time Parameterized R-tree (TPR-tree) [1,27] constructs conservative bounding rectangles in R-tree. They enclose a set of moving points. PR-tree [1,28] is similar to TPR-tree except that it represents the moving object as spatial extents. Velocity Constrained Indexing R-tree (VCI R-tree) [1] imposes a restriction on maximum speed of moving objects. It is capable of addressing the problems in continuous queries execution. Spatio-Temporal Self Adjusting R-tree (STAR-tree) [1] is same as TPR-tree except that it is able to self adjust without user input when the query performance degrades. R^{EXP}-tree [1] has extended TPR-tree to use expiration times to manage moving objects. The expired entries are removed from the index using a lazy technique. TPR*-tree [1] is same as TPR-tree except that it uses insert and deletion algorithms of R*-tree. TPR-tree and TPR*-trees are generalized in Spatio-Temporal Prediction Tree (STP-tree) [2]. Future locations of moving objects are indexed using arbitrary polynomial type moving functions in STP-tree. Adaptive Network R-tree (ANR-tree) [2,29] is used to index future trajectories of objects moving in a constrained network.

V. Indexing Past, Present And Future Positions

R^{PPF}-Tree, $PCFI^{+}$ - index, BBx-index, UTR-tree, $STCB^{+}$-tree and PPFI index structures fall into this group. R^{PPF}-tree [2,30] indexes locations of moving objects at all timestamps. The past positions of a moving object are captured in R^{PPF}–tree by the application of partial persistence to the TPR-tree. The future locations are calculated using interpolation of the past locations of a moving object using a linear function. The time-parameterized bounding rectangle (TPBR) in R^{PPF}–tree is used for current time. The past-current-future-index (PCFI) [2,30,31] is built using TPR*-tree and SETI. Moving object space is split into non-overlapping cells in PCFI. Present locations and velocities of a moving object are indexed using TPR*-tree and is maintained for each cell. PCFI indexes historical data using a sparse on-disk R*-tree. BBx-index [30,32] uses Bx-tree to support the present and future. Uncertain Trajectory R-tree (UTR-tree) [2] is an extension of the MON-tree with uncertainty that belongs to a constrained network. Spatio-Temporal Compressed B+ - tree (STCB+ - tree) [2] indexes trajectories of moving spatiotemporal objects using multiple compressed B+-trees. PPFI [2] indexes road network using 2D R*-tree. It uses hash structure to predict near-future locations and to describe the recent states of moving objects.

VI. Indexing Aggregate Historical Spatiotemporal Data

Many real-life applications need to access summarized spatiotemporal data. Pre-aggregation of raw data is required in these applications due to involvement of huge amount of spatiotemporal data organized in multidimensional data structure. Both spatial and temporal dimensions can be associated with dimension hierarchies. Queries require aggregation of the data along the dimension hierarchies to satisfy some of the query conditions. Various indexing structures [33,34] for spatiotemporal aggregate data are aR-tree, a3DR-tree, aRB-tree, aHRB-tree and a3DRB-tree. The original R-tree is extended in aggregate R-tree (aR-Tree) [33,34]. Each intermediate node in R-tree is augmented with storage of aggregated data for sub-branch under the node. aR-tree is generalized to three dimensional space in aggregate 3DR-tree (a3DR-tree) [33,34]. Both temporal and spatial dimensions can be indexed simultaneously in a3DR-tree. The drawbacks of a3DR-tree are its huge size that impacts query performance and wasting storage area by storing MBR for each change in aggregate data. The problems in a3DR-tree are rectified in aggregate RB-tree (aRB-tree) [33,34] by using R-tree to index aggregate spatial data and creation of a B-tree to each entry of the R-tree for storing historical aggregate data. If finest granularity in spatial dimension is considered, the regions become volatile. An approach to handle volatile regions is the creation of a new R-tree for each change. The aggregate Historical RB-tree (aHRB-tree) [33,34] is a combination of aRB-trees and HR-trees. The valid period of a spatiotemporal object during its lifespan is indicated by each node in HR-tree. A node is duplicated in HR-tree for each change in its entries. This causes redundancy of data in aHRB-tree. The aggregate 3DRB-tree (a3DRB-tree)[34] eliminates the data redundancy problem in aHRB-tree by combining 3DR-trees and B-trees.

VII. Miscellaneous Indexing Structures

Other indexing structures are Po-tree [35,36] and AIRSTD [35]. Po-tree is used to index real-time spatiotemporal data. It uses Kd-tree to index spatial aspects and B+ - tree to index the temporal aspects of the data. AIRSTD is an acronym for "Approach for Indexing and Retrieving Spatio-Temporal Data". It is a combination of 'Change Temporal Index' (CTI) and R-tree. CTI is used for temporal data indexing and R-tree is used for indexing spatial aspects.

VIII. Conclusions

All the spatiotemporal data indexing structures for accessing spatiotemporal data are collected and reviewed in this paper. They are organized into different categories and their purpose, focus and basis on which they are built is described. Indexing structures suitable for timestamp queries, range queries, trajectory queries and aggregate queries are specified. Standard bench mark datasets for evaluation and comparision of various spatiotemporal indexing structures are need of the hour to establish future direction for further research.

References

[1]. Mohamed F. Mokbel, Thanaa M. Ghanem and Walid G. Aref " Spatio-temporal Access Methods", IEEE Data Engg, Bull., 26(2):40-49,2003.

[2]. Long-Van Nguyen-Dinh, Walid G. Arif and Mohamed F. Mokbel " Spatio-Temporal Access Methods: Part 2(2003-2010). IEEE Data Engineering, Bulletin:

[3]. Peter Kuba, " Data Structures for Spatial Data Mining", FIMU-RS-200-05, September 2001.

[4]. Yufei Tao and Dimitris Papadias, " The MV3R-Tree: A Spatio-Temporal Access Method for Timestamp and Interval Queris", Proceeding of the 27th VLDB conference, Roma, Italy, 2001.

[5]. Lei Chen, Vincent Oria and M. Tamar Ozsu, " A Multi-level Index Structure for Video Databases", Proceedings of Internaltional Workshop on Multimdia Transformation Systems, October 2002,USA, PP:28-37.

[6]. Yannis Theodoridis, Michael Vazirgiannis and Timos Sellis, " Spatio-Temporal Indexing for Large Multimedia Applications", Proceedings of the Third Internaltional Workshop on Multimdia Transformation Systems, June 1996, PP:441-448

[7]. D. Pfoser, C. S. Jensen, and Y. Theodoridis. Novel Approaches in Query Processing for Moving Object Trajectories. In *Proc. of the Intl. Conf. on Very Large Data Bases, VLDB*, pages 395–406, Sept. 2000

[8]. P. Zhou, D. Zhang, B. Salzberg, G. Cooperman, and G. Kollios. Close pair queries in moving object databases. In *GIS*, pages 2–11, 2005.

[9]. E. Frentzos. Indexing objects moving on fixed networks. In *SSTD*, pages 289–305, 2003.

[10]. V. T. De Almeida and R. H. G¨uting. Indexing the trajectories of moving objects in networks. *Geoinformatica*, 9(1):33–60, 2005.

[11]. Y. Tao and D. Papadias. Efficient Historical R-trees. In *Proc. of the Intl. Conf. on Scientific and Statistical Database Management, SSDBM*, pages 223–232, July 2001.

[12]. Marios Hadjieleftheriou, George Kollios, Vassilis J. Tsotras and Dimitrios Gunopulos, " Indexing Spatio-temporal Archives", The VLDB Journal, Vol -….

[13]. Marios Hadjieleftheriou, George Kollios, Dimitrios Gunopulos and Vassilis J. Tsotras, "Efficient Indexing of Spatiotemporal Objects", Proceedings of the 8th International Conference on Extending Database Technology (EDBT), Springer-Verilog London, UK 2002, PP:251-268.

[14]. J. Ni and C. V. Ravishankar. PA-tree: A parametric indexing scheme for spatio-temporal trajectories. In *SSTD*, 2005.

[15]. T. T. T. Le and B. G. Nickerson. Efficient search of moving objects on a planar graph. In *GIS*, pages 1–4, 2008.

[16]. D. Pfoser, C. S. Jensen, and Y. Theodoridis. Novel Approaches in Query Processing for Moving Object Trajectories. In *Proc. of the Intl. Conf. on Very Large Data Bases, VLDB, pages 395-406, Sept. 2000*

[17]. L. Wang, Y. Zheng, X. Xie, and W.-Y. Ma. A flexible spatio-temporal indexing scheme for large-scale GPS track retrieval. In *MDM*, pages 1–8, 2008.

[18]. C. Jensen, H. Lu, and B. Yang. Indexing the trajectories of moving objects in symbolic indoor space. In *SSTD*, 2009.

[19]. Shengnan Ke, Jun Gong, Songnian, Qingzhu, Xintao Liu and Yeting Zhang, "A Hybrid Spatiotemporal Indexing Method for Tracjectory Databases", Sensors, July 2014.

[20]. X. Xiong, M. Mokbel, and W. G. Aref. LUGrid: Update-tolerant grid-based indexing for moving objects. In *MDM*, page 13, 2006.

[21]. Y. N. Silva, X. Xiong, and W. G. Aref. The RUM-tree: Supporting frequent updates in R-trees using memos. *VLDB J.*, 18(3):719–738, 2009.

[22]. K.-S. Kim, S.-W. Kim, T.-W. Kim, and K.-J. Li. Fast indexing and updating method for moving objects on road networks. *Web Information Systems Engineering Workshops*, 0:34–42, 2003.

[23]. J. Tayeb, O¨ . Ulusoy, and O.Wolfson. A Quadtree-BasedDynamic Attribute IndexingMethod. *The Computer Journal*, 41(3):185–200, 1998.

[24]. J. Dittrich, L. Blunschi, and M. A. Vaz Salles. Indexing moving objects using short-lived throwaway indexes. In *SSTD*, pages 189–207, 2009.

[25]. G. Kollios, D. Gunopulos, and V. J. Tsotras. On Indexing Mobile Objects. In *Proc. of the ACM Symp. on Principles of Database Systems, PODS*, pages 261–272, June 1999.

[26]. . Porkaew, I. Lazaridis, and S. Mehrotra. Querying Mobile Objects in Spatio-Temporal Databases. In *Proc. of the Intl. Symp. on Advances in Spatial and Temporal Databases, SSTD*, pages 59–78, Redondo Beach, CA, July 2001.

[27]. S. Saltenis, C. S. Jensen, S. T. Leutenegger, and M. A. Lopez. Indexing the Positions of Continuously Moving Objects. In *Proc. of the ACM Intl. Conf. on Management of Data, SIGMOD*, pages 331–342, May 2000.

[28]. M. Cai and P. Revesz. Parametric R-Tree: An Index Structure for Moving Objects. In *Proc. of the Intl. Conf. on Management of Data, COMAD*, Dec. 2000.

[29]. J.-D. Chen and X.-F. Meng. Indexing future trajectories of moving objects in a constrained network. *J. Comput. Sci. Technol.*, 22(2):245–251, 2007.

[30]. K Appathurai and S. Karthikeyan "A Survey on Spatio-Temporal Access Methods" , International Journal of Computer Applications (0975 – 8887), Vol-18, No-4, March 2011.

[31]. Z.-H. Liu, X.-L. Liu, J.-W. Ge, and H.-Y. Bae. Indexing large moving objects from past to future with PCFI+-index. In *COMAD*, pages 131–137, 2005.

[32]. D. Lin, C. Jensen, B. Ooi, and S. ˇ Saltenis. Efficient indexing of the historical, present, and future positions of moving objects. In *MDM*, pages 59–66, 2005.

[33]. Dimitris Papadias, Yufei Tao, Panos Kalnis and Jun Zhang, " Indexing Spatio-Temporal Data Warehoses",Proceedings of 18th International Conference on Data Engineering, San Jose, March 2002, PP:166-175.

[34]. Dimitris Papadias, Yufei Tao, Jun Zhang, Nikos Mamoulis, Qiongmao Shen and Jimeng Sun, " Indexing and Retrieving of Historical Aggregate Information about Moving Objects", Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 2002.

[35]. Hatem F. Halaoui, " AIRSTD: An Approach for Indexing and retrieving Spatio-Temporal Data", SITIS 2006.

[36]. Guillaume Noel, Sylvie Servigne and Robert Laurini, " Real-Time Spatiotemporal Data Indexing Structure", 7th AGILE Conference on Geographic Information Sciences, 29 april – 1 may, 2004, Greece.