Visualizing Learning Behavior as Spatio-Temporal Trajectories

Kevin Fuchs and Peter A. Henning[1]

Abstract: The digitalization of teaching and learning has become an increasing desire for schools and universities. In order to apply digital media purposefully, educational organizations need to understand if and how students make use of digital contents and platforms. In the following we present a technique that uses arbitrary logging data as they may be present in any ICT systems that are commonly used to distribute digital learning contents. It transforms arbitrary data into spatio-temporal trajectories that can be analyzed only on the basis of their geometric relationships and characteristics. Through this we lift heterogeneous data to a highly abstract level. In an example, we illustrate how we can distinguish different types of users regarding temporal patterns and the learners' mobility. We are also able to recognize groups of students working on similar topics. We mostly understand the current state of our system as a tool that can give both researchers and teachers the possibility to examine student's behavior on a qualitative basis. In an outlook we furthermore describe how the system can be extended to support automatic clustering of learning behaviors.

Keywords: learning analytics, spatio-temporal database, e-learning, distant learning, computers in education, data mining

1 Introduction

Progress in the field of ICT has brought new opportunities for the delivery of information and the construction of knowledge. Especially the worldwide web enables us to spread multimedia content for mobile and distant learning. Consequently, in the sector of education desires have grown to integrate digital media into the curricula. To utilize digital media efficiently, purposefully and economically, educational organizations need to address the habits and behaviors of their students. Schools and universities operate diverse platforms to provide digital learning content. But often they do this in a kind of blind flight not knowing how their students make use of those offers. On the other hand schools and universities lack time and resources to run broad empirical studies on these questions.The respective platforms may provide logging and statistical functionality. However, they cannot be utilized without expensive preparation and often respective data has to be aggregated from different heterogeneous sources.

Basically, we need a method for data abstraction allowing us to join heterogeneous data sources and reduce complexity. In this paper we introduce the Hypercube Database – in

[1] Institute for Computers in Education, Karlsruhe University of Applied Science, Germany

doi:10.18420/in2017_171

the following abbriviated as HCDB, which is based on a spatio-temporal database with which we model students' behavior and measurements of influencing factors in the form of spatio-temporal trajectories. HCDB is based on a data model that describes any of n variables equally as a dimension in a n-dimensional space. No particular scale level is needed. Each individual is assigned a n-dimensional time-evolving vector drawing a n-dimensional trajectory in that space. Because all elements of that vector are treated equally, HCDB is indifferent of what each dimension represents. A vector element may be a variable indicating a student's position within a learning environment – for example a course that a student has visited, a file she has downloaded or a video he has watched. It may be a score a student has achieved on a specific item or it may represent an influencing factor like the student's location, the device she is using or even stress measurement data.

All this data is abstracted to n-dimensional hyper-polylines which are purely geometrical objects. This reduces complexity in so far as we can analyze these trajectories only with respect to their geometrical characteristics as well as their spatio-temporal similarities and differences. HCDB focuses especially on the time dimension, considering time a dominant factor for learning processes. Learning and teaching always induce the transformation of semantically linked content into a linear sequence of actions. Therefore, we expect time coherences to give us an insight into many aspects of learning. In the following we explain how we can use HCDB for a qualitative form of learning analytics. We will show how we can identify common patterns of behavior by abstracting arbitrary data to spatio-temporal structures. Researchers may use their findings from HCDB as a preliminary stage for further quantitative and empirical analysis. In a concluding section we will further give an outlook on how we intend to enhance our system such that an automatic clustering of learning histories can be performed.

2 The Hypercube Model

The data structure that is used by HCDB origins from the former INTUITEL project (Intelligent Tutoring Interface for Technology enhanced Learning) [He14b]. It was a research project funded by the European Commission including twelve European partners. The INTUITEL group developed a universal interface to enhance common learning management systems such that they work as adaptive learning environments. A key element of INTUITEL is its Hypercube Model which describes the learner's cognitive position within a learning environment as a hyper-space vector [FHM16]. INTUITEL uses this vector to determine the learner's cognitive distance with respect to predefined learning pathways in order to generate learning recommendations for the learner. The Hypercube Model describes a n-dimensional space with n being the number of knowledge objects (KO) in a learning environment. Knowledge Objects may represent atomic items in a learning management system like text items, exercises, test modules or video items. Knowledge Objects may also be logical elements containing other KOs. This may be courses or folders in a learning management system. For HCDB we enhanced this model by adding arbitrary dimensions for other data items. We first elaborate on the Hypercube Model as it is used by INTUITEL and then explain how we enhanced it for HCDB.

Each of the n Hypercube dimensions is assigned a numeric value representing the state of progress that a learner has performed on the according KO at a certain point in time expressed by a value from the interval $[0, 1]$. A learner's position in this space is a time-evolving vector $L = (l_1, ..., l_n)$ drawing a trajectory in the n-dimensional space. This model can be envisioned as a n-dimensional hypercube with each edge representing the interval $[0, 1]$. This is also known as the "fuzzy hypercube" [Ko90]. The trajectory of any vector L is located within that hypercube. Figure 1 illustrates a hypercube with $n = 4$. Each edge represents a knowledge object a learner may attend to. The colored arrows describe exemplary transitions of a learner between KOs. In case the learner finishes one KO by the other in a disciplined way the learner's behavior results in a movement along the edges of the hypercube. In case the learner switches between KOs and/or does not process them completely, her movements result in a trajectory in the inner space of the hypercube. For didactic reasons a teacher may consider recommended learning pathways within that hyperspace. The INTUITEL system is capable of generating learning recommendations based on the distance between a learner's previous, personal learning pathway and the recommended one [He14a].

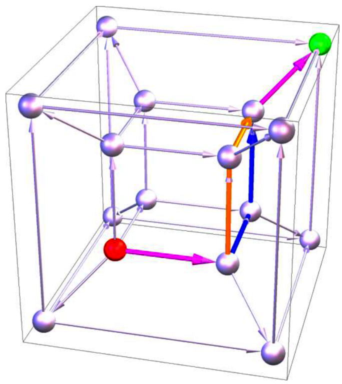

Fig. 1: 4D-hypercube for four KOs, coloured arrows describing possible learning pathways

3 The Advanced Hypercube Model

The initial purpose of the Hypercube Model was the representation of learning pathways and the according generation of learning recommendations. For HCDB we enhanced this model by considering not only a learner's progress on KOs but also any kind of data that can be of interest as far as learning analytics is concerned. The Hypercube Model is enhanced by k additional dimensions. This describes a $(n + k)$-dimensional space with n being the number of KOs in a learning environment and k being the number of additional data items. Each of these k dimensions may be assigned an arbitrary numeric value at a particular point in time. A learner's position in this space is now a time-dependent vector $M = (m_1, ..., m_n, m_{n+1} ... m_{n+k})$ forming a trajectory in the $(n + k)$-dimensional space. In case the k dimensions are normalized – which is not necessarily required – this enhanced model too converges to the fuzzy hypercube [Ko90]. Without normalization the hypercube simply becomes a hypercuboid. Learners' movements through this $(n + k)$-dimensional space define trajectories which we modeled with a spatio-temporal database as we will describe in the remainder of this paper.

4 History of Spatio-Temporal Databases

Before developing the design of HCDB, we evaluated existing research in the field of spatio-temporal databases (STDB). Actually, our spatio-temporal trajectories are high-dimensional objects that cannot be handled by common STDBs. Most STDBs are designed for real-world objects with two or three spatial dimensions. Purely temporal databases are for example the ARCADIA database for clinical applications [Co95], Calanda for time series with financial data [Sc95], ChronoLog running on top of a standard Oracle database [Bo94], HDBMS [Sa87], TDBMS [Ta97] and TimeDB for general purpose which is based on the ATSQL2 query language [St98, CRS06a, CRS06b].

The field of spatio-temporal databases is mostly dominated by Geographical Information Systems (GIS), Network and Facility Management, Land Information Systems (LIS) and Image Processing [AR99]. For example GRASS GIS [Ne12] and GeoToolKit [Ba00] are Geographical Information Systems while the CONCERT database focuses on management of raster images [RSS98, Re97]. The SECONDO database is a multi-purpose system for spatio-temporal data [AGB06]. Due to the nature of their subject these systems mostly provide support for only two or three spatial dimensions. The DEDALE database is capable of dealing with higher dimensions and is based on a constraint database technique that describes spatio-temporal objects as point sets defined by logical constraints [Ri03, Gr97].

5 Implementation of the Hypercube Database

All databases dealing only with two or three spatial dimensions are not an option for HCDB due to its high-dimensional space. The aforementioned DEDALE system appears to be an interesting candidate because of its constraint approach that can be exploited for high numbers of dimensions [GSR98]. However, the constraint database approach is most appropriate for querying geometric objects containing infinite point sets whereas it is less suitable for querying continuous trajectories. We therefore developed our own database but we used the temporal database TimeDB as its back end building the spatio-temporal functionality upon it. Figure 2 shows the fundamental parts of the system which is subdivided into the Vector Module, the Hypercube Module and the Database Access Module.

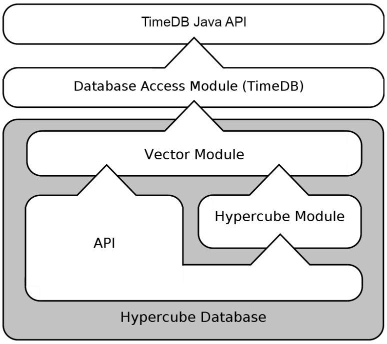

Fig. 2: Architecture of the Hypercube Database

5.1 Database Access Module

The system is built in Java and uses the TimeDB Java library which is managed by the ***Database Access Module***. TimeDB itself provides temporal support only for database tuples but not attribute-wise. The Database Access Module built upon TimeDB provides an interface for the Vector Module with which temporal support for single attributes is achieved.

5.2 Vector Module

The task of the ***Vector Module*** is the transformation of single measuring points into temporal vectors and storing it via the Database Accessor Module. Consider an individual for which we want to measure the values of m variables over time. For each variable we measure values at arbitrary points in time. For each variable we regard the lastly measured value as valid until a new value is measured. This way, we get a m-dimensional time-dependent vector for each individual. The listing below describes the transformation of measuring points into vector representation.We illustrate the subsequent insertion of measuring points and the evolution of the vectors for a trajectory with three variables a_1, a_2 and a_3.

At the beginning all variables have an initial value e.g 0 from start to eternity.

$$a_1 = 0 \; for \; t \; \in [0, forever)$$
$$a_2 = 0 \; for \; t \; \in [0, forever)$$
$$a_3 = 0 \; for \; t \; \in [0, forever)$$

Now we insert the measuring points $a_2 = 3$ at the time point t_1, $a_2 = 5$ at t_2 and $a_1 = 7$ at t_3

$$a_1 = \begin{cases} 0 & for\ t \in [0, t_3) \\ 7 & for\ t \in [t_3, forever) \end{cases}$$

$$a_2 = \begin{cases} 0 & for\ t \in [0, t_1) \\ 3 & for\ t \in [t_1, t_2) \\ 5 & for\ t \in [t_2, forever) \end{cases}$$

$$a_3 = 0\ for\ t \in [0, forever)$$

5.3 Hypercube Module

The ***Hypercube Module*** finally is responsible for three tasks as described below:

Interpolation: By default, the Vector Module represents its vector elements as period-wise constant values. Additionally, single dimensions of a trajectory can be interpolated by a linear function between each pair of consecutive measuring points.

Trajectory Clustering: To identify similar learning histories, the Hypercube Module is supposed to implement algorithms for trajectory clustering. This implies the calculation of similarity measures to be used for generic cluster algorithms.

Spatio-Temporal Indexing: For trajectory clustering we have to implement algorithms to examine trajectories with respect to spatio-temporal nearness and characteristics. For this purpose, similarity metrics have to be calculated and we need to provide efficient search algorithms. As these algorithms operate in a high-dimensional search space they have to be based on spatio-temporal indexing techniques.

5.4 API

The API provides access to the functionality of the Hypercube Database firstly in the form of Java methods, secondly in the form of a REST interface that can be accessed over a network. Both access methods provide functionality for the insertion and deletion of measuring points as well as for querying over trajectories. The Java API provides an additional interface to address the TimeDB back end directly with ATSQL queries. Moreover, trajectories can be visualized as we elaborate in the next section. It is possible to use the REST interface for real-time recording of data. For example a learning management system may address the REST interface asynchronously and transmit data about learners' actions.

5.5 State of Implementation

The Database Accessor Module and the Vector Module have been fully implemented. Equally, all parts of the API that are related to these two modules are complete. By the date of this publication the Hypercube Module is still in progress. This concerns the Indexing and the Clustering part. There is good reason why the Hypercube Module can not be implemented without careful research in the run-up and why we decide to publish the current state of this project. First there is the challenge of spatio-temporal indexing. Other spatio-temporal databases have implemented such indexing techniques. However, most of these solutions are not appropriate for the Hypercube Database due to its high-dimensional space. Furthermore, we intend to implement cluster algorithms to identify similar learning behaviors. For this, we first have to clarify which similarity metrics and cluster algorithms we should consider. We elaborate on both indexing techniques and cluster algorithms in the concluding *Future Work* section. As far as similarity metrics are concerned we decided to take a preliminary step visualizing the data as it is held by the Vector Module. We use this visualization to examine learners' behaviors visually. By this we get an orientation of possibly interesting patterns from which we may derive similarity metrics.

6 Visualization of Students' Behavior

We performed two tests of the Hypercube Database by recording logging data in two different learning platforms.

6.1 Students' Behavior on a Video Streaming Platform

The first platform was a video streaming site for lecture recordings. This platform contained 55 different video files and 200 users and provided logging data over a period of four semesters. The logging data was extracted and processed from the internal log files of the platform and it resulted in the following data items for each user:

system: operating system of the user (Windows, Mac OS, Linux, Android, etc.)
subnet ip: IP subnet of the user corresponding to the subnet mask 255.0.0.0
video title: title of the video file watched
seen percentage: percentage of minutes a user has watched of a specific video.

The above listed data contained 10 distinct operating systems, 33 different subnets and 55 video titles. The most obvious strategy for the conversion of this data into the Hypercube Model would conform to the following data model: 10 dimensions for the operating system with each dimension swapping between the values 0 and 1 (in use or not in use); 33 dimensions for the subnets having the value 0 or 1 (in use or not in use); 55 dimensions for the video titles with a value from 0 to 100 (seen percentage).

This would result in a 98-dimensional space. In fact our goal is to implement clustering algorithms that are able to handle such high dimensions. However, our intermediate purpose is to analyze the data visually just in order to obtain indications on the kind of similarities that occur in such data and how we can detect them algorithmically. The problem arising is not necessarily the high dimensions. It is rather the fact that the single dimensions would only be sparsely occupied. This is because a single video title is only retrieved occasionally and for a relatively short time compared to the overall period of four semesters. So if we visualize this data, it would be difficult for the human eye to recognize any patterns. We therefore modified the representation as follows: The Hypercube Model requires numeric values which means that string values have to be transformed into numbers. However, this is just a formal prerequisite. We decided to simply enumerate over the distinct string values for each data item. The result were the following four dimensions:

system: an integer between 1 and 10 (10 distinct operating systems)
subnet ip: an integer number between 1 and 33 (33 distinct subnets)
presentation title: an integer from 1 to 55 (55 distinct video titles)
seen percentage: an integer between 0 and 100

This way we reduced the dimensionality of the Hypercube Model massively by turning coordinate axises into values and merging them into shared dimensions. By this transformation we obtain four-dimensional time-evolving vectors for each student. We visualized each dimension of the vector as a time line which represents its values in the form of a color map. In this example each operating system is represented by one in 10 different colors, each subnet IP by one in 33 colors and each video title by one in 55 colors. The seen-percentage value is represented by a color scale that equates to the spectrum of light running from purple (0%) to red (100%). Additionally, for each student a normalized histogram is shown that indicates her frequency of activity. This enables us to categorize students' behavior by the following criteria:

regular users repeatedly used the video service during periods of several months. Figure 3 visualizes three representative individuals of that group.
occasional users watched the lecture recordings only occasionally. A visualization example of three occasional learners is illustrated by figure 4 .
dense users were inactive during longer time periods but used the video platform intensively during short time periods which indicates that they used the lecture recordings for specific purposes like the preparation for an exam. See figure 5 where the outlined areas represent a time period of 10 days.
mobile users were indicated by a fluctuation of the "system" and "subnetip" variable. Mobile learners were mostly present among the group of regular users. A visualization of three representative mobile users is shown in figure 6.
stationary users rarely or never changed their location or system. For example some dense users showed stationary behavior (see the third dense user in figure 5). One reason might be that when students prepare for exams they often do this in a ritualized way to discipline

themselves which involves learning in a particular place like in a library or at home. A visualization of three representative stationary users is shown in figure 7.

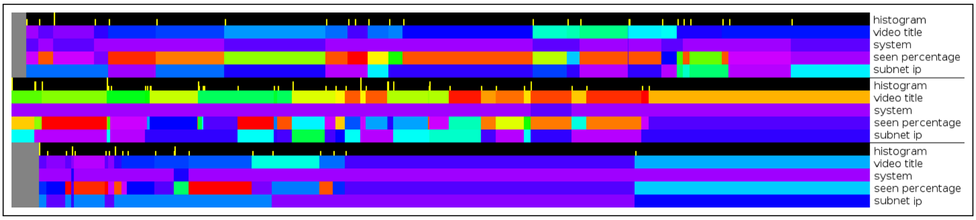

Fig. 3: regular users corresponding to a time period of 6 months (gray means no data is available)

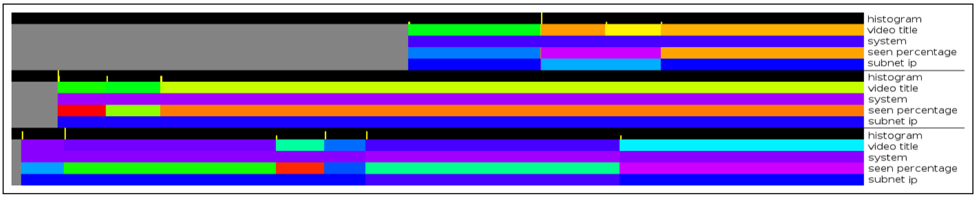

Fig. 4: occasional users corresponding to a time period of 6 months (gray means no data is available)

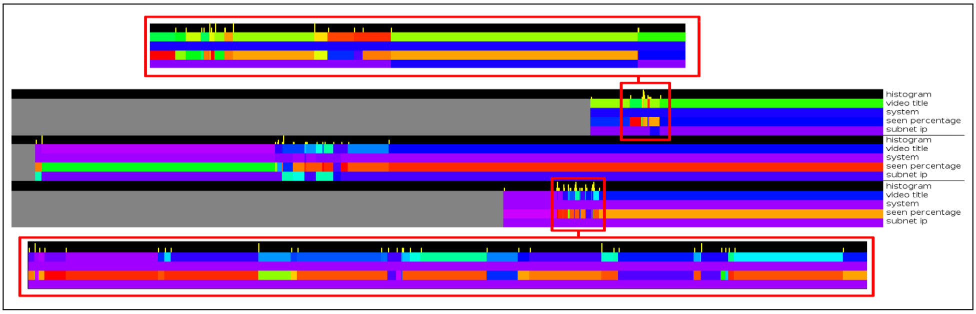

Fig. 5: dense users corresponding to a time period of 5 months (gray means no data is available)

Note that the HCDB holds a value for a variable until it is updated by a new event. This means that if a student watches for example 25% of a video and if he is inactive for the following two weeks, the visualization will show the respective color for a period of two weeks. How data is kept depends on the way that updates are inserted.

6.2 Spatial Discrimination to Identify Student Groups

In another example we were able to identify students that cooperated working on a set of exercises. It is only a small set of data and therefore we only use it to demonstrate how in principle spatial discrimination can be used to identify student groups. 15 students had to edit 47 articles in a Semantic MediaWiki application. The articles were grouped by topics and the students were distributed among these topics. The MediaWiki Software provided a logging function to record when a student viewed or edited an article. For the transfer

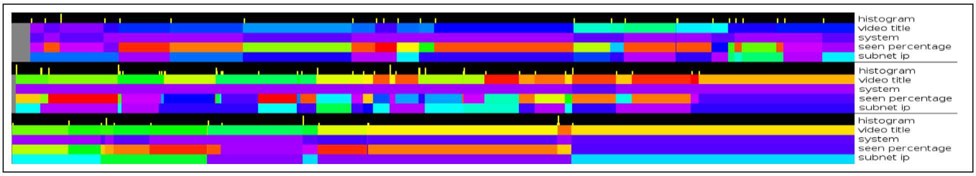

Fig. 6: mobile users corresponding to a time period of 4 months (gray means no data is available)

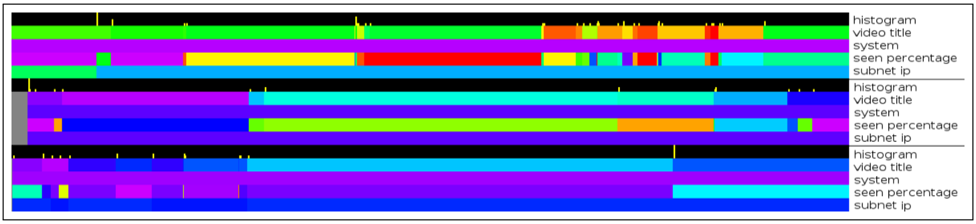

Fig. 7: stationary users corresponding to a time period of 4 months (gray means no data is available)

of the logging data we enumerated the articles. Each topic group was given a continuous interval. By this the visualization of each topic group corresponded to a subarea in the color spectrum. In terms of the Hypercube Model this equates to subspaces in one dimension. If a combination of m attributes was partitioned this way the result would be a set of m-dimensional subspaces. This method of grouping allows us to use spatial discrimination to identify phenomena that are correlated to such groups.

In this very simple example we can clearly identify students and their topic groups as it is illustrated by the example shown in figure 8. Each trajectory represents a single student. The first row of each trajectory indicates that the student has viewed an article, the second row recorded articles that were edited. At a glance we can distinguish three topic groups: bluish (trajectory 1 and 2), greenish (trajectory 3–5) and orange-colored (trajectory 6–8). Trajectory 6 represents a student that quit the course very early and therefore shows no more activity. Trajectory 2 and 8 represent students that joined the course later. Again, note that HCDB stores a value until it is updated by a new measuring point. If a student edits an article and if he is inactive for, lets say, the following two weeks, the visualization will show the color corresponding to the edited article for a period of two weeks.

7 Future Work: Spatio-Temporal Indexing and Clustering

Our first task for future work is the implementation of spatio-temporal indexing. Many of existing indexing techniques are based on the R-Tree family [Gu84, BM12] for multidimensional spatial indexing. [GS05] lists e.g. the 3D R-tree, the HR-tree, the RT-tree and the MR-tree. Moreover – for indexing moving objects with respect to the current time and the near future – [GS05] refers to TPR-trees, multilevel partition trees, kinetic B-trees and

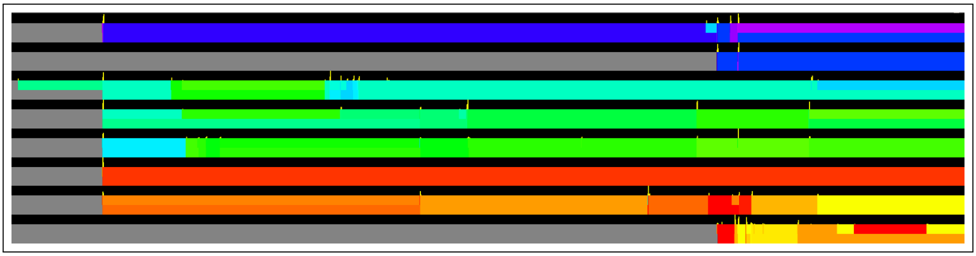

Fig. 8: stationary users corresponding to a time period of 3 months (gray means no data is available)

kinetic external range trees. The usefulness of those indexing techniques strongly depends on the kind of data and the kind of queries to be performed. In the case of the HCDB we are less interested in querying point sets like ”select the geographic region that was covered by the storm between 5 am and 7pm“. Such a query would be useful within a Geographical Information System and would return a point set as a geometric object altering over time. But in our case we are mostly interested in queries referring to entire trajectories like ”select all trajectories near to trajectory x“. For indexing and querying trajectories there are two techniques available: The Spatio Temporal R-tree (STR-tree) and the Trajectory Bundle Tree (TB-tree). Both index structures are appropriate for performing point, range and nearest-neighbor queries as well as trajectory-based queries [GS05].

However, tree-based index structures tend to perform poorly with high-dimensional spaces. This is because such search trees partition their search space by bounding boxes. The more dimensions we have, the more these bounding boxes tend to overlap. So to find a specific item, more sub-branches of the search tree have to be accessed. The X-tree is based on the R-tree and better optimized for multiple dimensions [BKK96]. However, our subject is spatio-temporal trajectories and not point sets. Trajectories own some particular characteristics we can utilize for efficient indexing: trajectories grow in a monotone way along the time dimension. Once a trajectory segment is inserted it is not altered or deleted and trajectory segments are usually inserted chronologically meaning that an index structure does not have to be reorganized for past insertions. Therefore, a good alternative is a grid-based index structure as it was implemented by SETI (Scalable and Efficient Trajectory Index) [CEP03]. SETI uses a two-level approach which treats the spatial and the temporal indexing separately. A grid index partitions the space by static, non-overlapping spatial static cells. The dimension of these cells corresponds to the dimension of the space. Such a grid index is especially interesting for two reasons: First — as temporal indexing is treated separately from spatial indexing — we can keep the temporal indexing mechanism of our TimeDB back end. Second the developers of SETI show that their technique ”does not suffer from the curse of dimensionality“ [CEP03].

Our second future task aims at automated clustering of trajectories. By the date of this publication it is not yet clear to the authors which cluster algorithms are most appropriate. We first have to identify useful similarity metrics. The visualization of data — as elaborated in this paper — provides a qualitative insight into a variety of phenomena and we expect this to give us orientation about possible metrics.

To give an impression of the various issues in the field of trajectory clustering we briefly introduce other researchers' work in this area. Byoung-Kee Yi et al. introduced a method to discover similar patterns in time sequences [YJF98]. They especially considered time warping techniques to investigate similarities that occur in different time sequences at different time periods. Nanni and Pedreschi worked on density clustering focusing on the time dimension [NP06]. Their approach is based on the OPTICS algorithm (Ordering Points To Identify the Clustering Structure) [An99] which is an enhanced version of the DBSCAN algorithm (Density-Based Spatial Clustering of Applications with Noise) [Es96]. Furthermore, Jae-Gil Lee et al. addressed the problem of similar sub-trajectories instead of treating trajectories only as a whole [LHW07].

8 Conclusion

It must be outlined that the visualization technique we described here can be regarded as a special representation of time series data that could also be achieved with more simple methods. It does not yet justify the enormous expense of a spatio-temporal database implementation. However, the use of spatio-temporal database technologies obtains its justification in case of large data sets and the need of high processing speed. The capability of spatio-temporal indexing and the provision of efficient search algorithms provides a unique solution in a special use case of learning analytics that we name “real-time pathway prediction”. Our long-term goal is to implement a system that is able to model the previous and partial learning pathway of a learner and compare it to other learners' pathways in the past. This would facilitate learning pathway recommender systems whose recommendations are based on learning behaviors of other learners for example on MOOC platforms. If this analysis is supposed to be processed in real-time and even on large data sets – as it is the case in a MOOC scenario – it becomes clear that this can only be achieved by the use of special indexing and search techniques.

The findings we presented and the system we built actually describe an intermediate state of the Hypercube Database. Our long-term objective is the automatic discovery of similar learning behaviors. For this we have to implement spatio-temporal indexing and clustering techniques. Concerning this we have to find metrics that provide information about similarities and dissimilarities of trajectories in the multidimensional space of the Hypercube Model. For this purpose and to get an insight into the data kept by the Hypercube Database, we implemented the visualization technique as described before.

In our example we only described categories of learners like ”regular“, ”occasional“, ”dense“, ”mobile“ and ”stationary“ but without quantifying their occurrences. For quantification we would first need to define objective measures for what a ”regular“, ”occasional“, ”dense“, ”mobile“ and ”stationary“ learner is. However, the current state of the system and its visualization capabilities already allow qualitative analysis. Abstracting heterogeneous data to spatio-temporal structures detaches our perspective from single variables. Instead we can consider only differences, similarities and temporal changes to discover patterns and characteristics. Moreover, we sketched the facility of grouping data into value ranges for the purpose of spatial discrimination. This allows us to uncover correlations between individuals. In this light we consider the current state of the Hypercube Database as a useful method for teachers and course administrators to examine students' behavior. Furthermore, we regard it as a research tool to provide preliminary qualitative findings that can be used as a basis for empiric research.

References

[AGB06] Güting, R. H.; Almeida, V.T. de; Behr, T.: Querying Moving Objects in SECONDO. 7th Int. Conf. on Mobile Data Management, 47-51, 2006

[An99] Ankerst, M. et al.: ordering points to identify the clustering structure. Proc. ACM SIGMOD Int. Conf. on Management of data, 1999

[AR99] Abraham T., Roddik, J. F.: Survey of Spatio-Temporal Databases. GeoInformatica vol 3, no. 1, 61–99, 1999

[Ba00] Balovnev, O. et al.: Extending GeoToolKit to Access Distributed Spatial Data and Operations. Proc. 12th Int. Conf. on Scientific and Statistical Database Management, 2000

[BKK96] Berchtold, S., Keim, D. A.; Kriegel, H.: The X-Tree : An Index Structure for High-Dimensional Data. Proc. 22nd Int. Conf. on Very Large Data-Bases, 1996

[BM12] Balasubramanian, L.; Sugumaran, M.: A State-of-Art in R-Tree Variants for Spatial Indexing. Int. Journal of ComputerApplications, vol. 42 no.20 975–8887, 2012

[Bo94] Böhlen, M.:Managing Temporal Knowledge in Deductive Databases. Dissertation, Federal Institute of Technology Zurich, 1994

[CEP03] Chakka, V. P.; Everspaugh, A.C.; Patel, J.M.: Indexing large trajectory data sets with SETI. Proc. Conf. on Innovative Data Systems Research, 2003

[Co95] Combi ,C. et al.: Querying temporal clinical databases with different time granularities: the GCH-OSQL language. Proc. Annual Symposium on Computer Applications in Medical Care, 326–330, 1995

[CRS06a] Carvalho, A.; Ribeiro, C.; Sousa:, A.: A Spatio-temporal Database System Based on TimeDB and Oracle Spatial. Research and Practical Issues of Enterprise Information Systems, vol. 205, 11-20, 2006

[CRS06b] Carvalho, A.; Ribeiro, C.; Sousa:, A.: Spatial TimeDB - Valid Time Support in Spatial DBMS. Proc. 2nd Int. Advanced Database Conf., 2006

[Es96] Ester, M.: A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining, 1996

[FHM16] Fuchs, K.; Henning, P.A.: Hartmann, M.: INTUITEL and the Hypercube Model - Developing Adaptive Learning Environments. Journal of Systemics, Cybernetics and Informatics, 2016

[GS05] Güting, R. H.; Schneider, M.: Moving Objects Databases. Morgan Kaufmann Publishers, 2005

[GSR98] Grumbach, S.; Segoufin, L.; Rigaux, P.: Efficient multi-dimensional data handling in constraint databases. BDA, 1998

[Gr97] Grumbach, S. et al.: DEDALE, A Spatial Constraint Database. DBPL, 38-59, 1997

[Gu84] Guttmann, A.: R-trees: a dynamic index structure for spatial searching. Proc. ACM SIGMOD Int. Conf. on Management of data, 47–57, 1984

[He14a] Henning, P.A. et al.: Learning Pathway Recommendation based on a Pedagogical Ontology and its Implementation in Moodle. Proc. DeLFI 2014 Conf., GI Lecture Notes in Informatics, 2014

[He14b] Henning, P.A. et al.: INTUITEL – Intelligent Tutorial Interface for Technology Enhanced Learning. Proc 22nd UMAP Conf., 2014

[Ko90] Kosko, B.: Fuzziness vs. Probability. Int. Journal of General Systems, vol. 17, 211-240, 1990.

[LHW07] Lee, J-G; Han, J.; Whang, K-Y.: Trajectory clustering: a partition-and-group framework. Proc. ACM SIGMOD Int. Conf. on Management of data, 2007

[Ne12] Neteler M. et al.: GRASS GIS: A multi-purpose open source GIS. Environmental Modelling & Software 31, 124–130, 2012

[NP06] Nanni M.; Pedreschi, D.: Time-focused clustering of trajectories of moving objects. Journal of Intelligent Information Systems vo.l 27, no. 3, 267-289 2006

[Re97] Relly, L.et al.: Physical Database Design for Raster Images in Concert. 5th Int. Symposium on Spatial Databases, 1997

[Ri03] Rigaux, P. et al.: Building a constraint-based spatial database system: model, languages, and implementation. Inf. Syst. 28(6): 563-595, 2003

[RSS98] Relly, L.; Schuldt, H.; Schek, H.: Exporting Database Functionality - The CONCERT Way. IEEE Data Eng. Bull 01/1998; 21:43-51, 1998

[Sa87] Sarda, N. L. : Design of an Information System Using a Historical Database Management System. Proc. 8th. Annual Int. Conf. on Information Systems, 86-96, 1987

[Sc95] Schmidt, D. et al: Calanda - a Complete Solution for Time Series Management in Banking, UBS, Zurich, 1995

[St98] Steiner, A.: A Generalisation Approach to Temporal Data Models and their Implementations. Dissertation, Swiss Federal Institute of Technology Zurich, 1998

[Ta97] Tansel, A. U.: Temporal Relational Data Model. IEEE Transactions on Knowledge and Data Engineering, vol. 9 no. 3, 1997

[YJF98] Yi, B-K.; Jagadish, H.V.; Faloutsos C.: Efficient retrieval of similar time sequences under time warping. Proc. 14th Int. Conf. on Data Engineering, 1998