A temporal GIS for field based environmental modeling

Sören Gebbert[a], Edzer Pebesma[b]

[a] *Thünen Institute of Climate-Smart Agriculture, Bundesallee 50, D-38116 Braunschweig, Germany*
[b] *Institute for Geoinformatics, University of Muenster, Weseler Strasse 253, 48151 Muenster, Germany*

Abstract

Time in geographic information systems has been a research theme for more than two decades, resulting in comprehensive theoretical work, many research prototypes and several working solutions. However, none of the available solutions provides the ability to manage, analyze, process and visualize large environmental spatio-temporal datasets and the investigation and assessment of temporal relationships between them. We present in this paper a freely available field based temporal GIS (TGRASS) that fulfills these requirements. Our approach is based on the integration of time in the open source Geographic Resources Analysis Support System (GRASS). We introduce the concept of a space time dataset that is defined as a collection of time stamped raster, voxel or vector data. A dedicated set of spatio-temporal tools was implemented to manage, process and analyze space time datasets and their temporal and spatial relationships. We demonstrate the temporal GIS and environmental modeling capabilities of TGRASS by analyzing a multi-decadal European climate dataset.

Keywords:
Temporal GIS, spatio-temporal modeling, GRASS GIS

1. Introduction

The integration of time in geographic information systems has been an ongoing research theme for 25 years. The book *Time in Geographic Information Systems* (Langran, 1992) marked a first milestone in temporal GIS. Since then, the literature about temporal GIS concepts and related topics like spatio-temporal database models has grown rapidly. Comprehensive overviews about spatio-temporal database models and temporal GIS approaches are available in Pelekis et al. (2004), O'Sullivan (2005), Ott and Swiaczny (2001).

The temporal GIS approach for environmental modeling that we present in this study focuses on the field based view on geographical data. We follow in our approach the field definition of Galton (2001): A spatial field is a mapping from spatial locations to values that may be any kind of data structures. The field provides a coverage of space using irreducible minimal regions, for example represented as a pixel. A mapping that assigns a value to each location at each time is called a spatio-temporal field. The distinction between an object based and field based view of geographic information is an important concept in geoinformation science. A good overview about this distinction with comprehensive theoretical work is given in Galton (2001, 2004), Goodchild and Gopal (1989), Goodchild (1992).

Field based temporal GIS has been a key technology for integrated assessment modeling that is common in the climate change research (Christakos et al., 2001). With the availability of high resolution environmental data sets with continental to global extent containing continuous field measurements and output data from physical, chemical or statistical models, a strong need has emerged to efficiently manage, analyze, process and visualize such big data.

Several spatio-temporal environmental modeling software systems and temporal GIS solutions are available that include STempo (Peuquet and Hardisty, 2010), GeoViz Toolkit (Hardisty, 2013), PCRaster (PCRaster team, 2012), TerraME (de Senna Carneiro et al., 2013), the R environment for statistical computing (R) (R Development Core Team, 2012), TerraLib (Câmara et al., 2008), Climate Data Operators (CDO) (Schulzweida, 2013), Arc Hydro Groundwater (Aquaveo LLC, 2013) and STEMgis (Discovery Software Ltd., 2013). Most of these solutions have a dedicated purpose that is spatio-temporal visualization, statistical analysis, water management, raster time series processing or climate data analysis. Yuan (2009) stated that most temporal GIS technology developed are still in the research phase or have an emphasis on mapping. Exceptions are the TerraLib and the R environment. Because of its modular approach the R environment can be enhanced with several spatial, temporal and spatio-temporal packages. For example the *spacetime* package (Pebesma, 2012) in conjunction with packages *sp, xts, rgeos* and *raster* transform R into a feature rich spatio-temporal GIS environment with modeling, statistical analysis and visualization capabilities.

However, to process massive datasets that do not fit into the main memory[1], R still requires a spatio-temporal database backend and is therefore not well suited yet for large-scale field based spatio-temporal modeling. The main aim of the TerraLib class and functions library is to enable the development of new generation GIS applications. TerraLib implements the basic infrastructure for spatio-temporal analysis and modeling and supports several different spatio-temporal data types (events, mobile objects, and evolving regions) (Câmara et al., 2008). On top of TerraLib, TerraME de Senna Carneiro et al. (2013) offers the capability to model nature-society interactions, using multi-scale concepts.

None of the available solutions provide large-scale field based, spatio-temporal environmental modeling capabilities that are based on a comprehensive set of spatio-temporal GIS management, processing and analysis tools. With the exception of the R environment available solutions do not support the analysis of relationships between spatio-temporal fields that are used in environmental modeling.

The aim of this paper is to describe a field based temporal GIS, based on the Geographic Resources Analysis Support System (GRASS), to efficiently manage, visualize, process, model and analyze large spatio-temporal fields and their spatio-temporal relationships. An additional aim is the interoperability between our temporal GIS (TGRASS) and the spatio-temporal modeling and analyzing environments R and CDO as well as ParaView (Kitware Inc., 2013a). The management, analysis, modeling, processing and visualization capabilities of our approach are demonstrated by analyzing a large climate dataset provided by the European Climate Assessment and Dataset project (ECA&D).

2. Related work

The temporal GIS approach presented in this paper follows the field based world view using two and three spatial and one temporal dimension as defined in Galton (2004). A comprehensive field based temporal GIS must support different kind of fields that are common in environmental modeling. Common spatial fields may have two or three dimensions. Such spatial fields are regular gridded, irregular gridded or of object type. A continuous field that maps

[1]With the exception of the *raster* package

location to spatial objects is defined as an Object field (Cova et al., 2002). Object fields are introduced in Galton (2001) and formulated more generally in Cova et al. (2002). Regular gridded fields can be represented as raster (2D) or voxel (3D) data. Irregular gridded fields can be represented as two or three dimensional point clouds, triangulated irregular networks (TIN) or Voronoi diagrams. These kind of fields are a specific form of object fields since they are built upon vector features like points, lines and polygons. Object fields may also contain spatial objects representing for example a watershed or a viewshed.

In our study the spatio-temporal fields are organized using time stamped spatial fields. This is commonly known as a snapshot approach. It has been utilized in several temporal GIS implementations because of its simplicity and the ability to extend existing spatial GIS that are layer based. Following the snapshot approach to integrate time in a spatial GIS, time stamps are assigned to spatial fields. Hence all cells (2D or 3D) or objects in a spatial field share the same time stamp. We will use the term *snapshot* and *time stamped spatial field* interchangeably in our paper. The concept of space time datasets was introduced to efficiently manage time stamped spatial fields. Space time datasets represent spatio-temporal fields in TGRASS. They are defined as a collection of time stamped spatial fields (snapshots) from which they derive their spatial and temporal extent. The common snapshot approach was extended in TGRASS so that each time stamped spatial field can have a different spatial and temporal extent. Temporal as well as spatial relationship computation between time stamped spatial fields is supported to allow the investigation of spatio-temporal interactions between them.

Space time cubes were introduced with two spatial (x, y) and one temporal (t) dimension. Space time cubes are often utilized to analyze and visualize space time paths resulting from the movement of individuals or objects in space and time. The space time cubes in TGRASS represent spatio-temporal fields, build upon three dimensional pixels (voxels). Forer (1998) denoted these kind of voxels as taxels to emphasize the specific nature of the time dimension. We denote this three dimensional spatio-temporal field representation as space time voxel cube. It can be seen as a special case of a space time dataset with restricted properties. The benefit of space time voxel cubes is the availability of several tools in GRASS that can perform spatio-temporal operations on them, for instance spatio-temporal map calculation as defined in Jeremy et al. (2005). Mitasova et al. (2011) utilized the voxel analysis capabilities of GRASS to analyze time series data.

2.1. Time in space time datasets

Several different models of time in geographic information systems have been developed. Models of time can be linear or cyclic, discrete or continuous, supporting branching or multiple perspectives. A comprehensive overview about different times in GIS is given in Frank (1998).

A field based temporal GIS must represent how fields are measured in time. Temperature for example is measured at time instances but the mean temperature is computed for time intervals. Precipitation and Greenhouse Gas (GHG) emissions are measured in time intervals. The system must be aware of calendar time to manage and analyze interaction between measured fields. Environmental models may use time for simulation with no fixed reference, hence relative time must be supported.

TGRASS uses the concept of linear, discrete time represented by time instances and time intervals. Time intervals and time instances represent the time stamps of spatial fields. The interval time model supports the occurrence of gaps between intervals. Time intervals are allowed to overlap or contain each other and can contain time instances. Time intervals can be unequally spaced. Time is measured using the Gregorian calendar time, also called absolute time, conform to ISO 8601[2] and as relative time defined by an integer and a unit of type year, month, day, hour, minute or second. The smallest supported temporal granule is a second. The definition of absolute and relative time follows the temporal database concepts collected in Dyreson et al. (1994).

Time intervals in our approach are designed to easily detect gaps. Intervals consist of a start time instance and an end time instance. The end time is not part of the time interval and represents the start time of a potential successor. Hence the time interval is a left closed right open interval. In case the end time of an interval is the start time of a second interval no gaps exist between them. Space time voxel cubes support only non-overlapping time intervals.

2.1.1. Temporal granularity

An important concept in temporal databases is the temporal granularity. A glossary about temporal granularity is available in Bettini et al. (1998).

[2] `http://en.wikipedia.org/wiki/ISO_8601`

The temporal granularity of a space time dataset is defined in TGRASS as the largest common divider granule of time intervals and gaps between intervals or instances from all time stamped spatial fields that are collected in a space time dataset. It is represented as a number of seconds, minutes, hours, days, weeks, months or years. The temporal granularity is computed automatically for each space time dataset.

2.1.2. Temporal topology

The temporal topology describes temporal relations between time stamps represented by interval time or time instances. Several algorithms in TGRASS need to check the temporal topology of space time datasets for validity. The computation of the temporal topology of a space time dataset is based on temporal logic introduced in Allen (1983) shown in Figure 1. A valid temporal topology allows only the following temporal relationships: follows/precedes and after/before.

	A in relation to B	B in relation to A
A B	equivalent	equivalent
A B	follows/adjacent	precedes/adjacent
A B	overlaps	overlapped
A B	after	before
A B	during	contains
A B	starts	started
A B	finishes	finished

Figure 1: Temporal relations between time intervals (Allen, 1983)

2.1.3. Temporal sampling

The investigation of spatio-temporal relations between space time datasets is a core concept in our temporal GIS approach. Relations between space time datasets are based on temporal relationships between time stamped spatial fields. To compute temporal relationships a simplified approach based on Allen (1983) is used, shown in Figure 2. We have chosen a different naming scheme to aggregate several temporal relations into single ones:

- *start* includes *equivalent*, *during*, *starts*, *started*, *finishes* and *overlaps*. This relationship can occur when the start time of a time interval or a time instance is located in a second time interval.

- *overlap* includes *overlaps* and *overlapped*.
- *contain* includes *contains*, *started* and *finished*.
- *during* includes *during*, *starts* and *finishes*.

In TGRASS we denote the process to identify temporally related spatial fields of two space time datasets as temporal sampling. The sampling methods are described in Figure 3.

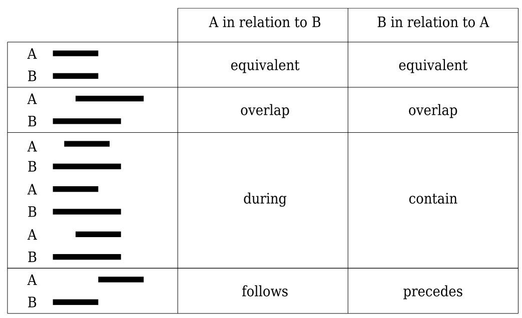
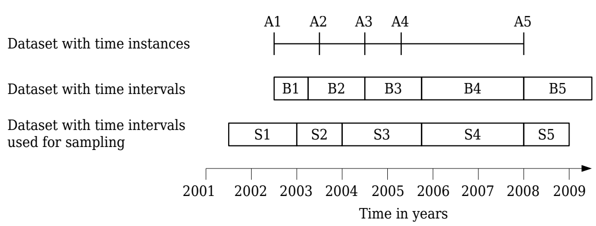


Figure 2: Simplified temporal relationship scheme for space time dataset sampling.



Sampling time instances

	start
S1	A1
S2	A2
S3	A3,A4
S4	—
S5	A5

Sampling time intervals

	start	during	contain	overlap	equal
S1	B1	—	—	B1	—
S2	B2	—	—	B1,B2	—
S3	B3	B3	—	B2	—
S4	B4	—	—	—	B4
S5	B5	—	B5	—	—

Figure 3: Methods of space time dataset sampling. Visualized are the temporal relations from time stamped spatial fields A and B to S. Hence A1 **start**s in S1, B3 is **during** S3, B4 **equal**s S4, B1 **overlap**s S1.

2.1.4. States, Events and Processes

Multiple concepts of states, events and processes have been developed. In our approach a single snapshot represents a specific state of a part of the world at a discrete time instance or in a time interval. Langran (1992) stated that changes between states are defined as events. An event transforms one state into the next, hence change can not be represented using a snapshot approach. The Event-Based Spatiotemporal Data Model (ESTDM) (Peuquet and Duan, 1995) uses time as an organizational basis to store event based changes. Sparse raster structures are used to represent the differences to a base raster layer that defines the state at the beginning of the time series. Worboys (1998) made the distinction that field based approaches allow the definition of processes and the object based approach allows the definition of events. Yuan (2001) claims that processes are a sequence of dynamically related states and events are the occurrence of something significant such as a flood, a storm events or a wildfire. Events may consists of multiple processes and processes may be part of different events.

TGRASS supports the detection of events in spatio-temporal fields. Temporally related snapshots can be identified and differences between them can be computed. TGRASS does not use a specific sparse data structure to store differences between spatial fields. The computational effort to detect differences is much higher than in the ESTDM approach and the data storage is not as efficient. However, TGRASS supports in contrast to the ESTDM approach the computation of differences and the storage of 2D and 3D gridded spatial fields and fields of spatial objects. With the introduction of space time datasets in TGRASS we are able to efficiently manage and analyze processes that were defined by Yuan (2001).

2.2. GRASS GIS

The spatial GIS to integrate time must support spatial fields that are used in environmental modeling. In addition spatial querying, analysis and processing tools must be available. The design of the spatial GIS must provide a well documented Application Programming Interface (API) to enable a strong[3] integration of time. Reusing existing spatial analysis tools and algorithms in spatio-temporal work flows must be supported to avoid redundant

[3] Strong in the meaning that functionality of the temporal GIS framework can be integrated in the core functionality of the chosen GIS

implementations.

We have chosen the Geographic Resources Analysis Support System GRASS for time integration. GRASS is an Open Source Geographical Information System that supports all needed spatial and integration specific requirements. Neteler et al. (2012) stated:

> Due to the scientific background of many of its contributors, and its historical background, GRASS is well equipped for environmental modeling, and at the same time it retains the usefulness for a multi-purpose GIS environment.

In addition to the GRASS GIS website[4] the text book by Neteler and Mitasova (2008) provides detailed information about this open source GIS. GRASS GIS has been utilized in many spatio-temporal environmental scientific applications (Mitasova et al., 1995, 2011, Neteler, 2005, 2010, Zorer et al., 2011). A comprehensive overview about GRASS GIS and its application in environmental modeling is available in Neteler et al. (2012).

3. The integration of time in GRASS GIS

3.1. Implementation

According to (Langran, 1992, page 5) the fundamental functions of a temporal GIS are:

- *Inventory:* Storing a complete description of the study area, and account for changes in both the physical world and computer storage.
- *Analysis:* Explain, exploit, or forecast the components contained by the process at work in a region.
- *Update:* Superseding outdated information with current information.
- *Quality control:* Evaluate whether new data are logically consistent with previous versions and states.
- *Scheduling:* Identifying or anticipating threshold database states, which trigger predefined system responses.

[4] `http://grass.osgeo.org`

- *Display:* Generating a static or dynamic map, or a tabular summary, of temporal processes at work in region.

Except for scheduling, all the requirements of a temporal GIS specified above were implemented with a focus on inventory, analysis and quality. Besides of the environmental modeling requirements, design rules to integrate time in GRASS GIS were considered. An important integration aspect was to avoid the break of existing functionality. To avoid redundancy existing modules and libraries were reused for spatio-temporal field processing. Our implementation follows the GRASS GIS design rule *Create small and fast modules for a specific purpose and combine them to manage complex tasks*.

A single spatial field is usually denoted as a layer in common GIS. However, we will use the GRASS GIS specific notation *raster map*, *3D raster map* and *vector map* in this paper. Two and three dimensional regular gridded fields are referred as raster and 3D raster maps. Irregular gridded spatial fields and object fields are referred as vector maps.

The integration of time in GRASS GIS was based on the combination of a new dedicated Python library that implements the temporal API, the definition of the temporal database structure using SQL statements, and a set of new Python modules, see Figure 4. The resulting temporal geographical information system is called TGRASS in our paper to distinguish between the temporal and non-temporal version of GRASS GIS. We will use the notation *TGRASS* and *our temporal GIS* interchangeably in our paper.

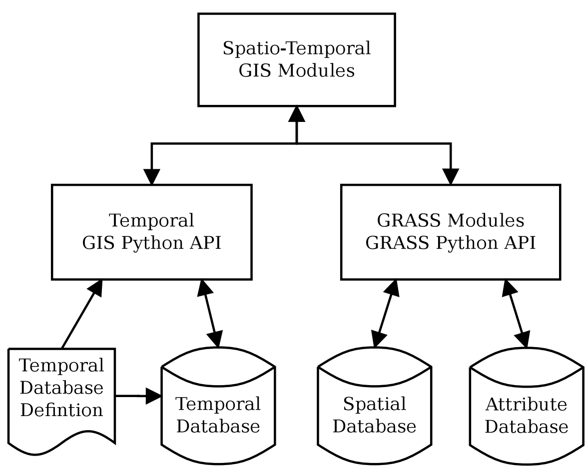


Figure 4: TGRASS API and modules

To assure spatial database compatibility of TGRASS with existing GRASS databases, a dual storage concept was implemented. The spatial and attribute data storage concept of GRASS was not modified. All spatial data is stored in the GRASS spatial database using the existing GRASS specific storage format. Vector attributes are stored in SQL databases. In addition a dedicated SQL database (temporal database) was introduced in TGRASS to store only temporal GIS related metadata.

3.1.1. Adding time stamps to maps

The first step to implement our temporal GIS was to integrate time stamp support for raster, 3D raster and vector maps and therefore the design of the temporal database. The existing time stamp mechanism for raster and 3D raster maps was reused and extended to support vector maps. In TGRASS, maps[5] can be registered and unregistered in the temporal database. When a map is registered, its unique id, the spatio-temporal extent and map type specific metadata are stored in the temporal database. This concept leads to redundant storage, since this data is stored in the GRASS spatial database as well. The benefit of this storage scheme is that it allows complex SQL queries using the spatio-temporal extent and metadata information for map selection. It was not considered to choose a non-redundant storage scheme, since that would require a rewrite of the GRASS core library functionality.

3.1.2. Space Time Datasets

The introduction of three map type specific spatio-temporal data types in TGRASS was the second integration step:

- Space Time Raster Datasets (STRDS) represent collections of time stamped raster maps
- Space Time 3D Raster Datasets (STR3DS) represent collections of time stamped 3D raster maps
- Space Time Vector Datasets (STVDS) represent collections of time stamped vector maps

Space time datasets (STDS) represent spatio-temporal fields in TGRASS. They are stored as table structures in the temporal database and can be

[5] Maps of type raster, 3D raster and vector.

created, updated and deleted. Raster, 3D raster and vector maps can be registered in several different space time datasets at the same time. The spatio-temporal extent as well as the granularity and the map type specific metadata of space time datasets is automatically computed from registered time stamped maps. The correctness of time stamps and the temporal topological validity is checked automatically. Cross referencing between time stamped maps and space time datasets was implemented to assure temporal data integrity and consistency.

3.1.3. Space Time Voxel Cubes

The third step was the introduction of time as the third dimension in the 3D raster GRASS C library to enable the support for space time voxel cubes. Space time voxel cubes support the same temporal types and time stamps as space time raster datasets. Space time raster datasets with valid temporal topology and interval time can be converted into space time voxel cubes. Every space time voxel cube can be converted into a space time raster dataset. The conversion is performed without information loss. Space time voxel cubes have equidistant sample resolutions for each axis (x, y, t). The unit of the spatial axis depends on the projection of the GRASS location. The unit of the temporal axis depends on the chosen temporal unit that can be of type years, months, days, hours, minutes or seconds. The axis specific spatial resolution is stored as double precision floating point values. In case of absolute time, the temporal resolution is stored as years, months or days with fractions of days representing hours, minutes and seconds relative to the date *Jan. 1. 1900 00:00:00 UTC*. This assures the correct temporal alignment of space time voxel cubes using the same temporal unit but different start or end times. All existing 3D raster modules can be used to process space time voxel cubes. This includes modules for cross section computation, uni-variate and zonal statistical analysis, 3D mask creation and 3D point sampling. Map calculation that allows spatio-temporal algebraic operations described in Jeremy et al. (2005) is supported as well. A limitation is that 3D map calculation is only allowed between space time cubes with the same temporal unit.

3.1.4. Spatio-temporal modules

In the last step temporal and spatio-temporal modules were implemented to provide spatio-temporal management, querying, analysis, processing, export, import and conversion functionality. These modules were designed to accept space time datasets as input for processing among maps of different type, numerical and textual parameter and files. Usually new space time datasets or textual contents are created as output. The modular concept of TGRASS allows nesting of temporal and spatial modules. Hence, the result of a spatio-temporal query created with a temporal module can be used as input parameter for spatial processing modules. The naming concept of TGRASS modules follows the established module naming convention of GRASS GIS:

- *t.* prefix:* Modules for temporal analysis and database management
- *t.rast.* prefix:* Modules processing space time raster datasets
- *t.rast3d.* prefix* Modules processing space time 3D raster datasets
- *t.vect.* prefix:* Modules processing space time vector datasets

An overview of implemented modules is given in Appendix A.

3.1.5. Visualization and data handling

TGRASS supports the direct visualization of raster time series. To create sophisticated animations including several space time raster and vector datasets, the display modules[6] can be utilized in conjunction with the temporal sampling module *t.sample*. Based on our temporal Python library new visualization modules have been implemented by Kratochvilova (2013):

- *g.gui.animate* to visualize and animate multiple space time raster and vector datasets
- *g.gui.timeline* to visually analyze the temporal topology of multiple space time raster and vector datasets

TGRASS was designed to handle and store high resolution, continental scale data represented as maps and space time datasets. It was successfully tested using different environmental datasets containing up to 150.000 maps with hourly, daily, monthly and yearly temporal resolution. Space time datasets that handle more than 20.000 maps were successfully used in spatio-temporal processing like aggregation and sampling. Limiting factors are the number and size of files and directories the used file system can manage and the number and size of tables the temporal database can handle.

[6] d.mon, d.rast, d.vect, d.title, d.text, d.legend and many more

3.2. Interoperability

Interoperability of a temporal GIS with existing spatio-temporal modeling, analysis and visualization applications multiplies its usefulness. Interoperability avoids redundant implementation of the same functionality and allows the user to combine different applications to perform complex tasks a single application is not capable of. Export interfaces for space time datasets and statistical spatio-temporal data to the following open source applications are provided:

- R environment for statistical computing (R)
- Climate Data Operator (CDO)
- ParaView

The statistical analysis spatio-temporal modules of TGRASS support output formats that can be directly imported in R for further analysis. With the introduction of space time voxel cubes a new export module to create NetCDF files was implemented. This export module *r3.out.netcdf* supports the export of spatial volumes and space time voxel cubes as NetCDF files. Comprehensive projection information as well as data and axis descriptions are provided in the NetCDF file, following the Climate and Forcast (CF) Conventions version 1.6[7]. This assures seamless processing, analysis and visualization of space time voxel cubes in R, CDO and ParaView. Additionally, the export of space time raster datasets as ParaView time series data using the legacy VTK (Kitware Inc., 2013b) format is provided.

4. Software availability

The source code of our implementation is licensed under the Gnu Public License (GPL) version 2 and is part of GRASS GIS 7. It is available via the software versioning and revision control system subversion[8] starting from GRASS GIS 7 revision 52369. The source code can be inspected using the GRASS GIS online source code browser[9]. Detailed compiling and installation

[7] `http://cf-pcmdi.llnl.gov/documents/cf-conventions/1.6/cf-conventions-multi.html`

[8] `http://trac.osgeo.org/grass/wiki/DownloadSource#GRASS7`

[9] `http://trac.osgeo.org/grass/browser/grass/trunk`

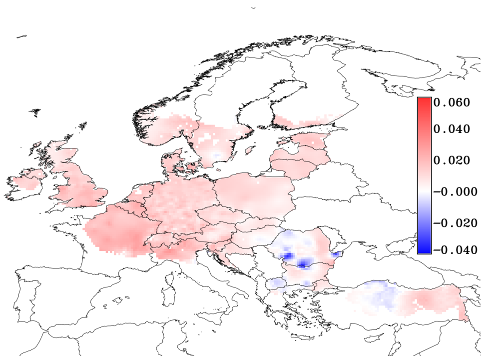
instructions as well as needed requirements are available at the GRASS GIS web site[10]. The author of TGRASS is Sören Gebbert.

5. Validation and verification

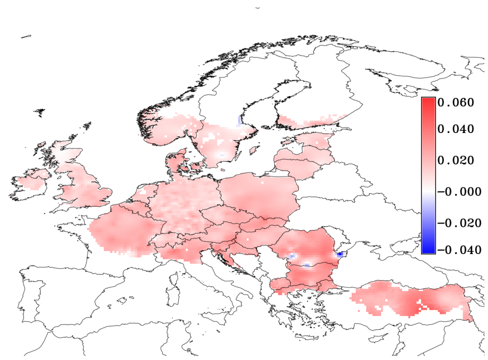
The functionality of the temporal GIS Python library was validated using automated Python tests that are directly integrated with the library source code. Additionally, more than 100 tests to validate the temporal and spatio-temporal modules were implemented using shell scripts.

The seasonal and yearly mean temperatures from 1950 - 2011 of the temperate climate zone of the European Union and Turkey were analyzed, to verify the inventory, analysis, modeling and display functionality of TGRASS as well as the data exchange capabilities with R and ParaView. A detailed description with code examples is available in Appendix B. The E-OBS datasets of daily temperature and daily precipitation with a spatial resolution of 0.25 degrees (Haylock et al., 2008) was used for this analysis. The daily mean temperature data was imported and registered in a new space time raster dataset with 22644 maps using the modules *r.in.gdal*, *t.create* and *t.register*. The next step was the monthly, seasonal and yearly aggregation of the daily average temperature data with *t.rast.aggregate*. The module *t.rast.extract* was used to extract specific space time raster datasets for each season. Then for each season specific *STRDS* the linear regression slope was computed from 1950 - 2011 with *t.rast.series*. The resulting maps are visualized in Figure 5. The module *t.rast.univar* was used to compute the mean seasonal temperature time series of the temperate climate zone. The output of *t.rast.univar* was imported into R to create the visualization shown in Figure 6a. Three vector points were created using the coordinates of the three capital cities Berlin, London and Paris. The points were used to sample the seasonal mean temperature space time raster datasets using *t.vect.observe.strds*. The resulting data were extracted with *t.vect.db.select* and visualized with R, see figures 6b, 6c and 6d. The 5 year mean temperature of the yearly temperature space time raster dataset was computed with *r3.mapcalc* after the conversion of the *STRDS* into a space time voxel cube with *t.rast.to.rast3*. The resulting space time voxel cube with a temporal extent from 1952 to 2008 and a granularity of one year was exported as netCDF file with *r3.out.netcdf* and visualized

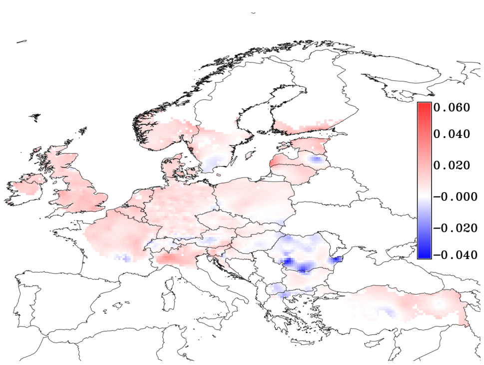
[10] `http://grass.osgeo.org/`



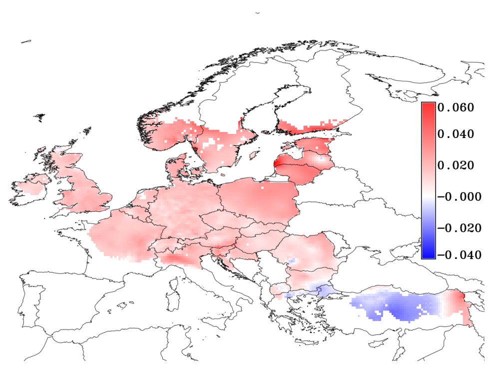
(a) Linear regression slope of spring mean temperature



(b) Linear regression slope of summer mean temperature

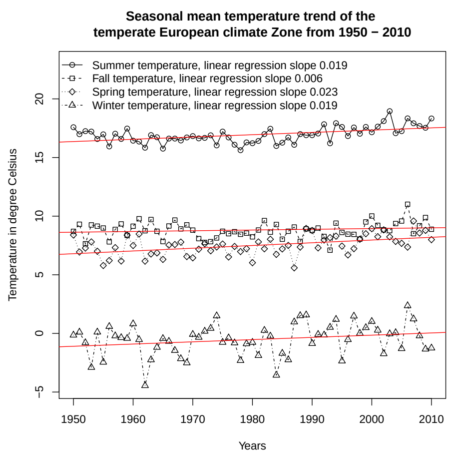


(c) Linear regression slope of fall mean temperature

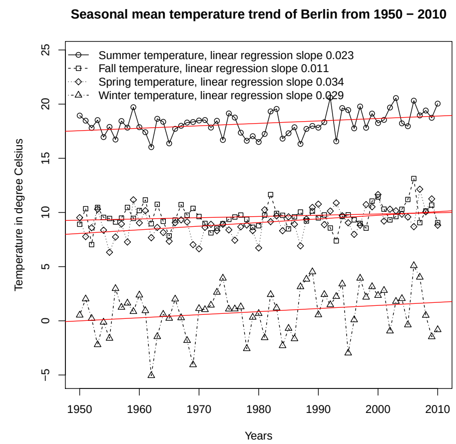


(d) Linear regression slope of winter mean temperature

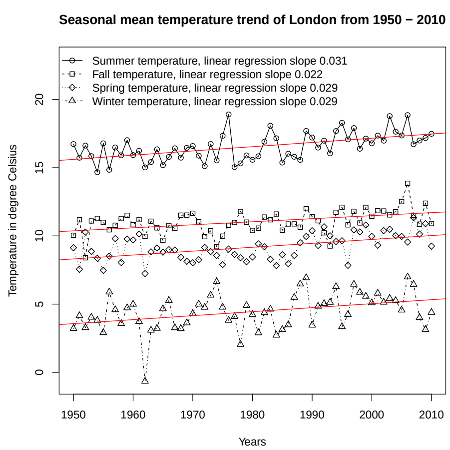
Figure 5: The linear regression slope computed for all seasons from 1950 - 2011. Red color indicates rising temperature, blue indicates falling temperature. Units are degree Celsius per year.



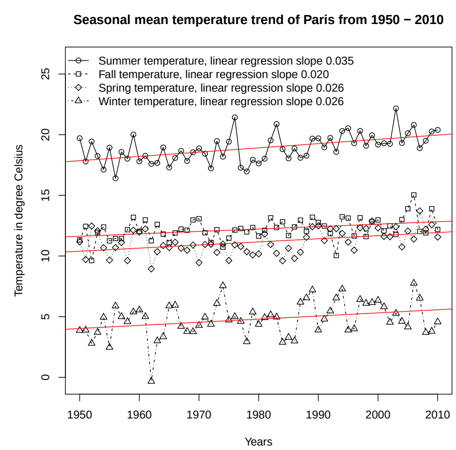
(a) Seasonal temperature trend for Europe



(b) Seasonal temperature trend for Berlin



(c) Seasonal temperature trend for London



(d) Seasonal temperature trend for Paris

Figure 6: Mean temperature trend for the temperate climate zone of the European Union and the capitals Berlin, London and Paris.

with ParaView. Four screenshots of the ParaView time series visualization are shown in Figure 7.

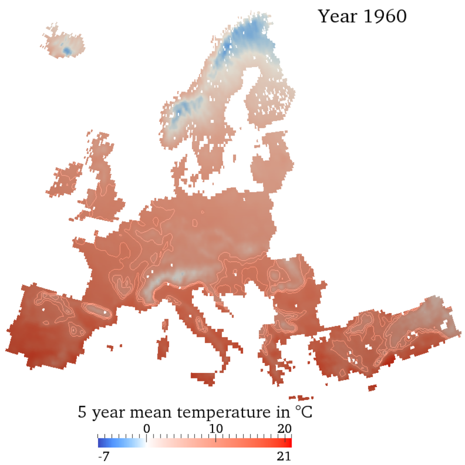
6. Discussion

This paper presents TGRASS, a temporal GIS for field based environmental modeling. We demonstrate that an existing spatial geographic information system can be modified into a field based temporal GIS. An extended snapshot approach allows the reuse of the spatial management, analysis, processing and visualization capabilities of GRASS for spatio-temporal tasks.

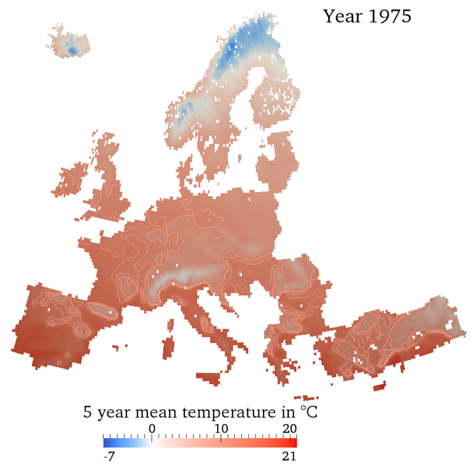
GRASS provides support for different spatial fields represented as raster, 3D raster and vector maps. We integrated the time dimension in GRASS by implementing time stamp support for all spatial fields. Spatio-temporal fields are represented in TGRASS as space time datasets. Space time datasets allow the efficient handling, analysis and processing of massive data using simple temporal GIS commands. Dedicated modules for temporal management simplify the handling of space time datasets and time stamped maps. The decision to use linear discrete time instances and intervals as time stamps for spatial fields allows a broad application in environmental modeling. Cyclic time can be emulated using scripts that loop over space time datasets. Different branches from branching time can be represented by scenario specific space time datasets. However, the linear discrete time approach leads to management overhead for models that are based on cyclic or branching time.

Our approach allows the investigation of spatio-temporal relations between space time datasets of different kind (raster, 3D raster and vector) using a combination of temporal sampling methods and spatial sampling modules. The module *t.sample* was designed to describe the temporal relationships between space time datasets. The textual output of this module can be used as input for several spatial modules that perform spatial sampling for example: *r3.cross.rast, v.what.rast, v.what.rast3, v.what.vect, v.rast.stats*.

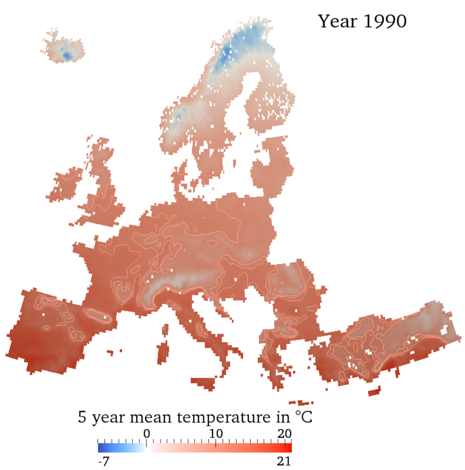
Map calculations between space time datasets of type raster or 3D raster are supported by dedicated modules using different temporal sampling methods. The module *r3.mapcalc* can be used for sophisticated spatio-temporal map calculations of space time voxel cubes. Neighborhood analysis as provided for space time voxel cubes using *r3.mapcalc* are not supported for space time raster and 3D raster datasets. However, space time raster datasets with valid temporal topology and interval time can be converted into space time voxel cubes using the module *t.rast.to.rast3*.



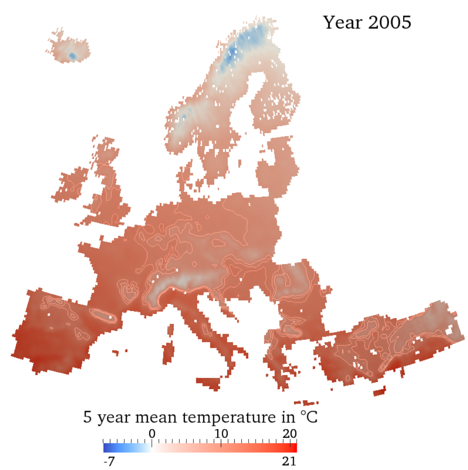
(a) Five year mean temperature in 1960



(b) Five year mean temperature in 1975



(c) Five year mean temperature in 1990



(d) Five year mean temperature in 2005

Figure 7: Four screenshots showing the five year mean temperature of the European Union and Turkey visualized with ParaView as key frame animation from 1952 to 2008. Contour lines have been created for 9, 10 and 11 degree Celsius.

Spatio-temporal queries are supported for all space time datasets using a combination of SQL WHERE statements for temporal related selection and the spatial and attribute querying capabilities of spatial modules. In addition, simple spatio-temporal and attribute specific queries are supported as SQL WHERE statements, since the temporal and spatial extent as well as map specific metadata is stored in the temporal SQL database.

TGRASS supports states, events and processes. States are an essential part of our temporal GIS approach. Events can be detected by combining the temporal topology capabilities of TGRASS with its spatial overlay functionality. Hence differences between temporally related time stamped maps can be computed and stored, regardless of the type of the maps. However, our approach lacks efficiency in storage and computation in comparison to the ESTDM approach. The representation of a process (Yuan, 2001) is available in TGRASS using space time datasets.

In the context of field-based modelling, a remaining future challenge is to better visualize space time datasets, and to handle events and nested processes more efficiently. In the broader context of modelling spatio-temporal phenomena, challenges the integration of our field based modelling environment with non-field based phenomena such as trajectories, lattice data, and (marked) point patterns (Stasch et al., 2014).

7. Conclusion

We implemented TGRASS, a temporal GIS for field based environmental modeling based on the open source geographical information system GRASS. The introduction of space time datasets that represent spatio-temporal fields in TGRASS, allows the efficient management and processing of massive environmental data and the analysis of relations between spatio-temporal fields. A comprehensive tool set for spatio-temporal management, analysis, processing and visualization is now available with the implementation of several temporal and spatio-temporal modules and the possibility to combine spatial modules with temporal modules. Our temporal GIS supports the import and export of the widely used spatio-temporal data format netCDF to assure data interoperability to existing sophisticated spatio-temporal analysis and visualization software CDO and ParaView. The structured textual output of the analysis modules allows the direct processing and visualization with the R statistical environment. The analysis of a massive climate dataset has demonstrated the environmental modeling capabilities of our approach.

Acknowledgements

We would like to thank Annette Freibauer and Rene Dechow for their help and critical review of this work.

Allen, J. F., 1983. Maintaining knowledge about temporal intervals. Communications of the ACM 26 (11), 832–843.

Aquaveo LLC, 2013. Arc Hydro Groundwater.
URL `http://www.aquaveo.com/archydro-groundwater`

Bettini, C., Dyreson, C. E., Evans, W. S., Snodgrass, R. T., 1998. A Glossary of Time Granularity Concepts. Lecture Notes in Computer Science 1399.

Câmara, G., Vinhas, L., Ferreira, K. R., De, G. R., Cartaxo, R., Souza, M., Miguel, A., Monteiro, V., Carvalho, M. T. D., Casanova, M. A., De, U. M., 2008. Terralib: An open source gis library for large-scale environmental and socio-economic applications.
URL `http://www.terralib.org/`

Christakos, G., Bogaert, P., Serre, M. L., 2001. Temporal GIS: advanced functions for field-based applications. Springer.

Cova, T., Church, R., Goodchild, M., 2002. Extending geographical representation to include fields of spatial objects. International Journal of Geographical Information Science 16, 509—532.

de Senna Carneiro, T. G., de Andrade, P. R., Cmara, G., Monteiro, A. M. V., Pereira, R. R., 2013. An extensible toolbox for modeling naturesociety interactions. Environmental Modelling & Software 46 (0), 104 – 117.
URL `http://www.sciencedirect.com/science/article/pii/S1364815213000534`

Discovery Software Ltd., 2013. STEMgis: a Temporal Geographic Information System. 11 St. Mary's Park, Paignton, Devon, TQ4 7DA, United Kingdom.
URL `http://www.discoverysoftware.co.uk/STEMgis.htm`

Dyreson, C., Grandi, F., Käfer, W., Kline, N., Lorentzos, N., Mitsopoulos, Y., Montanari, A., Nonen, D., Peressi, E., Pernici, B., Roddick, J. F., Sarda, N. L., Scalas, M. R., Segev, A., Snodgrass, R. T., Soo, M. D., Tansel, A., Tiberio, P., Wiederhold, G., Mar. 1994. A consensus glossary of temporal database concepts. SIGMOD Rec. 23 (1), 52–64.

Forer, P., 1998. Geometric approaches to the nexus of time, space, and microprocess: implementing a practical model for mundane socio-spatial systems. In: Egenhofer MJ, Golledge RG (eds) Spatial and temporal reasoning in Geographic Information Systems. Oxford University Press, New York/Oxford, pp. 171 – 190.

Frank, U., 1998. Different types of "Time" in GIS. In: Egenhofer MJ, Golledge RG (eds) Spatial and temporal reasoning in Geographic Information Systems. Oxford University Press, New York/Oxford, pp. 40 – 62.

Galton, A., 2001. A Formal Theory of Objects and Fields. Spatial Information Theory. Foundations of Geographic Information Science : International Conference, COSIT 2001 Morro Bay, CA, USA, September 19-23, 2001. Proceedings, 458–473.

Galton, A., 2004. Fields and Objects in Space, Time, and Space-time. Spatial Cognition & Computation 4 (1), 39–68.

Goodchild, M., Gopal, S., 1989. Modeling error in objects and fields. Accuracy of Spatial Databases, 107–114.

Goodchild, M. F., May 1992. Geographical data modeling. Computers & Geosciences 18 (4), 401–408.

Hardisty, F., 2013. GeoViz: toolkit for geographic visualization and analysis. URL `http://code.google.com/p/geoviz/`

Haylock, M. R., Hofstra, N., Tank, A. M. G. K., Klok, E. J., Jones, P. D., New, M., Oct. 2008. A European daily high-resolution gridded data set of surface temperature and precipitation for 1950 - 2006. Journal of Geophysical Research 113 (D20).

Jeremy, M., Roland, V., Dana, T. C., 2005. Cubic map algebra functions for spatio-temporal analysis. Cartography and Geographic Information Science 32 (1), 17–32.

Kitware Inc., 2013a. ParaView: an open-source, multi-platform data analysis and visualization application.
URL `http://www.paraview.org/`

Kitware Inc., 2013b. Visualization Toolkit (VTK): an open-source, freely available software system for 3d computer graphics, image processing and visualization.
URL http://www.vtk.org/

Kratochvilova, A., 2013. Visualization of spatio-temporal data in grass gis. Master's thesis, Faculty of Civil Engineering, Czech Technical University in Prague.
URL http://geo.fsv.cvut.cz/proj/dp/2013/anna-kratochvilova-dp-2013.pdf

Langran, G., Mar. 1992. Time in geographic information systems. Taylor & Francis.

Mitasova, H., Hardin, E., Starek, M., Harmon, R. S., Overton, M., 2011. Landscape dynamics from lidar data time series. In: Hengl, T., Evans, I. S., Wilson, J. P., Gould, M. (Eds.), Geomorphometry 2011. Redlands, CA, pp. 3–6.

Mitasova, H., Mitas, L., Brown, W., Gerdes, D., Kosinovsky, I., Baker, T., Jul. 1995. Modelling spatially and temporally distributed phenomena: new methods and tools for GRASS GIS. International Journal of Geographical Information Science 9 (4), 433–446.

Neteler, M., 2005. Time series processing of MODIS satellite data for landscape epidemiological applications. International Journal of Geoinformatics. Special Issue on FOSS/GRASS 2004 & GIS-IDEAS 2004 1 (1), 133–138.

Neteler, M., Jan. 2010. Estimating Daily Land Surface Temperatures in Mountainous Environments by Reconstructed MODIS LST Data. Remote Sensing 2 (1), 333–351.

Neteler, M., Bowman, M. H., Landa, M., Metz, M., 2012. GRASS GIS: A multi-purpose open source GIS. Environmental Modelling & Software 31, 124–130.

Neteler, M., Mitasova, H., Nov. 2008. Open Source GIS: A GRASS GIS Approach, 3rd Edition. Springer, New York.
URL http://www.grassbook.org/

O'Sullivan, D., Dec. 2005. Geographical information science: time changes everything. Progress in Human Geography 29 (6), 749 –756.

Ott, T., Swiaczny, F., 2001. Time-integrative geographic information systems: management and analysis of spatio-temporal data. Springer.

PCRaster team, 2012. PCRaster: a collection of tools and software libraries for environmental modeling.
URL http://pcraster.geo.uu.nl/

Pebesma, E., 2012. spacetime: Spatio-Temporal Data in R. Journal of Statistical Software 51 (7).

Pelekis, N., Theodoulidis, B., Kopanakis, I., Theodoridis, Y., Jun. 2004. Literature review of spatio-temporal database models. The Knowledge Engineering Review 19 (3), 235 – 274.

Peuquet, D., Duan, N., Jan. 1995. An event-based spatiotemporal data model (ESTDM) for temporal analysis of geographical data. International Journal of Geographical Information Science 9 (1), 7–24.

Peuquet, D. J., Hardisty, F., 2010. STempo: an interactive visualization and statistical environment for discovery and analysis of space-time patterns.
URL http://www.geovista.psu.edu/stempo/

R Development Core Team, 2012. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0.
URL http://www.R-project.org

Schulzweida, U., 2013. Climate Data Operators: a large tool set for working on climate and NWP model data.
URL http://code.zmaw.de/projects/cdo

Stasch, C., Scheider, S., Pebesma, E., Kuhn, W., 2014. Meaningful spatial prediction and aggregation. Environmental Modelling & Software 51 (0), 149 – 165.
URL http://www.sciencedirect.com/science/article/pii/S1364815213001977

Worboys, M. F., 1998. A generic model for spatio-bitemporal geographic information. In: Egenhofer MJ, Golledge RG (eds) Spatial and temporal reasoning in Geographic Information Systems. Oxford University Press, New York/Oxford, pp. 25 – 39.

Yuan, M., Apr. 2001. Representing Complex Geographic Phenomena in GIS. Cartography and Geographic Information Science, 83–96.

Yuan, M., 2009. Challenges and critical issues for temporal GIS research and technologies. In: Hassan A. Karimi (eds) Handbook of Research on Geoinformatics. IGI Global, Hershey, pp. 144–153.

Zorer, R., Rocchini, D., Delucchi, L., Zottele, F., Meggio, F., Neteler, M., Jul. 2011. Use of multi-annual MODIS land surface temperature data for the characterization of the heat requirements for grapevine varieties. In: Analysis of Multi-temporal Remote Sensing Images (Multi-Temp), 2011 6th International Workshop on the. IEEE, pp. 225–228.

Appendix A. Temporal GRASS GIS Modules

Table A.1: Modules for temporal analysis and database management

Module	Description
t.connect	Sets or shows the temporal GIS database connection information of the current mapset
t.create	Create the structure of a new space time dataset in the temporal GIS database
t.info	Lists information about space time datasets and maps registered in the temporal GIS database
t.list	Lists space time datasets and maps that are registered in the temporal database
t.register	Register raster, vector and 3D raster maps in the temporal GIS database or additionally in a space time dataset
t.remove	Remove space time datasets from the temporal GIS database
t.rename	Renames a space time dataset
t.sample	Sample input space time dataset(s) with a sample space time dataset and print the result
t.support	Modifies and update the metadata of a space time dataset
t.topology	List temporal relations of the maps in a space time dataset
t.unregister	Unregister raster, vector and 3D raster maps from the temporal GIS database or a specific space time dataset

Table A.2: Space time raster dataset processing modules

Module	Description
t.rast.aggregate	Create a new space time raster dataset from the aggregated data of an existing space time raster dataset
t.rast.aggregate.ds	Aggregated data of an existing space time raster dataset using the temporal topology of a second space time dataset
t.rast.colors	Creates/modifies the color table associated with each raster map of the space time raster dataset
t.rast.export	Export a space time raster dataset
t.rast.extract	Extract a subset of a space time raster dataset
t.rast.gapfill	Replace gaps in a space time raster dataset with interpolated raster maps
t.rast.import	Import a space time raster dataset
t.rast.list	List registered maps of a space time raster dataset
t.rast.mapcalc	Perform r.mapcalc computations of temporal related raster maps in space time raster datasets
t.rast.out.vtk	Export a space time raster dataset as VTK time series
t.rast.series	Perform different aggregation algorithms from r.series on all or a subset of raster maps in a space time raster dataset
t.rast.to.rast3	Convert a space time raster dataset into a 3D raster map representing a space time voxel cube
t.rast.univar	Calculates univariate statistics from the non-null cells for each registered raster map of a space time raster dataset

Table A.3: Space time 3D raster dataset processing modules

Module	Description
t.rast3d.extract	Extract a subset of a space time 3D raster dataset
t.rast3d.list	List registered maps of a space time 3D raster dataset
t.rast3d.mapcalc	Perform r3.mapcalc computations of temporal related 3D raster maps in space time 3D raster datasets
t.rast3d.univar	Calculates uni-variate statistics from the non-null cells for each registered 3D raster map of a space time 3D raster dataset

Table A.4: Space time vector dataset processing modules

Module	Description
t.vect.db.select	Prints attributes of vector maps registered in a space time vector dataset
t.vect.export	Export a space time vector dataset
t.vect.extract	Extract a subset of a space time vector dataset
t.vect.import	Import a space time vector dataset
t.vect.list	List registered maps of a space time vector dataset
t.vect.observe.strds	Observe specific locations in a space time raster dataset over a periode of time using vector points
t.vect.what.strds	Sample a space time raster dataset at spatio-temporal locations of a space time vector dataset
t.vect.univar	Compute uni-variate statistics of a space time vector dataset based on a single attribute row

Appendix B. Analyzing seasonal mean temperatures in the temperate climate zone of Europe

This is a detailed description of the mean temperature E-OBS dataset analysis that was used to verify the capabilities of TGRASS. The following workflow was performed on a 64Bit AMD Linux system. GRASS in version 7 was compiled and installed from the source code. All commands must be executed in the GRASS Unix command shell.

The E-OBS temperature and precipitation gridded datasets, provided by the ECA&D as netCDF files, has a daily temporal resolution and a spatial resolution of 0.25 degrees. The dataset was download as several compressed netCDF files from the ECA&D website[11]. Each netCDF file was imported into GRASS GIS using the module *r.in.gdal* with the specification of the band number offset to assure chronological numbering of the imported raster maps. The flag $-o$ indicates that the projection check should be skipped to allow the import. The reason for this is that the ECAD netCDF files do not include projection informations.

```
r.in.gdal -o input=tg_0.25deg_reg_1950-1964_v5.0.nc \
    output=temperature_mean offset=0

```

[11] http://eca.knmi.nl/download/ensembles/data/Grid_0.25deg_reg/

```
r.in.gdal -o input=tg_0.25deg_reg_1965-1979_v5.0.nc \
    output=temperature_mean offset=5479

r.in.gdal -o input=tg_0.25deg_reg_1980-1994_v5.0.nc \
    output=temperature_mean offset=10957

r.in.gdal -o input=tg_0.25deg_reg_1995-2011_v5.0.nc \
    output=temperature_mean offset=16436
```

A space time raster dataset named *temperature_mean_1950_2011_daily* was created to simplify the handling of more than 22000 raster maps:

```
t.create type=strds output=temperature_mean_1950_2011_daily \
     temporal=absolute \
     title="European mean temperature 1950-2011" \
     description="The European daily mean temperature from 1950 - 2011"
```

A small Python script was implemented that created the input text file for the module *t.register* to support the registration of all imported raster maps in the space time raster dataset *temperature_mean_1950_2011_daily*. To generate the interval time stamps the start date was set to the first of January 1950 using a time increment of one day:

```
cat > ECAD_climate_data_timeseries_1950_2011.py << EOF
file = open("map_list.txt", "w")
for i in range(22461):
    file.write("temperature_mean.%i\n" % (i + 1))
file.close()
EOF

python ECAD_climate_data_timeseries_1950_2011.py

t.register -i type=rast input=temperature_mean_1950_2011_daily \
    file=map_list.txt start=1950-01-01 increment="1 day"
```

The daily data was aggregated with the module *t.rast.aggregate* to monthly, seasonal and yearly granularity. Using the module *g.region* the correct region and resolution for temporal aggregation was set. Spatial aggregation was not required in this case.

```
g.region -p rast=temperature_mean.1

t.rast.aggregate input=temperature_mean_1950_2011_daily \
    method=average \
    output=temperature_mean_1950_2011_monthly \
    base=temperature_mean_monthly \
    granularity="1 month"

t.rast.aggregate input=temperature_mean_1950_2011_monthly \
    method=average \
    output=temperature_mean_1950_2011_seasonal \
    base=temperature_mean_seasonal \
    granularity="3 months"\
```

```
    where="start_time >= '1950-03-01' and start_time < '2011-02-01'"

t.rast.aggregate input=temperature_mean_1950_2011_monthly \
    method=average \
    output=temperature_mean_1950_2011_yearly \
    base=temperature_mean_yearly \
    granularity="1 year" \
    where="start_time < '2011-01-01'"
```

The temperature raster maps of spring, summer, fall and winter were extracted separately using SQL WHERE statements. This results in new space time raster datasets. Additionally the unit of the temperature was converted from 0.01 degree Celsius into 1 degree Celsius using a raster map calculation expression. The following command was repeated for each season using different monthly offsets and output naming:

```
t.rast.extract input=temperature_mean_1950_2011_seasonal \
    output=temperature_mean_1950_2011_spring \
    where="start_time = datetime(start_time, 'start of year', '2 month')" \
    expression="temperature_mean_1950_2011_seasonal / 100.0" \
    base=temp_mean_spring
```

A mask was applied to analyze the seasonal temperature for the temperate climate zone. The mask was based on the extraction of the thermal climate zone using the GRASS map calculator *r.mapcalc* and the thermal climate zone map provided by *Environmental conditions* of Food Insecurity, Poverty and Environment Global GIS Database (FGGD)[12]. Figure B.8 shows the simplified climate zones map that was used. The temperature trend of spring, summer, fall and winter was computed with *t.rast.series* using the option *slope* to compute the linear regression slope for each season. This slope represents the average temperature change per year in degree Celsius. The following command combination was repeated for each season:

```
t.rast.series input=temperature_mean_1950_2011_spring \
    output=temperature_mean_1950_2011_spring_slope \
    method=slope
```

Additionally the color for each map was set to the predefined GRASS color table *differences* by taking the temperature range of all maps into account.

```
r.colors map=temperature_mean_1950_2011_fall_slope,
    temperature_mean_1950_2011_spring_slope,
    temperature_mean_1950_2011_summer_slope,
    temperature_mean_1950_2011_winter_slope \
    color=difference
```

[12] http://www.fao.org/geonetwork/srv/en/metadata.show?id=14056

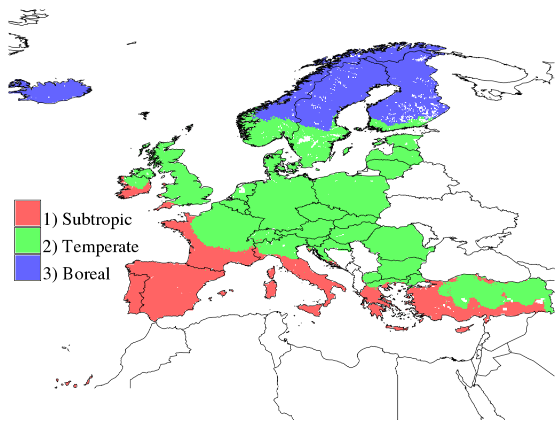


Figure B.8: Simplified thermal climate zones of the European Union and Turkey derived from the thermal world climate zones provided by *Environmental conditions* of Food Insecurity, Poverty and Environment Global GIS Database.

Since the output of the module *t.rast.series* is a raster map, the common display modules of GRASS can be used for visualization. For each season the temperature trend, the trend legend and the administrative borders were shown. The administrative boundaries are part of the Eurostat NUTS[13] dataset. The following commands were repeated for each season. Results are shown in Figure 5.

```
d.mon start=wx0
d.rast map=temperature_mean_1950_2011_spring_slope
d.vect map=administrative_boundaries
d.legend map=temperature_mean_1950_2011_spring_slope \
    at=55,95,83,85 range=-0.04,0.06
```

The module *t.rast.univar* was used to compute the mean seasonal temperature for the temperate climate zone. The analysis resulting in Figure 6a was performed with the combination of the module *t.rast.univar* and R for plotting time series data.

To analyze the seasonal temperature trend of the cities Paris, London and Berlin, a simple text file with latitude/longitude coordinates was created and imported as vector points map with *v.in.ascii*.

[13] `http://epp.eurostat.ec.europa.eu/portal/page/portal/gisco_Geographical_information_maps/popups/references/administrative_units_statistical_units_1`

```
cat > capital_coords.txt << EOF
Paris|48.856667|2.351667
London|51.50939|-0.11832
Berlin|52.518611|13.408056
EOF

v.in.ascii input=capital_coords.txt output=observations \
    x=3 y=2 columns="capital TEXT, y DOUBLE, x DOUBLE"
```

The imported vector points *observations* were used to sample the spring, summer, fall and winter mean temperature using the module *t.vect.observer.strds*. New space time vector datasets were created to store for each time stamped raster map the point sampled values in time stamped attribute tables. The module *t.vect.db.select* was used to extract the temperature history for each season and city. The following command was used to create the observations for Paris in spring:

```
t.vect.observe.strds input=observations \
    output=spring_observations \
    vector=spring_observations_1950_2011 \
    strds=temperature_mean_1950_2011_spring \
    column=temperature

t.vect.db.select input=spring_observations where="cat = 1" \
    column=temperature
```

The output of *t.vect.db.select* for each season and capital was imported into the R statistical environment to create Figure 6b, 6c and 6d. This is a shortened version of the resulting output of *t.vect.db.select*:

```
start_time|end_time|temperature
1950-03-01 00:00:00|1950-06-01 00:00:00|11.1725734767
1951-03-01 00:00:00|1951-06-01 00:00:00|9.7168243728
1952-03-01 00:00:00|1952-06-01 00:00:00|12.4741541219
...
2008-03-01 00:00:00|2008-06-01 00:00:00|12.2055340502
2009-03-01 00:00:00|2009-06-01 00:00:00|12.5818673835
2010-03-01 00:00:00|2010-06-01 00:00:00|11.5653046595
```

The yearly aggregated mean temperature space time dataset was converted into a space time voxel cube to perform spatio-temporal map calculations. The goal was to compute the 5 year mean temperature for each voxel and to analyze it visually in ParaView for the European Union and Turkey. The first steps was to adjust the MASK and region settings with *r.mask* and *g.region* followed by the transformation using the module *t.rast.to.rast3*. Then the space time voxel cube map calculation was performed with the module *r3.mapcalc*. The temporal unit and the time stamp must be explicitly set with *r3.support* and *t.register*. Finally the resulting space time voxel cube was exported as netCDF file and visualized with ParaView, see Figure 7. The commands were executed in the following order:

```
r.mask thermal_climate_zones_europe

g.region -p3 rast=temperature_mean.1

t.rast.to.rast3 input=temperature_mean_1950_2011_yearly \
    output=vol_1y_mean

g.region -p3 rast3=vol_1y_mean

r3.mapcalc expression="vol_5y_mean = (vol_1y_mean[0,0,-2] + \
                                      vol_1y_mean[0,0,-1] + \
                                      vol_1y_mean[0,0, 0] + \
                                      vol_1y_mean[0,0, 1] + \
                                      vol_1y_mean[0,0, 2])/500.0"

t.register type=rast3 map=vol_5y_mean start="1950-01-01" end="2011-01-01"
r3.support map=vol_5y_mean vunit="years"

r3.out.netcdf input=vol_5y_mean output=vol_5y_mean.nc null=-1000
```