UNRAVELLING TIME-LAGGED TELECONNECTIONS BETWEEN SEA SURFACE TEMPERATURE AND VEGETATION ANOMALIES

MICHAEL WESS

Mat.-No.: 0540427

Supervisor:
Univ.Prof. Dr.rer.nat. Clement Atzberger

Institute of Surveying, Remote Sensing and Land Information
University of Natural Resources and Life Sciences, Vienna

March 2014 – Version 1.1

Michael Wess: *Unravelling time-lagged teleconnections between sea surface temperature and vegetation anomalies*

Dedicated to Jaqueline Sunshine

ABSTRACT

A lot of studies have already shown a correlation (or anti-correlation) between the Sea Surface Temperature (SST) and the NDVI in different parts of the world, especially in regions affected by the Southern Oscillation (SO), the atmospheric component of the El Niño cycle. The goal of this thesis is to provide a basis for future studies regarding short and mid-term NDVI forecasts based on SST data. We want to try if Recurrence Quantification Analysis (RQA) is able to to identify, characterize and quantify possible (time-lagged) relationships between NDVI and SST, based on weekly world-wide data from the years between 1985 and 2006. SST data was adopted from the AVHRR Pathfinder 5.0 project, NDVI information was provided by the GIMMS project. Both time series were smoothed, the NDVI series were grouped into 1135 classes, for each class the mean NDVI series was determined and then the anomalies of both datasets were calculated and normalized. After that the diagonal-wise RQA indices DET, RR and L (with $l_{min} = 5$) for diagonals with a time lag of 0 to 48 weeks were calculated for each SST-NDVI pair, based on Cross Recurrence Plots (CRPs) with a threshold of 0.8, a dimension of 20 and time delay of 1.

Map plots of the resulting RQA indices clearly show no random distribution, but distinctive structures and patterns highlighting different sea areas with potential (tele-)connections with the NDVI classes. A detailed interpretation of these patterns however goes beyond the scope of this thesis. Future work is necessary, preferably together with experts in oceanography and climatology, to be able to verify and explain those structures. To us, RQA seems to be a promising approach 1) to automatically identify time-lagged relationships between climate variables and the NDVI (and thus agricultural production), 2) to quantify those relationships and 3) ultimately to develop forecast models for crop conditions based on remotely sensed data using this method.

ZUSAMMENFASSUNG

Viele Studien haben bereits eine Korrelation (oder Antikorrelation) zwischen Meeresoberflächentemperatur (Sea Surface Temperature) und NDVI in unterschiedlichsten Teilen der Welt festgestellt, besonders in Gebieten, die von der sogenannten Southern Oscillation (SO), der atmosphärischen Komponente des El Niño-Kreislaufs, betroffen sind. Ziel dieser Arbeit ist es, eine Grundlage für weitere Studien über

kurz- und mittelfristige NDVI-Vorhersagen auf Basis der Meeresoberflächentemperatur zu schaffen. Wir wollen überprüfen, ob die Methode Recurrence Quantification Analysis (RQA) geeignet ist, um mögliche zeitversetzte Zusammenhänge zwischen NDVI und Meerestemperatur zu identifizieren, zu charakterisieren und zu quantifizieren, basierend auf weltweiten, wöchentlichen Messdaten der Jahre 1985 bis 2006. Quelle für die Temperaturdaten war das AVHRR Pathfinder 5.0-Projekt, die NDVI-Daten stammen aus dem GIMMS-Projekt. Beide Zeitreihen wurden zunächst geglättet, dann die NDVI-Daten in 1135 Klassen unterteilt, für jede Klasse die mittlere NDVI-Signatur bestimmt und zuletzt für alle Zeitreihen die Anomalien berechnet und normalisiert. Danach wurde für jedes SST-NDVI-Paar ein Cross Recurrence Plot (CRP) mit den einem *threshold* von 0.8, einer *dimension* von 20 und eine *time delay* von 1 erstellt und daraus die diagonalweisen RQA-Indizes DET, RR und L (mit $l_{min} = 5$) für die Diagonalen mit einem Zeitversatz von 0 bis 48 Wochen ermittelt.

Kartendarstellungen der berechneten RQA-Indizes zeigen deutlich keine zufällige Verteilung, sondern unterscheidbare Strukturen, die verschiedene Meeresgebiete mit möglichen Zusammenhängen mit den NDVI-Klassen ausweisen. Eine detaillierte Interpretation dieser Muster geht aber über den Umfang dieser Arbeit hinaus. Hier sind weitere Forschungen notwendig, am besten in Zusammenarbeit mit Experten für Ozeanographie und Klimatologie, um die Ergebnisse verifizieren und interpretieren zu können. Für uns scheint RQA jedoch ein vielversprechender Ansatz 1) zur automatisierten Identifikation von zeitversetzten Zusammenhängen zwischen Klimavariablen und dem NDVI (und damit der landwirtschaftlichen Produktion), 2) zur Quantifizierung dieser Zusammenhänge und 3) letztendlich zur Entwicklung von Prognosemodellen für den Zustand der Vegetation basierend auf Fernerkundungsdaten zu sein.

There is a theory which states that if ever anyone discovers exactly what the Universe is for and why it is here, it will instantly disappear and be replaced by something even more bizarre and inexplicable. There is another theory which states that this has already happened.

— Douglas Adams [1]

ACKNOWLEDGMENTS

I wish to thank various people for their contribution to this thesis: My supervisor, Univ.Prof. Dr.rer.nat. Clement Atzberger, for his patient guidance, encouragement and useful critiques of my work, and Dr. Anja Klisch, Dipl.-Ing. Matteo Mattiuzzi, Dr. Markus Stöhr and Dipl.-Ing. Dr. Jan Zabloudil for their help in handling the big amount of data necessary for this project.

I would also like to extend my thanks to Dr. Norbert Marwan for providing me with his helpful MATLAB CRP Toolbox and his advice in its application, and to Prof. Dr.-Ing. André Miede for releasing this beautiful LaTeX theme and his support with my typesetting questions.

Finally, I wish to thank my parents for their support and encouragement throughout my study, my brothers who introduced me to LaTeX and LyX, and in particular my wife – I still wouldn't have finished this project by now without her.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LISTINGS

ACRONYMS

AVHRR Advanced Very High Resolution Radiometer

CRP Cross Recurrence Plot

ENSO El Niño-Southern Oscillation

GAC Global Area Coverage

GHRSST Group for High-Resolution Sea Surface Temperature

GIMMS Global Inventory Modeling and Mapping Studies

HDF	Hierarchical Data Format
LOI	Line Of Identity
NDVI	Normalized Difference Vegetation Index
NODC	US National Oceanographic Data Center
RP	Recurrence Plot
RQA	Recurrence Quantification Analysis
SO	Southern Oscillation
SST	Sea Surface Temperature
TIFF	Tagged Image File Format
VSC	Vienna Scientific Cluster

SYMBOLS AND NOTATIONS

ϵ	threshold or radius of neighborhood of a recurrence plot
CRM	Cross Recurrence Matrix
DET	Determinism
L	Average Diagonal Line Length
RM	Recurrence Matrix
RR	Recurrence Rate
T_{rec}	Recurrence Period
approx.	approximately
e. g.	for example
etc.	*et cetera*, and so on
i. e.	this is

Part I

INTRODUCTION

1 MOTIVATION

Especially as a result of today's growing concern about the global climate change and its possible impacts on agricultural production and human development in general, understanding the complex relationships between climate and vegetation dynamics has become a major task of Earth sciences [2] and thus of the remote sensing community. In large parts of the world, water limitation is the primary factor responsible for interannual variations in performance of rainfed crops, and hundreds of millions of people depend upon the success of these crops for their livelihood [3]. Consequently, a lot of effort is devoted to the implementation of forecast systems and prediction models of crop conditions. Such models can be based on known time-lagged correlations between different climate variables like precipitation, temperature or air pressure to estimate the future performance of rainfed crops.

A lot of studies have already shown a correlation (or anti-correlation) between the Sea Surface Temperature (SST) and the Normalized Difference Vegetation Index (NDVI) in different parts of the world [4, 3, 2, 5], especially in regions affected by the Southern Oscillation (SO), the atmospheric component of the El Niño cycle. Consequently, it might be possible to use this relationship to develop, calibrate and validate crop condition forecast models. As SSTs are partially predictable up to 2 years ahead of time [6], a new method of exploring its relationship with world-wide vegetation dynamics might contribute to the development of forecast models.

Recurrence Quantification Analysis (RQA), an extension of Recurrence Plots (RPs), is a powerful method to analyze, quantify and compare the (possibly nonlinear) behavior of dynamical systems [7]. It is well-known in various fields like earth sciences [8], chemistry [9], sociology [10] or mechanics [11], but apart from Li et al., who used RQA to measure determinism and predictability of NDVI series and its spatial patterns [12], it is barely used by the remote sensing community.

As this work uses datasets covering the whole Earth's surface, possible correlations between SST and NDVI can occur on a smaller spatial scale, because coastal waters and vegetation areas nearby the shore might be affected by the same climatical processes, but also on greater or even global scales. These existing apparent "links" between climate variables measured at places located hundreds of kilometers apart from each other are referred to as *teleconnections*.

The American Meteorological Society defines teleconnections as

1. *"a linkage between weather changes occurring in widely separated regions of the globe"* and

2. *"a significant positive or negative correlation in the fluctuations of a field at widely separated points [...] most commonly applied to variability on monthly and longer timescales; The name refers to the fact that such correlations suggest that information is propagating between the distant points through the atmosphere"* [13].

A lot of the well-studied teleconnections (e. g. the emblematic linkage between the surface-level air pressures at Darwin, Australia and Tahiti) are related with the El Niño-Southern Oscillation (ENSO). It has been shown, for example, that a recognizable negative correlation between the ENSO cycle (measured using positive mean SST anomalies) and vegetation vigour (represented by the NDVI) in the south-east USA exists [2], or that precipitation and hence the NDVI in sizeable areas of Africa, South America and Australia were associated with tropical Pacific SST anomalies [4]. Also other (time-lagged) teleconnections between ENSO and climate parameters such as precipitation, temperature and air pressure have been documented for several regions of the world [14, 15]. Consequently, one might be able to exploit relationships like that for short and mid-term early warning systems and forecast models.

2

OBJECTIVES

The goal of this thesis is to provide a basis for future studies regarding short and mid-term NDVI forecasts based on SST data. We want to test if RQA is a suitable method to identify, characterize and quantify possible (time-lagged) relationships between the NDVI and the SST based on world-wide data. Forecast or prediction models of any kind are not part of this work, but might be the focus of future research projects.

Part II

MATERIAL

3 SST DATA

3.1 DATA SOURCE

A valuable source for the Sea Surface Temperature data was found in the Advanced Very High Resolution Radiometer (AVHRR) Pathfinder Version 5.0 project realized by the US National Oceanographic Data Center (NODC) and Group for High-Resolution Sea Surface Temperature (GHRSST) available at

`http://pathfinder.nodc.noaa.gov`.

This data collection is described in detail by Casey et al. [16].

The NODC's Pathfinder 5.0 project provides daily, 5-day, 7-day, 8-day, monthly and yearly SST data from the years 1985 to 2006 downloadable for free on the internet; each observation is availably for both day- (10 am or 2 pm depending on the satellite) and nighttime. The datasets offer a ground resolution of up to approx. 4×4 km per pixel which leads to raster files consisting of 8192×4096 pixels covering nearly the whole earth surface. The data is available in the Hierarchical Data Format (HDF) in separate files. As georeferencing is not crucial for this work, any available projection information was neglected.

This work is based on the 7-day (weekly) SST data recorded during daytime. Figure 1 shows an example of the Pathfinder 5.0 SST data from summer 1985.

For each temporal resolution, the Pathfinder 5.0 project provides (amongst others) the following parameters:

"ALL-PIXEL" SST: The all-pixel SST files contain values for each ground pixel, including those contaminated with clouds or other sources of error. The SST value in each pixel location is an average of the highest quality AVHRR Global Area Coverage (GAC) observations available in each roughly 4 km bin. The SST values are stored in a 16-bit format and can be converted to °C using the formula

$$y = 0.075x - 3 \tag{1}$$

where x is the 16-bit and y the °C value.

OVERALL QUALITY FLAG: The Overall Quality Flag is a relative assignment of SST quality based on a hierarchical suite of tests. The Quality Flag varies from 0 to 7, with 0 being the lowest and 7 the highest quality. A detailed description of the applied quality tests is available by Kilpatrick et al. [17].

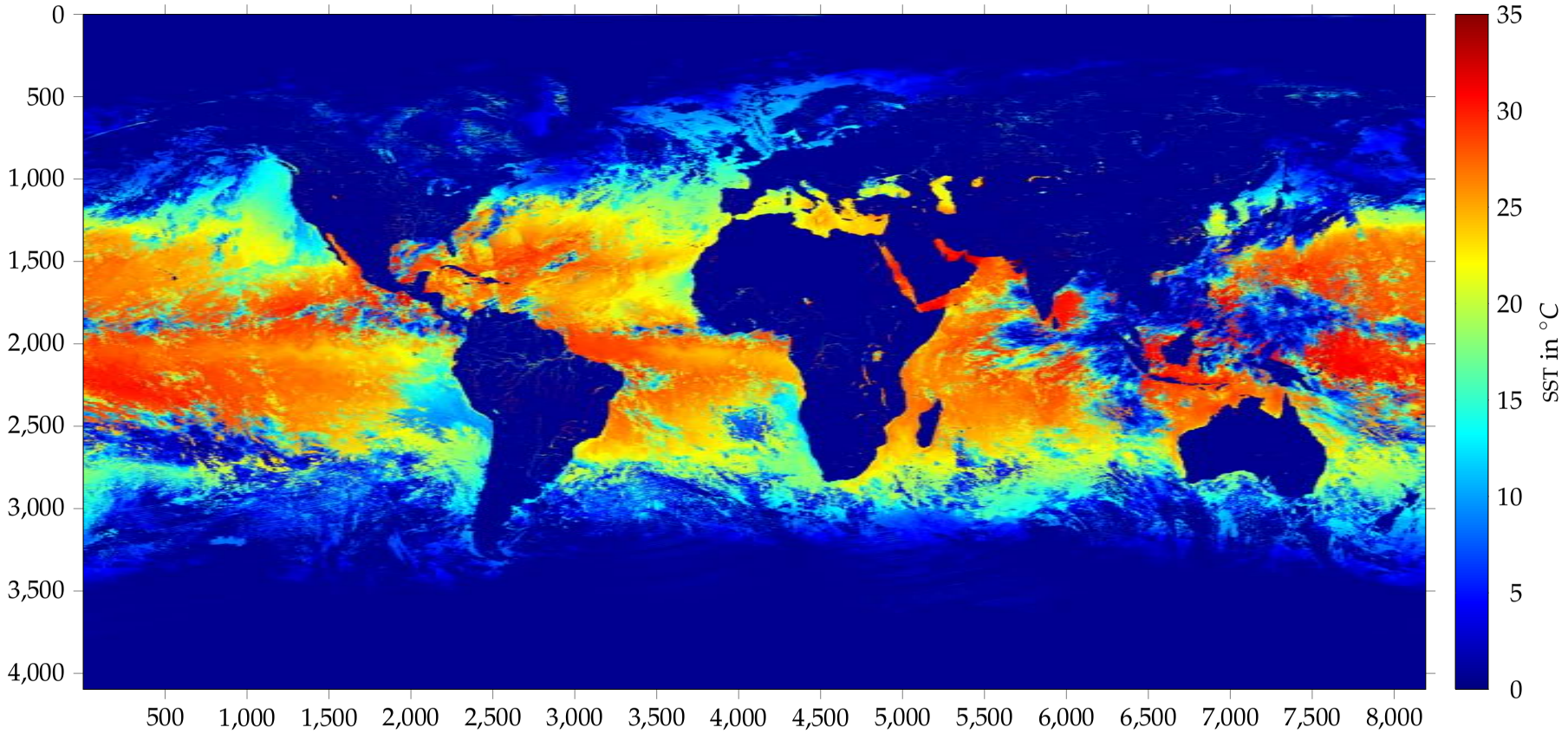


Figure 1: SST in °C during summer 1985

Additionally, a land-water mask which classifies each pixel as either a land or water pixel is part of the Pathfinder 5.0 dataset.

As a result of leap years and a change in week numbering during the recording period, a total of 1148 weekly SST values per pixel is available for the years 1985 to 2006 in seperate files.

3.2 MATLAB IMPORT

As a first preprocessing step, the HDF files were sorted chronologically and imported into MATLAB. Because of memory limitations, a seperate MATLAB variable file for each pixel row was created, resulting in a total number of 4096 files, each containing one row of 8192 pixels with 1148 SST observations, the same number of quality flags, and the corresponding pixel row of the land-water mask.

3.3 SMOOTHING, INTERPOLATION AND RESCALING

Because of leap years and a change in week numbering during the recording period, the number of values per year varies between 52 and 53. Additionally, as new years usually do not start with Mondays only, it would be difficult to compare the observations from one year with the ones of the others because they represent a different day of the year. Therefore, the complete SST dataset was rescaled and interpolated to daily values using a MATLAB implementation of a smoothing and interpolating algorithm developed by Eilers [18] based on the ideas of Whittaker [19]. This algorithm has often shown to perform well on remotely sensed time series [20, 21]. For the smoothing process, each value was assumed to represent Sundays during the years 1985 and 1989 and Wednesdays starting from 1990. Then, the missing (daily) values between those days were interpolated using the above-mentioned smooting algorithm with a smoothing parameter of $\lambda = 10^4$ and an order of differences of $d = 2$. These parameters were selected by visually inspecting sample plots (e. g. like Figure 2).

Since part of the SST values were of lower quality, only the ones with an Overall Quality Flag of 4 and above were used as input variables for the interpolation, the others were treated as missing values as well. This was achieved by assigning a weight of 1 to values with Overall Quality Flag ≥ 4 and a weight of 0 to the others (and also the missing days). After the applying the smoothing and interpolation algorithm, the extended dataset consisted of 8035 daily SST values for each pixel. Figure 2 shows an example of the smoothing and interpolation process, where the green circles represent SST values with a Quality Flag ≥ 4 which were therefore taken into account; all the other values were interpolated.

To rescale the dataset back to weekly values, the SST values of the 4^{th}, 11^{th}, 18^{th} and 25^{th} day of each month were extracted, resulting in

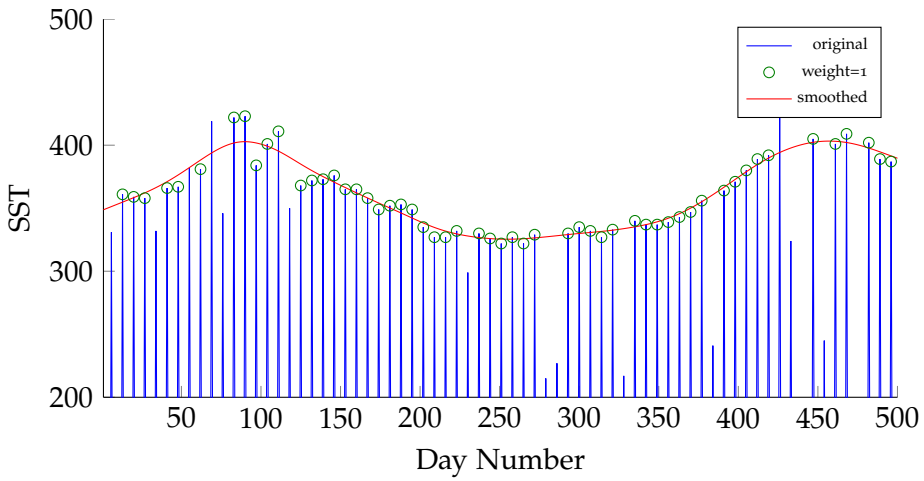


Figure 2: Smoothing and interpolation of the SST values

$22 \cdot 12 \cdot 4 = 1056$ SST observations per pixel for the 22 analyzed years (1985-2006). This procedure ensures comparability because now e. g. the 28^{th} observation of each year represents the same day of the year (in this case the 25^{th} of July).

3.4 ANOMALY AND Z-SCORE CALCULATION

Naturally, the SST is subject to seasonal fluctuations. For correlation analysis, only deviations from the "usual" values for a certain time of the year are of interest – the so-called *anomalies*. To eliminate the seasonal component, the mean values of each week $(1 \ldots 48)$ of the year $(1 \ldots 22)$ were calculated. This resulting "average year" was then deducted from the original values leading to a time series of SST anomalies.

As a last step, the z-scores z_t of each time series were calculated using the formula

$$z_t = \frac{x_t - \mu_w}{\sigma_w} \qquad w = 1 \ldots 48 \tag{2}$$

where μ_w and σ_w are the mean and the standard deviation of each week of the "mean year". Figure 3 shows the resulting values after each processing step.

The total resulting anomaly time series of each pixel has now a mean of $\mu_{total} = 0$ and standard deviation of $\sigma_{total} = 1$ and can thus be compared to other time series with different (original) value ranges – like NDVI anomalies (after they were preprocessed in the same way).

The MATLAB function (prep_sst) used for smoothing, interpolation and rescaling is described in Section A.1.

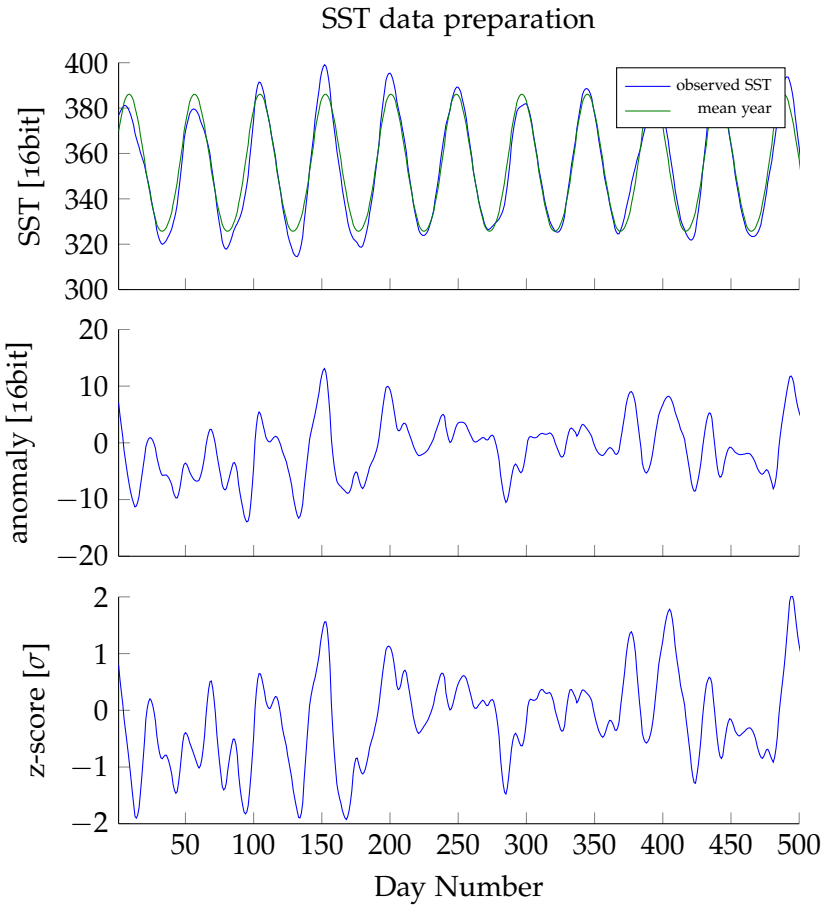


Figure 3: Anomaly and z-score calculation for SST values

3.5 SST DATA COMPRESSION

Due to memory and processing time issues, the amount of input SST data had to be reduced as a last step. To fasten the correlation calculation by a factor of 16, only every 4^{th} SST pixel (starting from the 2^{nd}) was selected. This compression results in a considerable loss of data (now 2048×1024 pixel), but was necessary to overcome difficulties with the amount of required storage space and processing time.

To even further reduce processing times and avoid unnecessary calculations, the polar regions were removed from the SST dataset (approx. starting from 70° N and 60° S). The final SST dataset now consisted of 2048×700 pixels (Figure 4).

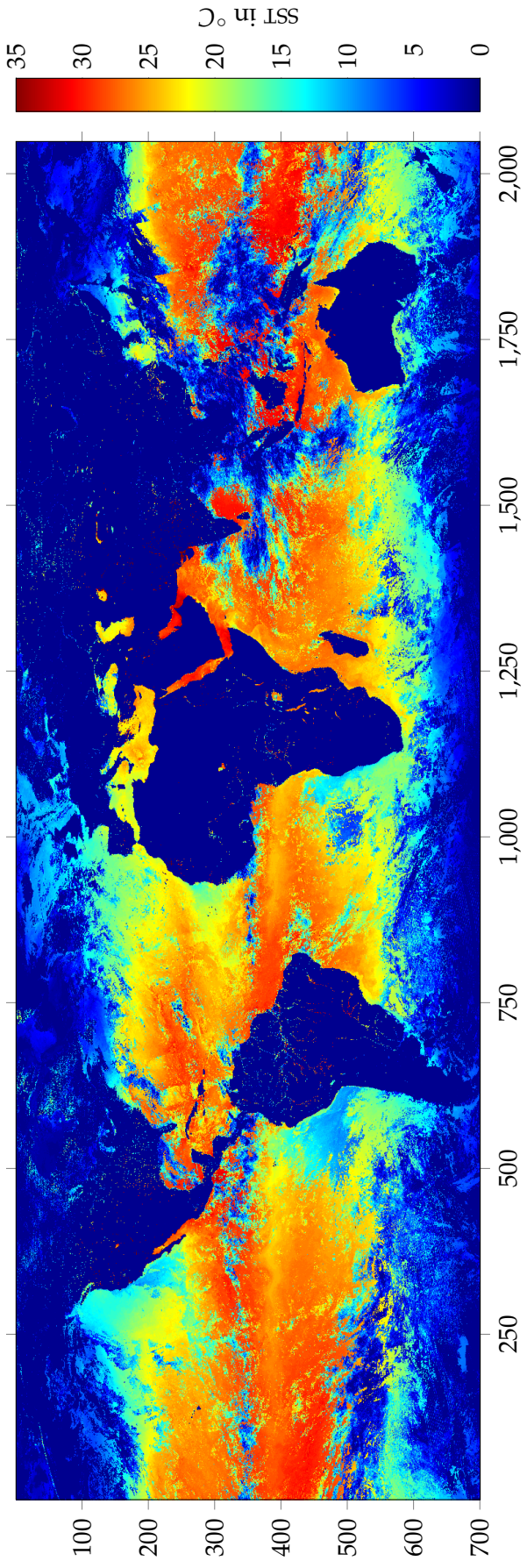


Figure 4: Final spatial extent and resolution of the SST dataset (showing the summer of 1985, in °C)

4 NDVI DATA

4.1 BASICS

The Normalized Difference Vegetation Index (NDVI) is a simple vegetation index which can easily be calculated using spectral reflectance data (in most cases remotely sensed). It was designed to provide a quick and easy overview of the spatial distribution of vegetated areas and has become one of the best-known and most-used vegetation indices in multi- and hyperspectral remote sensing.

The history of the NDVI dates back to the 1960s when researchers in the US were struggling with satellite-derived spectral signals confounded by differences in solar zenith angle across their extensive study area. Their approach to "normalize" the effects of the changing solar zenith angle resulted in the NDVI [22].

The NDVI is calculated using the formula

$$\text{NDVI} = \frac{\text{NIR} - \text{RED}}{\text{NIR} + \text{RED}} \tag{3}$$

where NIR is the reflectance in the near infrared wavelength range (typically ranging from approx. 0.7 to 1.0 µm) and RED the reflectance in the visible red range (approx. 0.6 to 0.7 µm). As Equation 3 shows, NDVI values can vary between -1 to 1, where higher NDVI values indicate higher vegetation density, because the chlorophyll of live green plants strongly absorbs visible solar radiation and tends to reflect most of the near-infrared light.

The NDVI has been found to correlate significantly with annual and monthly rainfall totals and has been used to predict crop yields several times [23, 24, 25]. This usefulness has led to a wide-spread usage of this index.

4.2 DATA SOURCE

Similar to the SST data, NDVI datasets are also available on the internet. For this project, data provided by the Global Inventory Modeling and Mapping Studies (GIMMS) project was used. It is available at

`http://glcf.umd.edu/data/gimms/`

and consists of satellite-derived NDVI data corrected for calibration, view geometry, volcanic aerosols, and other effects not related to vegetation change, covering the years from 1981 to 2006. A detailed

description of this dataset is available by Tucker et al. [26, 27], the correction method is described by Pinzon et al. [28].

The Global Inventory Modeling and Mapping Studies (GIMMS) project uses data obtained by the Advanced Very High Resolution Radiometer (AVHRR) sensor and provides (amongst others) global bi-weekly NDVI datasets availableas separate Tagged Image File Format (TIFF) files. For each month, there are two files, one representing the days from the 1^{st} to the 15^{th} and one from the 16^{th} day to the end of the month. The dataset is provided with a ground resolution of approximately 8×8 km leading to raster files with 4950×2091 pixels. For each temporal resolution, the GIMMS project provides the following parameters:

CORRECTED GIMMS NDVI: The NDVI values are corrected for residual sensor degradation/intercalibration differences, distortions caused by persistent cloud cover globally, solar zenithangle and viewing angle effects due to satellite drift, volcanic aerosols, missing data in the Northern Hemisphere during winter using interpolation due to high solar zenith angles, and low signal to noise ratios due to sub-pixel cloud contamination and water vapor. They are scaled to range from -1000 to 1000, while water pixels are assigned a value of -10000. To recover the original NDVI, the formula

$$y = \frac{x}{1000} \tag{4}$$

where x is the stored and y the original value, has to be applied to the data.

FLAG: A flag value is available for every NDVI observation and ranges from 0 to 6 with 0 and 1 meaning good values, 2 and 3 values retrieved from interpolation, 4 and 5 values retrieved from average seasonal profile and 6 pointing out missing values.

For the complete analyzed time period, a total of 528 NDVI values was available in separate files with corresponding flags. Figure 5 shows an example of the NDVI dataset from summer 1985.

4.3 MATLAB IMPORT

As a first preprocessing step, the original files were sorted chronologically and imported into MATLAB. Because of memory limitations, a seperate MATLAB variable file for each pixel row was created, resulting in a total number of 2091 files, each containing one row of 4950 pixels with 528 NDVI observations and the same number of flags.

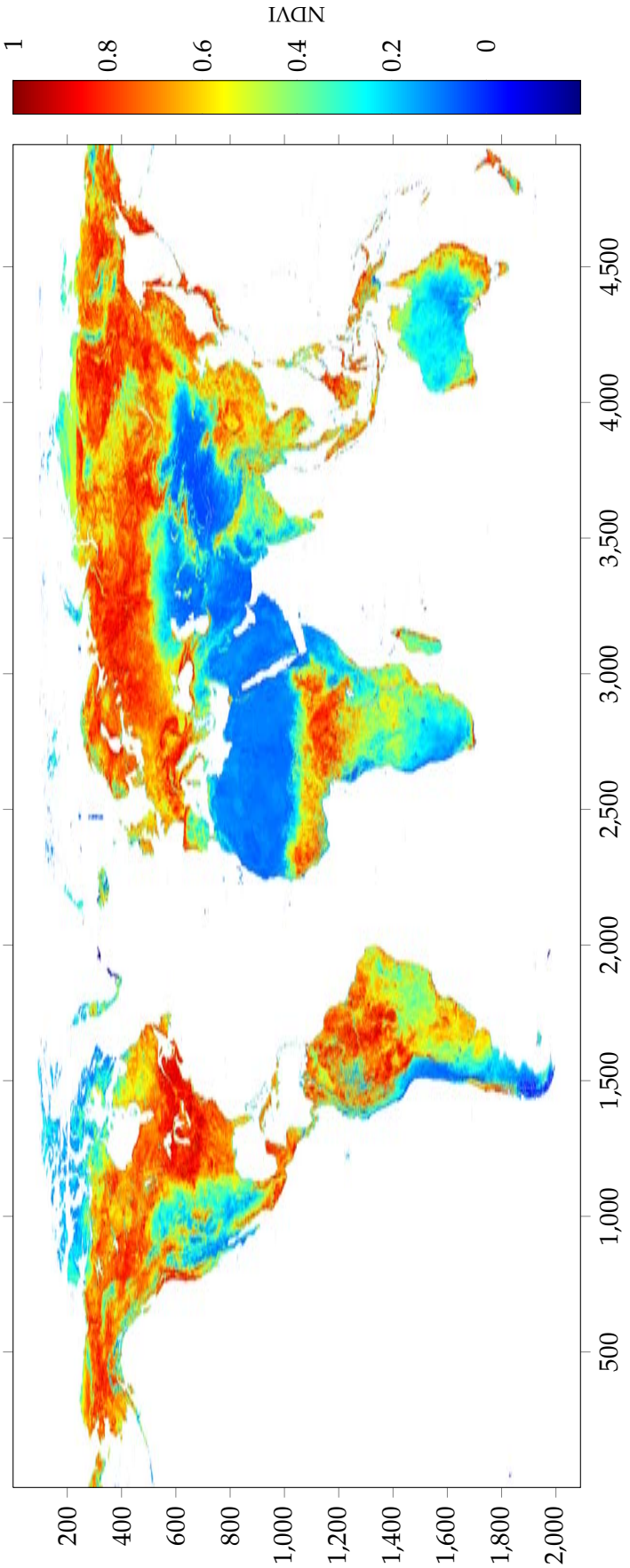


Figure 5: NDVI during summer 1985

4.4 SMOOTHING, INTERPOLATION AND RESCALING

As described in Section 3.1, the final SST anomaly dataset consists of pixels with 1056 weekly SST anomaly values, 48 per year for the 22 analyzed years. To allow correlation calculations, the bi-weekly NDVI dataset had to be interpolated to weekly values, too. To enable the use of the same smoothing algorithm by Eilers (see Section 3.3), the existing NDVI values were assumed to represent the 8^{th} and the 23^{rd} day of each month. Then the time series were interpolated and smoothed, this time using a smoothing parameter of $\lambda = 10^5$ and an order of differences of $d = 2$. After the applying the algorithm, the extended dataset consisted of 8035 daily NDVI values for each pixel.

To rescale the dataset back to weekly values, again the NDVI values of the 4^{th}, 11^{th}, 18^{th} and 25^{th} day of each month were extracted, resulting in $22 \cdot 12 \cdot 4 = 1056$ NDVI observations per pixel for the 22 analyzed years (1985-2006) – just as the SST dataset.

4.5 DATA CLASSIFICATION

To further reduce the amount of input data before the correlation calculation, the complete NDVI anomaly dataset was spatially classified using MATLAB implementations of the kmeans [29, 30] and the mean-shift [31, 32, 33] algorithms. The amount of classes was chosen in a way that each class consists of an average of 2500 pixels. With an input dataset of 2,810,691 land pixels (a size of 4950×2091 and approx. 30% land pixels), the classification resultes in 1135 classes and is shown in Figure 6. After that, the mean NDVI time series was calculated for each class by calculating the average NDVI value for each week.

4.6 ANOMALY AND Z-SCORE CALCULATION

Similar to the SST, also the NDVI is subject to seasonal variations. To eliminate the seasonal component, the mean values of each week $(1 \ldots 48)$ of the year $(1 \ldots 22)$ were calculated. This calculated "average year" was then deducted from the original values leading to a time series of SST anomalies.

As a last step, the z-score z_t of the time series of each NDVI class was calculated using Equation 2 from above.

The total resulting anomaly time series of each class has now a mean of $\mu_{total} = 0$ and standard deviation of $\sigma_{total} = 1$ and can thus be compared to other time series with different (original) value ranges – like SST anomalies.

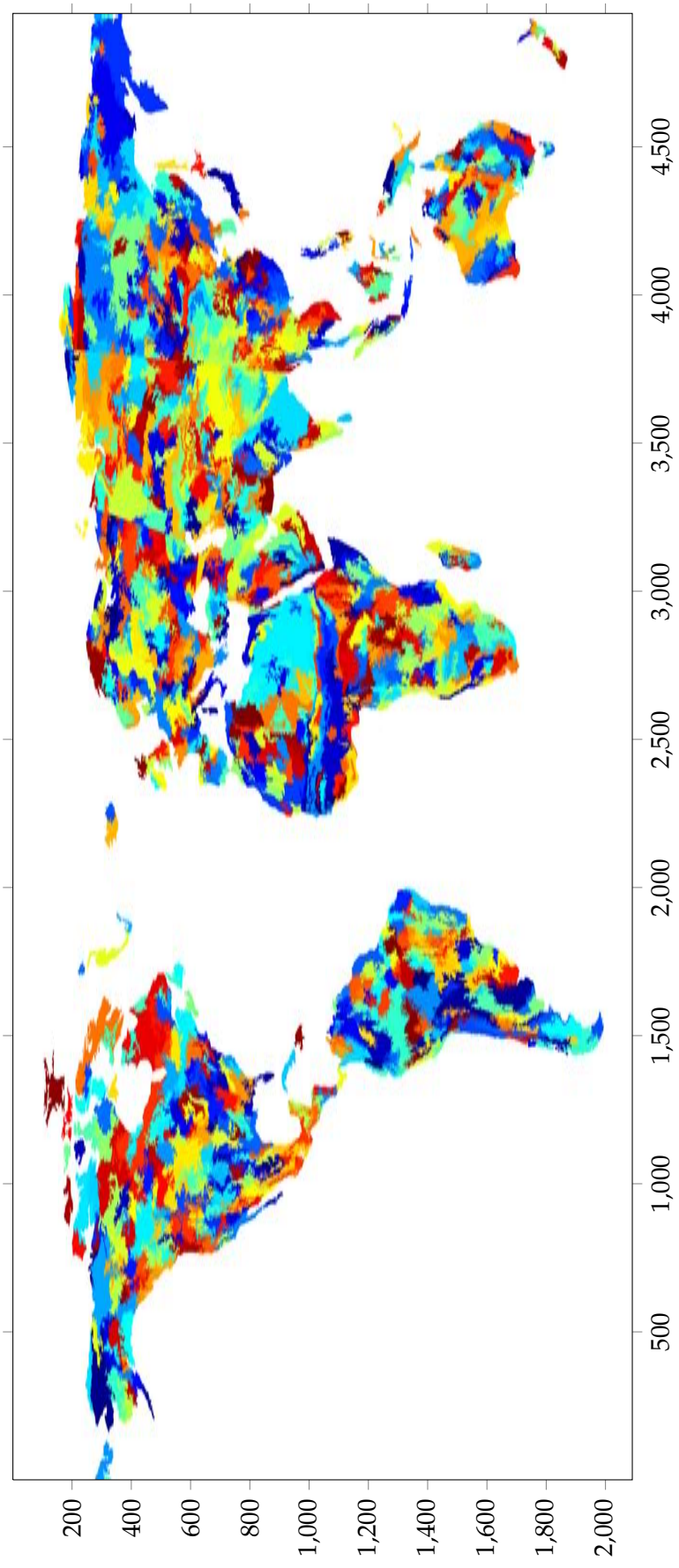


Figure 6: Result of NDVI classification (1135 classes)

Part III

METHODS

5 BASIC CONCEPT OF RECURRENCE PLOTS

5.1 INTRODUCTION

Recurrence Plots (RPs) are powerful tools to analyze, quantify and compare the behavior of dynamical systems [7]. Although they are well-known in various fields like earth sciences [8], chemistry [9], sociology [10] or mechanics [11], but apart from Li et al., who used RQA to measure determinism and predictability of NDVI series and its spatial patterns [12], they are barely used by the remote sensing community.

RPs are based on the formal concept of recurrences discovered by Poincaré in 1890 [34] and were introduced in 1987 by Eckmann et al. [35] to visualize recurrences of multidimensional dynamical systems [7]. Eckmann et al. described recurrence plots as "... an array of dots in a [...] square, where a dot is placed at i, j whenever x_j is sufficiently close to x_i" [35]. This statement can also be written as

$$RM_{i,j} = \begin{cases} 1: & x_i \approx x_j, \\ 0: & x_i \not\approx x_j, \end{cases} \qquad i, j = 1 \dots N \tag{5}$$

and results in a $N \times N$ Recurrence Matrix RM with values of 1 where the states of the system at times i and j are similar to each other and values of 0 where they are not.[1] So the basic concept of RPs is a series of pairwise tests where each of the N observed states x_i of the system is compared to all the others (and itself) x_j and the corresponding value of the RM is set either to 1 or 0 (and filled black or white in visual representations accordingly).

5.1.1 *Example 1 – Circular Motion*

Let's assume one is observing a circular motion with constant angular velocity and a period of $T = 10\,\mathrm{s}$. If its position (represented by the angle φ) is measured 100 times using a sampling rate of $\Delta t = 1\,\mathrm{s}$, the observed values will look similar to the ones shown in Table 1.

Now Equation 5 is used to calculate the Recurrence Matrix. Table 2 displays the calculation steps for $i = 1$ and $j = 1 \dots 21$ ($x_1 = 0$ has to be compared to x_1, x_2, ..., x_{21}), and Figure 7 shows the corresponding part of the RP.[2]

1 In literature (e. g. by Marwan et al. [7]), often the Heaviside step function $\Theta(k)$ is used. It results in 0 if $k < 0$ and in 1 otherwise.

2 The initial state of the system is at $t = 0$, but the indices i and j start with 1.

After applying Equation 5 to the whole data set, the resulting RP of this process looks like the one shown in Figure 8. Per convention, the two axes of a RP are faced rightwards and upwards [7] and are both time axes. As Equation 5 already states, the middle diagonal is always black in RPs because for $i = j$, each value is compared to itself, which always results in a black dot. The other diagonal lines parallel to the main diagonal reach from the lower left to the upper right side of the RP and appear every ten steps (i. e. every ten seconds). They show that after that time, the motion's evolvement in time is completely the same as it was 10 seconds before. In other words, the process has a recurrence period of $T_{\text{rec}} = 10\,\text{s}$.[3]

Table 1: Angles φ of a circular motion with $T = 10\,\text{s}$ for $t = 0 \ldots 13$ and $\Delta t = 1\,\text{s}$

t (s)	0	1	2	3	4	5	6
φ (rad)	0	0.2π	0.4π	0.6π	0.8π	π	1.2π
t (s)	7	8	9	10	11	12	13
φ (rad)	1.4π	1.6π	1.8π	$2\pi = 0$	0.2π	0.4π	0.6π
⋮							⋮

Table 2: Calculation of the RM at $i = 1$ and $j = 1 \ldots 21$

j	CALCULATION	$RM_{1,j}$	j	CALCULATION	$RM_{1,j}$
1	$0 \approx 0$	1	12	$0 \not\approx 0.2\pi$	0
2	$0 \not\approx 0.2\pi$	0	12	$0 \not\approx 0.4\pi$	0
3	$0 \not\approx 0.4\pi$	0	14	$0 \not\approx 0.6\pi$	0
4	$0 \not\approx 0.6\pi$	0	15	$0 \not\approx 0.8\pi$	0
5	$0 \not\approx 0.8\pi$	0	16	$0 \not\approx \pi$	0
6	$0 \not\approx \pi$	0	17	$0 \not\approx 1.2\pi$	0
7	$0 \not\approx 1.2\pi$	0	18	$0 \not\approx 1.4\pi$	0
8	$0 \not\approx 1.4\pi$	0	19	$0 \not\approx 1.6\pi$	0
9	$0 \not\approx 1.6\pi$	0	20	$0 \not\approx 1.8\pi$	0
10	$0 \not\approx 1.8\pi$	0	21	$0 \approx 2\pi \approx 0$	1
11	$0 \approx 2\pi \approx 0$	1	⋮		⋮

5.2 THRESHOLD ϵ

Unlike the example in Section 5.1.1, natural processes rarely return *exactly* to one of their former states. Eckmann et al. defined that "… a

3 T_{rec} equals the offset between the diagonals in the RP in Figure 8.

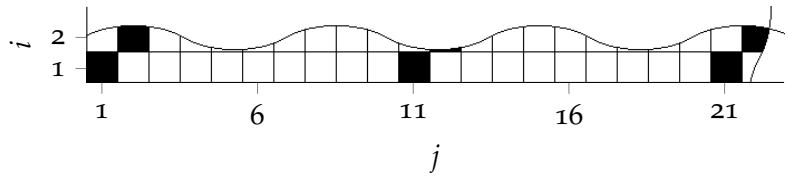


Figure 7: Part of the RP ($i = 1$ and $j = 1 \dots 21$) of a circular motion with $T = 10\,\mathrm{s}$ and $\Delta t = 1\,\mathrm{s}$

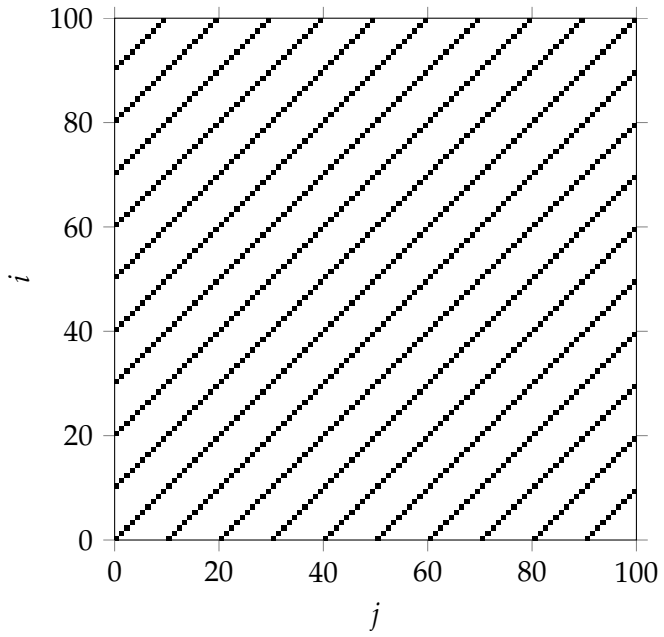


Figure 8: RP of a circular motion with $T = 10\,\mathrm{s}$ and $\Delta t = 1\,\mathrm{s}$

dot is placed at i, j whenever x_j is sufficiently close to x_i" [35], but as *sufficiently close* is not a very exact definition, at this point the first crucial parameter of RPs is introduced: The *threshold* ϵ. Including this parameter, Equation 5 changes to

$$RM_{i,j} = \begin{cases} 1: & \epsilon \geq |x_i - x_j|, \\ 0: & \epsilon < |x_i - x_j|, \end{cases} \qquad i, j = 1 \dots N. \tag{6}$$

This equation states that the value of the RM at position i, j is set to 1 if x_j and x_i are closer to each other than ϵ, otherwise to 0.
It is obvious that the parameter ϵ is crucial when working with RPs: A high threshold results in more black points and might render the RP useless, while a RP with a threshold chosen too low might reveal no information at all, because no values are to be found *sufficiently close* to any others.[4] Additionally, a threshold chosen too high might include points into the neighborhood which are simply consecutive points in the original time series or the reconstructed trajectory (for

4 For an example, see Figure 11.

trajectory reconstruction, see Section 5.6). This effect is called *tangential motion* and can greatly affect the appearance of a RP [7].

For the definition of ϵ, value range and scale of the analyzed time series need to be considered. Several ideas have been proposed in literature for the selection of ϵ [7]:

- a few percent of the maximum phase space diameter [36][5]
- not more than 10 % of the mean or maximum phase space diameter [37, 38]
- in a way that approx. 1 % of the total points are recurrence points [39]
- five times larger than the standard deviation of the observational noise [40]

Still, the appropriate choice of ϵ depends strongly on the nature of the studied system [7].

It is recommended to *normalize* the dataset before calculating RPs – in this case, the unit of ϵ is *standard deviations* σ. Normalization is especially important when comparing different parameters or when working with Cross Recurrence Plots (CRPs) – see Section 6.1. Generally, the threshold has to be chosen with caution in respect to the dynamics of the analyzed dataset.

5.2.1 *Example 2 – Sine Function*

In this example, we assume a dynamical process or system whose evolution in time can be described by the function $y_t = \sin t$. For $\Delta t = 1\,\mathrm{s}$, some observations are shown in Table 3 and Figure 9.

Table 3: Values of $y_t = \sin t$ for $t = 0 \ldots 11$ and $\Delta t = 1\,\mathrm{s}$

t	0	1	2	3	4	5
y_t	0.0000	0.8415	0.9093	0.1411	−0.7568	−0.9589
t	6	7	8	9	10	11
y_t	−0.2794	0.6570	0.9894	0.4121	−0.5440	−1.0000
⋮						⋮

For obtaining the *RM* for these observations, again each value has to be compare to all the other ones (and itself). With an assumed threshold of $\epsilon = 0.3$, we calculate the first row ($i = 1$, $j = 1 \ldots 21$) of the *RM* as shown in Table 4. This calculation is applied to the rest of

5 The phase space is a space which contains all possible states of a system – see Section 5.5

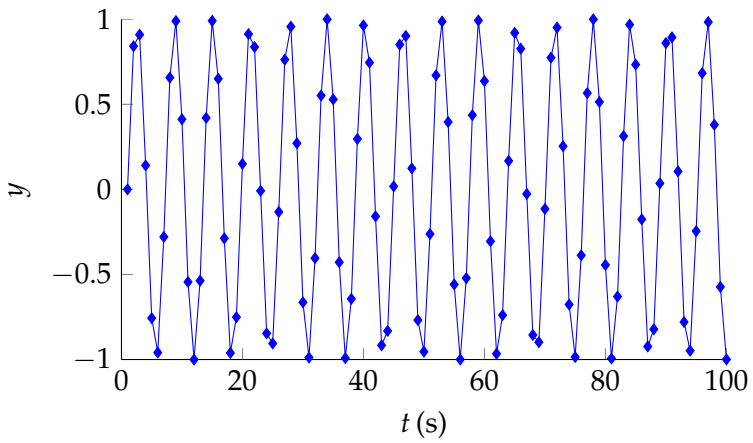


Figure 9: Plot of $y_t = \sin t$ for $t = 1 \dots 100$ and $\Delta t = 1\,\text{s}$

the observations too to completely fill the RM. If we want to display the RP, we just plot the RM and fill each position where $RM_{i,j} = 1$ black and $RM_{i,j} = 0$ white.

Figure 10 shows what the resulting Recurrence Plot looks like.

Table 4: Calculation of the RM at $i = 1$

j	CALCULATION	$RM_{1,j}$	j	CALCULATION	$RM_{1,j}$
1	$\lvert 0.000 - 0.000 \rvert \leq 0.3$	1	12	$\lvert 0.000 + 1.000 \rvert > 0.3$	0
2	$\lvert 0.000 - 0.842 \rvert > 0.3$	0	13	$\lvert 0.000 + 0.537 \rvert > 0.3$	0
3	$\lvert 0.000 - 0.909 \rvert > 0.3$	0	14	$\lvert 0.000 - 0.420 \rvert > 0.3$	0
4	$\lvert 0.000 - 0.141 \rvert \leq 0.3$	1	15	$\lvert 0.000 - 0.991 \rvert > 0.3$	0
5	$\lvert 0.000 + 0.757 \rvert > 0.3$	0	16	$\lvert 0.000 - 0.650 \rvert > 0.3$	0
6	$\lvert 0.000 + 0.959 \rvert > 0.3$	0	17	$\lvert 0.000 + 0.288 \rvert \leq 0.3$	1
7	$\lvert 0.000 + 0.279 \rvert \leq 0.3$	1	18	$\lvert 0.000 + 0.961 \rvert > 0.3$	0
8	$\lvert 0.000 - 0.657 \rvert > 0.3$	0	19	$\lvert 0.000 + 0.751 \rvert > 0.3$	0
9	$\lvert 0.000 - 0.989 \rvert > 0.3$	0	20	$\lvert 0.000 - 0.150 \rvert \leq 0.3$	1
10	$\lvert 0.000 - 0.412 \rvert > 0.3$	0	21	$\lvert 0.000 - 0.913 \rvert > 0.3$	0
11	$\lvert 0.000 + 0.544 \rvert > 0.3$	0	⋮		⋮

A pattern is clearly visible also in this RP, but unlike the one in Figure 8, the diagonals are not only faced from the lower left to the upper right, but also from the upper left to the lower right. This effect is a result of the characteristics of the sine function where a state *sufficiently close* recurs during both the decreasing and the increasing value range, thus the diagonals are faced in both directions.[6]

To visualize the influence of the threshold, Figure 11 shows RPs of this sample function calculated with different thresholds.

6 On the contrary, in the example in Section 5.1.1 the underlying function has only increasing ranges, so the diagonal lines are only faced in one direction.

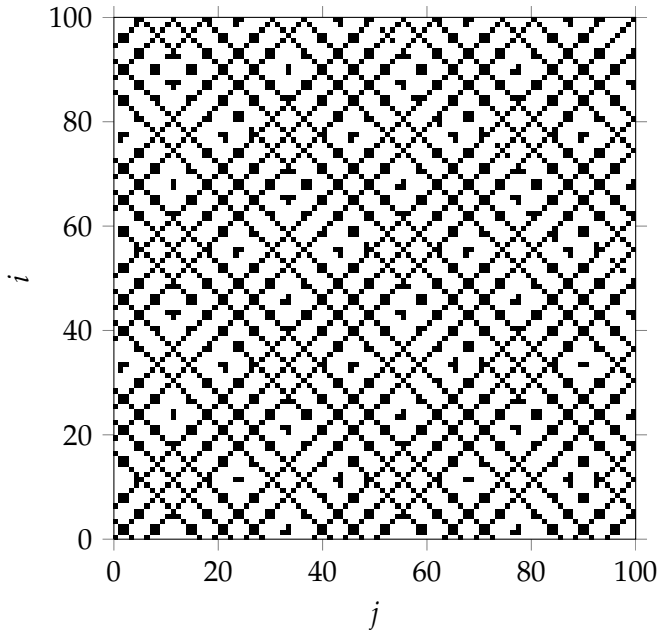


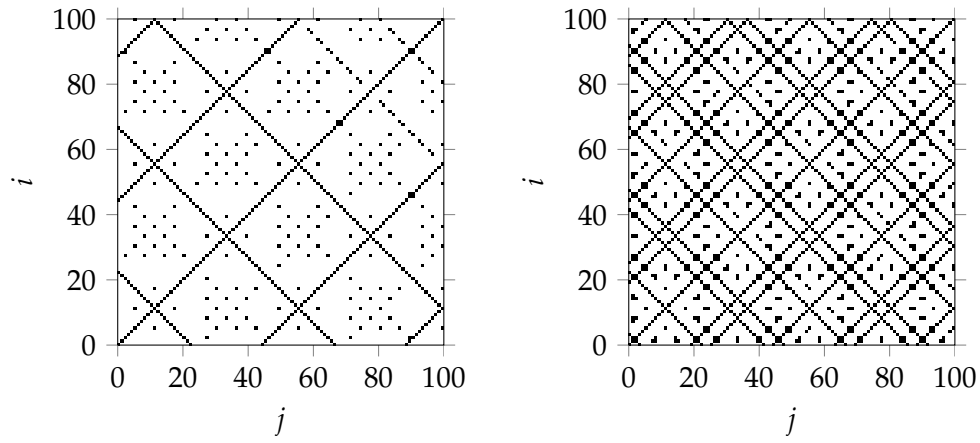
Figure 10: RP of $y_t = \sin t$ with $\epsilon = 0.3$

5.3 BASIC CHARACTERISTICS OF RECURRENCE PLOTS

In Figure 8, Figure 10 and Figure 11, some fundamental characteristics of RPs can be identified:

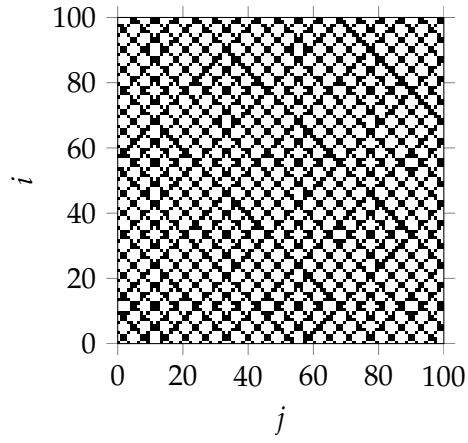
- RPs do always have a black main diagonal, the so-called Line Of Identity (LOI), because along this line $i = j$ and thus each value is compared to itself.
- RPs are always symmetrical with respect to the LOI ($RM_{i,j} \equiv RM_{j,i}$). It makes no difference which of the axes is plotted rightwards and which upwards.
- Diagonal lines are of particular interest in RPs, as they reveal time periods where the process evolves similar to a former/future period over more than one time step.
- The normal distance between diagonal lines can provide information about a possible recurrence period T_{rec} of the analyzed system.
- The selection of a suitable treshold ϵ is crucial when working with RPs, as it heavily influences their looks.

Having these characteristics in mind, a visual interpretation of a RP might lead to a considerable amount of information about the dynamics of the studied process or system. However, not only visual interpretation is possible, but a variety of possibilities to quantify recurrence characteristics of the underlying system based on RPs was

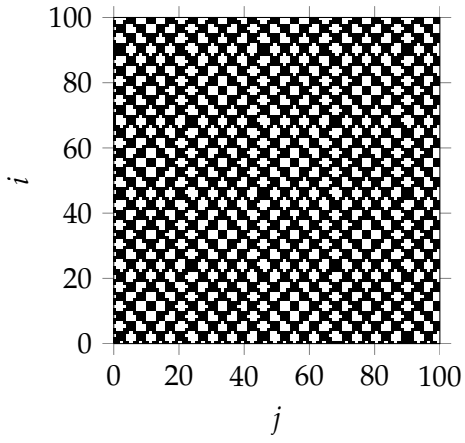


(a) $\epsilon = 0.05$

(b) $\epsilon = 0.2$



(c) $\epsilon = 0.5$



(d) $\epsilon = 0.9$

Figure 11: RPs of $y_t = \sin t$ with different thresholds

developed – the Recurrence Quantification Analysis (RQA) – see Chapter 7.

5.4 SYSTEMS WITH MORE THAN ONE PARAMETER

In the examples in Section 5.1.1 and Section 5.2.1, the time-evolution laws of the studied dynamical processes were known beforehand, and so it was clear that one single parameter completely described the states of these systems. To know if a certain observed state is a recurrence or not, we just needed to compare each observed value to the other ones. But for most natural processes and systems, the exact time-evolution law and/or the total number of parameters involved are either not known, or only one or a few of the parameters are (easily) measurable. In other words, we try to observe the state of a system by measuring one of its parameters, but we know that the system's state is defined by more. Obviously, a recurrence of (only) the

measured parameter to one of its former values does not necessarily evoke a recurrence of the whole system.

5.4.1 *Example 3 – Daily Air Temperature Oscillation*

Consider the system "weather", whose state is (amongst many others) defined by the parameter air temperature. If we want to find recurrences in a temperature profile (i. e. the temperature measured on e. g. an hourly basis), we may find the same temperatures during sunrise in the morning and after sunset in the evening. Although the temperature values alone would indicate recurrences, the system as a whole is clearly at completely different states at those times, because additional, unknown parameters (which have different values at both times) are involved (in this case they might be air pressure, humidity etc.).

5.5 MULTIDIMENSIONAL PHASE SPACES

A phase space is a space in which all possible states of a system are represented, with each possible state of the system corresponding to one unique point in the phase space [41]. Its dimension d ($d \in \mathbb{R}$) depends on the number of parameters involved (in the examples in Section 5.1.1 and Section 5.2.1, it was 1 in both cases; in Section 5.4.1 it was unknown). As the system evolves over time, its state follows a line (*trajectory*) between imaginary "points" (states of the system) in this phase space.

The real strength of Recurrence Plots is revealed when studying complex dynamical systems or processes whose states are defined by several parameters. As natural processes often have a lot of variables but not all of them are measurable, or the total number of variables and thus the phase space dimension is not known, the trajectory has to be reconstructed out of often only one observed variable – this is done using the time delay method (see Section 5.6). With RPs, even characteristics of systems with high-dimensional phase spaces can be visualized in two-dimensional plots – that's one of the main advantages of RPs.

To allow RPs to work with phase spaces with $d > 1$, the different parameters $x_1(t)$, $x_2(t)$, $x_3(t)$, $\dots x_d(t)$ that define the trajectory in the d-dimensional phase space are considered to form a vector $\vec{x}_t$ and Equation 6 is extended to

$$RM_{i,j} = \begin{cases} 1: & \epsilon \geq ||\vec{x}_i - \vec{x}_j||, \\ 0: & \epsilon < ||\vec{x}_i - \vec{x}_j||, \end{cases} \qquad i, j = 1 \dots N. \tag{7}$$

To decide whether two values are sufficiently close to each other to result in a black dot in the RP, it is not enough to compare the absolute

difference between two values x_i and x_j with ϵ (because they are now vectors, $\vec{x}_i$ and $\vec{x}_j$), so $|\,.\,|$ is extended to a norm $||\,.\,||$ (see Section 5.7).

5.5.1 *Example 4 – Lorenz System*

A system with a three-dimensional phase space is for example the Lorenz system. It was introduced by Lorenz in 1963 as a simplified mathematical model for atmospheric convection [42] and serves as an excellent example for systems with multidimensional phase spaces in a lot of publications on RPs (e. g. [7, 35, 43]). The Lorenz System consists of three equations:

$$\begin{aligned} \frac{\mathrm{d}x_1}{\mathrm{d}t} &= \sigma(x_2 - x_1), \\ \frac{\mathrm{d}x_2}{\mathrm{d}t} &= x_1(\rho - x_3) - x_2, \\ \frac{\mathrm{d}x_3}{\mathrm{d}t} &= x_1 x_2 - \beta x_3. \end{aligned} \tag{8}$$

Using a MATLAB implementation [43] of these equations, the system parameters $\sigma = 10$, $\beta = \frac{8}{3}$ and $\rho = 28$ and the initial values $x_1 = 6$, $x_2 = 9$ and $x_3 = 25$, a plot (the phase space portrait) for $t = 0 \ldots 2000$ and $\Delta t = 0.02$ is shown in Figure 12.

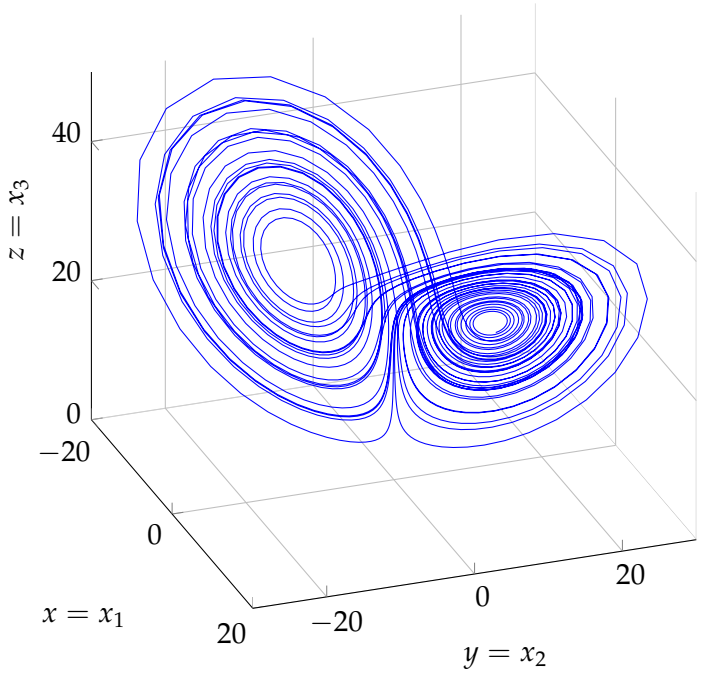


Figure 12: Phase space portrait of the Lorenz system with $\sigma = 10$, $\beta = \frac{8}{3}$, $\rho = 28$, $x_{1,0} = 6$, $x_{2,0} = 9$ and $x_{3,0} = 25$ for $t = 0 \ldots 2000$ and $\Delta t = 0.02$

5.6 TRAJECTORY RECONSTRUCTION – THE TIME DELAY METHOD

Let's assume we want to analyze a natural process which behaves like the Lorenz system, but the underlying equations are not known and only the variable x_1 was measured. Of course we can now construct a RP like in Section 5.1.1 and Section 5.2.1 just by using the observed values of x_1 as inputs. The resulting RP will show black dots for all the times t at which x_1 recurs close to a former value. But as the phase space portrait in Figure 12 shows, it is obvious that the RP would not at all reflect the characteristics of the system if x_2 and x_3 are not considered.

So the phase space trajectory has to be somehow reconstructed out of only the values of x_1. This is done by using the time delay method [7, 43] developed by Takens [44]:

$$\vec{x_t} = (u_t, u_{t+\tau}, u_{t+2\tau}, \ldots, u_{t+(m-1)\tau}) \tag{9}$$

where m is the embedding dimension and τ is the (index-based) time delay. It can be shown that if m is chosen in a way that $m > 2d + 1$ with d being the dimension of the original phase space, the reconstructed phase space trajectory is sufficient enough for subsequent data analysis [43].[7] If d is unknown, the embedding dimension should be increased until the majority of recurrence points in the resulting RP form diagonal structures [7].

It needs to be considered that depending on the embedding parameters m and τ, the number of points (i. e. states) on the trajectory decreases to

$$N_2 = N - \tau \cdot (m - 1) \tag{10}$$

where N is the length of the original and N_2 the length of the reconstructed time series. This means that the number of observations for analyses using the time delay method needs to be considerably higher than the one necessary for studying processes with one-dimensional phase spaces.

5.6.1 *Example 4 – Lorenz System (continued)*

If we now reconstruct the original trajectory of Figure 12 using a time delay of $\tau = 6$ and an embedding dimension of $m = 3$,[8] we get the resulting trajectory shown in Figure 13. Although the reconstructed trajectory differs from the original its main characteristics are still nearly the same and it is sufficient enough for the analysis of recurrences.

7 This means that both the original and the reconstructed trajectory can be considered to represent the same dynamical system in different coordinate systems [7]

8 In reality, at least a dimension of $m = 7$ would be required here as the original phase space dimension $d = 3$ and $m > 2d + 1$ needs to be fulfilled. To maintain the possibility of showing the reconstructed trajectory in a three-dimensional plot, in this case an embedding dimension of $m = 3$ was chosen.

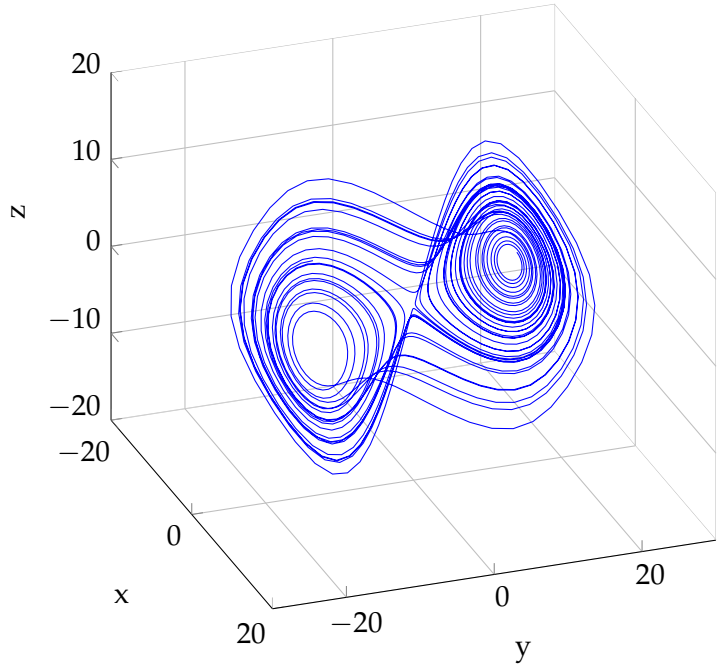


Figure 13: Phase space portrait of the reconstructed system with $\tau = 6$ and $m = 3$ for $t = 1 \dots 2000$ and $\Delta t = 0.02$

Figure 14 compares a RP calculated using the normalized original time series – the vectors $\vec{x}_t$ are formed out of $(x_{1,t}, x_{2,t}, x_{3,t})$ – with a RP calculated using a normalized reconstructed trajectory with $\tau = 6$ and $m = 7$.

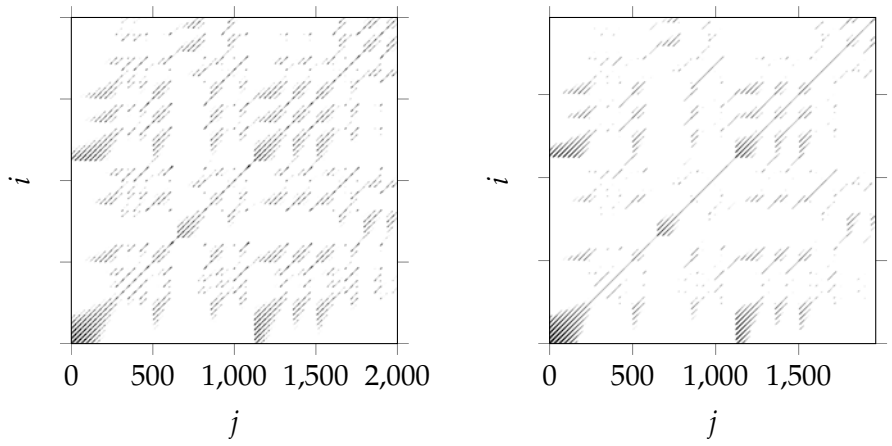
Despite the fact that the RP in Figure 14b has fewer values than the one in Figure 14a (because of Equation 10 – see there), the general characteristics of the two RPs look rather similar.

5.7 NEIGHBORHOOD – NORMS

The time delay method does not just take the *current* value u_t into account, but also the *future* values $u_{t+\tau}, u_{t+2\tau}, \dots, u_{t+(m-1)\tau}$. In case of $\tau = 2$ and $m = 3$, the vector $\vec{x}_t$ has three components and consists of the values $(u_{t,} u_{t+2}, u_{t+4})$. This vector is then compared to the other vectors formed at each time t. In practice, this means that not only a single value x_i has to be sufficiently close to another one x_j to result in a black dot in the $RM_{i,j}$, but also the *later* value pairs at (in case of $\tau = 2$ and $m = 3$) x_{i+2}, x_{j+2} and x_{i+4}, x_{j+4} have to lie in the neighborhood of each other. How exactly this neighborhood condition is determined, depends on the selected *norm*.

To illustrate the different norms, a small time range of the (original) first parameter x_1 (which was later used for the reconstruction) from Section 5.5.1 and Section 5.6.1 was extracted and is shown in Table 5.

Now the time delay method is used to reconstruct the phase space trajectory using the values x_t for $t = 200 \dots 229$ (due to Equation 10,



(a) RP based on the original time series, with $\epsilon = 0.3\sigma$

(b) RP based on the reconstructed trajectory with $\tau = 6$, $m = 7$ and $\epsilon = 0.3\sigma$

Figure 14: RPs of the Lorenz System based on the normalized original and reconstructed phase space trajectory

the number reduces to 225), a time delay of $\tau = 4$ and an embedding dimension $m = 2$.[9] Table 6 and Figure 15 show the resulting reconstructed trajectory.

To determine if two specific points on the trajectory at $t_1 = i = 200$ and $t_2 = j = 218$ for $\epsilon = 5$ form a recurrence point at $RM_{i,j}$ or not, the vectors $\vec{x_{200}} = (x_{200}, x_{204})$ and $\vec{x_{218}} = (x_{218}, x_{222})$ have to be compared. This can be done using one of the following norms:

- Maximum Norm (L_∞-Norm): This norm calculates the absolute differences between the component of the vectors $\vec{x}_i$ and $\vec{x}_j$ and checks if the biggest difference is less than or equal to the threshold ϵ:

$$max(|x_i - x_j|, |x_{i+\tau} - x_{j+\tau}|, \ldots, |x_{i+(m-1)\tau} - x_{j+(m-1)\tau}|) \leq \epsilon \tag{11}$$

  In the case of the example for $i = 200$, $j = 218$ and $\epsilon = 5$:

$$\begin{aligned} max(|9.4617 - 6.8331|, |6.2511 - 9.9336|) &\leq 5 \\ max(2.6286, 3.6825) &\leq 5 \\ 3.6825 &\leq 5 \end{aligned}$$

- Euclidean Norm (L_2-Norm): This norm calculates the euclidean distance for the vectors $\vec{x}_i$ and $\vec{x}_j$ and checks if it is less than or equal to the threshold ϵ:

$$\sqrt{(x_i - x_j)^2 + (x_{i+\tau} - x_{j+\tau})^2 + \cdots + (x_{i+(m-1)\tau} - x_{j+(m-1)\tau})^2} < \epsilon \tag{12}$$

9 Again, the embedding dimension of $m = 2$ is chosen to be able to show the trajectory in a 2-dimensional plot. In reality, a higher dimension would be necessary.

Table 5: Parameter x_1 for $t = 200 \ldots 229$

t	VALUE	t	VALUE	t	VALUE
200	9.4617	210	4.3187	220	8.2726
201	8.5944	211	4.3445	221	9.0869
202	7.7438	212	4.4550	222	9.9336
203	9.9529	213	4.6475	223	10.7741
204	6.2511	214	4.9208	224	11.5553
205	5.6554	215	5.2749	225	12.2124
206	5.1727	216	5.7111	226	12.6749
207	4.8029	217	6.2305	227	12.8790
208	4.5416	218	6.8331	228	12.7818
209	4.3825	219	7.5163	229	12.3744

In the case of the example for $i = 200$, $j = 218$ and $\epsilon = 5$:

$$\begin{aligned} \sqrt{(9.4617 - 6.8331)^2 + (6.2511 - 9.9336)^2} &\leq 5 \\ \sqrt{6.9095 + 13.5608} &\leq 5 \\ 4.5244 &\leq 5 \end{aligned}$$

- Minimum Norm (L_1-Norm): This norm calculates the absolute differences between the component of the vectors $\vec{x}_i$ and $\vec{x}_j$ and checks if the sum of them is less than or equal to the threshold ϵ:

$$|x_i - x_j| + |x_{i+\tau} - x_{j+\tau}| + \cdots + |x_{i+(m-1)\tau} - x_{j+(m-1)\tau}| \leq \epsilon \quad (13)$$

In the case of the example for $i = 200$, $j = 218$ and $\epsilon = 5$:

$$\begin{aligned} |9.4617 - 6.8331| + |6.2511 - 9.9336| &\leq 5 \\ 2.6286 + 3.6825 &\leq 5 \\ 6.3111 &\not\leq 5 \end{aligned}$$

Obviously, the resulting RM for $\epsilon = 5$ would have the value 1 (i. e. show a black dot) at position $i = 200$ and $j = 218$ if the maximum (L_∞) or the euclidean (L_2) norm was chosen; a selection of the minimum (L_1) norm would result in no recurrence point at this position.

Figure 15 illustrates the sizes and shapes of the three norms in the 2-dimensional case.[10] It can be seen that the maximum (L_∞) norm results in the most recurrence points while the minimum (L_1) norm is the most restrictive.

10 Obviously, the use of these embedding parameters ϵ, m and τ would include a lot of *consecutive* points on the trajectory in the neighborhood (*tangential motion)*. This is because $m = 2$ was chosen to be able to plot the trajectory in a 2-dimensional plot; in reality, at least $m = 7$ would be necessary here, which would decrease this effect.

Table 6: Coordinates of the reconstructed trajectory $x = x_t$ and $y = x_{t+\tau}$

	X	Y		X	Y
t	x_t	$x_{t+\tau}$	t	x_t	$x_{t+\tau}$
200	9.4617	6.2511	213	4.6475	6.2305
201	8.5944	5.6554	214	4.9208	6.8331
202	7.7438	5.1727	215	5.2749	7.5163
203	6.9529	4.8029	216	5.7111	8.2726
204	6.2511	4.5416	217	6.2305	9.0869
205	5.6554	4.3825	218	6.8331	9.9336
206	5.1727	4.3187	219	7.5163	10.7741
207	4.8029	4.3445	220	8.2726	11.5553
208	4.5416	4.4550	221	9.0869	12.2124
209	4.3825	4.6475	222	9.9336	12.6749
210	4.3187	4.9208	223	10.7741	12.8790
211	4.3445	5.2749	224	11.5553	12.7818
212	4.4550	5.7111	225	12.2124	12.3744

Other possibilities for the interpretation of the neighborhood size ϵ (e. g. fixed amount of neighbors) can be found in the literature [7, 35].

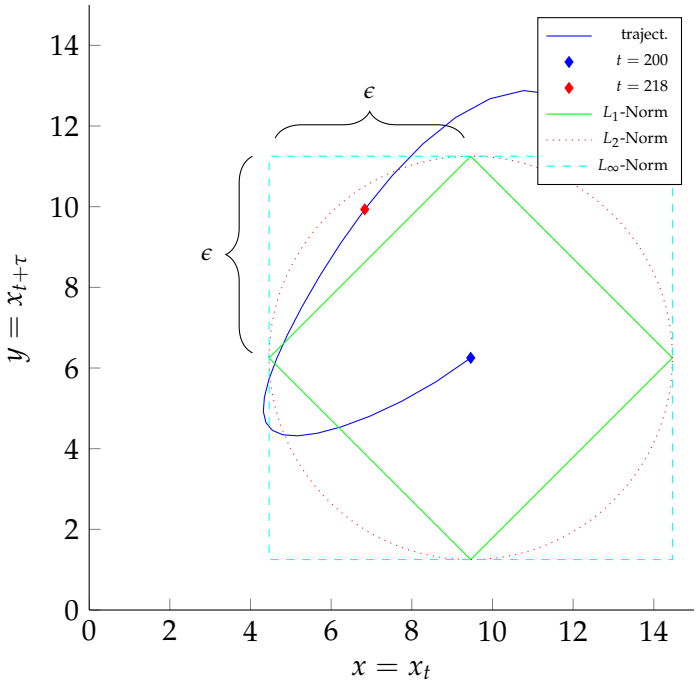


Figure 15: Reconstructed trajectory $x = x_t$ and $y = x_{t+\tau}$ for $t = 200 \ldots 225$ and sizes and shapes of the different norms at $t = 200$

6 BASIC CONCEPT OF CROSS RECURRENCE PLOTS

6.1 INTRODUCTION

Cross Recurrence Plots (CRPs) can be considered the bivariate extension of Recurrence Plots [7] and can be used to analyse the relationship of two different systems by comparing their states [45]. Analogously to RPs (Equation 7), CRPs consist of pairwise tests between the states of two different systems, and a Cross Recurrence Matrix CRM is calculated accordingly:

$$CRM_{i,j} = \begin{cases} 1: & \epsilon \geq ||\vec{x}_i - \vec{y}_j||, \\ 0: & \epsilon < ||\vec{x}_i - \vec{y}_j||, \end{cases} \qquad i = 1 \dots N,\ j = 1 \dots M \tag{14}$$

The basic concept of CRPs is similar to the one of RPs (e. g. embedding parameters, trajectories in phase space, time delay method, neighborhood – see Section 5.1), but there are some differences:

- As the two time series do not need to have the same length ($N \neq M$ is possible), the resulting Cross Recurrence Matrix CRM is not necessarily a square matrix.
- The two axes of the CRP are not interchangeable, as the time of one of the processes is plotted on the rightwards-faced and the one of the other one on the upwards-faced axis.
- The CRP does not have a black main diagonal (LOI) in most cases, because likely $\vec{x}_i \neq \vec{y}_i$ and $\vec{x}_j \neq \vec{y}_j$.

Another important point that needs to be considered when working with Cross Recurrence Plots is that both processes need to be represented in the same phase space, because a CRP looks for those times when a state of the first system recurs to one of the other system [7]. If the time delay method is used for the trajecotry reconstruction, the embedding parameters are the same for both processes. Also only one value for the treshold ϵ can be chosen; so either the two observed processes are similar in value range and dynamics, or the data needs to be normalized. Only this procedure ensures comparabilty between the values of the different processes.

Similar to RPs, diagonal structures are of particular interest in CRPs, too. They reveal times at which the first process evolved in a similar way the other process did at the same or another time. Also for example nonlinear changing of time scales can be clearly identified (*bowed* lines in the CRP). A visual interpretation of a CRP can provide

a lot of information about a possible relationship between the two processes (e. g. correlation, time lag/shift, etc.).

CRPs allow the detection of nonlinear similarities in short an non-stationary time series with high noise levels, as they often occur e. g. in life or earth sciences [7]. Other extensions of RPs (e. g. the Joint Recurrence Plot or the Order Matrix) can be found in literature.

6.1.1 *Example 5 – Two Sine Functions with changing Frequencies*

To illustrate the principle of CRPs, an example similar to the one created by Marwan et al. [7] is shown here. Consider the two sine functions $f_t = \sin t$ and $g_t = \sin t^2$ ($\Delta t = 0.01$) as shown in Figure 16. The second one differs from the first by the quadratic transformation $t_2 = t^2$, but they still are clearly related.

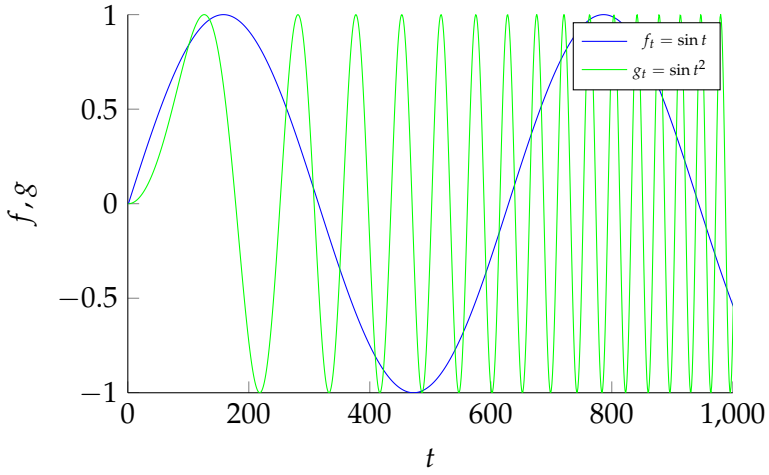


Figure 16: Sine functions $f_t = \sin t$ and $g_t = \sin t^2$ with $\Delta t = 0.01$

The CRP ($m = 1$, $\epsilon = 0.1\sigma$, L_∞-Norm) for these two (normalized) functions (Figure 17) clearly shows the nonlinear relationship between the two processes (undetected by linear correlation coefficients, e. g. Pearson's $r^2 = 0.0894$).

6.2 POSITIVE AND NEGATIVE-SIGNED TRAJECTORIES

Equation 14 checks if the trajectory points $\vec{x}_i$ and $\vec{y}_j$ are close to each other, which allows to check for positive correlation of the two studied processes. To be able to look for negative correlation/anti-correlation too, Equation 14 can be altered to

$$CRM^{-}_{i,j} = \begin{cases} 1: & \epsilon \geq ||\vec{x}_i + \vec{y}_j||, \\ 0: & \epsilon < ||\vec{x}_i + \vec{y}_j||, \end{cases} \qquad i = 1 \dots N,\ J = 1 \dots M. \quad (15)$$

This means that for the data series of two different processes, two CRPs can be calculated: One using Equation 14 to check for positive

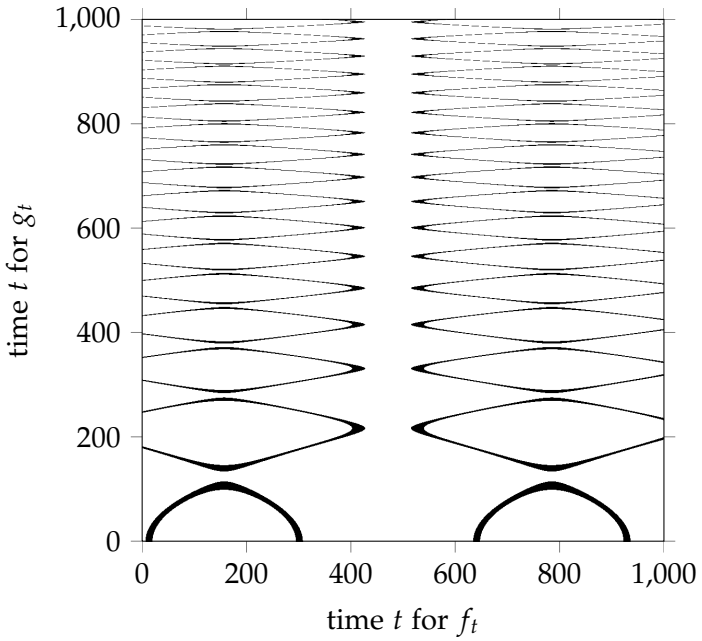


Figure 17: CRP with $m = 1$, $\epsilon = 0.1\sigma$ and L_∞-norm of the (normalized) functions $f_t = \sin t$ and $g_t = \sin t^2$ with $\Delta t = 0.01$

correlation and a second one based on Equation 15 to find a possible anti-correlation. To distinguish between the two resulting CRMs (or RMs), the superscript indices $+$ and $-$ are added – like $RM^{+}_{i,j}$ or $CRM^{-}_{i,j}$.

7 RECURRENCE QUANTIFICATION ANALYSIS

As mentioned above, visual interpretation of Recurrence Plots and Cross Recurrence Plots can tell a lot about the nature and relationship of the analyzed process(es). However, in order to go beyond the visual impressions yielded by RPs and CRPs, several measures of complexity which quantify the small scale structures, have been proposed [46, 38, 47] and are known as *Recurrence Quantification Analysis (RQA)* [45]. Three of the many measures proposed in literature will be used in this work: The Recurrence Rate RR, the Determinism DET and the Average Diagonal Line Length L.

7.1 RECURRENCE RATE

The Recurrence Rate RR is the simplest RQA measure. It corresponds to the percentage of recurrence points in relation to the total number of points; it therefore measures the recurrence point density. It is calculated using the formula

$$RR = \frac{1}{N^2} \sum_{i,j=1}^{N} RM_{i,j}. \tag{16}$$

The same applies for CRPs; here, the RR is sometimes referred to as the cross correlation sum

$$RR = CC_2 = \frac{1}{N \cdot M} \sum_{i=1\ldots N}^{j=1\ldots M} CRM_{i,j}. \tag{17}$$

7.2 DETERMINISM

The Determinism DET is the ratio of recurrence points that form a diagonal structure with a minimum length l_{min} to all recurrence points. Highly correlated processes cause longer diagonal lines: In case of RPs, if a process often returns to a former state and evolves in time similar to the way it did before, the number and length of diagonal structures increases, and with that, the predictability and the DET of the process; In case of CRPs, if the two processes run in parallel for a certain period (also possibly shifted in time), the same applies. DET is calculated using the formula

$$DET = \frac{\sum_{l=l_{min}}^{N} l \cdot D(l)}{\sum_{l=1}^{N} l \cdot D(l)} \tag{18}$$

where $D(l)$ is the number of diagonals with the length l. Equation 18 clearly shows that for $l_{min} = 1$, $DET = 1$ because all recurrence points form "diagonals" with a length of at least 1. For $l_{min} > 1$, the resulting value of DET depends on the shape of the distribution of the lengths of the diagonals. While choosing l_{min}, it has to be taken into account that with a rising l_{min} the histogram can become very sparse and thus lead to a decreased reliabilty of DET [7].

When working with an embedding dimension of $m > 1$ and trajectory reconstruction using the time delay method (see Section 5.6), a diagonal with a length of l shows that a segment of the reconstructed trajectory is *sufficiently close* to another segment at a different time for l consecutive time steps. When looking at the original values used for the reconstruction, this means that for $l + (m-1) \cdot \tau$ time steps these values recur to those of another time; the closer this diagonal is to the center, the smaller is the difference between these two times.

Analogously, the DET can be calculated for CRPs.

7.3 AVERAGE DIAGONAL LINE LENGTH

The Average Diagonal Line Length L of diagonals with a length of at least l_{min} is the average time two segments of the trajectory are close to each other and can be interpreted as the mean prediction time [7]. It is calculated using the formula

$$L = \frac{\sum_{l=l_{min}}^{N} l \cdot D(l)}{\sum_{l=l_{min}}^{N} D(l)} \tag{19}$$

where $D(l)$ is the number of diagonals with the length l.

7.4 DIAGONAL-WISE MEASURES

As like in the chapters above, all these measures can be calculated for the whole RP or CRP, but also for each diagonal parallel to the LOI separately. For a certain line parallel to the main diagonal with a distance of τ, the diagonal-wise measures are called τ-RR, τ-DET and τ-L (or RR_τ, DET_τ and L_τ) accordingly (with $\tau = 0$ corresponding to the LOI).

Especially when working with CRPs, diagonal-wise measures can reveal a lot of information about a possible time-shifted correlation between the two processes. A high RR_τ means that the trajectory of the second process often visits the same phase space regions as the trajectory of the first one did τ time steps before. Additionally, a high DET_τ shows that with a time lag τ, the trajectories visit the same phase space regions *at least once* for more than l_{min} consecutive time steps and seldom, if ever, for less than l_{min} consecutive time steps.

To distuingish between RQA-calculations for both possible RPs/CRPs (positive or negative correlation), the superscript indices $+$ and $-$ are added to the measure, e. g. RR_τ^+, L_τ^-, DET^+ or RR^-.

In CRPs, a distance of τ between two diagonals represents a "shift" or time-lag between the two studied processes, where the direction is given by the order of the two variables.

Part IV

APPLICATION, RESULTS AND CONCLUSION

8 CORRELATION CALCULATION

The correlation calculation was carried out on the Vienna Scientific Cluster (VSC), a cluster computer located at and operated by the Technical University of Vienna[1]. After extensive testing and code optimizations, individual jobs were submitted to compare the anomalies of each NDVI class to every SST anomaly series. Additionally, we wanted to test if there is a difference if we just use NDVI data from the summer months June, July, August and September or from winter (i. e. from the months October, November, December and January) and set the rest to zero in the CRPs (see Figure 18). To calculate the CRPs , we chose the parameters threshold $\epsilon = 0.8$, dimension $m = 20$ and a time delay of 1, and calculated the diagonal-wise RQA indices DET_τ^+, DET_τ^-, RR_τ^+, RR_τ^-, L_τ^+ and L_τ^- with $l_{min} = 5$ for diagonals with time lags $\tau = 0 \ldots 48$ weeks (=one year; see below) between SST and NDVI. Doing this for the three cases *summer*, *winter* and *all*, this led to an amount of $6 \times 3 \times 49 = 882$ RQA indices for each SST-NDVI-pair and thus to approx. 1.03^{12} individual correlation calculations. The total processing time was about 350,000 core-hours, which decreased to around two weeks on the VSC thanks to parallel processing. The resulting dataset had a size of about 4 TB, and it took another few days to transfer it back to our local system from the VSC.

As explained in Section 5.1 ff, Cross Recurrence Plots include every possible time shift between two time series. Since we just want to see how the SST influences the NDVI, half of the CRPs (the half where the SST anomalies are compared to NDVI anomalies which occured earlier in time) are of no interest, and thus the RQA parameters were not calculated for these diagonals. Additionally, we did not consider the part of the CRPs representing correlations between events that happened more than 48 weeks (=one year) apart. So for most of the diagonals of the CRPs, we didn't calculate the RQA indices (see Figure 19) because we considered a correlation of events with a time lag with more than one year seems to be rather unlikely. As a result of the excluding most of the diagonals of and the seasonal NDVI value selection, RQA calculations are based on CRPs similar to Figure 20 for *summer* and *winter* and to Figure 19 for *all*.

The code used to calculate the RQA indices is shown and described in Section A.2.

1 More accurately on the VSC-1; additional information can be found at `http://www.vsc.ac.at`.

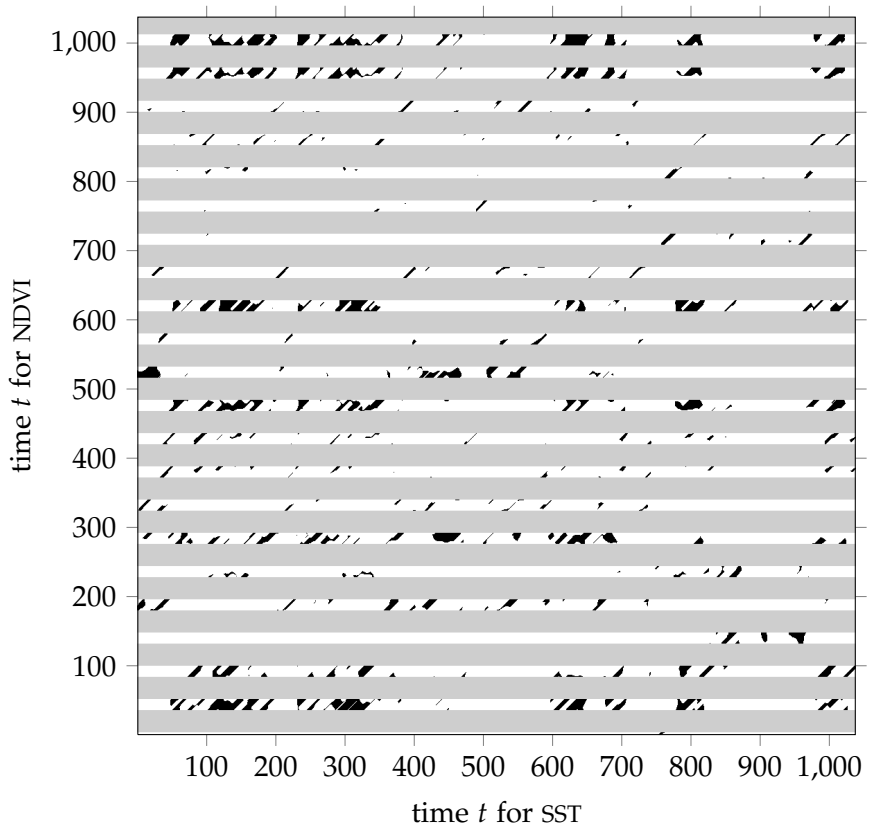


Figure 18: Just NDVI anomaly values from *winter* taken into account (other seasons marked gray) of a CRP

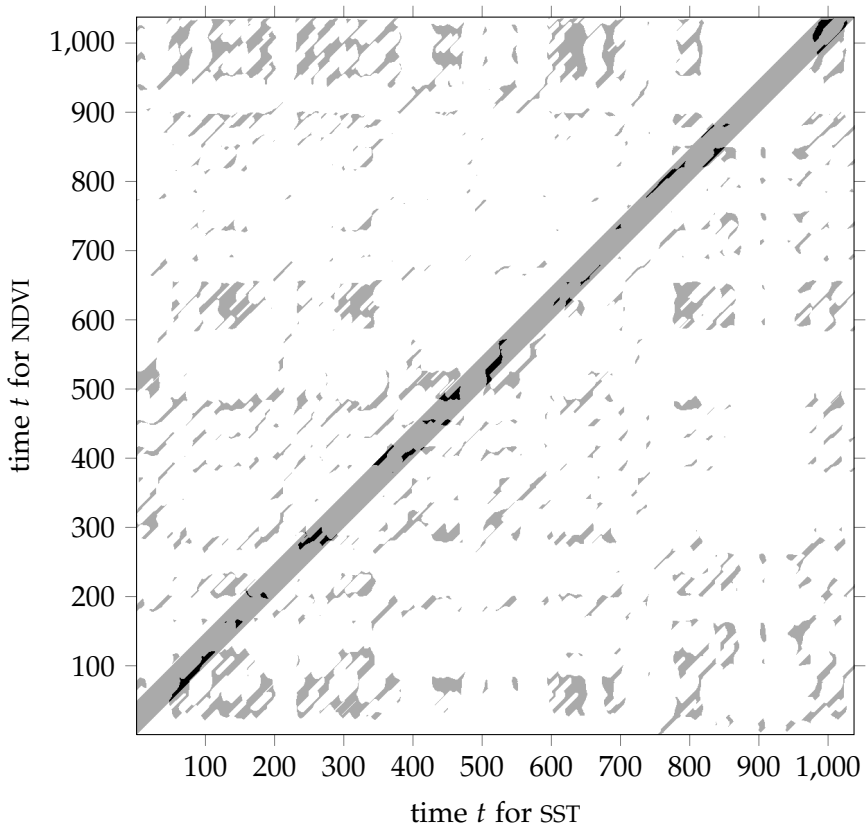


Figure 19: Diagonals the RQA indices were calculated for (shown in black-/gray) of a CRP; for *all* NDVI values

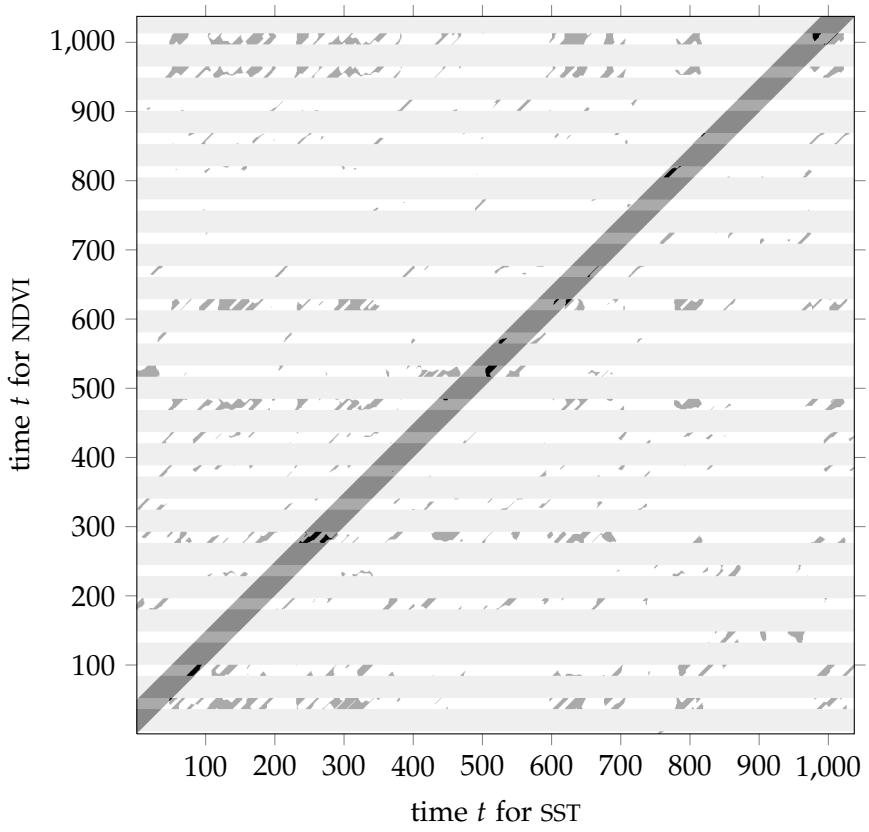


Figure 20: Diagonals the RQA indices were calculated for (medium gray/black) and NDVI anomaly values just from *winter* combined (other seasons marked light/dark gray) of a CRP

9 RESULTS AND INTERPRETATION

9.1 OVERVIEW

For a first overview of sea areas with a potential influence on vegetated areas, the average Recurrence Rates over all the NDVI classes per continent and time lags of 0 to 10, 10 to 20, 20 to 30, 30 to 48 and 10 to 48 weeks were calculated. Figure 21 and Figure 22 show two of those average *RR* images for Western Asia and Europe. In both images, a sea area located in the Pacific Ocean close to the equator shows the highest Recurrence Rates, although it seems to be a slightly different part in both images.

The actual result of the correlation calculation were 882 different "maps" for each NDVI class, showing one specific positive or negative RQA index for one specific time lag, one subset of NDVI input values and the whole sea surface.

Figure 23, Figure 24 and Figure 25 are examples of such a maps, showing the three different RQA indices for different time lags and summer, winter and the whole time period.

These figures clearly illustrate that

- distinctive structures or patterns are formed by all of the three RQA indices and their distribution is far from random,
- the structures look different for NDVI classes located in different parts of the world and
- the Recurrence Rate seems to be the least noisy and most promising index.

9.2 EXAMPLE - SOUTHERN FRANCE (EUROPE)

Since a visual representation of the complete resulting dataset is not possible due to its size, this section focuses on an NDVI class located in the south of France, Europe. It was chosen because it was exceptionally affected by a severe drought during summer 2003 (with more than 70,000 casualties all over Europe [48]), and RQA possibly allows to find any connections with this event in the SST dataset. Figure 26 displays the size and location of this NDVI class; of course, the same data is available for all the other NDVI classes too.

Although the NDVI anomaly series of this class is obtained by calculating the z-score of the mean of all the NDVI time series from each

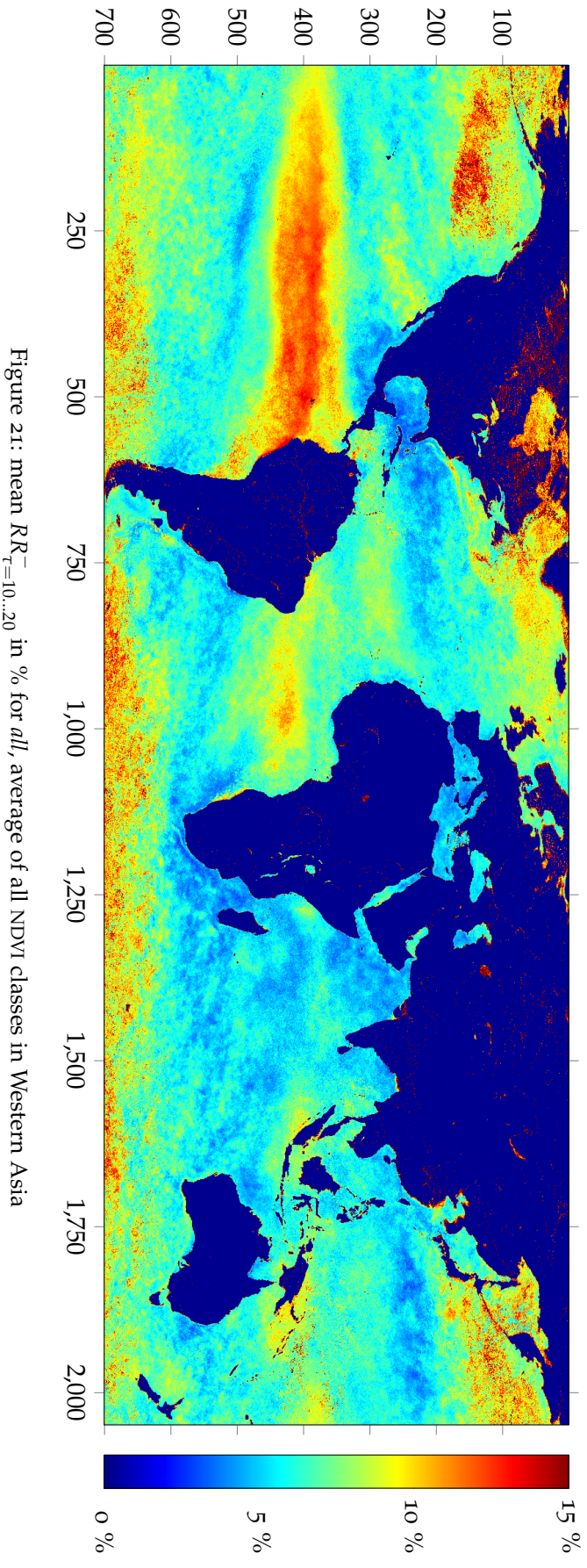


Figure 21: mean $RR^{-}_{\tau=10...20}$ in % for *all*, average of all NDVI classes in Western Asia

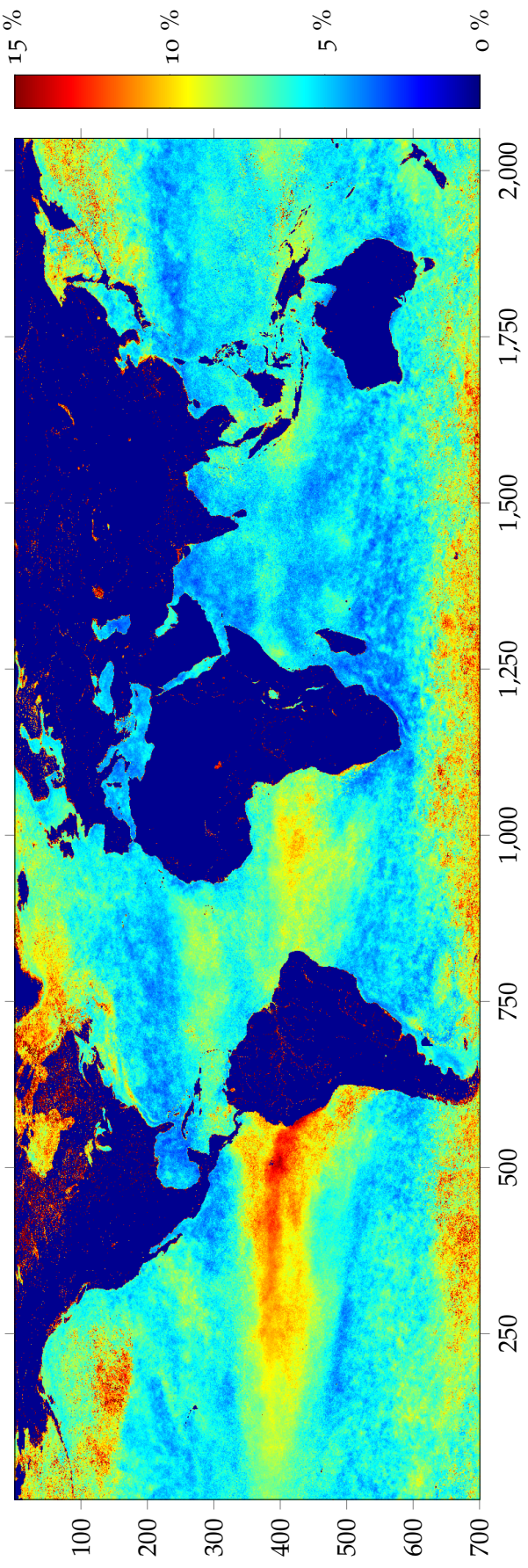


Figure 22: mean $RR^{-}_{\tau=10..20}$ in % for *all*, average of all NDVI classes in Europe

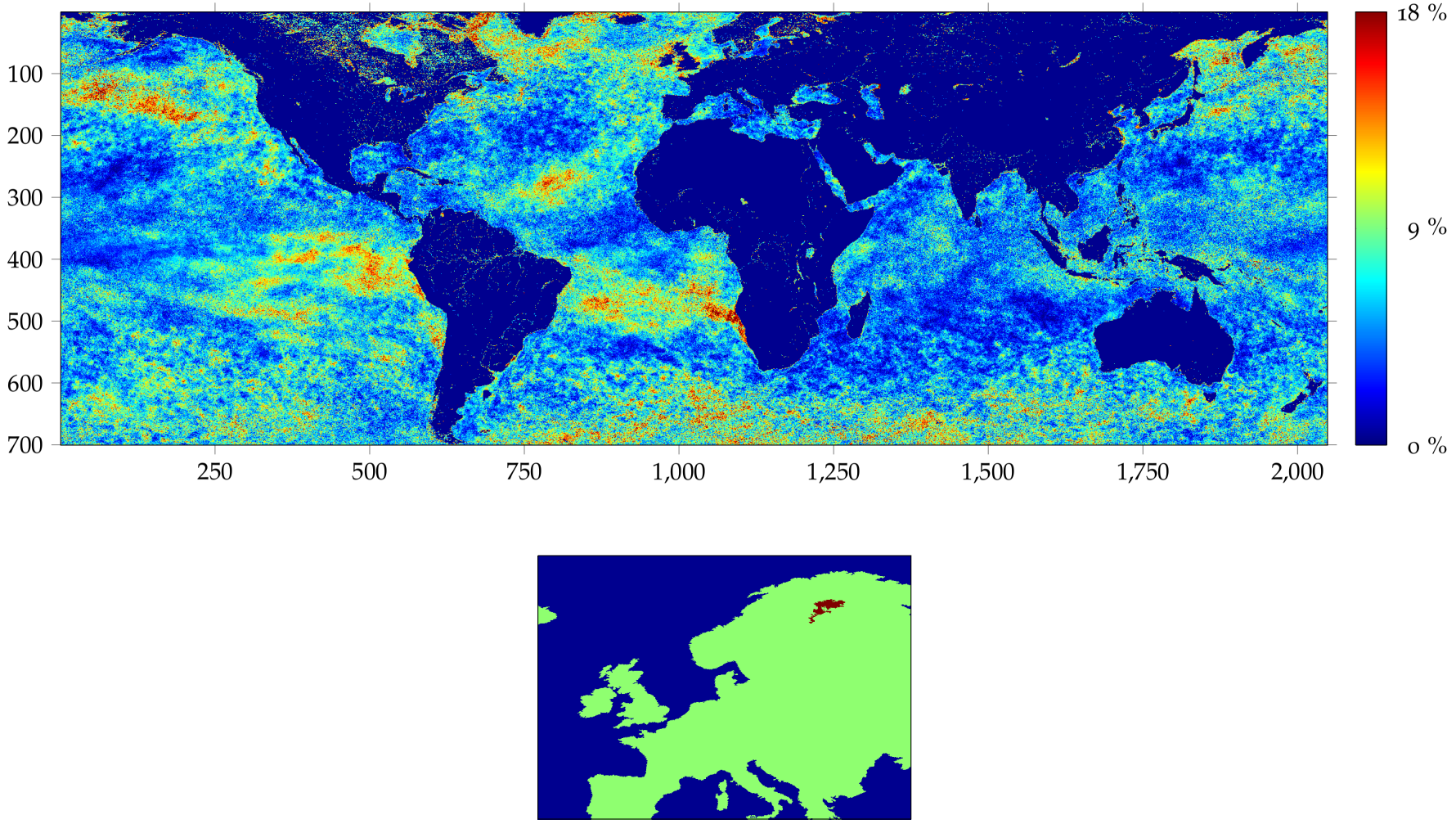


Figure 23: $RR^{+}_{\tau=2}$ in % for *all*, NDVI class in Northern Europe

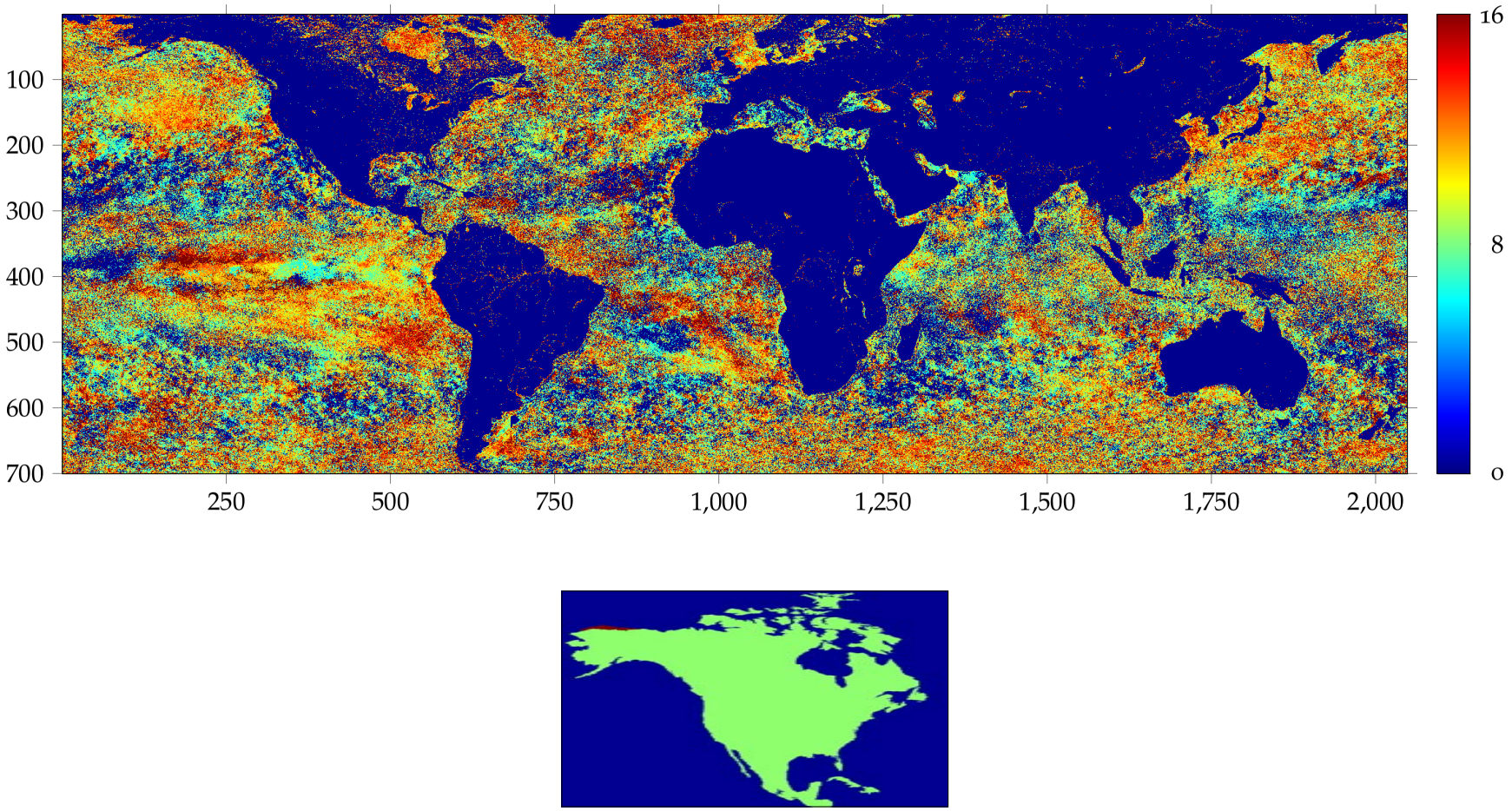


Figure 24: $L^{+}_{\tau=4}$ for *winter*, NDVI class in North America

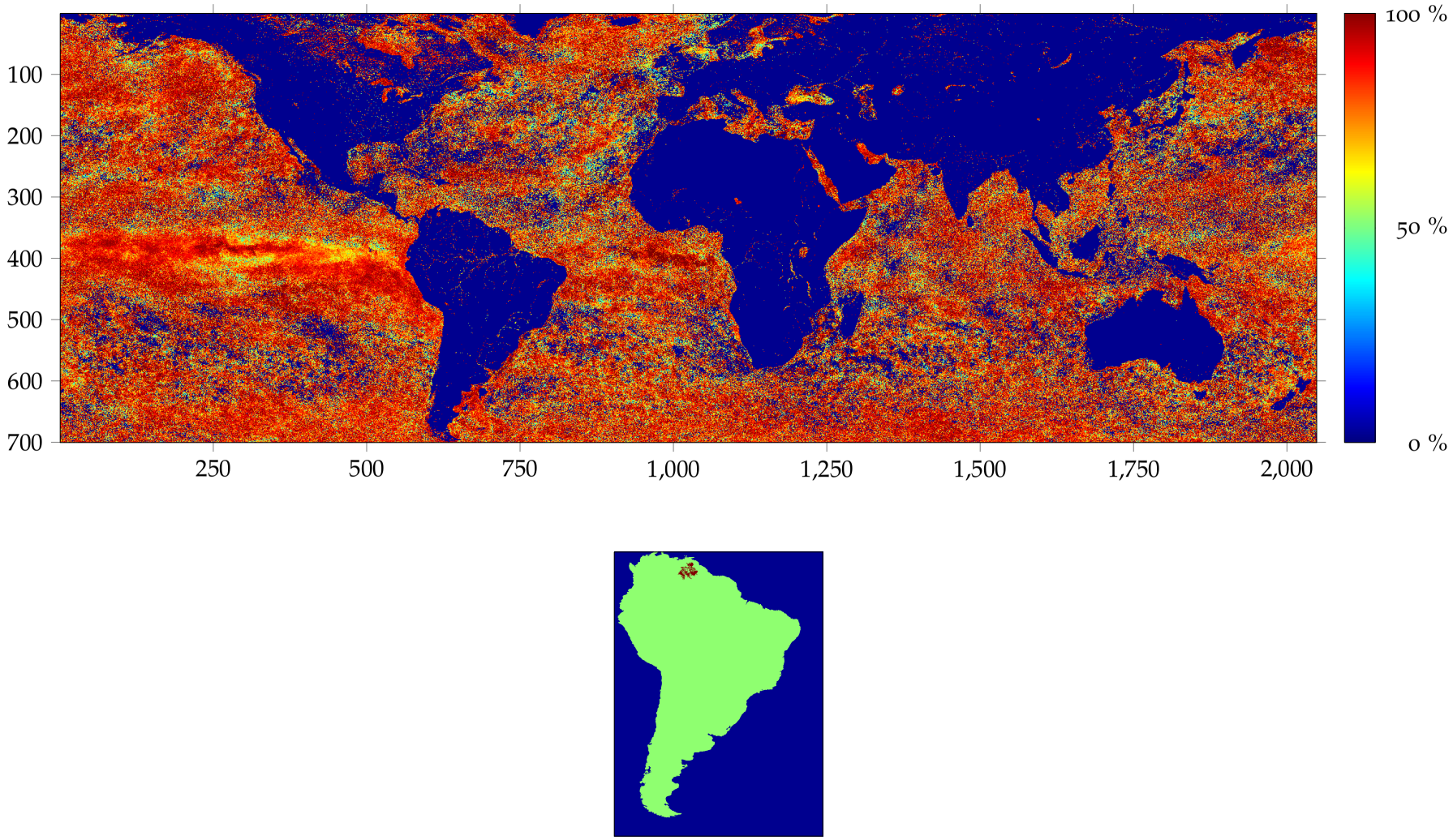


Figure 25: $DET^{-}_{\tau=10}$ in % for *summer*, NDVI class in South America

Figure 26: Size and location of the France NDVI class in Europe

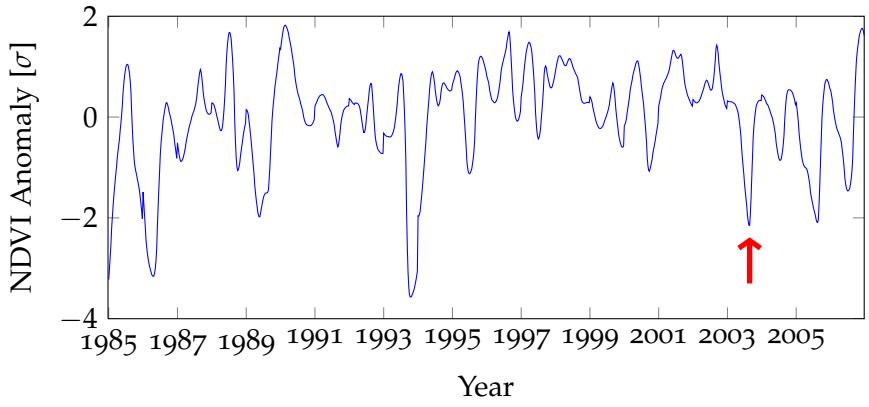


Figure 27: NDVI anomalies France; drop in summer 2003 at week 896

individual pixel of this class, it clearly shows a drop during this particular time (see Figure 27).

Since the Recurrence Rate RR has shown to be the most promising RQA index for this application, these results should now be examined more in detail. Figure 28, Figure 29, Figure 30 and Figure 31 display the Recurrence Rates based on the positive and negative CRPs for a time lag of 10 weeks for *summer*, *winter* and *all*. Again, the RR values form clear structures all over the oceans, but their interpretation is rather difficult. However, they do not seem to change a lot if *all* values or just the ones from *summer* or *winter* are taken into account; the structures seem to remain similar.

Another thing that catches one's eye is the fact that across the oceans, areas with high RR^+_{10} and RR^-_{10} seem to somehow "alternate". To illustrate this observation, Figure 32 shows areas with $RR^+_{10} > 15\,\%$ and/or $RR^-_{10} > 15\,\%$. Obviously, sea areas tend to either have a high RR^+ or RR^-, not both at the same time.

To look into the results even further, a five pixels out of these areas were chosen and their SST-NDVI anomaly signatures compared to the RQA results (Figure 33). In the SST-NDVI anomaly plots (Figure 34, Figure 35, Figure 36, Figure 37 and Figure 38), no clear relationships between those two time series are visible for all of the pixels, but the

development ofRR^+ and RR^- over the time lags from 0 to 48 weeks often reveal distinct maxima. It was therefore decided to smoothe the RR values of each SST pixel using the above-mentioned smoother (Section 3.3) with $\lambda = 10^{-0.4}$ and $d = 2$. Then the time lags with the maximum RR^+ and RR^- were determined for each pixel and are displayed in Figure 39 and Figure 40. Clearly also the time lag with the highest RR are distributed following certain patterns all across the sea surface.

Since in some cases the maximum RR occurs with a time lag of less than 5 weeks, additionally the maximum RR between time lags of 6 and 15 weeks might be of interest (Figure 41 and Figure 42).

Generally, an almost unlimited amount of different map plots or images can be generated out of the RQA results. Since we lack the necessary knowledge to interpret the resulting patterns and to compare them to known phenomena, just a small part of the possibilities are shown here. Further analysis, validation and interpretation may be the focus of future projects.

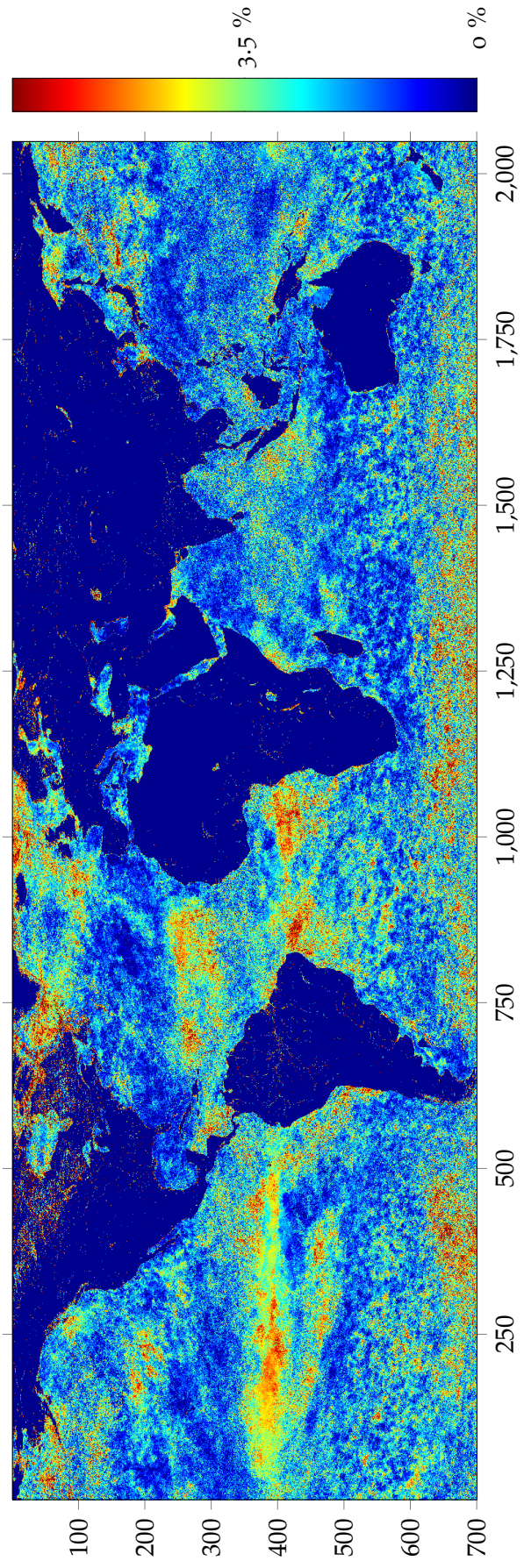


Figure 28: RR_{10}^{+} in % for *summer*, NDVI class France

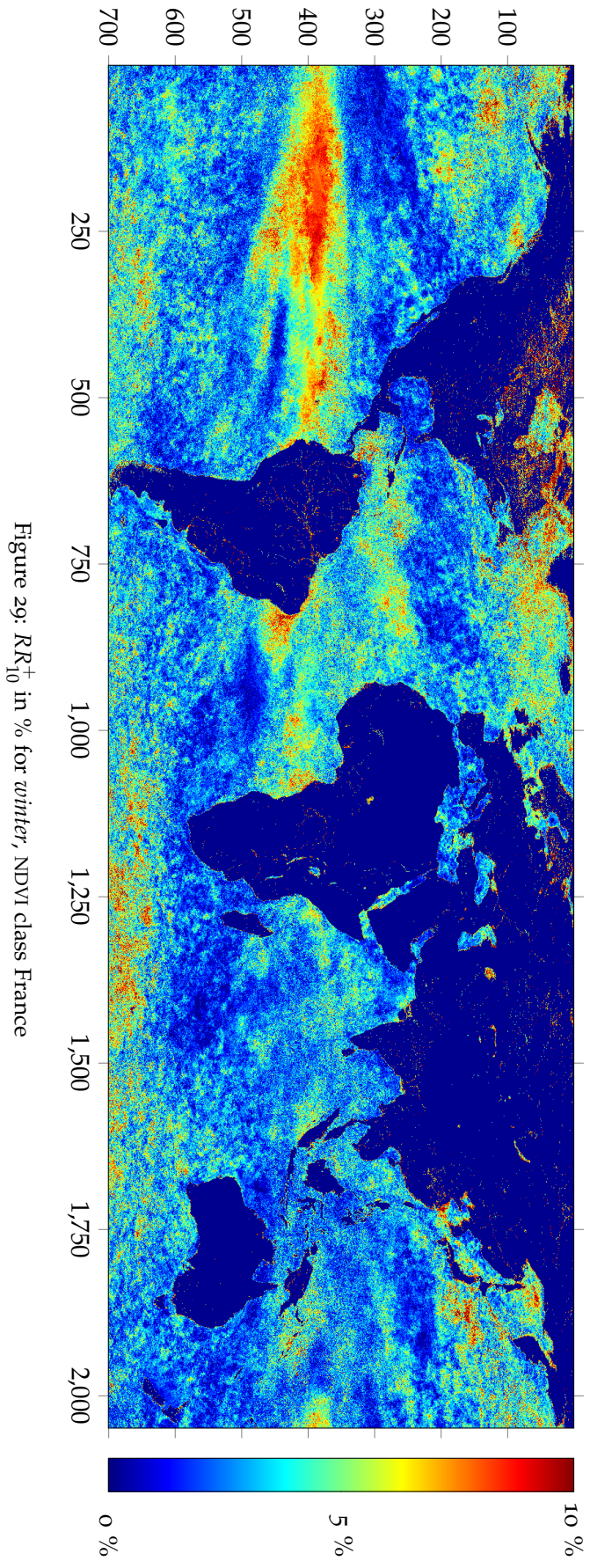


Figure 29: RR_{10}^{+} in % for *winter*, NDVI class France

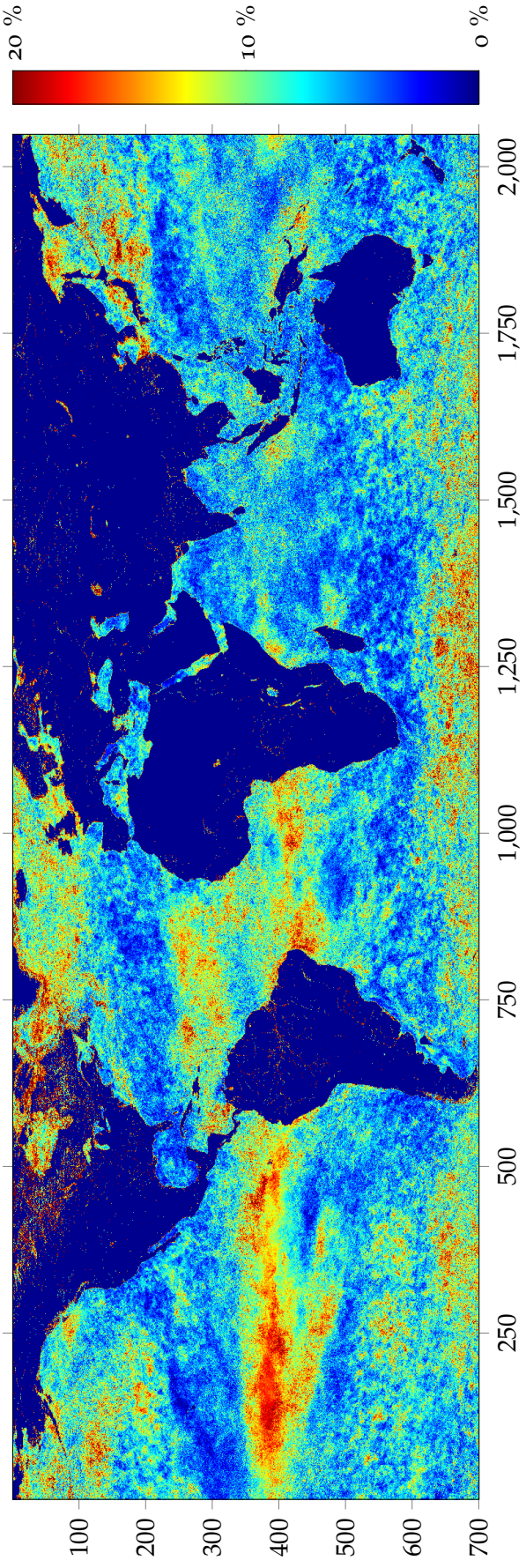


Figure 30: RR^{+}_{10} in % for *all*, NDVI class France

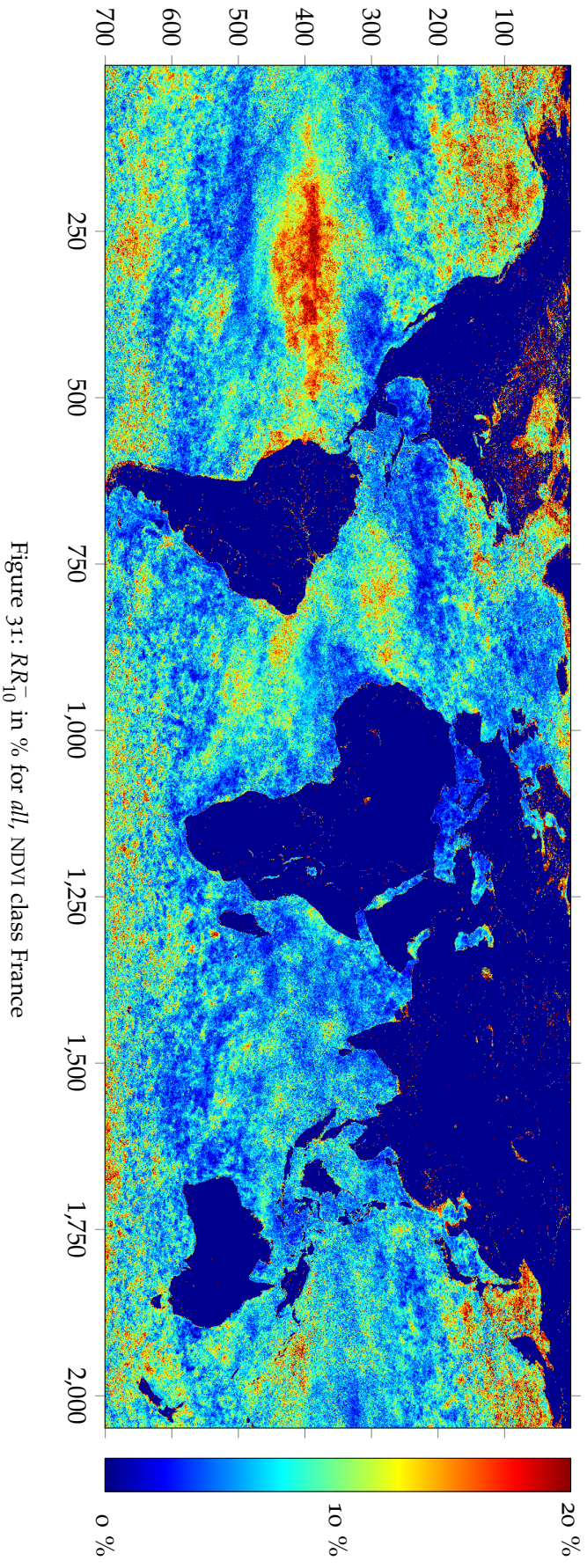


Figure 31: RR_{10}^{-} in % for *all*, NDVI class France

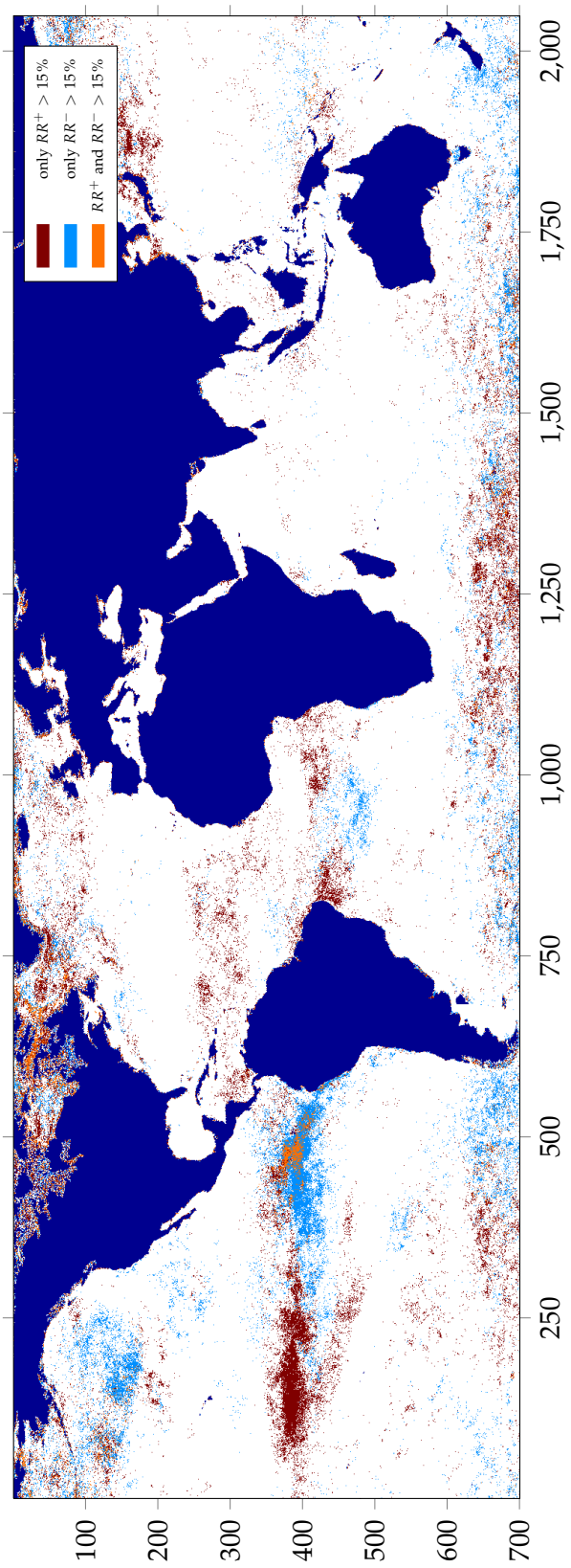


Figure 32: Areas with RR^+_{10} and $RR^-_{10} > 15\,\%$ for *all*, NDVI class France

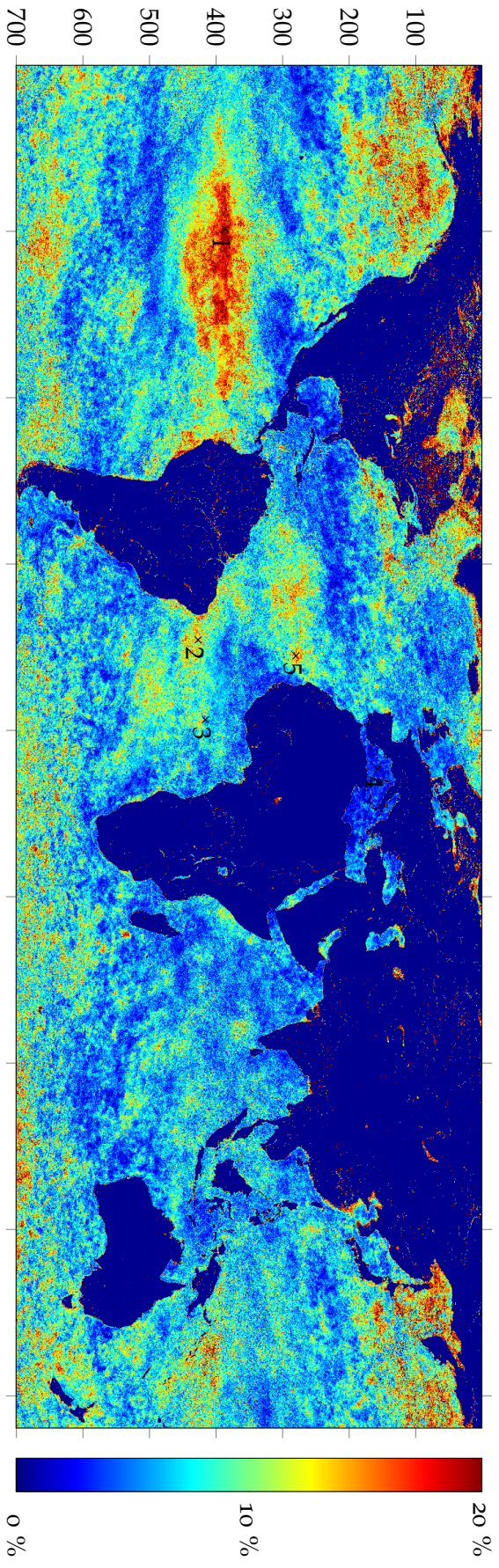


Figure 33: RR_{10}^{-} in % for *all*, NDVI class France, with locations of 5 selected SST pixels

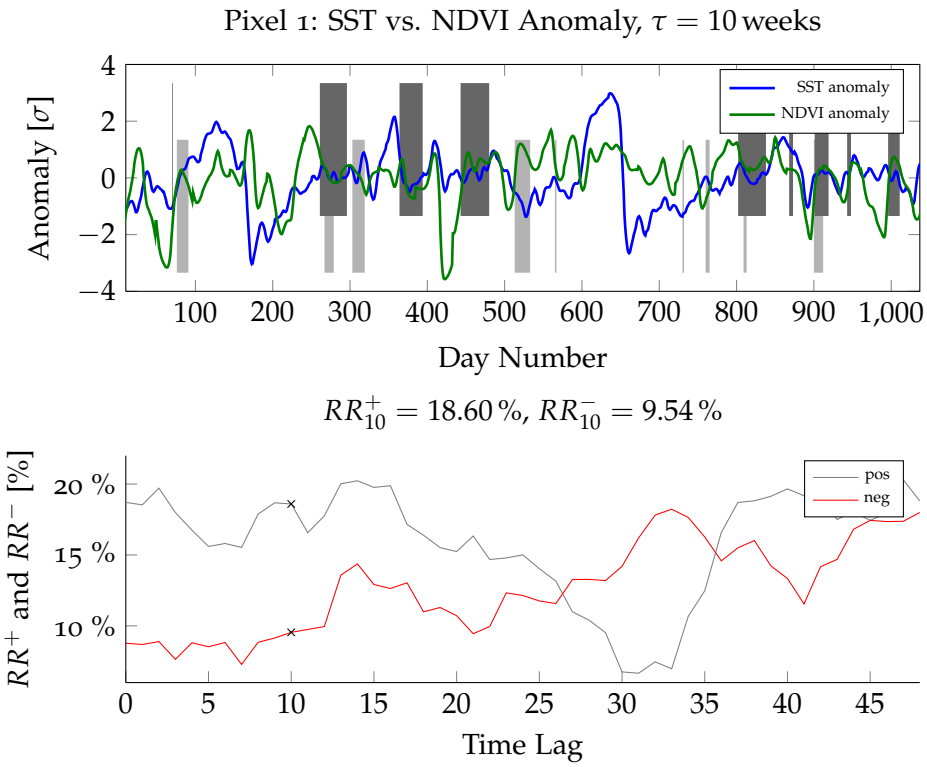


Figure 34: Calculation of CRM^+_{10} (dark gray areas) and CRM^-_{10} (light gray areas) and development of $RR^+_{0...48}$ and $RR^-_{0...48}$ for SST pixel 1

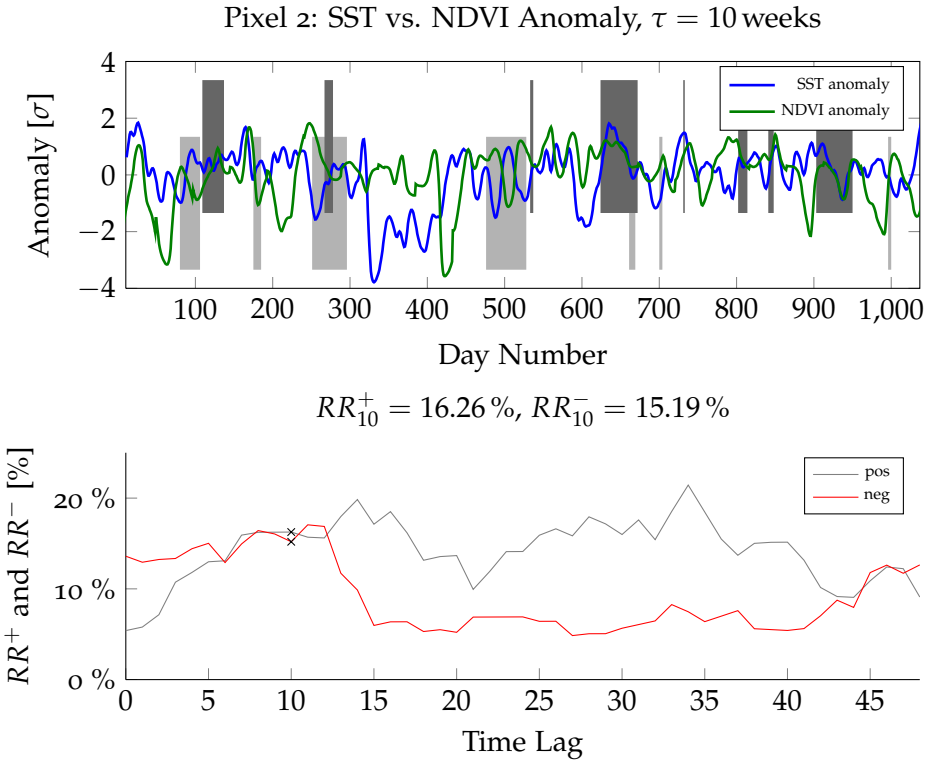


Figure 35: Calculation of CRM^+_{10} (dark gray areas) and CRM^-_{10} (light gray areas) and development of $RR^+_{0...48}$ and $RR^-_{0...48}$ for SST pixel 2

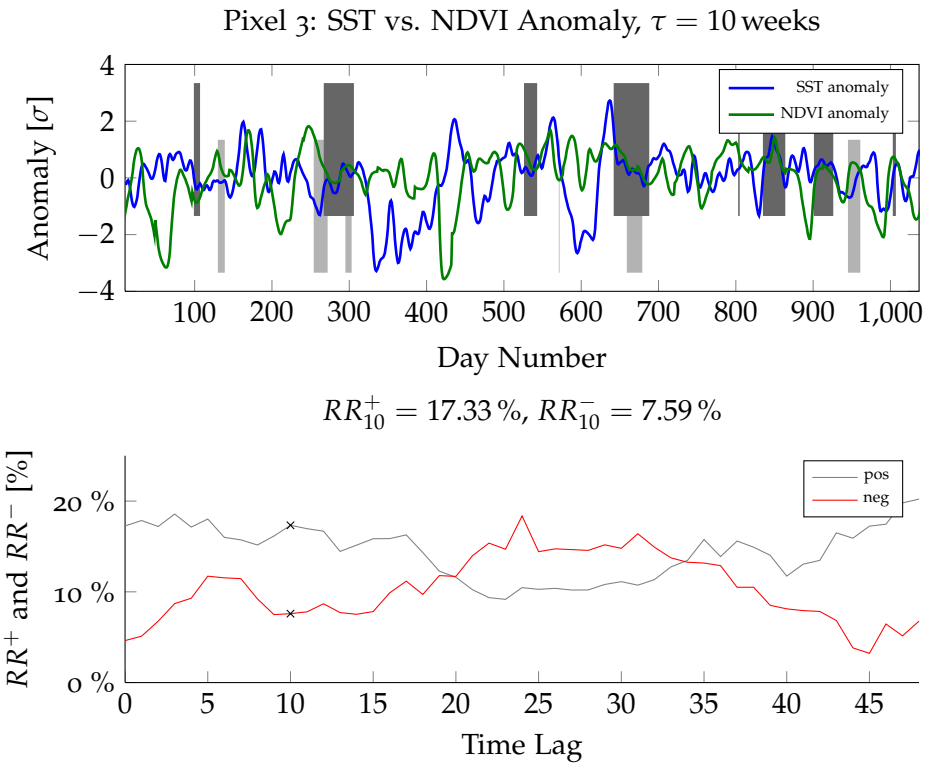


Figure 36: Calculation of CRM^+_{10} (dark gray areas) and CRM^-_{10} (light gray areas) and development of $RR^+_{0\ldots48}$ and $RR^-_{0\ldots48}$ for SST pixel 3

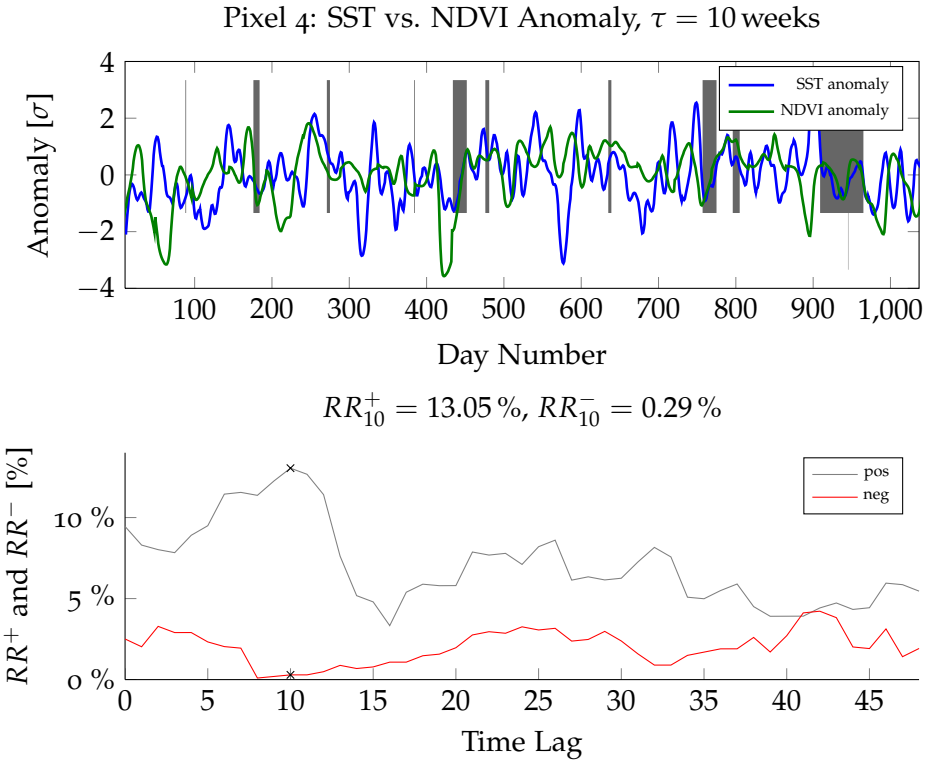


Figure 37: Calculation of CRM^+_{10} (dark gray areas) and CRM^-_{10} (light gray areas) and development of $RR^+_{0\ldots48}$ and $RR^-_{0\ldots48}$ for SST pixel 4

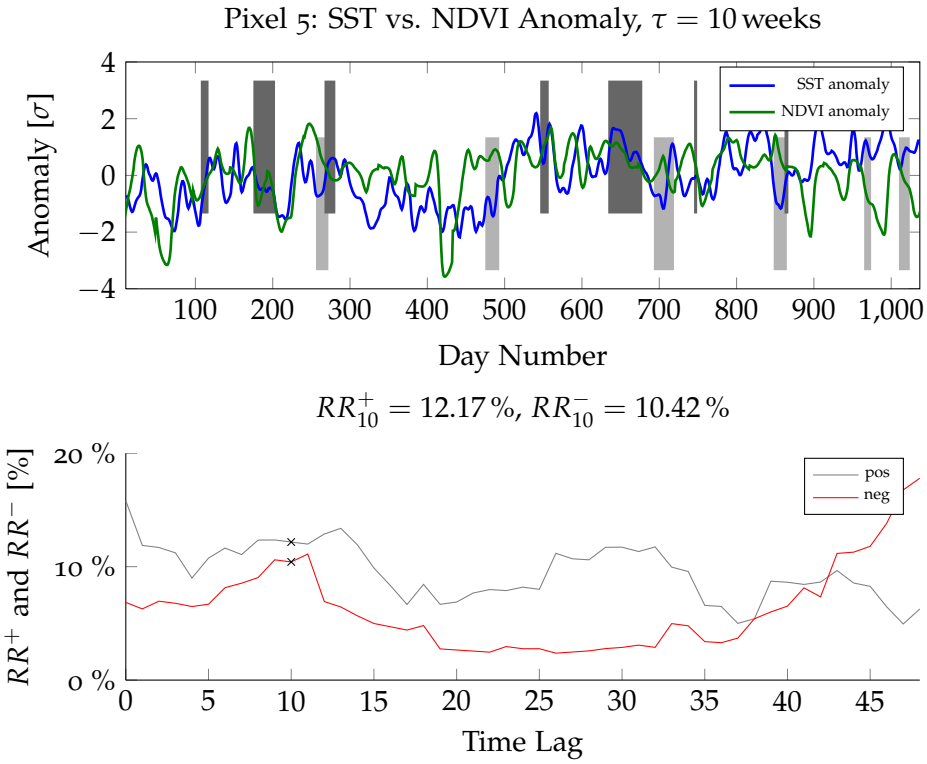


Figure 38: Calculation of CRM^{+}_{10} (dark gray areas) and CRM^{-}_{10} (light gray areas) and development of $RR^{+}_{0...48}$ and $RR^{-}_{0...48}$ for SST pixel 5

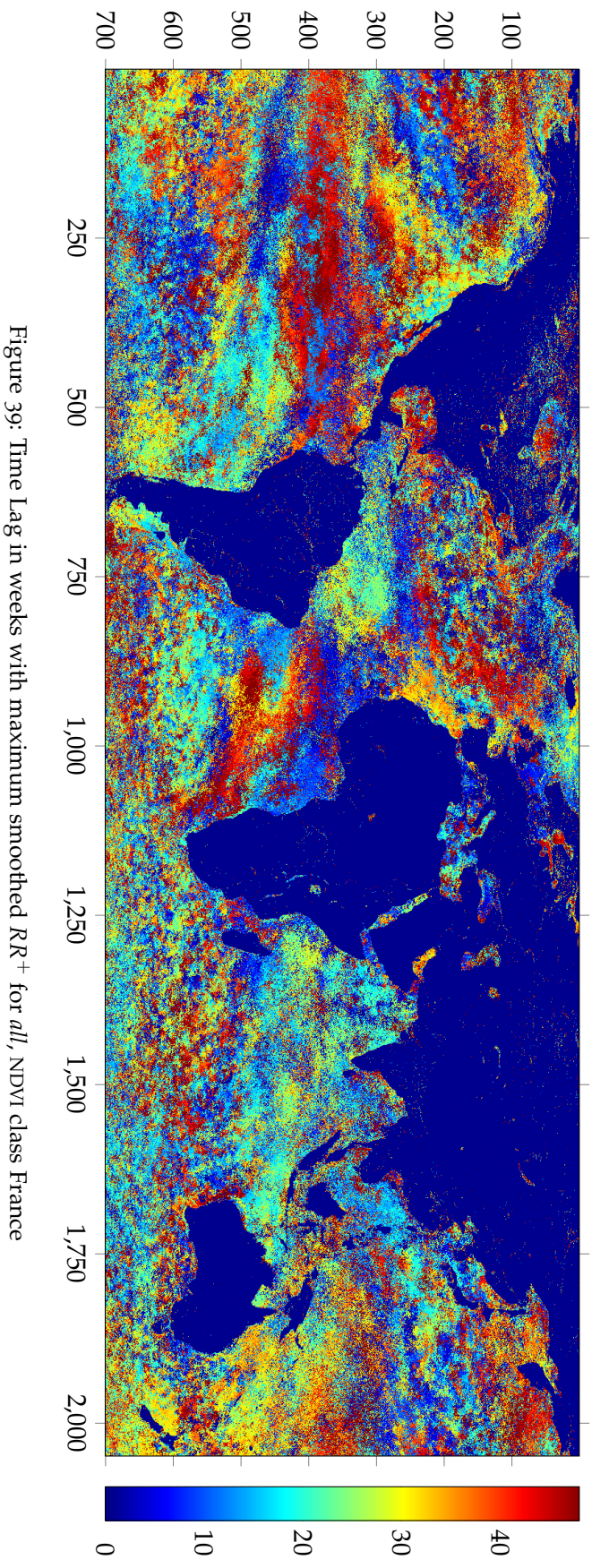


Figure 39: Time Lag in weeks with maximum smoothed RR^+ for *all*, NDVI class France

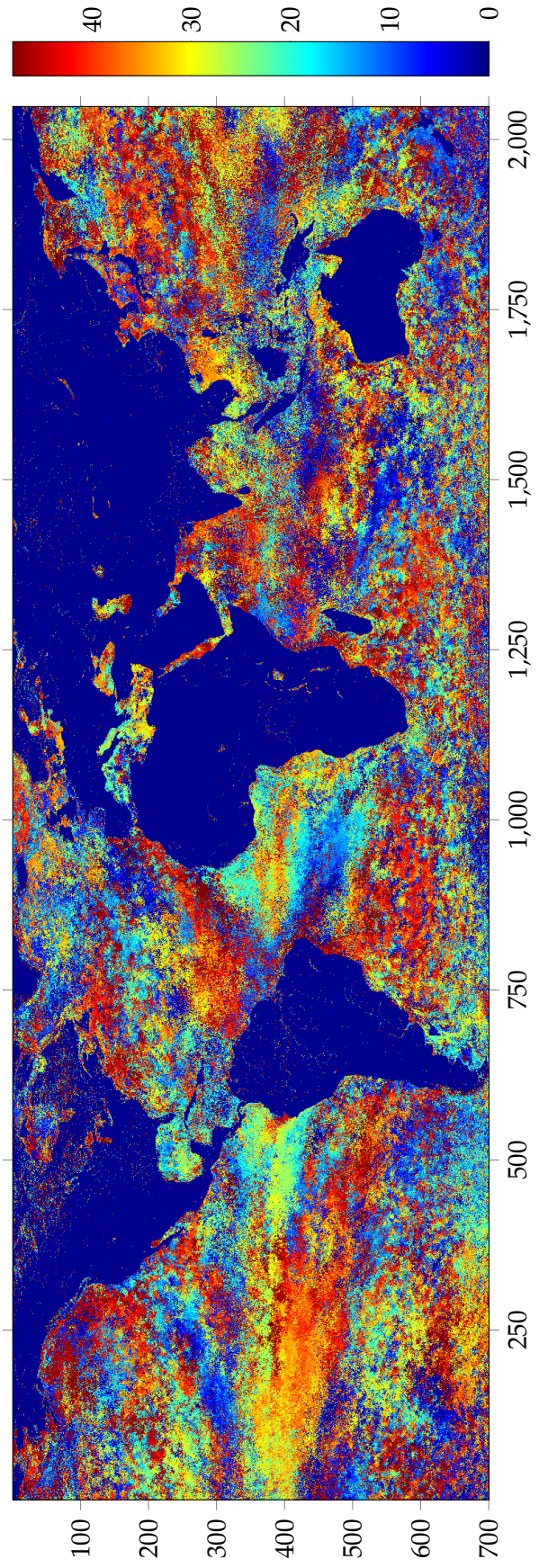


Figure 40: Time Lag in weeks with maximum smoothed RR^- for *all*, NDVI class France

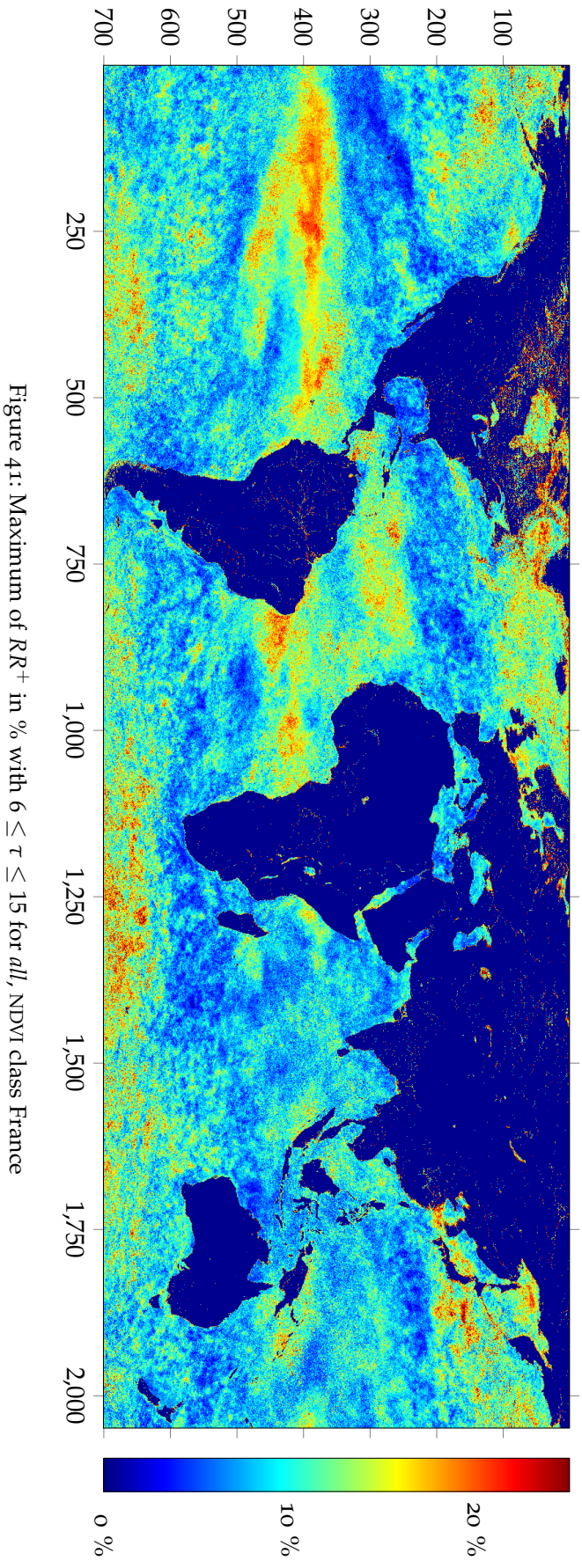


Figure 41: Maximum of RR^+ in % with $6 \leq \tau \leq 15$ for *all*, NDVI class France

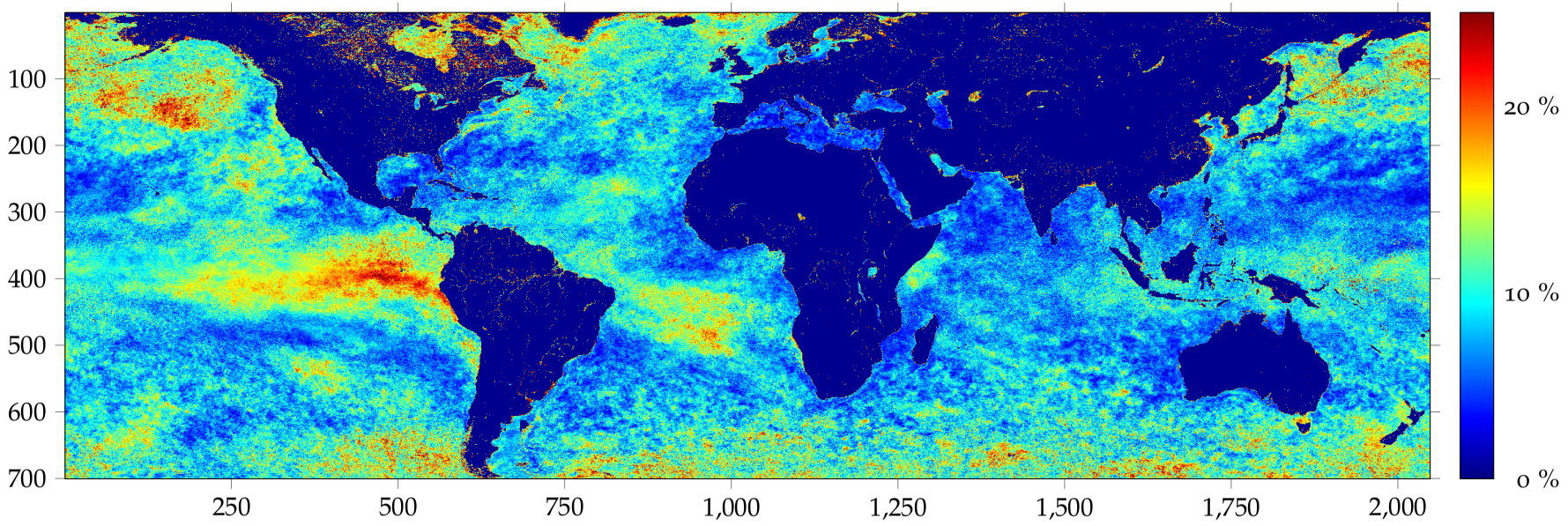


Figure 42: Maximum of RR^{-} in % with $6 \leq \tau \leq 15$ for *all*, NDVI class France

10 CONCLUSION AND OUTLOOK

Because of the huge amount of data and the large number of individual parameters, only a small percentage of the total results of the calculations could be inspected in detail. Nevertheless, we think this approach is promising and worth being pursued in future research projects.

Since the use of CRPs and RQA together with remote sensing data is relatively new, we learned plenty of things in the course of this work:

- Applications like that heavily depend on the quality of the underlying datasets. Additional quality tests, improved filtering and validation checks during preprocessing might lead to better results.
- RPs and CRPs are based on a variety of different parameters. Slight adjustments of these can greatly affect the nature of the plots and thus lead to completely different RQA results.
- The resulting RQA indices are clearly not randomly distributed all across the oceans, but show distinctive patterns and structures. However, interpreting these patterns is rather difficult.
- The Recurrence Rate RR seems to be the most stable and least "noisy" RQA index of the three that were tested (DET, RR and L).
- Calculating the RQA indices for the whole period or replacing parts of the CRPs with zeros (*summer* or *winter*) seems to alter the general value range of the indices, but the resulting structures remain more or less the same.
- There are vast possibilities of further processing and illustrating the results (e. g. map plots, animations, additional filtering or smoothing, additional calculations etc.).
- Limited resources required a drastic reduction of the amount of input data. Classifying the NDVI and cropping of the SST dataset was not planned beforehand, but became essential during resource planning.

Additional tasks that go beyond the scope of this thesis, but might be in the focus of future projects include

1. showing our preprocessing chain, set of CRP parameters and calculations to experienced users of RPs, CRPs and RQA,

2. comparing our outcome to the results of studies with similar objectives, but different approaches of achieving them,

3. presenting our results to specialists in oceanography and climatology, since we lack the knowledge to identify possible well-known climate phenomena our results might represent,

4. focusing on (smaller) regions with well-studied teleconnections to be able to work with higher spatial resolutions and to verify the results more easily and

5. testing different RQA indices, CRP parameters etc.

Part V

APPENDIX

A MATLAB FUNCTIONS

A.1 PREPROCESSING FUNCTION

The function `prep_sst` was used to preprocess the SST dataset, i. e. to interpolate, smoothe and filter to daily values, rescale back to weekly values, get the mean and standard deviation and calculate the anomalies.

A similar function exists for the preprocessing of the NDVI dataset.

Function 1: `prep_sst`

```
function prep_sst
%  Function to preprocess SST DATA


%  Definitions
numd = 8035; %  number of days
perl = 48; %  period length (number of weeks per year)
years = 22; %  number of years

lambda=10^4; %  smoothing parameter
d=2; % order of differences (for smoothing)

%  load Time Matrix (from tm_path)
load([tm_path,'\TM.mat']);
%  (amongst others) two variables (each with 8035=daily ...
    values):
%   daynum_sst:  0=missing, 1=day with SST value
%   daynum_outp: 0=not needed, 1=day for analysis

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5

%  Read SST Row files in a given path (path_mfiles)
matfiles = dir(fullfile(path_mfiles, '*.mat'));

%  Loop through all files
for i=1:numel(matfiles)

    disp(['Processing file ', num2str(i),...
        ' of ', num2str(numel(matfiles)), '...']);

    %  Load variables out of each file
    %  (quality flag, land mask, SST values)
    load(fullfile(path_mfiles, matfiles(i).name),...
        'qual', 'sst', 'land');

    %  Preallocate output
    sstw_sm_n = zeros(1, size(sst, 2), perl*years, 'double');

    %  Loop through all Pixels
    for x = 1:size(sst, 2)

        disp(['X: ',num2str(x)]);

        %  if current pixel is a water Pixel
```

```
44          if ~land(1,x)
45
46              %  Prepare Data
47              sst_cur=double(squeeze(sst(1, x, :)));
48              qual_cur=squeeze(qual(1, x, :));
49
50              %  Preallocate daily weights
51              %  (just zeros; for smoothing)
52              w_d=false(numd, 1);
53
54              %  Preallocate daily SST values
55              sst_d=zeros(numd, 1, 'double');
56
57              %  Assign a weight of 1 to values
58              %  with quality flag > 3
59              w_w=qual_cur>3;
60
61              %  Populate daily weights/SST values with
62              %  the weekly values
63              w_d(daynum_sst(:,2))=w_w;
64              sst_d(daynum_sst(:,2))=sst_cur;
65
66              %  apply smoother
67              sst_d_s=func_whitsmw(sst_d, w_d, lambda, d);
68
69              %  Select weekly values
70              sst_w_s = sst_d_s(daynum_outp);
71
72              %  Reshape to matrix for mean
73              %  calculation and normalization
74              sst_w_sr = reshape(sst_w_s, [perl, years]);
75
76              %  calculate mean year
77              m_sst_w_s = mean(sst_w_sr, 2);
78
79              %  Calculate Standard Deviation
80              std_sst_w_s=std(sst_w_sr, 1, 2);
81
82              %  Repeat mean year and std dev for subtraction
83              m_sst_w_s = repmat(m_sst_w_s, [years,1]);
84              std_sst_w_s = repmat(std_sst_w_s, [years,1]);
85
86              %  Calculate ZSCORE
87              sst_w_s_n = (sst_w_s-m_sst_w_s)./std_sst_w_s;
88
89              %  Allocate to output variable
90              sstw_sm_n(1, x, :) = ...
91                  reshape(sst_w_s_n, [1,1,numel(sst_w_s_n)]);
92          end
93      end
94
95
96      %  Prepare output path (path_write)
97      fn_ready = [path_write, matfiles(i).name];
98
99      %  Save File
100     save(fn_ready, 'sstw_sm_n', 'land');
101     disp(['File ', fn_ready, ' written.']);
102     disp(' ');
103 end
```

In lines 6–11, the total number of days `numd`, the number of weeks per year `perl`, the total number of years `years` and the smoothing parameters `lambda` and `d` are defined. Then the time matrix file where binary variables `daynum_sst`, `daynum_outp` and `daynum_ndvi` used

to extract the required days out of the total 8035 are stored is loaded. In line 22, all the MATLAB files in the input folder `path_mfiles` are listed; each of these files holds a $1 \times 2048 \times 1148$ pixel row with SST values and is named in a way that the reading order is correct. Then the function loops through all the SST files found, opens them in line 32 and loads the variables `sst` (the 1148 SST values), `land` (the land-water-mask) and the quality flag `qual` for each SST value. After that, the output matrix `sstw_sm_n` is preallocated; note the number of SST observations (1056).

As the values are processed pixel-wise, the function loops through all the pixels in line 39, and checks if the current pixel is a water or land pixel to avoid unnecessary processing time. In line 47 and 48, the dimensions of the current SST and quality flag values `sst_cur` and `qual_cur` (which both are $1 \times 1 \times 1148$) are reduced to 1148×1 using the MATLAB function `squeeze`.

The smoothing algorithm `func_whitsmw` accepts a weight parameter (1 or 0) for each input value: 0 marks a value as missing and for interpolation, while 1 tells the smoother to use this value. After preallocating the variables for the weights for each day and the daily SST values in lines 5 and 55, a weight vector `w_w` is created for the weekly SST observations, where values with a quality flag greater than 3 are assigned 1, the rest 0. This vector is then rescaled to daily weights in line 63 using the second column of the matrix `daynum_sst`, which selects the 1148 days the observed values belong to out of the 8035 days and assigns them the weigh of `w_w`. The same is done with the SST values and the vector `sst_d` in line 64.

Now the smoothing and interpolating algorithm `func_whitsmw` is applied to the data using the weight vector `w_d` and the parameters `lambda` and `d`. A detailed description of this algorithm is available by Eilers [18]. Line 70 rescales the resulting data back to weekly values using the matrix `daynum_outp` which defines the 4^{th}, 11^{th}, 18^{th} and 25^{th} day of each month and stores them in `sst_w_s`.

As a last step, the anomalies are calculated: Line 74 reshapes `sst_w_s` to a 48×22 matrix and assigns the result to `sst_w_sr` to calculate the average of each month `m_sst_w_s` in line 76 and the standard deviation `std_sst_w_s` in line 80. To subtract the mean and standard deviation from all the years, the resulting 48×1 vectors are replicated 22 times to dimensions of 1056×1. The zscore is calculated in line 87 (see Equation 2), the resulting matrix is reshaped back to dimensions of $1 \times 1 \times 1056$ and stored in the output matrix `sstw_sm_n` at the current pixel position `x`; then the next pixel is processed in the same way.

After processing the last pixel of each row, the anomalies are saved in `path_write` and a message is displayed.

A.2 CRP AND RQA FUNCTIONS

The processing chain for calculating the CRPs and diagonal-wise RQA indices was split over several separate MATLAB functions to improve its handling. The first function, which is called to start the process, is called `telecon`.

Function 2: `telecon`

```
function telecon(sstnum, classnum, ndviw_sm_n)
%  Function to calculate all (diagonalwise)
%  RQA measures bewteen an NDVI time series and
%  an SST Pixel Row


%  define Parameters
m=20;
t=1;
e=0.8;

%  Load definitions for summer/winter
%  (path_sumwin needs to be set)
load([path_sumwin,'sumwin.mat']);
%  This file contains two variables:
%   summr: defines the summer weeks
%   wintr: defines the winter weeks

%  Create Variables for CRPs
i=(1:(1056-m+1))';
j=0:t:(m-1);
i=reshape(i(:,ones(1,m))+...
    j(ones((1056-m+1),1),:),m*(1056-m+1),1);
ds=eye(m);

%  Load selected SST data file
load([path_mfiles, num2str(sstnum, '%03d'),'.mat'],...
    'sstw_sm_n', 'land');

for x=1:2048
    %  Loop through pixels

    %  If it is no land pixel...
    if land(1,x)~=1

        %  ... Create positive and negative CRP
        Xp = calc_crp(sstw_sm_n(:,x,:), ...
            ndviw_sm_n, m, e, i, ds);
        Xn = calc_crp(sstw_sm_n(:,x,:), ...
            -ndviw_sm_n, m, e, i, ds);

        %%%%%  POSITIVE, SUMMER ONLY
        Xp_now=Xp;
        tsg = true(1,(1056-m+1));
        tsg(summr(1:(1056-m+1)))=0;
        Xp_now(tsg, :)=0;

        %  Calculate diagonal-wise RQA
        DETp_summer(1,x,:)=calc_crqad_diag_DET(Xp_now);
        RRp_summer(1,x,:)=calc_crqad_diag_RR(Xp_now);
        Lp_summer(1,x,:)=calc_crqad_diag_L(Xp_now);


        %%%%%  POSITIVE, WINTER ONLY
```

```
55          Xp_now=Xp;
56          tsg = true(1,(1056-m+1));
57          tsg(wintr(1:(1056-m+1)))=0;
58          Xp_now(tsg, :)=0;
59
60          %  Calculate diagonal-wise RQA
61          DETp_winter(1,x,:)=calc_crqad_diag_DET(Xp_now);
62          RRp_winter(1,x,:)=calc_crqad_diag_RR(Xp_now);
63          Lp_winter(1,x,:)=calc_crqad_diag_L(Xp_now);
64
65
66          %%%%%  POSITIVE, ALL
67          Xp_now=Xp;
68
69          %  Calculate diagonal-wise RQA
70          DETp_all(1,x,:)=calc_crqad_diag_DET(Xp_now);
71          RRp_all(1,x,:)=calc_crqad_diag_RR(Xp_now);
72          Lp_all(1,x,:)=calc_crqad_diag_L(Xp_now);
73
74
75          %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
76
77          %%%%%  NEGATIVE, SUMMER
78          Xn_now=Xn;
79          tsg = true(1,(1056-m+1));
80          tsg(summr(1:(1056-m+1)))=0;
81          Xn_now(tsg, :)=0;
82
83          %  Calculate diagonal-wise RQA
84          DETn_summer(1,x,:)=calc_crqad_diag_DET(Xn_now);
85          RRn_summer(1,x,:)=calc_crqad_diag_RR(Xn_now);
86          Ln_summer(1,x,:)=calc_crqad_diag_L(Xn_now);
87
88
89          %%%%%  NEGATIVE, WINTER
90          Xn_now=Xn;
91          tsg = true(1,(1056-m+1));
92          tsg(wintr(1:(1056-m+1)))=0;
93          Xn_now(tsg, :)=0;
94
95          %  Calculate diagonal-wise RQA
96          DETn_winter(1,x,:)=calc_crqad_diag_DET(Xn_now);
97          RRn_winter(1,x,:)=calc_crqad_diag_RR(Xn_now);
98          Ln_winter(1,x,:)=calc_crqad_diag_L(Xn_now);
99
100
101         %%%%%  NEGATIVE, ALL
102         Xn_now=Xn;
103
104         %  Calculate diagonal-wise RQA
105         DETn_all(1,x,:)=calc_crqad_diag_DET(Xn_now);
106         RRn_all(1,x,:)=calc_crqad_diag_RR(Xn_now);
107         Ln_all(1,x,:)=calc_crqad_diag_L(Xn_now);
108
109
110     end
111
112 end
113
114 %  Save results to output path (path_results)
115 save([path_results,...
116     '/Class_',num2str(classnum, '%03d'),'/',...
117     num2str(sstnum, '%03d'),'.mat'], ...
118     'DETp_summer', 'RRp_summer', 'Lp_summer', ...
119     'DETp_winter', 'RRp_winter', 'Lp_winter', ...
120     'DETp_all', 'RRp_all', 'Lp_all', ...
```

```
121       'DETn_summer', 'RRn_summer', 'Ln_summer', ...
122       'DETn_winter', 'RRn_winter', 'Ln_winter', ...
123       'DETn_all', 'RRn_all', 'Ln_all');
124
125   end
```

The function `telecon` expects 3 input arguments: `sstnum` (the number of the pixel row of the SST dataset, `ndvi_w_sm_n` (a weekly and smoothed NDVI anomaly time series) and `classnum` (the number of the NDVI class; just to save the outputs into the correct folder).

In lines 8–10, the desired values are assigned to the variables dimension `m`, time lag `t` and threshold `e`. Then the file `sumwin.mat` which contains two vectors (`summr` and `wintr`) for selecting appropriate months for the summer and winter period is loaded. Lines 20–24 create some embedding vectors for the later CRP calculation (done already here to save processing time). After that, the pixel row indicated by `sstnum` which contains the SST anomaly data is loaded.

Line 30 loops through all the pixels (1 . . . 2048) in the SST row. If the current pixel is not a land pixel (line 34), the positive and negative CRPs are calculated by calling the function `calc_crp` two times and passing the current SST anomaly time series `sstw_sm_n(:,x,:)`, the NDVI anomalies `ndviw_sm_n` and some CRP and RQA parameters as arguments to this function (line 37–40). The first time, the positive CRP `Xp` is calculated, while the second time the NDVI anomaly time series is passed to the function with a negative sign to recieve the negative CRP `Xn`.

To calculate RQA indices for the different times *summer*, *winter* and *all*, the rows of the two resulting CRPs `Xp` and `Xn` which should not be taken into consideration are then set to zero (e. g. lines 43–46 or 78–81) and the three indices are calculated by handing the new CRPs (`Xp_now` and `Xn_now`) over to the functions `calc_crqad_diag_DET`, `calc_crqad_diag_RR` and `calc_crqad_diag_L` (e. g. lines 49–51).

In the end, the resulting parameters are saved class-wise in separate folders (lines 115 ff).

Function 3: `calc_crp`

```
1  function X=calc_crp(x, y, m, e, i, ds)
2  %  Calculate Cross Recurrence Plot
3  %  (c) by Norbert Marwan,
4  %  http://tocsy.pik-potsdam.de/CRPtoolbox/
5  %  edited by Michael Wess
6
7
8  %  Remove singleton dimensions
9  x=reshape(x, [1056, 1, 1]);
10 y=reshape(y, [1056, 1, 1]);
11
12 %  Determine length after applying time
13 %   lag and dimension (may be shorter!)
14 NX=(1056-m+1);
15
16 %  Create new matrices
```

```
17  x1=x(i);
18  x2=reshape(x1,NX,m);
19  y1=y(i);
20  y2=reshape(y1,NX,m);
21
22  %  Switch vectors
23  y2=y2*ds;
24
25  %  Determine new sizes of the embedded vectors
26  [NX, ~] = size(x2);
27
28  %  Calculate indices
29  px = permute(x2, [ 1 3 2 ]);
30  py = permute(y2, [ 3 1 2 ]);
31  on=ones(1,NX);
32
33  %  Calculate Differences
34  s1 = px(:,on,:) - py(on,:,:);
35
36  %  Calculate absolute differences
37  s = max(abs(s1),[],3);
38
39  %  Apply threshold
40  X2=s<e;
41
42  %  Transpose and convert to uint8
43  X=uint8(X2)';
44
45  end
```

The function `calc_crp` expects the two time series `x` and `y` and the CRP and RQA parameters as input arguments. At first, any singleton dimensions of the two time series are removed (lines 9 and 10). In line 14, the size of the CRP `NX` with the seleted dimension is calculated. Then, two new matrices are created, which are populated with the shifted input data (lines 17–31). Then the absolute differences are calculated (line 37) and the threshold is applied to create the CRP (line 40). As a last step, the resulting *CRM* `X2` is transposed and converted to class `uint8`.

The three functions `calc_crqad_diag_DET`, `calc_crqad_diag_RR` and `calc_crqad_diag_L` are used to calculate the diagonal-wise RQA indices based on the two CRPs `Xp` and `Xn`.

Function 4: `calc_crqad_diag_DET`

```
1  function DET=calc_crqad_diag_DET(X)
2  %  Calculate Determinism
3  %  (c) by Norbert Marwan,
4  %  http://tocsy.pik-potsdam.de/CRPtoolbox/
5  %  edited by Michael Wess
6
7  %  Define parameters
8  lmin=5;
9  w=48;
10
11 i=1;
12 for j=0:-1:-w
13     %  Loop from 0 to -w
14
15     clear z z0 z1
```

```
16
17      %  Get current diagonal
18      k=double(diag(X,j));
19
20      %  Detect changes
21      z=diff(k);
22      if ~isempty(find(~(z-1), 1))
23          z0(:,1)=find(~(z-1));
24      else
25          z0=0;
26      end
27
28      if ~isempty(find(~(z+1), 1))
29          z1=find(~(z+1));
30      else
31          z1=0;
32      end
33
34      %  Correct lengths of vectors
35      if z0(1)>z1(1)
36          z0(2:end+1,1)=z0(1:end);z0(1)=0;
37      end
38      if length(z0)>length(z1)
39          z1(end+1)=length(k);
40      end
41
42      %  Calculate line lengths and sort
43      l=sort(z1-z0);
44
45      %  Apply lmin
46      l1=l(l>lmin);
47
48      %  Calculate Determinism
49      DET(i)=sum(l1)/sum(k);
50
51      %  Increment counter
52      i=i+1;
53
54  end
55
56  %  Set NaNs to 0
57  DET(isnan(DET))=0;
58
59  end
```

Function 5: `calc_crqad_diag_RR`

```
1   function RR=calc_crqad_diag_RR(X)
2   %  Calculate Average Diagonal Line Length
3   %  (c) by Norbert Marwan,
4   %  http://tocsy.pik-potsdam.de/CRPtoolbox/
5   %  edited by Michael Wess
6
7   %  Define parameters
8   w=48;
9
10  i=1;
11  for j=0:-1:-w
12      %  Loop from 0 to -w
13
14      %  Get current diagonal
15      k=double(diag(X,j));
16
17      %  Calculate RR
```

```
18      RR(i)=sum(k)/length(k);
19
20      %  Increment counter
21      i=i+1;
22  end
23
24  %  Set NaNs to 0
25  RR(isnan(RR))=0;
26
27  end
```

Function 6: `calc_crqad_diag_L`

```
1   function L=calc_crqad_diag_L(X)
2   %  Calculate Average Diagonal Line Length
3   %  (c) by Norbert Marwan,
4   %  http://tocsy.pik-potsdam.de/CRPtoolbox/
5   %  edited by Michael Wess
6
7   %  Define parameters
8   lmin=5;
9   w=48;
10
11  i=1;
12  for j=0:-1:-w
13      %  Loop from 0 to -w
14
15      clear z z0 z1
16
17      %  Get current diagonal
18      k=double(diag(X,j));
19
20      %  Detect changes
21      z=diff(k);
22      if ~isempty(find(~(z-1), 1))
23          z0(:,1)=find(~(z-1));
24      else
25          z0=0;
26      end
27
28      if ~isempty(find(~(z+1), 1))
29          z1=find(~(z+1));
30      else
31          z1=0;
32      end
33
34      %  Correct lengths of vectors
35      if z0(1)>z1(1)
36          z0(2:end+1,1)=z0(1:end);z0(1)=0;
37      end
38      if length(z0)>length(z1)
39          z1(end+1)=length(k);
40      end
41
42      %  Calculate line lengths and sort
43      l=sort(z1-z0);
44
45      %  Apply lmin
46      l1=l(l>lmin);
47
48      %  Calculate Average
49      L(i)=mean(l1);
50
51      %  Increment counter
```

```
52      i=i+1;
53
54  end
55
56  %  Set NaNs to 0
57  L(isnan(L))=0;
58
59  end
```

To calculate DET and L, at first the lengths of the diagonal structures are calculated, then the threshold `lmin` is applied and then lengths of diagonals $>$ lmin to the total length of diagonal structures (line 49 of `calc_crqad_diag_DET`) and the mean lengths of diagonal structures $>$ lmin (line 49 of `calc_crqad_diag_L`) are calculated.

To obtain RR, the total length of diagonal structures is divided by the total length of the current CRP diagonal (line 18).

BIBLIOGRAPHY

[1] D. Adams. *The Restaurant at the End of the Universe*. Pan Books, 1980. (Cited on page vii.)

[2] J. Mennis. Exploring relationships between ENSO and vegetation vigour in the south-east USA using AVHRR data. *International Journal of Remote Sensing*, 22(16):3077–3092, 2001. (Cited on pages 3 and 4.)

[3] J. Verdin, C. Funk, R. Klaver, and D. Roberts. Exploring the correlation between Southern Africa NDVI and Pacific sea surface temperatures: results for the 1998 maize growing season. *International Journal of Remote Sensing*, 20(10):2117–2124, 1999. (Cited on page 3.)

[4] R.B. Myneni, S.O. Los, and C.J. Tucker. Satellite-based identification of linked vegetation index and sea surface temperature anomaly areas from 1982-1990 for Africa, Australia and South America. *Geophysical Research Letters*, 23(7):729–732, 1996. (Cited on pages 3 and 4.)

[5] J. Cho, P.J.-F. Yeh, Y.-W. Lee, H. Kim, T. Oki, S. Kanae, W. Kime, and K. Otsuki. A study on the relationship between Atlantic sea surface temperature and Amazonian greenness. *Ecological Informatics*, 5:367–378, 2010. (Cited on page 3.)

[6] D. Chen, M.A. Cane, A. Kaplan, S.E. Zebiak, and D. Huang. Predictability of El Niño over the past 148 years. *Nature*, 428(6984): 733–736, 2004. (Cited on page 3.)

[7] N. Marwan, M.C. Romano, M. Thiel, and J. Kurths. Recurrence plots for the analysis of complex systems. *Physics Reports*, 438: 237–329, 2007. (Cited on pages 3, 25, 26, 28, 33, 34, 38, 41, 42, and 46.)

[8] M.H. Trauth, B. Bookhagen, N. Marwan, and M.R. Strecker. Multiple landslide clusters record Quaternary climate changes in the northwestern Argentine Andes. *Palaeogeography, Palaeoclimatology, Palaeoecology*, 194:109–121, 2003. (Cited on pages 3 and 25.)

[9] A. Yadav, V.K. Jayaraman, M. Kale, and U. Kulkarni-Kale. Phylogenetic Clustering of Protein Sequences Using Recurrence Quantification Analysis. *Advanced Science Letters*, 19(5):1336–1339, 2013. (Cited on pages 3 and 25.)

[10] S.A. Brian, L.P. Salamanca, J. Whitehill, J.S. Reilly, M.S. Bartlett, D. Angus, and J. Wiles. Using recurrence plots to visualize the temporal dynamics of tutor/student interactions. In *2012 IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL)*, 2012. (Cited on pages 3 and 25.)

[11] S. Dia, S. Fontaine, and M. Renner. Using Recurrence Plots for Determinism Analysis of Blade–Disk Tribometer. In *Proceedings of the SEM Annual Conference*, 2009. (Cited on pages 3 and 25.)

[12] S.C. Li, Z.Q. Zhao, and F.Y. Liu. Identifying spatial pattern of NDVI series dynamics using recurrence quantification analysis. *European Physical Journal Special Letters*, 164:127–139, 2008. (Cited on pages 3 and 25.)

[13] American Meteorological Society. Glossary of Meteorology. URL `http://glossary.ametsoc.org/wiki/Teleconnection`. 25.03.2013. (Cited on page 4.)

[14] C.F. Ropelewski and M.S. Halpert. North American precipitation and temperature patterns associated with the El Niño/Southern Oscillation (ENSO). *Monthly Weather Review*, 114:2352–2362, 1986. (Cited on page 4.)

[15] C.F. Ropelewski and M.S. Halpert. Global and regional scale precipitation patterns associated with the El Niño/Southern Oscillation. *Monthly Weather Review*, 115:1606–1626, 1987. (Cited on page 4.)

[16] K.S. Casey, T.B. Brandon, P. Cornillon, and R. Evans. The Past, Present and Future of the AVHRR Pathfinder SST Program. In V. Barale, J.F.R. Gower, and L. Alberotanza, editors, *Oceanography from Space: Revisited*. Springer, 2010. (Cited on page 9.)

[17] K. Kilpatrick, G.P. Podesta, M. Arbelo, R. Evans, V. Halliwell, and J. Brown. Errors in high-latitude SSTs and other geophysical products linked to NOAA-14 AVHRR channel 4 problems. *Geophysical Research Letters*, 30(11):1548, 2003. (Cited on page 9.)

[18] P.H.C. Eilers. A Perfect Smoother. *Analytical Chemistry*, 75(14): 3631–3636, 2003. (Cited on pages 11, 20, and 83.)

[19] E.T. Whittaker. On a New Method of Graduation. *Proceedings of the Edinburgh Mathematical Society*, 41:63–75, 1923. (Cited on page 11.)

[20] P.M. Atkinson, C. Jeganathan, J. Dash, and C. Atzberger. Intercomparison of four models for smoothing satellite sensor time-series data to estimate vegetation phenology. *Remote*, 123:400–417, 2012. (Cited on page 11.)

[21] C. Atzberger and P.H.C. Eilers. A time series for monitoring vegetation activity and phenology at 10-daily time steps covering large parts of South America. *International Journal of Digital Earth*, 4(5):365–386, 2011. (Cited on page 11.)

[22] F.J. Kriegler, W.A. Malila, R.F. Nalepka, and W. W. Richardson. Preprocessing transformations and their effects on multispectral recognition. *Proceedings of the Sixth International Symposium on Remote Sensing of Environment*, 1:97–131, 1969. (Cited on page 17.)

[23] M.S. Rasmussen. Assessment of millet yields and production in northern Burkina Faso using integrated NDVI from the AVHRR. *International Journal of Remote Sensing*, 13:3431–3442, 1992. (Cited on page 17.)

[24] S.M.E. Groten. NDVI-crop monitoring and early yield assessment of Burkina Faso. *International Journal of Remote Sensing*, 14: 1495–1515, 1993. (Cited on page 17.)

[25] L.S. Unganai and F.N. Kogan. Drought monitoring and corn yield estimation in Southern Africa from AVHRR data. *Remote Sensing of Environment*, 63:219–232, 1998. (Cited on page 17.)

[26] C.J. Tucker, J.E. Pinzon, and M.E. Brown. NDVI 1985–2006. In *Global Inventory Modeling and Mapping Studies*. Global Land Cover Facility, University of Maryland, 2004. (Cited on page 18.)

[27] C.J. Tucker, J. Pinzon, M.E. Brown, D. Slayback, E.W. Pak, R. Mahoney, E. Vermote, and N. Saleous. An Extended AVHRR 8-km NDVI Data Set Compatible with MODIS and SPOT Vegetation NDVI Data. *International Journal of Remo*, 26(20):4485–4498, 2005. (Cited on page 18.)

[28] J. Pinzon, J. Brown, and C.J. Tucker. *Satellite time series correction of orbital drift artifacts using empirical mode decomposition*, volume 5 of *Interdisciplinary Mathematical Sciences*. World Scientific Publishing Co. Pte. Ltd., 2005. (Cited on page 18.)

[29] G. A. F. Seber. *Multivariate Observations*. John Wiley & Sons, Inc., 1984. (Cited on page 20.)

[30] H. Spath. *Cluster Dissection and Analysis: Theory, FORTRAN Programs, Examples*. Halsted Press, 1985. (Cited on page 20.)

[31] B. Georgescu P. Meer. Edge detection with embedded confidence. *IEEE Trans. Pattern Anal. Machine Intell.*, 23:1351–1365, 2001. (Cited on page 20.)

[32] P. Meer C. Christoudias, B. Georgescu. Synergism in low-level vision. In *16th International Conference on Pattern Recognition*, volume IV, pages 150–155, 2002. (Cited on page 20.)

[33] P. Meer D. Comanicu. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Machine Intell.*, 24:603–619, 2002. (Cited on page 20.)

[34] H. Poincaré. Sur la problème des trois corps et les équations de la dynamique. *Acta Mathematica*, 13:1–271, 1890. (Cited on page 25.)

[35] J.-P. Eckmann, S. Oliffsonn Kamphorst, and D. Ruelle. Recurrence Plots of Dynamical Systems. *Europhysics Letters*, 4(9):973–977, 1987. (Cited on pages 25, 26, 27, 33, and 38.)

[36] G.M. Mindlin and R. Gilmore. Topological analysis and synthesis of chaotic time series. *Physica D*, 58(1–4):229–242, 1992. (Cited on page 28.)

[37] M. Koebbe and G. Mayer-Kress. Use of recurrence plots in the analysis of time-series data. In M. Casdagli and S. Eubank, editors, *Proceedings of SFI Studies in the Science of Complexity, vol. XXI*, pages 229–242, Reading, MA, 1992. Addison-Weasly. (Cited on page 28.)

[38] J.P. Zilbut and C.L. Webber Jr. Embeddings and delays as derived from quantification of recurrence plots. *Physics*, 171(3–4):199–203, 1992. (Cited on pages 28 and 45.)

[39] J.P. Zilbut, J.-M. Zaldivar-Comenges, and F. Strozzi. Recurrence quantification based liapunov exponents for monitoring divergence in experimental data. *Physics Letters A*, 297:173–181, 2002. (Cited on page 28.)

[40] M. Thiel, M.C. Romano, J. Kurths, R. Meucci, E. Allaria, and F.T. Arecchi. Influence of observational noise on the recurrence quantification analysis. *Physica D*, 171:138–152, 2002. (Cited on page 28.)

[41] Wikipedia. Phase Space, 4 2013. URL `http://en.wikipedia.org/wiki/Phase_space`. (Cited on page 32.)

[42] E. Lorenz. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20(2):130–141, 1963. (Cited on page 33.)

[43] M.H. Trauth. *MATLAB Recipes for Earth Sciences*. Springer, 2006. (Cited on pages 33 and 34.)

[44] F. Takens. *Detecting strange attractors in turbulence*. Springer, Berlin, 1981. (Cited on page 34.)

[45] N. Marwan and J. Kurths. Nonlinear analysis of bivariate data with cross recurrence plots. *Physics Letters A*, 302(5–6):299–307, 2002. (Cited on pages 41 and 45.)

[46] N. Marwan, N. Wessel, U. Meyerfeldt, A. Schirdewan, and J. Kurths. Recurrence plot based measures of complexity and its application to heart rate variability data. *Physical Review E*, 66 (2):026702, 2002. (Cited on page 45.)

[47] C.L. Webber Jr. and J.P. Zilbut. Dynamical assessment of physiologica. systems and states using recurrence plot strategies. *Journal of Applied Physiology*, 76(2):965–973, 1994. (Cited on page 45.)

[48] J.-M. Robine, S. L. Cheung, S. Le Roy, H. Van Oyen, C. Griffiths, J.-P. Michel, and F. R. Herrmann. Death toll exceeded 70,000 in Europe during the summer of 2003. *Comptes Rendus Biologies*, 331 (2):171–178, 2008. (Cited on page 55.)

STATUTORY DECLARATION

I herewith declare that I have completed this thesis independently, using only the literature and aids specified in the bibliography. The thesis in this form or in any other form has not been submitted to any examination body.

Vienna, March 2014

Michael Wess

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography *"The Elements of Typographic Style"*. `classicthesis` is available for both LaTeX and LyX:

`http://code.google.com/p/classicthesis/`

Final Version as of March 18, 2014 (`classicthesis` Version 1.1).