MARSHALL PLAN FELLOWSHIP FINAL REPORT

Tri-Space Approach to Exploratory Visualization of Multispectral and Multivariate Imagery

By:

Timothy Schempp

San Diego State University, Department of Geography

San Diego, USA

M.S. Thesis Advisor:

Dr. André Skupin

San Diego State University, Department of Geography

San Diego, USA

Marshall Plan Scholarship Research Program Supervisors:

Dr. Gernot Paulus
Dr. Karl-Heinrich Anders

Carinthia University of Applied Sciences, Austria

Table of Contents

Table of Figures

List of Abbreviations

BMU – Best Matching Unit

EDA – Exploratory Data Analysis

KDD – Knowledge Discovery in Databases

MDS – Multidimensional Scaling

PCA – Principal Component Analysis

SOM – Self-Organizing Map

A_LT – Attribute over Locus and Time

AT_L – Attribute and Time over Locus

L_AT – Locus over Attribute and Time

LA_T – Locus and Attribute over Time

LT_A – Locus and Time over Attribute

T-LA – Time over Locus and Time

Tri-Space Approach to Exploratory Visualization of Multispectral and Multivariate Imagery

Abstract

Modeling the dynamic nature of phenomena over time is a powerful tool that can be used to further expand research and understanding of behavior. Continued development of methods and interfaces that explicitly visualize change within multitemporal and multivariate data support applications in many domains such as epidemiology, law enforcement, precision agriculture, environmental monitoring, and urban studies. The tri-space framework provides an alternative means of analyzing multivariate and multitemporal data by manipulating its structure to yield six distinct perspectives which can be visualized through dimension reduction techniques such as self-organizing maps (SOMs). Imagery analysis presents a compelling avenue for this methodology as it is a data source commonly used in a wide variety of environmental and urban studies that exploit its multitemporal and multivariate attributes. This research focuses on the conceptualization and development of software that encapsulates the tri-space approach in an exploratory data analysis environment to facilitate inductive investigation of imagery data. Implementing the software as a web-based platform further serves to increase accessibility and distribution capabilities to enable the extraction of insights. This study integrates and extends existing strands of collaborative research resulting from the Marshall Plan Foundation and San Diego State University which have contributed to this developmental project.

1 Introduction

In this era of big data, researchers face an ever-surmounting complexity in the quantity and dimensionality of their data due to technological advancements. This is the reality for research involving every conceivable type of multivariate and multitemporal data, including crime statistics, climate data, unstructured (text) data, and many others. Imagery data are no exception to this trend, especially as unmanned aerial vehicles (UAVs) continue to provide an abundance of low-cost, rapid acquisitions of imagery for monitoring phenomena. The multivariate nature of imagery relates to its *spectral resolution*, and the specific segments of the electromagnetic spectrum that are captured (Jensen 2007). The majority of remote sensing sensors capture multispectral imagery, which consists of anywhere from three to ten spectral channels. Newer state-of-the-art systems can capture hyperspectral imagery containing up to hundreds of thousands of spectral channels. Both of these data acquisition advancements demonstrate a substantial technological increase in volume and complexity, as higher spectral and temporal resolutions relate to data with higher dimensionality. Analysis techniques must also adapt to ensure that valuable insights are not obscured by this increase of data but are made obvious.

Classification maps are one of the most common and useful products that are derived by remote sensing analysts. This is possible by classifying the multispectral and hyperspectral attributes of an image pixel through either supervised or unsupervised classification (Jensen 2007). These maps serve as an important input for many spatial analysis procedures and as informative tools to aid decision-makers. While this process is familiar with any remote sensing analyst, it can act as a 'black box' in that there are many parameters, methods, and additional operations that are utilized before data creation or map production. After imagery has been transformed into a land cover raster file, this process is repeated at different times, creating a multitemporal series of land cover maps. These can be input to a change identification analysis which can explicitly show the changes that occur in between any two images within the dataset, e.g. between the oldest and newest images. The derived products can show pixels that experience *land cover change*, which actually represents a movement across the multispectral attribute space (Chen *et al.* 2003). These conventional analyses do not enable the explicit visualization of this movement. Instead, analysis typically classifies a pixel, where this project attempts to visualize a pixel's attribute trajectory to determine whether it's experiencing change or is remaining stable. For the case of environmental monitoring, this could yield insights related to degradation, pollution, or ecosystem change by observing trends, trajectories, and classification of pixels. The comparison of classification scheme outputs can also be impeded by a lack of ground truth or reference points, the inability to validate results.

These nuances suggest why new approaches are required to handle large quantities of data, detect patterns, and emphasize even subtle changes. The tri-space framework provides the means for analyzing and visualizing the behavior of these n-dimensional multitemporal data by manipulating its underlying structure to present six unique perspectives relating to its discrete objects, attributes, and temporal slices (Skupin 2010). These six perspectives are most effectively visualized using dimensionality reduction (DR) techniques, such as self-organizing maps (SOMs), multidimensional scaling (MDS) (Kohonen 1982; Torgerson 1952). This study sought to develop a web-based platform that synthesizes these conceptual and computational approaches to enable a flexible and accessible means of analyzing imagery data in an exploratory fashion. After development, applying this platform as a case study that explores different kinds of environmental monitoring data in different scales and resolutions provides a substantive aim, and establishes this project's relevance to geography.

2 Literature Review

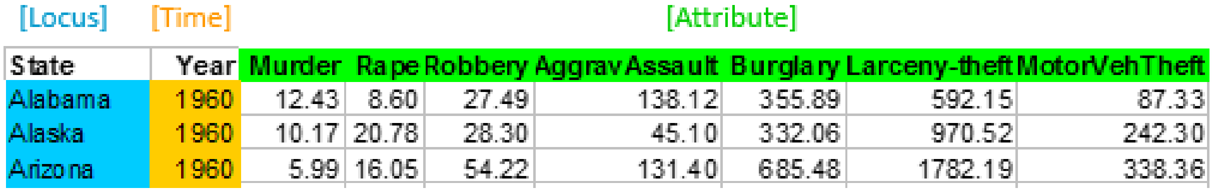
2.1 Conceptual Frameworks

This section of the literature review describes the conceptual frameworks that are utilized in this study. These different frameworks are integrated through software development as an engineering task to conduct this study's research-oriented goals. It is important to explain these frameworks in detail to establish where this research resides in the realm of image analysis and its relevance.

2.1.1 Tri-Space Framework

The tri-space framework and recent studies employing this methodology currently utilize discrete spatiotemporal data in structured and unstructured forms, which are typically represented by tabular datasets. These studies have explored the relationships and dynamics within topics such as demographics, air pollution, and snow water equivalency (Skupin & Hagelman 2005; Kolovos *et al.* 2010; Wang *et al.* 2013). Conversely, imagery can be considered as a subset of continuous geographic data, which is not typically stored in a tabular format rather inside of a two-dimensional array. In order to extend this framework to build this study's proposed imagery analysis tool, a number of components must be synthesized. The conceptual approaches of tri-space and exploratory data analysis (EDA) are integrated into an interactive, data-driven visualization environment (Skupin 2010; Tukey 1997). This is enabled with the computational approach of self-organizing maps (SOMs) and supported with visualization of a parallel coordinate plot (Kohonen 1982; Inselberg 1985). The application of tri-space to any spatiotemporal dataset can utilize visualizations SOMs, a type of artificial neural network, or other dimensionality reduction techniques such as principal component analysis (PCA) or multidimensional scaling (MDS) to produce up to six unique perspectives with each one providing improved opportunities for detecting change (Kohonen 1982; Pearson 1901; Torgerson 1952; Skupin 2010; Thompson 2017). While this framework has been applied to many kinds of multivariate and multi-temporal data, this research seeks to expand the framework into the imagery analysis domain (Lehrer 2013; Schabus 2013; Thompson 2017).

The tri-space is a conceptual approach that permits the manipulation of the identity, temporal, and attribute aspects of the data to produce a variety of unique perspectives on multivariate and multitemporal data. Tabular data structures and object-oriented structures can be used within the tri-space approach because they can explicitly organize and transform the data by manipulating these three different aspects. T tabular data structure shown in Figure 1 includes the three components of tri-space exist within spatiotemporal datasets as a locus, attribute, and time (Kolovos *et al.* 2010; Skupin 2010).


[Locus]
[Time]
[Attribute]

State	Year	Murder	Rape	Robbery	AggravAssault	Burglary	Larceny-theft	MotorVehTheft
Alabama	1960	12.43	8.60	27.49	138.12	355.89	592.15	87.33
Alaska	1960	10.17	20.78	28.30	45.10	332.06	970.52	242.30
Arizona	1960	5.99	16.05	54.22	131.40	685.48	1782.19	338.36

Figure 1. Tri-space components in tabular data (Skupin Unpublished, with permission)

The locus property typically relates to a distinct entity, either geographic or non-geographic. In previous research, the locus has represented states with multitemporal crime data attributes, raster cells of pollution monitoring, and raster cells of snow water equivalent attribute data

(Skupin & Hagelman 2005; Skupin 2010; Kolovos *et al.* 2010; Wang *et al.* 2013). In this study, the loci represent the geographic extents of pixels or cells within a given imagery dataset. The other fields associated with a particular record refer to time and attribute spaces, or in the contest of this study are the different multispectral channels multiple temporal stages. The six high-dimensional perspectives that can be derived include (Skupin 2010):

- *Locus* and *time* **over** *attribute* (LT_A)
- *Time* **over** *locus* and *attribute* (T_LA)
- *Attribute* and *time* **over** *locus* (AT_L)
- *Locus* **over** *attribute* and *time* (L_AT)
- *Locus* and *attribute* **over** *time* (LA_T)
- *Attribute* **over** *locus* and *time* (A_LT)

The simultaneous view of all perspectives can facilitate a greater comprehensive understanding of data behavior (Skupin 2010; Thompson 2017). These six perspectives when analyzed individually, offer major differences in terms of insights available. For example, the *LT_A* perspective highlights diverging and converging trajectories of demographic data (Skupin & Hagelman 2003). Additionally, the *L_AT* perspective enables the observation of the broad similarities between loci (image pixels) in the dataset. Applying the perspective of *AT_L* to imagery data would demonstrate the dynamic relationships among the multispectral channels across time. The *LT_A* perspective would explicitly show temporal trajectories of loci across the attribute space (Figure 2). De-aggregating the *LA_T* perspective into component planes through dynamic selection would yield differences according to which pixel and multispectral combination were selected. By presenting these different perspectives simultaneously, a meaningful exploration of how pixels and their attributes change over time and space can occur. Depending on the normalization scheme used, relationships between the magnitudes behavior can be explored. Dynamic selection by a user is a critical component, as their analysis can yield further questions and insights from observing the data in these different perspectives in a platform that supports the EDA framework. Normalization is another important consideration, as previous research has focused on normalization within the *LA_T* perspective which enhances the temporal signatures of the dataset (Thompson 2017). This normalization technique may also be the most appropriate for the study at hand, but normalization to all of the other perspectives was implemented.

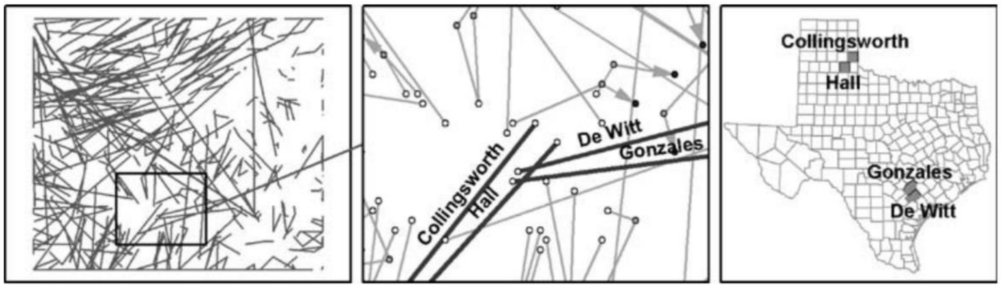


Figure 2. Converging and diverging LT_A trajectories (Skupin & Hagelman 2005)

2.1.2 Exploratory Data Analysis (EDA)

EDA embodies graphic display and visualization of data results; as statistics alone can be inadequate in describing data behavior (Tukey 1977; Anscombe 1973). For example, descriptive statistics super-imposed upon visualizations can yield deeper insights and be more meaningful to a researcher than a table of statistical results, especially when observing trends over time and space. EDA has also been employed by researchers in the discipline of geography, as a way to facilitate inductive research questions about the data (Andrienko & Adrienko 2006). Analyzing temporal data benefits from EDA methods as dynamically constraining and filtering the data at various temporal scales and ranges can present meaningful patterns. EDA contributes a framework for synthesizing the interaction between cognition and computation through the dynamic linking of data through multiple perspectives (Xu *et al.* 2006). To evaluate a tool which utilizes the EDA framework, subject-matter experts are the most qualified to assess its effectiveness. Their familiarity with the underlying data allows them to conduct more robust analysis with the tool, in order to confirm truths or generate hypotheses through inductive reasoning (Tukey 1977). The standards and principles of this framework will be adhered to during the user interface development stage.

2.1.3 Knowledge Discovery in Databases (KDD)

Traditional approaches to data analytics for extracting knowledge and insights have not scaled in order to match the complexity and quantity of the data that is generated from satellites, smartphones, UAVs, and the wide variety of other sensors available. Knowledge discovery in databases (KDD) provides a procedural framework for deriving meaning from raw data through five main steps: selection, preprocessing, transformation, data mining, interpretation, and evaluation (Frawley *et al.* 1991; Ester *et al.* 1995; Fayyad *et al.* 1996). This framework represents the overall approach that this research utilizes to uncovered nuanced trends and relationships within a multitemporal layer stack of imagery data, it is illustrated in Figure 3. It is important to note that process is not linear, rather constantly refined through interaction and many iterations (Fayyad *et al.* 1996).

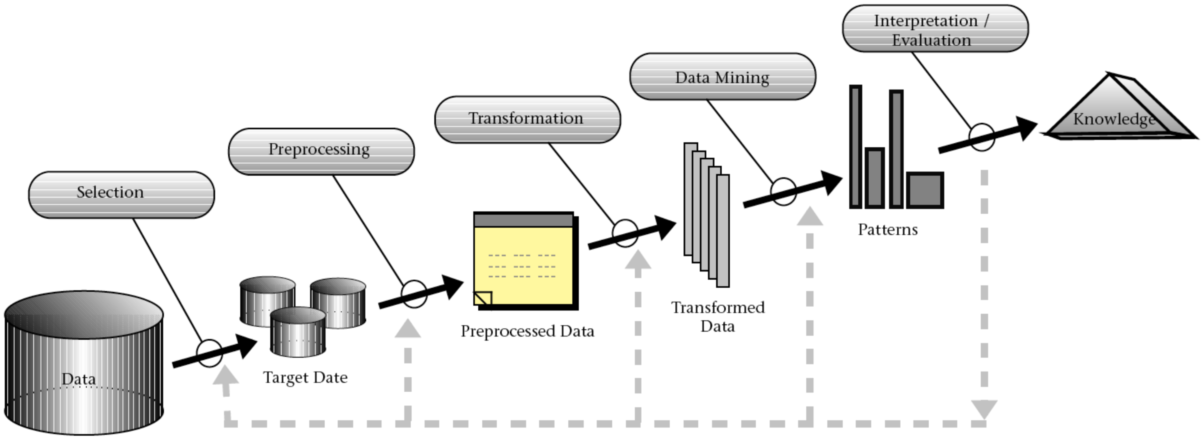


Figure 3. The KDD process (Fayyad et al. 1996)

2.2 Computational Approaches

Dimensionality reduction provides an array of *holistic* approaches to analyzing high-dimensional data. It includes a variety of techniques such as principal components analysis (PCA), multidimensional scaling (MDS), isomaps, topic modeling, and machine learning neural networks like self-organizing maps (Pearson 1901; Torgerson 1952; Tenenbaum, de Silva, & Langford 2000; Papadimitriou *et al.* 1998; Kohonen 1982). These are also commonly used in remote sensing, for classification of hyperspectral imagery. These different techniques' ability to analyze high-dimensional data provide one set of approaches for visualizing the six tri-space perspectives.

2.2.1 Self-Organizing Maps (SOMs)

Self-organizing maps (Figure 4) can automatically process, organize, and visualize multivariate data across an abstract space where proximity and distance are metrics of similarity across n-dimensions, as opposed to geographic space (Kohonen 1982). Samples of the dataset are used to generate the structure of the neural network; this refers to the *training* or m*achine learning process* (Kohonen 1990). One primary assumption of these methods is that human interpretation is required either prior to SOM-use (supervised) or after (unsupervised) in order to extract insights, so in-depth knowledge of the data and subject of interest is critical. Software libraries which simplify usage procedures can generate an atmosphere of uncertainty and skepticism surrounding these methods and their results. Familiar approaches to geography, like *labeling*, need to be considered carefully as they may not be appropriate to data in these high-dimension spaces. While SOM techniques have been used with imagery data, they are traditionally used for unsupervised classification of cell clusters in a single image (Zhong *et al.* 2006). This study will perform classification across all images in the time series. SOMs provide a means of displaying all six tri-space perspectives simultaneously, which yields a unique series of visualizations displaying the imagery cells moving through hyperspectral space. They are most useful when trying to organize and visualize a large number of input vectors if a sparse number of input vectors are encountered a different DR technique such as multidimensional scaling (MDS) may be preferable.

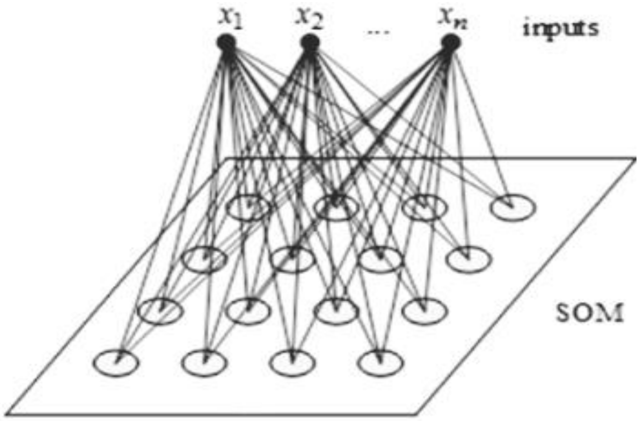


Figure 4. SOM structure (Kurasova & Molyte 2011)

2.2.2 Multidimensional Scaling (MDS)

Multidimensional scaling is a nonlinear DR technique that has optimal conditions for visualizing n-dimensional data (Torgerson 1952). If there are a relatively small number of input vectors, e.g. ten or twenty input vectors instead of thousands, the resultant visualization is more interpretable than a SOM. Comparisons can be observed by the proximity of similar to a scatter plot, it operates similarly to a scatter plot, as when only dealing with two dimensions it behaves in the exact same way. Objects are placed in the resultant 2-D space relate to the similarity between all dimensions of attribute data (Torgerson 1952). Figure 2 demonstrates how different techniques might be best used to visualize the different tri-space perspectives. Instead of all six outputs displaying SOMs, the bottom-left and bottom-right panes which contain a sparse number of input vectors utilize MDS. While MDS and SOMs both rely on topology to visualize comparisons, sometimes it is useful to look at the plotted data something that is easily implemented for n-dimensional data with a parallel coordinate plot.

2.2.3 Parallel Coordinate Plots

Parallel coordinate plots provide the capability to generate a two-dimensional visualization of complex multivariate data that could prove useful as an auxiliary perspective for exploring imagery data (Inselberg 1985; 1998). Spectral signature analysis within the domain of remote sensing commonly applies a similar strategy of visualization to show reflectance or digital number value signatures across axes that represent continuous hyperspectral and multispectral bands within the data. Conversely, parallel coordinate plots involve distinct variables that are plotted across the x-axis in a parallel fashion (Inselberg 1985). It permits one to observe vector values across many wavelengths or other variables that associated with each cell or cell region (Figure 5). Implementing this technique with interactive selection is an important step towards amplifying cognition when a user is exploring both pixels and neurons within the attribute space by acting as an auxiliary visual for interpreting and contextualizing the resultant tri-space perspectives.

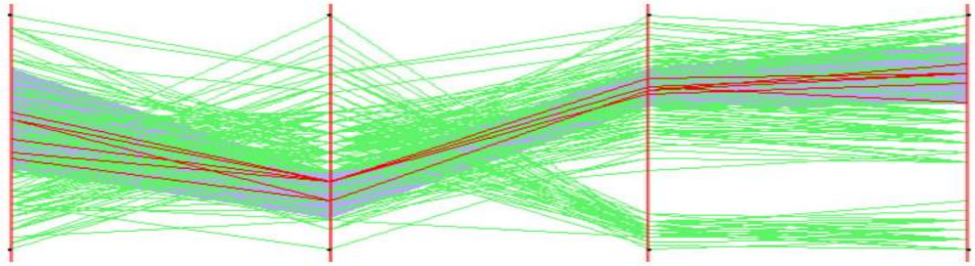


Figure 5. Parallel coordinate plot with four dimensions (Siirtola & Räihä 2006)

2.2.3 K-Means Clustering

Clustering techniques provide a means to enhance the user's ability to interpret the SOM and MDS outputs. Because the two-dimensional outputs are actually showing topological relationships across high-dimensional space, clustering supports cognition by creating boundaries and classes within these spaces. K-Means clustering is a popular algorithm that is commonly used for performing clustering on geographic data. Because SOMs and MDS outputs are topologic structures, k-means can be applied to its symbology to find non-obvious groupings of input vectors or neurons. The *SOMatic Viewer* implementation of the K-Means algorithm is used in this project, as it relies on many of the class definitions that already exist in the rest of *SOMatic* (Rainer 2013). This will be used to find clusters and perform classification on the neurons in the SOMs.

2.3 Integrating Methods

Software libraries reflect the majority of developments in recent years for SOM, MDS, and parallel coordinate plot usage and visualization as they provide a means for distribution and access to these approaches for researchers. There are many iterations and implementations of libraries that utilize SOM visualizations, but only the relevant ones to this project are discussed. The first SOM library, *SOM_PAK*, was originally developed through collective efforts from Kohonen (the original conceptualizer of SOM) and the Helsinki University of Technology. It is now a public-domain library that contributes the basis for most if not all future iterations. A modernized software library, *SOMatic*, has been developed through collaborative efforts between San Diego State University and Carinthia University of Applied Sciences (Rainer 2013; Spöcklberger 2013; Thompson 2017; Kowatsch 2017). It is a Java library that has been designed for a specialized tri-space approach to multi-perspective data visualizations. This project utilizes *SOMatic* for generating SOM visualizations, as it has been improved to handle the capability of generating the six simultaneous visualizations. Finally, a *tri-space* library has been created to handle the different transformations of data into the different tri-space perspectives. Input data can be in any perspective and the library outputs the data transformed into the other five perspectives. The Space-Time-Attribute Analysis tool created by Diansheng *et al.* served as an example of how one might achieve the integration of these different approaches into a single interface (2006). *Parcoords.js* is a JavaScript library that effectively implements a parallel coordinate plot with embedded features such as axis querying and subsetting.

3 Study Area

This project focuses on a very small subset of San Diego County, the master-planned community known as San Elijo Hills (Figure 6). This subset was selected for this project for a variety of reasons, which all seek to reduce the complexity of the input data for the initial proof-of-concept developed throughout this project. This contiguous region has experienced a great deal of change during the three captured time periods – 1993, 2003, and 2013. For software development purposes, a limited number of pixel objects was desired, the San Elijo Hills municipal area is comprised of 1,235 pixels which make for an adequate starting point in regards to the computational resources required to process and interact with the data.

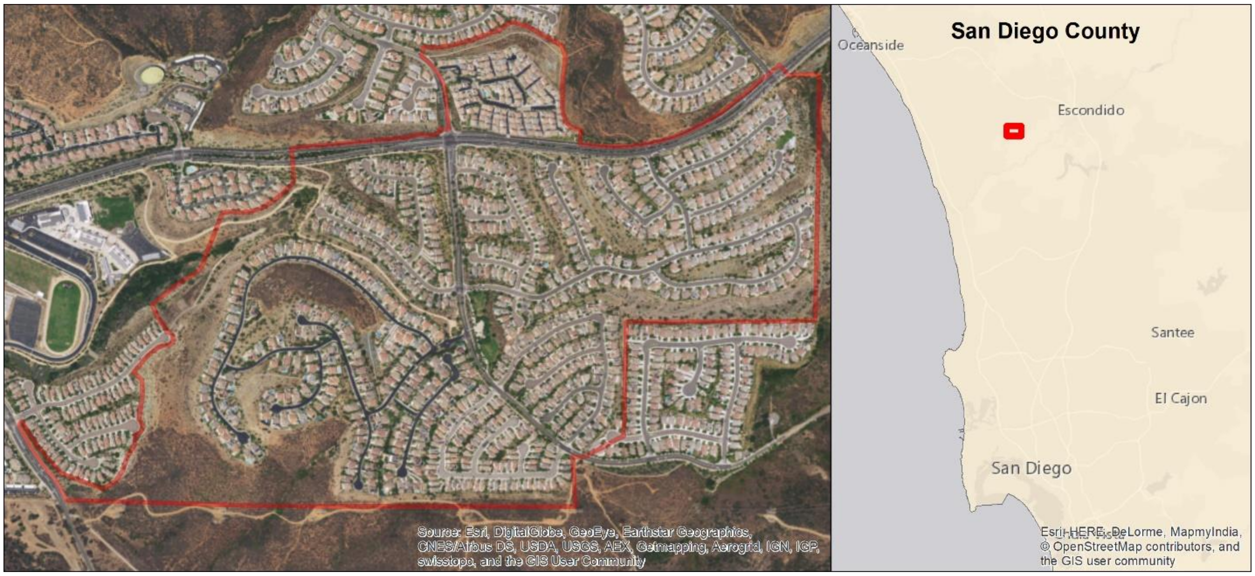

San Diego County
Oceanside
Escondido
Santee
El Cajon
San Diego
Source: Esri, DigitalGlobe, GeoEye, Earthstar Geographics, CNES/Airbus DS, USDA, USGS, AEX, Getmapping, Aerogrid, IGN, IGP, swisstopo, and the GIS User Community
Esri, HERE, DeLorme, MapmyIndia, © OpenStreetMap contributors, and the GIS user community

Figure 6. San Elijo Hills, San Diego study area

4 Research Questions

This research aims to develop a web-based platform that can perform the processing, distribution, and visualization of imagery data in accordance with the tri-space framework. Two main research questions are pursued in the course of this project.

5.1 How can the tri-space framework be extended into the imagery analysis domain?

Answering this question relates to the workflow schema that must be designed which takes input image data and converts it into all the other tri-space perspectives. To build a reusable, generalizable platform, this process must be highly automated and conscious of computational resources. Finding a balanced level of parallelization in regards to processing time and performance will be investigated. Other considerations towards the user-interface design are required to answer the interactive visualization component of the platform. Finally, the capability to project different classification schemes onto trained SOMs create insightful overlays which show pixel trajectories through classifier boundaries has been integrated. This describes how the frameworks and methods are all implemented.

5.2 What insights can be derived from imagery analysis utilizing tri-space?

This question seeks to answer the substantive components of the research. A look into what insights can be extracted from each of the tri-space perspectives is performed. This explores how the spatial and temporal resolution characteristics of imagery data impact the platform. Also, discovering how do different tri-space normalizations can impact results.

5.3 How can implementing tri-space into a web-based platform be accomplished?

One goal was to determine what geographic scales this approach could be reasonably applied. This was something encountered in different stages of the methodology workflow, some reducing the scales of analytics. This describes what issues are resolved through deploying technological solutions and data subsetting.

5 Data

The approach taken in this project has attempted to keep all workflows, procedures, and subroutines as generalizable as possible. While the technology has not been fully developed by the end of this project, most processes but the selection/naming of input data and the generation of the leaflet layers objects in the proof-of-concept's interface have been implemented.

5.1 Platforms and Sensors

Imagery data is generated by a wide variety of platforms and sensors that are utilized to accomplish different kinds of goals. Common platforms include satellites, airplanes, UAVs, handheld, or even fixed structural locations. For implementation and testing, Landsat satellite platforms and their data products were primarily used as a functional and standardized data source.

Landsat 8 contains eight visible bands, a panchromatic band, and two thermal bands. Landsat 7 contains six visible bands, a panchromatic band, and one thermal band. USGS provides Landsat data in three kinds of forms: raw data, top-of-atmosphere reflectance, and surface reflectance. In the context of this project's goals to detect land cover change, the surface reflectance option was utilized as this set of visible light bands are corrected to remove the interference from atmospheric irradiance while providing additional spectral indices. These outputs are composed of these bands, of which have different spatial resolutions: 15m^2 for the panchromatic channel, 100m^2 for the thermal bands (resampled to 30m^2), and the rest are 30m^2.

UAVs and their sensors do not adhere to any enforceable standards other than weight limitations. These are traditionally equipped with any kind of sensor from a standard RGB digital camera to that of a hyperspectral sensor for precision agriculture (CITATION). These are increasingly becoming the preferred way to capture high-resolution imagery in a quick deployment fashion. This translates to them having high temporal resolution, as repeat flights can be taken within the same day, hour, or minutes depending on the number of platforms, power usage, and the size of the study area.

5.2 Standards and Considerations

It is important that all input data for analysis by this tool utilizes the same geographic extent and pixel size – *spatial resolution*. In order to create unique objects from the pixels in the L_AT, LT_A, and LA_T perspectives, a pixel must be consistent in the geographic space it corresponds to across all time slices. Ideally, the images should be registered to each other meaning they are geometrically rectified.

Another important consideration is the spectral and radiometric resolution of the sensors that generate the input data. This project's approach to analysis should come from data sets that utilize consistent sensors, at the same time of day, the same day of the year if performing an annual study, and preferably with the same amount of cloud cover and radiation within the atmosphere. This limits much of the variation that can influence the detection of change within a pixel across all of its bands. This means that even the difference between using Landsat 7 and Landsat 8 data in the same model would influence results, as their band designations actually cover different ranges within the electromagnetic spectrum. Similarly, the radiometric resolutions relate to the depth of the data that a sensor can capture. Using sensors that have different radiometric depths, makes their data incompatible with each other unless the more detailed data is scaled down to the lower resolution data, but this can also influence results.

5.3 Formats

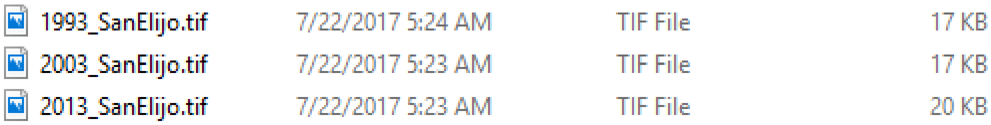
Imagery data comes in in a wide variety of formats. But one of the most common formats that both satellite and UAV data come in is that of Geographic Tagged Image File Format (GeoTIFF). There are also formats like Network Common Data Form (NetCDF) and or Hierarchical Data Format (HDF) that are suited for multitemporal datasets. Because the open source GDAL library package can natively process GeoTIFF files, and its widespread use as a standard format for both satellite and UAV output data, it was chosen as the primary input data for the implemented workflows.

Another data standard utilized is that of Geographic JavaScript Object Notation (GeoJSON). This allows for Leaflet to act as the primary library that enables interactive, scalable visualization. It is important to note here that this also enforces the use of the Web Mercator projection, as this the default and standard projection used in web mapping applications. It enables the vector layers to be created in leaflet as points, polygons, and lines.

This study also looked at various options for improving the performance of the standard GeoJSON vector layers. Two potential solutions found include *VectorGrid*, a Leaflet plugin, and *TopoJSON*, a topology-enabled variant of GeoJSON. The VectorGrid converts a GeoJSON file into a vector tiles format, which is very similar to raster tiles in that it enables scalable detail which depends on the current extent. TopoJSON takes a different approach as it reduces redundancy by storing related geometry only once, which makes it seem like a very appealing solution for this project since all vector layers are either neurons in a hexagonal lattice or pixel extents in a rectangular grid. However, Leaflet extensions that utilize TopoJSON cannot visualize it natively, rather they convert a TopoJSON file into a GeoJSON file in the browser, and therefore the reduced encoding is not actually utilized. Although explored, neither were successfully implemented in the proof-of-concept, but these technologies offer an alternative solution to achieving improved performance and a reduction in the computational workload in the browser.

5.4 Naming Convention

This project utilizes a file naming convention for the original input image data that is used throughout the preprocessing and data mining procedures. This naming convention persists even in the proof-of-concept user interface (UI). The proof-of-concept only focuses on a dataset with relatively coarse temporal resolution, three time slices with a ten-year interval. This is reflected in the current implementation, but this section includes the current naming convention and the conceptualization for a more generalizable naming convention. The current implementation utilizes the following naming convention: *Year_Location.tif*. This is displayed in Figure 7.



1993_SanElijo.tif	7/22/2017 5:24 AM	TIF File	17 KB
2003_SanElijo.tif	7/22/2017 5:23 AM	TIF File	17 KB
2013_SanElijo.tif	7/22/2017 5:23 AM	TIF File	20 KB

Figure 7. Input data naming convention

This convention does not capture the higher temporal resolution aspects of data, this is something that must be addressed since this project would like to include UAV data sources in the next stage of its development. One consideration was to utilize the Landsat naming convention; however, this again only captures the year and the day, something that would not be applicable to hourly or more frequent UAV data collections. The naming convention settled upon would be to use the following convention: *Year.Month.Day.Hour.Minute_Location.tif*. This would allow input data to capture intervals as low as a minute or can use intervals of years. Users are not required to populate all temporal regions, i.e. *2003_SanElijo.tif* would still be a valid file name, as a tokenization split based off the '.' and '_' characters would only yield the year and location. If higher temporal attributes were provided, they would be captured in the tokenization and utilized.

6 Methodology

This section describes the project's methodological approach to development and implementation of the conceptual frameworks and computational approaches described in the literature review. The overall workflow presented utilizes the KDD framework as a logical procedure for knowledge discovery from multitemporal trends in imagery data. The input data used for this process was subsetted throughout the entire process, as the initial goal of almost an entire Landsat scene was too ambitious relative to computational resources.

6.1 Data Selection

The first step in the KDD framework is to perform a selection from the datasets that are available to the user. For this project, it relates to the selection criteria of Landsat 8 scene acquisition of the majority of San Diego County. Landsat satellites have a temporal resolution of sixteen days but are consistent with the time of day of their acquisitions. The Landsat 5 dataset used for the visualization in the proof-of-concept were all from April in the three separate years 1992, 2003, 2013.

6.1.1 Hydrologic Year

The Landsat 8 dataset used in the development of this project, adhered to the concept of a hydrologic year to capture temporal trends within the study area. A hydrologic year which begins on October 1st and ends September 30th is utilized in hydrology to measure changes in precipitation and vegetation. This was the basis for anniversary dates in the resultant multi-year dataset. The Landsat 8 dataset utilizes the following four dates as target anniversary dates for the entire dataset: October 1st, January 1st, April 1st, and July 1st. It begins with July 1st, 2013 and continues up until July 1st, 2017 as of the most recent data acquisition at the time of this project report. Obviously with the sixteen-day temporal resolution landing exactly on these dates over four years it not always possible, especially when considering uncontrollable interferences and obscurities such as cloud cover. Because this analysis tries to identify trends in the macro-temporal cycles across the dataset, there is a need to fix the micro-temporal cycles in an attempt to make anniversary dates actually line up. This concept is visited in section 6.2.2.

6.2 Data Preprocessing

The next step after selecting the data is to prepare it. There are many considerations one must make at this juncture. This section covers some of the assumptions and preprocessing procedures that the input data must be subjected to first before it can be transformed and processed with machine learning. Some of these processes may not be necessary or even possible with UAV data, but this project still includes them as Landsat datasets are a frequent source of data for researchers in many fields like geography, environmental science, and urban studies. This part still utilized both the Landsat 8 and 5 datasets.

6.2.1 Image Registration

Image registration, as previously mentioned in section 5.2, is essential for all of the input data for this process to work. First, there must be the same number of cells in all input images. Second, all cells must be of the same spatial resolution and cover the same geographic extent.

If this is not the case, then one must use resampling and georectification tools to generate a clean registration between all input images in the dataset. This is not an issue for Landsat datasets as all the images come pre-registered and always occupy the same geographic extents with the same spatial resolution. This is more of an issue when using UAV datasets for the tri-space approach. This must be considered and dealt with in a manual fashion as automation of this step is not within the scope of this project.

6.2.2 Image Interpolation

In order to achieve anniversary dates for the Landsat 8 dataset, an image interpolation technique was implemented. Linear interpolation was achieved using the following equation:

$$y_{target} = y_{pre} + \frac{(x_{target} - x_{pre})(y_{post} - y_{pre})}{(x_{post} - x_{pre})}$$

The *x* values refer to the number of days across the images with x_2 being the target date, and the *y* values would correspond to the array of reflectance values. Therefore, the set of values represented by y_2 would reside somewhere in between the y_1 and y_3 values. This serves two main purposes:

1. To minimize the micro-temporal cycles in the dataset to achieve anniversary dates.

2. To remove/reduce the influence of cloud cover in the images.

While the implementation of this technique was not necessary for the dataset used in the proof-of-concept, it was an important step to incorporate into the preprocessing workflow to manage future research involving data from Landsat or other satellite platforms. As completely clean full-scene Landsat data is uncommon, this approach was made scalable in that it can utilize an array of scenes both prior and after the target date for interpolation. So, scenes closest to the target date are utilized first, then if clouds are detected in either scene it iterates to the next or previous scene that is not obscured by cloud cover for that pixel and readjusts the equation parameters with the appropriate separation in days (Figure 8).

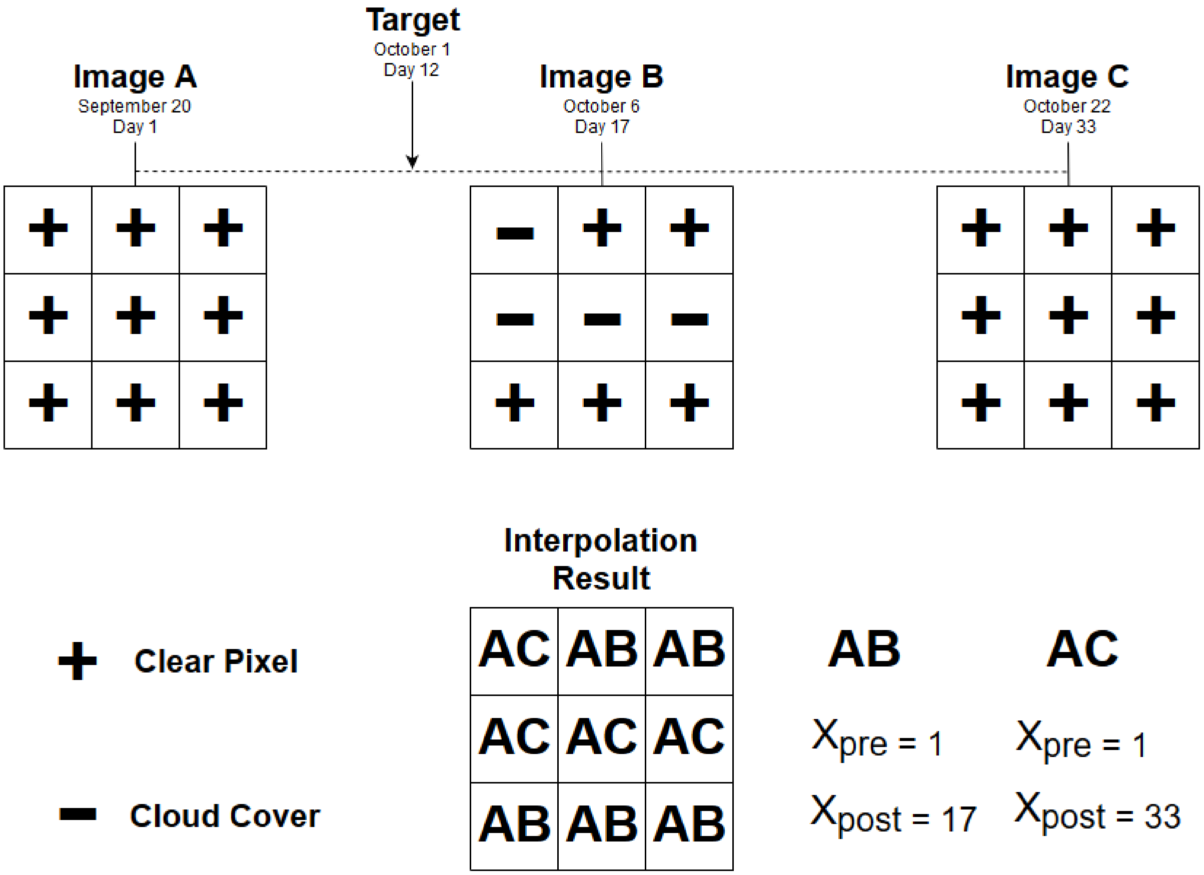


Figure 8. Image interpolation routine

6.2.3 Image Subsetting

Subsetting is a useful strategy when trying to reduce the volume of the data being analyzed. This step was utilized both in the Landsat 8 dataset, as well as in the Landsat 5 (proof-of-concept) dataset. For the full San Diego county study, all of the target images were subsetted to only include mainland San Diego, as well as have the eastern border equivalent to the smallest extent of all images. Because the focus of this study is to detect land cover change, including the huge number of pixels in the Pacific Ocean was deemed unnecessary and was excluded from the final Landsat 8 dataset. Similarly, in the San Elijo dataset only the pixels that were found within the municipal boundary were included in the analysis (Figure 9).

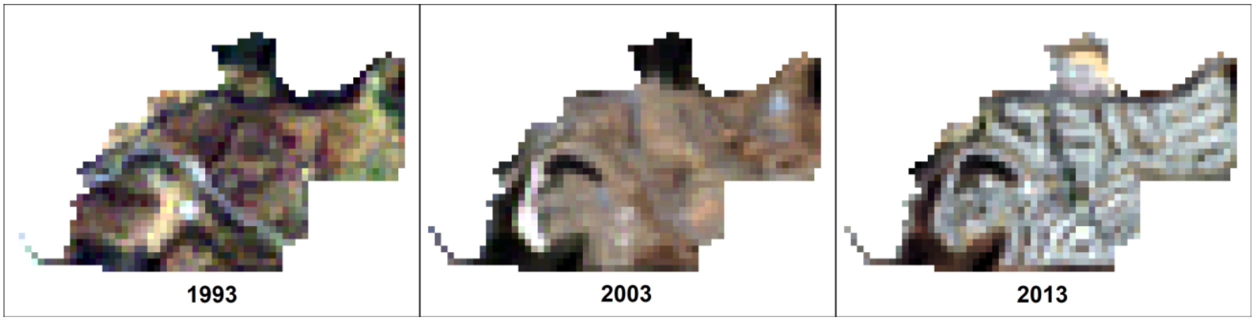


Figure 9. San Elijo boundary subset from Landsat

6.2.4 Band Selection

As discussed in section 5.1 there are many different bands available for use with the Landsat 8 platform. For the Landsat 8 dataset, the following bands were used in preprocessing: surface reflectance bands 1-7, band 9, and band 10. Band 8 was not utilized as it represents the panchromatic part of the spectrum, therefore it would be redundant to include. Band 11 was not included because of USGS recommendations in the Landsat 8 documentation which states that it is more susceptible to contamination from the atmosphere than Band 10. The spectral indices such as normalized difference vegetation index (NDVI), enhanced vegetation index (EVI), and others were considered, but it was determined with consultation with project advisors that solely focusing on the data that captures only the electromagnetic spectrum would be utilized. The San Elijo dataset also utilizes surface reflectance bands, specifically bands 1-5, and band 7. Band selection has been implemented as a parameter in the preprocessing software, so this parameter can be manipulated to include or exclude certain bands based on what is important to each study conducted.

6.3 Data Transformation

The next step after preparing the data through preprocessing is to transform it into a tabular format that can be analyzed through machine learning. The data must also be transformed into a format that it can be easily visualized in. This project utilizes Leaflet for the primary means of visualization, so these include converting images into raster tiles and GeoJSON formats for vector data. This is where an initial issue with the feasibility of geographic scales was encountered, the Trispace.jar library cannot handle large (1M+) tabular datasets. So only the Landsat 5 data was utilized after section 6.3.1. A subset of Bataquitos Lagoon and San Elijo containing 90,955 and 1,235 pixels respectively.

6.3.1 GeoTIFF(s) to CSV Conversion

The first input data in the processing workflow comes as GeoTIFF files. These must adhere to the naming convention described previously in section 5.3. In order to first prepare the data for its usage in *SOMatic*, it must be converted into a tabular data structure, then it can be transformed into all the Tri-Space perspectives. The GDAL function *gdal2xyz* is used to accomplish the conversion from a GeoTIFF image file into a CSV file. Each pixel in the GeoTIFF becomes a row in the CSV and its columns contain the pixel centroid and all of the values associated with each band. This procedure then parses through this CSV and removes the rows that do not contain data, as these do not provide any meaning when used to create the SOM. The final step is to merge all of the resulting CSV files into a single file (Figure 10).

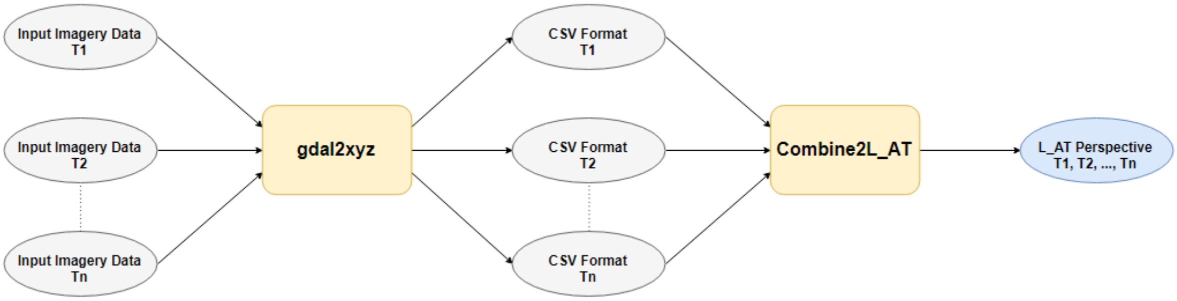


Figure 10. GeoTIFF to CSV conversion workflow

Each row in the table represents a pixel and the columns now reflect all band data across all time slices. This is the initial Tri-Space perspective that the data represents, *L_AT*. Each object in the table is a *locus* or a cell, and each of its attributes is related to a combination of a particular band and at a specific time.

6.3.2 Tri-Space Conversion

Once the data has been converted into one of the tri-space perspectives, it is then converted to the remaining five perspectives (Figure 11). Each of these perspectives hold the same data, but in different structures, as explained in section 2.1.1. These result in a number of different objects, which are all compared and analyzed with *SOMatic*. The current version of the Trispace.jar library expects the input perspective to be L_AT, it also cannot operate on datasets that have over one million records.

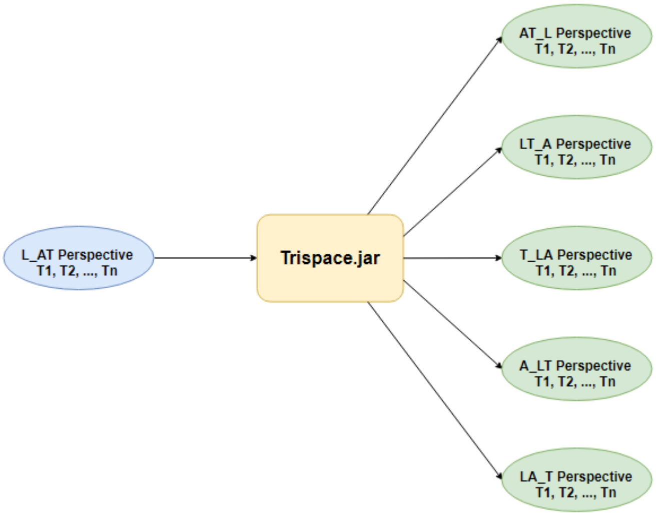

L_AT Perspective T1, T2, ..., Tn
Trispace.jar
AT_L Perspective T1, T2, ..., Tn
LT_A Perspective T1, T2, ..., Tn
T_LA Perspective T1, T2, ..., Tn
A_LT Perspective T1, T2, ..., Tn
LA_T Perspective T1, T2, ..., Tn

Figure 11. Trispace.jar converts between perspectives initialized with L_AT

This conversion does not perform any normalization, so performing a normalization for each perspective must occur. Using the raw numbers as inputs in *SOMatic* does not yield meaningful results. This study encountered two primary ways to implement this procedure, the first is the processing approach where an entire tabular dataset is loaded into memory with Processing’s *Table* class, then calls are made to each row in memory and they are transformed to a different perspective, this is how the software is currently implemented. The second approach considered for this process is utilizing the more traditional Java approach of line-by-line reading and writing. While the current implementation works well for a subset of the data, the Java approach is better suited for large datasets as not all entities are stored in memory, rather they are written to a file. The trispace.jar library also converts the perspectives into a normal tabular format, as well as a .DAT file that is directly used as input to *SOMatic*.

6.3.3 Tri-Space Normalization

In order to produce meaningful comparisons in *SOMatic* normalization must first take place. Because there are six perspectives each with their own normalization, this results in a total of thirty-six combinations of normalizations and perspectives. The procedure in the data transformation workflow has been implemented and is illustrated in Figure 12.

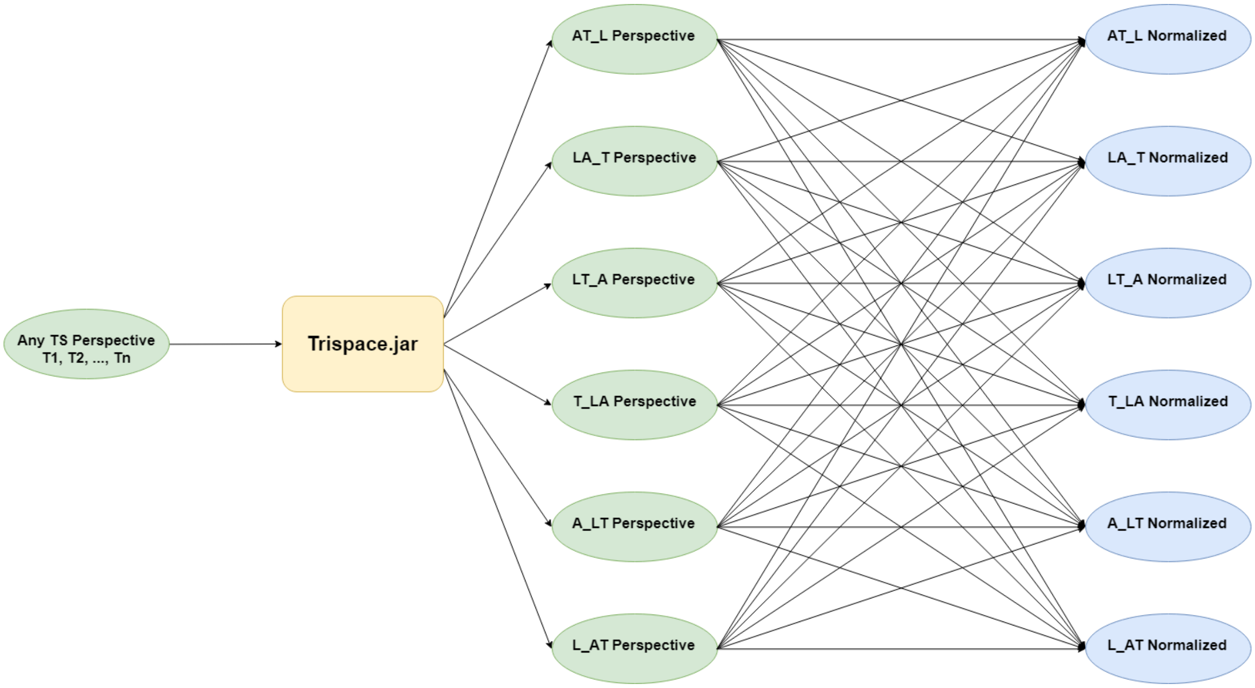


Figure 12. Six perspectives normalized each way result in 36 possible combinations

The feature scaling normalization technique was used to normalize each perspective which results in data that scales as a decimal number between 0 and 1. It is represented by the following equation:

$$x_{normalized} = \frac{(x - x_{min})}{(x_{max} - x_{min})}$$

Normalization to a particular perspective focuses in on a particular characteristic of the data and helps magnify different kinds of trends within the data (Figure 13). Part of this research is to explore what effect these different normalizations have when analyzing imagery data.

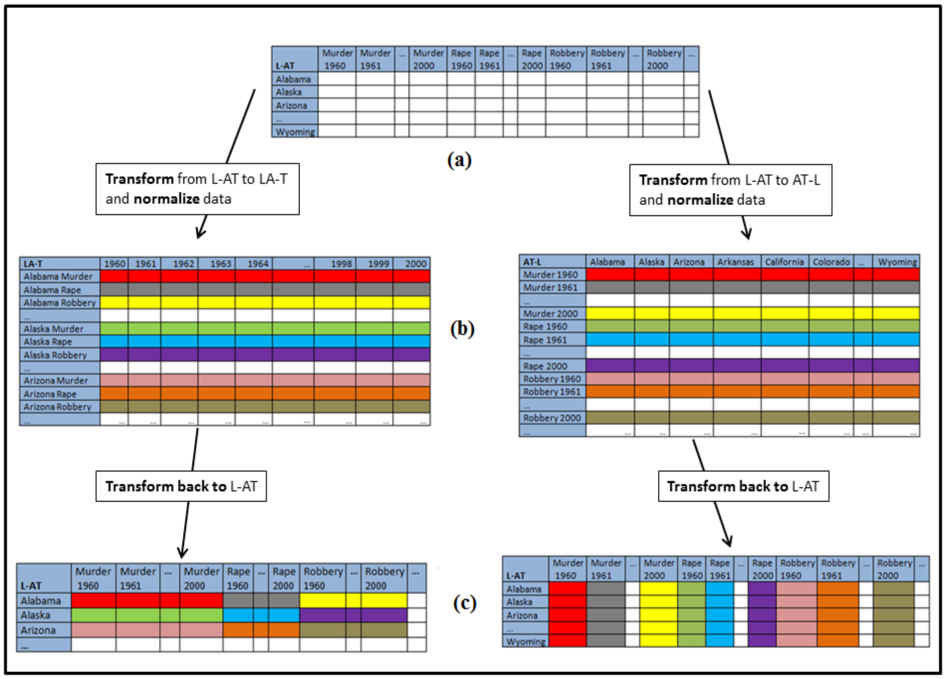


Figure 13. Different normalization highlights different trends in the data (Thompson 2017)

6.3.4 GeoTIFF to Leaflet Formats

Leaflet provides the means to visualize the imagery on a geographic map. But Leaflet cannot directly utilize GeoTIFFs, rather it can utilize raster tiles or image overlays. The first format, raster tiles, preserve the geographic coordinates of the imagery, meaning one can easily change between different basemaps and have context that is typically provided with all web mapping applications. A potential drawback of this format is that it typically requires larger storage space, especially as the number of scenes and band combinations is increased. Raster tiles also require a conversion, this can be achieved both through GDAL with the *gdal2tiles* function or with a software called *TileMill* which is a two-step process to get the resultant raster tiles. A raster tile is a hierarchical folder structure that relates to each zoom level and exponentially increases in file size as the zoom levels are increased. The image overlay format is more efficient to store, but it does not have geographic coordinates so it cannot be easily switched out with other basemaps ending up with less context. It also has some performance benefits, as it does not have to load in multiple files; it can zoom and scale a single image file as opposed to switching between them for different zoom levels.

6.3.5 GeoJSON

GeoJSON is utilized to display vector layers in Leaflet. Vector objects are used in all seven of the Leaflet panes, six of them for the tri-space perspectives and one for the geographic map. These objects are used to enable interactivity with the user, it allows for the selection of objects which also select corresponding objects in the other perspectives. The imagery data must be converted to a GeoJSON to allow for the user to select a pixel in the image, otherwise, Leaflet does not allow for direct interaction with a raster tile. This is accomplished by processing the data in ArcMap and using the *Fishnet* tool to generate a vector surface that corresponds to the pixels in the scene (Figure 14). The SOMs in the tri-space perspectives utilize GeoJSON for visualization in Leaflet. A useful output of *SOMatic* allows for direct transformation into a GeoJSON format of the SOM (Kowatsch 2017).

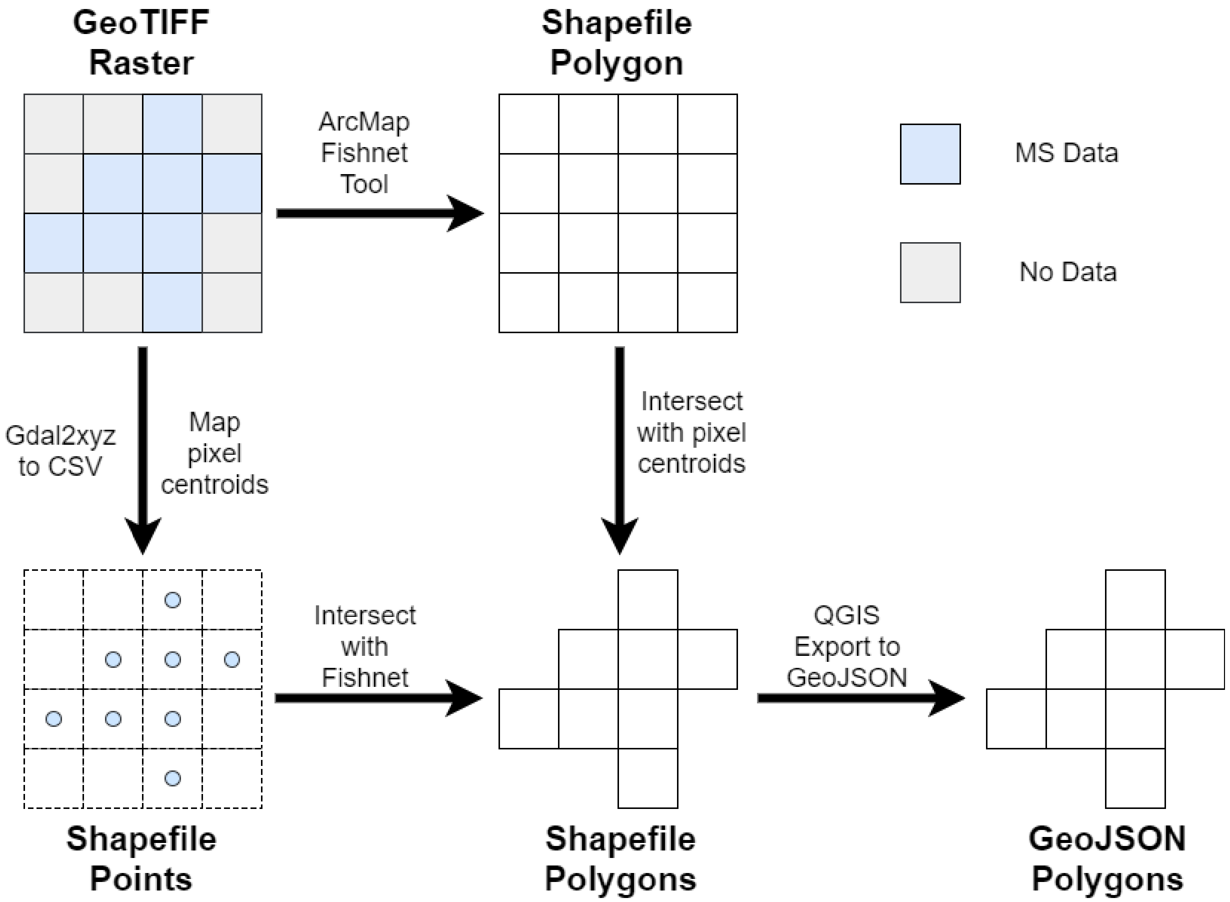


Figure 14. Deriving GeoJSON extents from a GeoTIFF

6.4 Data Mining

Data mining presents the most computationally complex part of the KDD workflow. For this project, it involves using machine learning and dimensionality reduction techniques to develop SOMs. These are further explored by performing k-means cluster analysis on them to derive more meaning. This section describes this process.

6.4.1 SOMatic

Section 3.3.3 describes what the input data to *SOMatic* looks like. At this stage, the original input imagery and transformed it into thirty-six unique tables which correspond to each tri-space normalization times each tri-space perspective. The main parameters that must be considered for implementation relate to the size of the SOM, the size of the neighborhood, the number of cores for parallelization, and the similarity measure.

SOMatic outputs a .COD file as well as a .SPRJ file that contains the data about the input parameters, the neurons and the SOM itself. Enhancements to *SOMatic*, have also enabled the direct output of GeoJSON files so they can be easily visualized in Quantum GIS (QGIS) or Leaflet as this project uses (Kowatsch 2017).

Another previous project that is the first attempt at a tri-space enabled analytical tool is that of *SOMGrid* (Thompson 2017). Some functionality from this project was utilized to handle the finding of the best matching unit (BMU) for each of the input vectors. The input vector to BMU (one to one) matching is saved in a tabular format, as part of enabling interactivity in the proof-of-concept. Conversely, the BMU to input vector table (one to many) was also saved in a similar fashion to also push the interactive capability of the prototype.

6.4.2 K-Means Clustering

In this same workflow, a subroutine was implemented that performs k-means clustering on all of the generated SOMs. This functionality is originally developed in the *SOMatic Viewer* but has a modified parameter for *k* (Rainer 2013). Instead of performing solely the calculation for a k number of classes, it iteratively decrements from a user-chosen parameter for *k* to *k = 2*. It stores all this information in additional columns in the BMU table. Additionally, the sum of squared errors (SSE) table is stored separately to allow for visualization of this metric to while the user is selecting an appropriate number for *k*. It is important to use the same similarity measure parameter as was used for the generation of the SOM, which in the case of the proof-of-concept was Euclidean distance.

6.5 Data Interpretation

This is the final step in the KDD framework, it enables an analyst to interpret the results from the data mining computations. Visualization is a catalyst of successfully providing the means for interpretation. While statistics and machine learning quality metrics are also useful for extracting insights, they can also be visualized which makes their understanding more intuitive and comparable. This section discusses visualization techniques and how interactivity can drive more meaningful ways of interpretation through the EDA framework.

6.5.1 Data Visualization

As mentioned in previous sections, Leaflet is the primary tool used to deploy the geographic map and all of the tri-space perspectives in the UI. This allows for scalable extents with zooming, panning, and selection functionality already built-in. Here only the San Elijo Landsat 5 scenes of 1993, 2003, and 2013 were used as a feasible data source.

The geographic map contains the imagery in raster tile or image overlay formats, in addition to the underlying vector representations of each pixel that enable interactivity. Deciding what bands to include on a particular image depends on storage space and level of detail the analyst expects while pursuing analysis. In the case of the proof-of-concept, eight images were made available: true color, false color, and panchromatic single-band images for all six bands. The ability to incorporate custom band combinations would be ideal, but this can end up too computationally expensive as the number of possible images would then be 6^3, or 216 potential combinations. For this reason, it was not pursued in the course of this project.

Each of the tri-space perspectives utilized Leaflet as well for their display and to provide an interactive selection. This relies upon the GeoJSON export from *SOMatic* to visualize the SOMs. Leaflet makes it easy to symbolize each neuron according to either the number of input vectors mapped to it as well as the k-means class it corresponds to. Some of the sparser input vector sets may be better visualized with MDS but have only been implemented as SOMs for this project. Also, it easy to re-symbolize and change the underlying data depending on which normalization is selected. The underlying data is also provided in a table view, where it is possible to visualize both the input vector tables and the BMU tables.

Also included in the visualization is a time slider, which allows for the manipulation of the data within geographic map while highlighting the appropriate selection in the temporal tri-space perspectives. Finally, a parallel coordinate plot acts as an auxiliary display to view either the pixels in the imagery or to observe the different input vector objects for each tri-space perspective.

6.5.2 Interactivity

Interactivity is one of the key elements to creating an EDA-enabled visualization environment (Tukey 1977). This section goes into detail on each element in the UI and what effect interactivity should have on other elements.

The time slider is one of the most intuitive elements in the UI. Its interactivity directly manipulates which layers are available/displayed in the Leaflet geographic map. Another consideration is what should happen in the temporal-related tri-space perspectives. It certainly makes sense to highlight the appropriate neurons in *T_LA* because each neuron literally represents a particular time, but there is the case of when a neuron corresponds to more than a single time input vector which already throws some obscurity into the mix. There was also consideration towards if this should highlight neurons that correspond to that time in the *AT_L* and *LT_A* perspectives since each object in these perspectives directly relates to a particular time on the slider.

The geographic map can be thought of the primary display element in the UI. It has a direct association with the *L_AT*, *LT_A*, and *LA_T* tri-space perspectives. Original conceptualization envisioned that a two-way selection was the most logical method of implementation, but this also suffers from the same notion of pixels to neurons being a one to many relationship, where clicking one neuron may highlight multiple pixels, while one pixel only highlights a single neuron. Overcoming this obscurity with selection options may be the most intuitive way around this issue. Also, for the interaction with *LT_A* trajectories (the same pixel moving across all bands could be one option, or limiting it to whatever bands are currently visible in the geographic map makes sense. between them.

In all of the tri-space perspectives, the first letter or two letters that come before the underscore indicate the object for that perspective. Each object actually represents an input vector which corresponds to a neuron in the SOM as a one-to-many relationship. It is important to understand this concept as the interactivity with each perspective is detailed.

The first tri-space perspective refers to *L_AT* where each object relates to a pixel or cell in the imagery, and its attributes are all the bands at all time slices, which depend on the original input data.

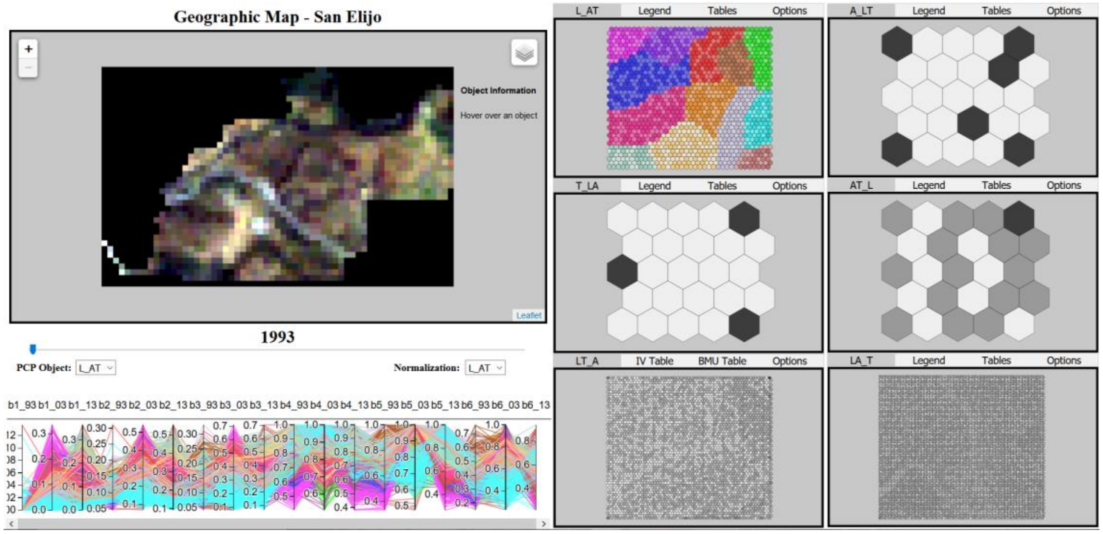
7 Results

This section describes the result of the project which comes as a working proof-of-concept visualization application. Screenshots of the UI are included as well as a description of its functionality and analytical capability. While not all perspectives symbology and interactivity were implemented, *L_AT* was focused on to demonstrate how further development would make the other perspectives more interpretable.

7.1 Proof-of-Concept

The proof-of-concept yielded by this project is a visualization application that can be run locally or placed on a server to facilitate a distributed platform for analysis. While much of the project was focused on developing a standardized workflow for preprocessing and processing, the proof-of-concept only visualizes the results. In addition to simply being a desktop/web application, it also serves as an expansion of the framework for Tri-Space UI design as it has more analytical capabilities built into it than *SOMGrid*. Figure 15 shows the UI that has been designed over the past summer, which integrates geographic space and the six Tri-Space perspectives as Leaflet panels in addition to the parallel coordinate plot. From this initialized state one can see that only *L_AT* has been implemented with k-means clustering capability. The SOM perspectives utilize a tabbed CSS structure to contain tabular data and options.

The UI provides an EDA-enabled environment for a user to interactively start to explore the data and find trends. The interactivity between the geographic map and the different perspectives enhance a user's cognition as they investigate individual or multiple pixels as well as clusters of neurons (Figure 16).


Geographic Map - San Elijo

1993

PCP Object: L_AT

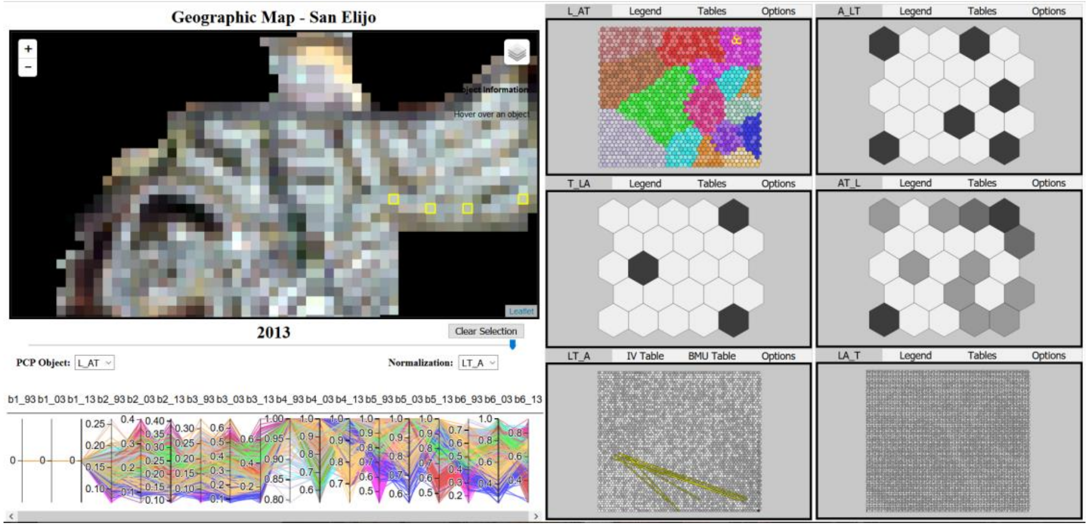
Normalization: L_AT

Object Information
Hover over an object
Leaflet
L_AT
Legend
Tables
Options
A_LT
T_LA
AT_L
LT_A
IV Table
BMU Table
LA_T
b1_93 b1_03 b1_13 b2_93 b2_03 b2_13 b3_93 b3_03 b3_13 b4_93 b4_03 b4_13 b5_93 b5_03 b5_13 b6_93 b6_03 b6_13

Figure 15. Tri-Space Image Analysis UI

7.1.2 Functionality and Capability

The user is able to simultaneously display all six Tri-Space perspectives and can explore different kinds of patterns within the data through controls and options. The time slider provides a way to manipulate the temporal aspects of the UI, in the current implementation it solely affects which images are displayed on the geographic map. Another option includes which normalization one desires to investigate, this is currently a global operation – it changes all 6 SOMs and the parallel coordinate plot data into whatever normalization perspective is selected.


Geographic Map - San Elijo

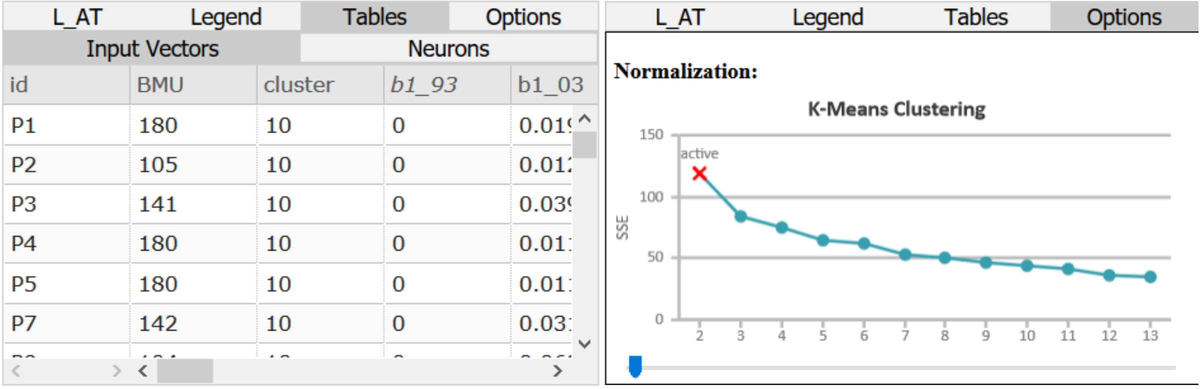
2013

PCP Object: L_AT

Normalization: LT_A

Figure 16. Interacting with pixels and neurons

This will be stored in each Leaflet pane's options window in future iterations. A tabular view of the data has also been implemented using the *SlickGrid.js* library (Figure 17). Another useful option is an adjuster for *k* in the *k-means* classes, users are provided with a graph of the sum of squared errors (SSE) and *k* that they can manipulate and change the SOM symbolization (Figure 17). Figure 18 demonstrates the results of changing the value for k.



id	BMU	cluster	*b1_93*	b1_03
P1	180	10	0	0.019
P2	105	10	0	0.012
P3	141	10	0	0.039
P4	180	10	0	0.011
P5	180	10	0	0.011
P7	142	10	0	0.031

Figure 17. Table view and k-means options

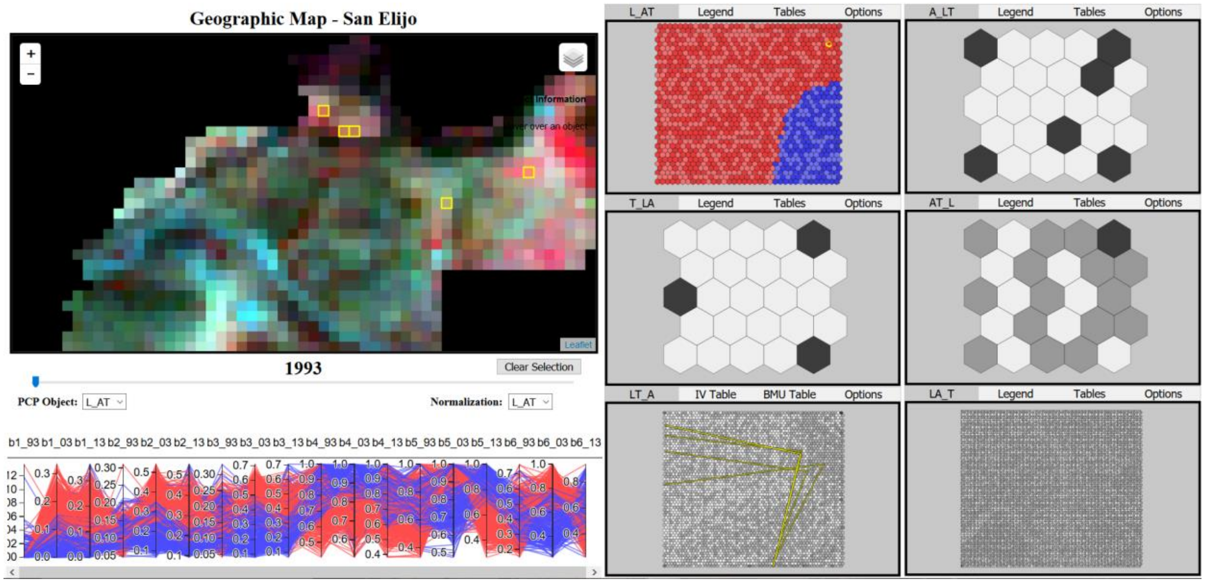


Figure 18. False-color 1993 layer with k value of 2

As mentioned in section 6.5.2 interaction has been implemented in various ways. Highlighting a pixel in the geographic map will highlight the corresponding neuron in *L_AT* and the trajectory in *LT_A* in a one-to-one relationship. Selecting a neuron in *L_AT* will highlight one or more pixels in the geographic map, and show all trajectories of selection in *LT_A*. Another interactive feature relates to the table of contents in the geographic map. Selecting a particular panchromatic band image will select the corresponding neuron in *A_LT* (Figure 19).

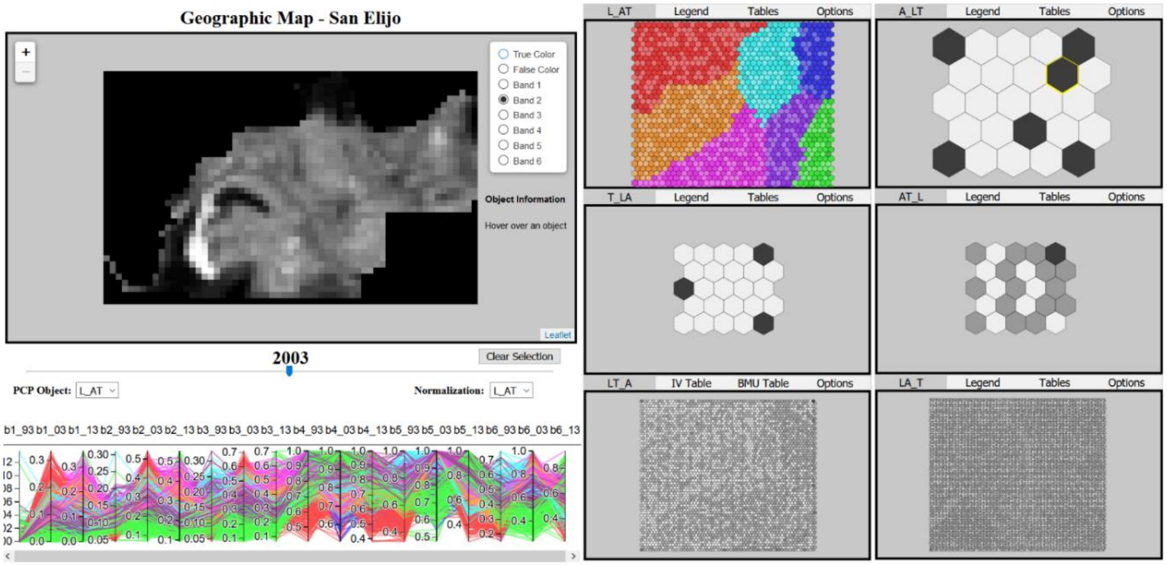

Geographic Map - San Elijo
○ True Color
○ False Color
○ Band 1
● Band 2
○ Band 3
○ Band 4
○ Band 5
○ Band 6
Object Information
Hover over an object
Leaflet
2003
Clear Selection
PCP Object: L_AT
Normalization: L_AT
L_AT Legend Tables Options
A_LT Legend Tables Options
T_LA Legend Tables Options
AT_L Legend Tables Options
LT_A IV Table BMU Table Options
LA_T Legend Tables Options

Figure 19. Selection of panchromatic band image

7.1.2 Testing and Validation

The section focuses on the validation of processing results and understanding performance. The following performance benchmarks are utilized to identify bottlenecks in the current version of the web application. The developer tools found in Google Chrome were utilized to generate Figure 20. One can see that outside of the initialization of the web application, normalization, and rescaling the Leaflet panes are the largest bottlenecks encountered in the proof-of-concept. Normalization is the most substantial and potentially avoidable bottleneck as it determines the symbology of every SOM in all 6 Leaflet panes, as well as replaces all the data in the tables and parallel coordinate plot. This is something that should be addressed with off-screen loading of states. The next largest bottleneck relates to panning and zooming of the high-density SOMs such as *LT_A*. Deploying a technology solution such as vector tiles may be the answer to improving this bottleneck.

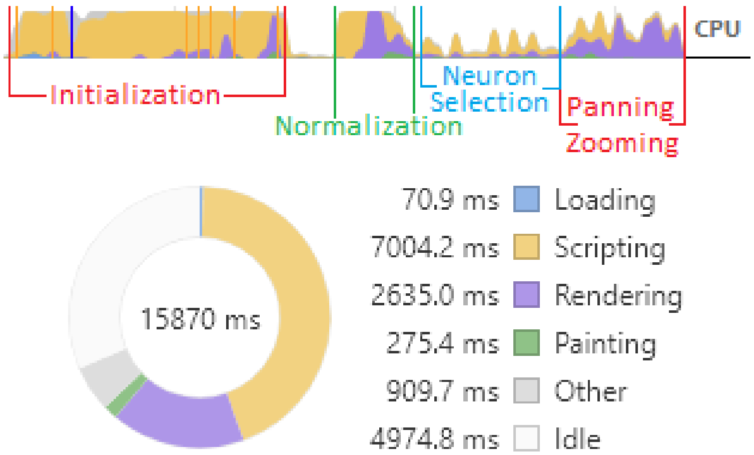
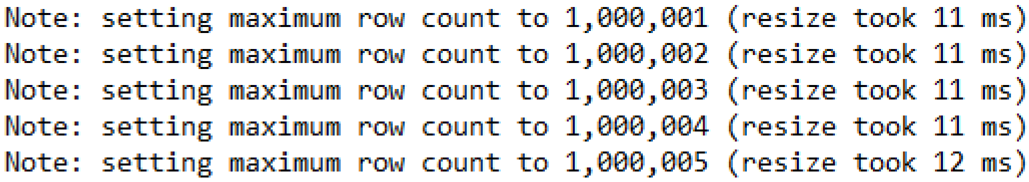


Figure 20. Google developer tools performance metrics

8 Conclusion, Limitations, and Future Work

Although the resultant proof-of-concept of this project was not able to fully realize all of the project goals that will be further pursued in the course of the thesis work, it did serve as an important expansion of the Tri-Space framework in regards to developing an interactive EDA-based web platform. It also yielded many important lessons that must be addressed in the future iterations of this project. Most issues that arose in the process of this study were associated with utilizing very large amounts of data in both the processing and visualization components. The proof-of-concept enables some analytical insights when looking for patterns within the San Elijo area in 1993, 2003, and 2013. One is able to link geographic space with the compressed attribute space visualized in *L_AT*. Also, one is able to visualize the trajectory for pixels across *LT_A* which facilitates exploring the temporal signatures of change of pixels. The current implementation permits one to observe the effects of using different normalizations and to overlay different granularities of k-means clustering onto the *L_AT* perspective.

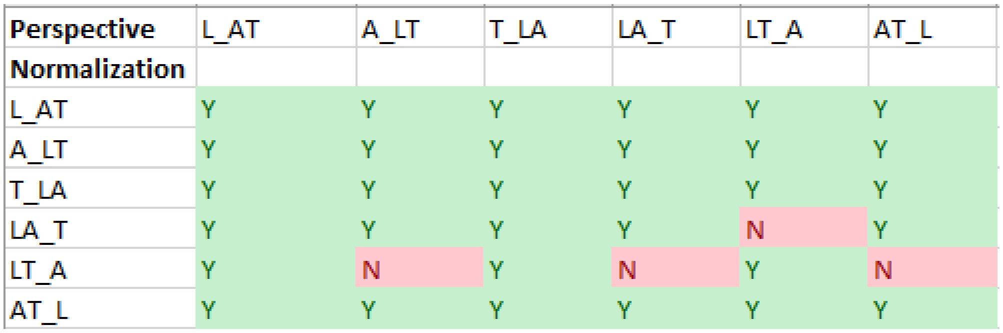
The current implementation is susceptible to many limitations, many of which are addressed in the following paragraph about future work. Limitations discussed here will appear in the same order as they did in the project workflow. The first limitation encountered related to how the TriSpace.jar library was implemented. Because it utilizes *Processing*'s native *Table* class, it stores an entire table into memory, this causes problems when utilizing very large datasets such as the entire Landsat 8 image of San Diego over 6 million pixels (Figure 21). This same issue was encountered when implementing the image interpolation procedure but was avoided by using the standard Java approach of *BufferedReader* and *FileWriter* classes, which continuously write data to file rather than hold it in memory. A solution that will avoid both of these procedures is to utilize a database to store these large quantities of data as they should neither be held in memory or in a local CSV file.



```
Note: setting maximum row count to 1,000,001 (resize took 11 ms)
Note: setting maximum row count to 1,000,002 (resize took 11 ms)
Note: setting maximum row count to 1,000,003 (resize took 11 ms)
Note: setting maximum row count to 1,000,004 (resize took 11 ms)
Note: setting maximum row count to 1,000,005 (resize took 12 ms)
```

Figure 21. Console print out when surpassing the Table limit (1,000,000)

The next limitation occurred in the SOM processing routine, where the all-zero vector was encountered. Cosine similarity is a more robust metric, however, testing encountered an error related to an "all-zero vector" that occurs in some of the normalization/perspective combinations (Figure 22). These result in a division by zero undefined error, something that must be addressed in an improved version of *SOMatic*'s cosine similarity functionality, which when encountering a division by zero will convert the zero into a very low value such as 0.00001. This project avoided this issue by using Euclidean distance for the similarity measure parameter.



Perspective	L_AT	A_LT	T_LA	LA_T	LT_A	AT_L
Normalization						
L_AT	Y	Y	Y	Y	Y	Y
A_LT	Y	Y	Y	Y	Y	Y
T_LA	Y	Y	Y	Y	Y	Y
LA_T	Y	Y	Y	Y	N	Y
LT_A	Y	N	Y	N	Y	N
AT_L	Y	Y	Y	Y	Y	Y

Figure 22. Table of normalization/perspective combinations containing all-zero vectors

The next limitation is related to performance in the proof-of-concept when displaying a large vector layer in Leaflet. This was eluded to in section 5.3, as other more efficient formats were investigated. TopoJSON for reasons described earlier, cannot actually be utilized by Leaflet as of version 1.2.0, rather it converts the format back into GeoJSON in the browser, which increases the computational workload instead of reducing it. VectorGrid does actually provide a potential solution to this issue, but the scaling parameter when outputting the SOM GeoJSON was not explored in totality. The primary issue faced when using VectorGrid was that the neurons were subjected to the Web Mercator projection's distortion in high latitudes. This can be overcome by adjusting the scaling factor in *SOMatic* to a much lower value so that this distortion does not occur.

This project will be continued in the form of a master's thesis which will be completed in May 2018. Here many of the limitations will be revisited, solutions will be re-implemented, and unexplored goals will be realized. First, overcoming the limitations encountered in the preprocessing and visualization components should be addressed. This includes utilizing a database for storage instead of local CSV tables, modifying the cosine similarity measure within *SOMatic*, and utilizing a more efficient means to interact with vector data in Leaflet. These changes should enable to proof-of-concept to be able to utilize much larger datasets, such as an entire Landsat 8 scene. This will facilitate more meaningful analysis and permit validation by comparison to other Landsat studies. Using UAV data sources would likely result in more challenges but would yield interesting results on how Tri-Space can detect change and trends in much higher spatial resolution data. Another important step will be to incorporate the use of MDS vector layers as an alternative to SOMs for the sparse input vectors. A SOM is not the appropriate means of visualization when there are a relatively small number of input vectors. Deploying an application server that handles input/inference of imagery data would be another useful endeavor. This way one could submit appropriately named GeoTIFF files and could use REST calls to retrieve inference results without directly interacting with a database. Finally, the UI design should be revisited and re-implemented. A high-level web development framework such as Angular will allow for modular development and should increase performance through CSS optimization and data streaming.

Acknowledgements

I offer my sincerest gratitude to the Austrian Marshall Plan Foundation, the Department of Geoinformation at Carinthia University of Applied Sciences, Dr. Gernot Paulus, Dr. Karl-Heinrich Anders, and Dr. Piotr Jankowski for providing this research opportunity. I would also like to recognize Dr. André Skupin, my mentor and thesis committee chair, for the time, knowledge, and patience he contributed towards realizing the goals of my graduate education and this project.

Financial support for this work was provided by the Austrian Marshall Plan Foundation. Any opinions, findings, conclusions, and recommendations expressed in this material are those of the author and do not necessarily reflect the views of the Marshall Plan Foundation.

References

Agafonkin V. (2016). Leaflet. Open-source software. Retrieved from http://leafletjs.com/.

Andrienko N & Andrienko G. (2006). Principles. In Exploratory Analysis of Spatial and Temporal Data. Berlin, Heidelberg: Springer Berlin Heidelberg. 461–633. doi:10.1007/3-540-31190-4_5

Anscombe FJ. (1973). Graphs in Statistical Analysis. *The American Statistician*. 27(1): 17–21. doi: 10.1080/00031305.1973.10478966

Chen J, Gong P, He C, Pu R, & Shi P. (2003). Land-Use/Land-Cover Change Detection Using Improved Change-Vector Analysis. *Photogrammetric Engineering & Remote Sensing*. 4: 369-379.

Guo D, Chen J, MacEachren AM, & Liao K. (2006). A Visualization System for Space-Time and Multivariate Patterns (VIS-STAMP). *EEE Transactions on Visualization and Computer Graphics*. 12(6): 1461-1474. doi:10.1109/TVCG.2006.85.A

Frawley WJ, Piatetsky-Shapiro G, Matheus CJ, and Smyth P. (1991). Knowledge Discovery in Databases: An Overview. In *Knowledge Discovery and Data Mining*, edited by Piatetsky-Shapiro G and Frawley B (Cambridge, Mass: AAAI Press/The MIT Press). 1-27.

GDAL. (2016). GDAL - Geospatial Data Abstraction Library: Version 2.1.3, Open Source Geospatial Foundation, http://gdal.osgeo.org

Gray J, Chaudhuri S, Bosworth A, Layman A, Reichart D, & Venkatrao M. (1997). Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals. *Data Mining and Knowledge Discovery*. 1: 29-53.

Inselberg A. (1985). The plane with parallel coordinates. *The Visual Computer*. 1(2): 69–91.

Inselberg A. (1998). Multidimensional detective. In *IEEE Proceedings of Information Visualization*. 100–107.

Jensen JR. (2007). *Remote Sensing of the Environment: An Earth Resource Perspective*. Upper Saddle River, NJ: Prentice Hall.

Kohonen T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*. 43(1), 59-69.

Kohonen T. (1990). The self-organizing map. *Proceedings of the IEEE*. 78(9): 1464–1480. doi:10.1109/5.58325

Kolovos A, Skupin A, Christakos G, & Jerrett M. (2010). Multi-Perspective Analysis and Spatiotemporal Mapping of Air Pollution Monitoring Data. *Environmental Science & Technology*. 44(17): 6738–44. doi: 10.1021/es1013328

Kowatsch F. (2017). SOMatic Trainer 2.0: Improved Parallelization Concepts and GeoJSON Integration for a Self-Organizing Maps Java Implementation. Thesis (MSc). Carinthia University of Applied Sciences.

Kurasova O & Molyte A. (2011). Quality of Quantization and Visualization of Vectors Obtained by Neural Gas and Self-Organizing Map. *Informatica*, Lith. Acad. Sci., 22, 115-134.

Lehrer G. (2013). Visualization Tool for Tri-Space concept and SOM. Thesis (BSc). Carinthia University of Applied Sciences. Retrieved from https://static1.squarespace.com/static/559921a3e4b02c1d7480f8f4/t/585c293f9f7456577905ccba/1482434884654/Lehrer.pdf

MacQueen J. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkely Symposium on Mathematical Statistics and Probability*. 1: 281-297.

Papadimitriou CH, Tamaki H, Raghavan P, & Vempala S. (1998). Latent Semantic Indexing: A Probabilistic Analysis. *Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*. 159-168.

Pearson K. (1901). On Lines and Planes of Closest Fit to Systems of Points in Space. *Philosophical Magazine*. 2: 559-572.

Rainer M. (2013). SOMatic Viewer: Implementation of an Interactive Self-Organizing Map Visualization Toolset in Processing and Java. Thesis (MSc). Carinthia University of Applied Sciences. Retrieved from https://static1.squarespace.com/static/559921a3e4b02c1d7480f8f4/t/585c2976d482e9dc5dd40d85/1482434949219/Rainer.pdf

Schabus S. (2014). Spatio-Temporal Data Mining for Pattern Recognition in Production Line Processes. Thesis (MSc). Carinthia University of Applied Sciences.

Siirtola H & Räihä KJ (2006). Interacting with Parallel Coordinates. Interacting with Computers. 18(6): 1278-1309. doi:10.1016/j.intcom.2006.03.006

Skupin A & Hagelman R. (2003). Attribute Space Visualization of Demographic Change. In *Proceedings of the 11th ACM International Symposium on Advances in Geographic Information Systems*. 56–62. New York, NY, USA: ACM, 2003. doi:10.1145/956676.956684

Skupin A & Hagelman R. (2005). Visualizing Demographic Trajectories with Self-Organizing Maps. *GeoInformatica*. 9(2):159-179. doi:0.1007/s10707-005-6670-2

Skupin A. (2010). Tri-Space: Conceptualization, Transformation, Visualization. Sixth International Conference on Geographic Information Science, Zürich, Switzerland, September 2010.

Spöcklberger M. (2013). SOMatic Trainer: Implementation of a Self-organizing Map Tool with Parallel Training for Processing, Applied to Carinthian Municipalities Census Data. Thesis (MSc). Carinthia University of Applied Sciences. Retrieved from https://static1.squarespace.com/static/559921a3e4b02c1d7480f8f4/t/585c29992e69cffa3fd07343/1482434975774/Spoeckelberger.pdf

Tenenbaum JB, del Silva V, & Langford JC. (2000). A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*. 290(5500): 3219-2323. doi:10.1126/science.290.5500.2319

Thompson G. (2017). Exploratory Analysis of Longitudinal Data: Design and Implementation of a Tri-Space Solution. Thesis (MSc). San Diego State University.

Torgerson WS. (1952). Multidimensional scaling: I. Theory and method. *Psychometrika*. 17(4): 401-419.

Tukey J. (1977). *Exploratory Data Analysis*. Reading, Massachusetts: Addison-Wesley Publishing Company.

Wang N, Biggs T, & Skupin A. (2013). Visualizing Gridded Time Series Data with Self-Organizing Maps: An Application to Multi-Year Snow Dynamics in the Northern Hemisphere. *Computers, Environment and Urban Systems*. 39(5): 107-120. doi:10.1016/j.compenvurbsys.2012.10.005

Xu Y, Prather JW, Hampton HM, Aumack EN, Dickson BG, & Sisk TD. (2006). Advanced Exploratory Data Analysis for Mapping Regional Canopy Cover. *Photogrammetric Engineering and Remote Sensing*. 72(1): 31-38.

Zhong Y, Zhang L, Huang B, & Li P. (2006). An Unsupervised Artificial Immune Classifier for Multi / Hyperspectral Remote Sensing Imagery. IEEE Transactions on Geoscience and Remote Sensing. 44(2): 420–431.