KRINGLE: Map Algebraic Functionalities

Project I: GIS Architecture and Algorithms

User Manual: Kringle (v 0.1)

A GIS Raster processing software developed as an open source application for map algebraic functionalities (Local, Focal and Zonal operations). This project report contains its functionalities and benefits as well as comparisons and way forward to develop this software further.

omkarpp@kth.se, shuoc@kth.se; ionmatai@kth.se; huzh@kth.se

Contents

1. Introduction

The project relates to the functionality of GIS that is preparation, processing and visualisation of geographic data, specially raster data. The format of data supported and the algorithms implemented and access provided to the user are defined in this project report. A user-friendly interface, low computing necessitude for the implemented algorithm of functionalities, proper visualisation are the goals of this project. Following are certain attributes of the application discussed in brief.

1.1. Java powered application

The application is JAVA powered. All the algorithms are devised in JAVA Development Kit and hence integrates all its advantages and benefits. The application is platform independent i.e. it can be executed on any platform that has a JVM installed on it. It has all built-in security features that allow programmers to write highly secure Java programs. It also prevents malicious software from compromising the Operating System (OS) because it keeps Java applications from interacting with Operating System resources.

Java programs that run on a Java Virtual Machine tend to perform slower than equivalent programs written in C++. The system neutrality of bytecode acts as a disadvantage where performance is concerned. This is because code optimization relies heavily on system-specific features. Since Java bytecode is system-neutral, it cannot be optimized for a specific hardware set.

Regarding correctness, since a Java program relies on the Java Virtual Machine to execute it, the JVM must be free of errors for the program to operate correctly. This reliance on the Java Virtual Machine introduces a possible point of failure for the program. Luckily, the Java Virtual Machine software is produced with very high standards, and therefore it isn't likely to ship with any errors. Regardless, a failure in the Java Virtual Machine is a possibility that should be considered.

1.2. Graphical user interface (Swing functionalities)

Swing is the collection of user interface components for the Java programs. It is part of Java foundation classes that are referred to as JFC. In simple words, Swing is the graphical user interface toolkit that is used for developing the windows-based java applications or programs. Swing is the successor of AWT which is known as Abstract window toolkit API for Java and AWT components are a mainly heavyweight.

Lightweight: Swing components are lightweight which helps in creating the UI lighter. Swings component allows it to plug into the operating system user interface framework that includes the mappings for screens or device and other user interactions like key press and mouse movements.

Plugging: It has a powerful component that can be extended to provide the support for the user interface that helps in good look and feel to the application. It refers to the highly modular-based architecture that allows it to plug into other customized implementations and framework for user interfaces. Its components are imported through a package called java.swing.

Manageable: It is easy to manage and configure. Its mechanism and composition pattern allows changing the settings at run time as well. The uniform changes can be provided to the user interface without doing any changes to application code.

MVC: They mainly follows the concept of MVC that is Model View Controller. With the help of this, we can do the changes in one component without impacting or touching other components. It is known as loosely coupled architecture as well.

Customizable: Swing controls can be easily customized. It can be changed and the visual appearance of the swing component application is independent of its internal representation.

1.3. Open Source coding

The source-code of the application is provided online and freely downloadable for usage and modifying to user needs. Git Hub is used as the repository for the project. Using GitHub is free if your project is

open source and includes a wiki and issue tracker that makes it easy to include more in-depth documentation and get feedback about your project. If you want to contribute, you just fork a project, make your changes and then send them a pull request using GitHub web interface.

2. Analysis: Map Algebraic functions

In a raster data set, each cell represents a value at a given location. A grid cell could represent anything – temperature values or precipitation volume. The map algebra tool is a cell-by-cell combination of raster data layers stacked on top of each other. A simple operation like addition or multiplication are applied to each raster cell location. Map algebra generates a new raster output based on the math-like expression. Map algebra can be defined as local, focal, zonal and global operations.

2.1. Local Operations

The value generated in the output raster is a function of cell values at the same location on in the input layers. When you take the temperature average in each cell using two raster grids, this is an example of a local operation.

Here are examples of operations that can be used between the two raster layers:

- Arithmetic operations (addition, subtraction, multiplication, division)
- Statistical operations (minimum, maximum, average, median)
- Relational operations (greater than, smaller than, equal to)
- Trigonometric operations (sine, cosine, tangent, arcsine)
- Exponential and logarithmic operations (exponent, logarithm)

Input1, Input2, … => Output

2.1.1. Applications: reclassification (by table, ranges, etc.)

On each layer, possible values for each cell are 1 or 0.

E.g., suitability of a region to a certain purpose by logical reasoning: different layers have binary values for each location, the result based on the logics on these values.

E.g., visualizing contours using DEM: Divide all values of pixels by 10, multiply the integer result by 10, subtract this value from the original; color each value according to the following: values 4 and 6, 3 and 7, 2 ja 8 and 1 ja 9 get the same color, 0 is white, 5 is black; one-meter contours.

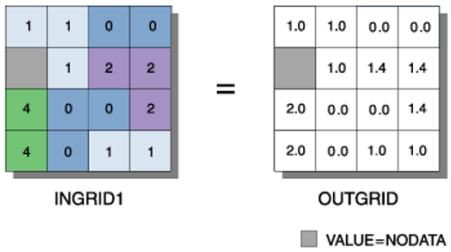

Figure 1. LocalRoot Operation

2.1.2. Local Operations:

LocalRating, LocalMaximum, LocalMinimum, LocalSum, LocalMean, LocalDifference, LocalProduct, LocalRatio, LocalRoot, LocalSine, LocalCosine, LocalTangent, LocalArcSine, LocalArcCosine, LocalArcTangent, LocalCombination, LocalVariety, LocalMajority, LocalMinority, etc.

2.2. Focal Operations

The focal operation is a spatial function that computes an output value of each cell using neighborhood values. Convolution, kernel and moving windows are examples of image processing techniques that use focal operations.

A moving window is a rectangular arrangement of cells that applies an operation to each cell in a raster dataset while shifting in position entirely. A neighbourhood operation is a spatial function where the output location, area and extent comes from areas larger than and adjacent to the input cells. For example, average neighbourhood operations smooth values in a map.

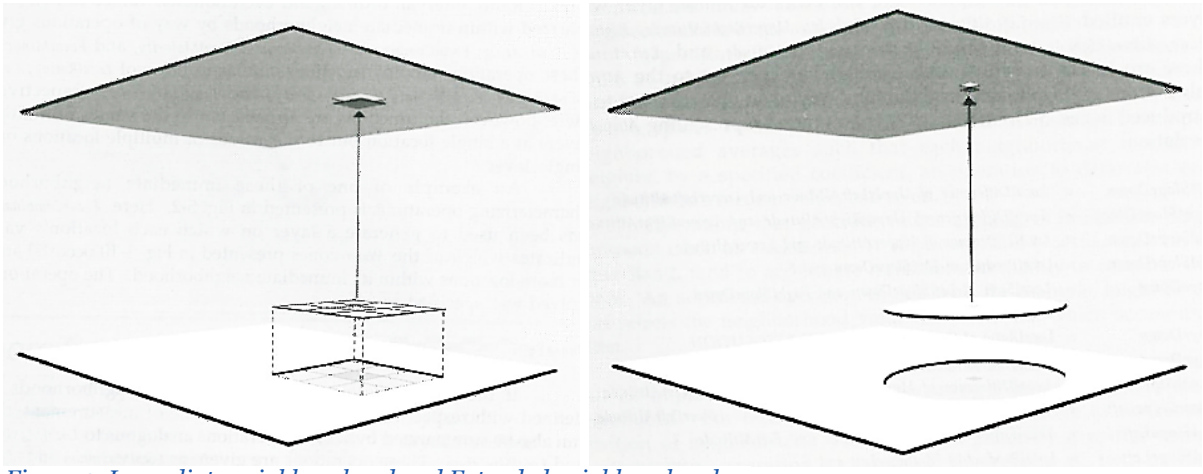

Figure 2. Immediate neighbourhood and Extended neighbourhood

Input1, neighborhood => Output

2.2.1. Applications of focal operations

Focal Variety: Focal Drainage:

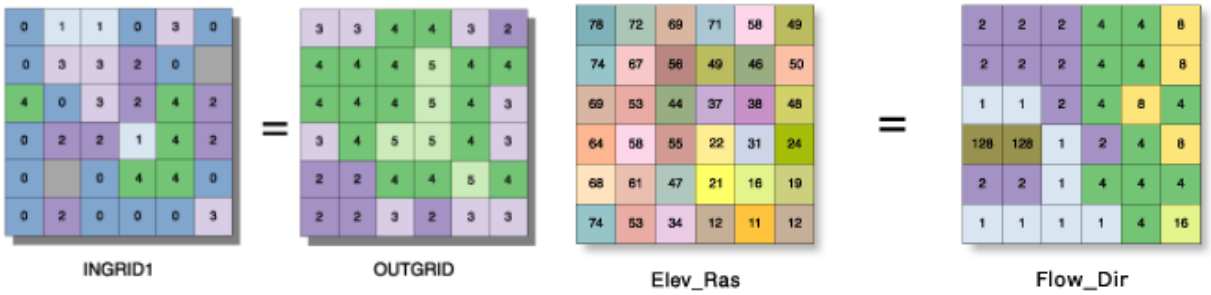

Figure 3. Illustration of focal variety and focal drainage operations

- Smoothing the map layer by FocalAverage; high and low values are smoothed away (continuous values)
- Generalization by filtering out individual pixels that differ from their surrounding (categorical values)
- Mobility simulation by producing a cost surface of mobility.

2.2.2. Other Focal Operations

Operations on a pixel and its 4 (or 8, or 32, or…) neighboring pixels are FocalRating, FocalCombination, FocalVariety, FocalMajority, FocalMinority, FocalMaximum, FocalMinimum, FocalSum, FocalMean, FocalProduct, FocalPercentage, FocalPercentile, FocalRanking, FocalInsularity, FocalProximity, FocalBearing, FocalNeighbor, FocalGravitation, IncrementalLength, IncrementalArea, IncrementalVolume, IncrementalGradient, IncrementalAspect, IncrementalDrainage, etc.

2.3. Zonal Operations

A zonal operation is a spatial function that computes an output value of each cell using the zone containing that cell. An example of a zone could be a watershed. When you want to calculate the total mean volume of precipitation in each watershed zone, this is an example of when you would use a zonal operation.

Input1, zones => Output

2.3.1. Applications of zonal operations

- Calculating the land cover temperature average.
- Calculate the amount of apartments in a block: ZonalSum.
- Calculate the most common soil type for each land cover area: ZonalMode (or ZonalMajority).

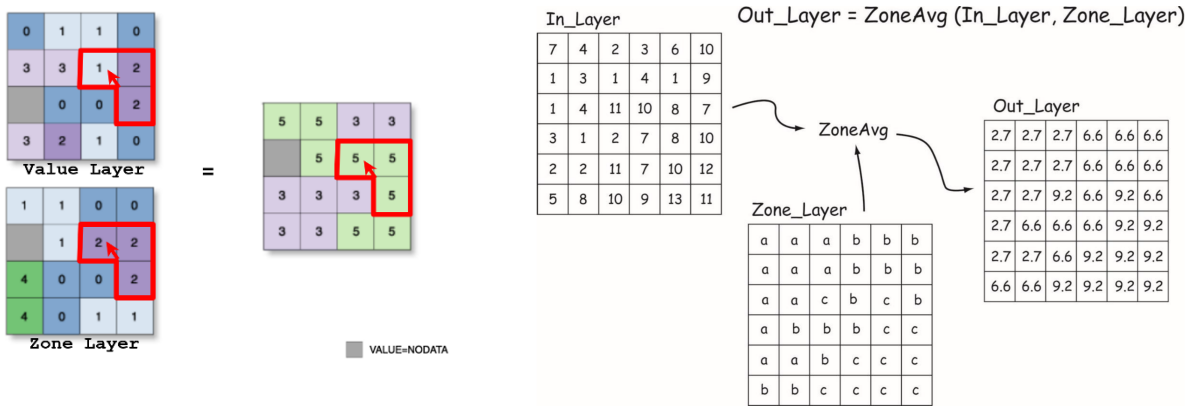

Figure 4. Illustration of zonal operations.

2.3.2. Other Zonal Operations

ZonalCombination, ZonalMajority, ZonalMaximum, ZonalMean, ZonalMinimum, ZonalMinority, ZonalProduct, ZonalRating, ZonalSum, ZonalVariety, ZonalPercentage, ZonalPercentile, ZonalRanking, etc.

3. Kringle: user manual for functionalities

The first version Kringle has already realized various functions, it is superior not only in the ASCII data processing, but also with a user-friendly software. This part demonstrate the instructions of this software and make you get familiar with it in 10 minutes.

3.1. Functions and algorithms

The interface is quite simple and clean, which makes this software easy to use. It mainly contains the menu bar, the contents viewer, display viewer and preview viewer. The menu bar relates to the function implement, contents viewer shows the file location, displayer view is used to display the file data or the processed data in image format, and preview viewer is a small window that shows the thumbnail of the displayed image, which allow users to have a look at how is the data like from a whole prospective and when there are more than one image, it's possible choose one to focus on. The contents viewer, display viewer and preview viewer are combined by JSplitPanel, which means that their sizes are not fixed, but adjustable.

3.1.1. The menu bar

- File Menu

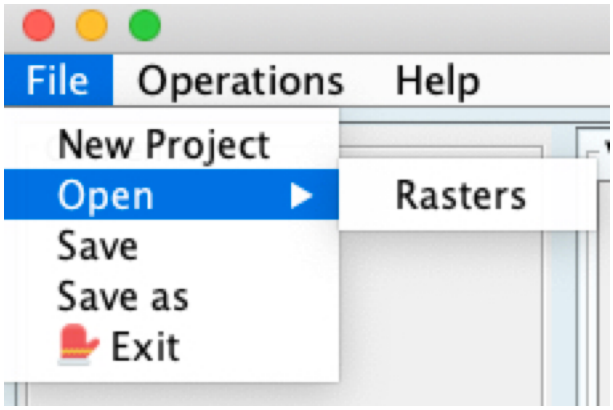

Figure 5. The organization of File menu.

- New project – Create a new project and start working on it. Every time before you implement any functions, click this menu. Also when you want to discard what you have done and restart a new project, this menu will help you clean the display viewer as well as preview viewer.

- Open – Open from an ASCII format file, and display it in an image format in the display viewer and add a thumbnail in the right preview panel. Meanwhile, the left content panel lists all the txt file under the same directory with the opened ASCII file.

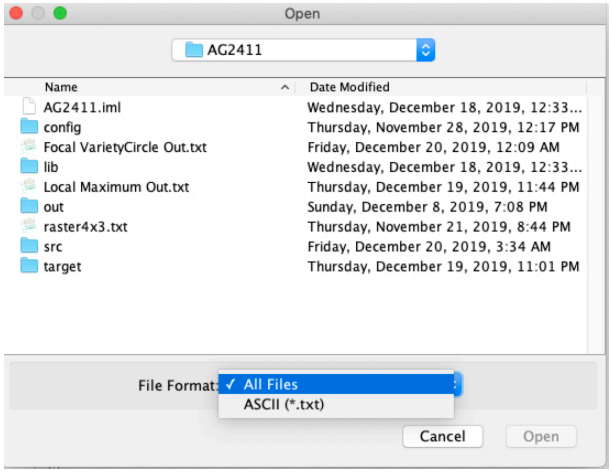

Figure 6. The open dialog.

- Save — Save the current display viewer as a txt file to current workspace.

- Save as – Save the display viewer as an image to your local path.

- Exit – Exit the software after finishing it, you will be asked one more time before exit.

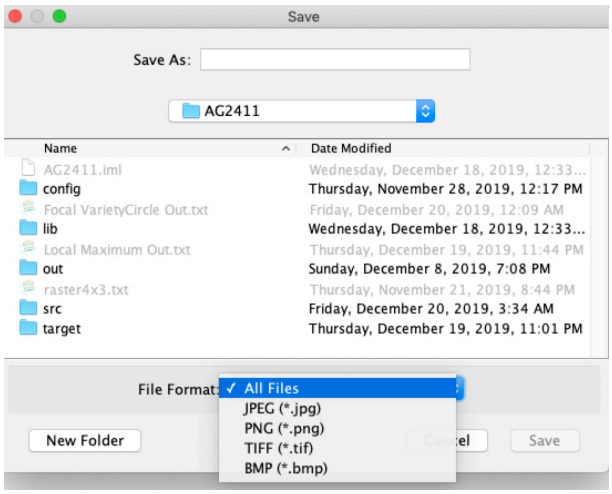

Figure 7. The Save as dialog.

- Operations Menu

  This menu is used for implementing the operations introduced before, choose one operation and the corresponding dialogue will pop up. Choose the input file(s) and output file, set the parameters and method, and wait for a few seconds until the progress bar completed, then the result will display in the display viewer automatically and meantime you get the result ASCII file in your local path.

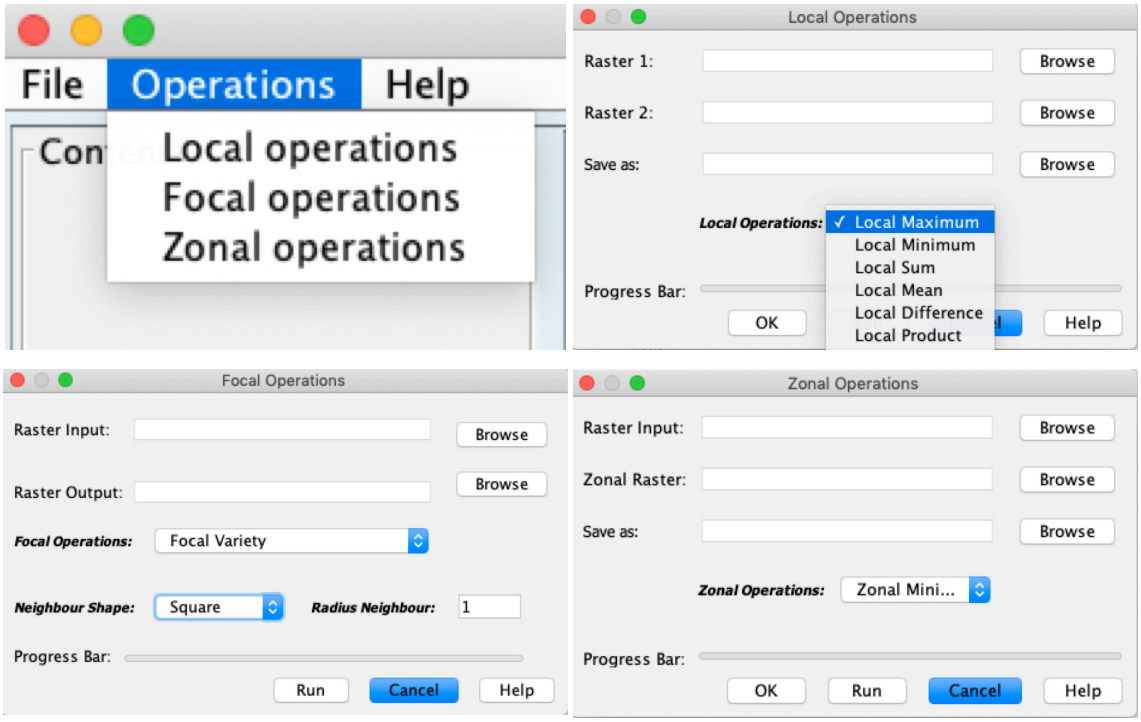

Figure 8. The organization of Operation menu and dialogs of each operation.

- Help Menu

  In this part, users can find the help function when they have problems.

  - Contact support—You can find our technical support information here, if you have some problems about this software which you cannot find in the help manual, feel free to contact them, usually you will get a reply within 24 hours.

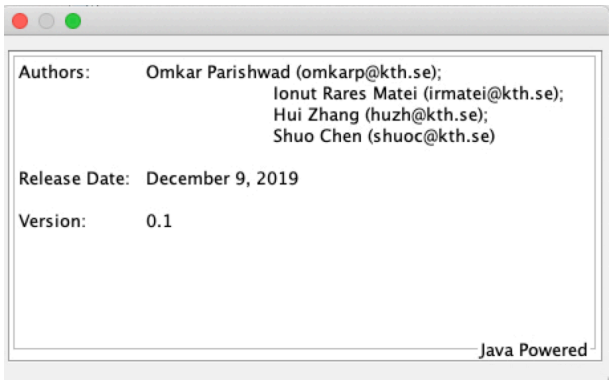

Figure 9. The content of Contact support.

  - Help manual – Here you can get the help manual which has a detailed instruction about the use of our software. You are strongly suggested to read it carefully before using this software.

  - Register – Register is the same important. You have to register for our software after the 15-day trial, please keep it in mind. Click the register menu and fill in your personal information, read our terms and conditions before clicking accept the terms and conditions, once you have clicked this, it means you will be satisfied that the agreement you are entering into suits you as well as us. Then you will receive a greeting email from us about the registration result.

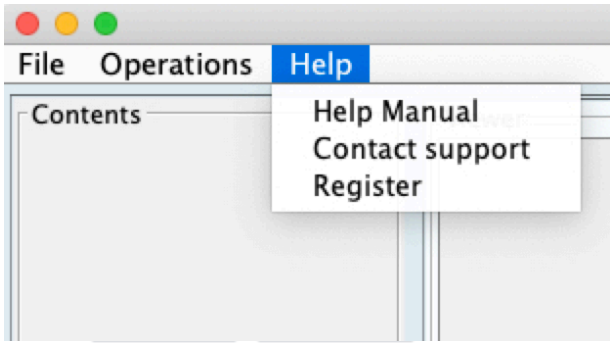

Figure 10. The organization of Help menu.

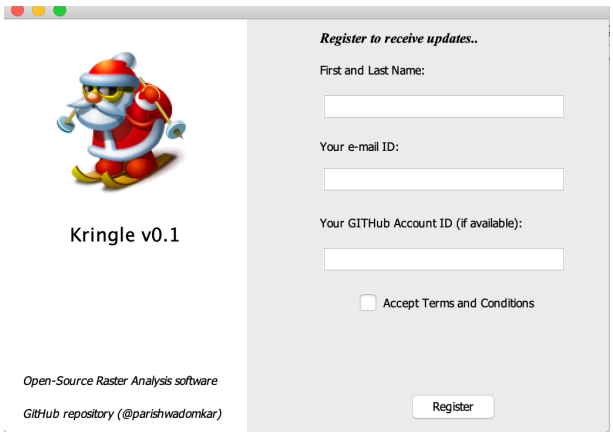

Figure 11. The content of Register dialog.

3.1.2. contents view

This view mainly used the JTree function, combined with java built-in file function to get access to the file name and file location. Get all the files and folders by looping and organized them in JTree format then display them to users.

In the contents view, you can have a look at the file locations. This view is very helpful to quickly find a specific data that you want to use, also it helps you to determine where you want to put the output file.

By double clicking the file, users can take a quick look at an input file or an output file in the image format.

3.1.3. display view

This view is designed for the showing the details of a specific layer, double clicked in the file tree or selected the preview panel. The shown image is zoomable and draggable . The name of the current displayed layer is shown at the bottom of the window, and also the coordinates of the mouse and the corresponding cell of the layer and its value.

3.1.4. preview view

This view is designed for showing the thumbnails of input layers and output layers, whenever an ASCII format file is opened or a specific map algebra operation has been conducted.

Those thumbnails are hosted by a JPanel in a JScrollPanel, which allows users to browse all the input and output files that have been processed. Every thumbnail is labeled and clickable, where single click allows users to check its details in display panel, and double click to remove it. And there's also a clear button above for removing all the thumbnails at once.

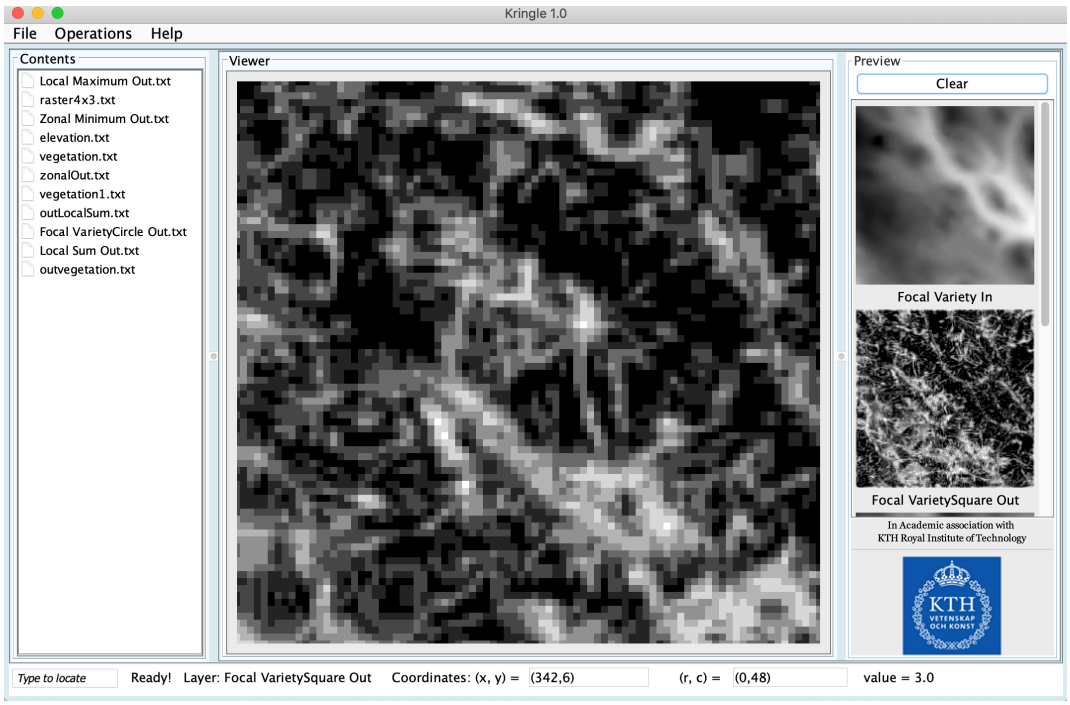

Figure 12. The main window of Kringle.

4. Way forward for Kringle: Comparison with other software

This comparison section is a benchmark that Kringle sets for itself as an open source image processing software, which is not available in the market due to the complexity of the algorithms for implementation. Discussed herewith are three of the most commonly used commercial applications: ArcGIS and ERDAS Imagine for raster processing.

Map algebraic functionalities are the basics of GIS technology which lead to applicability of remote sensing and data processing abilities for various applications. All the software available in the market as either open source or commercial have the map algebraic functionalities. All of them are highly evolved with an orientation towards applications. Its highly improbable that the user has a requirement or even knowledge for such specific operations and hence, they are not directly available in a software form. To understand this, we need to go over some commonly used applications.

4.1. ArcGIS Software (ESRI GIS Product)

The GIS application, used commonly called ArcMap has the same functionalities in its Spatial Analysis toolbox. It is the most advanced and popular commercial product in the market. It provides three basic tools for running its Map Algebra.

4.1.1. Raster Calculator

The Raster Calculator tool executes Map Algebra expressions. The tool has an easy-to-use calculator interface from which most Map Algebra statements can be created by simply clicking buttons. Raster Calculator can be used as a stand-alone tool, but it can also be used in ModelBuilder. As a result, the tool allows the power of Map Algebra to be integrated into ModelBuilder.

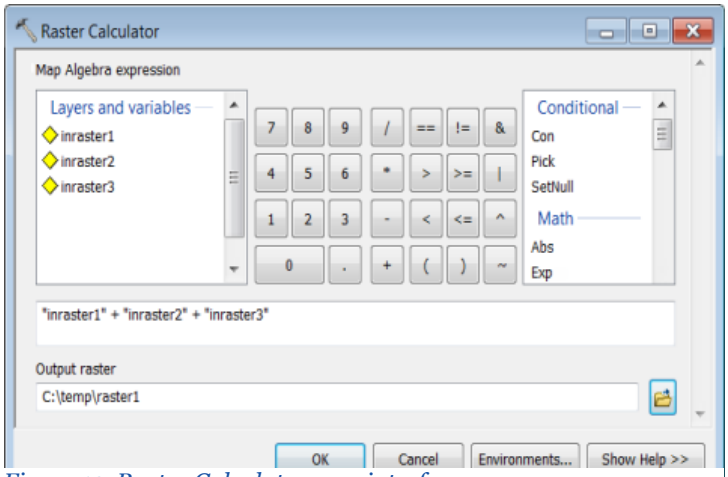

Figure 13. Raster Calculator user interface

The Raster Calculator tool is not intended to replace other Spatial Analyst tools. Continue to use the other tools for the appropriate calculations. For example, use the Slope tool to perform the slope calculations. The Raster Calculator tool is designed to execute single-line algebraic statements.

Since Raster Calculator is a geoprocessing tool, like all tools, it can be integrated into ModelBuilder.

4.1.2. Python window

The Python window is an efficient and convenient location to use geoprocessing tools and Python functionality from within ArcGIS. The Python commands run from this window can range from single lines of code to complex blocks with logic. The Python window also provides a place to access additional functionality using custom or third-party Python modules and libraries.

To launch the Python window, click the Python Window button Python on the Standard toolbar, or from the Geoprocessing drop-down menu, click Python window.

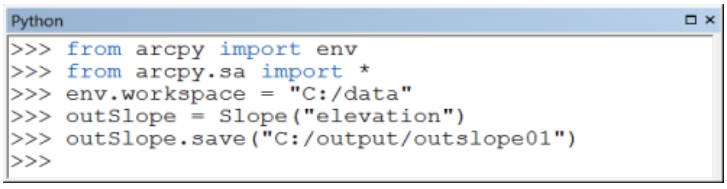

Figure 14. Python interface in ArcGIS.

In the above sequence of statements, the ArcPy site package, the geoprocessing environments, and the Spatial Analyst modules are imported; the workspace is set; the Slope tool

is run; and the output is permanently saved. Once a carriage return is entered at the end of a statement, that statement is immediately run.

Some features of the Python window include built-in line autocompletion, use of variables, and access to Python and ArcPy functionality.

4.1.3. Python integrated development environment (IDE)

Even though there is no limit to the number of statements that can be entered within the Python window, it may be cumbersome to create more complex models. The Spatial Analyst modules' tools, operators, functions, and classes can also be accessed from your favorite integrated development environment such as PythonWin. Start your preferred IDE and enter the desired statements.

In the following script, ArcPy, the geoprocessing environments, and the Spatial Analyst module are imported; variables are set; the extension is checked out; the Slope tool is run; and the output is saved.

All Spatial Analyst tools that output a raster are available in algebraic format. The dataset name can be used if it is in the Table of Contents window or in the current workspace; otherwise, the full path must be entered.

4.2. ERDAS Imagine by Hexagon Geospatial

The most popular and advanced raster processing Software. Again, oriented with an user requirement of application oriented approach. The map algebraic functionalities in it are as discussed herewith.

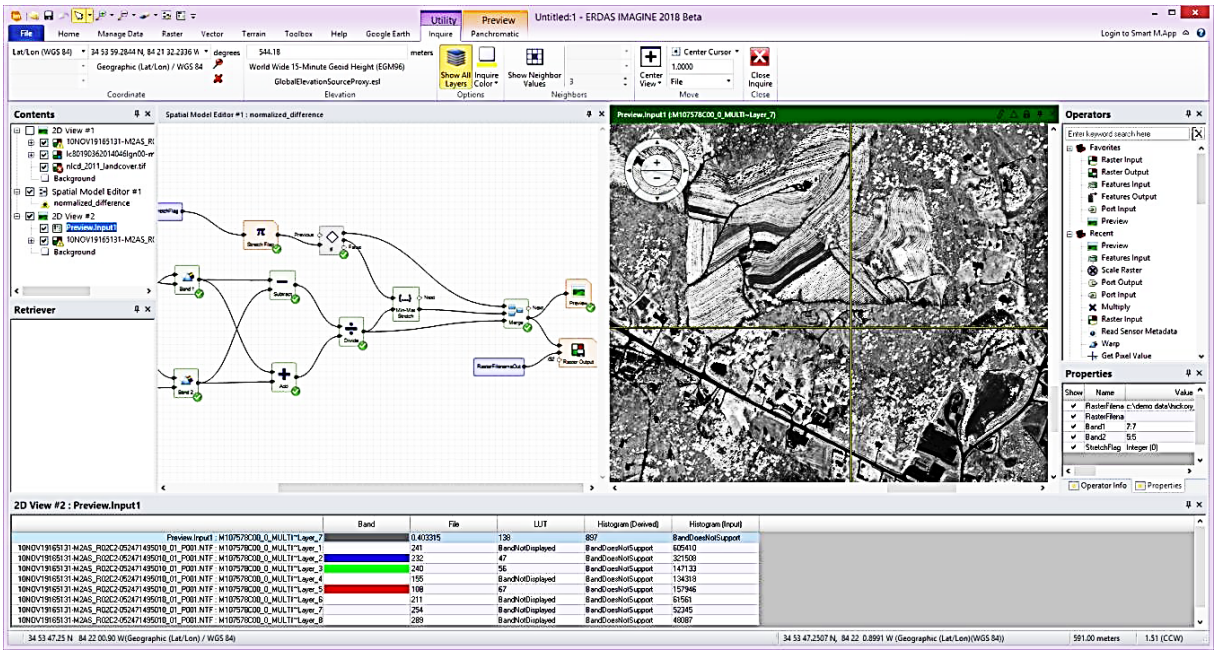

Figure 15. ERDAS Imagine interface

Being a commercial and an advanced raster processing application, it has raster calculator built in almost all its facades of applications. The model builder is again a part of the product as well, which provides an interface to integrate multiple map algebraic operators for the imagery.

Its famous for integrating the ArcPy operator and the interoperability for its outputs which is the eventual issue of many software available in the market due to their variability of formats generated. It also has machine learning features which supports the multitude of operators available at its disposal, integrated with its application-oriented user interface. Change detection, feature extraction, coherence detection as per applications of imagery (SAR), web services, etc. are some of the examples of tools available in its functionality.

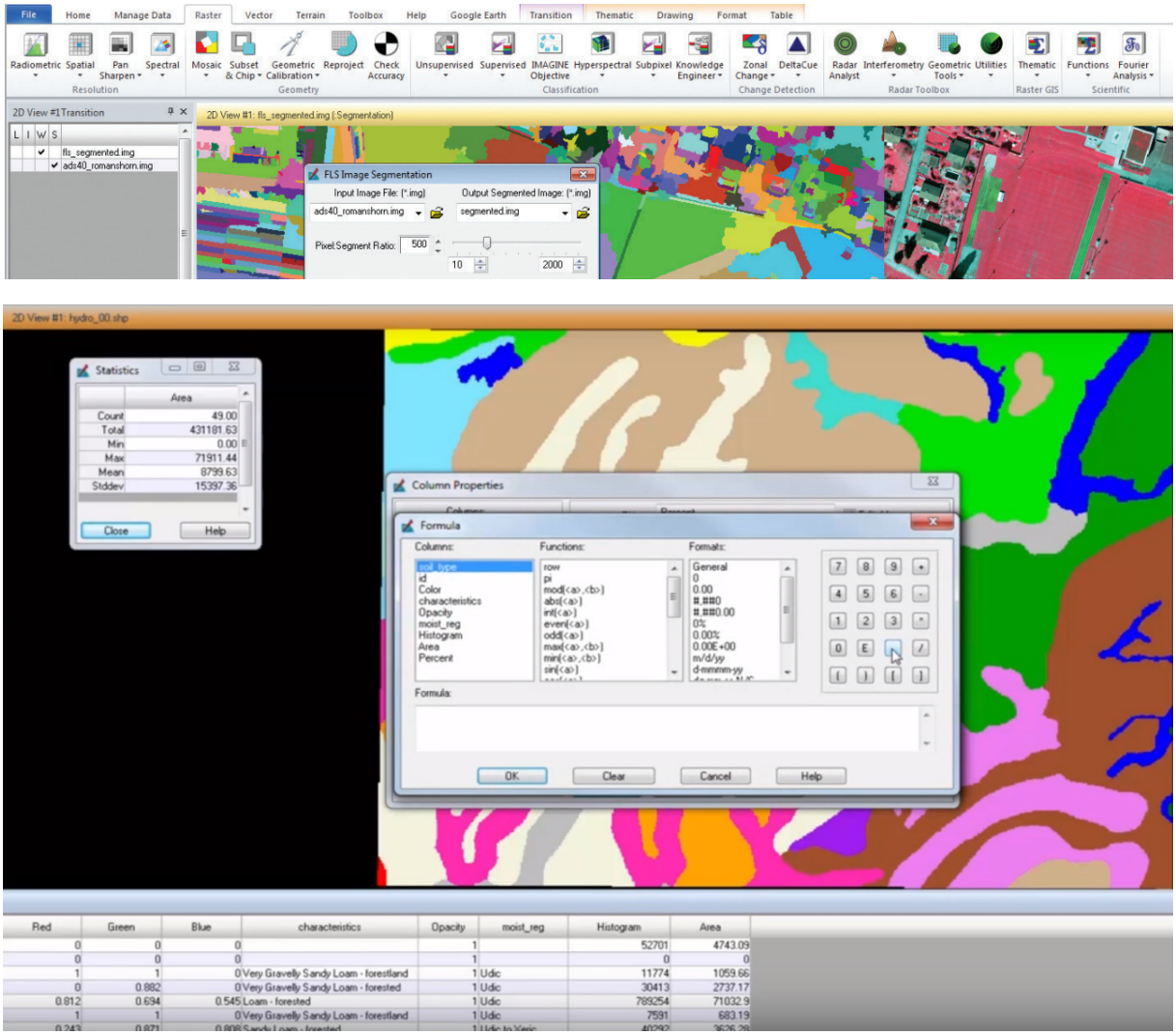

Figure 16. Raster Calculator unique application in ERDAS Imagine

5. Plans for future integration

GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside 40 million developers. This open source repository is perfect to host the source code of Kringle, which can then be advanced with proper documentation autogenerated online.

The registered Kringle users offers an added benefit of creating a database and adding them as a collaborator for providing the source code, allowing them with a chance to get to know the code and update it with authorisation. All the versions can be tested online and maintained as well.

MSc Transport and Geoinformation Technology

KTH Royal Institute of Technology, Sweden

Authors:

- Shuo Chen
- Omkar Parishwad
- Hui Zhang
- Ionut Rares Matei