N-E-RO

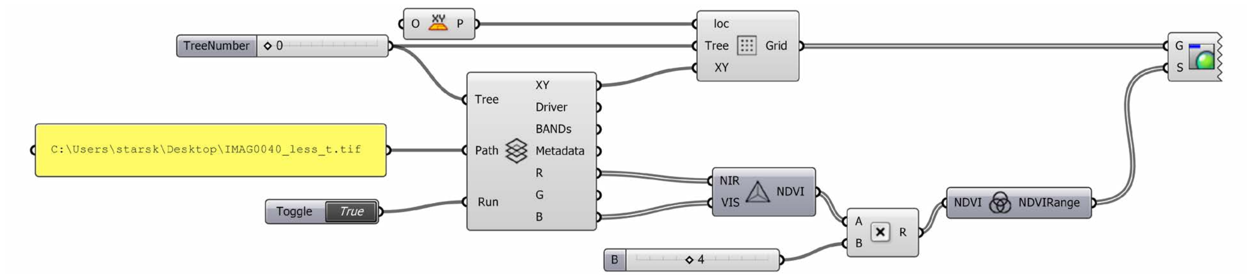
NETWORKING ENVIRONMENTAL ROBOTICS
[GRASSHOPPER LIBRARY DOCUMENTATION]
//V 0.6.2 BETA//

NOUMENA

INTRODUCTION

NERO (Networking Environmental Robotics) is a collaborative project to develop systems and tools to translate environmental conditions into data. As a part of the tool, we developed **NOUMENA** (V 0.6.2 beta) - an add-on for Grasshopper. Our objective was to offer an interactive tool to visualize environmental data collected by means of aerial robots.

This add-on contains different components be able to extract data from the single Image (JPEG format) and translate into Grasshopper and Rhinoceros interface. These components are completely developed in C# language, which integrates GDAL/OGR libraries in order to support an extensive range of raster and vector formats.



DEVELOPER:

The development of Grasshopper add-on was a collaborative work of team NOUMENA.
Team involves:

Starsky Lara: Development of NERO library and documentation.
Aldo Sollazzo: NERO project Coordinator, support NERO code development
Efilena Baseta: Documentation
Chirag Rangholia: Nero Project Manager, tester Nero library

mail: developer@noumena.io | web site : http://nero.noumena.io/

Licence OPEN SOURCE:

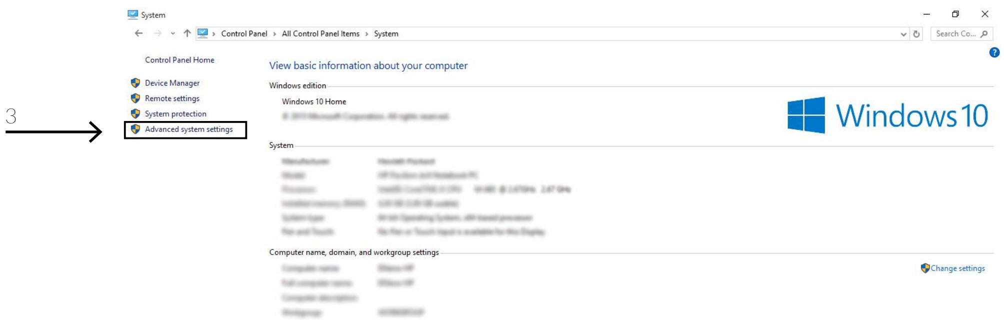
Installation

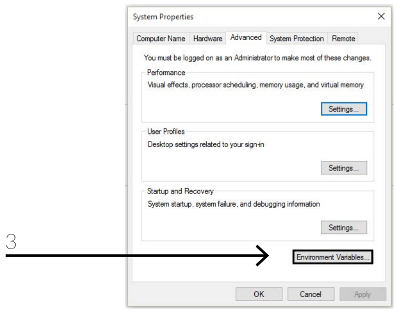
1- Download and Unzip NERO folder. Copy and paste the folder "GDAL" in drive C as follow: **C:\GDAL** (NOTE: for now the library works only in 64 bit computers!)

2- Go to your Control Panel: **Control Panel\System**

3- In System Go to the **Advanced System Settings**. In this **System Properties** panel Click on "**Environment Variables**" (refer to image below)

4- In the **System variables** panel choose "**Path**" and click "**Edit..**" (refer to image below)



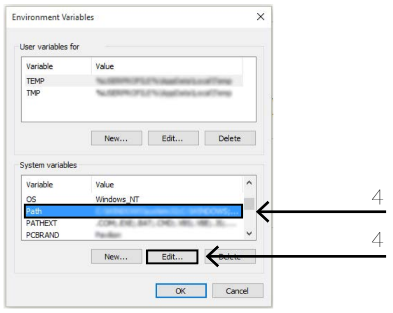


You must be logged on as an Administrator to make most of these changes.

Performance
Visual effects, processor scheduling, memory usage, and virtual memory

User Profiles
Desktop settings related to your sign-in

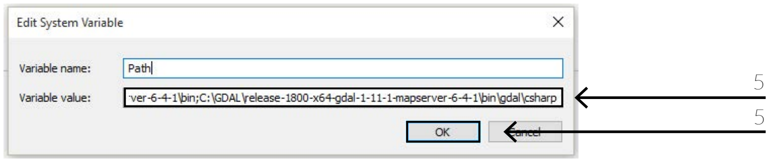
Startup and Recovery
System startup, system failure, and debugging information



Variable	Value
TEMP	
TMP	

Variable	Value
OS	Windows_NT
Path	
PATHEXT	
PCBRAND	

5- Copy and paste following path at the end of the existing path: **C:\GDAL\release-1800-x64-gdal-1-11-1-mapserver-6-4-1\bin;C:\GDAL\release-1800-x64-gdal-1-11-1-mapserver-6-4-1\bin\gdal\csharp**



Variable name: Path

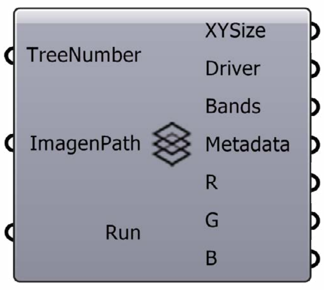
Variable value: ver-6-4-1\bin;C:\GDAL\release-1800-x64-gdal-1-11-1-mapserver-6-4-1\bin\gdal\csharp

Attention! Separate the new path and existing paths with "**;**" punctuation mark. The existing paths are different for each computer!
Please you must not replace the existing path, just copy and paste above mentioned path at the end of the existing path.

5- Drag and drop the GH files (dll and gha) from NERO source folder to the Components Folder of Grasshopper (Grasshopper -> File -> Special Folders -> Components Folder)

NOTE: Remember to Unlock each GH file (dll and gha)!

ReadRaster



DESCRIPTION

ReadRaster is first component to read data that is stored in the R, G, B channels (BANDs) of a raster image (a raster image is a dot matrix data structure representing a generally rectangular grid of pixels, or points of color).

This component will geo-process an image (e.g. TIFF) as a dot matrix data structure of pixels. This matrix data structure can be read by rows (running across) or columns (running down) of pixels. Each dot of the matrix contains a set of numerical values relative to R, G, B channels. Depending on the type of picture and the information stored in it, each dot can also contain other channels such as: near Infrared (NIR), electromagnetic spectrum, cirrus, panchromatic and thermal infrared.

INPUTS

TreeNumber: Settings for the organization of the matrix data structure. The settings are: “0” as input for Flattened DataTree, “1” as input for Grafted DataTree by columns where rows are included into the list.”
NOTE: The output data structure is organized by the data tree structure of Grasshopper

ImagePath: “The path as string from the image file stored on your computer. The file has to be part of the GDAL Raster Formats. Check the GDAL Raster Formats in the following link: http://www.gdal.org/formats_list.html”

RunRaster: “Run the component”

OUTPUTS

XYSize: “Size of the width and the height of the image. The units are in pixels by “x” and “y” respectively.”

ImageDriver: “Type of driver used by GDAL in order to read the DataSet of the image. Find the drivers that GDAL can read here: http://www.gdal.org/formats_list.html.”

Bands: “Number of Bands (channels) stored in the file.”

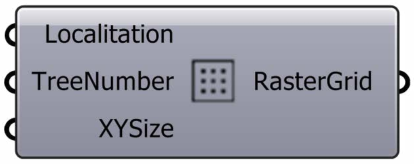
Metadata: “Provides descriptive metadata to facilitate information about the file”

R: “Red BAND (Optional channel).”

G: “Green BAND (Optional channel).”

B: “Blue BAND (Optional channel).”

RasterGrid



DESCRIPTION

RasterGrid is to display the data of specific BANDs, providing numeric informations about the pixels. It enables the visualization of all the information, such as illuminate values, monochrome values, colour values, 3d values or numeric values, assigned each data to the corresponding coordinates of the grid.

This component will create a grid of points according to the Raster dot matrix data structure, generating “x” and “y” coordinates for each pixel. We aimed to create vector features from the image identifying every small variation of each pixel.

INPUTS

Localization: “Construction Plane of the grid in the 3d space.”

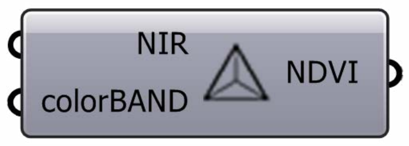
TreeNumber: “Settings for the organization of the matrix data structure. The settings are: “0” as input for Flatten DataTree, “1” as input for Grafted DataTree by columns where rows are included into the list.”

XYSize: “Size (in pixels “x, y”) of the width and the height of the image. The actual size of the grid depends on the Rhino Unit. For example: 1pixel = 1 RhinoUnit.”

OUTPUTS

RasterGrid: “Geometrical grid of 3d points according to the Raster dot matrix data structure. This grid processes the data in the same order as the ReadRaster component.”

NDVICalculator



DESCRIPTION

NDVICalculator is to calculate to extract "Normalized Difference Vegetation Index" (NDVI) values from the each pixel. NDVI is directly related to the photosynthetic capacity and hence energy absorption of plants. NDVI is one of the most commonly used vegetation indices.

The NDVI is calculated from the following individual measurements:
NDVI = (NIR – VIS) / (NIR + VIS)

Where VIS and NIR stand for the spectral reflectance measurements acquired in the visible (red) and near-infrared regions, respectively.

This component is based on the "The NDVI formula", which can be found in the following link;
https://en.wikipedia.org/wiki/Normalized_Difference_Vegetation_Index

Attention:
A basic math error is caused in the NDVICalculator when the image file comes from a camera with blue or red filter! Though with some modifications the process seems to be still valid. You can find more information about how to fix the original erroneous images on following link;
http://publiclab.org/wiki/mobius-infragram-erroneous

See an example of the error here;
http://publiclab.org/notes/warren/04-10-2014/mobius-action-cam-infragram-tests

INPUTS

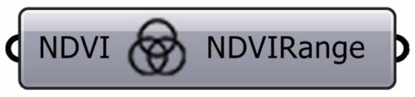
NIR : "NIR is the spectral reflectance measurements acquired in the near-infrared band."

VIS : "VIS is the spectral reflectance measurements acquired in the visible (red) band."

OUTPUTS

NDVI: "The Normalized Difference Vegetation Index" as a value between -1 and +1."

NDVIColorRange



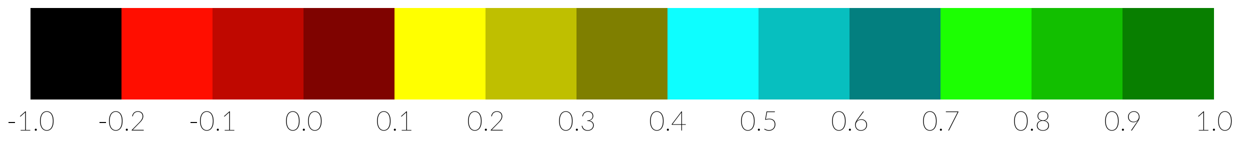
DESCRIPTION

NDVIColorRange is to display the "Normalized Difference Vegetation Index" as False Colour Mapping. The vegetation index value (NDVI input) must be a number between -1 and +1. Depending on the value of the number, a particular colour is assigned to that pixel. Each pixel in the original image is processed through NDVICalculator component and assigns RGB colour. The process of assigning colours depending on the value is called False Colour Mapping.

This component will create a numeric domain colour range based on the following source: https://en.wikipedia.org/wiki/Normalized_Difference_Vegetation_Index#/media/File:NDVI_062003.png.

The information processed from our NDVICalculator component will be used as unique input for the NDVIColorRange. This component will process the data information from the NDVICalculator in order to rebuild a false colour image that represents the vegetation index.

Color Range :



INPUTS

NDVI: "The NDVI from NDVICalculator is the unique input for this component. The data must be a number between -1 and +1."

OUTPUTS

NDVIRange : "Transforms the RGB color data of "The Normalized Difference Vegetation Index" into False Color Mapping."