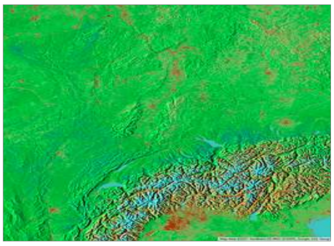
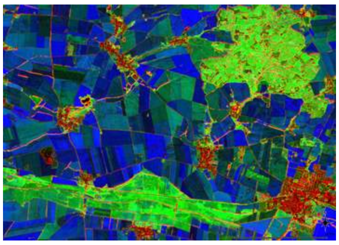
ESIS EcoSystem Integrity Service



Traits

Calibration, Reduction,
Kernel, Time Series,



Methods

Zones, Connections,
Classes, Objects

Software

Library, Interface,
Algorithms

► **Peter Selsam**

► **Angela Lausch**

► **Jan Bumberger**

Department Monitoring- und Erkundungstechnologien
Department Monitoring and Exploration Technologies

Helmholtz-Zentrum für Umweltforschung GmbH – UFZ
Helmholtz Centre for Environmental Research GmbH – UFZ
Permoserstraße 15, 04318 Leipzig, Germany

peter.selsam@ufz.de, www.ufz.de

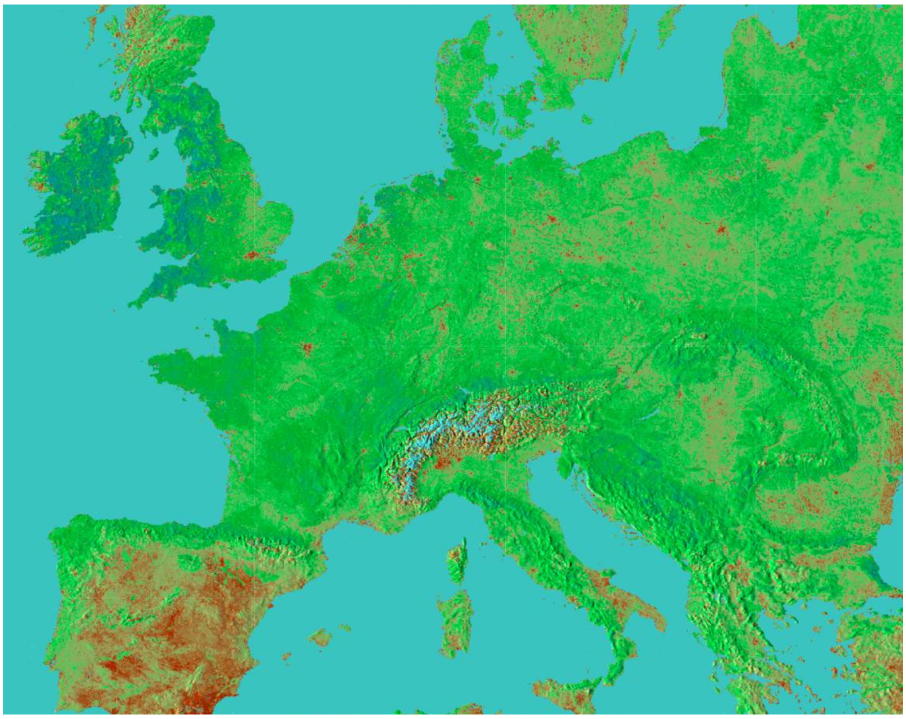
ESIS

The Ecosystem Integrity Service (ESIS) was designed to link, analyze and model data from the natural environment. At present, land-use borders, landscape diversity, development and the main types of landcover are the main objectives.

Vegetation index of Central Europe superimposed with shading from an elevation model

Elevation data from SRTM mission (2001), image data Landsat-8, 2014-2020

Vegetation index NIRv (red and infrared). Values between 0.0 (turquoise), 0.1 (red brown) to 0.4 (dark green)



Traits

In ESIS, “traits” are well-defined features associated with a specific place and time. Traits should be scale-invariant, sensor-independent and globally applicable. Each location can be described by numerous traits.

Imalys

The software library Imalys (Image Analysis) was developed to derive traits from satellite images. Imalys offers generic processes for specific tasks, wrappers for known processes and a workflow that can be automated.

Tutorial

The following chapters describe the application of the processes in Imalys and give hints on how to link them. The tutorial includes executable examples that can serve as a starting point for own solutions.

Methods

Some traits provide new insights into the structure of a landscape, others use new software techniques. Background, methods and motivation of the most important processes are described in “Methods.”

Nomenclature

Implemented processes are marked by gray boxes. The process names are used as an identifier, as a command, as a filename, and as a field name in tables. The names are short and can have a much wider meaning in general usage. In the text they are marked as →link.

The following chapters also contain processes that are not yet released. They are brightly colored.

Prozess	*Implemented processes*
Prozess	*Development*

Some processes need figures as a parameter. [N] stands for natural numbers, [R] for rational figures (floats). A hint is given, if Zero is excluded,

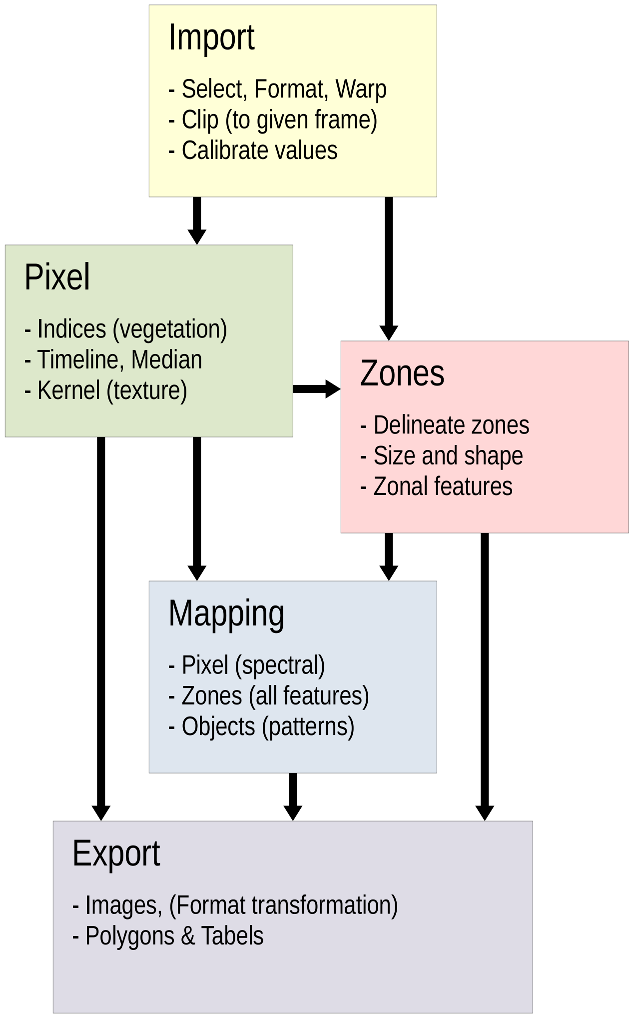
Imalys Workflow

Imalys was designed as a library providing methods for image analysis. The processes are designed to run on a server without a graphical interface.

Imalys provides processes for importing and exporting raster and vector data. Numerous formats can be read and written.

Between import and export, there are processes for the transformation of individual pixels, the delineation of land-use borders and the classification of image content.

All processes use the same interface, but are logically interdependent. An overview of all processes is shown in Fig.XXX



The graphic shows the most important paths in the Imalys workflow. Image data are imported into a working directory and stored in ENVI format. Pixel-oriented processes generate new bands from the raw data. Image pixels are grouped into zones and stored in vector format. Pixels or zones are classified and merged into objects. Finally the results can be exported in a selectable format.

Except for the import, none of the steps is mandatory. Each process stores its result under its own name in the working directory. Each result can be a reference for another process. Whether the chain makes sense is up to the user to decide. Imalys includes processes for all stages of image analysis.

Pixels	Imalys contains three groups of routines that can modify or combine pixels. “Indices” combine different bands (frequencies, colors) from an image to a new feature, e.g. a vegetation index. “Time series” compare equal bands at different times to find outliers or trends. “Kernels” scan the local environment of each pixel for spectral differences, patterns or similarities.
Zones	Many processes in Imalys use zones. Zones are contiguous areas in the image with largely identical pixel characteristics. Zones follow the OBIA concept. In addition to their spectral features, zones have features of shape, size and environment. Imalys uses these features to characterize structural elements.
	Zones completely cover the image. If zones have been formed, the image can be fully represented by polygons and attributes, such as a map. Changes and spatial concentrations are easier to detect in this form.
Objects	Imalys also offers the possibility to link different zones to larger “objects”. Objects are contiguous combinations of different zones. Which zones will be part of an object is the result of a pattern analysis. Objects are defined by their pattern of different zones.
Classes	Imalys includes classification routines for pixels, zones, and objects. In all cases, these routines are self-calibrating and do not require training. The routines analyze the feature space for frequent combinations and assign a user-defined number of classes based on the distances in the feature space.
Control	Imalys consists of executable programs which are called as a command and controlled by parameters. The most important parameter is a script that can contain as many commands and parameters as desired. Imalys executes the commands in the specified order. Alternatively, commands and parameters can also be passed directly. The library includes a graphical interface that can create the scripts without format errors.
Source code	Imalys is freely available as code under GNU license. Imalys was programmed in Free Pascal and relies in its current form on a Linux environment. The code meets the practical needs of our working group. Additions and extensions are most welcome.

Processes

Run Imalys

xImalys script	*runs a "xImalys" commands chain*
Home	*Initialize a new process chain*

Import

Home	*Initialize a new process chain*
Import:	*Select and calibrate image data*
clipping = true	*Restrict import to a selected frame (option)*
quality = true	*Mask out image errors and clouds (option)*
scaling = true	*Value calibration and athmospheric correction (option)*
Warp:	*Reprojection with high level interpolation*

Indices, Pixel Arithmetics

Reduce	*reduce multitemporal or multispectral images*
NirV:	*Vegetation index „Near Infrared Vegetation" (NIRv)*
Principal:	*First principal component, dimensional reduction*

Statistics, Trends

Mean:	*Arithmetic mean, bands or images*
Median:	*Most common values from image stacks*
Difference:	*Eukledian distance of two n-dimensional properties*
Variance:	*Variance based on standard deviation*
Regression:	*Regression based on standard deviation*

Diversity, Contrast

Kernel	*Assign new pixel values using a moving window process*
Texture:	*Standard texture process*
Normal:	*Normalized texture (values 0…1)*

Inverse:	*Inverse Difference Moment (IDM)*
Roughness:	*Rao's diversity based on pixels*
Entropie:	*Rao's Diversity based on classes*
Lowpass:	*Lowpass filter with Gaussian kernel*
Laplace	*Enhance local contrast*

Zones

Index	*Delineated homogeneous image elements*

Features

Features	*extend existing attributes with new properties*
Size:	*Size of single zones given as natural logarithm*
Dendrites:	*Quotient of zone perimeter and cell size*
Diversity	*Spectral diversity for all neighbor zones*
Proportion:	*Size difference between central zone and all neighbors*
Relation:	*Quotient of cell perimeter and number of neighbors*
Append:	*Attributes from all images listed as →import*
Diffusion:	*Smoothe properties in a zonal network*
Raster:	*Control image with one band for each attribute*

Mapping

Pixel	*Classify spectral combinations*
Zonal	*Classify spectral and spatial properties*
Fabric	*Classify spatial patterns of different zones*

Export

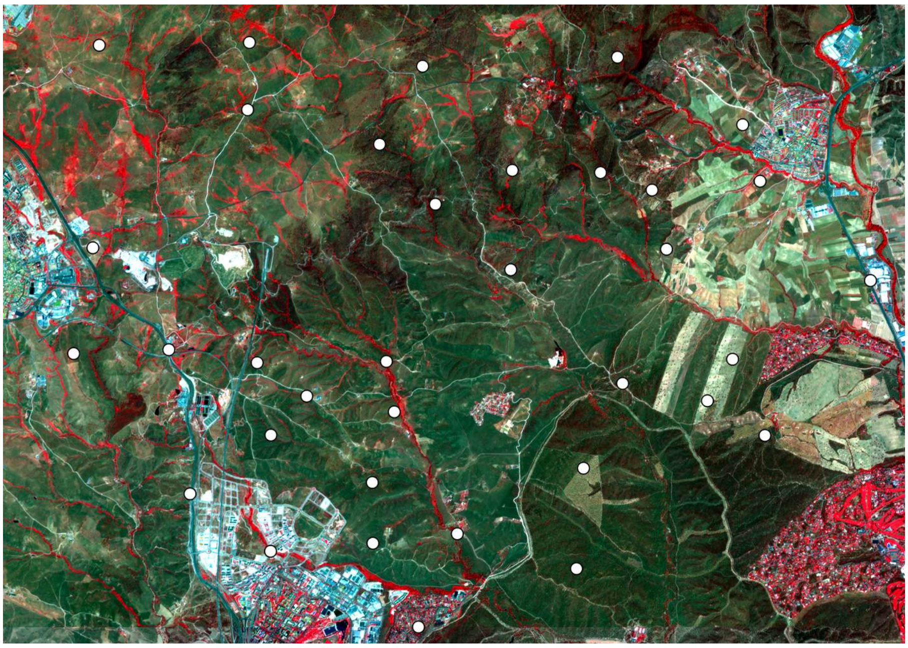
Export	*Transform and store the most recent processing result*

Software Installation

Source	Executables of all commands are available in the directory “binaries” as “xImalys.” “xImalys” was compiled for a Linux environment under Debian / Ubuntu. All commands described here are compiled together. “xImalys” can be called directly as a command. Write permissions are only required for the Imalys home directory.
Source Code	The sources are available in the “units” directory. The code is written in Object Pascal and uses the Free Pascal Component Library and the Free Pascal Run-Time Library, the graphical script editor also uses Lazutils and Lazarus Component Library, which are available from “https://www.lazarus-ide.org/.” The “lazarus dependency package” installs Free Pascal, all required libraries and the GNU debugger.
Imalys Home	“xImalys” creates a directory “~/.imalys” in the user’s home directory. “xImalys” uses this directory as a working directory for intermediate results and for logs. The working directory can be changed at any time with the →Home command at the begin of the command script.
Dependencies	“xImalys” requires simple system commands such as “cp”, Python version 3 or higher and the GDAL library. Python should be a basic part of all Linux installations. “xImalys” expects the GDAL library under “/usr/bin/.” If it is not installed, the GDAL library can be obtained from GitHub “https://github.com/OSGeo/GDAL”, or alternatively directly from the Open Source Geospatial Foundation “https://www.osgeo.org/.”
Interface	“xImalys” does not have a graphical interface, it is controlled by the command line. “xImalys” expects the name of a text file (command script) containing commands and parameters as the first parameter. Commands and parameters are described in the following chapters. Due to the close connection to the GDAL library, Quantum GIS is recommended for control purposes.

Selection and Import of Image Data

In Imalys, all image data is first transformed into a uniform format and stored in a local working directory where the data can be processed quickly.



Selection points and image data for entomlogical investigations north of Madrid

Sensor: Sentinel-2
Bands: 8-4-3 (Infrared)
Date: 12 June 2021
Calibration: TOA Reflectance
Section: Test area
Source: ESA

Image format

The import takes over additional functions. Images can be cropped (→ Clip) or merged, the coordinate system can be changed and the pixel size adjusted (→ Warp). The raw data can be calibrated to reflectance or other defined values such as radiation or heat (→ Scale). The import can read about 30 different image formats.

The typical result are raster data calibrated as reflectance in the working directory. Internally, all images are stored in ENVI format. Imalys thus complies with the requirements of the European Space Agency (ESA).

Combination

If images from different sources are to be processed together, it may be necessary to bring them to a common frame (→ Clip) and to the same pixel size and projection (→ Warp).

Disturbances

Satellite images almost always have gaps due to clouds and other image disturbances. The NASA (Landsat) image data are supplied with a quality filter that marks clouds and cloud shadows with sufficient security (→ Mask). Unfortunately, the ESA offers nothing comparable for Sentinel-2.

Median

Satellite images are taken regularly. The → Median process returns the most common value of a time series. Exceptions such as clouds and cloud shadows are eliminated if the exceptions are randomly distributed over the image. The selection can accept images with minor disturbances, which can greatly increase the number of usable images. The median then returns the most frequent value for a limited period of time. Its combination may be superior to a single picture with a random acquisition time.

Elevation model

Apart from the reflection, other physical characteristics of the depicted surfaces can also be detected with remote sensing. Height, slope and exposure of a surface can be measured from elevation measurements (→Elevation). Elevation data requires special data sources such as the SRTM or the ASTER mission.

Temperature

Landsat includes a sensor that registers the heat radiation of the surface. Heat radiation from Landsat images can be calibrated with →Temperature to temperatures in °K.

Application

The different processes for preparing the raw data can only be called during import.

Home	*Create and select a working directory*

header

each script starts with “IMALYS”

home

creates and initializes a (new) working directory.

Mandatory part of each script

[IMALYS] starts the script

[home] creates a working directory if necessary and empties it

The working directory has been implemented to allow fast processing of data that may only be available through a service or a slow connection. It should be directly accessible. Each script can use its own working directory. Each instance of Imalys needs its own working directory. Write permissions are required for the working directory.

directory = path

below→home creates a new working directory if necessary and gives it the name “path”

clear = true

below→home (option) deletes the contents of the working directory before the next steps

Import:	*Select and calibrate image data*

import

Transforms image data into a common format and stores them as “import” in the working directory

Import and calibrate bands

[import] combines six bands as provided by the USGS to an new image

[clipping] applies a frame given by a shape file

[quality] applies a filter for disturbed pixels

[scaling] transforms the image values to TOA reflectance

```
*recent.imalys
~/.imalys

IMALYS
home
      directory=/home/c7sepe2/.imalys/
      clear=true
import
      image=/home/c7sepe2/Temp/LC08_L2SP_193024_20210901_20210909_02_T1_SR_B2.TIF
      image=/home/c7sepe2/Temp/LC08_L2SP_193024_20210901_20210909_02_T1_SR_B3.TIF
      image=/home/c7sepe2/Temp/LC08_L2SP_193024_20210901_20210909_02_T1_SR_B4.TIF
      image=/home/c7sepe2/Temp/LC08_L2SP_193024_20210901_20210909_02_T1_SR_B5.TIF
      image=/home/c7sepe2/Temp/LC08_L2SP_193024_20210901_20210909_02_T1_SR_B6.TIF
      image=/home/c7sepe2/Temp/LC08_L2SP_193024_20210901_20210909_02_T1_SR_B7.TIF
      EPSG=32633
clipping=true
      frame=/home/c7sepe2/SIPATH/VCT_Sipath/VCT_Sipath_de2_32632_10km_Hull.shp
      EPSG=32633
quality=true
      format=raster
      filter=31
      mask=LC08_L2SP_193024_20210901_20210909_02_T1_QA_PIXEL.TIF
scaling=true
      factor=2.75e-05
      offset=-0.2

Plain Text   Tab Width: 8   Ln 23, Col 1   INS
```

image = file

below → import takes the image “file” into the working directory.

“file” can be a band or a multispectral image. The parameter can be repeated for any number of images. If the coordinates and the frame of the imported images are equal → Import generates a multispectral image from all transmitted bands. If this is not the case the process tries to overlap equal bands of the different images to one result image (→ merge).

Many images contain undefined areas and image errors. → Import tries to detect undefined areas and sets them to NoData. Imalys considers Zero as a valid value. NoData is ignored in all cases.

lowband = number

[1…N] below → import selects only bands beginning at “number” (option)

highband = number

[1…N] below → import selects only bands up to “number” (option)

Many providers store each image band in a separate file. In this case, only selected bands need to be imported. If this is not the case, the required bands can be selected individually.

extend = path

below → import searches in the directory “path” for metadata concerning the acquisition time of the image (Option)

The date of the image acquisition can be hard to find in the provider’s metadata. Imalys automatically takes them from the file name and saves them in its own metadata (ENVI header). All processes that perform time comparisons need this information.

Imalys can recognizes the date only if the file name of the provider has not been changed. The date can be recorded subsequently with → Extend. The command requires a pathname to the provider’s metadata.

clipping = true	*Restrict import to a selected frame (option)*
clipping = true	below →Import activates the image section (option)
frame = geometry	creates a rectangular frame from all points in “geometry”
	The frame can be passed as a polygon or as a collection of vector points. The projection of the frame is automatically adjusted to the image.
	The result is always a rectangular frame around all vertices of the given geometry. The process ignores vertices outside the image. The result is the intersection of image and frame. Empty intersections cause an error.

quality = true	*Mask out image errors and clouds (option)*
quality = true	below →Import activates the quality mask (option)
	→Mask transfers ESA and USGS image quality masks to the image data. Individual settings “filter” can be selected. The mask sets all pixels selected by the parameter “filter” to NoData. NoData is ignored by all processes.
format = raster | vector	NASA / USGS deliver a quality mask in raster format for Landsat, the ESA mask for Sentinel-2 is a polygon.
filter = number	[1…N] below →Import uses a binary filter to select certain features from the Landsat mask. Securely recognized clouds and cloud shadows need “number” = 31. For Sentinel-2 the mask discriminates between clouds and haze.
mask = file	uses the file “file” as a mask

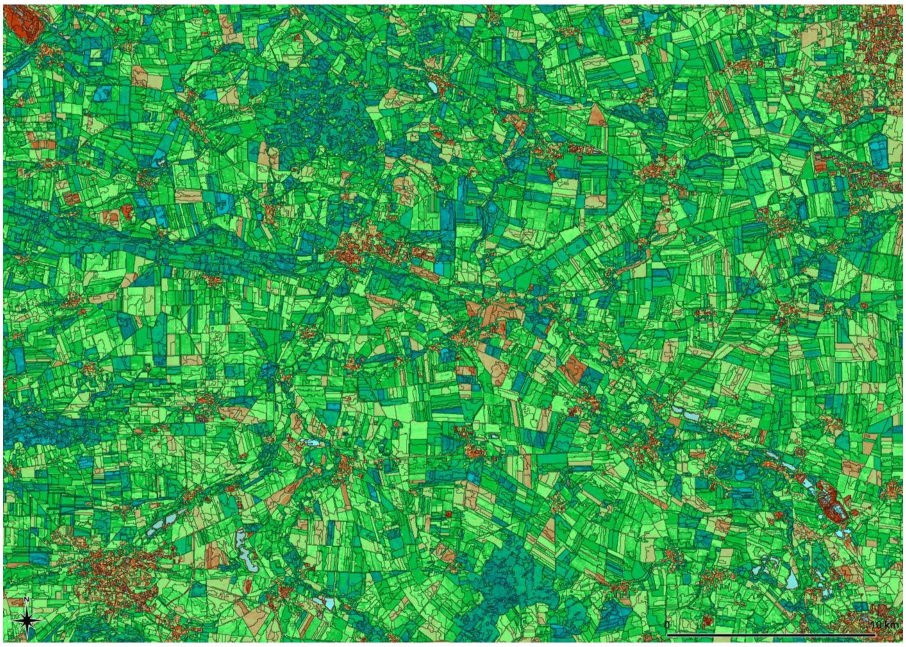
scaling = true	*Value calibration and atmospheric correction (option)*
scaling = true	below →Import activates the calibration (option)
	Images from public providers (ESA, USGS) include calibration data that can be used to calculate reflectance, radiation, temperature, altitude, etc. from the raw data. “Reflectance” is defined as the ratio between incoming and reflected light. It is a pure number between 0 and 1. “Radiation” is the energy emitted by a surface in [W/m²]. The influence of the atmosphere is corrected to “Top Of Atmosphere” (TOA). Increasingly, additional parameters such as haze, sun angle, azimuth or the distance between sun and earth are available.
factor = figure	[R > 0] sets the scaling factor to “figure”
offset = figure	[R] sets the offset to “figure”
	As an example, two values for converting the raw data to reflectance are given:

Landsat 5,7,8	Factor: 2.75e-4	Offset: -0.2
Sentinel-2	Factor: 1e-4	Offset: 0

	The values depend on the sensor and the distribution. →Scale implements equal factor and offset for all bands. For older Landsat distributions, each band must be calibrated with a different parameter.
	Landsat is currently the only freely accessible sensor that provides temperature data with high spatial resolution.
Warp:	*Reprojection with high level interpolation*
warp = true	below →import initializes a reprojection of the image data
	Optical satellite images are supplied as bands with square pixels, almost all are projected on an elevation model. All Imalys functions that determine kernels, neighborhoods or areas assume projected data with square pixels. →Warp can change the projection of the image, e.g. from geodetic to UTM. →Warp can also change only the pixel size.
EPSG = number	[1…N] below →import passes the EPSG code “number” as a new projection
size = figure	[R > 0] below →import selects the pixel size “figure” for the length and width of the pixels (option)
style = lanzcos	selects an interpolation according to Lanzcos (option)
	The reprojection of image data can be done “soft”, i.e. with interpolated values, or “hard”, according to the “nearest neighbor” principle. Imalys projects scalar image data (reflection, elevation) softly with a cubic spline interpolation and thematic data (maps, classes) hardly. Alternatively, an interpolation according to Lanzcos can be chosen.

Indices and Pixel Arithmetics

The ratio of two, three or rarely four bands is traditionally used as an indicator for features that are not directly visible. The most well-known are the vegetation indices (NDVI, EVI, LAI, ...). These indices are based on the reflections at different frequencies (bands) and can be applied to individual pixels.



Vegetation Index
Near Infrared Vegetation NIRv

Narrow black lines mark boundaries of land use (zones).

The NIRv values range between 0.0 (turquoise) and 0.4 (dark green)

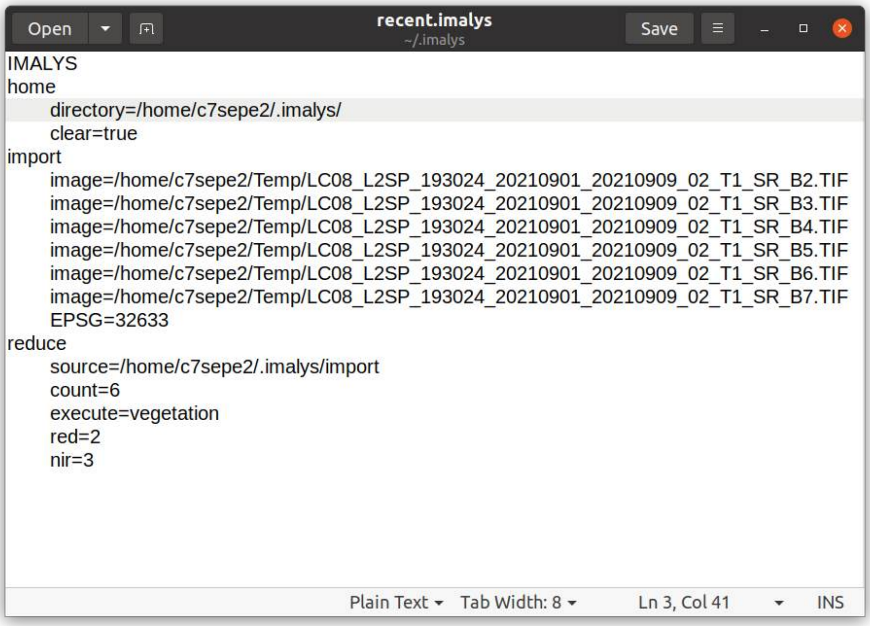
"Hohes Holz" and "Großer Bruch" in the Bode catchment area, Landsat-8, First half of the growing season 2014-2020

Vegetation index	Imalys has implemented two vegetation indices, one for the metabolic rate of green plants (→NIRv) and one for the relative coverage of the landscape with green leaves (→LAI). The user should take into account that all vegetation indices are approximations. They assume spectral dependencies and also depend on the type of sensor.
Soil moisture	In addition to special data sources, there are numerous other methods to derive proxies for features such as soil moisture from the visible image data. Imalys implements an estimation for soil moisture →Moisture.
Reduce	*reduce multitemporal or multispectral images*
source = file	below →reduce links the image "file" to the "process"

Calculate a vegetation index

[import] provides six USGS bands

[reduce] transforms them to a single band with the vegetation index NIRv



```
IMALYS
home
	directory=/home/c7sepe2/.imalys/
	clear=true
import
	image=/home/c7sepe2/Temp/LC08_L2SP_193024_20210901_20210909_02_T1_SR_B2.TIF
	image=/home/c7sepe2/Temp/LC08_L2SP_193024_20210901_20210909_02_T1_SR_B3.TIF
	image=/home/c7sepe2/Temp/LC08_L2SP_193024_20210901_20210909_02_T1_SR_B4.TIF
	image=/home/c7sepe2/Temp/LC08_L2SP_193024_20210901_20210909_02_T1_SR_B5.TIF
	image=/home/c7sepe2/Temp/LC08_L2SP_193024_20210901_20210909_02_T1_SR_B6.TIF
	image=/home/c7sepe2/Temp/LC08_L2SP_193024_20210901_20210909_02_T1_SR_B7.TIF
	EPSG=32633
reduce
	source=/home/c7sepe2/.imalys/import
	count=6
	execute=vegetation
	red=2
	nir=3
```


The image source must have been saved with →Import.

execute = process	below →reduce applies “process” to the source “file” and saves the result to the working directory. The result is named like the process.
	→Reduce can run more than one process at a time. The source is shared by all processes. The results are stored as different layers with the name of the process.

NirV:	*Vegetation index „Near Infrared Vegetation“ (NIRv)*
	$(v_i - v_r) / (v_i + v_r) \cdot v_i$ *v: value (reflection); r,i: color red, infrared;*

execute = vegetation	below →reduce activates the vegetation index “Near Infrared Vegetation” (NIRv) for the metabolic rate of green plants
	The NIRv is based on the fluorescence emitted by the photosynthesis of green plants. Fluorescence is easily visible, but infrared radiation can also come from other sources so there is room for misinterpretation. The process requires an indication which bands reflect the color red and near infrared.
red = number	[1…N] below →reduce takes the band “number” as color red
nir = number	[1…N] below →reduce takes the band “number” as Near Infrared

LAI	*Proxy for Leaf Area Index*

execute = LAI	below →reduce activates the vegetation index LAI for covering the surface with leaves.

	The leaf area index is defined as the total leaf area per unit area, even if leaves overlap several times. The estimate must be calibrated for each different land cover. If time series are provided, →NirV and →LAI are calculated from the first principal component (→Principal) of the Red and Infrared bands of all provided images.
red = number	[1…N] below →reduce takes the ban “number” as color red
nir = number	[1…N] below →reduce takes the band “number” as Near Infrared

Moisture	*Soil moisture value*

execute = moisture	below →reduce calculates the soil moisture from ...
	Special combinations of optical data have been suggested to derive soil moisture from image data, others use heat radiation or radar backscatter. The ESA has launched a specialized sensor.

Principal:	*First principal component, dimensional reduction*
	$\sqrt{\sum_i v_i^2}$ *v: pixel value* *i: bands*

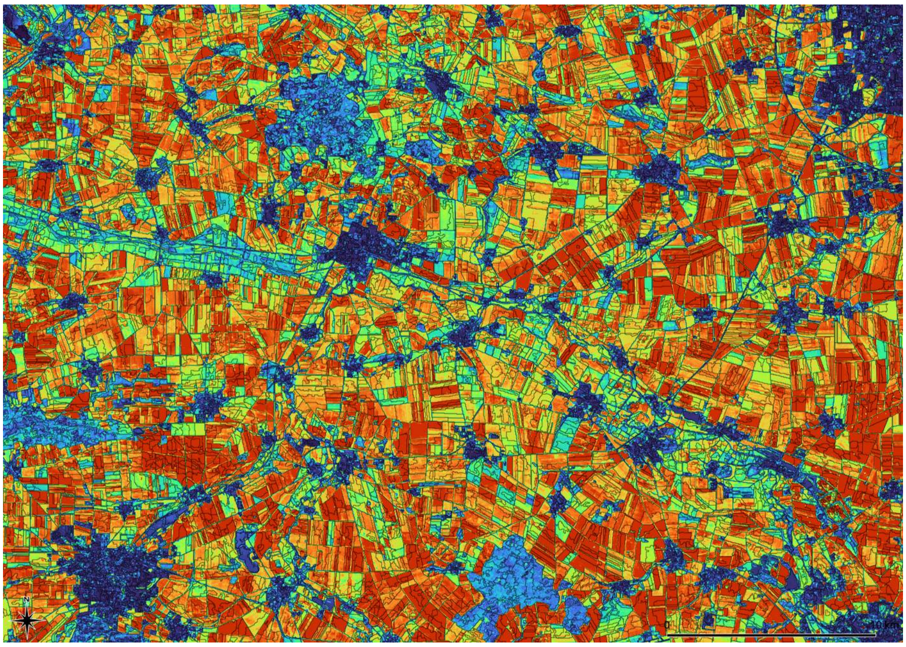
execute = principal	below →Reduce calculates the first principal component of all bands from “source”
	→Principal reflects the brightness or density of all image bands. Imalys uses the first principal component of all bands as brightness in several other cases.

Time Series and Periods

Image data from remote sensing are snapshots. For time series they will be analysed individually. To characterize a typical state they will be summarized by means of a statistical process. → Reduce can merge selected images or bands to new images. With “typical” features or extract statistical indicators.

Median

Clouds are almost everywhere random in time. If the majority of single pixels are undisturbed, the median returns a typical value for the provided period of time, even if the acquisition time of each pixel is unknown.



Variance Over 7 Years

Variance of mean values over the growing season 2014-2020. Values between 0.0 (blue) and 0.52 (red). Small changes in settlement and forest areas are clearly visible.

“Hohes Holz” and “Großer Bruch” in the Bode catchment area, Landsat-8, Growing season 2014-2020

Season

In mid-latitudes there are pronounced seasonal periods. The seasonal changes are often greater than the changes over the years. In order to depict typical features of a landscape or to search for outliers, the season of the images should be taken into account. The → median fails when large natural or anthropogenic changes such as harvest occur. Images from equal seasons at different years provide images with typical features for the chosen period.

Time intervals

If time series filtered with the → median are used, the majority of the recordings must be error-free for each single pixel. With Landsat there are 1-2 usable recordings per quarter in almost each year and with Sentinel-2 at least twice as many. Sentinel 2 and Landsat 8 have been in operation since 2016 and 2014 respectively. Time series with several dozen recordings can be made. Landsat TM and OLI allow time series over 35 years.

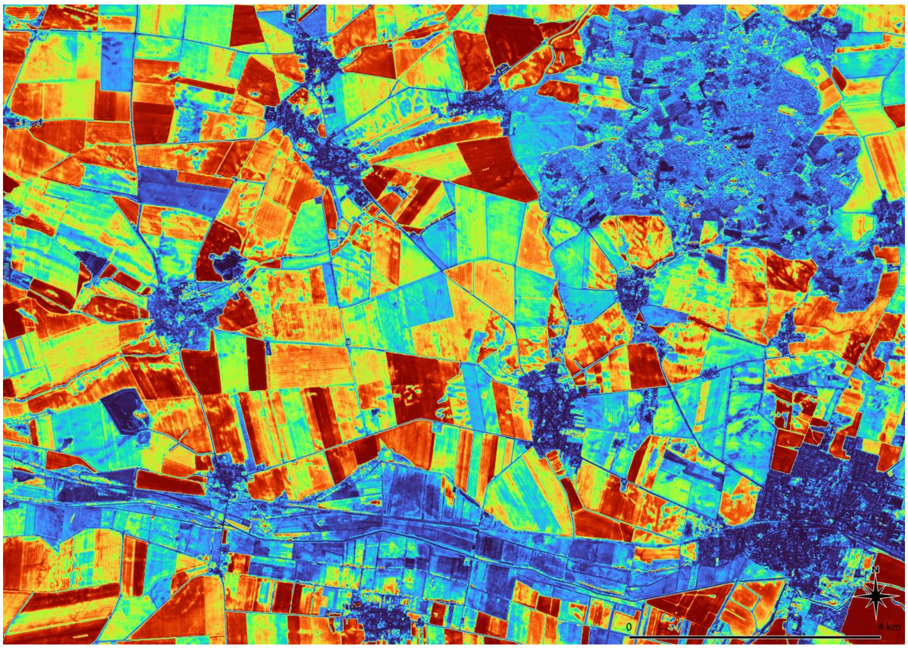
Classes

The extent and direction of periodic changes is a specific feature of many landscape types. Settlement areas show little periodic change, deciduous forests show significant but regular fluctuations and arable land both changes, high and irregular. Permanent grassland can be depicted by its periodicity.

Change

→Reduce provides routines to look for changes. “Difference” compares all pixels in two pictures, “Variance” and “Regression” detect typical fluctuations and quantify trends. Both require at least three different time periods or states.

Outlier in time series are often also outlier in the space, e.g. construction sites or forest fires. Outliers in space are easy to see, especially if previously visible boundaries (→Zones) have been created. If possible, the analysis should always use both aspects.



Variance of Differences

Time periods May – July and August – October over the years 2017 to 2021

Settlements show the smallest seasonal variance, forest (NO) and permanent grassland (S) moderate variance. Agriculture show all grades (harvest)

Sensor: Sentinel-2
Years: 2017-2021
Bands: 2, 3, 4, 8
Values: 0.0 (Blue) – (Red) 0.014

Optical sensors are not suitable to detect changes over days or weeks. Radar (e.g. Sentinel-1 in C-Band) provide an image (→backscatter) every 2-3 days. Using radar the date of a rapid change (e.g. harvest) can be determined but the nature of the change has to be recognized in another way.

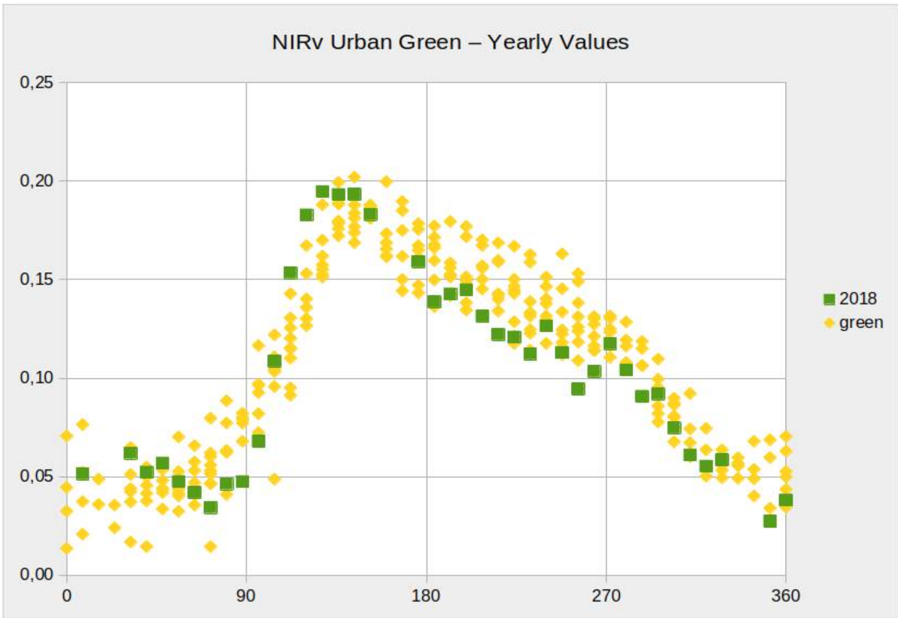
Radar backscatter and polarization are very different from optical images. Smooth, built-up objects usually show a high backscatter. Corner reflectors e.g. power lines can mimic much larger objects.

Sensors such as MODIS Terra provide optical data on a daily basis, but with at least 10 times lower spatial resolution than Landsat-8 or Sentinel-2. Combining temporal high-resolution images with spatial high-resolution images allows to detect sudden changes and help to adjust single recordings to typical states of annual changes thus making random recording dates easier to interpret.

Annual cycle for the vegetation index NIRv

Weekly survey of all green areas in Leipzig over the years 2011-2020. Measurements in the dry year 2018 are highlighted in green.

MODIS Terra, Public green spaces in Leipzig



Mean:	*Arithmetic mean, bands or images* $(\sum_i v_i) / n$ *v: value; i: intems* *n: item count*

execute = mean — below → reduce determines the average value of all bands from “source”

→ Mean is the arithmetic mean of all provided image bands. For multispectral images, → mean is calculated for each band separately. The process returns a multispectral image of mean values.

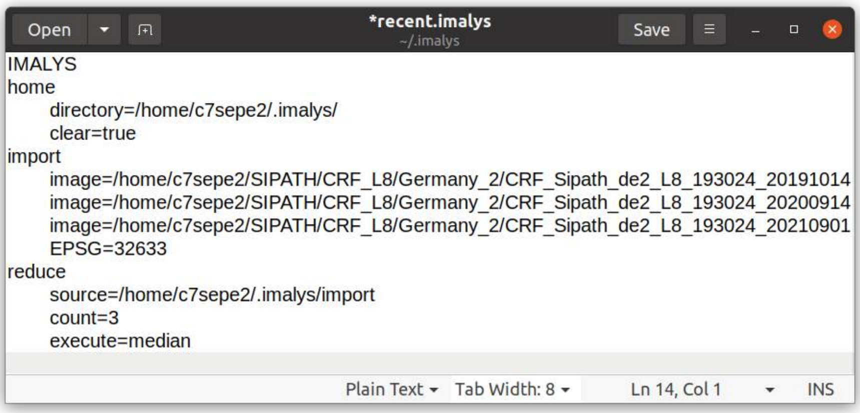
Median:	*Most common values from image stacks* *– sort values* *– chose value at the center*

execute = median — below → Reduce determines the median of all bands from “source”

Median of time series

[import] re-imports tree calibrated images of different years

[reduce] returns the median of the tree images for each band




```
IMALYS
home
	directory=/home/c7sepe2/.imalys/
	clear=true
import
	image=/home/c7sepe2/SIPATH/CRF_L8/Germany_2/CRF_Sipath_de2_L8_193024_20191014
	image=/home/c7sepe2/SIPATH/CRF_L8/Germany_2/CRF_Sipath_de2_L8_193024_20200914
	image=/home/c7sepe2/SIPATH/CRF_L8/Germany_2/CRF_Sipath_de2_L8_193024_20210901
	EPSG=32633
reduce
	source=/home/c7sepe2/.imalys/import
	count=3
	execute=median
```

The →Median is defined as the most common value of all inputs. For each pixel, the process sorts all bands according to their value. The most common value is then in the middle of the sorted order. The result is the "typical" value within the selected time period.

If multispectral image series are provided →Median forms the result for each band separately and returns a multispectral image with the median for each band.

Outlier:	*Detect values far from standard deviation*

execute = outlier

below →Reduce determines pixels with the strongest deviation from the mean

Single, strongly divergent pixel or periods in a time series can be found with classic statistics if the source data are distributed approximately normally. In the case of optical data, this is the case.

Difference:	*Eukedian distance of two n-dimensional properties*

execute = difference

below →Reduce returns the difference between two bands or images.

→Difference returns the difference between the values of two images or bands. Exactly two images or bands must be provided. For multispectral images →Difference generates a result for each band separately and returns a multispectral image of the differences for all bands.

Variance:	*Variance based on standard deviation*

$$\left(\sum_f v^2 - \left(\sum_f v\right)^2/n\right) / (n-1)$$

v: value; n: feature count
f: features (bands)

execute = variance

below →Reduce determines the variance in all bands of "source"

Satellite images of optical sensors are recorded regularly and time series are generated. The →Variance command determines the variance of individual pixels based on a standard distribution for all bands in source.

For multispectral images, →Variance determines the variance for each band separately and returns the result as a multispectral image of the variances. The result can be further reduced to a one band image with the first principal component of all bands using →Principal.

For this calculation, Imalys uses an identical transformation of the Gaussian equations, which is easier to calculate and allows data sets of arbitrary size.

Regression:	*Regression based on standard deviation*

$$\left(\sum_f t \cdot v - \sum_f t \sum_f v/n\right) / \left(\sum_f v^2 - \left(\sum_f v\right)^2/n\right)$$

v: cell value; n: feature count
t: time; f: features (bands)

execute = regression

below →Reduce returns the regression of all bands in "source"

→Regression returns the regression of individual pixels of all bands in source. →Regression uses the temporal distance of the recordings from the metadata of the images. To do this, the images must have been imported with →Import or have been dated afterwards with →Extend.

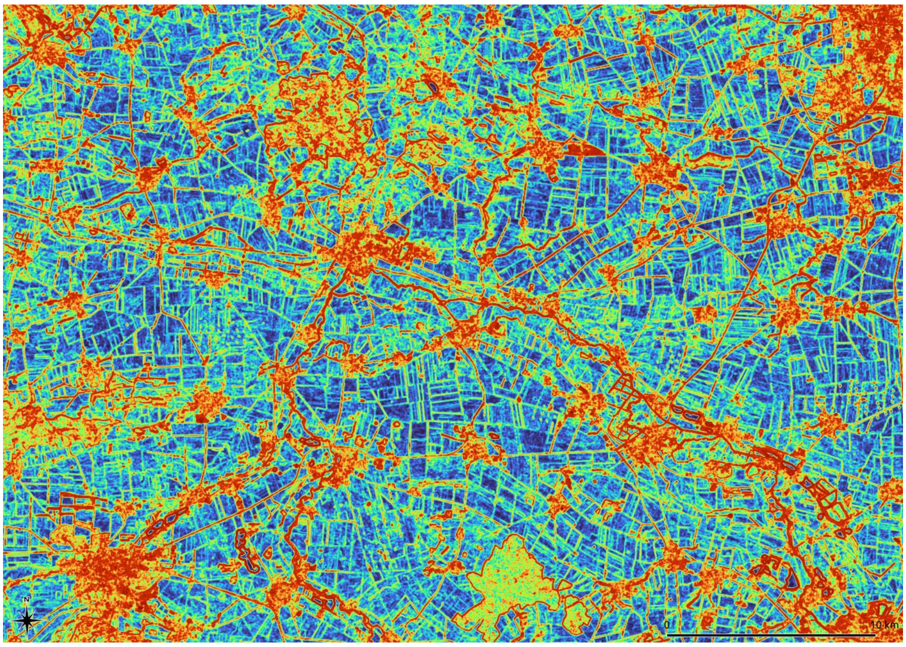
Similar to →Variance →Regression determines the regression for each band separately if multispectral images are provided and returns a multispectral regressions. Similar to the →Variance process, the selection of the scenes must include all important development states for a reliable result.

Kovarianz

HotSpots:	*Spatial isolated feature combination*

Diversity and Contrast

Kernel processes assign a new value to each pixel in each band. The value is compiled from a small window around a central pixel of the image (kernel). Kernels can be used to determine the local roughness of an image but also to modify the contrasts or enhance an elevation model.



Normalized Texture

First principal component of the normalized textures of all optical bands in a 5x5 kernel.

Sensor: Landsat-8
Values: 0 (blue) to 0.42 (red)
Growing season 2014-2020

“Hohes Holz” und “Großer Bruch” in the Bode catchment area,

Concerning biology, diversity is defined as the probability to register different species at one place. Indicators of landscape diversity have been a focus of development. Imalys offers the choice between different methods to estimate landscape diversity from image data.

Spectral diversity or texture is traditionally used as a measure of ecological diversity In remote sensing. Imalys provides additional methods to quantify spatial distribution, shape and temporal changes of depicted structures.

Texture

Textures return the “roughness” of an image. In the simplest case, this is the spectral difference between adjacent pixels in a small window (kernel). The kernel is systematically dragged over the whole image. Each kernel defines one result pixel. In addition to the classic texture, the normalized texture (→Normal) and the Inverse Difference Moment (IDM) according to Haralik (→Inverse) are also implemented.

Rao’s Diversity

Texture as a measure of diversity has the disadvantage that even a monoculture can show a “rough” surface with a high texture. Rao’s approach evaluates spectral and spatial differences simultaneously. Regular patterns show lower values as a texture would return. Unfortunately Rao’s approach is complex to calculate.

Imalys implements two variants of Rao's approach. The pixel-oriented version →Roughness differs little from the texture in practice. The class-oriented version →Entropy is closer to the biological definition, but requires a classification of land cover.

Rao's Entropy

→Entropy below →Mapping can cluster image data fully automatically and uses the result to calculate diversity using Rao's approach. A clustering of the spectral combinations is sufficient because the nature of the classes need not be known. The process only needs to determine the frequency of the different classes in the kernel and their spectral distances. Rao's →Entropy can only be called below →Mapping, since the classification and diversity have to be computed together.

Zones

Diversity can also be calculated for zones (→Diversity). Since zones combine small-scale regions, this process resembles Rao's approach with classes.

Contrast

Structural features of image data can be strengthened or weakened using kernels. Imalys implements a Laplace transformation to amplify small-scale structures (→Laplace) and the opposite, a LowPass filter with a Gaussian kernel to enhance larger structures but also reduce noise (→LowPass)

Kernel	*Assign new pixel values using a moving window process*

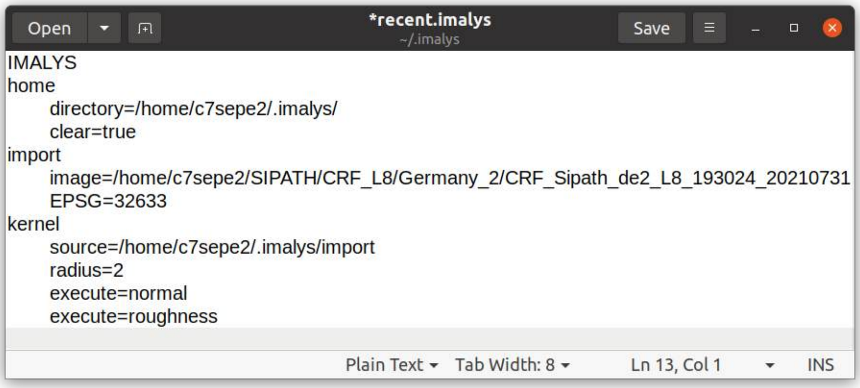
kernel

The command initializes a kernel, associates it with image data (source) and saves the result under the name of the kernel process (execute = ...).

Apply two kernel processes

[import] re-imports a calibrated image

[kernel] applies the kernel processes (normal, roughness) using the same image




```
IMALYS
home
	directory=/home/c7sepe2/.imalys/
	clear=true
import
	image=/home/c7sepe2/SIPATH/CRF_L8/Germany_2/CRF_Sipath_de2_L8_193024_20210731
	EPSG=32633
kernel
	source=/home/c7sepe2/.imalys/import
	radius=2
	execute=normal
	execute=roughness
```

source = file

links the image "file" to the process

All →Kernel processes require image data stored with →Import.

If multispectral images are provided →Kernel first forms the first principal component of all bands and thus calculates the kernel. The result always consists of one band.

radius = number

[1…N] below →Kernel controls kernel radius. The "radius" is the number of pixels between the central pixel and the kernel edge. The pixel in the center is not counted.

For all kernels a radius must be specified. The kernel diameter is (R·2+1) with “R” as radius. “R=1” as radius gives a kernel of 3x3 pixels, “R=2” a kernel of 5x5 pixels (radius + center + radius).

Texture: *Standard texture process*

$$\sqrt{\sum_{b} (v_i - v_j)^2}$$

V: pixel value; i,j: neighbors; b: bands

execute = texture

below →Kernel creates a new image with the result of the →Texture kernel. The kernel uses a Gaussian standard distribution, large kernels are approximated by an iteration, otherwise the process times could become extremely long.

→Texture is the usual form to determine diversity

Normal: *Normalized texture (values 0…1)*

$$\sqrt{\sum_{b} ((v_i - v_j) / (v_i + v_j))^2}$$

r: pixel value; i,j: neighbors; b: bands

execute = normal

below →Kernel creates a new image with a texture normalized to brightness. →Normal uses the modulation of (p1-p2) / (p1+p2) with p1 and p2 as values of two pixels instead of the difference (p1-p2) as →Texture does.

Textures based on spectral differences show very low values in dark regions (forests) and very high ones in bright regions like industrial buildings. →Normal determines the texture relative to the brightness.

Inverse: *Inverse Difference Moment (IDM)*

$$\sqrt{\sum_{b} 1 / (v_i - v_j)^2}$$

r: pixel value; i,j: neighbors; b: bands

execute = inverse

below →Kernel creates a new image with the Inverse Difference Moment (IDM) proposed by Haralik (ZZZ).

The IDM is particularly high in dark regions and low in bright regions. It can complement →Texture and has proven itself in the analysis of settlement structures.

Roughness: *Rao‘s diversity based on pixels*

$$V_p = \sum_{i\,j} d_{ij} \times p_i \times p_j$$

Vp: Pixel value; d: spectral distance; p: frequency; i,j: pixel combinations

execute = roughness

below →Kernel creates a new image with Rao’s ß-Diversity on pixel basis.

As →Texture does, Rao's approach evaluates the spectral difference of individual pixels, but compares not only neighboring pixels but all pixels within the kernel. Small-scale patterns that repeat in the kernel do little to increase the result.

Entropie:	*Rao's Diversity based on classes*	
	$V_p = \sum_{k\,l} d_{kl} \times p_k \times p_l$	*V_p: Pixel value; d: spectral distance; p: frequency; k,l: class combinations*

execute = entropy — below →Mapping creates a new image with Rao's Diversity based on an automatic classification.

The →Entropy process requires a classification of the image data. The easiest way to call the process is together with the classification. Therefore →Entropy can only be called below →Mapping. The parameters used to classify pixels must also be specified. Alternatively →Entropy can use an existing classification, which can be specified with the parameter "external."

model = pixel — →Mapping creates a new image with an automatic classification based on pixels (see →Mapping)

features = number — [1…N] below →Mapping creates "number" clusters

samples = number — [1…N] below →Mapping uses "number" samples to define the classes

external = directory — links the classification in "directory" to the →Entropy process. The class definition is not recalculated. The classification in "directory" must be pixel based.

Elevation:	*Height, slope and aspect of terrain surface*

Elevation data are not very variable and were therefore collected in campaigns. All satellite recordings are based on radar interferometry. The results of the SRTM mission are meanwhile available worldwide in the highest resolution. ASTER data is available with approx. 100 meters pixel size. Combined data sets for optimal accuracy are available.

In addition to the absolute height (→elevation), the slope (→slope) of the landscape can be used as an indicator. The exposure (→aspect) may be strongly correlated with vegetation.

Kernel definitions can be too complex to be clearly presented in a form. In the following the kernel is symbolized by a ⨀ (dot-operator) where each pixel within the kernel is multiplied with a different factor.

Lowpass:	*Lowpass filter with Gaussian kernel*	
	$V_p = \bigodot_{R,\,i,j}(d_{ij})$	*$\bigodot_R$: Gaussian Kernel Product, Radius „R"; V_p: Result value; d_{ij}: Pixel densities at kernel position i,j*

execute = lowpass — below →Kernel reduces the local contrast of the image data according to the selected "radius"

→LowPass uses a kernel ⨀ with a normalized Gauss distribution. The kernel radius is set to two standard deviations. The process reduces the local contrasts with the Gaussian distribution in the kernel and can fix small bugs. The kernel size can be selected freely. Imalys implements large kernels through an iterative process to significantly reduce the processing time.

Laplace	*Enhance local contrast*	
	$V_p = \underset{i,j}{\odot_R}(d_{ij}) - \underset{i,j}{\odot_S}(d_{ij})$	*$\odot_{RS}$: Gauß Kernel Product, Radius R (large), S (small)* *V_p: Result value; d_{ij}: pixel density at kernel position i,j*

execute = lapalce — below →Kernel enhances the contrast of the image according to the selected inputs “inner” and “outer”

A Laplace Transformation enhances the image contrast and can make lines and closed shapes clearly visible. Imalys implements the transformation as the difference of two Gaussian distributions with different radius.

inner = number — [1…N] sets “number” as the radius for the inner →Kernel radius

outer = number — [1…N] sets “number” as radius for the outer →Kernel radius

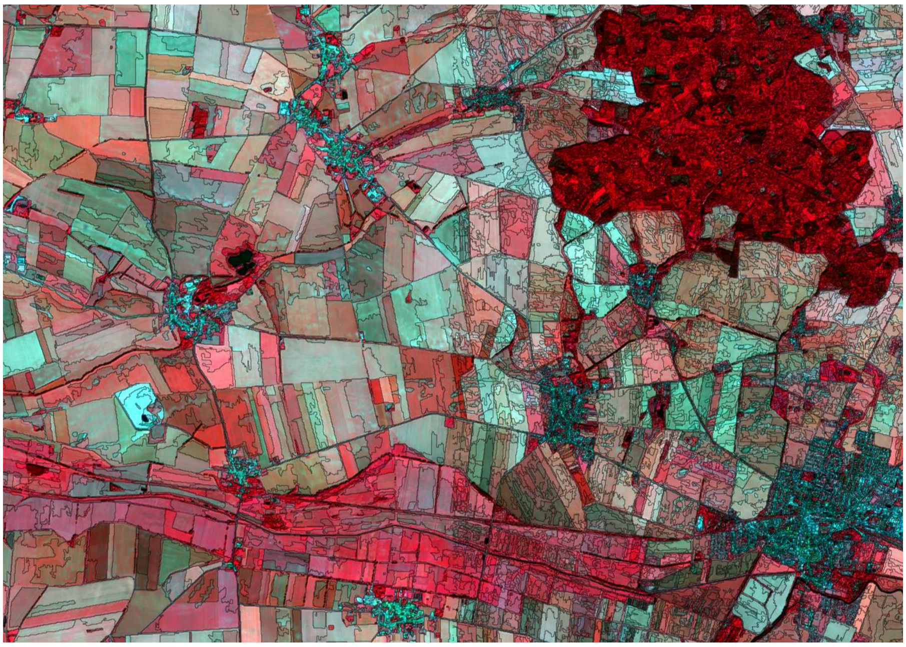
Zones

Imalys implements a process that divides images into zones with broadly homogeneous spectral characteristics (→ Index). The process depends only on the image data and a parameter for the mean size of the zones. Zones have a geometry, attributes and individual neighbors. The borders between zones represent borders of land use. The local density of the borders is used as an ecological indicator ("effective mesh size" or "coherence degree" (ZZZ)).

Structural landscape elements

Zones with different combinations of optical characteristics are marked by narrow dark lines. After creation zones can be processed independently of the image data.

"Hohes Holz" and "Großer Bruch" in the Bode catchment area, Landsat-8 images, first half of the growing season 2014-2020



Using pattern analysis in the image, areas with largely the same optical characteristics can be defined (→ Index). They are hereinafter referred to as "zones." The technique is also known as Object Based Image Analysis (OBIA) (ZZZ).

Delineation

The process is based on an iterated watershed algorithm. Details are explained in cape. "Methods". The algorithm can process any type of image, regardless of the image source (microwave, altitude data, light), the scale or the number of bands. The algorithm sets borders preferably at places with maximum contrast.

Size, Borders

The average size of the zones "size" can be freely selected. The process starts with "zones" of individual pixels and gradually removes borders between existing zones until a threshold is reached. The order in which the borders are removed does not depend on the final size. This means that larger zones can serve as second-degree order for smaller ones. The larger borders are always the same.

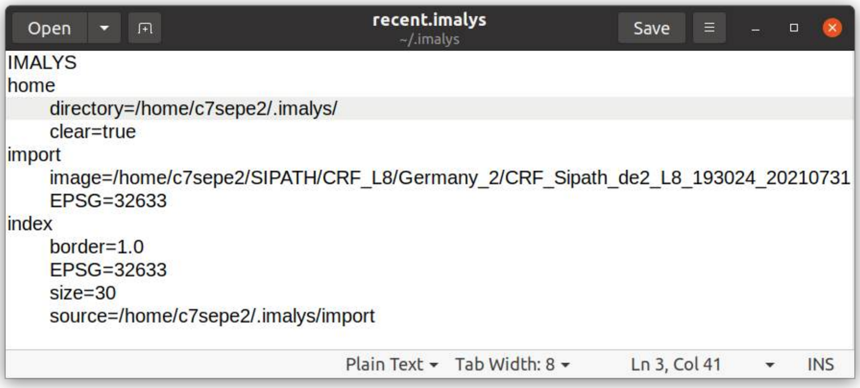
Input data	If bands with very different value ranges are used, the bands with the largest numerical values dominate the position of the border. It may be useful to scale the values of the different bands in front of the processing (→Equalize) so all features will have the same influence on the result.
Definition	The internal definition consists of an image with the zone ID as value (index, index.hdr), a WKT file with the coordinates of the borders (vector.csv), an attribute table (index.bit) and a table of all links between adjacent zones (topology.bit).
Attributes	???

Index	*Delineated homogeneous image elements*
index	The command creates a seamless network of zones that completely covers the image. The zones are assigned with the spectral signatures of the image data as attributes. The attributes can be expanded (→Features).

Crate Zones

[import] provides a calibrated image

[index] delineates zones from the import

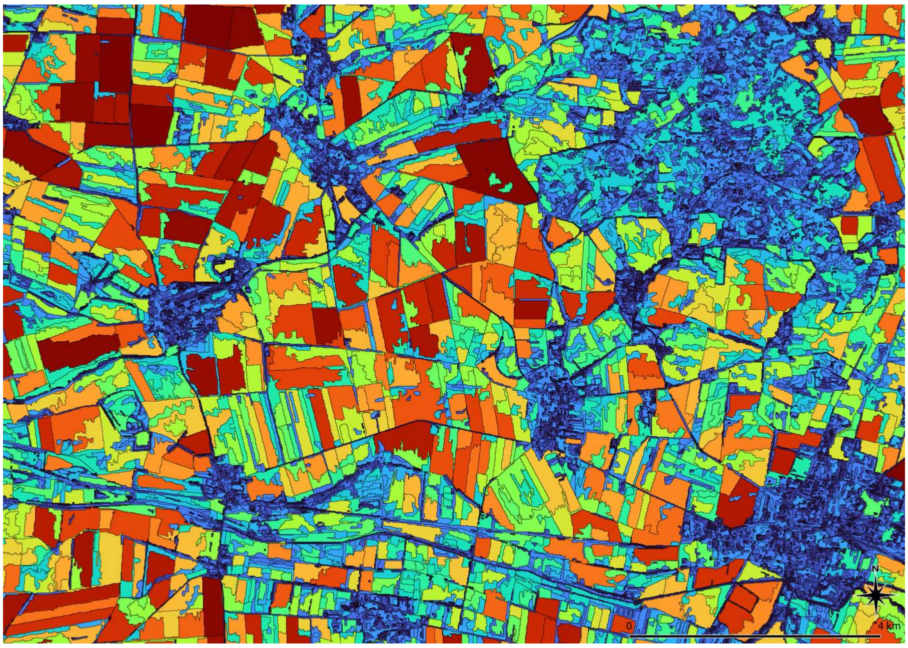


	→Index stores the geometry, attributes and contacts of the zones in an internal format in the working directory. →Export copies the internal data to another directory and creates a vector file. The vector format can be selected
source = image	below →Index links all bands from “image” to the process. The process assumes that the image data are included with →Import. The number of bands is not limited.
EPSG = number	[1…N] below →Index provides the projection of all imported raster and vector data to the process. The input is mandatory, but mainly serves for control purposes.
size = number	[1…N] below →Index uses “number” as the mean size of the zones. The input is not an absolute value but is combined with the image contrast to control the process. “size = 1” returns very small zones, the parameter is not bounded upwards.

border = figure	[0 < R ≤ 1] below →Index regulates the sensitivity for color contrast. With values close to zero, very large zones can be created. Zero itself is excluded. The default is One (option).
external = directory	takes over zones from the folder “directory”. “Directory” must have been created with the →Export command.

Zonal Attributes

Structural features of the landscape can be derived from geometry and connection of the zones. Some features describe individual zones such as →Size (area) or →Dendrites (shape), others describe the connections to adjacent zones such as →Relation (density of neighbors) or →Diversity (spectral differences). With →Features, additional image data can be transferred to existing zones as new attributes.



Dendritic Cell Form

Ratio of volume and area of single zones.

Small or narrow zones have small values, large or compact zones have large values. The quotient does not depend on the parameters of cell formation.

Process: Dendrites
Values: 0.05 – 1.0 (blue – red)
Location: Bode catchment area
Sensor: Sentinel-2
Years: 2017-2021, May – Oct.

Attributes are organized in a table like attributes of vector data. Attributes can also be exported as an image using the →Raster process.

Size

The size of the zones (→Size) depends on a selectable parameter and thus can't be used to compare images. Therefore all other structural attributes return relative values that are much less dependent on the absolute size of the zones.

Shape and size

The process →Index automatically adopts all spectral bands of the selected images as attributes for the zones. →Features adds new features to the existing attributes. Shape and connections of the zones are obtained directly from the geometry of the zones. In addition arbitrary image data such as height or →Kernel results can be linked to the zones as new attributes. →Features was therefore implemented as an independent process. The command can be invoked as often as desired with different parameters.

Connections

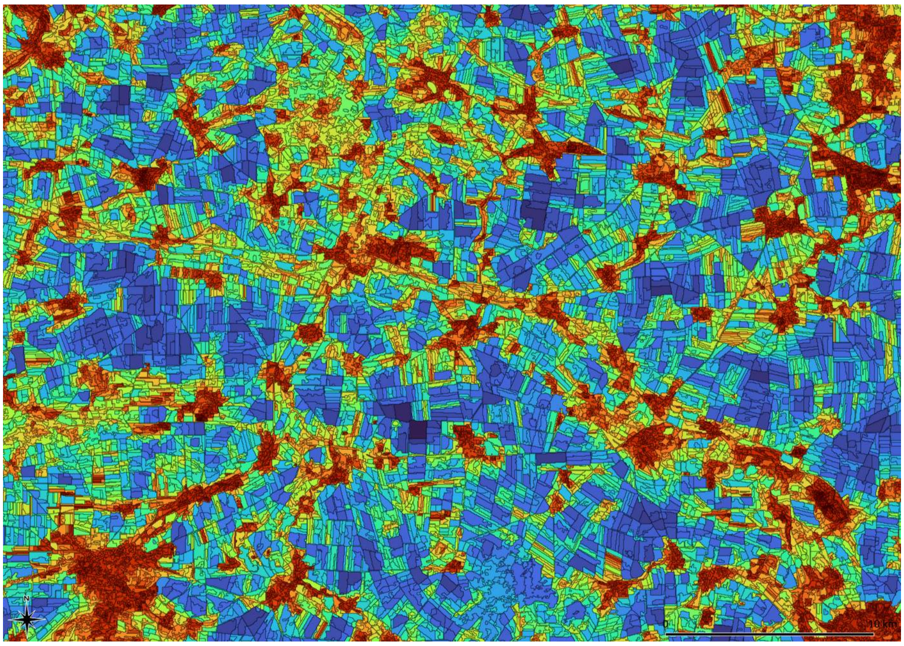
Two other processes compare shape and size of the zones with the local environment. →Proportion returns the ratio of the central zone to all connected zones. The value is greater than one if the central zone is larger than the

	mean of the neighbors. →Relation returns the ratio between perimeter and number of connected zones. As with →Dendrites, size and environment mix in the value of →Relation.
Local concentration	Shape and size of the zones are rarely randomly distributed, they form regions or corridors with similar characteristics. →Diffusion strengthens locally dominant features similar to a low pass filter and thus makes focal points more visible. The algorithm follows Alan Turing's suggestion to understand patterns as a consequence of diffusing fluids (ZZZ).

Local concentration for the Dendrites attribute

Regional balance of values over 5 levels enhances the visibility of focal points and corridors.

Process: Dendrites
Process: Dissemination
Years: 2014 – 2020
Values: 0.09 – 5.6 (red – blue)
Sensor: Landsat-8
Site: Bode catchment area



If corridors, i.e. paths with favorable conditions for exchange, are to be found, a hydrological drainage model →Runoff is available, which can not only use elevation data. A classification (→Pixel, →Zonal) or image objects (→Fabric) can also be used to record large scale spatial links.

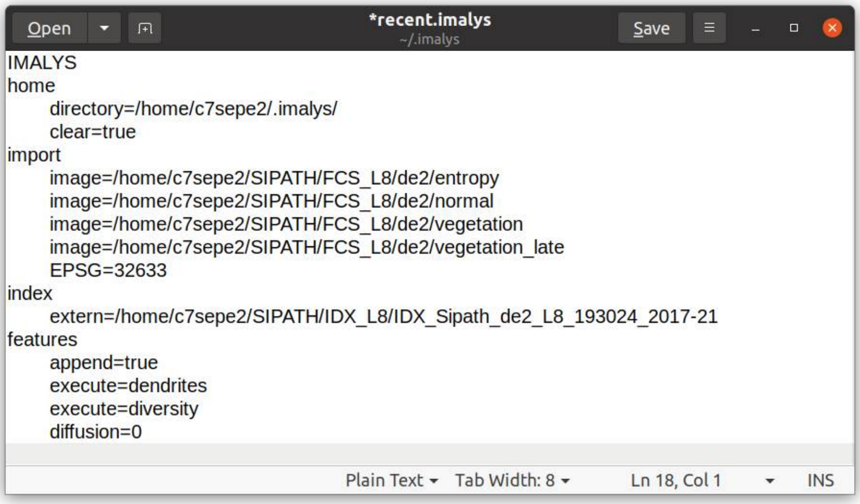
Features	*extend existing attributes with new properties*
features	Extends an existing attribute table with new values. If →Features is executed immediately after →Index, the definition of the zones is stored in the working directory. In all other cases, a directory with the zone definition must be specified with →Resume.

Zonal Features From Different Sources

[import] adds four images with kernel results as new attributes

[index] imports existing zones

[features] add attributes from the images below [import] and two for zonal attributes (dendrites, diversity)




```
IMALYS
home
	directory=/home/c7sepe2/.imalys/
	clear=true
import
	image=/home/c7sepe2/SIPATH/FCS_L8/de2/entropy
	image=/home/c7sepe2/SIPATH/FCS_L8/de2/normal
	image=/home/c7sepe2/SIPATH/FCS_L8/de2/vegetation
	image=/home/c7sepe2/SIPATH/FCS_L8/de2/vegetation_late
	EPSG=32633
index
	extern=/home/c7sepe2/SIPATH/IDX_L8/IDX_Sipath_de2_L8_193024_2017-21
features
	append=true
	execute=dendrites
	execute=diversity
	diffusion=0
```

→Index generates attributes from all spectral bands of the provided images without a specific command. These attributes are always available. If new attributes should be added from external image data, the images must be provided as →Import, not as “source”. Shape and size attributes (→Size, →Dendrites, ...) can be created at any time.

resume = directory	below →features selects “directory” as the source for the zone definition. →Resume assumes that “directory” was created with →Export.

Size:	*Size of single zones given as natural logarithm*	
	ln(s)	*s: size*

execute = size	below →Features creates an attribute with the absolute area of the zones. To overcome large size differences, the area is given as the natural logarithm. Other scales then require only an additive constant.

Dendrites:	*Quotient of zone perimeter and cell size*	
	p / s	*p: perimeter; s: size*

execute = dendrites	below →Features creates an attribute with the ratio of the circumference to area of the zones. Since the area of a zone grows faster than its circumference, the attribute returns a mixture of area and shape. It serves as a measure of spatial diversity. Small and narrow zones have the largest values, large compact zones the smallest.

Diversity	*Spectral diversity for all neighbor zones*	
	$\sqrt{\sum(v_i - v_n)^2 \cdot p_{in}}$	*v: spectral value; i,n: central, neighbor* *p: common border length*

execute = diversity	→Features generates an attribute with the spectral diversity of the zones.

Similar to pixels, →Diversity measures the spectral distance to all connected zones as the distance in the n-dimensional feature space. Since the borders to each zone differ in length →Diversity scales the spectral distance with the length of the border and uses the mean of all scaled distances as diversity.

→Diversity reduces the sum of all distances at the border of the zone in relation to the size of the zone. In this context the sum of all borders within the zone is taken as size of the zone.

The result is the average of all spectral distances between two pixels within and at the edge of a zone in the n-dimensional feature space.

→Diversity is similar to Rao's approach to diversity (→Entrope). Diversity is assessed according to its size and the spatial distribution of the zones.

Proportion: *Size difference between central zone and all neighbors*

$$\left(\sum_j \ln(s_j) - \ln(s_i)\right) / n$$

s_i: size of central zone; s_j: size of neighbor zone; n: number of neighbors;

execute = propotion — below →Features creates an attribute with the ratio of the sizes between the central zone and all neighbors. The result is negative if the central zone is smaller than the average of its neighbours.

The process determines the ratio of the areas between the central zone and all neighbors as the difference of the natural logarithms of their sizes. The result is the mean of all differences, the length of the boundary is not taken into account.

"Proportion" resembles a texture of zones sizes.

Relation: *Quotient of cell perimeter and number of neighbors*

$$p / n$$

p: perimeter; n: number of neighbors

execute = relation — below →Features creates an attribute with the ratio between perimeter and number of neighbor zones.

If all connected zones are of the same size, the result is independent of the size of the central zone. If this is not the case, →Relation indicates differences in the shape of the zones.

Similar to →Dendrites, →Relation integrates shape and size properties.

Append: *Attributes from all images listed as →import*

append = true — below →Features takes the spectral signature of all bands below →Import as new attributes.

If →Append is run independently of →Index, attributes can be added from image data that only need to cover the same area as the original image data.

→Append can use any type of image data, including maps or masks. Projection and pixel size must be adjusted if they differ from the original image.

Diffusion:	*Smoothe properties in a zonal network*
	$\sum_t a \cdot s \sum_j (a_j \cdot b - a_i \cdot b)$ *a: attribute; s: size; b: common border length; i,j: own, neighbor cell; t: time steps*

diffusion = number

[1…N] below →features controls the range of a value equalization. →Diffusion=0 has no effect. The parameter affects all new attributes.

The algorithm for value equalization in zones mimics diffusion through membranes (borders). In the process, features “migrate” into the neighboring zone like soluble substances and mix with the existing concentrations. The intensity of diffusion depends on the length of the common border and the number of iterations. The size of the zones does not matter.

The value equalization is controlled only by the “number” of iterations. Each iteration includes a new layer of contributing zones. The influence of distant zones on the central zone diminishes with distance. Entries over 10 are not forbidden, but rarely have a visible effect.

Raster:	*Control image with one band for each attribute*

raster = true

below →features creates an image with one band for each attribute.

→Raster does not create features but maps them. The process translates all stored attributes into bands with the value of the attribute as density. This allows to visualize attributes and their combination.

Correlation

It is not easy to determine temporal, spatial or combined spatio-temporal linking of images and other measurements. ESIS offers the process →Rank, which allows to compare any measurement series by means of a rank correlation. For this purpose, the time of the data acquisition or the location must be known in both datasets. At best, of course, both.

→Rank works with each distribution and can therefore analyze any pattern. As an alternative, the degree of correlation can be estimated according to X^2 →ChiSquare. Strictly speaking, X^2 presupposes standard distribution, but may then be more selective.

Rank	*Pattern recognition*
ChiSquare	*Correlation*

Classification

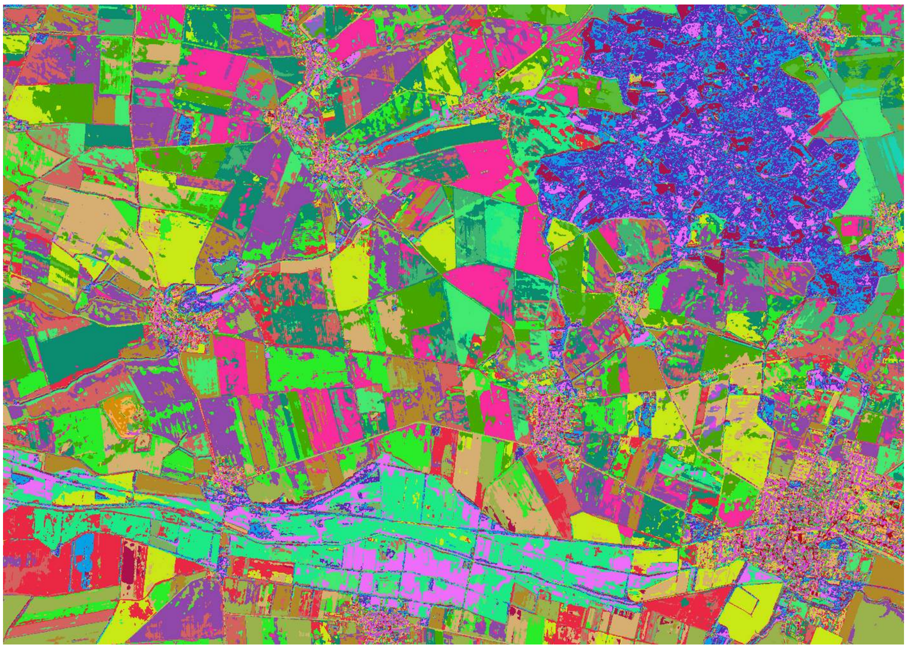
Imalys implements a fully automatic classification of image features →Mapping. The process can use pixels or zones and in an advanced version zones can be combined to "objects".

Self-Adjusting Classification

30 spectral combinations (patterns) separated by a fully automatic process according to Teuvo Kohonen

The colors shown are random.

Sensor: Sentinel-2
Years: 2017-2021
Channels: 2, 3, 4, 8
Seasons: May – July
and August – October
Process: Mapping
Color code: Random
Values: Classes 1...30



Deriving basic land use types from image data can be unreliable because land use types are defined by their purpose and not by their appearance in the image. Machine learning can recognize almost any pattern, but needs to be trained with examples. Only trained patterns are recognized. The training may take longer than a manual classification.

Levels

Imalys implements methods to arrange image features of all kinds into separate groups or clusters (→Mapping). The result reflects feature combinations that are common in the classified image. →Mapping can classify images at three levels:

(1) Pixels based on their spectral combinations
(2) Zones based using zonal attribures
(3) Objects based on connections of classified zones

The principle is the same in each case. Features or properties create a multi dimensional feature space. Local concentrations of feature combinations are detected and classified using a neuronal network according the suggestion of Teuvo Kohonen (ZZZ)

Pixels

Unsupervised classification of pixels based on spectral combination is a standard procedure (→Pixel). The result depends mainly on the selection and quality of the image base (see chap. Time Series).

Zones

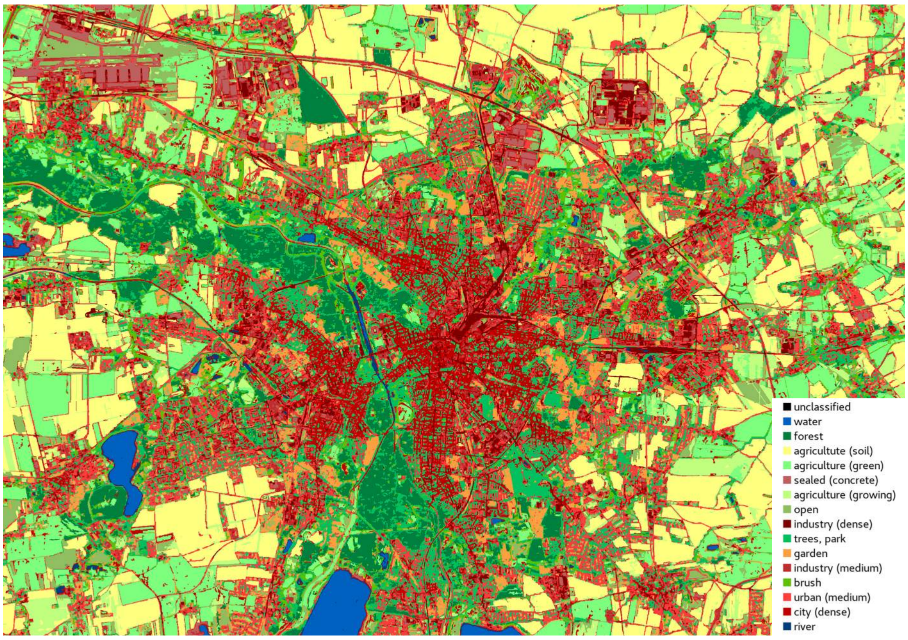
Zones must be created (→Index) in front of the classification (→Zonal) but zonal properties provide extended features derived from size, shape and connections of the zones. The spectral features are the mean of all pixels that contribute to one zone. Zones and classes complement each other. Zones summarize typical pixel features at a limited area. Classification sorts them into a manageable list of feature combinations. Typical problems with clustering pixels such as "pepper and salt" patterns do not occur anymore.

Image Objects I

Spatial patterns of different zones were reduced by self-calibrating process to 30 patterns, which were assigned to 16 classes.

Process and parameters are identical to Fig (II)

Image data: Sentinel-2
Date: 16.9.2018
Bands: 8-4-3-2
Process: →Fabric
Values: Classes (see legend)



Objects

Classified zones can be combined by a second level classification to form "objects" (→Fabric). Objects are defined only by the frequency or intensity of the connections between their zones. The connection intensities form characteristic patterns that can be classified. The result describes typical pattern of different spectral and spatial combinations. These classes are hereinafter referred to as "objects". For details please see chap. "Methods".

Object Size

The size of the objects is not limited. Simple patterns of small zones can be repeated over a large area to form one large object. Large zones usually form objects of one dominant zonal class and many smaller ligands. In practice the size of the zones should be selected in such a way that homogeneous objects do not decay into many zones and heterogeneous objects can be represented by a sufficient number of small zones.

Seasons

Spectral differentiation of natural areas becomes more reliable when images from different seasons are used together. Different seasons may originate from different years. A logical combination of very different parameters, including color, shape, distribution and development of surface characteristics, may provide a robust alternative to machine learning without extensive training (ZZZ Ellen).

Examples

The purely statistical object classification without any training (→Fabric) was sufficient to separate roof surfaces, streets, parks, small gardens and other elements from two images of the city of Leipzig with very different scales and sources. Fig. I shows the result for an satellite image, Fig. II the result for an areal image combined with an elevation model. In both cases the parametrization was identical.

Image Objects II

Spatial patterns from different zones were reduced by a self-calibrating analysis to 30 patterns, which were assigned to 13 different classes.

Process and parameters are identical to Fig (I)

Image data: Infrared aerial photographs and elevation model of the State of Saxony.

Transfer

Due to its statistical technique, the method only generates definitions for objects that are common in the image. Other landscapes or different light invalidate the definitions. The technology has the advantage that it does not require any training or prior information. The result is determined only by the image data. It has the disadvantage that the results can only be transferred to images with the same structure. The technical implementation is described in detail in chap. “Methods”.

Mapping	*Map image features to a set of feature combinations*

mapping

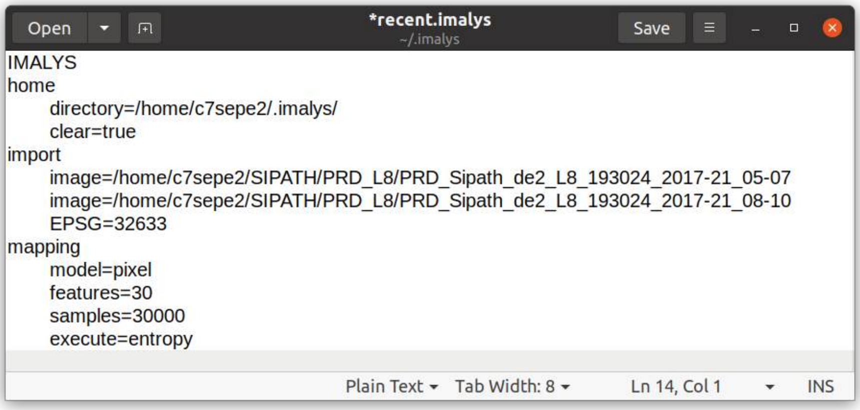
creates an automatic classification (clustering) of the provided image data. The result and the class definition can be saved in a separate directory with →Export.

Self-Adjusting Classes

[import] provides two calibrated images

[mapping] classifies single pixels of the import to 30 classes using spectral combinations

[execute=entropy] calculates Rao's diversity based on classes



All →Mapping processes require image data stored with →Import (ENVI format). →Mapping includes three different processes (→Pixel, →Zonal and →Fabric)

features = number

[1…N] below →mapping controls the "number" of classes. The number of classes should not be too large. Overclassification reduces accuracy. Two classes per desired land use feature have proven their worth.

Coincidence can confuse statistical analysis. It happens that one class more or less (!) significantly improves the quality.

sample = number

[1…N] below →mapping controls the "number" of samples for the class definition.

To find clusters in the feature space →Mapping uses samples from the image data. They are selected from the picture at random places and reduced in such a way that their spatial distances are largely the same. Samples make the classification much faster than each pixel has to be evaluated individually. Around 1000 samples per desired class have proved their worth.

equalize = true

below →mapping normalizes all values or attributes to the value range [0...1]. "equalize" should only be used if images or attributes with very different values have to be processed.

external = directoty

below →mapping takes over a class definition stored in "directory" and applies it to the →Import. The classification type (→Pixel, →Zonal, →Fabric) must match the image data.

Pixel	*Classify spectral combinations*

model = pixel

below →mapping selects a pixel-oriented classification of the image data.

→Pixel uses all bands of the provided image. The bands will mostly be spectral bands. If other bands are involved, the value range of the bands should be considered. Very small values would not affect the classes, since their distances in the feature space are insignificant. Very large values will dominate the derived classes.

Zonal	*Classify spectral and spatial properties*
model = zonal	below →mapping selects the classification of zones based on their attributes (→Features). The zones and their attributes must exist. The process takes into account all attributes, including those that were subsequently created with →Features. As with →Pixels, the value range of the attributes should be comparable. On the other hand, attributes with particularly high values can be used on purpose to dominate the classes.

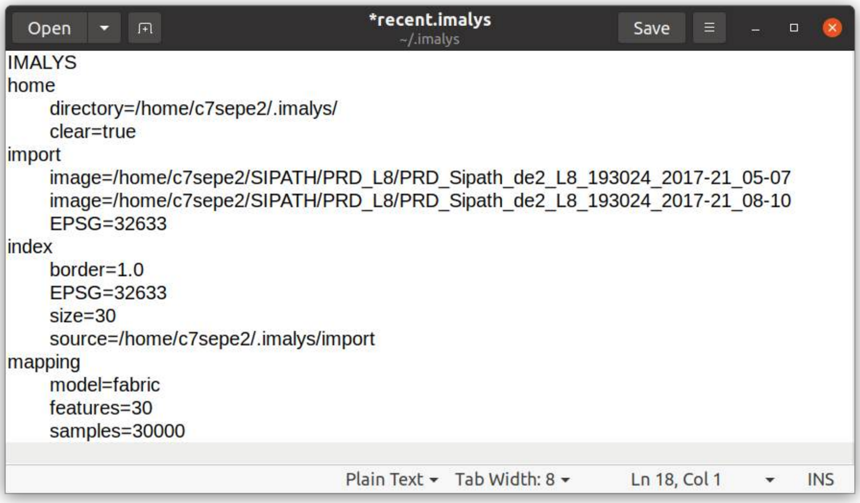
Fabric	*Classify spatial patterns of different zones*
model = fabric	below →mapping classifies the linkage between different zones and forms object classes. The zones and their attributes must exist.

Classify Image Objects

[import] provides two calibrated images

[index] delineates zones using the import

[mapping] classifies the zones by means of spectral combinations and combines different zones to objects by means of their spatial pattern [model=fabric].




```
IMALYS
home
	directory=/home/c7sepe2/.imalys/
	clear=true
import
	image=/home/c7sepe2/SIPATH/PRD_L8/PRD_Sipath_de2_L8_193024_2017-21_05-07
	image=/home/c7sepe2/SIPATH/PRD_L8/PRD_Sipath_de2_L8_193024_2017-21_08-10
	EPSG=32633
index
	border=1.0
	EPSG=32633
	size=30
	source=/home/c7sepe2/.imalys/import
mapping
	model=fabric
	features=30
	samples=30000
```

Real landscape types with a specific use are rarely completely homogeneous but consist of different parts, such as a plantation might consist of trees, green spaces and roads. Object classes use the specific composition of the different zones to describe the class.

As for zonal classes, the number of object classes should not be too large. Random differences in the appearance of objects can quickly become larger than the difference between two object classes.

double = true	below →mapping expands the search for suitable zones for the object formation in space

Export

Export allows to save the results at a selected place. Imalys stores the results of all processes in the working directory. From there they can be transferred to another location. The user can choose from a variety of formats.

The →Export command is context-dependent. It basically takes over the result of the last given command and saves it in the selected format. The export format is controlled by the extension in the result name. →Export can be repeated after each command to save new results.

Special conditions apply to zones, classes and vectors.

In the working directory, Imalys stores all raster data in the ENVI format [Chapter XYZ]. Vector data and tables are stored in WKT format. Both formats support fast and easy processing.

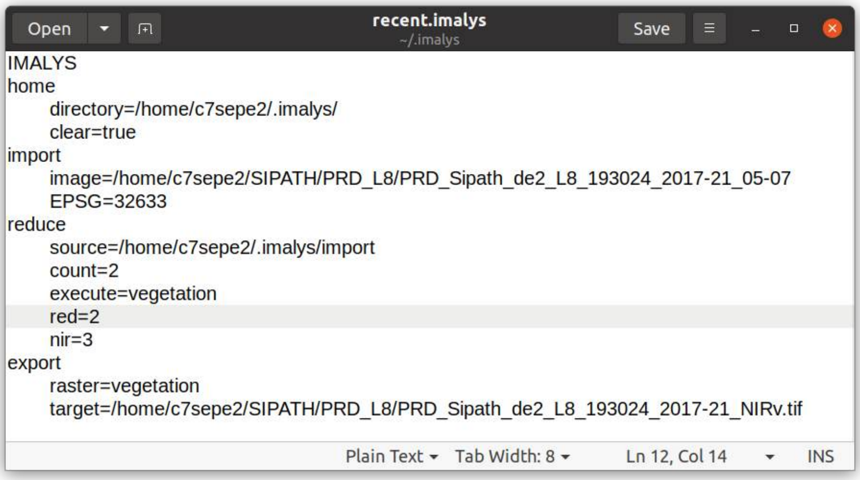
Export	*Transform and store the most recent processing result*
export	initializes the data export

Export GEO TIFF

[import] provides a calibrated image

[reduce] calculates the vegetation index NIRv

[export] saves the result as GEO-Tiff at a selected place



format = index

below →Export stores the most recent zones definition and creates a vector copy. →Export copies the complete zone definition to a separate directory with the name given by “target” but without the extension. The definition includes a raster file (index, index.hdr), the attributes as a binary table (index.bit), a table with links between the zones (topology.bit), and the vertices of the polygons (vector.csv). The five files should not be changed or separated.

At the same time “index” creates a copy of the zonal geometry and attributes using a vector format. The name given by “target” is used. The extension of “target” selects the vector format.

format = mapping	below →Export stores the result of a classification as a raster file along with the class definition (mapping.bit) to a separate directory. In principle, the class definition can be used for any other image that share the same bands. However, fully automatic classes will only provide satisfactory results for very similarly structured images.
format = raster	below →Export exports the last created image using the name given with “target.” The result format is controlled by the extension of the name. No extension generates a result in ENVI format (default).
format = vector	below →Export stores the last created vectors using the name given with “target”. The format is controlled by the extension of the name. No extension chooses the ESRI shape format (default).
EPSG = number	[1…N] below →Export selects “number” as the projection for the result. “number” must be a valid EPSG code (option).
target = name	below →Export selects “name” as pathname for the result. The extension of “name” controls the format. Without extension, →Export uses the ENVI or the ESRI shape format.

Methods

Zoning

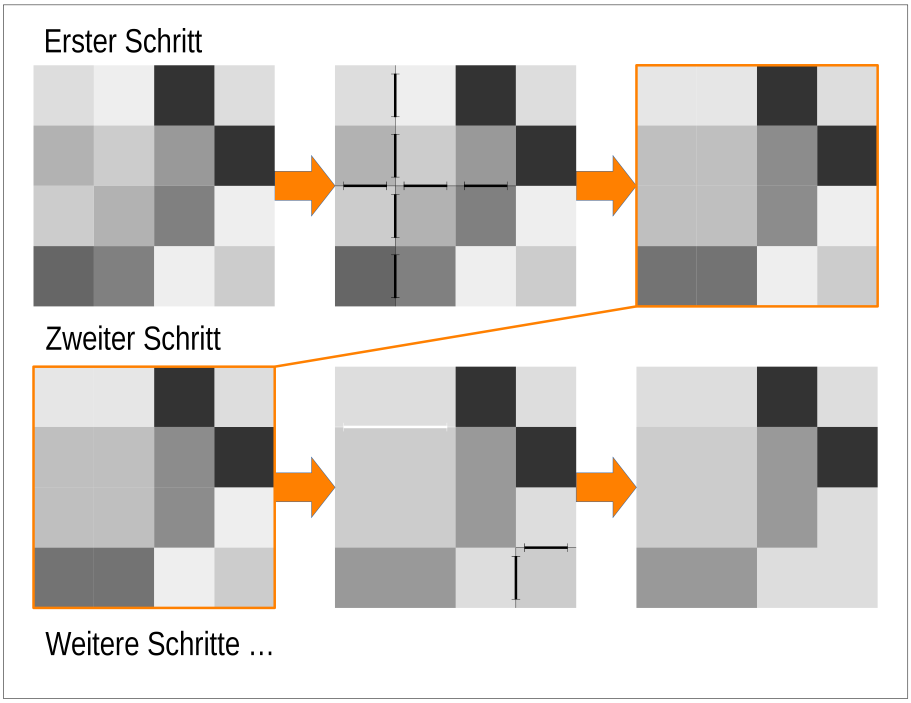
Imalys creates “zones” with an iterated watershed process. The first step merges the most similar pixel neighbors by linking them with a common ID. A threshold can suppress the merge. The threshold can be influenced by the user.

Create New Zones

Shown are strongly enlarged pixels and their grayscales

Above: Maximum similarity between seven pixel pairs ⇒ 7 borders (black lines) are deleted

Bottom: Maximum similarity in 3 cases ⇒ two borders (black lines) are deleted, one border (white line) between two larger areas is retained



As “threshold”, Imalys uses the first principal component of all normalized brightness modulations (a-b) / (a+b) with a,b as the brightness values. Normalization promotes differentiation in dark areas of the image. The principal component promotes the effect of contrasts that occur only in one band.

After the first iteration, which combines only individual pixels into first zones, the process combines existing zones into larger ones. Again, only the locally most similar zones are merged and again the above threshold must be adhered to. In addition, the increasing size of the zones acts as an inhibiting factor when merging zones. The zones grow in the course of repeated steps until the process no longer finds suitable precursors.

→Index uses an exponential function of the cellular size to inhibit the merge. The exponent can be selected. Small exponents (close to zero) reduce the influence of the size on a possible merge. This creates zones that are mainly controlled by contrast. They are largely homogeneous in spectral terms and can become very large. Edges are better recognized. Large exponents (close to one) lead to zones with smaller size differences. The spectral differences within the zones are larger.

The combination of brightness and area is necessary to create zones with reasonable areas. A threshold for only one parameter would separate the image into individual pixels and “infinitely” large zones.

Mapping

Kohonen

The basis of the mapping is the Euclidean distance in the n-dimensional feature space, similar to the IsoClass method. Each neuron represents a separate class. According to Kohonen's suggestion (ZZZ), the neurons have individual properties, not only connections as neurons of the perceptron type would have. One of the individual properties is a receptive field, a section of the feature space in which the neurons can recognize features. For the rest, they're blind. This property enhances their ability to depict small but common differences.

Model

Regarding zones two abstractions take place. One are the spectral features. Zones have an spectral composition like pixels but the value is the mean of all pixels connected to one zone. Local differences and texture might be lost. On the other hand the class definition gains information from size, shape and connection of the zones. A specific color combination in a small zone might gain a different meaning than in a large one.

Objects

Objects (→Fabric) are an extreme abstraction of this enhancement. The class definition depends only on the frequency of borders between different zonal classes. The zonal classes are only defined by their spectral composition. The whole form and size stuff is done by the connections between the zones. The border length is counted as pixel borders. This includes all pixels, also pixel borders within one zone. This principle introduces the necessary plasticity in the object definition.

Objects can consist of simple patterns that are repeated over a large area depicted by small zones. Objects can also consist of one large but homogeneous zone. In the first case, the object is defined mainly by the connections between zones, in the second internal borders of the large area dominate the definition and the connections play a minor role (see chap. Object Generation). In practice there is a smooth transition between both extremes.

Since the object class definition depends only on different frequencies, each zonal class may occur in each object class. Non-specific zonal classes such as “shadows” can be defined in all object classes. They are not characteristic for the object, but make it complete.

Object Generation

→Fabric combines two processes in three steps. Pixels are combined to zones, zones are classified by means of their spectral features and finally

classified zones are combined to objects with specific spatial combinations of different zones.

Image Objects

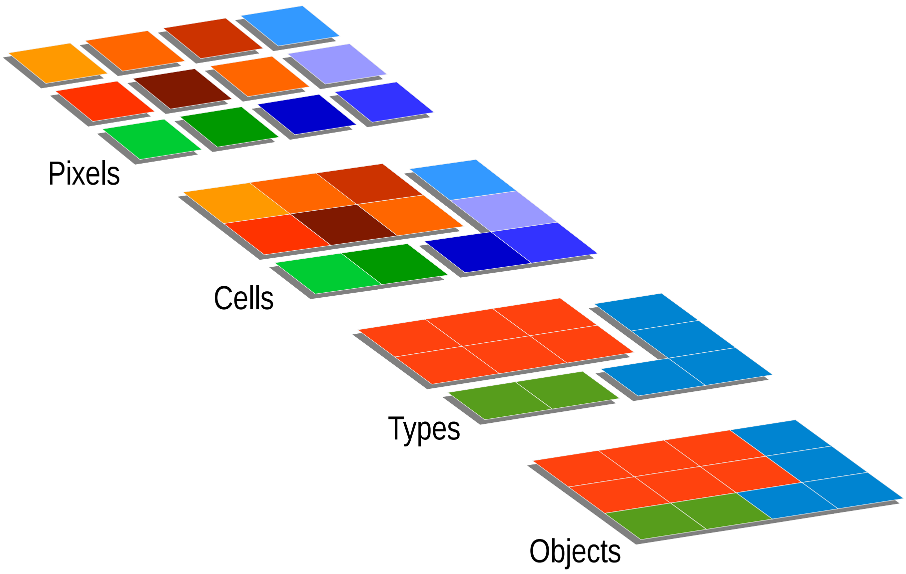
Image objects are created in three steps:

(1) Pixels of similar spectral combination are combined into zones

(2) Zones are classified into clusters with typical spectral combination (types)

(3) Types are aggregated to form objects according to their connections to other zones

Each zonal type can be part of each object definition



Spatial features

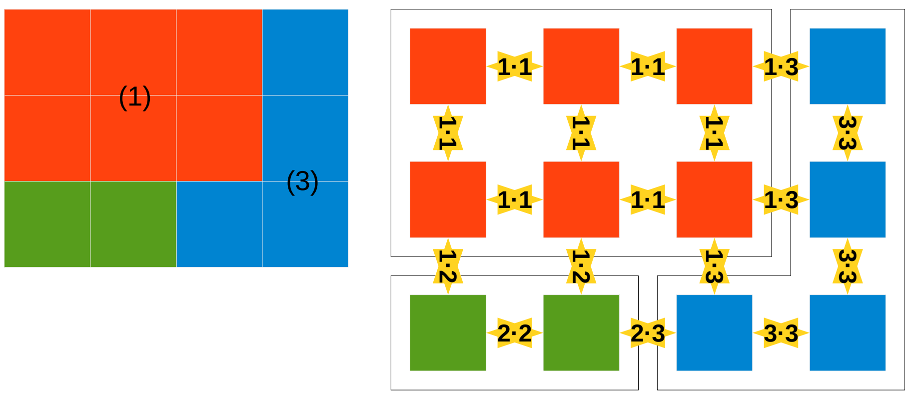
Images of the earth's surface are structured. Pixels with nearly the same spectral combination are not randomly distributed but form clusters and sometimes regular patterns. The →Index process combines regions with pixels of nearly the same spectral combination into "zones." Zones have spectral features like pixels.

Spectral features

During the second step, the →Zonal process classifies the characteristics of individual zones and assigns zonal classes. In this case the classification depends only on the spectral characteristics of the zones and the normalized texture (→Normal) between zonal pixels. After the second step zones map the spatial distribution of image features and the zonal classes map the spectral distribution. →Fabric combines both to a class definition that includes size, shape and typical patterns of visible structures in the image.

Object Definition

An object consisting of three zones (red-green-blue) with a total of 12 pixels is characterized by 17 contacts (yellow symbols). The object definition is based solely on the nature and frequency of these contacts.



Combination

→Fabric registers all class combinations between two pixels for each zone. "All" includes also pixels within the zone. The frequency of the class combi-

nations at all borders between two pixels form a matrix that can be used for a second level classification. →Fabric arranges them almost like spectral classes (→Mapping).

Examples

One typical result are objects that are dominated by one large zone whereas the connected zones are small and unspecific. Water bodies or agricultural ares need a class definition of this kind. The other extreme are small zones that only have contact to a few other classes. The class definition is dominated by their connections. They reproduce small-scale patterns that can be repeated over a large area. The definition will combine zones with different spectral compositions but similar connection thus defining regular patterns. Object classes of this type are found in settlements or forests.

Since the object definition uses only frequencies there is a smooth transition between small-scale patterns and large zones with unspecific connections.

Control

Imalys was designed as a library containing executable programs and their source code. The program “xImalys” is controlled by a command line. The most important parameter is the filename for a script (text) that can contain a long chain of processes and their parameters.

Working directory

A process chain (Script) is bound to a working directory. It is created or emptied at the beginning of the command chain and stores all intermediate results. Imalys can run in any number of instances if each instance is linked to its own working directory.

In some cases, it may be useful to collect intermediate results from more than one process chain in the working directory and export only the final result. For this reason, the “clear” parameter for the working directory can be disabled (→Home).

Import, Export

The →Import transforms and checks the images. All other processes assume that they include verified data. If intermediate results are to be stored externally, the ENVI format must be used (→Export). Classes and zones use their own export processes that store the results as an image or vector in a separate directory. For classes, this is mainly the class definition, for zones, attributes, links, coordinates and zone IDs are saved.

Data format

With the exception of →Export, all commands are free to select their input data and save their result under a fixed name in the working directory. The name is identical to the command. However, some processes produce more than one result. The format for raster data is always RAW binary with header (ENVI format) and WKT (CSV with format line) for vectors and tables. Zones and classes also use a binary table format (.bit) which allows very fast access. The formats support easy processing. All metadata must include a valid projection (CRS), image data require the date of the data acquisition.

NoData, values

Imalys commands assume that thematic data (classes, maps) are stored with natural numbers (integer) whereas scalar data (images, elevations, ...) are stored with real numbers (float). For natural numbers, zero indicates undefined areas of the image, for scalar data, the value (1/0) = (NoData) is used.

Commands

Scripts consist of commands and parameters. Commands have their own line. Parameters must follow their command. Parameters consist of an identifier and a value separated by a “=” character. The “=” character defines a line as a parameter line. Commands can be used in any order and repeated as often as desired. Only the import at the beginning of the chain is mandatory. Parameters always have a preset. It is changed by the entry in the script.

Control

Imalys reports progress and errors via standard-IO. When “xImalys” is called in a shell, the output can be observed directly. Imalys stores a progress re-

	port to a file "process.log" at the working directory. The command chain is stored twice. In the working directory as "recent.imalys" and together with the result and named like the result.

xImalys script	*runs a "xImalys" command chain*
xImalys	calls the program
script	passes the path name of a text file containing processes and parameters. The Processes are executed in the given order.

Home	*Initialize a new process chain*
directory = path	below →Home sets "path" as the working directory and creates the directory if necessary. This line is mandatory!
clear = true	below →Home clears the working directory of all entries (option)

Changes

0: 2022-05-01: Draft
1: 2022-07-10: Methods, Formulas
2: 2022-08-13: Reformation by theme
3: 2022-08-31: Breakdown by Traits – Application – Method
4: 2022-09-08: Unification of terms
5: 2022-09-14: Graphics revised,
6: 2022-10-02: Tutorial in blocks
7: 2022-10-05: Chapter “control”
8: 2022-10-06: Chapter “Installation”