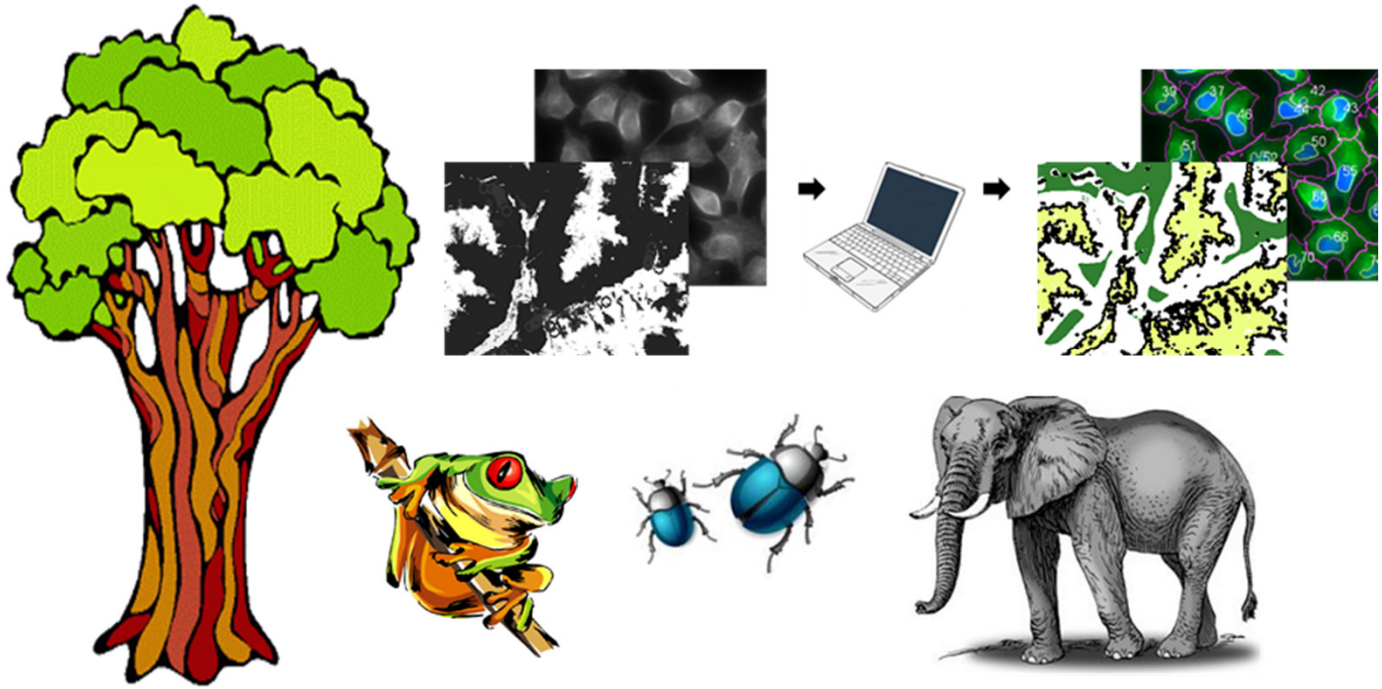
Imperial College London

Image Processing



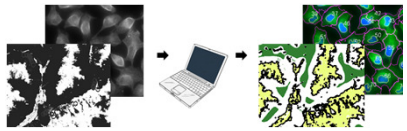
in Ecology

CMEE

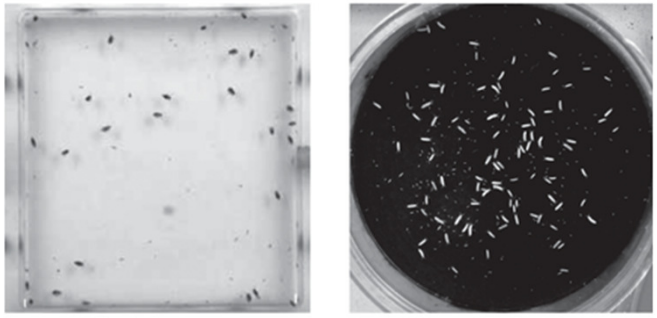
Véronique Lefebvre

v.lefebvre@imperial.ac.uk

Image processing in Ecology



Analysis of photos and videos from experiments:



Organism counting

Species classification

Analysis of landscapes from spatial data:

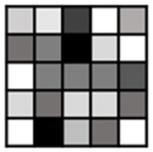
Aerial photos

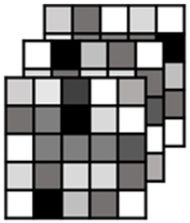
Satellite images

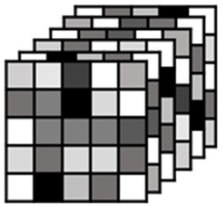
Types of digital images

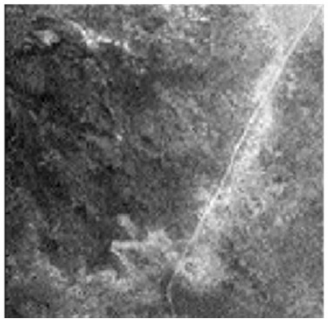
Digital images can be manipulated like N-dimensional arrays

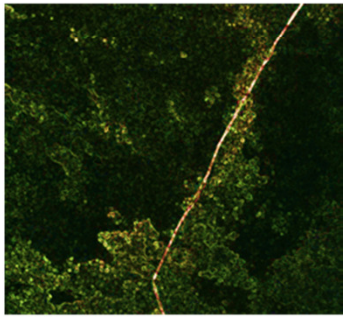
a00	a01	a02
a10	a11	a12
a20	a21	a22











Binary image
e.g. habitat / non habitat map
1 layer, 2 values

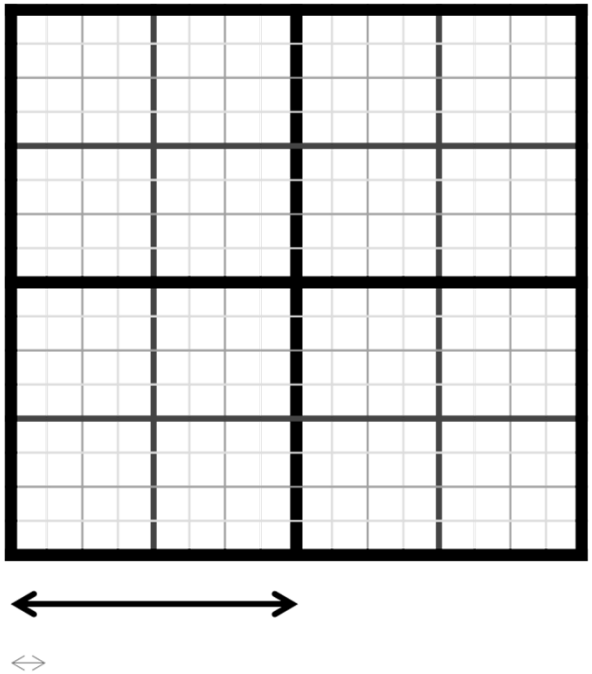
Grayscale image
e.g. NDVI
1 layer, range of values

Colour image
e.g. colour photo
3 layers, range of values

Multispectral image
e.g. satellite
N layers (~ 5 , 9), range of values

Digital image properties

- Number of levels (quantization)
- Spatial resolution: number of independent pixel values per unit length



4 resolutions

Remote sensing:

NDVI (normalized difference vegetation index), 30 m per pixel (Landsat)

Visible light, 10 m per pixel (Spot)

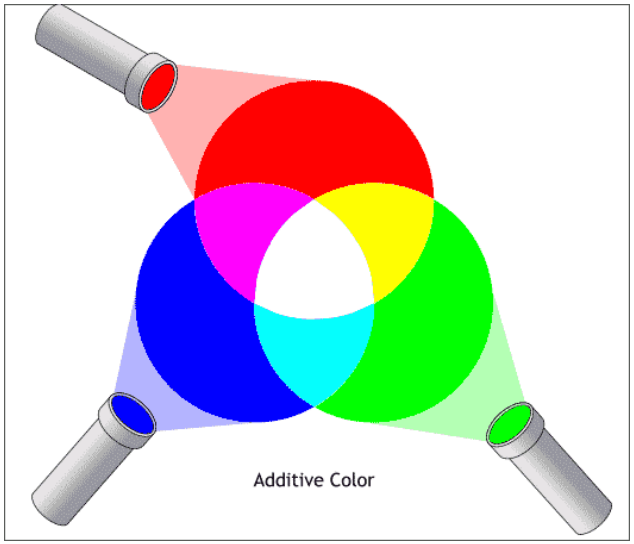
Digital image properties

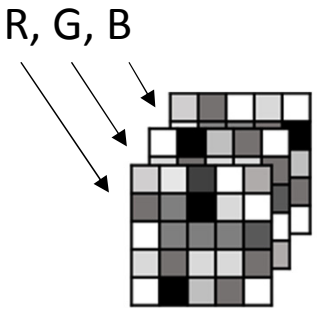
Colour is encoded by a triplet at each pixel locations:

Layer 1: red

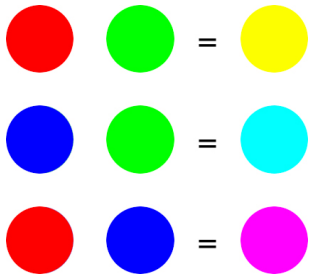
Layer 2: green

Layer 3: blue





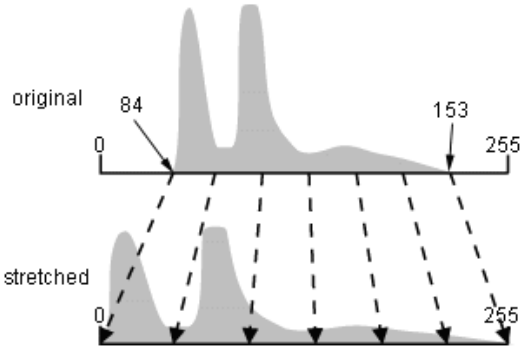
R	G	B	
0	0	0	
0.5	0.5	0.5	
1	1	1	
1	0	0	
0	1	0	
0	0	1	
1	1	0	
1	0	0	
0	1	1	
0	0.5	0.1	



Additive colour mixing: as more colour is added the result gets lighter and tends to white.

Image enhancement - Intensity

- Contrast stretching: remap pixel values to the entire intensity range



Histogram range adjustments:

Point / pixel operations

Image enhancement – Colour

- Removing colour cast: neutral / white balance
- E.g. with assumptions that shadows are grey
- By adjusting the 3 layers separately (curves tool)
- Important for e.g. comparing aerial photographs taken in different weathers

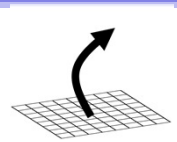
Blue cast

Green cast

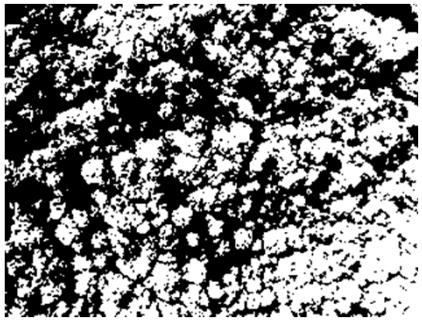
Point / pixel operations

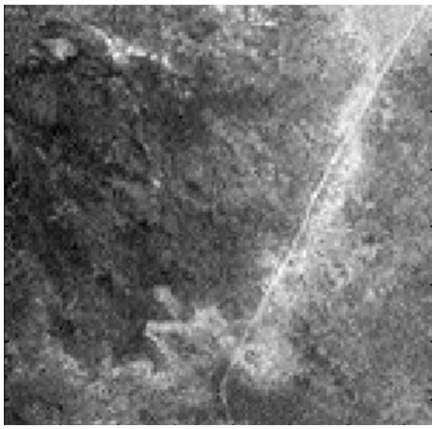
http://gimp-savvy.com/BOOK/index.html?node61.html

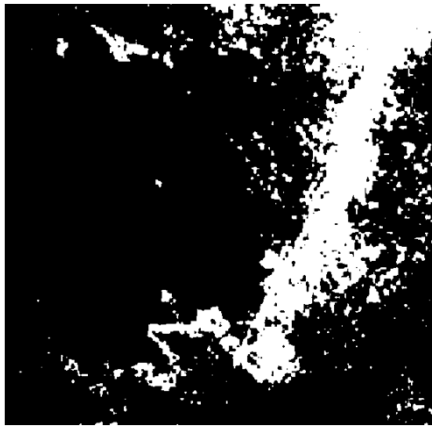
Feature Extraction - Threshold

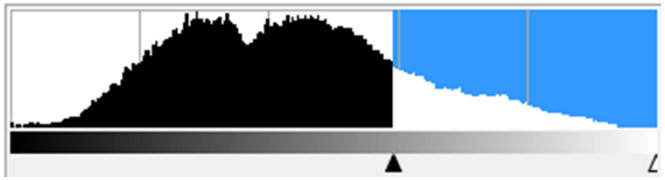


- Extract regions with higher or lower intensity than a threshold to obtain a mask
- E.g. to get a habitat / non habitat map, or delineate objects
- However, impractical as a first step due to image noise





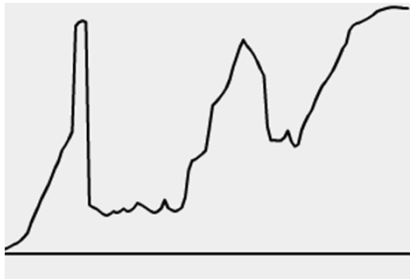


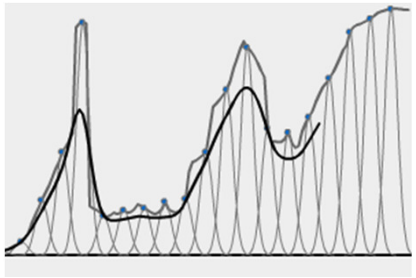


Point / pixel operations

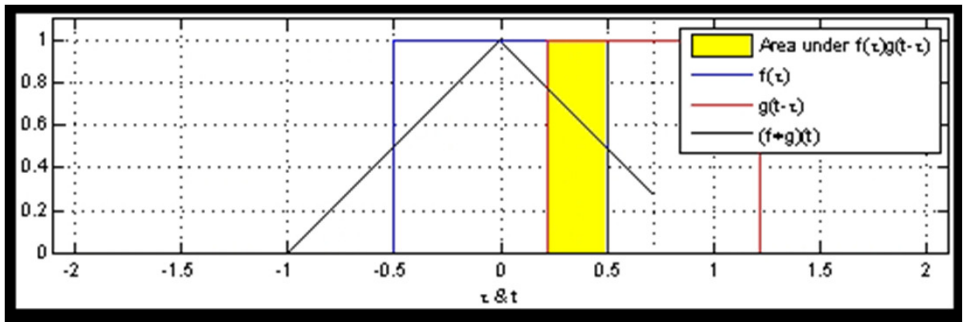
Image enhancement – filtering

- Smoothing
- Similarly to smoothing a 1D signal to obtain local trends, we can smooth an image
- With convolution





Average filter

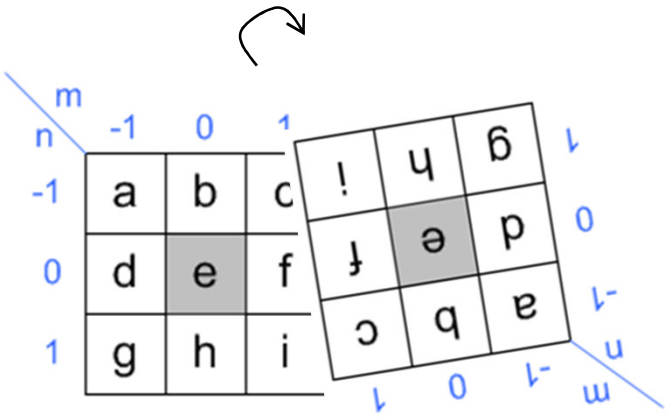


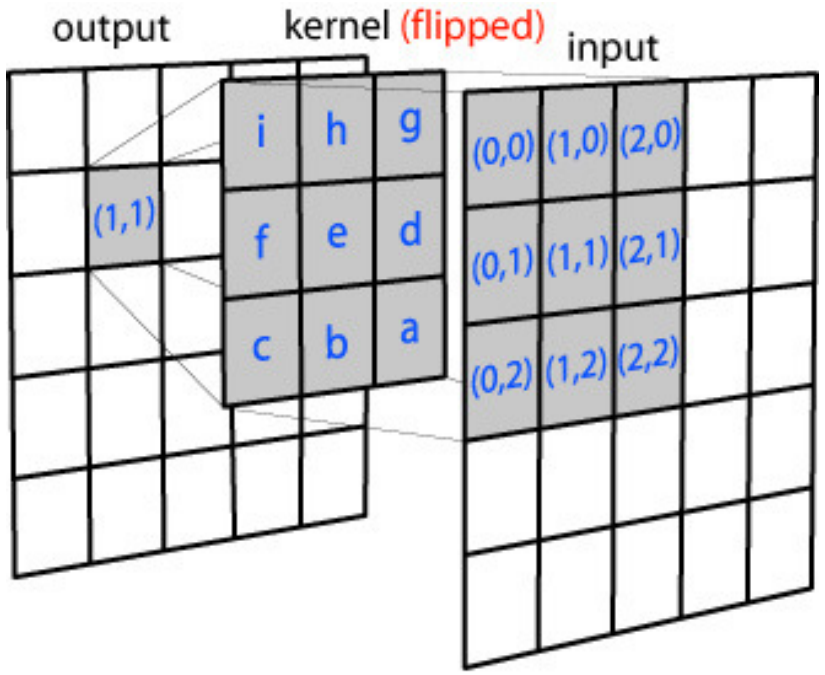
The convolution of two vectors, u and v, represents the area of overlap under the points as v slides across u

For vector of same lengths, the nieme element:
w(n) = u(1)*v(n)+u(2)*v(n-1)+ ... +u(n)*v(1)

Image enhancement – filtering

- 2D convolution
- notice that the filter matrix is flipped both horizontal and vertical direction before multiplying the overlapped input data



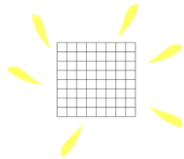


$$y[m,n] = x[m,n] * h[m,n] = \sum_{j=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} x[i,j] \cdot h[m-i, n-j]$$

$$y[1,1] = \sum_{j=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} x[i,j] \cdot h[1-i, 1-j]$$

$$\begin{aligned} = \; & x[0,0] \cdot h[1,1] + x[1,0] \cdot h[0,1] + x[2,0] \cdot h[-1,1] \\ & + x[0,1] \cdot h[1,0] + x[1,1] \cdot h[0,0] + x[2,1] \cdot h[-1,0] \\ & + x[0,2] \cdot h[1,-1] + x[1,2] \cdot h[0,-1] + x[2,2] \cdot h[-1,-1] \end{aligned}$$

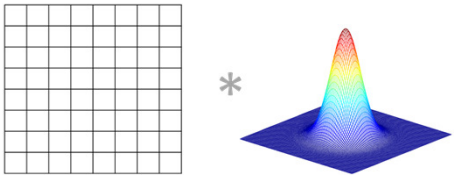
Image enhancement – filtering



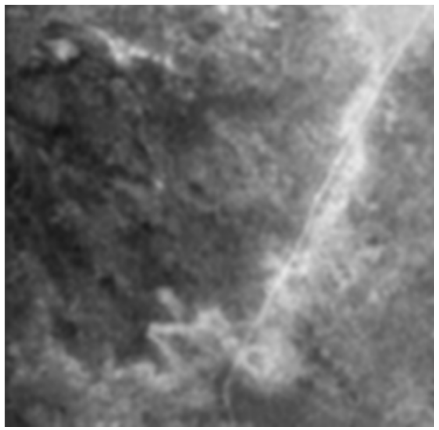
- Smoothing – Average filter

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

- Gaussian filter

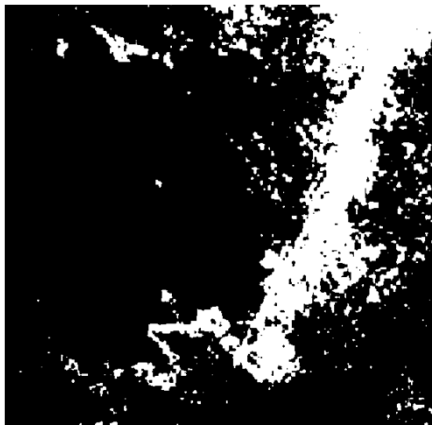


- Attenuates local variations



Gaussian blur

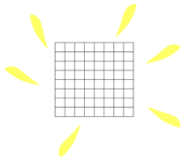
Threshold



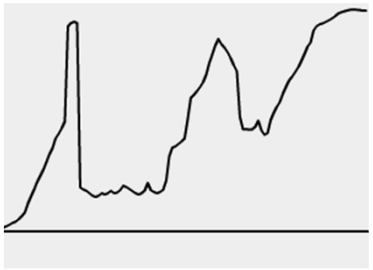
Smooth and threshold

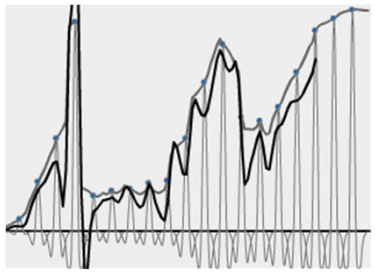
Neighbourhood operations

Image enhancement – filtering



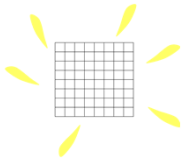
- Sharpen (Correct out of focus)
- Amplifies local variations



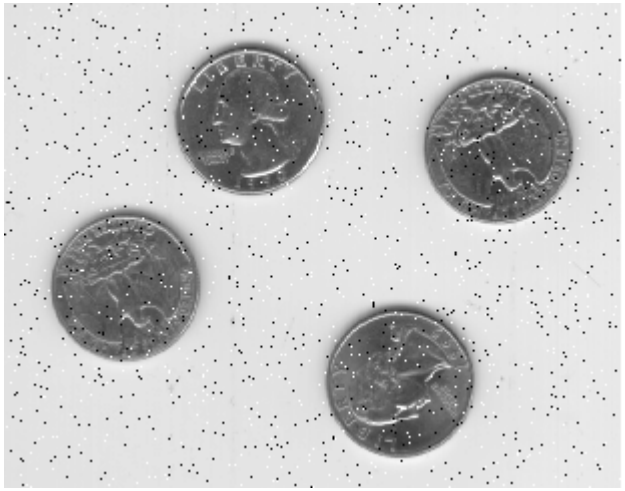


Neighbourhood operations

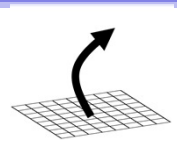
Image enhancement – filtering



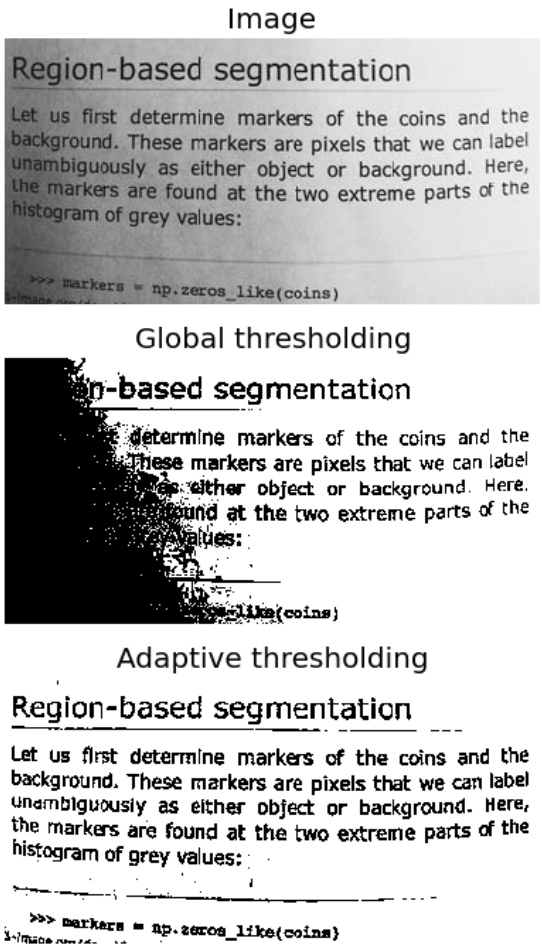
- Noise reduction: median filter / despeckle



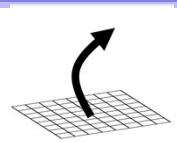
Feature extraction – Adaptive threshold



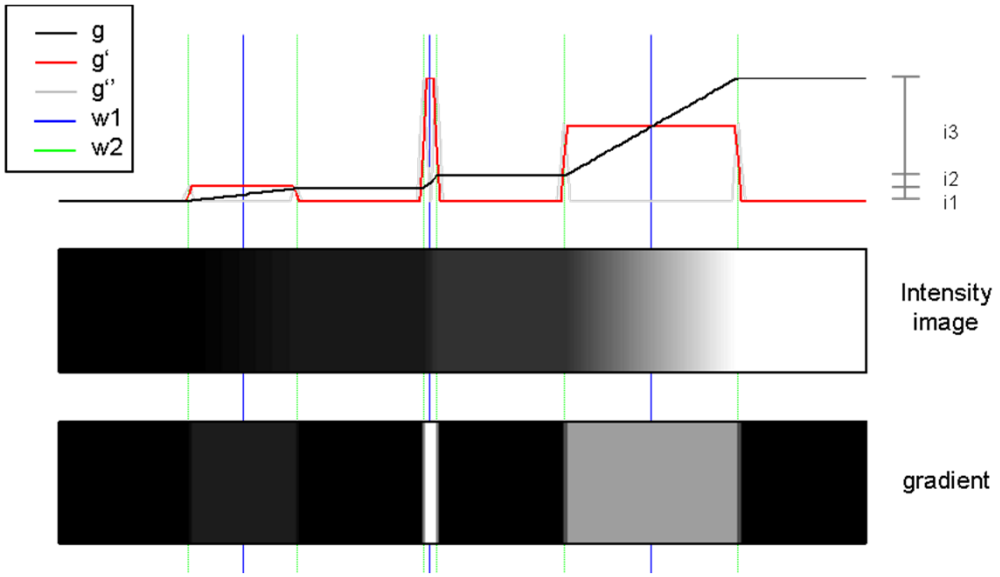
- Using local filter to optimise threshold value
- http://scikit-image.org/docs/dev/auto_examples/plot_threshold_adaptive.html#example-plot-threshold-adaptive-py



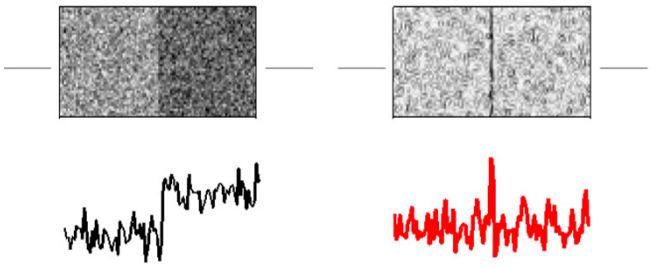
Feature extraction – Filtering – Edges



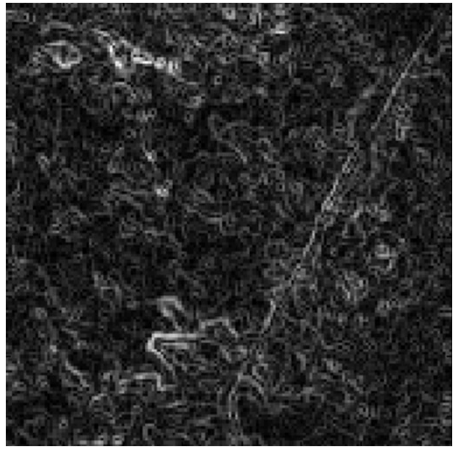
- Edge filter
- As in 1D, we differentiate to extract local intensity variations
- Gradient image



- However taking the gradient directly is often impractical due to noise

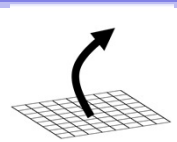


Neighbourhood operations



gradient

Feature extraction – Filtering – Edges



- Sobel filters:

1	0	-1
2	0	-2
1	0	-1

1	2	1
0	0	0
-1	-2	-1

- Horizontal and vertical

- Convolving with a 3 x 3 matrix -> summing the difference of the outermost columns (vertical filter)

a00	a01	a02
a10	a11	a12
a20	a21	a22

$*$

1	0	-1
2	0	-2
1	0	-1

$=$

-a00 + 0 + a02 + …

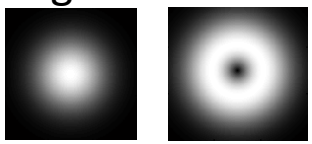
-2a10 + 0 + 2a12 + …

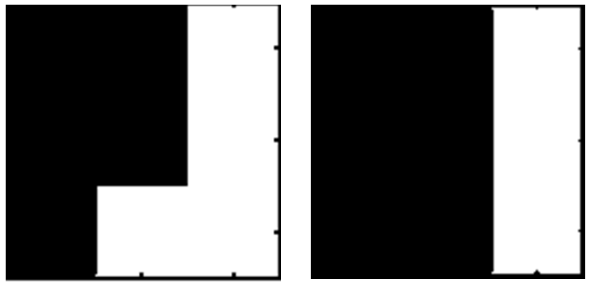
-a20 + 0 + a22

- Then horizontal filter (rows)

- Then square and sum to obtain gradient

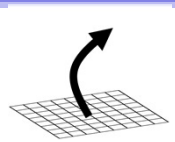
Original Gradient



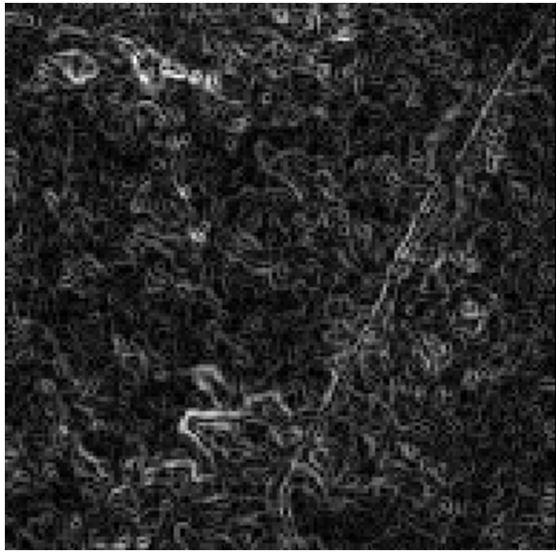


- A corner (2D) has a higher gradient than a line (1D)

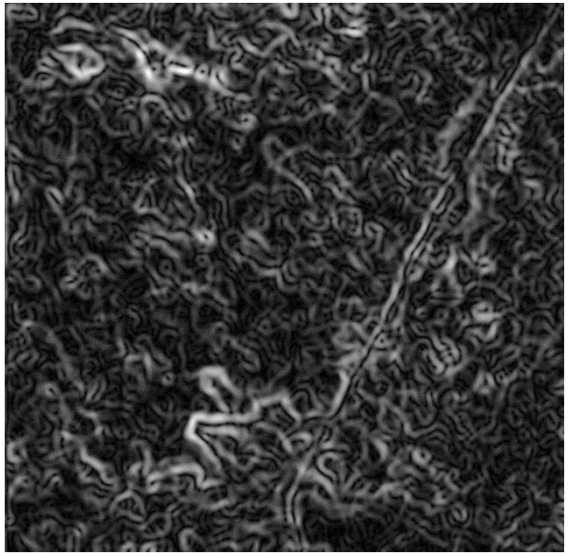
Feature extraction – Filtering – Edges



Gradient

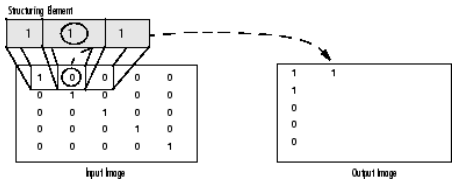


Gradient after smoothing

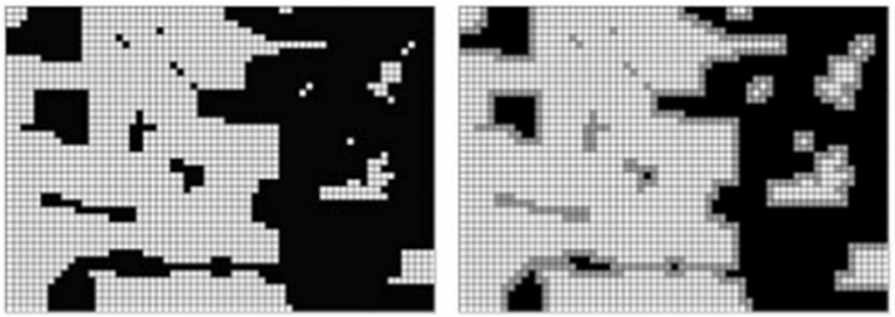


Neighbourhood operations

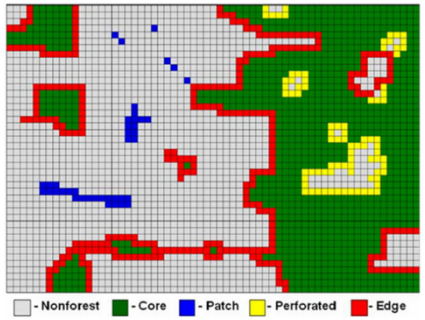
Image enhancement - Morphology



- Dilation / erosion
- structural elements
- Opening: erosion followed by dilation
  - Opening by reconstruction: erosion followed by morphological reconstruction
- Closing by reconstruction: dilation and reconstruction
- To “clean” the image: Remove background and foreground noise in binary images, create flat maxima in grayscale images



Erosion
(eroded pixels
in gray)



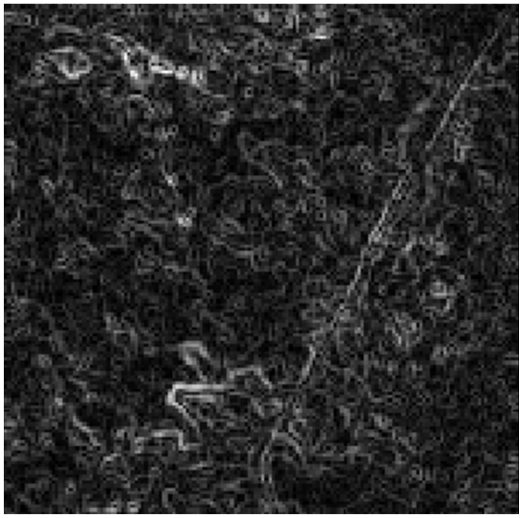
Successive
redilations within
mask remove small
background objects

Vogt, Peter, et al. "Mapping spatial patterns with morphological image processing." *Landscape Ecology* 22.2 (2007): 171-177.

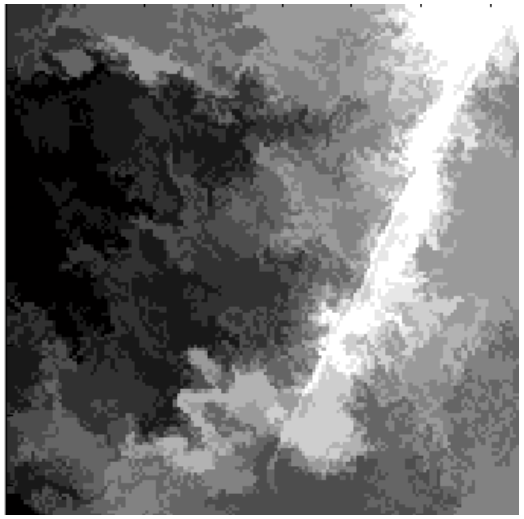
Morphological operations

Image enhancement - Morphology

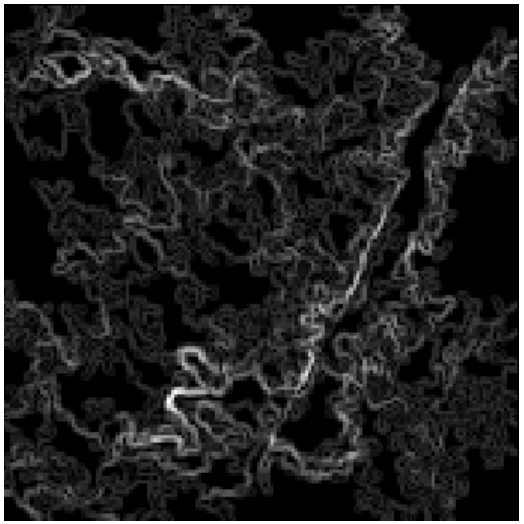
original



gradient

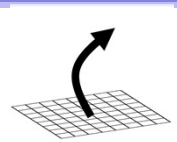


Morphologically reconstructed

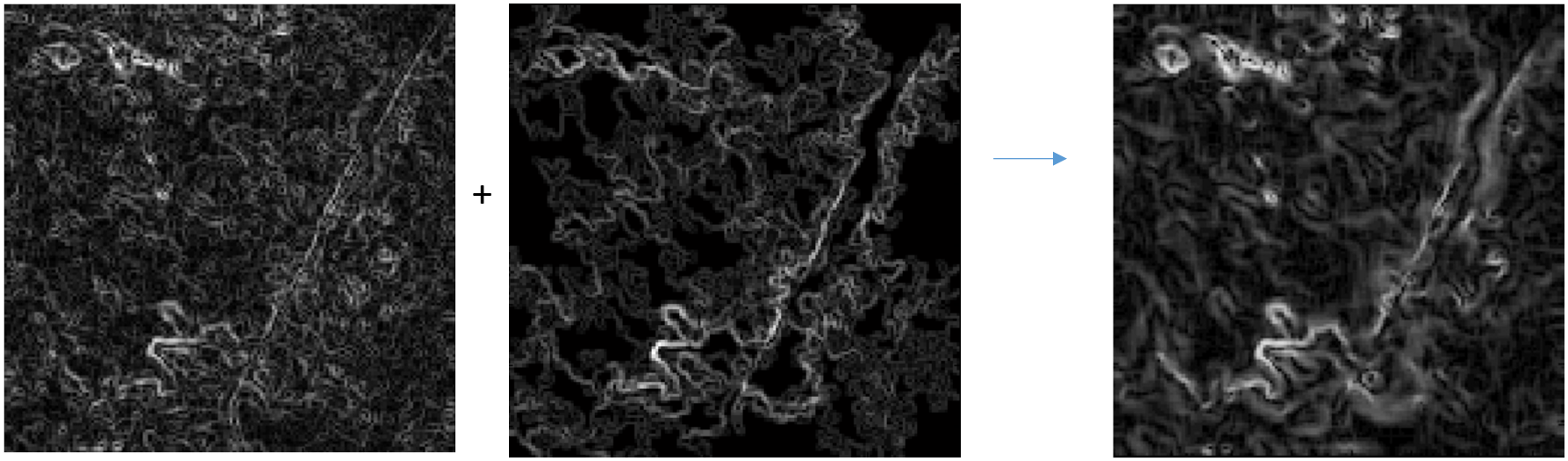


gradient

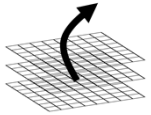
Feature extraction – Edges



- Loss of local but high variations in intensity after morphological reconstruction
- -> combine the two gradients



Feature extraction – multiple images

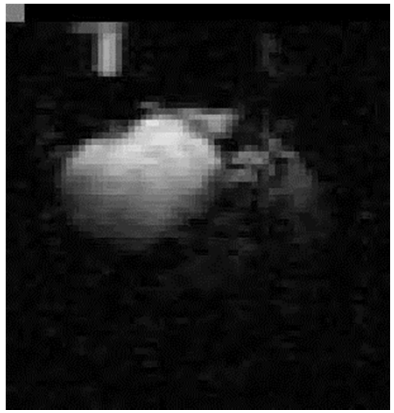


- Temporal sequence → take difference
- Panorama → stitch together

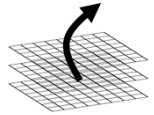
- Image arithmetic: images can be added, subtracted, multiplied element by element (E.g. multiply binary mask to original to keep intensity variations only in regions of interest)

- Example: measuring critical temperature threshold of ants (climate change vulnerability)
- Ants are filmed when subjected to lethal heat, by subtracting each frame to the previous one we can detect when they stop moving

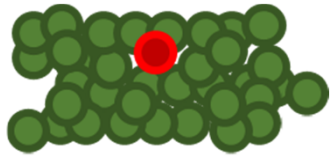
Unless the video camera moves…



Feature extraction – multiple images



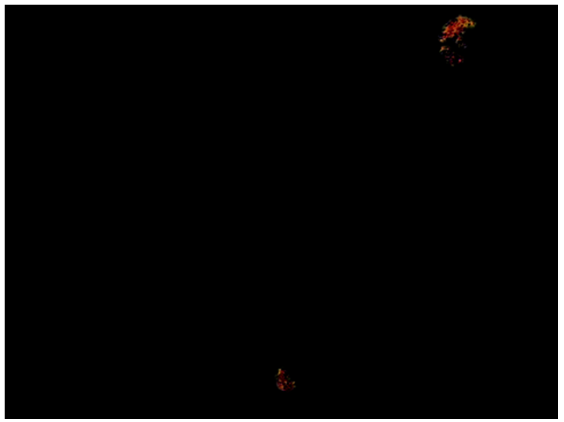
- Images alignment, registration
- First step for image sequences (after basic enhancing / de-noising)



- Example: detect tree flowering

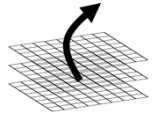
Time 1

Time k



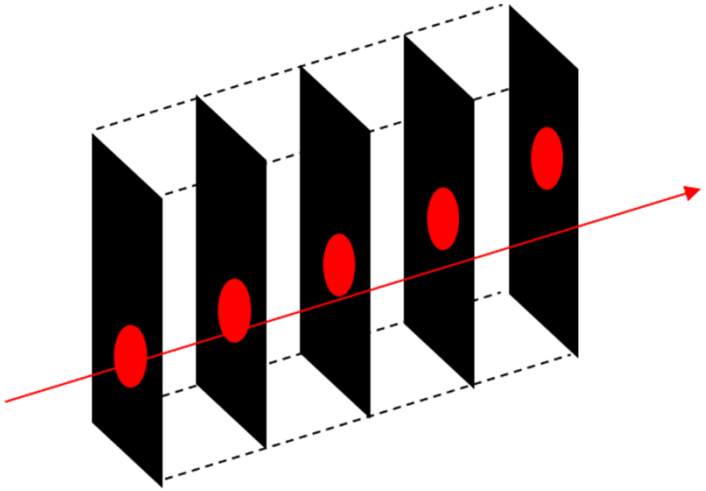
Difference

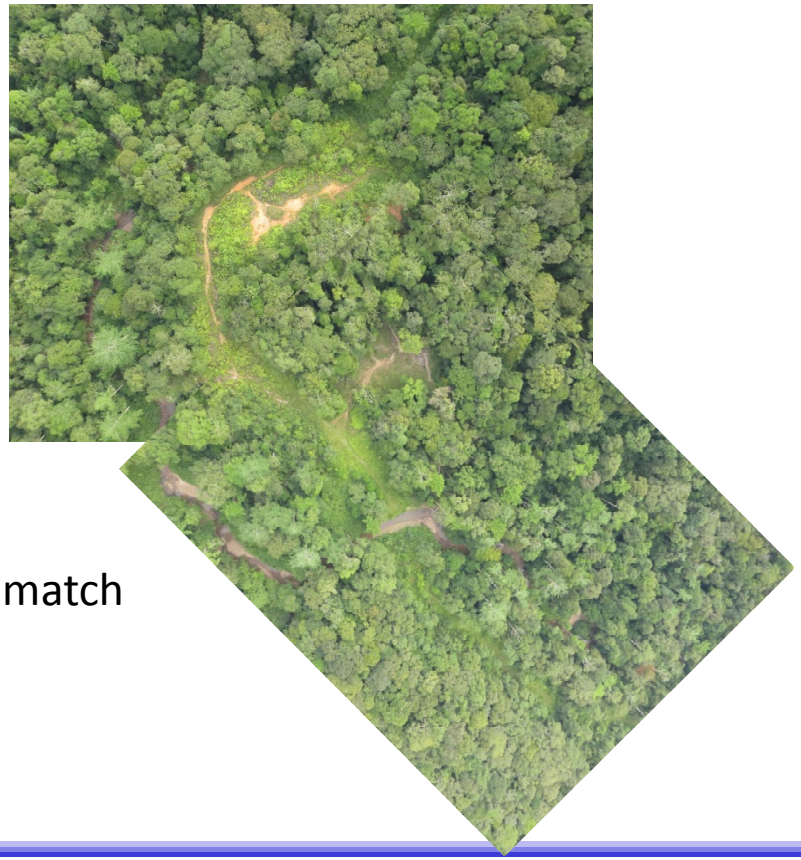
Image Registration



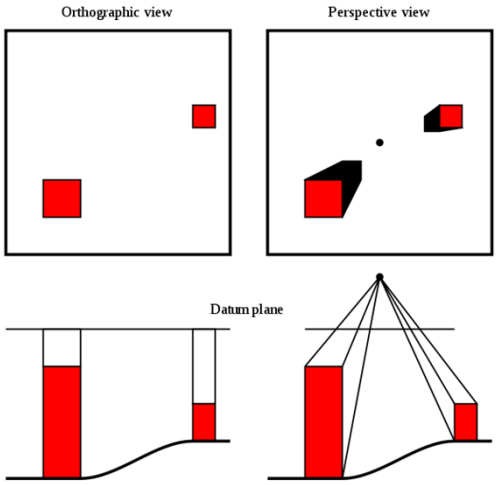
- Image registration: aligning two or more images of the same structure
- To correct for geometrical transformations

- Enable image difference in sequence
- Panorama stitching





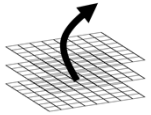
No match



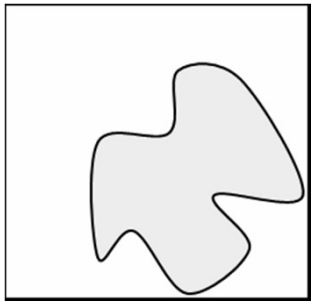
Unmanned Aerial Vehicles UAV

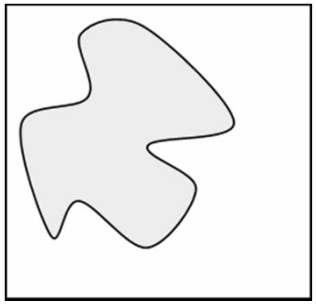
- Cheap, easy, great resolution, but moves a lot

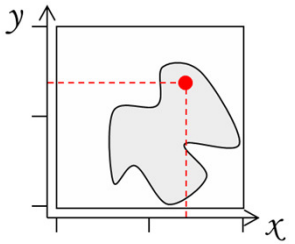
Image Registration

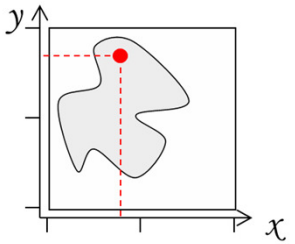


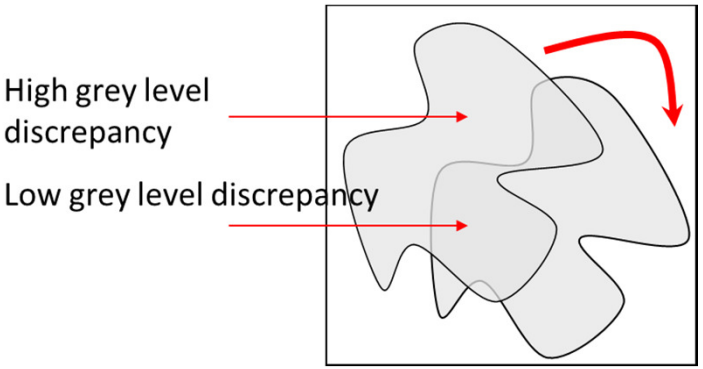
- Minimisation of grey level discrepancy (mutual information)
- To detect the geometrical transformations (linear algebra)

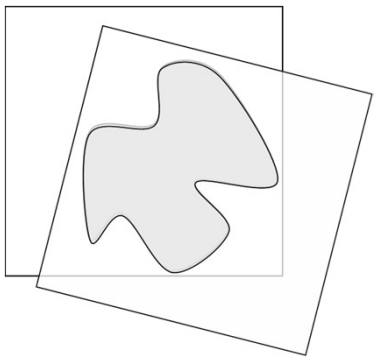












$$\begin{bmatrix} x\ new \\ y\ new \end{bmatrix} = \begin{bmatrix} V1 & V2 \\ V3 & V4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} V5 \\ V6 \end{bmatrix}$$

$$x\ new = V1x + V2y + V5$$

$$y\ new = V3x + V4y + V6$$

Image Affine transformations

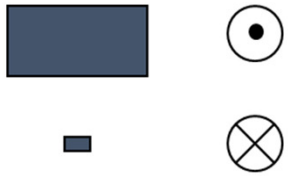
1) Vertical translation

$$\begin{cases} x_{new} = 1x + 0y + 0 \\ y_{new} = 0x + 1y + V6 \end{cases}$$

2) Horizontal translation

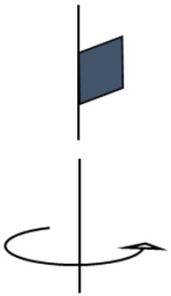
$$\begin{cases} x_{new} = 1x + 0y + V5 \\ y_{new} = 0x + 1y + 0 \end{cases}$$

3)Scaling



$$\begin{cases} x_{new} = V1x + 0y + 0 \\ y_{new} = 0x + V4y + 0 \end{cases}$$

1) Vertical shear



$$\begin{cases} x_{new} = 1x + 0y + 0 \\ y_{new} = V2x + V4y + 0 \end{cases}$$

2) Horizontal shear

$$\begin{cases} x_{new} = V1x + V3y + 0 \\ y_{new} = 0x + 1y + 0 \end{cases}$$

3) Rotation

$$\begin{cases} x_{new} = \cos(\theta)\, x - \sin(\theta)\, y + 0 \\ y_{new} = \sin(\theta)\, x + \cos(\theta)\, y + 0 \end{cases}$$

http://scikit-image.org/docs/dev/auto_examples/applications/plot_geometric.html#example-applications-plot-geometric-py

Interpolation

- After scaling, rotation, etc.. Estimate grid values
- Interpolation is the computation of points or values between ones that are known - or tabulated, using the surrounding points or values

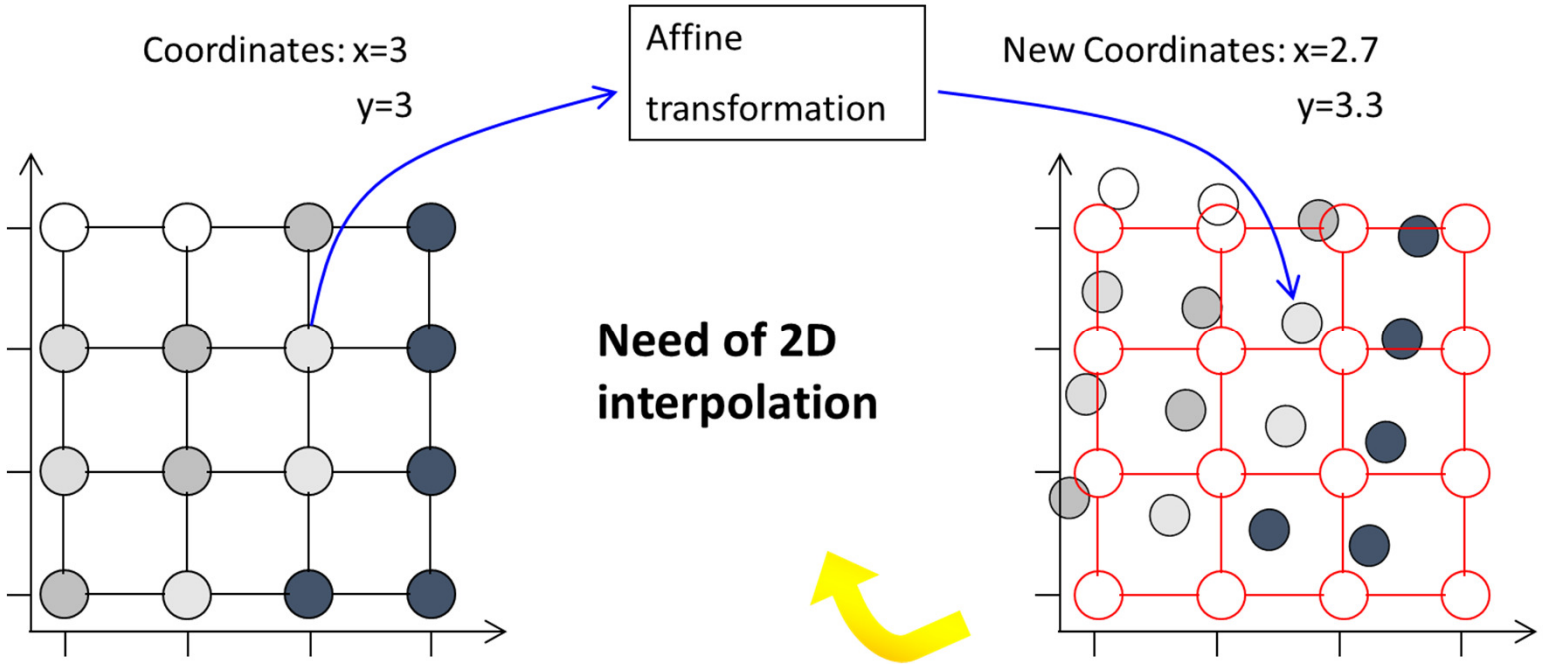
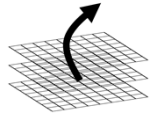


Image registration



- Rat brain fMRI video realigned:

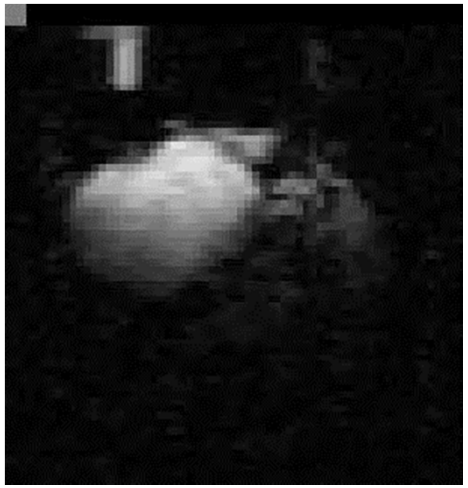


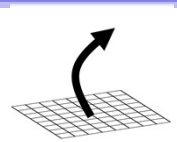
Image registration

- Stitching of aerial images:

Dedicated software to process UAV images: Agisoft Photoscan, visual SFM

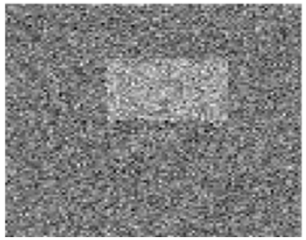
http://uk.mathworks.com/help/images/examples/registering-an-aerial-photo-to-an-orthophoto.html

Feature extraction – segmentation

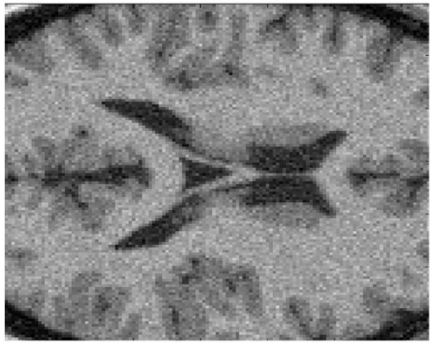


- Following the “pre-processing” (enhancement, basic feature extraction, registration) we can do some serious analysis
- Segmentation: finding and applying a mask on an image, to separate the object(s) of interest from “the rest”
- Delineate objects separately
- Create a labelled image (not necessarily ‘classified’ yet, objects are delineated but not identified)

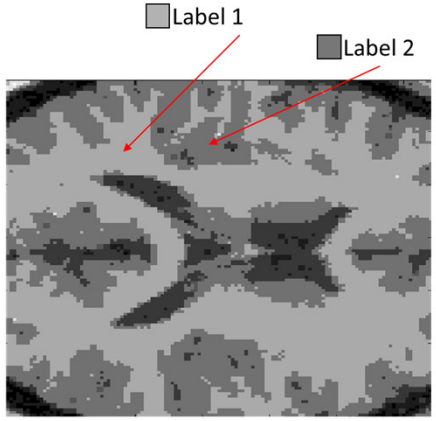
Noisy image made of ~ 150 grey levels



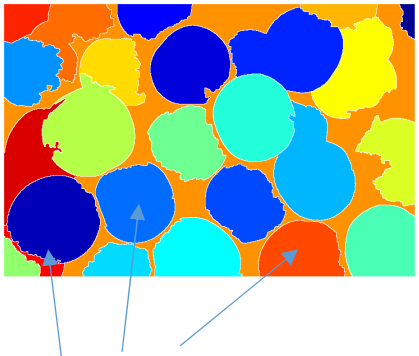
Mask, binary image (2 grey levels)



Original image

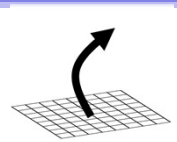


Segmentation



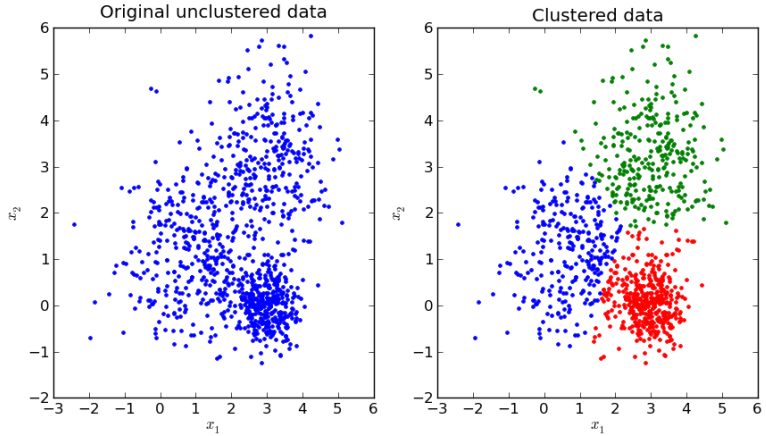
Different labels

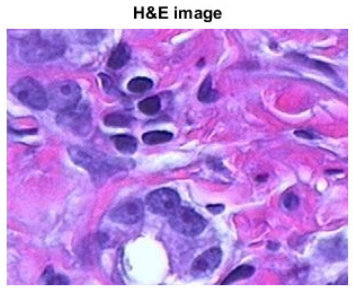
Feature extraction – segmentation

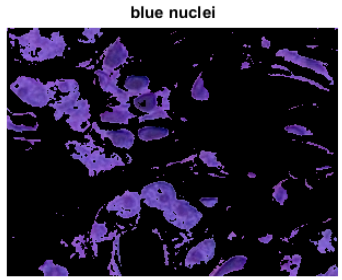


- Threshold: segmentation on histogram only
- More advanced segmentation techniques combine intensity with local properties (neighbourhood), and shape

- Clustering techniques, e.g. k-means
- Compute mean and variance of neighbourhood, cluster





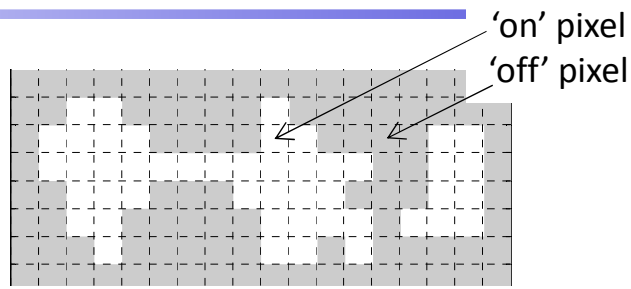


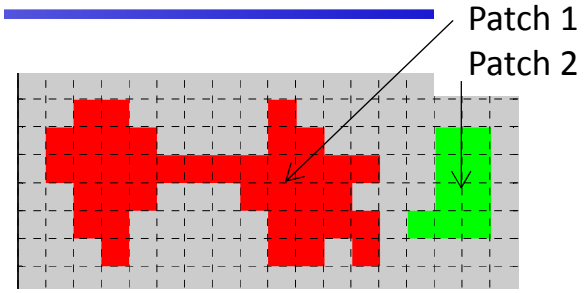
- http://uk.mathworks.com/help/images/examples/color-based-segmentation-using-k-means-clustering.html

- Takes into account local neighbourhood, but not shape / connection / spatial links

Feature extraction – binary segmentation

<u>Common technique:</u>
Connected Component Labelling (CCL)

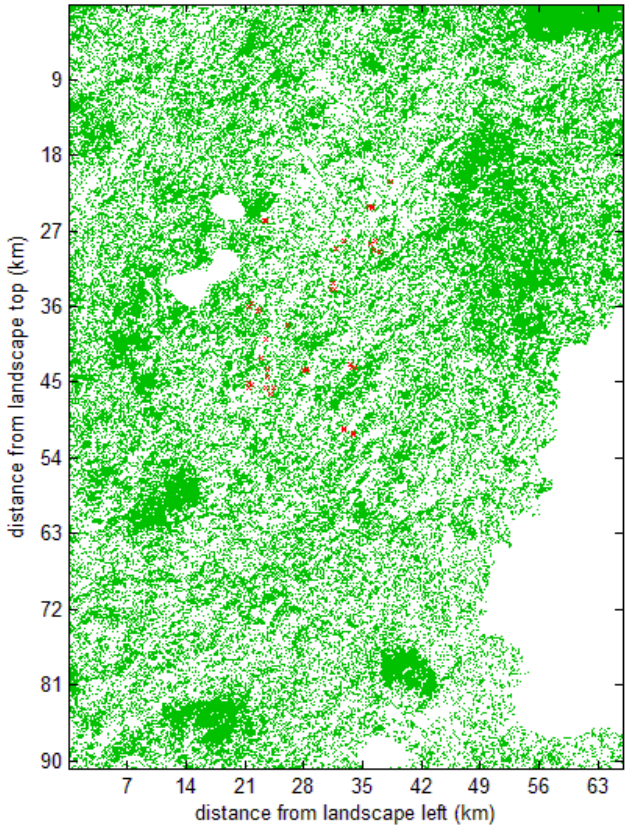




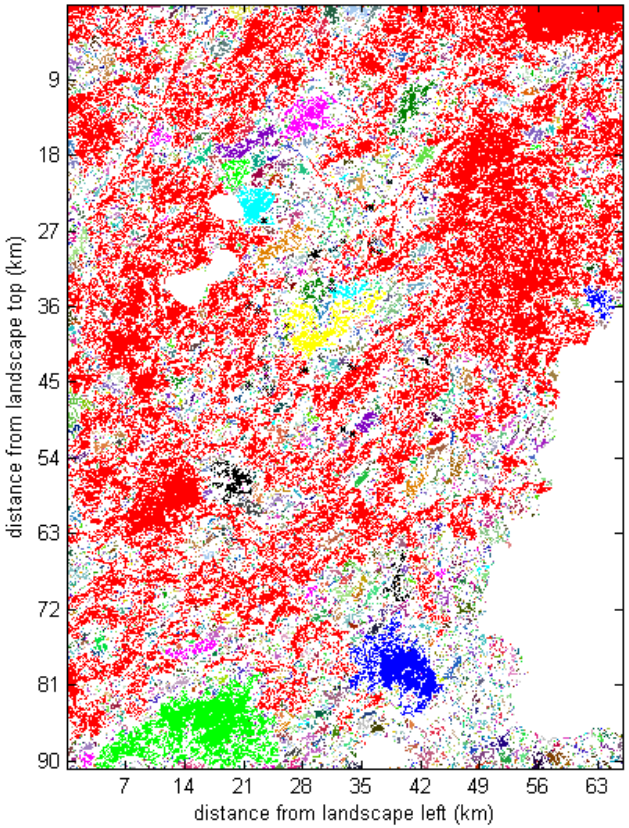
To separate objects in masks

But very sensitive to noise

Habitat binary map

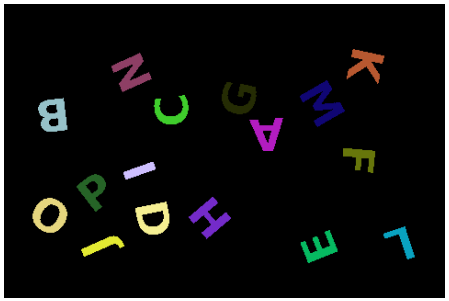


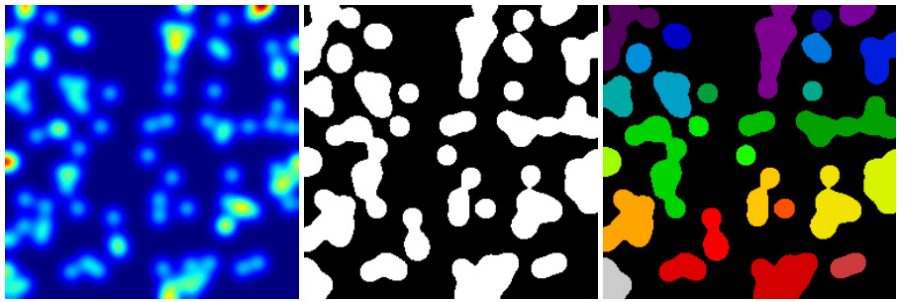
Connected component labelling



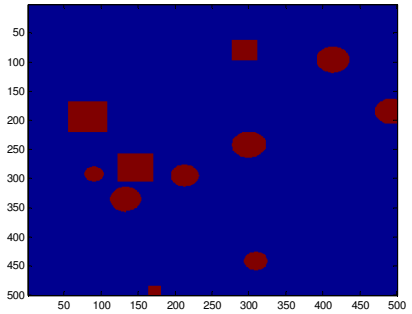
Feature identification – binary segmentation

- However Connected Component Labelling can often serve as a first step:

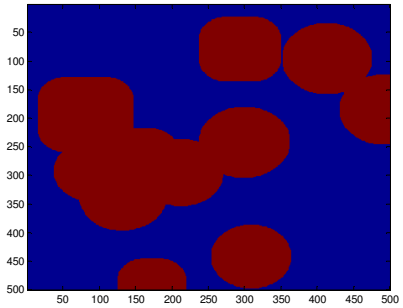




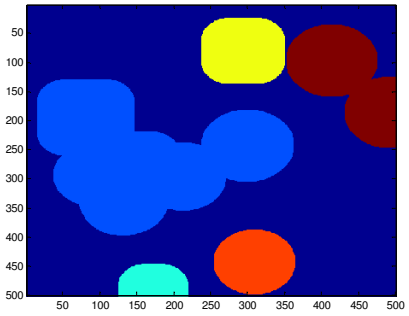
- Example: computing dispersal area:



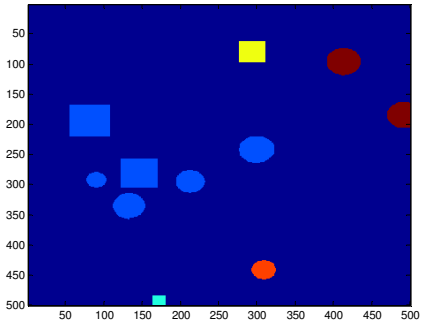
Habitat mask



Dilation



CCL



Multiply by mask:
Group of fragments
where dispersal is likely

Feature extraction – segmentation – morphology

- A robust and often used algorithm: Watershed
- Identifies homogeneous zones in a grayscale image (from the gradient)
- Identifies compact shapes in a binary image (from the distance image)
- Can be marker guided -> allows segmentation at different levels

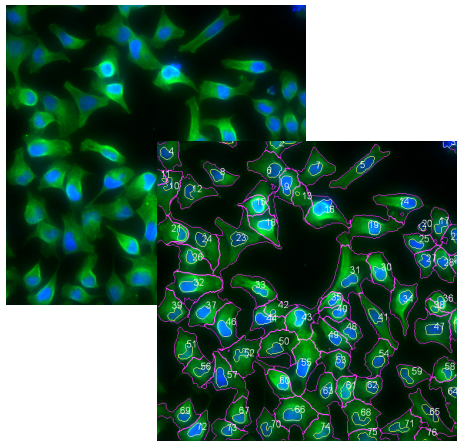
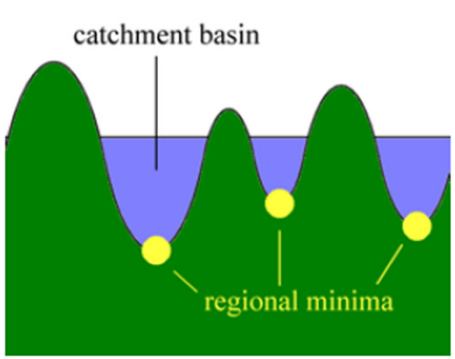
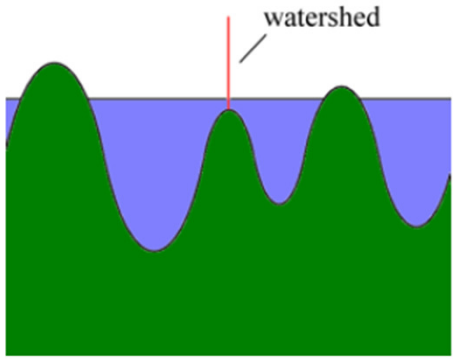
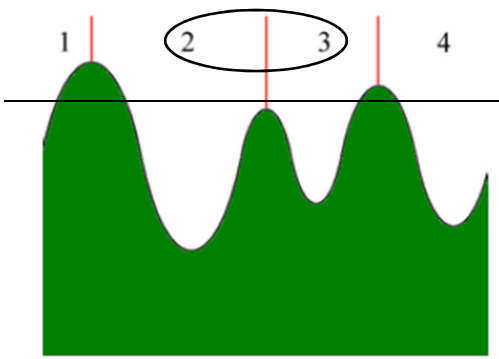


Image seen as a landscape







Guided: perception threshold

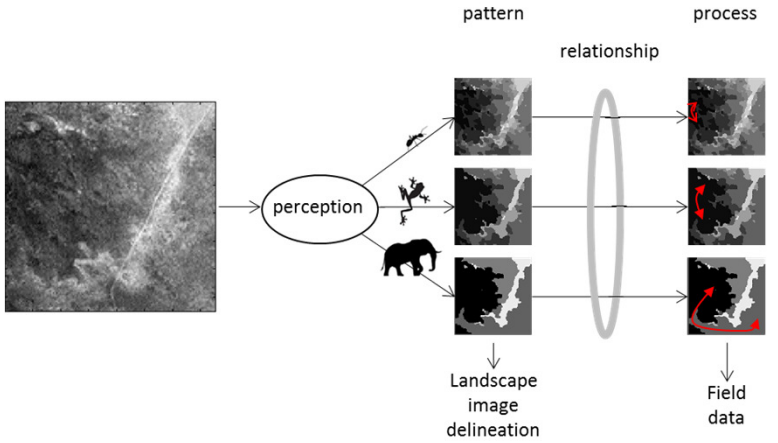
Golodetz, S., 2008a. Watersheds and Waterfalls (Part 1). *Overload*, 83, pp.4–9.

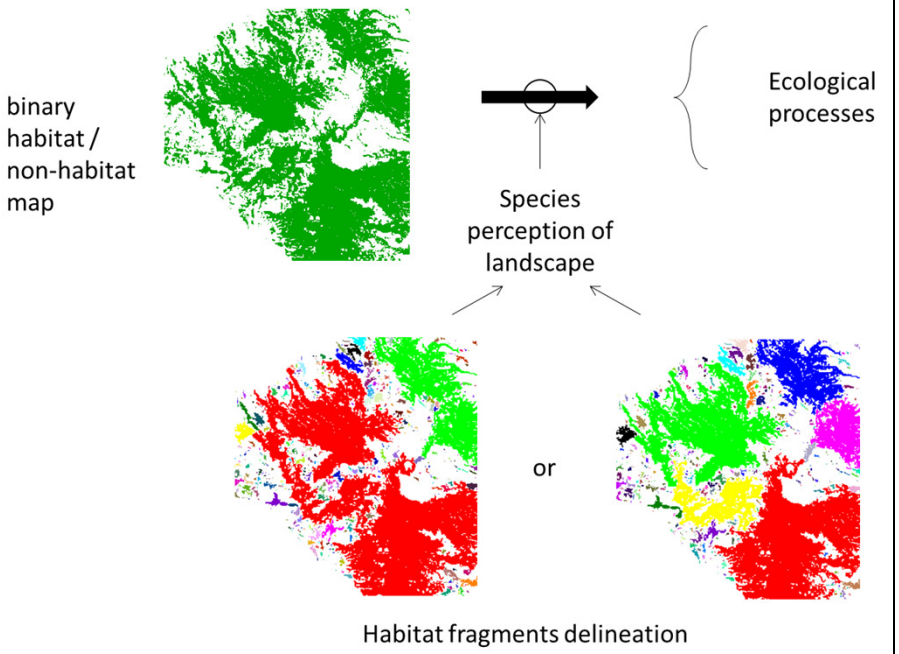
http://uk.mathworks.com/help/images/examples/marker-controlled-watershed-segmentation.html

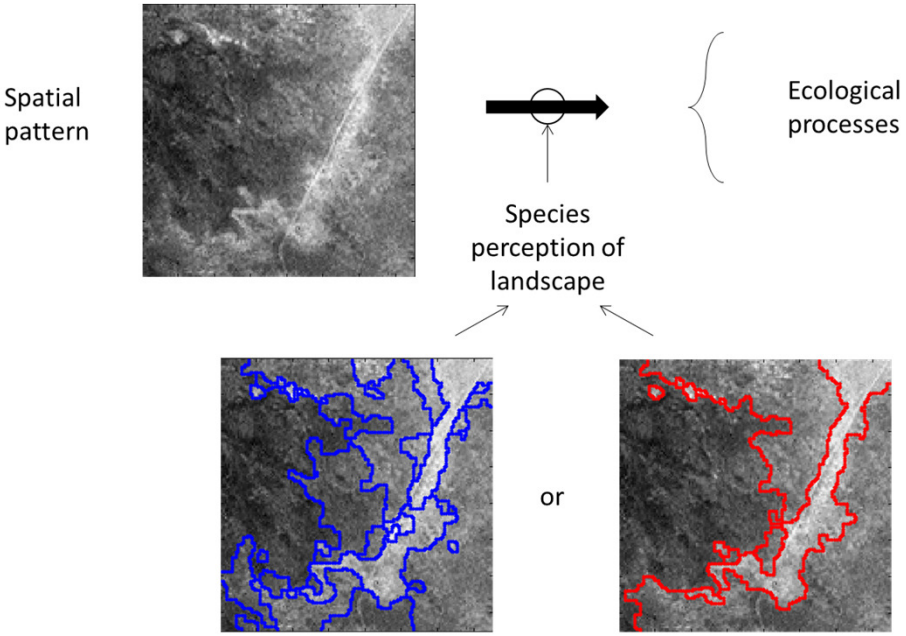
Marker-guided watershed

Example:
To find different land cover types in NDVI image
To find habitat fragments in habitat map

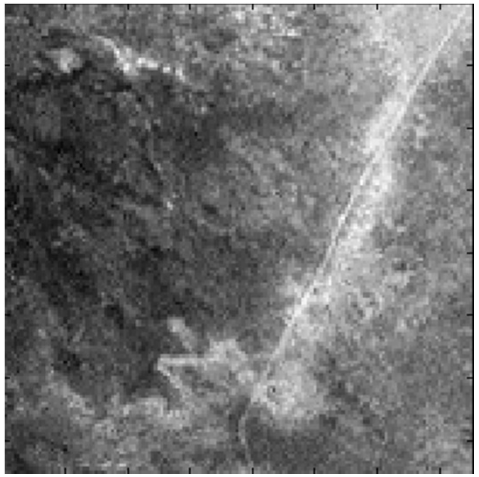
Multiple segmentations depending on species perception





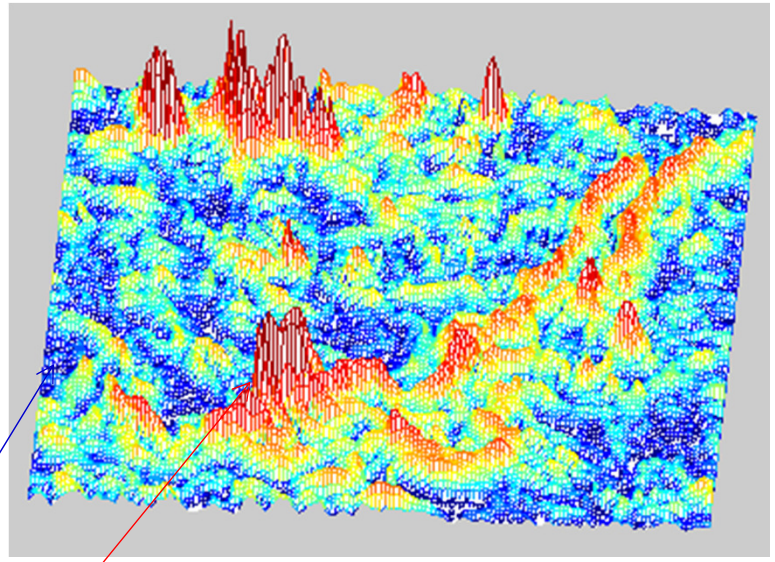


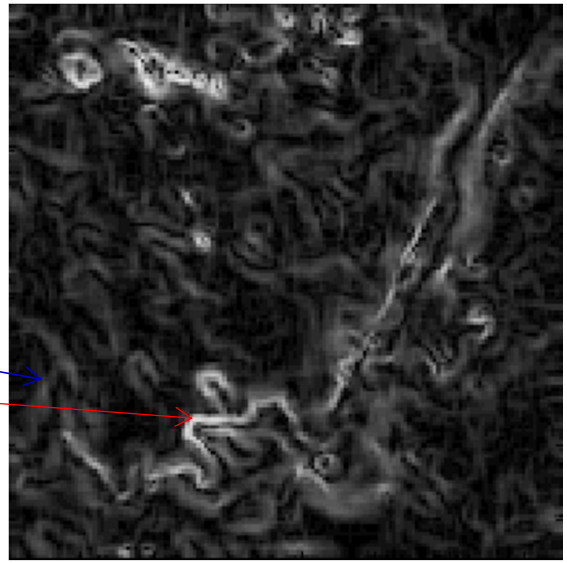
Watershed, grayscale



Gradient image

Segmentation image

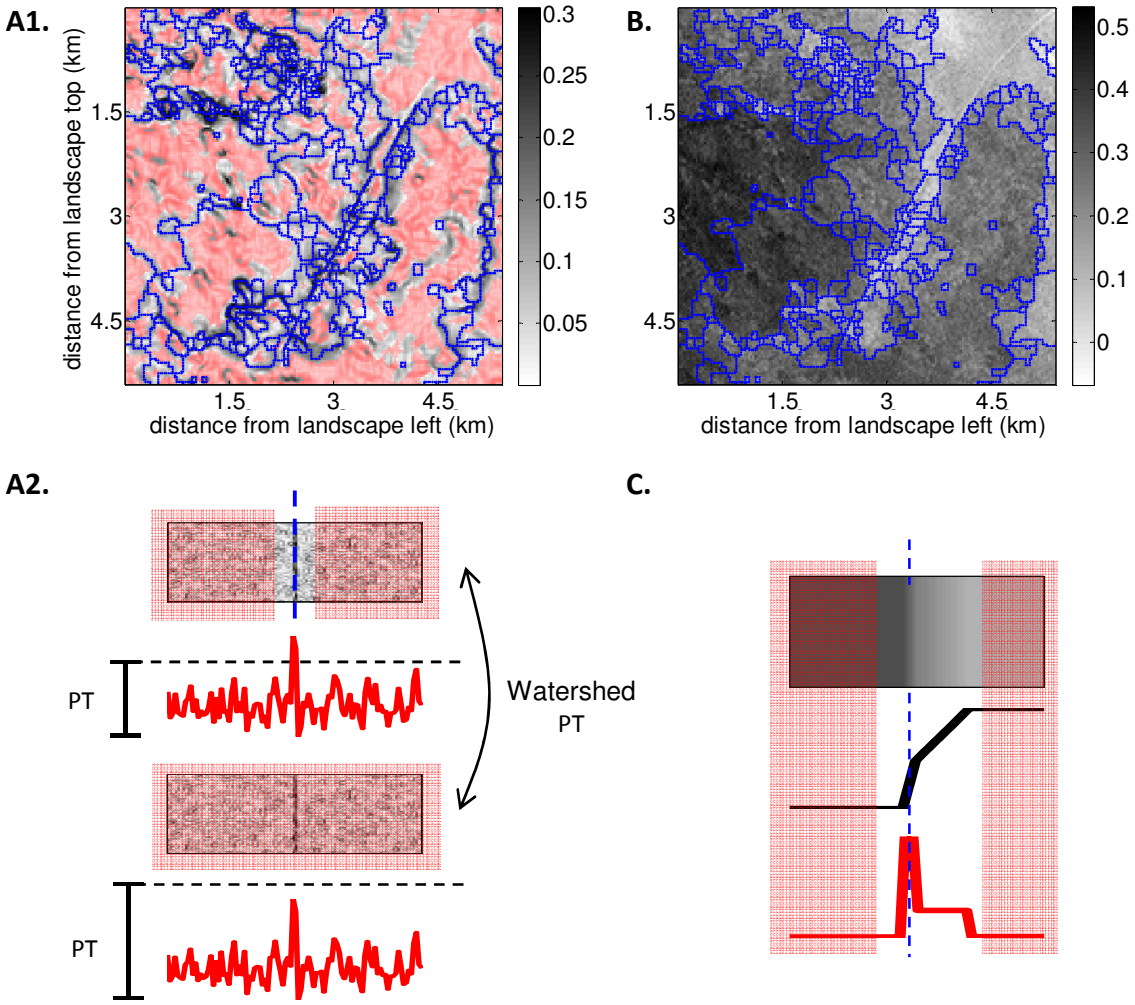




A patch: homogenous area

Boundary: discontinuity

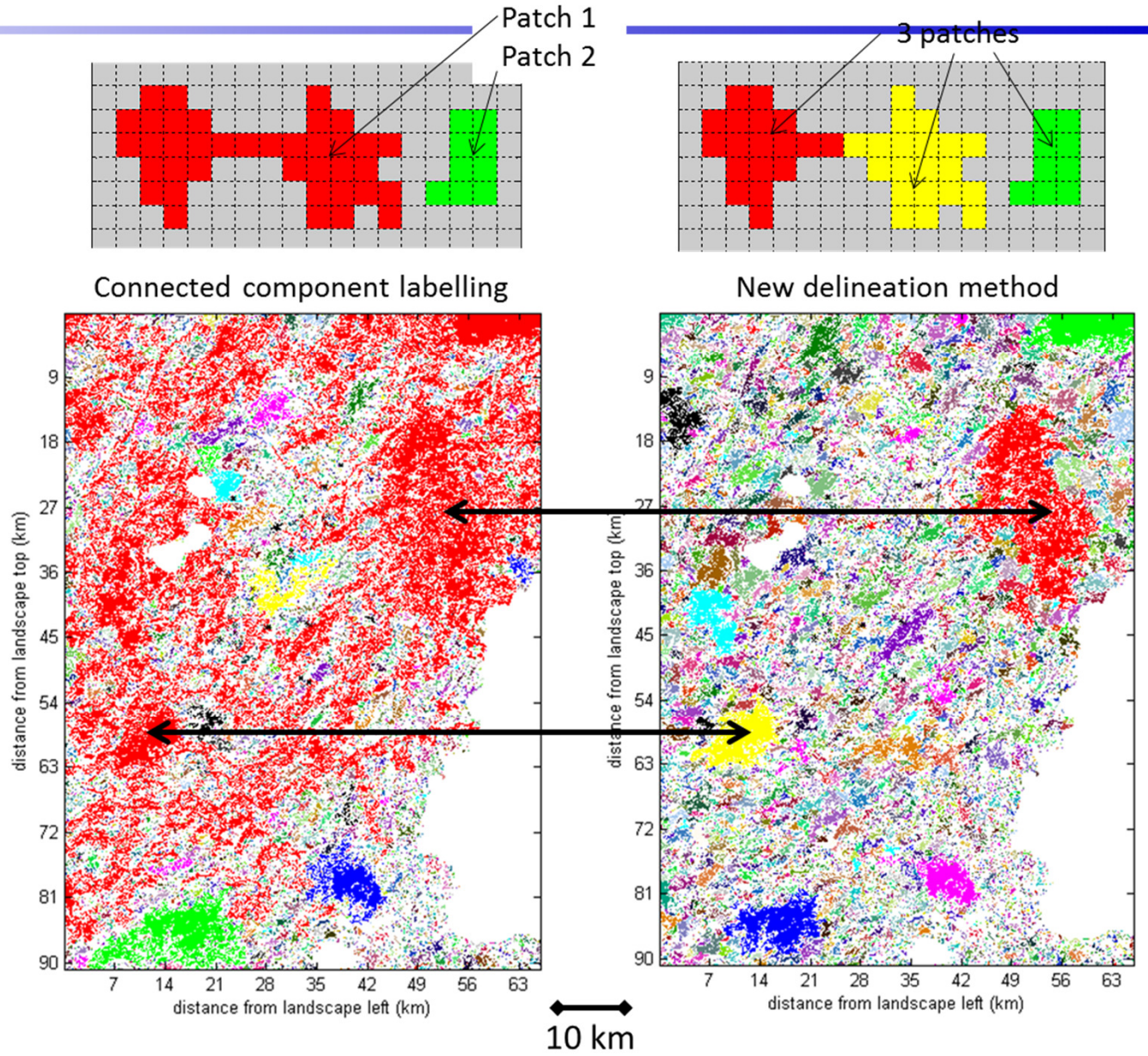
Watershed, grayscale



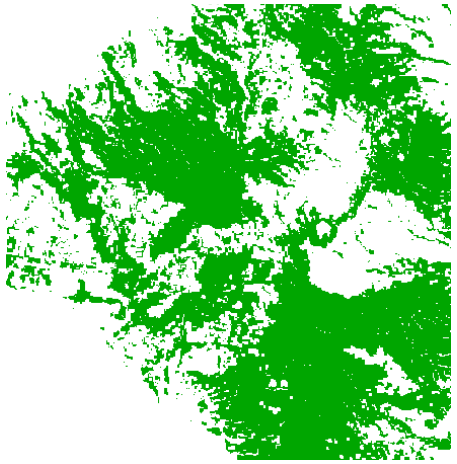
Watershed, binary

Method disconnects weakly connected pixels

Delineation of fragment is flexible



Watershed, binary



Distance map

Segmentation image

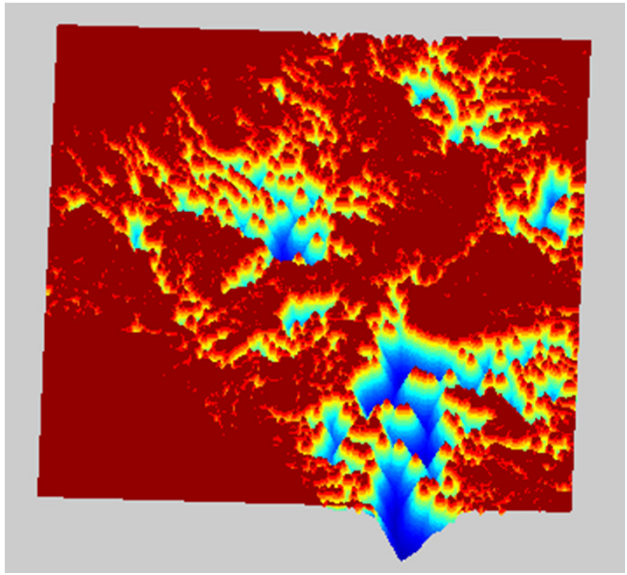
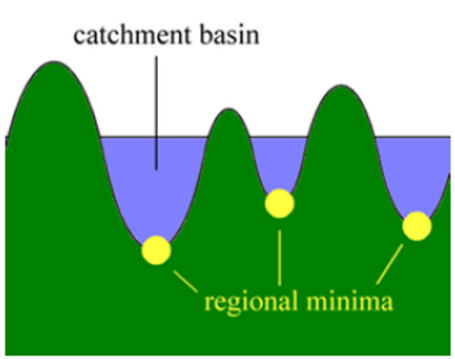
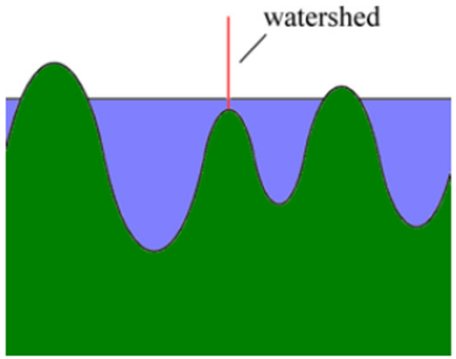
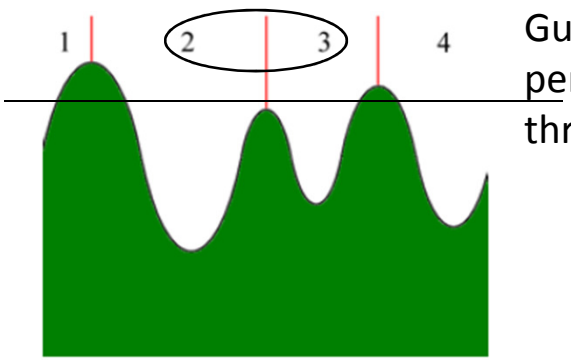


Image seen as a landscape



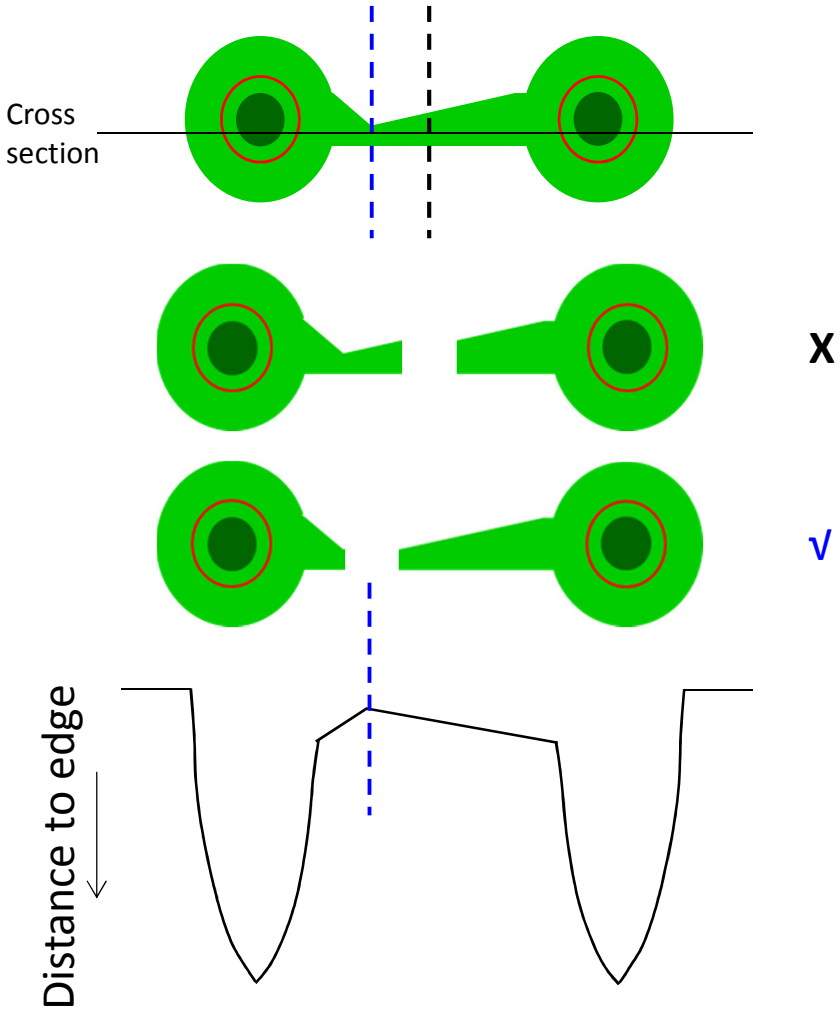




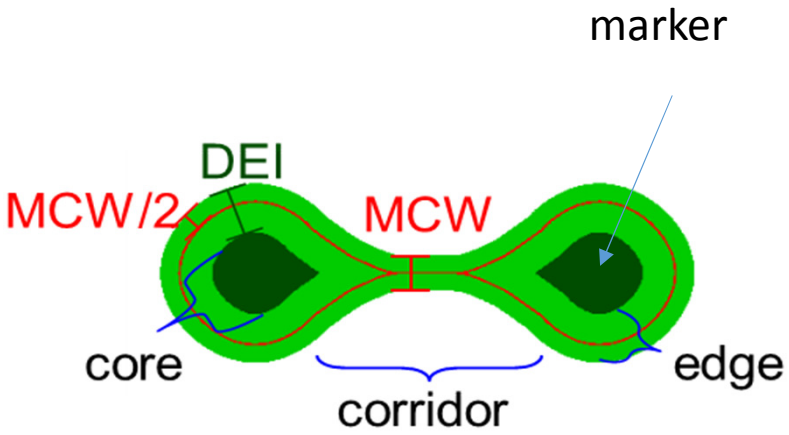
Guided: perception threshold

Golodetz, S., 2008a. Watersheds and Waterfalls (Part 1). *Overload*, 83, pp.4–9.

Watershed, binary



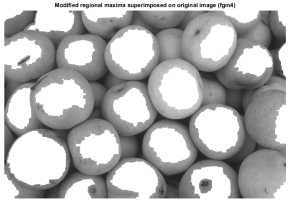
- Weak links should be disconnected at their thinnest section
  - ➢ distance to edge
- At the thinnest section the distance to edge goes through a regional maximum



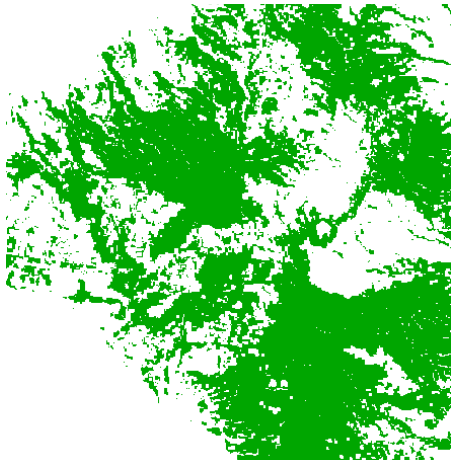
The watershed disconnects at the thinnest section of the link

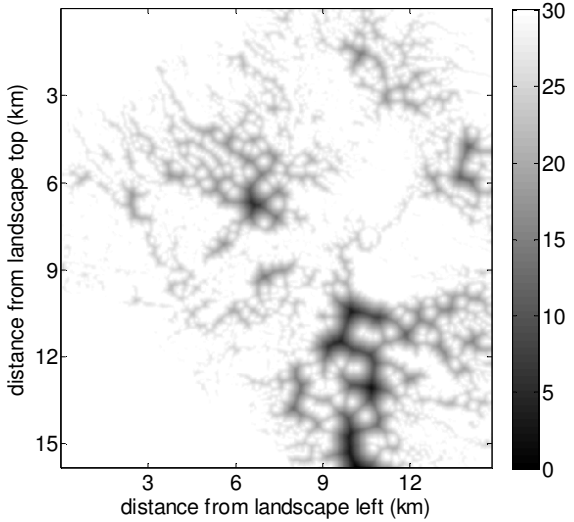
Finding markers for Watershed segmentation

- If markers are not used watershed leads to over-segmentation (every dent in gradient image)

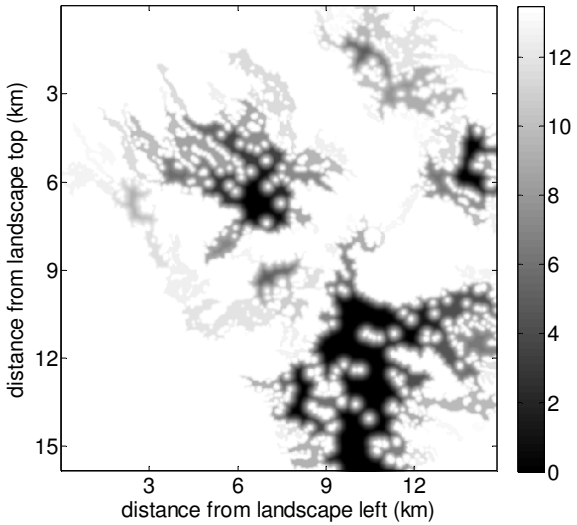


- Gradient image must be enhanced
- Morphology is useful to specify the markers (both gradient and distance image):
  - Reconstruction
  - Detection of regional maxima and minima
  - Impose minimum





Distance image



Threshold distance and impose min in basins: shows flow through landscape, connectivity

http://uk.mathworks.com/help/images/examples/marker-controlled-watershed-segmentation.html
http://uk.mathworks.com/help/images/morphological-reconstruction.html#brcw3z0

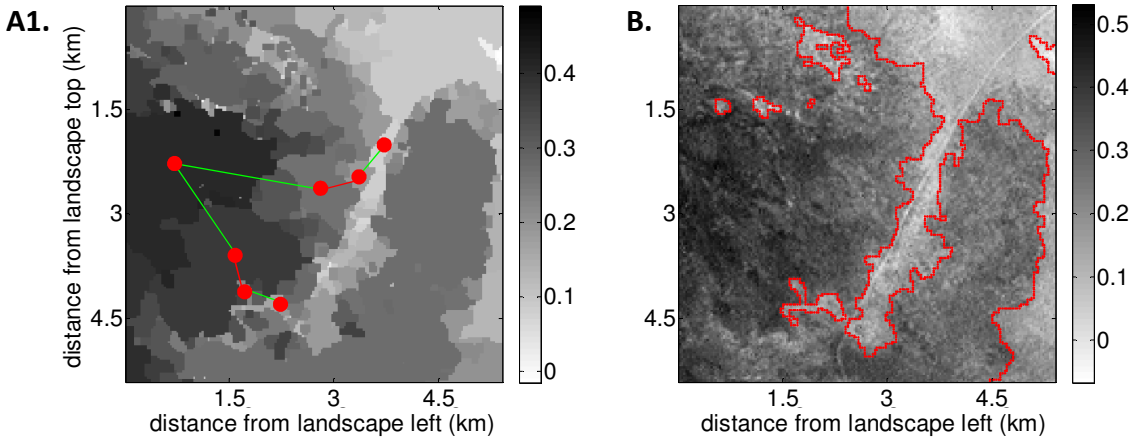
Waterfall, region merging, greyscale

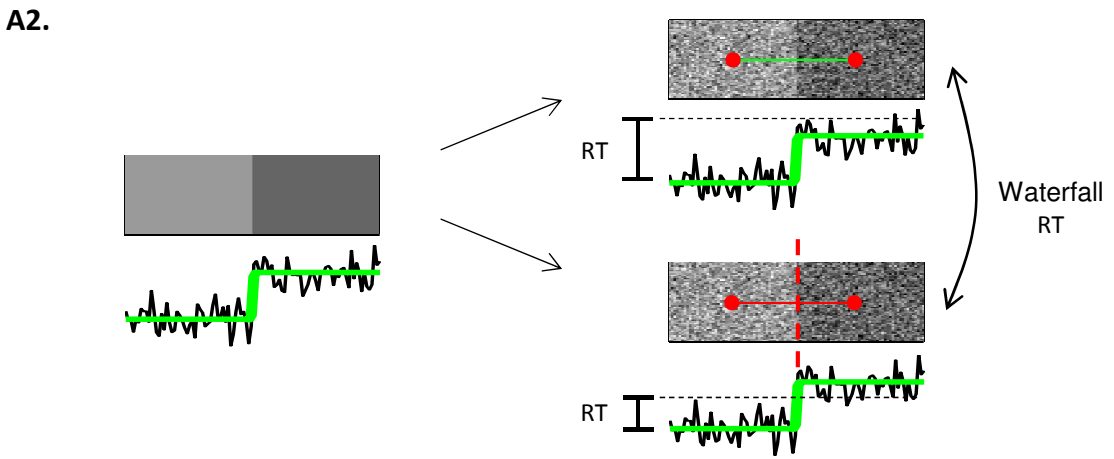
Even after enhancement and well chosen markers, watershed image may still be over segmented.

Use Waterfall algorithm.

Watershed: Delineations of local variations

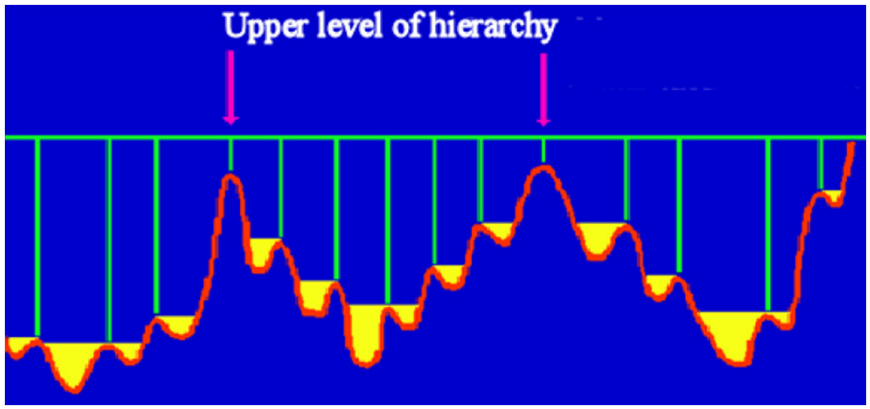
Waterfall: reconnect all regions of which difference is not relevant





Golodetz, S., 2008. Watersheds and Waterfalls (Part 2). *Overload*

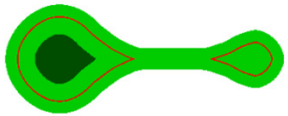
Waterfall, region merging, binary



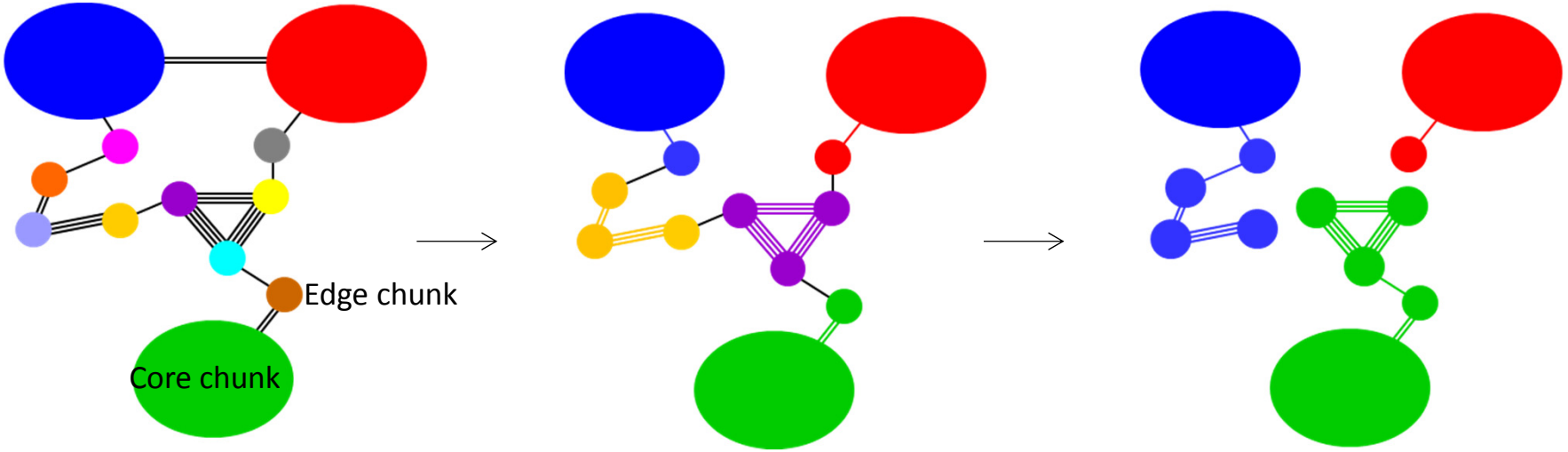
Beucher, S. & Marcotegui, B., 2009. P algorithm, a dramatic enhancement of the waterfall transformation

- Waterfall: find hierarchy of regional maxima
  - ➢ watershed on the regional adjacency graph
- graph connection values: width of habitat links

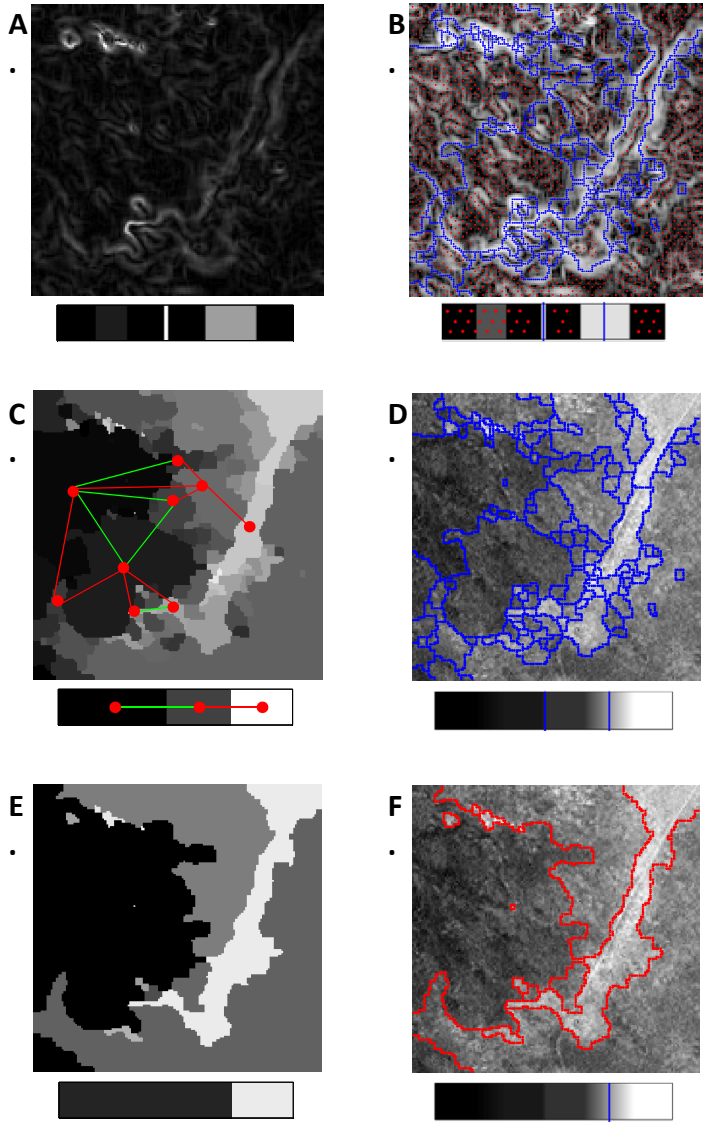
1 core region = 1 fragment



- Guided: reconnect edge chunks to a core fragment



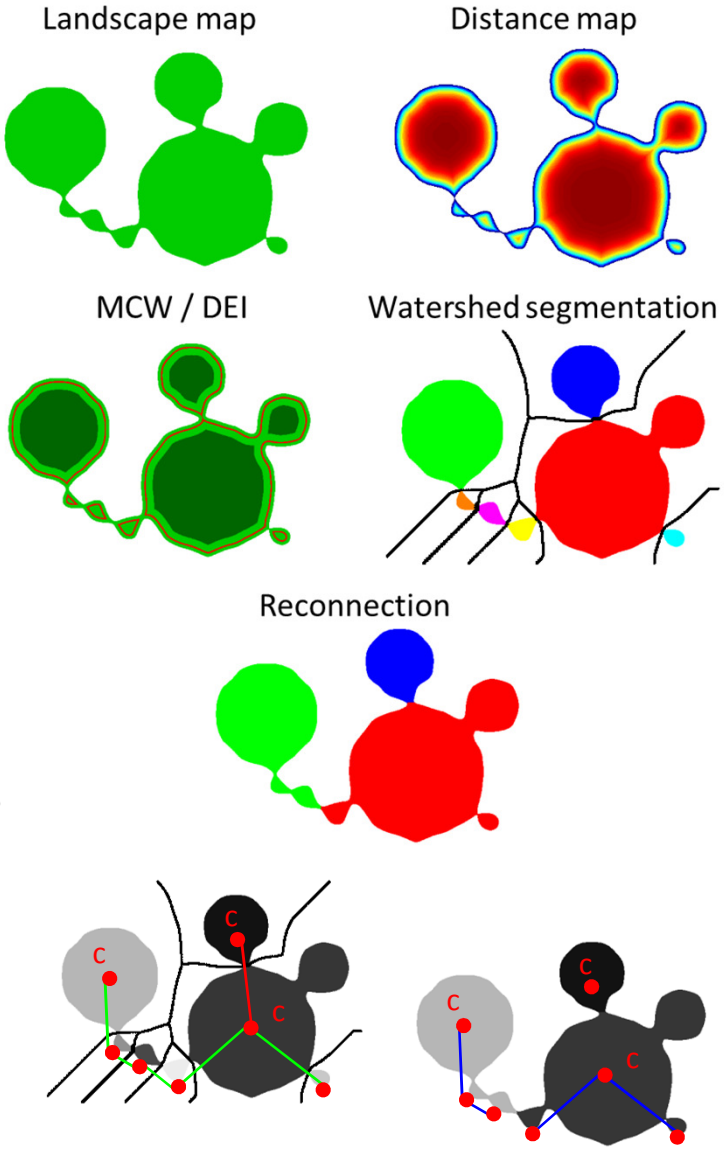
Summary of marker guided watershed / waterfall steps



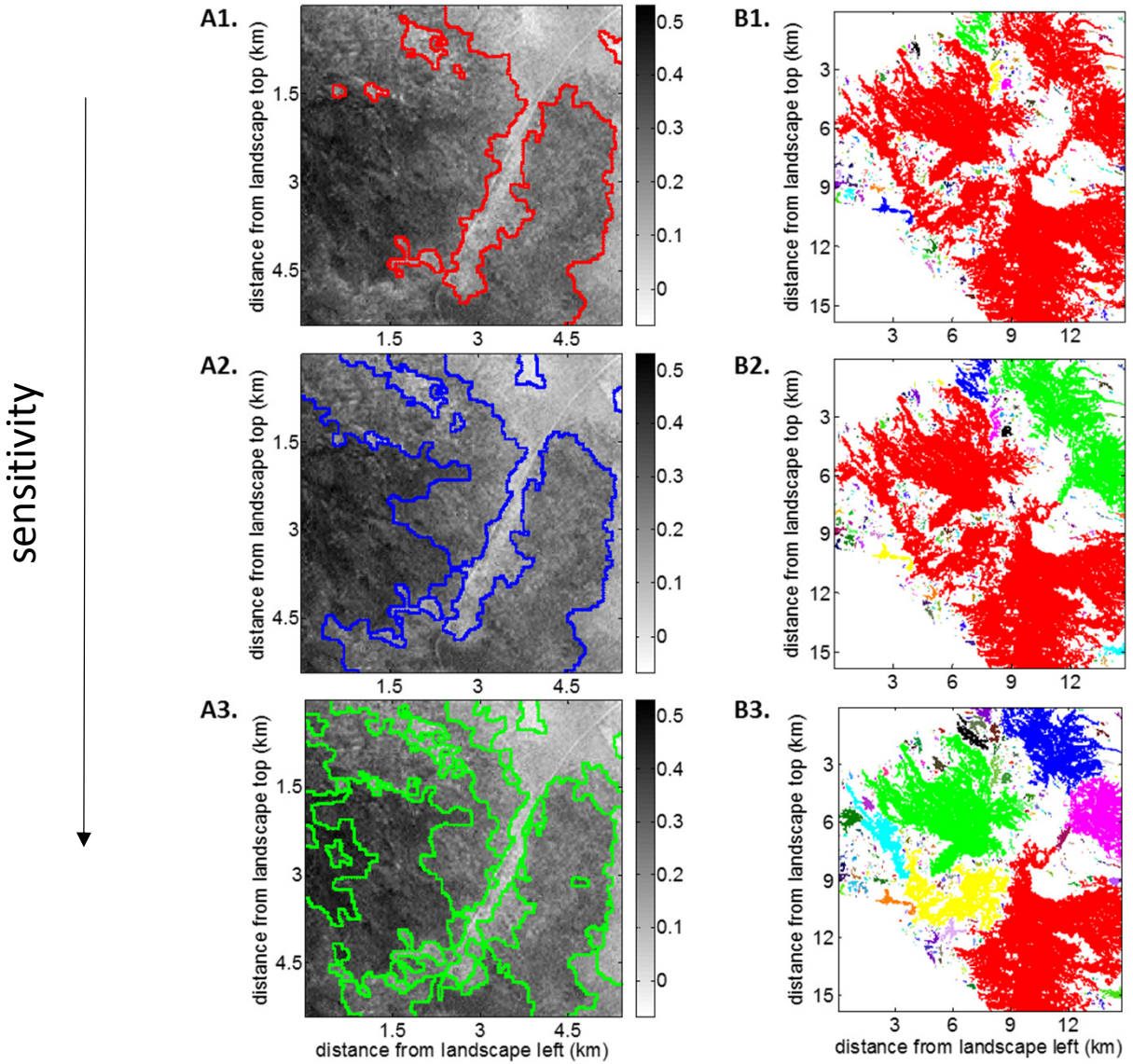
Find markers

Find basins

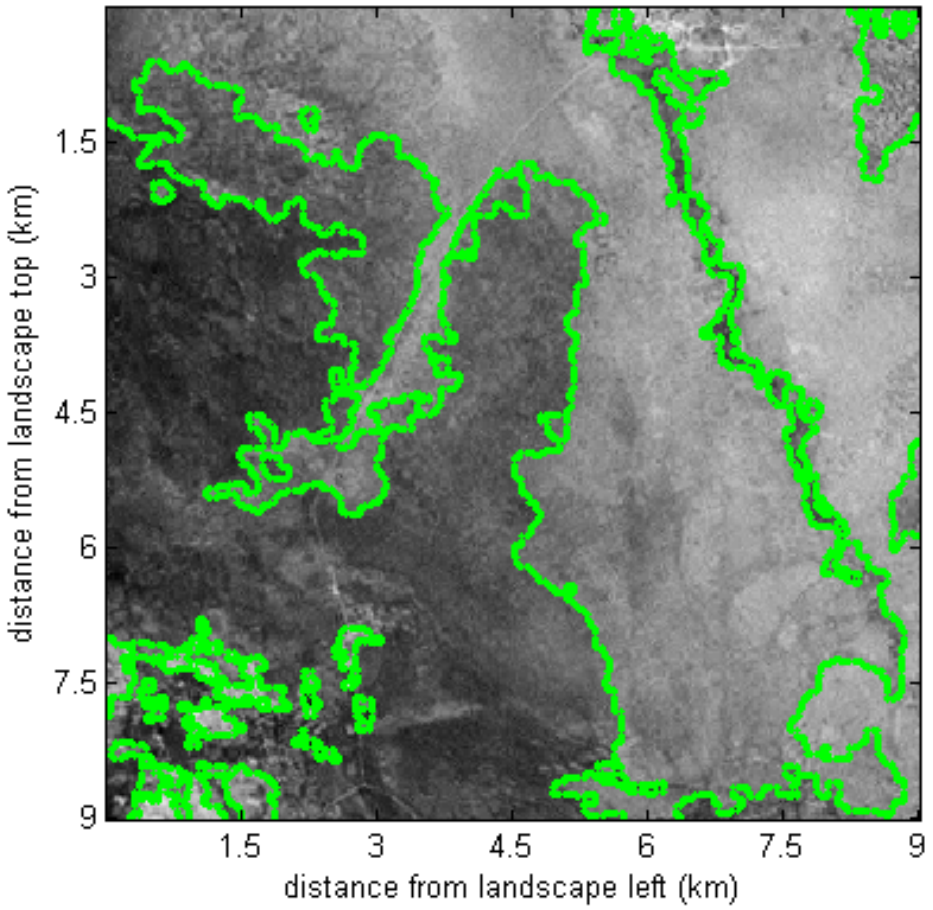
Reconnect similar regions

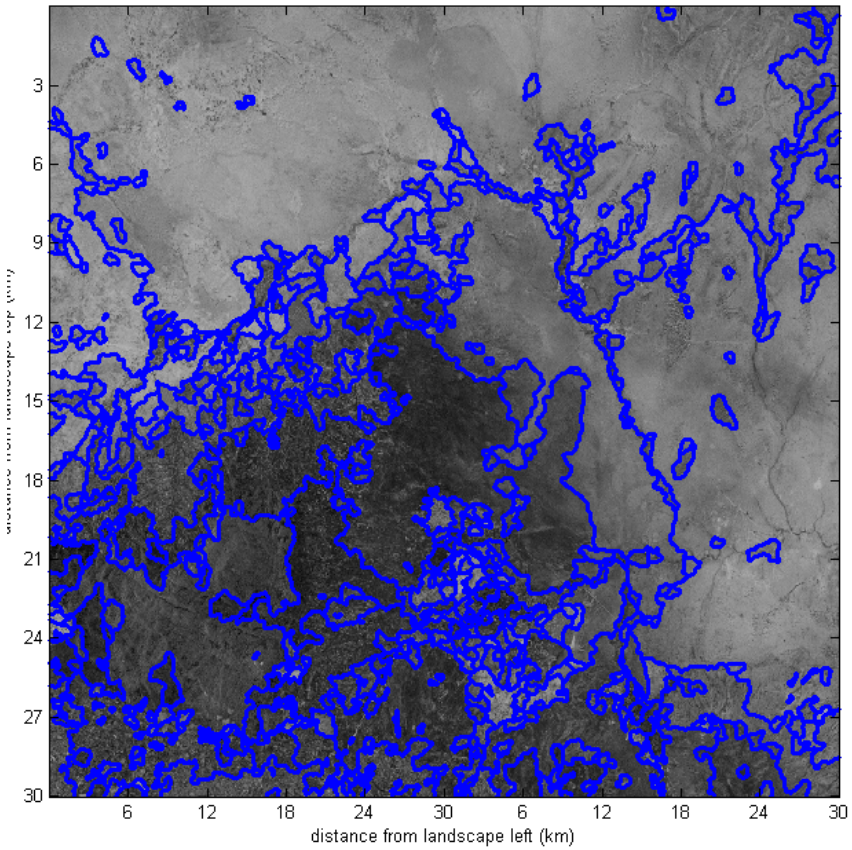


Watershed, waterfall results

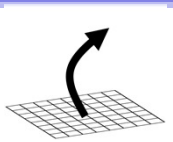


Example delineations, Watershed, Waterfall

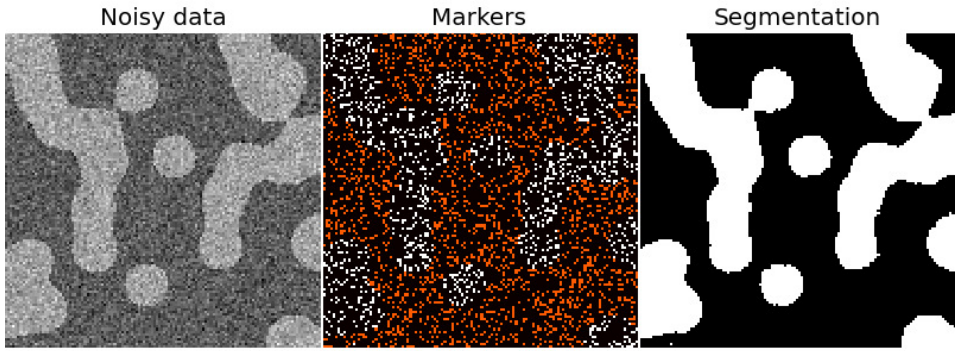




Similar segmentation algorithm: Random Walker

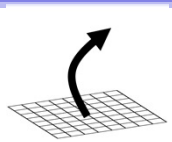


- Marker guided, markers determined by extreme values (e.g. tails of histogram, or morphology)
- Diffusion depending on gradient (difficult across high gradients), along the least resistance path
- Pixel gets label of marker that reaches it first
- Probabilistic approach

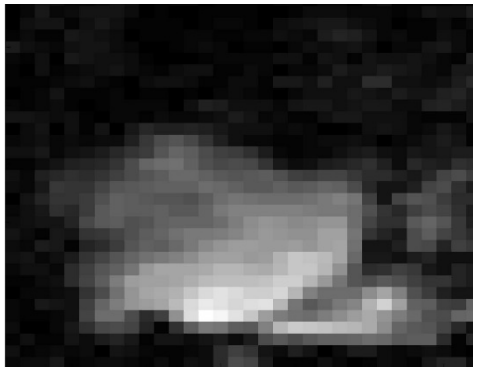


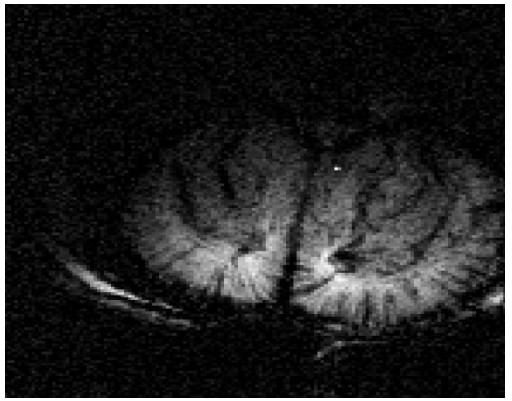
- Python example:
- http://scikit-image.org/docs/dev/auto_examples/plot_random_walker_segmentation.html#example-plot-random-walker-segmentation-py

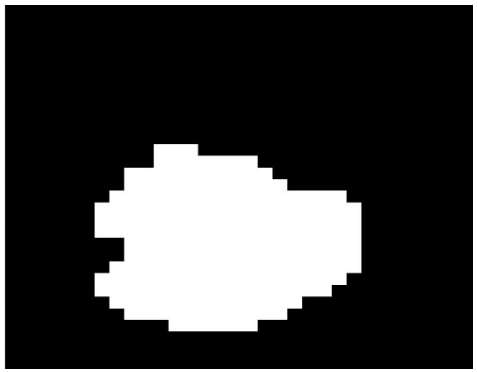
Another segmentation algorithm: graph cut

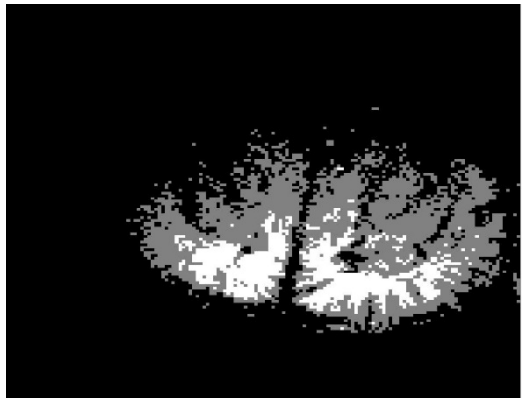


- Graph cut: segment foreground from background
- photographs

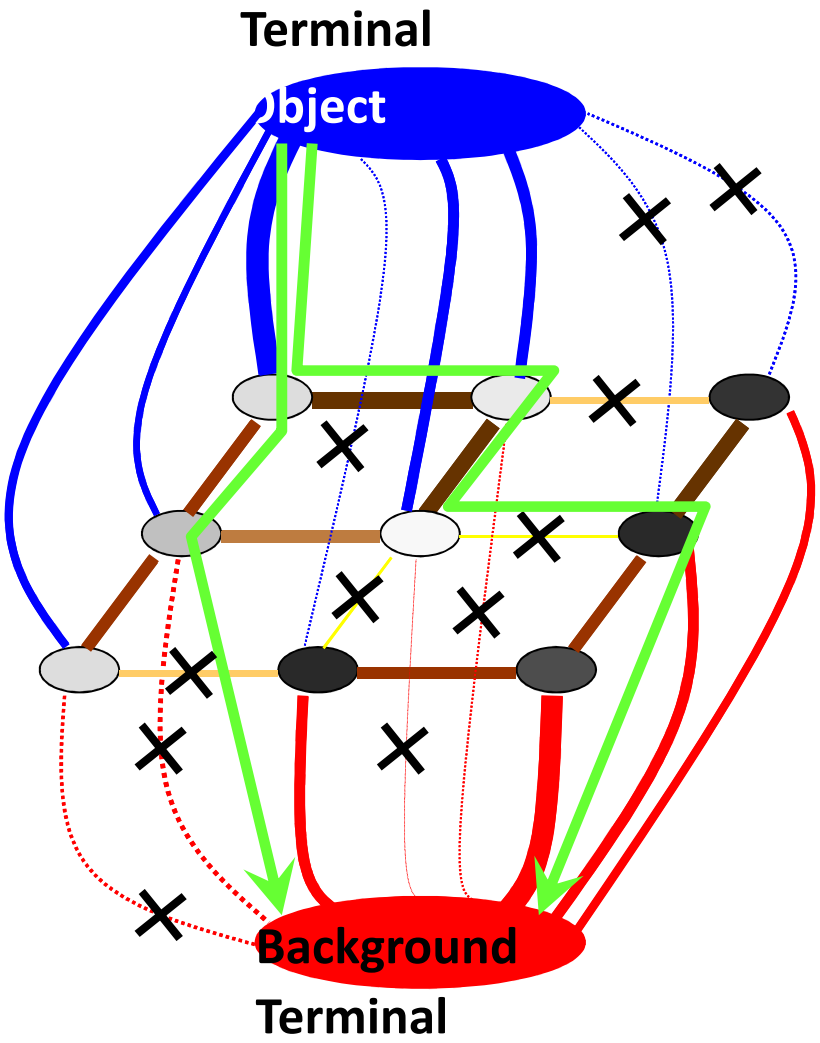






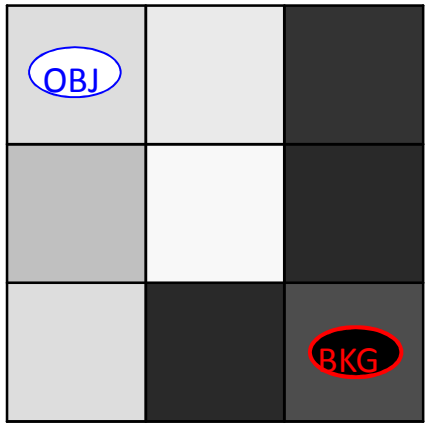


Graph Cut Algorithm

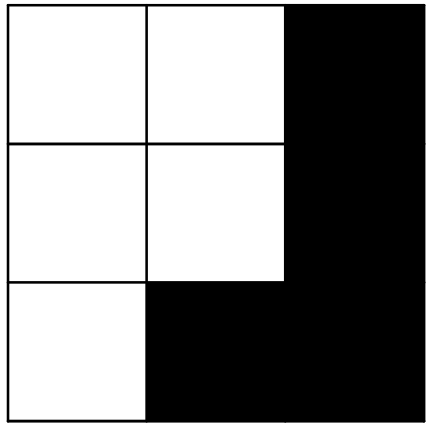


Graph

Image to segment



Result



All pixels are linked to the 2 terminals and to their neighbours. The cost of these links is high if the pixels are similar, low if they are not.

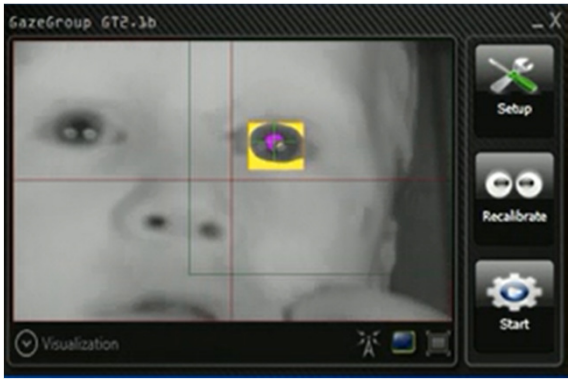
The algorithm finds all the possible paths from one terminal to the other, and cut the weakest link on each path.

Ultimately, all pixels are linked to only 1 terminal, hence there are 2 sets of pixels (background and object) = the image is segmented.

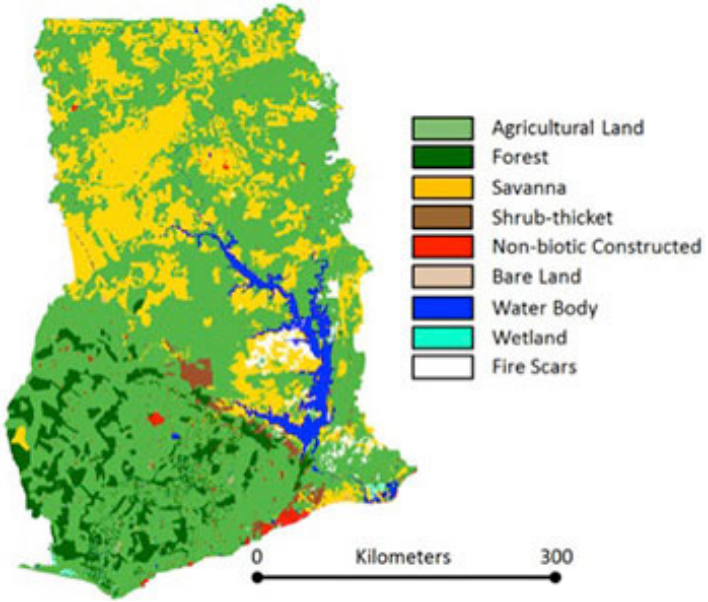
(Boycov et al. 2002)

Feature identification

- After segmentation of objects of interest -> identification
- E.g. how to extract only blobs, lines or corners ?



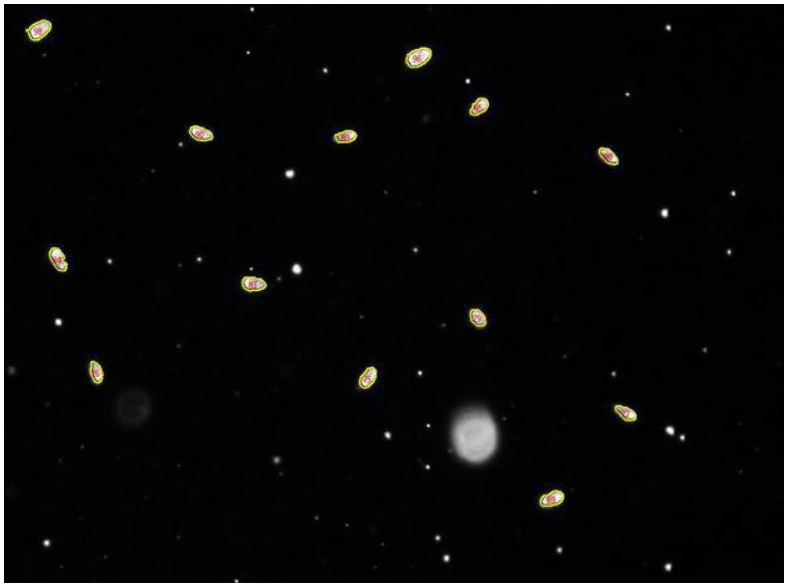
- Select a region of interest, containing a searched object (e.g. eye)
- Classifying objects w.r.t. shape, or other properties
- Distinguish letters in text
- E.g. identify roads and rivers
- Distinguish between different species



- Classify land cover from spectral signature, shape, prior knowledge …

Region properties

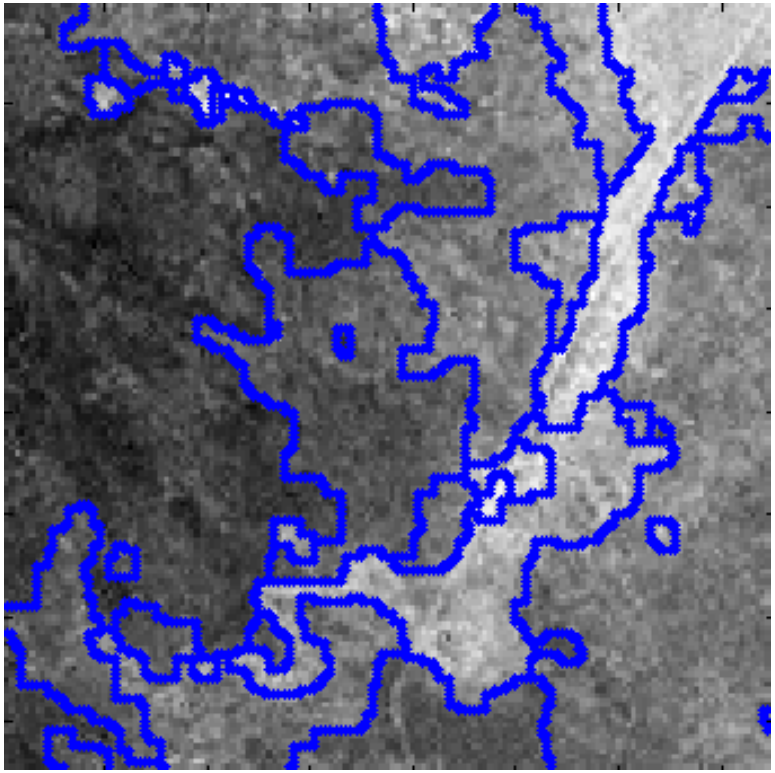
- Classify objects from segmented region properties
- area, perimeter, bounding box, shape, variance, etc..
- Region prop function



- e.g. cells of interest (specific shape)
- Test image of a *Tetrahymena thermophila* culture where identified cells are outlined in yellow and enumerated.

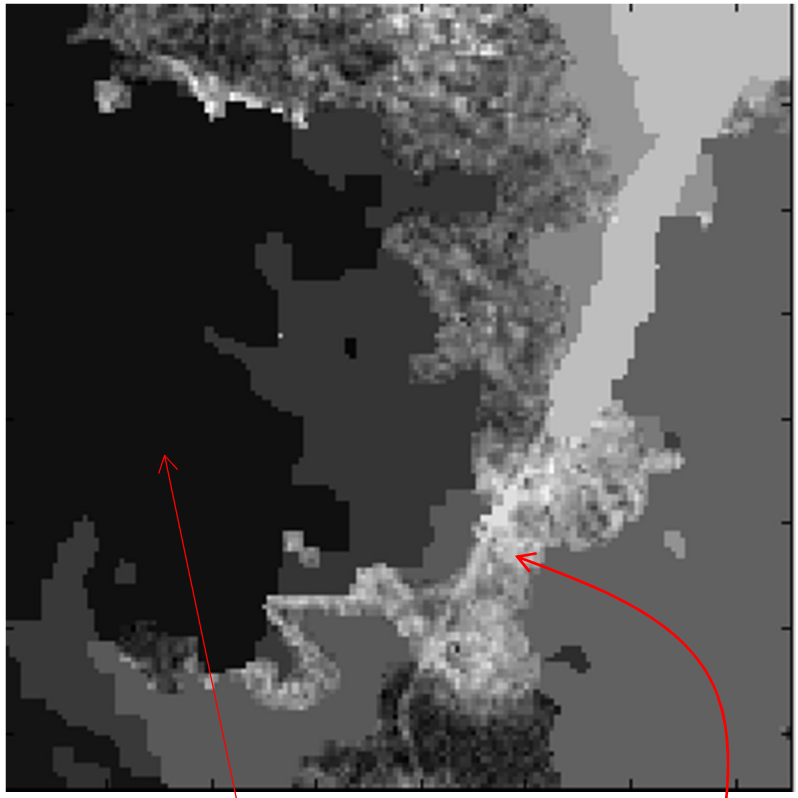
Region properties

- E.g. high variance patches, -> classified as ecotones



WSWF delineation

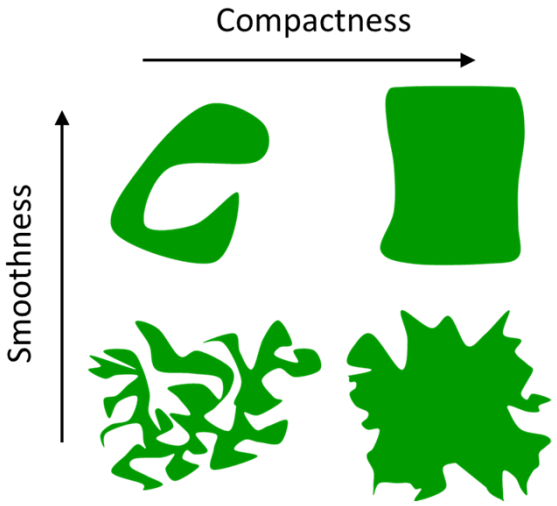
Patch statistics →



Low variance patches: average
High variance patches: grey scale (boundary zones?)

Region properties – shape

- Compactness, smoothness, e.g. forest fragments

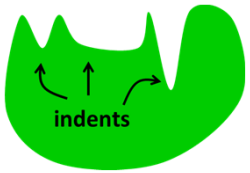


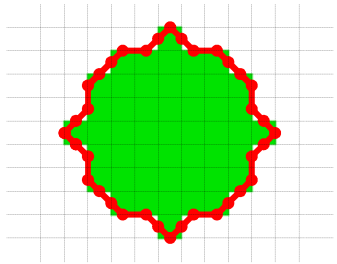
<u>Compact shape:</u>
effective in conserving resources

<u>Convoluted shapes:</u>
effective in enhancing interaction with the surroundings

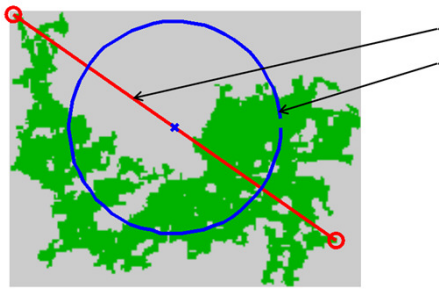
<u>Smooth shapes:</u>
Higher resistance to disturbance

Smoothness:





$$\frac{P - number\ of\ indents \times indent\ length}{P}$$



Longest distance within fragment
Circle of same area

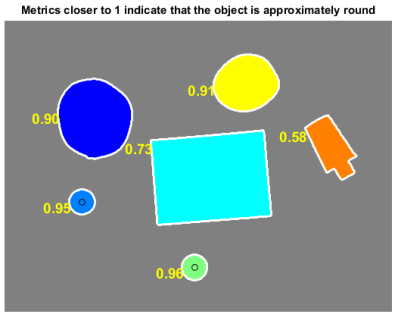
$$\frac{Longest\ length\ in\ fragment}{Equivalent\ circle\ diameter} = \frac{length}{2\sqrt{\frac{A}{\pi}}}$$

Compactness

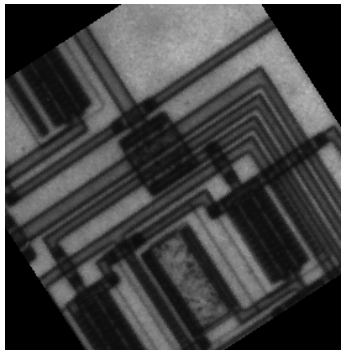
Feature identification

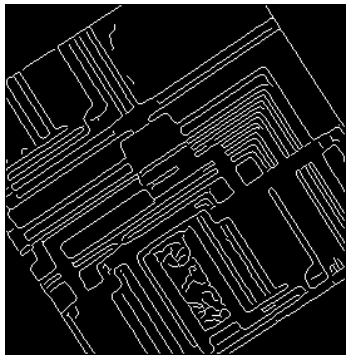
- From texture:
- http://uk.mathworks.com/help/images/examples/texture-segmentation-using-texture-filters.html

- Round and circular objects
- http://uk.mathworks.com/help/images/examples/detect-and-measure-circular-objects-in-an-image.html
- http://uk.mathworks.com/help/images/examples/identifying-round-objects.html



- Lines: http://uk.mathworks.com/help/images/hough-transform.html





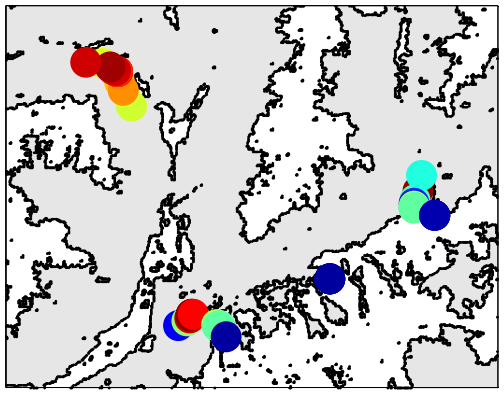
-

Habitat / edge identification

Studying species response to the forest edge:

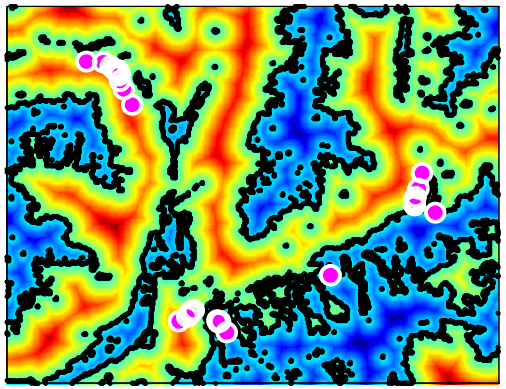
- Distance to nearest edge used as proxy for edge influence

Abundance of species i

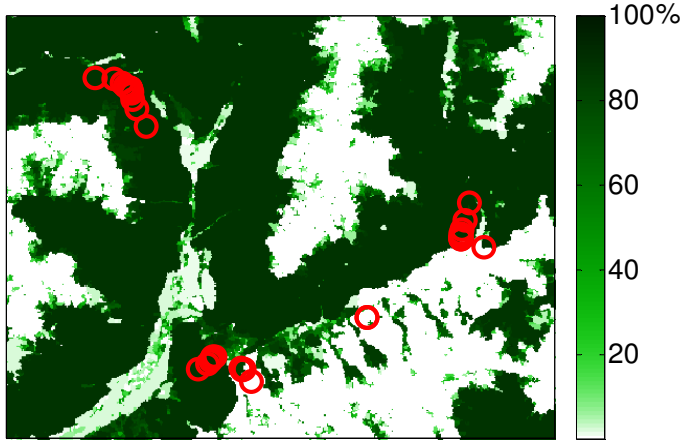


Robert M. Ewers and Raphael K. Didham, The Weta, 2004

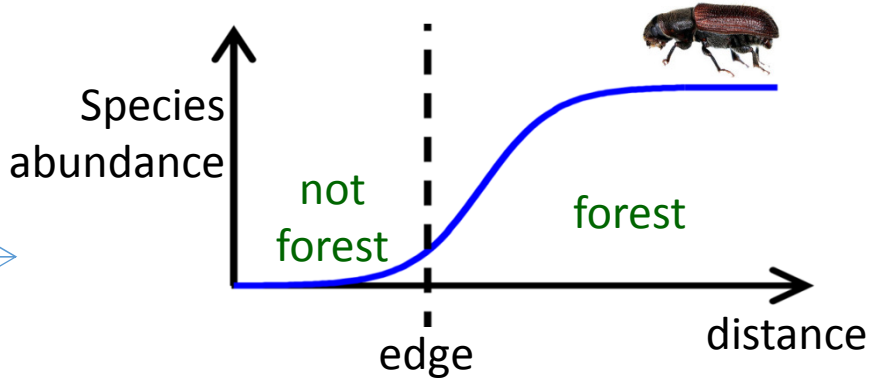
Distance Transform



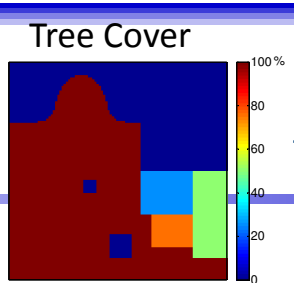
Tree Cover Map



5 km

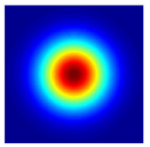


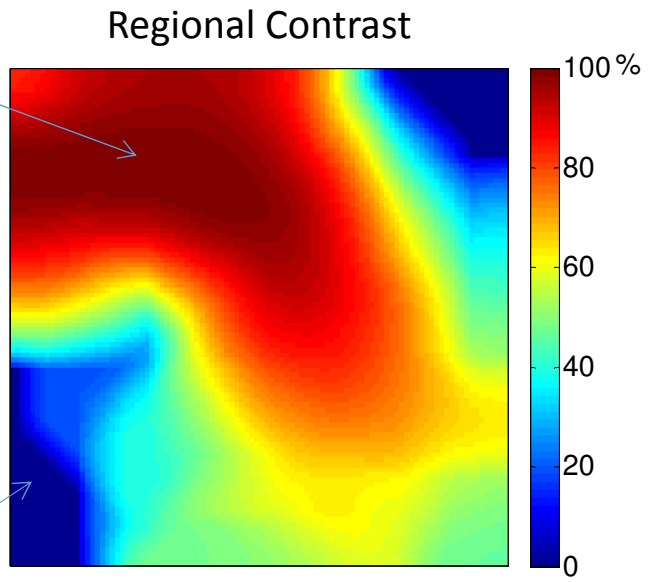
Edge influence / habitat quality identified from contrast and shape in Tree Cover image

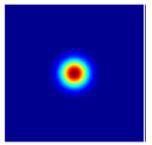


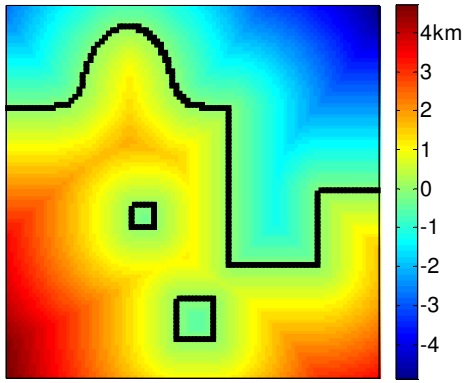
→ Filter radius (DEI) modulated with regional contrast

- smooth the tree cover image
  - to get multiple edge influence
- with local contrast dependent filter
  - To get the depth of edge (DEI) influence to vary with patch contrast

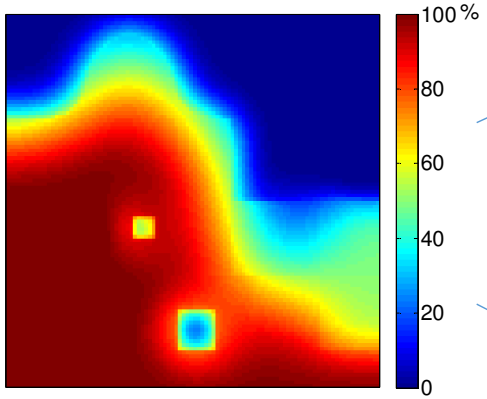


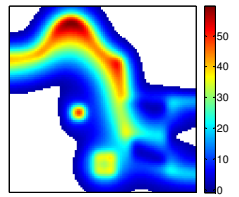


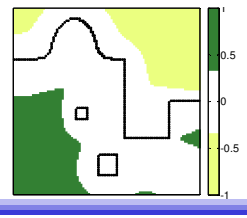




≠

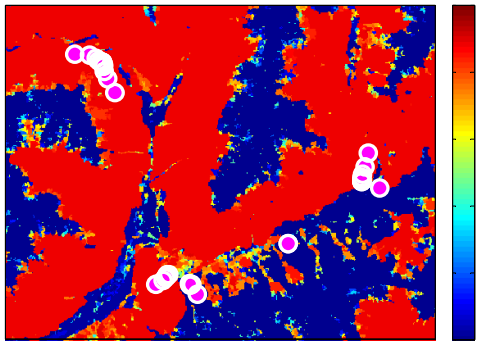




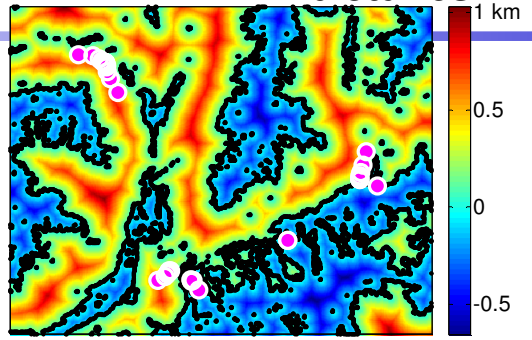


Habitat model results

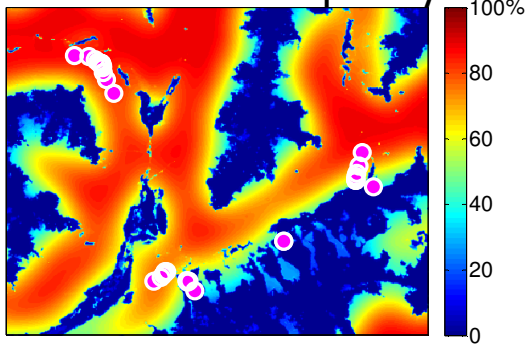
Tree Cover



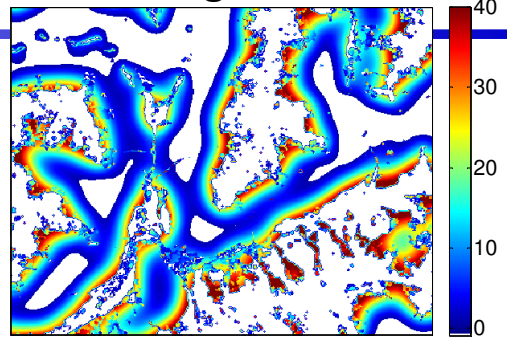
distance



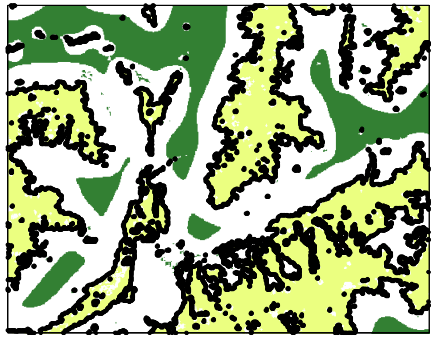
quality



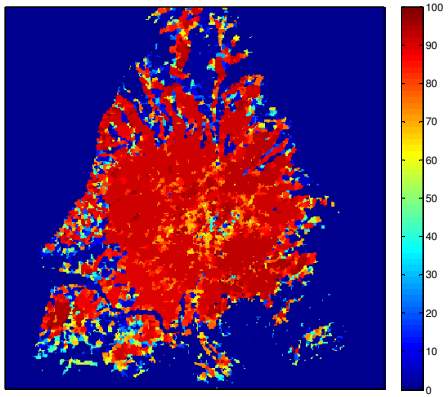
Edge influence



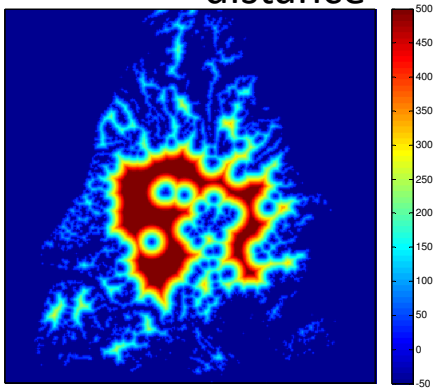
Core area



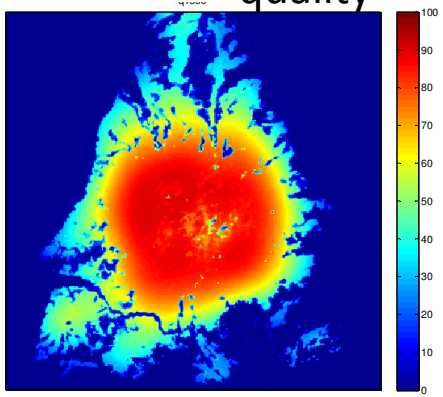
Tree Cover



distance



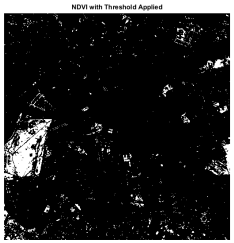
quality



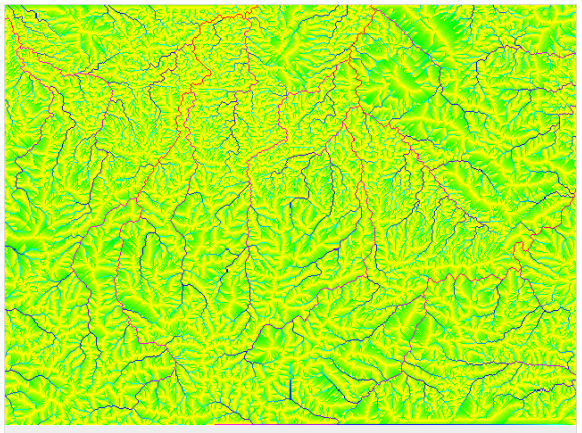
Feature identification

- Land cover classification:

- Finding vegetation
- http://uk.mathworks.com/help/images/examples/finding-vegetation-in-a-multispectral-image.html



- Finding river network:
- http://www.ing.unitn.it/~grass/docs/tutorial_62_en/htdocs/esercitazione/dtm/dtm4.html



-

Feature identification

- Extract local descriptors
- E.g. Local binary patterns (texture classification), gradients

- Machine learning algorithms to classify images
- E.g. kmeans

- Using training dataset/database

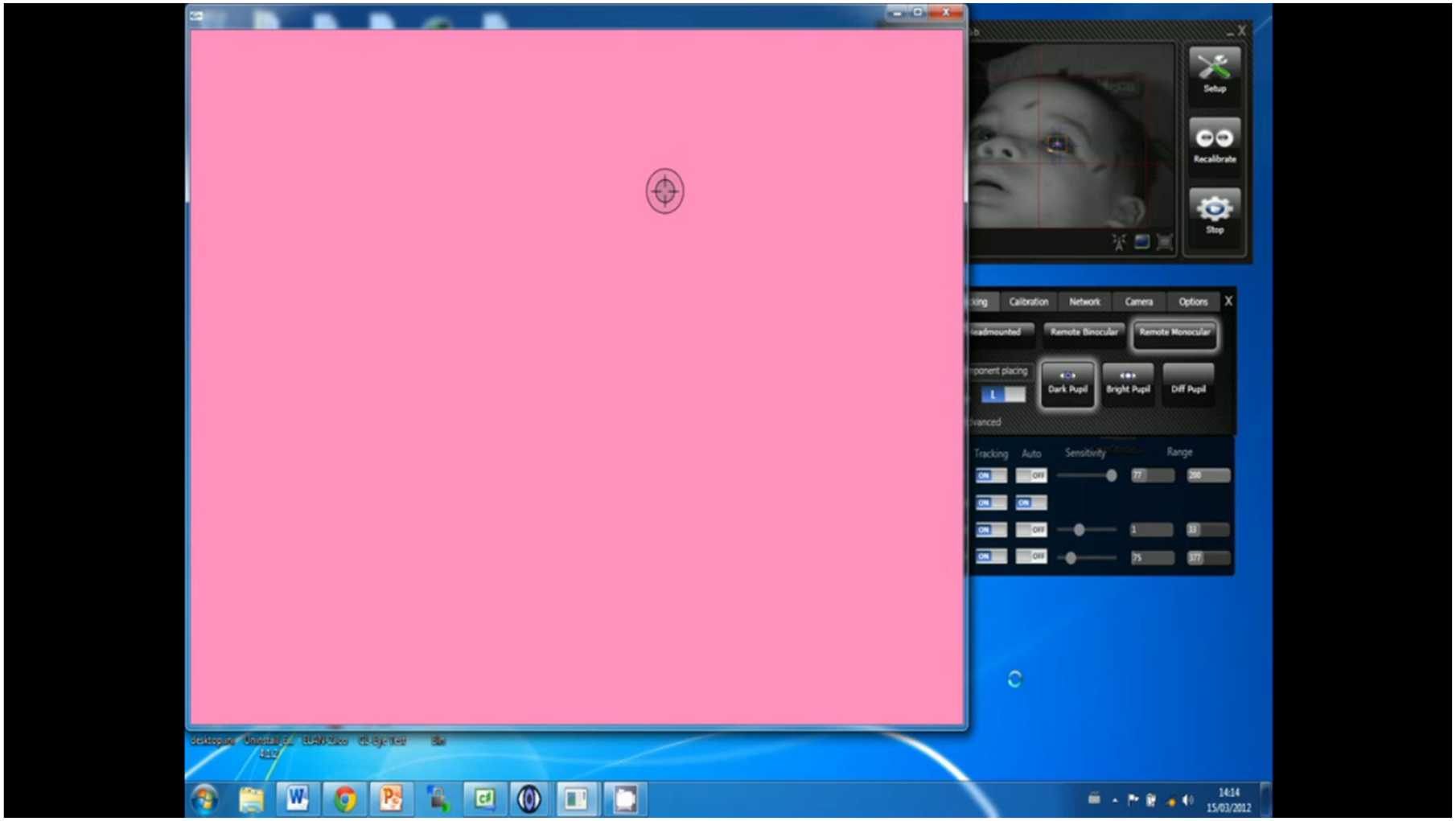
- E.g. can download nose/eye/face classifiers
- Haar cascade

- Face Detection using Haar Cascades: http://opencv-python-tutroals.readthedocs.org/en/latest/py_tutorials/py_objdetect/py_face_detection/py_face_detection.html#face-detection
- Cascade Classifier Training: http://docs.opencv.org/doc/user_guide/ug_traincascade.html

- Bag of features example (image classification):
- http://uk.mathworks.com/help/vision/examples/image-category-classification-using-bag-of-features.html
- Digit classification
- http://uk.mathworks.com/help/vision/examples/digit-classification-using-hog-features.html

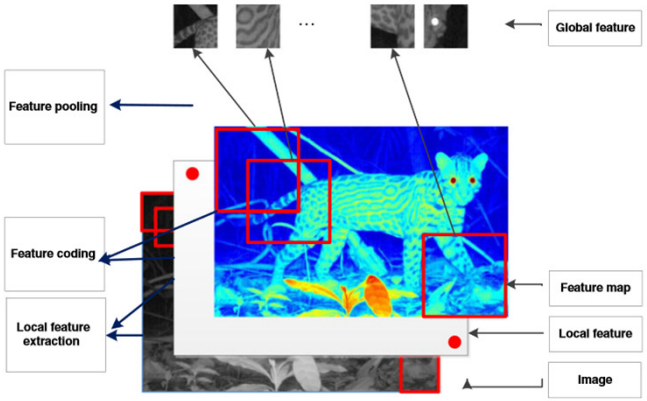
Eye Tracker

- Example: Tracking of eye and pupil in live video to follow baby gaze on screen

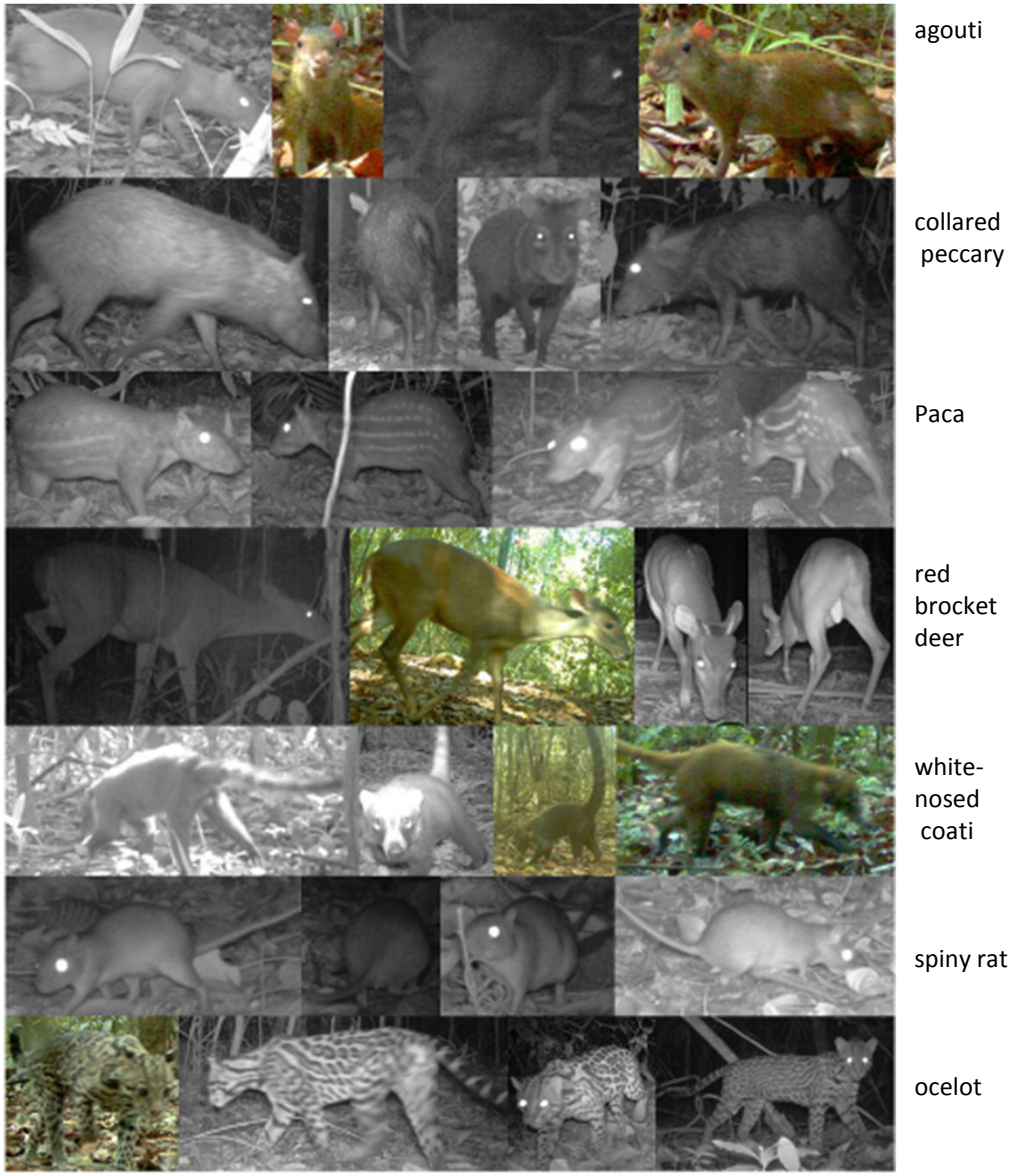


Camera traps – species identification

- Camera traps
- Automatic classification of different species
- Images vary in pose, illumination and scale
- Local features detection:
  - SIFT algorithm (Scale invariant feature transform, also used in panorama stitching)
  - LBP (Local Binary Patterns), texture analysis
- Image classification by SVM (Support Vector Machine)
  - spatial pyramid matching (SPM) kernel



Yu, Xiaoyuan, et al. "Automated identification of animal species in camera trap images." *EURASIP Journal on Image and Video Processing* 2013.1 (2013): 1-10.



Spatial pyramid, butterflies identification

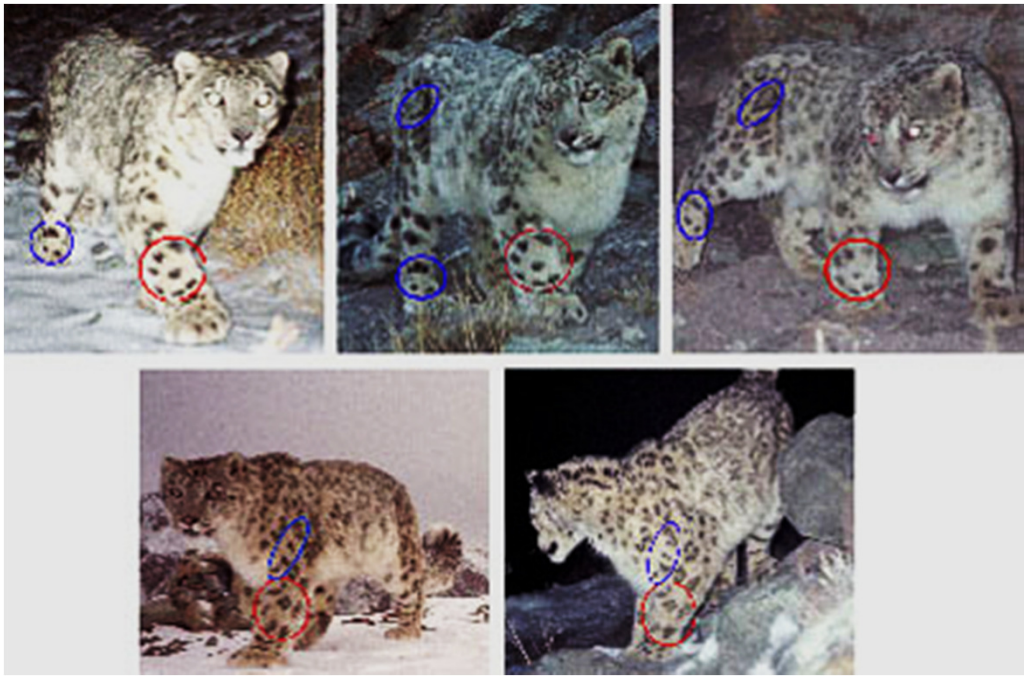
Peacock

Zebra

http://www-cvr.ai.uiuc.edu/ponce_grp/publication/paper/cvpr06b.pdf

Pattern matching – identify individuals

- Identification of snow leopards from pattern matching – not done automatically in this example, but it could be
- matching the leopard's spotting pattern against a photo library of animals whose identity is known from pictures taken earlier in the survey or from previous surveys.
- Identifying different snow leopards is most easily accomplished by comparing the rosette and spot patterns along each flank or side of the body
- http://www.snowleopardconservancy.org/text/conservation/conservation5.htm



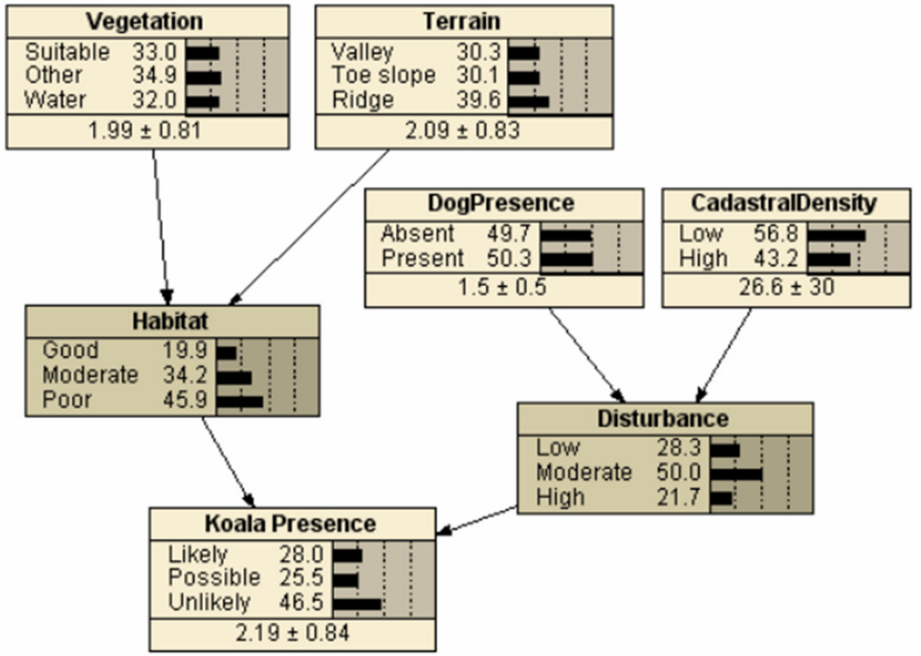
Top row: HNP-1
Bottom row: HP-3

Red lines: primary features
Blue lines: secondary

Identification of koalas from UAV infrared video

Identification of koalas from UAV infrared video

- Monitoring the movements and populations of koalas, dingos, feral pigs and other wildlife
- koalas identified from spectral signature and shape
- **"We use artificial intelligence and machine learning to develop counting and tracking algorithms with results so far showing a significant improvement in detection rates and more accurate population counts."**
- "Bayesian statistical modelling -> making informed decisions under uncertainty and to predict outcomes for koalas and other species"
- + spore traps from the UAV, to get pathogen distribution, plant biosecurity project



https://www.qut.edu.au/news/news?news-id=85595

Methods used - summary

- Enhancement
  - Contrast stretching
  - Neutral balance
  - Filtering:
    - Average, Gaussian, sharpen, edge (sobel, canny), texture, median (denoise), motion, etc..
  - Morphology (dilation, erosion, opening, closing, reconstruction, regional extrema, impose min)
- Geometry / transformations
  - registration
- Extraction
  - Segmentation:
    - K-means
    - Watershed / waterfall
    - Random walker
    - Graph cut
- Analysis / identification
  - Classifiers, SVM

Where to find code / software

- Python packages:
  - Scikit.image
  - openCV
  - SimpleCV
  - Ilastik
  - Mahotas
  - Numpy, Scipy, matplotlib
- ORFEO toolbox, integrates with QGIS: http://www.orfeo-toolbox.org/otb/
- GIMP (scripting possible)
- UAV: Agisoft Photoscan, visual SFM (free)

Bibliography / suggested reading

Articles

- Pennekamp, Frank, and Nicolas Schtickzelle. "Implementing image analysis in laboratory-based experimental systems for ecology and evolution: a hands-on guide." *Methods in Ecology and Evolution* 4.5 (2013): 483-492.

Image processing python tutorials

- Scipy/Numy: http://scipy-lectures.github.io/advanced/image_processing/
- Scikit.image: http://scipy-lectures.github.io/packages/scikit-image/index.html
- Mahotas: http://mahotas.readthedocs.org
- OpenCV:http://opencv-python-tutroals.readthedocs.org/en/latest/py_tutorials/py_imgproc/py_table_of_contents_imgproc/py_table_of_contents_imgproc.html
- http://docs.opencv.org/trunk/doc/py_tutorials/py_tutorials.html
- GRASS: http://grasswiki.osgeo.org/wiki/Image_classification, http://grasswiki.osgeo.org/wiki/Image_processing

Book

- Liu, Jian Guo, and Philippa Mason. *Essential image processing and GIS for remote sensing*. John Wiley & Sons, 2013.

Matlab documentation

- Image processing toolbox: http://uk.mathworks.com/help/images/index.html
  - Examples: http://uk.mathworks.com/help/images/examples.html
- Computer vision toolbox: http://uk.mathworks.com/help/vision/index.html
  - Examples: http://uk.mathworks.com/help/vision/examples.html