Imaging Approaches for Structural Traits in the Lab and Field

Tony Pridmore

University of Nottingham, UK.

BBSRC bioscience for the future

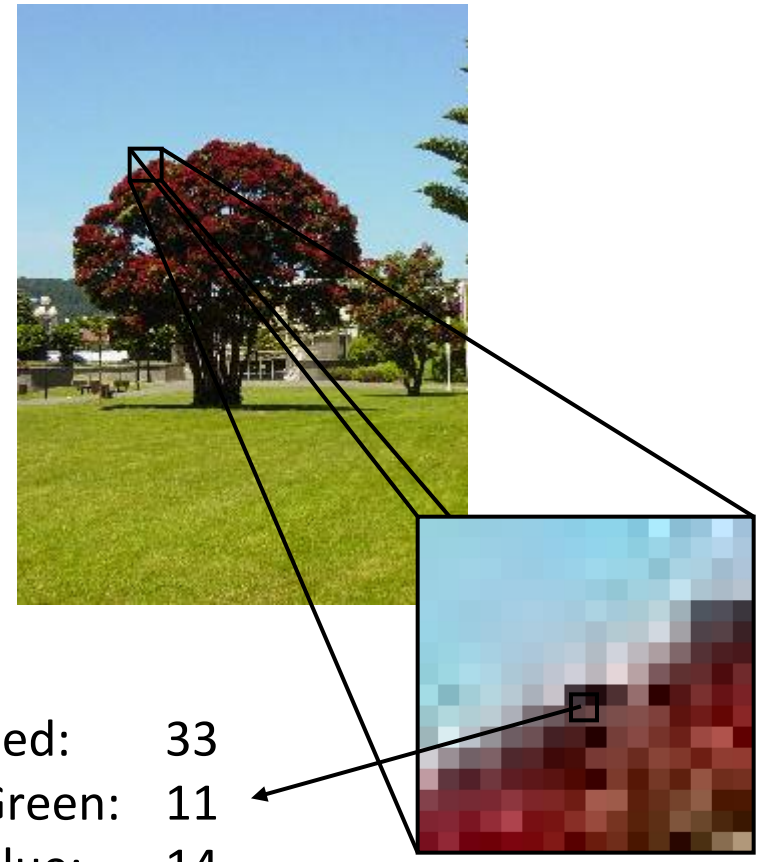
EPSRC Engineering and Physical Sciences Research Council

Digital Images

- Many file formats are in use, with different properties
- Resolution varies with device
- Colour may be represented in different ways

BUT

- All images are rectangular arrays of pixels
- Each pixel contains one (grey) or more (colour) values, e.g. RGB (red, green, blue)

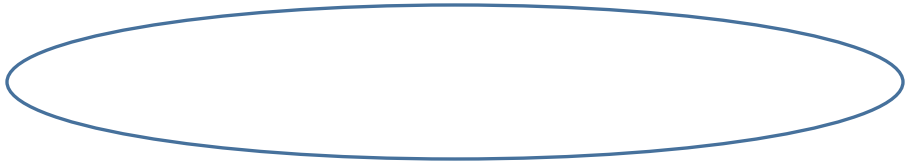


Vision & Computer Vision

- Pixel values are a function of
  - the reflectance and shape of the viewed object
  - illumination conditions
  - viewing geometry
- Biological vision inverts that function
- The goal of computer vision is to
  - understand how
  - build systems that do something similar and useful (e.g. recover plant traits)

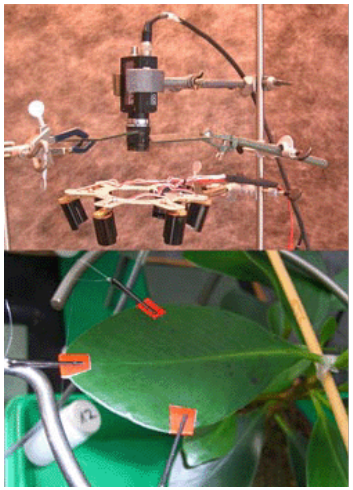
Vision & Computer Vision

- Pixel values are not enough, vision is impossible without prior knowledge or assumptions

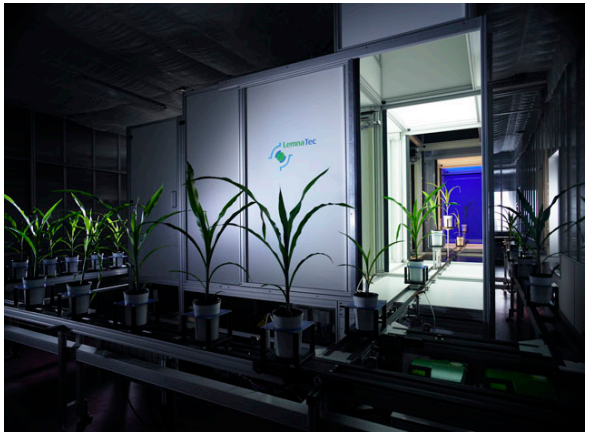


Every computer vision method makes assumptions: if those assumptions are not true, results will become unreliable

Phenotyping: Lab to Field



Phenotyping problems and environments are complex and variable: there is no single solution



Understanding and Designing

❑ To *understand* an image-based phenotyping method you must understand the relationships between

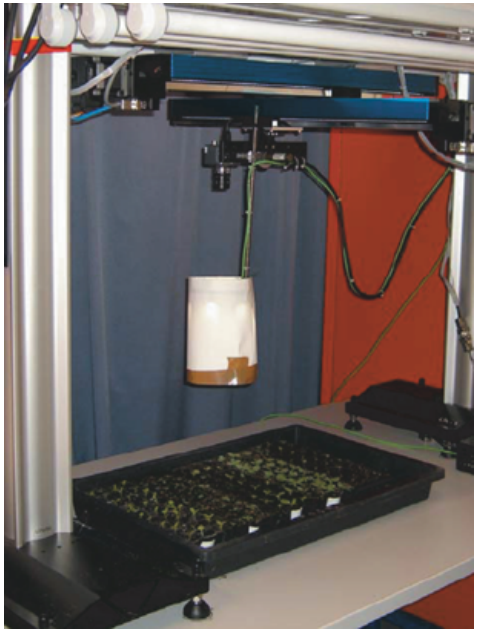
- the **objects** being imaged
- the **imaging environment**
- the **images** acquired

and the **assumptions** made by the image analysis method

❑ When *designing* an image-based phenotyping system you need to ensure that the assumptions used are **consistent** with the objects, imaging environment and images

Lab-based 2D Phenotyping

- Younger plants are often phenotyped under lab conditions.
- Cameras can sometimes be arranged so that the required traits are directly reflected by properties of some image region

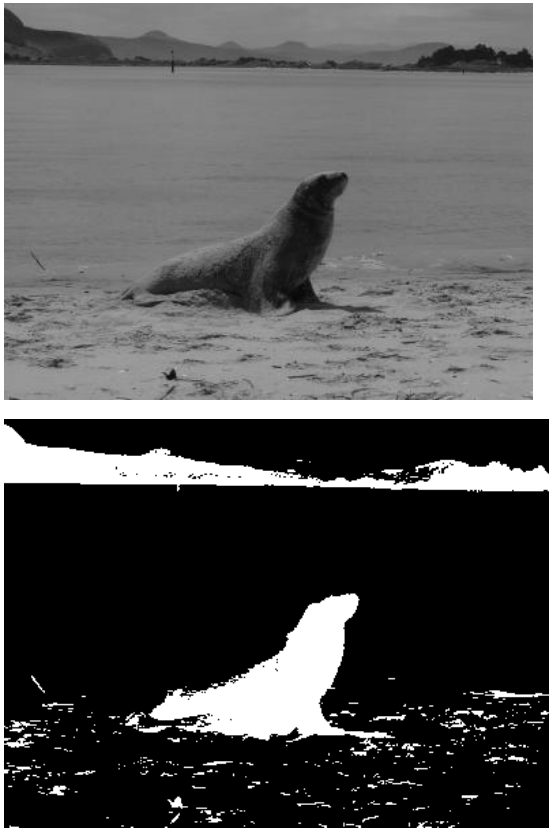


GROWSCREEN, FZ Julich

- Key task is to separate plant from background: *segmentation*

Thresholding: A Simple Method

- A dark object on a light background in a grey-level image
  - Choose a threshold value, T
  - Consider each pixel in turn
  - If the brightness at a pixel is less than T, that pixel is object
  - Otherwise it is part of the background
- Basic idea extends to colour; define sets of colour values that correspond to objects



Threshold, T = 96

Too Simple?

- The value of the threshold is very important
  - If it is too high, background pixels will be classified as foreground
  - If it is too low, object pixels will be considered background
- Assumes there are exactly two regions, with no overlap in their brightness – *is that true?*

$T = 128$

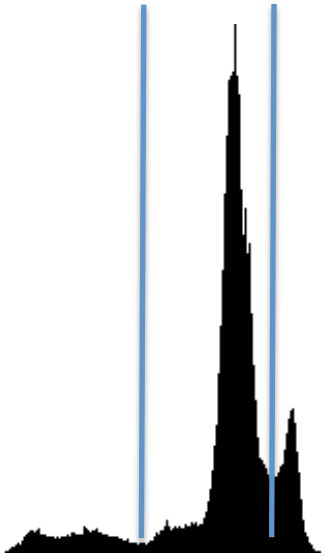
$T = 64$

Image Histograms

- Histograms can also be used to set thresholds automatically

  - set thresholds in the troughs between peaks

  - assumes each peak corresponds to a region, *is that true?*

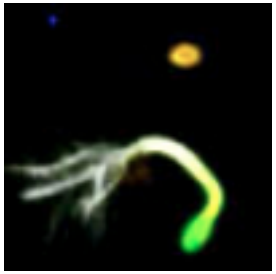


Thresholding

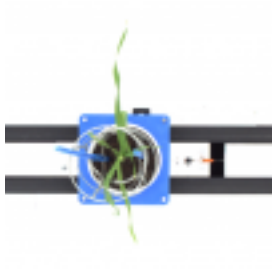
GROWSCREEN, FZ Julich

Lamina, Umea Uni.

Leaver, CIRAD



Germinator, Wageningen



HTP Pheno, Leibniz

Golzarian et al, Plant Methods 2011

http://www.plant-image-analysis.org

❑ Many lab-based systems rely on thresholding. They work because the **environment** is carefully controlled.

Moving away from the Lab

- Glasshouse and field environments are less controllable and so images more complex: assumptions must be less restrictive
- More complex background makes thresholding less accurate
- Assume leaf boundaries are marked by changes in the image:

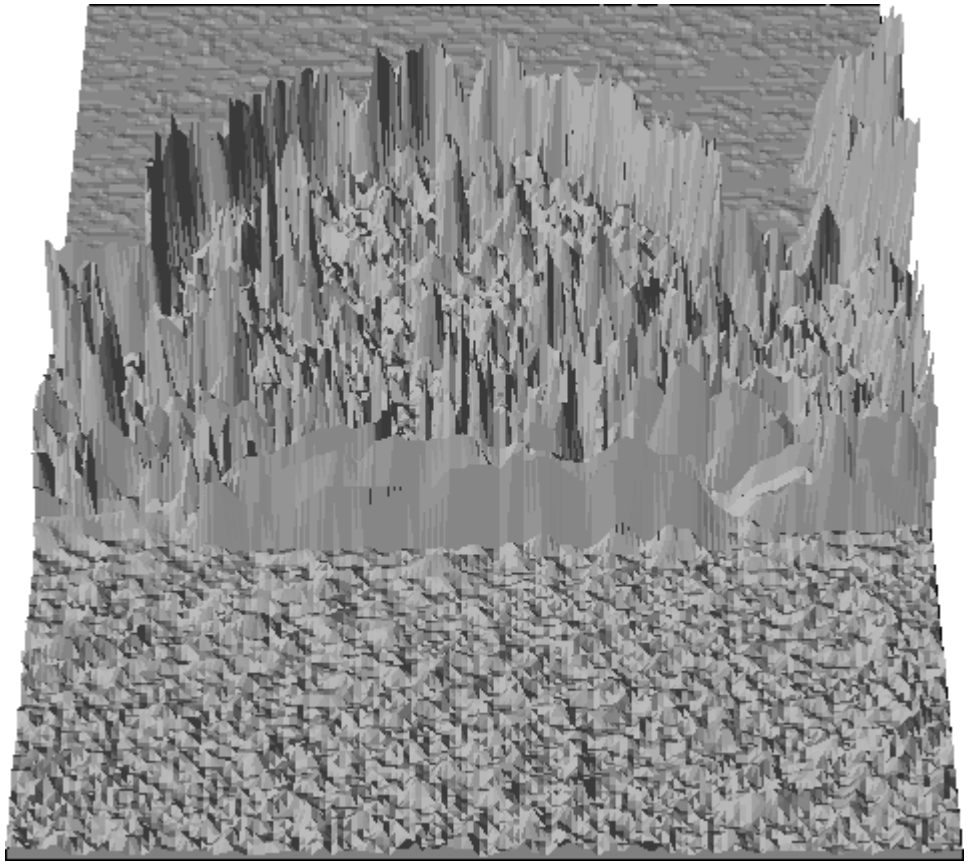
the Watershed algorithm

Images as Terrain: Watersheds

- We start by finding *image gradients*
  - Using e.g. the Sobel operators

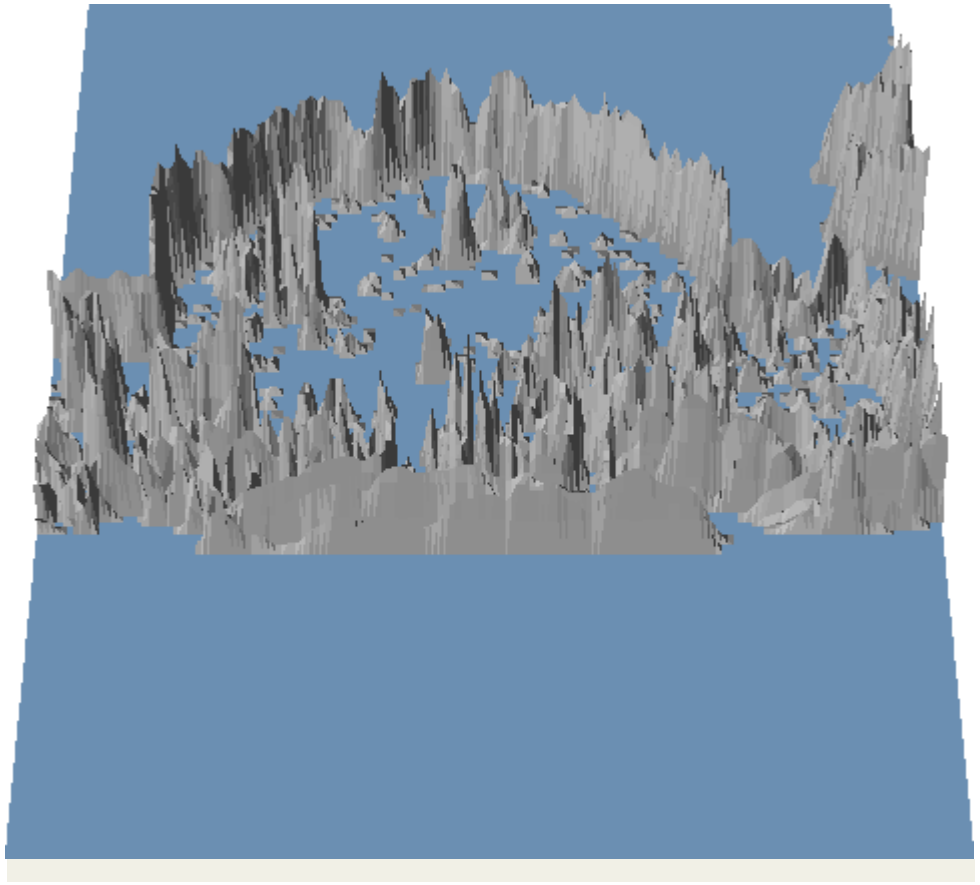
1	2	1
0	0	0
-1	-1	-1

  - This can be viewed as a 3D 'terrain'



Images as Terrain: Watersheds

- We then slowly flood the terrain
  - Flat areas of the image become areas of low gradient, so are valleys in the terrain
  - Edges in the image have high gradient and so are ridges in the terrain

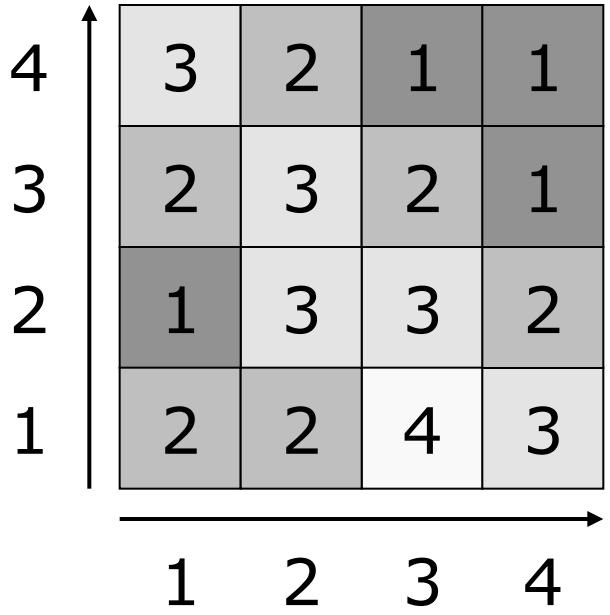


A Watershed Algorithm

- Sort the pixels from low to high
- For each pixel
  - If it's neighbours are all unlabelled, give it a new label
  - If it has neighbours with a single label, it gets that label
  - If it has neighbours with two or more labels, it is a watershed

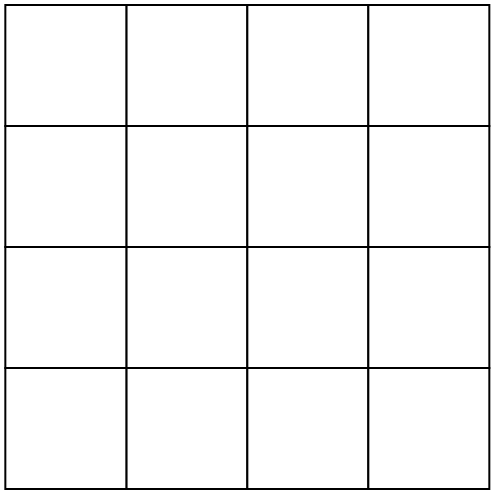
- This is a very basic version
- It can give 'thick' watersheds rather than fine lines
- It is sensitive to noise and so can generate lots of small regions
- It does show the basic plan though

Watersheds Example



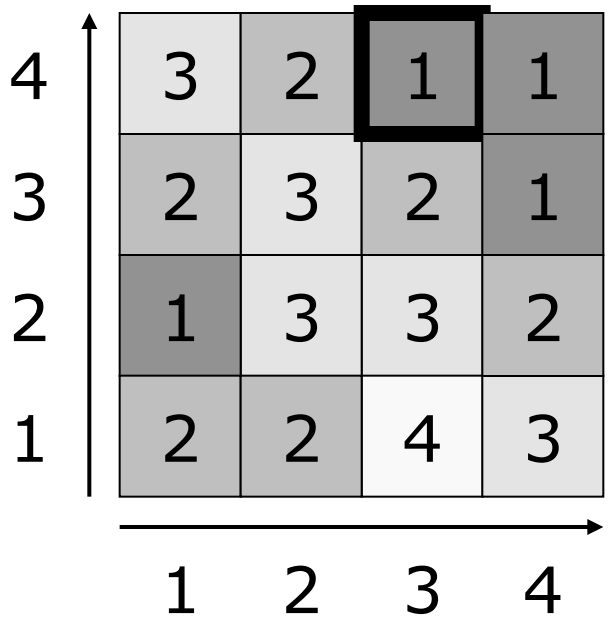
Sorted list:

(3,4) = 1
(4,4) = 1
(4,3) = 1
(1,2) = 1
(2,4) = 2
(1,3) = 2
(3,3) = 2
(4,2) = 2
(1,1) = 2
(2,1) = 2
(1,4) = 3
(2,3) = 3
(2,2) = 3
(3,2) = 3
(4,1) = 3
(3,1) = 4



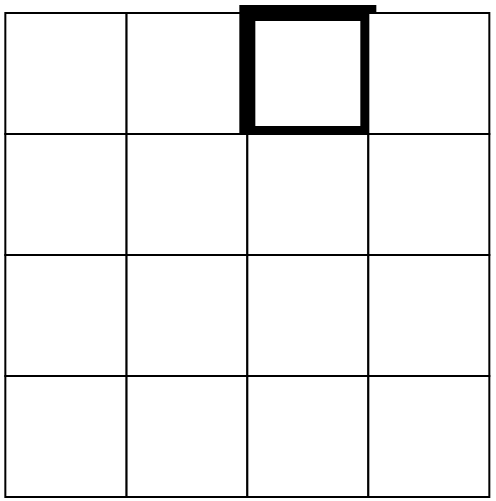
Labels

Watersheds Example



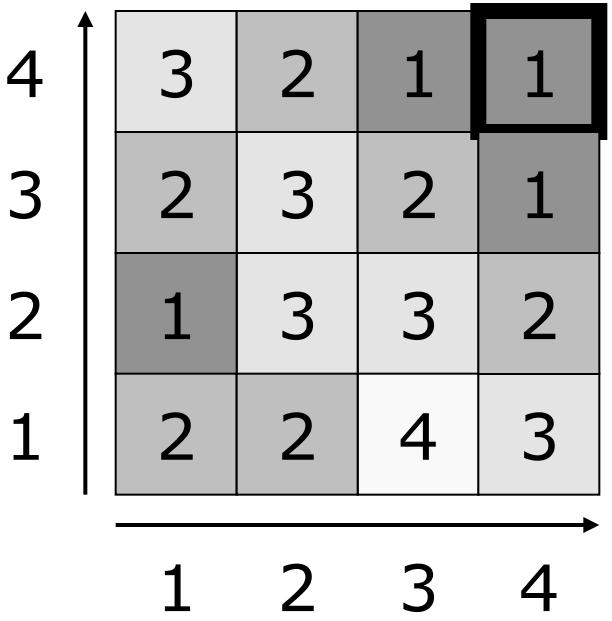
Sorted list:

(3,4) = 1
(4,4) = 1
(4,3) = 1
(1,2) = 1
(2,4) = 2
(1,3) = 2
(3,3) = 2
(4,2) = 2
(1,1) = 2
(2,1) = 2
(1,4) = 3
(2,3) = 3
(2,2) = 3
(3,2) = 3
(4,1) = 3
(3,1) = 4



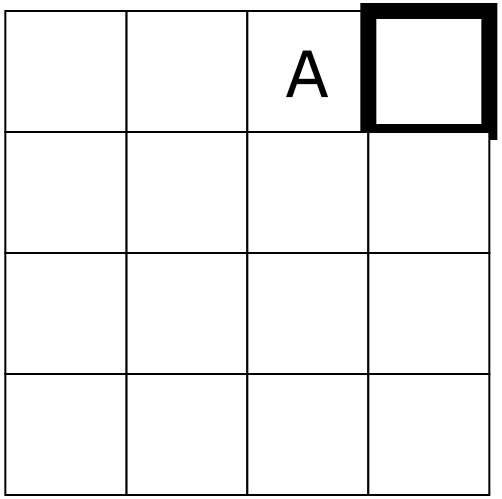
Labels

Watersheds Example



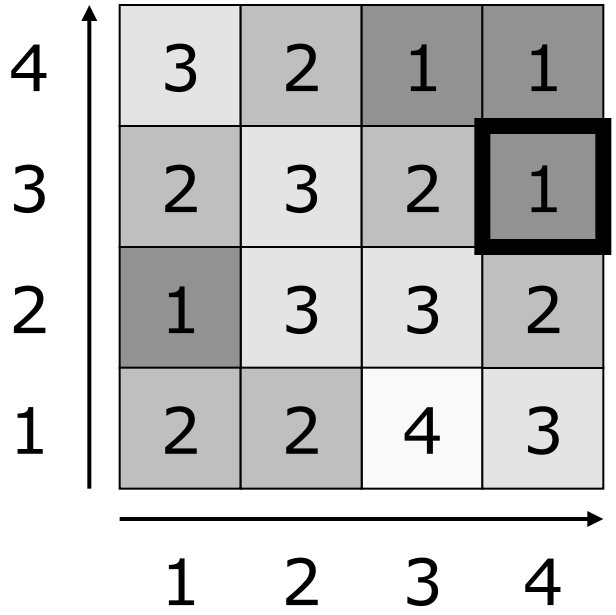
Sorted list:

(3,4) = 1
(4,4) = 1
(4,3) = 1
(1,2) = 1
(2,4) = 2
(1,3) = 2
(3,3) = 2
(4,2) = 2
(1,1) = 2
(2,1) = 2
(1,4) = 3
(2,3) = 3
(2,2) = 3
(3,2) = 3
(4,1) = 3
(3,1) = 4



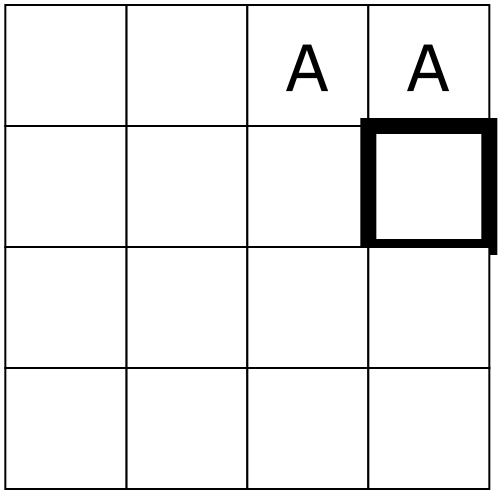
Labels

Watersheds Example



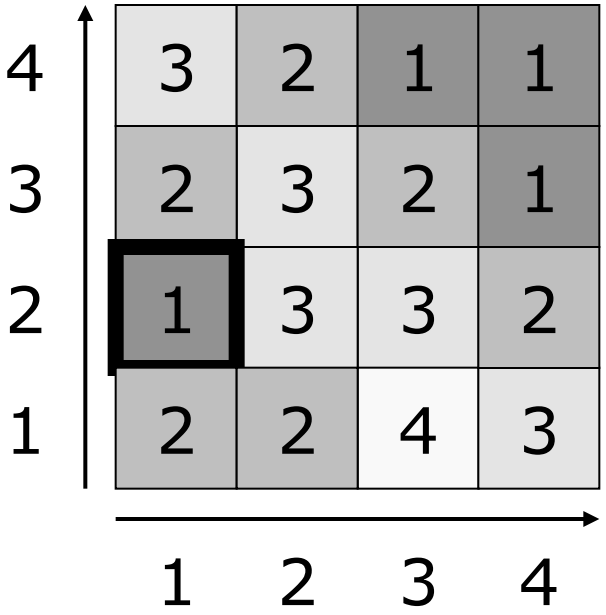
Sorted list:

(3,4) = 1
(4,4) = 1
(4,3) = 1
(1,2) = 1
(2,4) = 2
(1,3) = 2
(3,3) = 2
(4,2) = 2
(1,1) = 2
(2,1) = 2
(1,4) = 3
(2,3) = 3
(2,2) = 3
(3,2) = 3
(4,1) = 3
(3,1) = 4



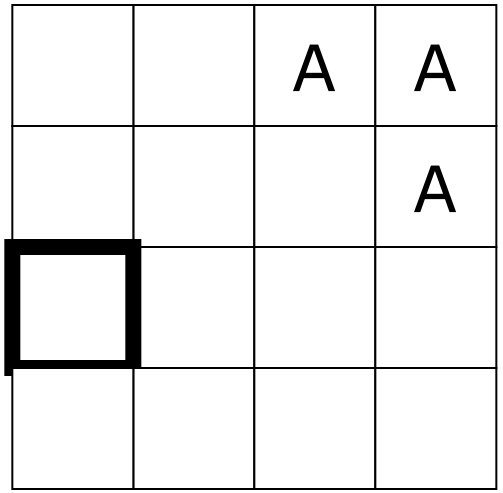
Labels

Watersheds Example



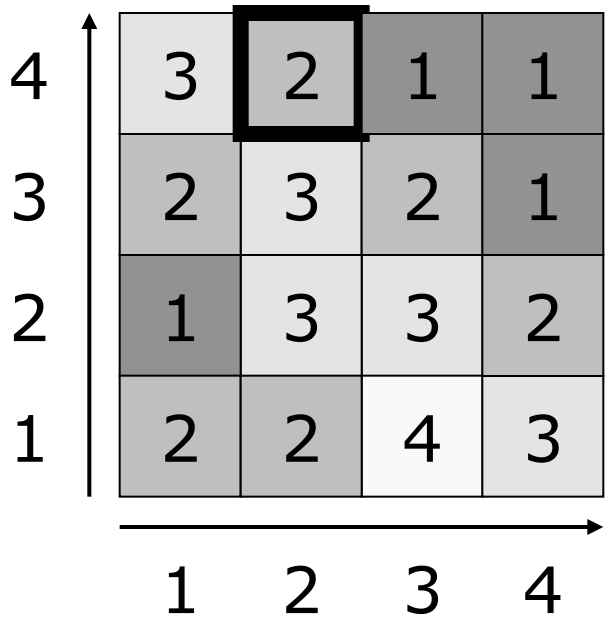
Sorted list:

(3,4) = 1
(4,4) = 1
(4,3) = 1
(1,2) = 1
(2,4) = 2
(1,3) = 2
(3,3) = 2
(4,2) = 2
(1,1) = 2
(2,1) = 2
(1,4) = 3
(2,3) = 3
(2,2) = 3
(3,2) = 3
(4,1) = 3
(3,1) = 4



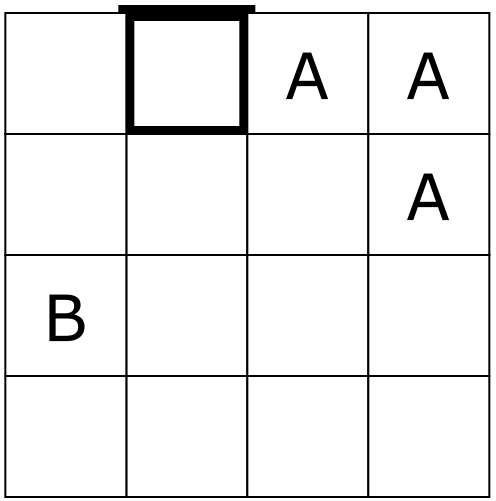
Labels

Watersheds Example



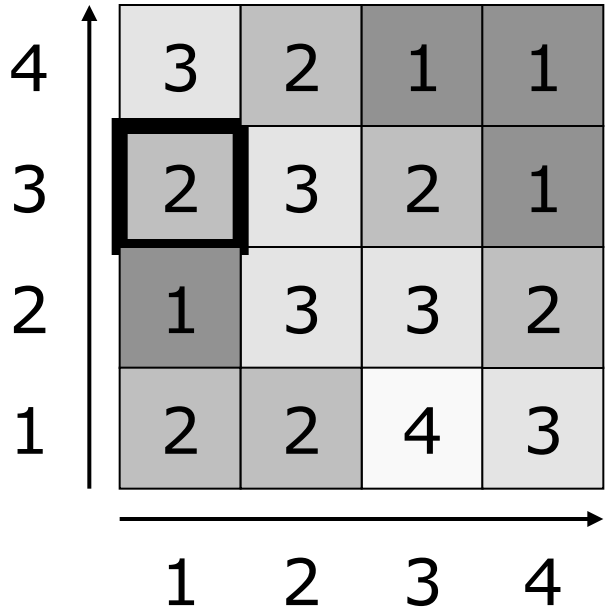
Sorted list:

(3,4) = 1
(4,4) = 1
(4,3) = 1
(1,2) = 1
(2,4) = 2
(1,3) = 2
(3,3) = 2
(4,2) = 2
(1,1) = 2
(2,1) = 2
(1,4) = 3
(2,3) = 3
(2,2) = 3
(3,2) = 3
(4,1) = 3
(3,1) = 4



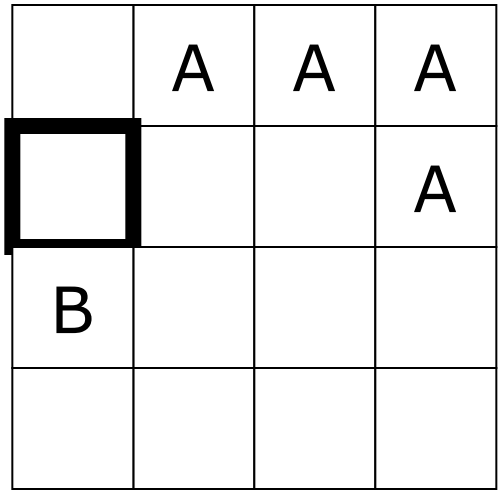
Labels

Watersheds Example



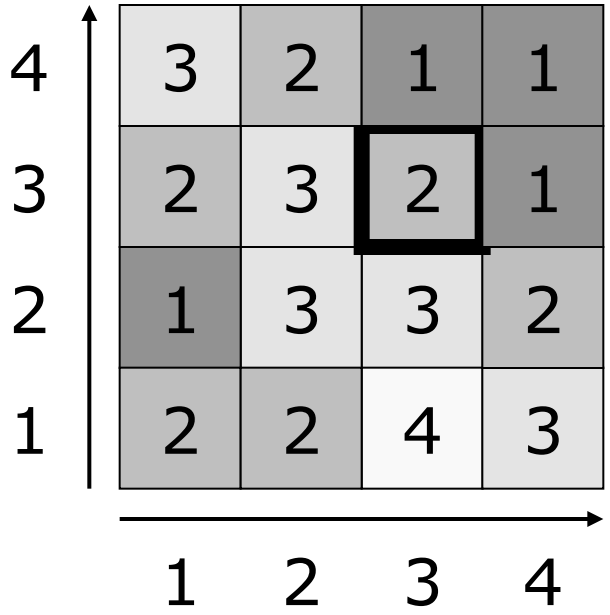
Sorted list:

(3,4) = 1
(4,4) = 1
(4,3) = 1
(1,2) = 1
(2,4) = 2
(1,3) = 2
(3,3) = 2
(4,2) = 2
(1,1) = 2
(2,1) = 2
(1,4) = 3
(2,3) = 3
(2,2) = 3
(3,2) = 3
(4,1) = 3
(3,1) = 4



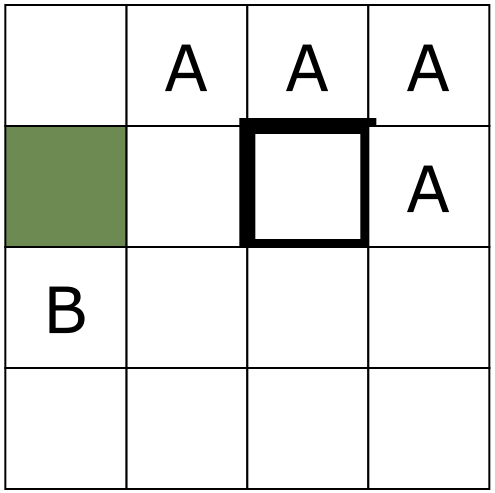
Labels

Watersheds Example



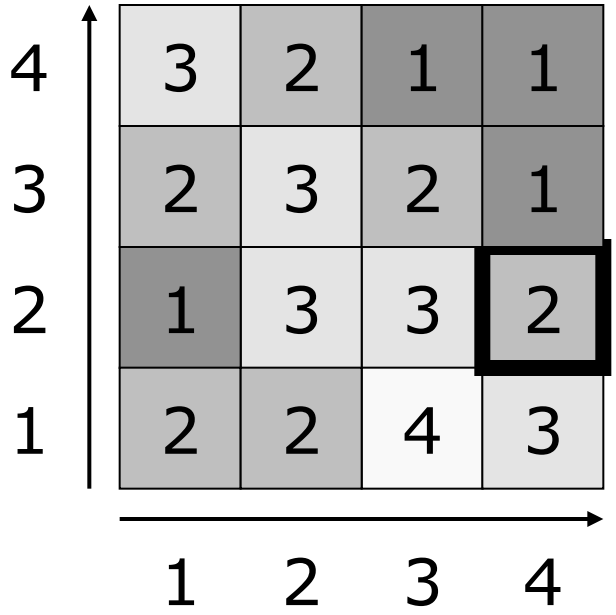
Sorted list:

(3,4) = 1
(4,4) = 1
(4,3) = 1
(1,2) = 1
(2,4) = 2
(1,3) = 2
(3,3) = 2
(4,2) = 2
(1,1) = 2
(2,1) = 2
(1,4) = 3
(2,3) = 3
(2,2) = 3
(3,2) = 3
(4,1) = 3
(3,1) = 4



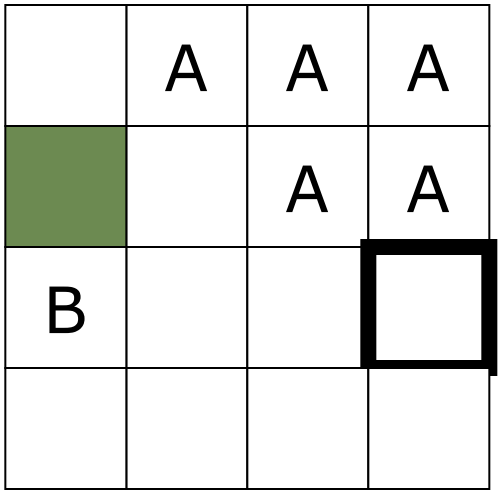
Labels

Watersheds Example



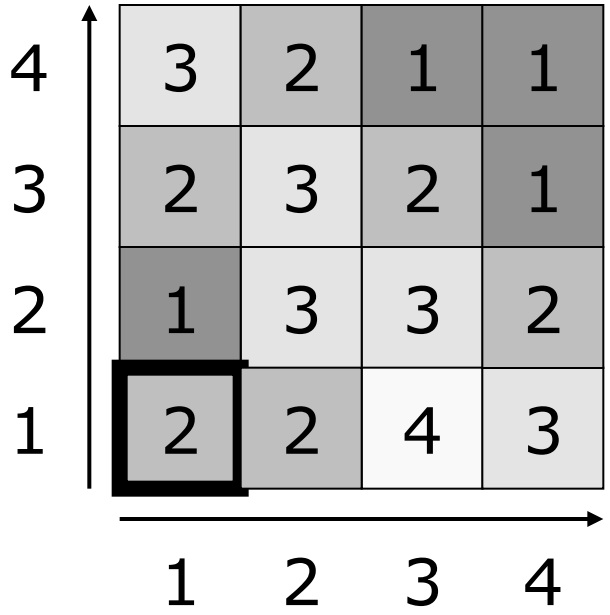
Sorted list:

(3,4) = 1
(4,4) = 1
(4,3) = 1
(1,2) = 1
(2,4) = 2
(1,3) = 2
(3,3) = 2
(4,2) = 2
(1,1) = 2
(2,1) = 2
(1,4) = 3
(2,3) = 3
(2,2) = 3
(3,2) = 3
(4,1) = 3
(3,1) = 4



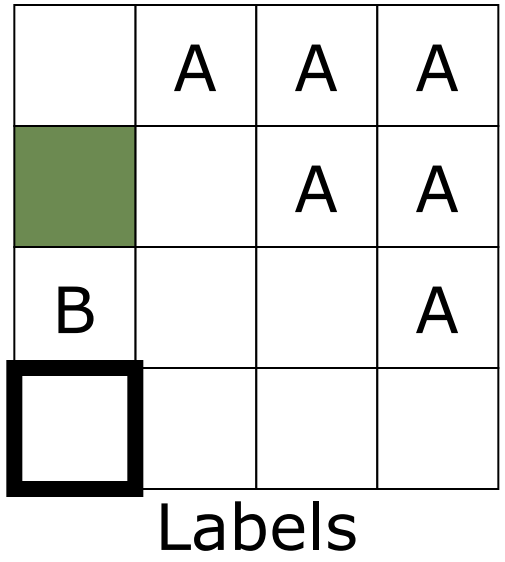
Labels

Watersheds Example



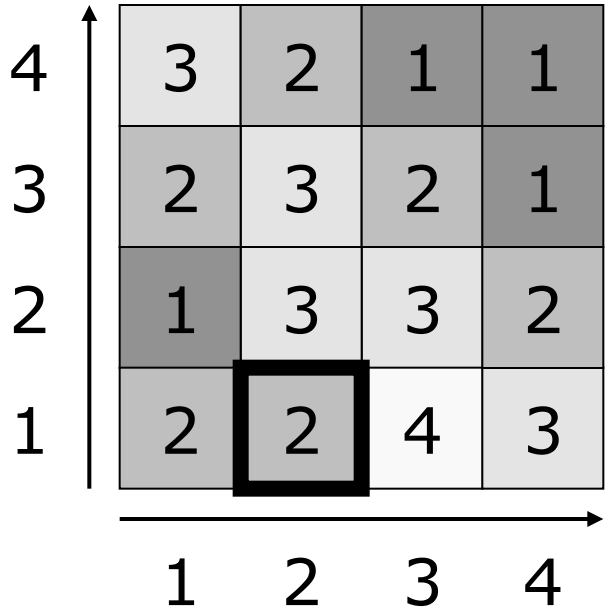
Sorted list:

(3,4) = 1
(4,4) = 1
(4,3) = 1
(1,2) = 1
(2,4) = 2
(1,3) = 2
(3,3) = 2
(4,2) = 2
(1,1) = 2
(2,1) = 2
(1,4) = 3
(2,3) = 3
(2,2) = 3
(3,2) = 3
(4,1) = 3
(3,1) = 4



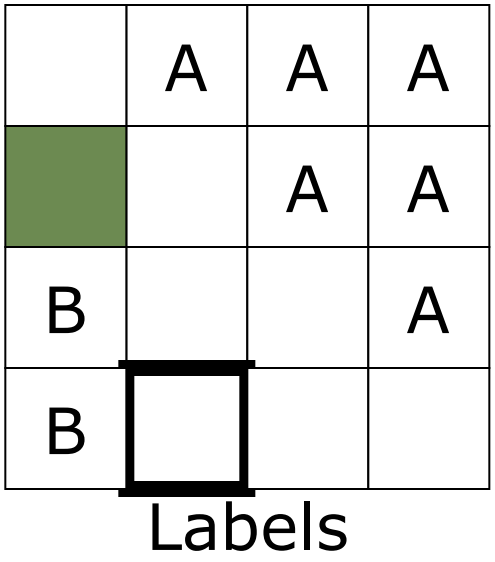
Labels

Watersheds Example



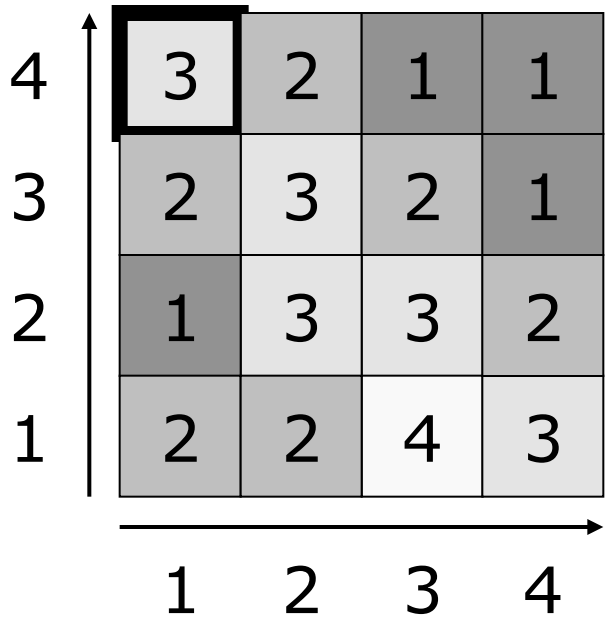
Sorted list:

(3,4) = 1
(4,4) = 1
(4,3) = 1
(1,2) = 1
(2,4) = 2
(1,3) = 2
(3,3) = 2
(4,2) = 2
(1,1) = 2
(2,1) = 2
(1,4) = 3
(2,3) = 3
(2,2) = 3
(3,2) = 3
(4,1) = 3
(3,1) = 4



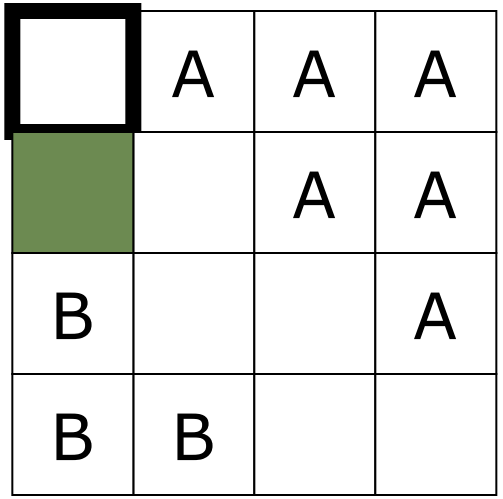
Labels

Watersheds Example



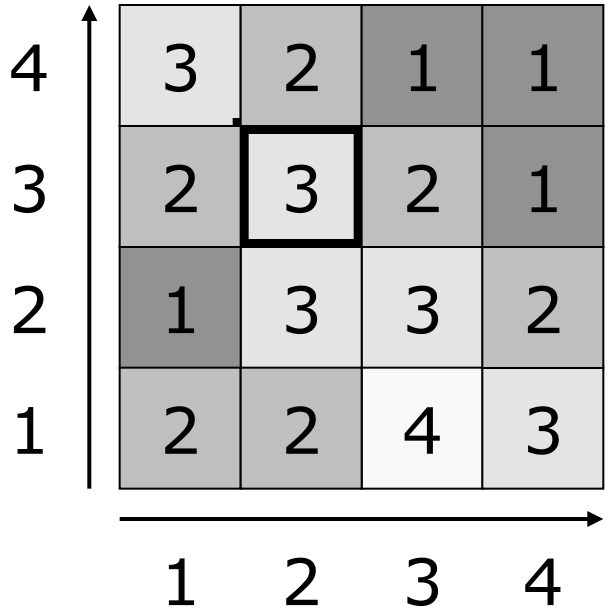
Sorted list:

(3,4) = 1
(4,4) = 1
(4,3) = 1
(1,2) = 1
(2,4) = 2
(1,3) = 2
(3,3) = 2
(4,2) = 2
(1,1) = 2
(2,1) = 2
(1,4) = 3
(2,3) = 3
(2,2) = 3
(3,2) = 3
(4,1) = 3
(3,1) = 4



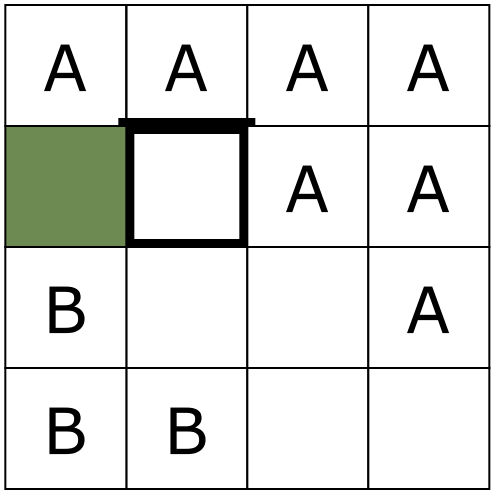
Labels

Watersheds Example



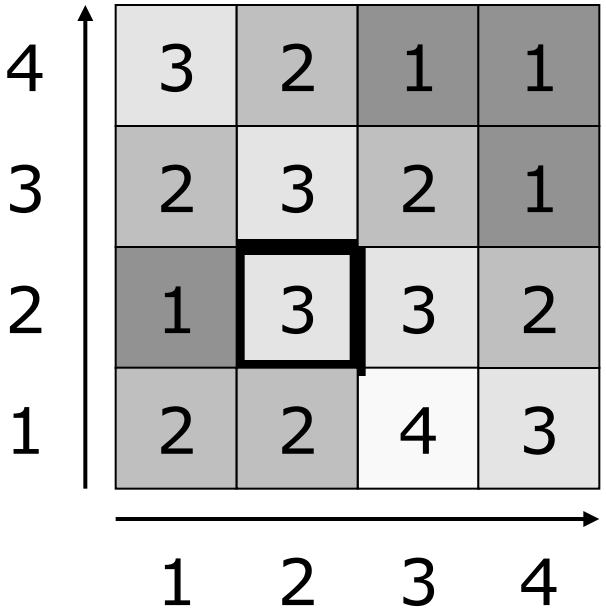
Sorted list:

(3,4) = 1
(4,4) = 1
(4,3) = 1
(1,2) = 1
(2,4) = 2
(1,3) = 2
(3,3) = 2
(4,2) = 2
(1,1) = 2
(2,1) = 2
(1,4) = 3
(2,3) = 3
(2,2) = 3
(3,2) = 3
(4,1) = 3
(3,1) = 4



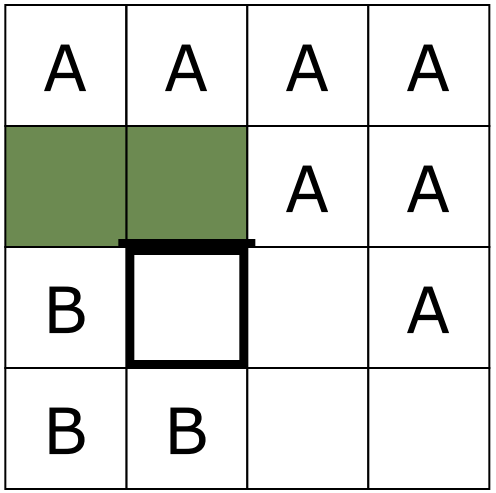
Labels

Watersheds Example



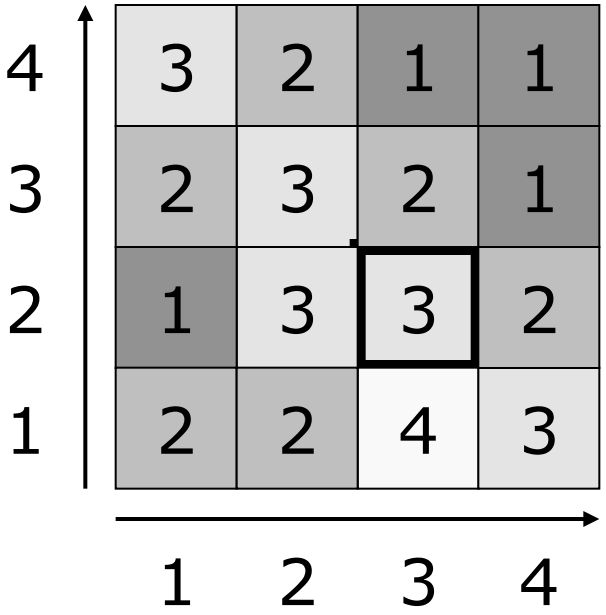
Sorted list:

(3,4) = 1
(4,4) = 1
(4,3) = 1
(1,2) = 1
(2,4) = 2
(1,3) = 2
(3,3) = 2
(4,2) = 2
(1,1) = 2
(2,1) = 2
(1,4) = 3
(2,3) = 3
(2,2) = 3
(3,2) = 3
(4,1) = 3
(3,1) = 4



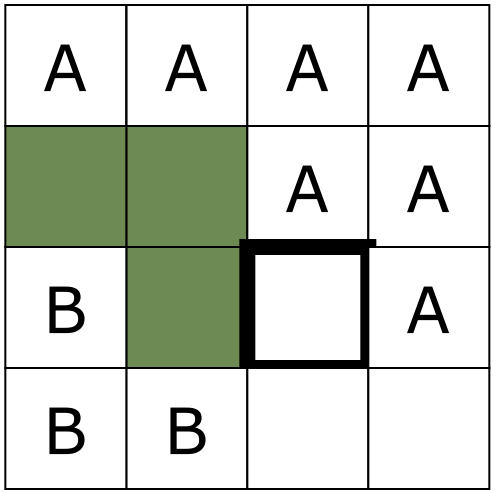
Labels

Watersheds Example



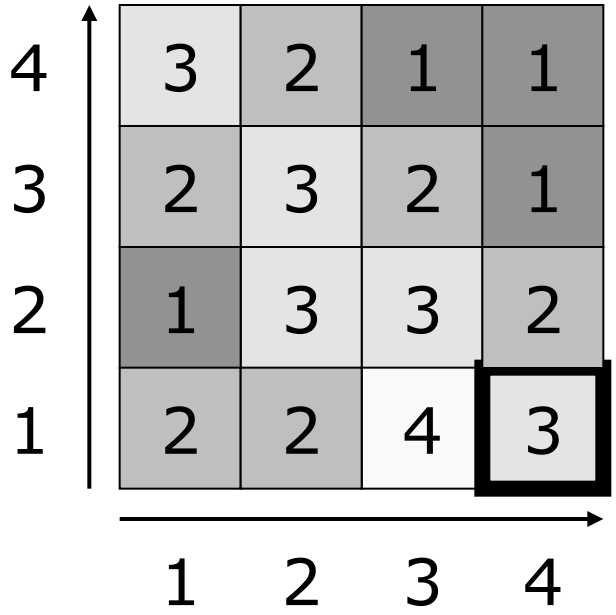
Sorted list:

(3,4) = 1
(4,4) = 1
(4,3) = 1
(1,2) = 1
(2,4) = 2
(1,3) = 2
(3,3) = 2
(4,2) = 2
(1,1) = 2
(2,1) = 2
(1,4) = 3
(2,3) = 3
(2,2) = 3
(3,2) = 3
(4,1) = 3
(3,1) = 4



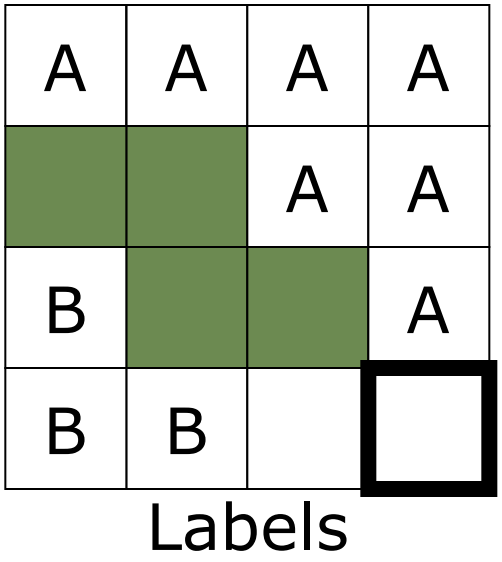
Labels

Watersheds Example



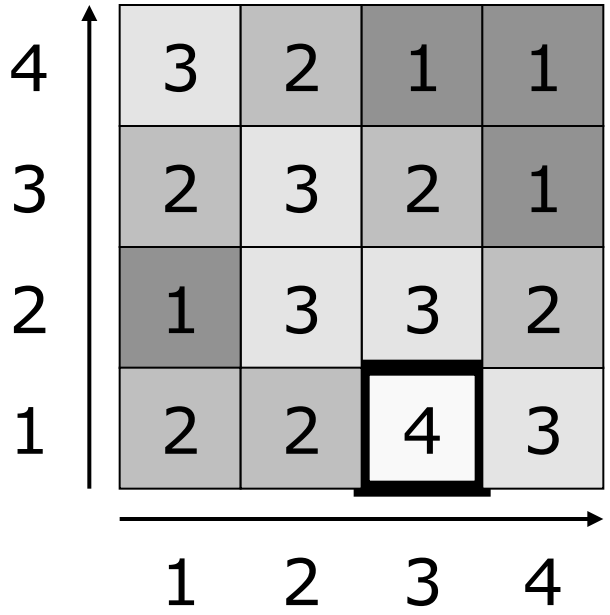
Sorted list:

(3,4) = 1
(4,4) = 1
(4,3) = 1
(1,2) = 1
(2,4) = 2
(1,3) = 2
(3,3) = 2
(4,2) = 2
(1,1) = 2
(2,1) = 2
(1,4) = 3
(2,3) = 3
(2,2) = 3
(3,2) = 3
(4,1) = 3
(3,1) = 4



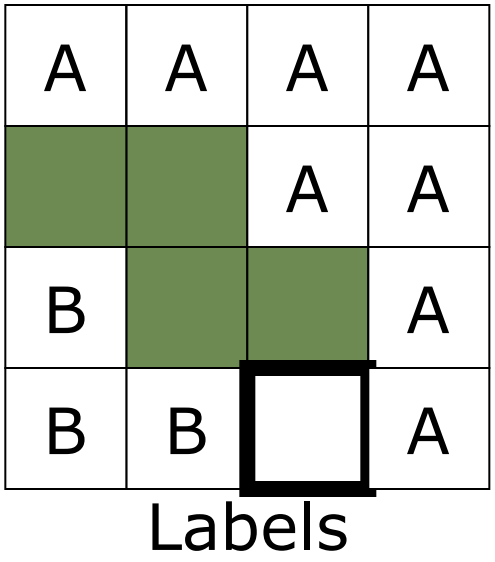
Labels

Watersheds Example



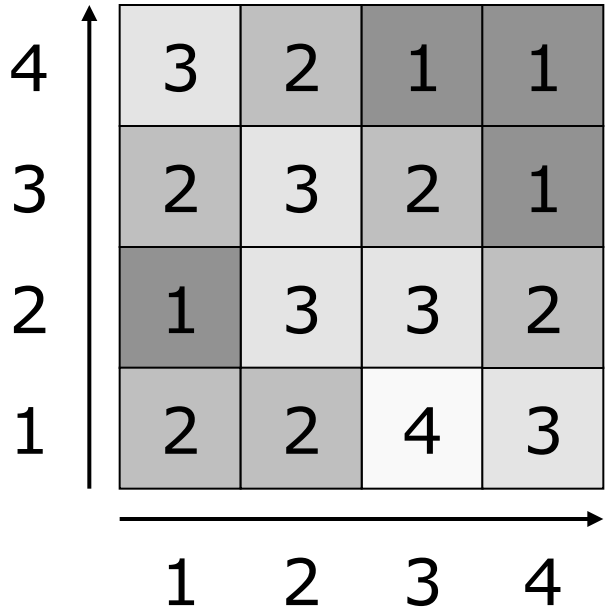
Sorted list:

(3,4) = 1
(4,4) = 1
(4,3) = 1
(1,2) = 1
(2,4) = 2
(1,3) = 2
(3,3) = 2
(4,2) = 2
(1,1) = 2
(2,1) = 2
(1,4) = 3
(2,3) = 3
(2,2) = 3
(3,2) = 3
(4,1) = 3
(3,1) = 4



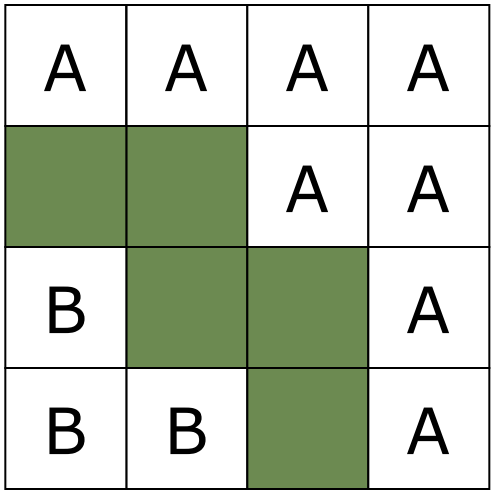
Labels

Watersheds Example



Sorted list:

(3,4) = 1
(4,4) = 1
(4,3) = 1
(1,2) = 1
(2,4) = 2
(1,3) = 2
(3,3) = 2
(4,2) = 2
(1,1) = 2
(2,1) = 2
(1,4) = 3
(2,3) = 3
(2,2) = 3
(3,2) = 3
(4,1) = 3
(3,1) = 4



Labels

Moving away from the Lab

- Flooding can be initialised in many ways, e.g. from approximate regions found by thresholding (Wang et al, 2005)

- Watersheds' assumption is quite often true, so the approach is quite widely applicable

3D Measurements

- Silhouette intersection or *Space Carving*

  - take multiple images from different views

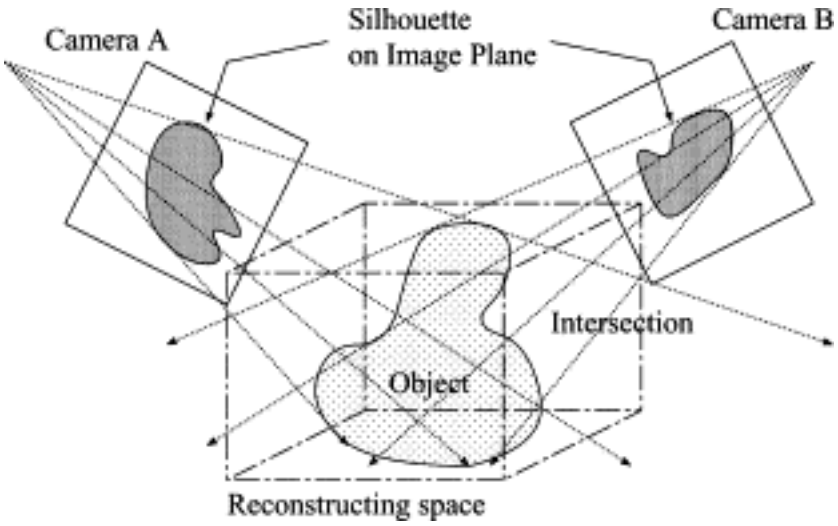
  - segment the object (plant) from the background to make a set of silhouettes

  - project each silhouette out into space to approximate the object shape (e.g. a circular silhouette will produce a cone)

  - intersect all the projections

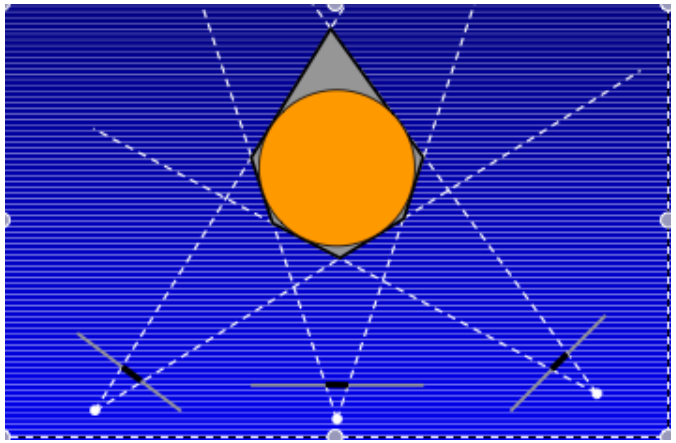
  - if there are enough images, the intersection will **closely approximate** the object

  - Many variations on the theme exist

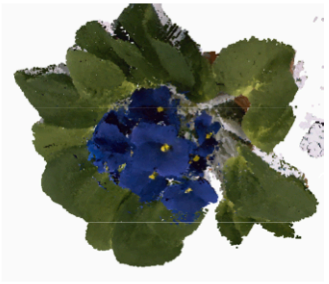


3D Measurements

- Silhouette intersection or *Space Carving*



Input Image (1 of 45)



Reconstruction

Reconstruction

Reconstruction

Source: S. Seitz

- can use multiple cameras, but a common approach is to rotate the plant in front of a single fixed camera

3D Measurements

- Binocular or Multiview Stereo

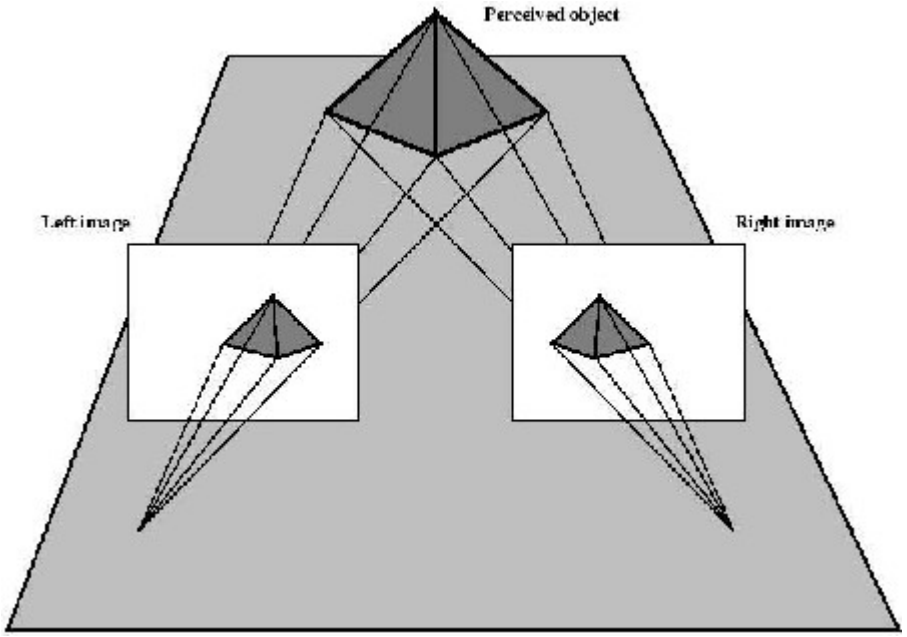
  - again take multiple images from different viewpoints

  - extract features from each image independently

  - match features between images

  - recover 3D position by triangulation

  - result is a 3D point cloud



3D Measurements

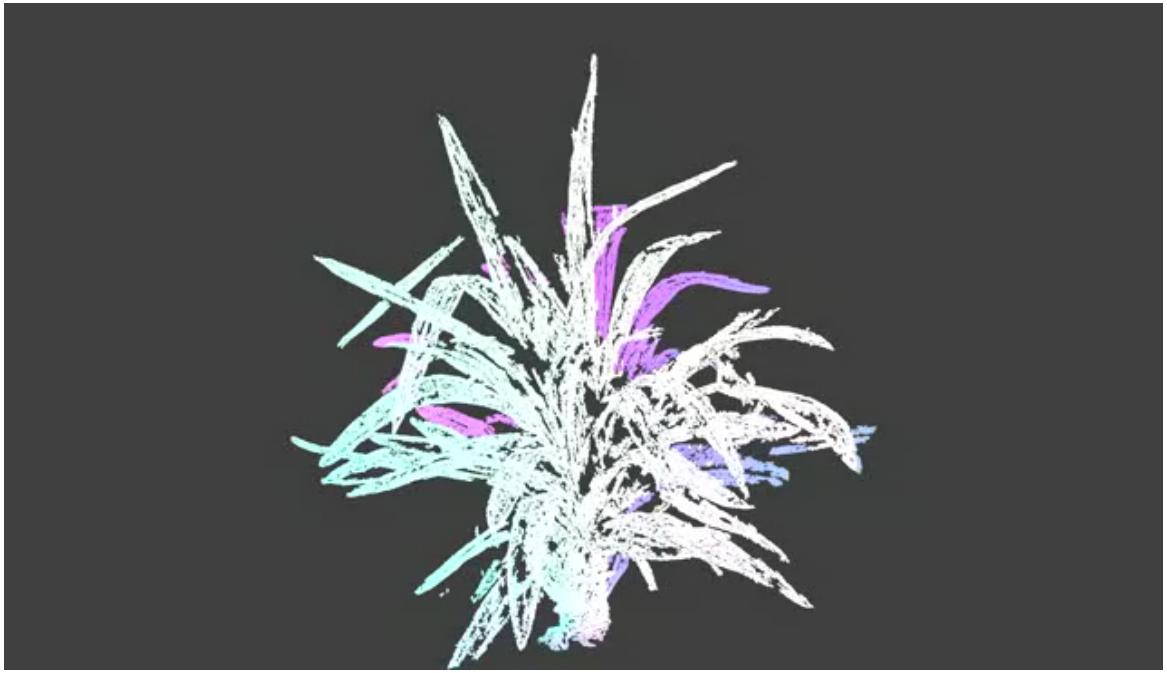
- SPICY project (Wageningen): glasshouse phenotyping of pepper plants

Colour (L) Colour (R) ToF Stereo+ToF

High quality depth map makes segmentation e.g. of leaves easier

3D Measurements

- Nottingham is developing multi-view stereo methods to reconstruct plant canopies for use in photosynthesis modelling



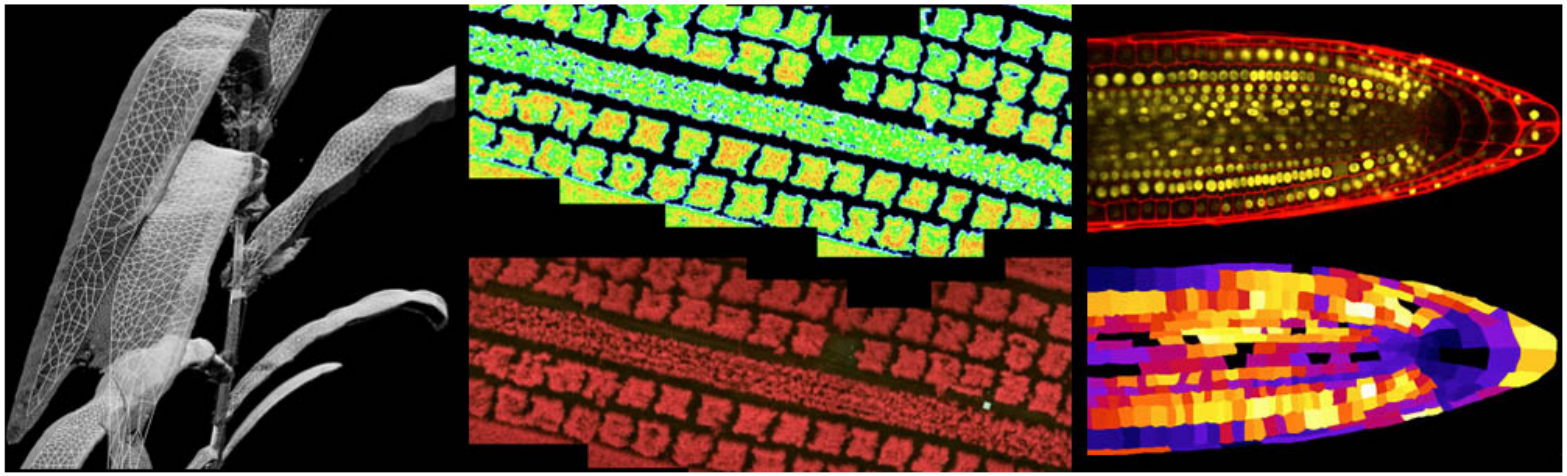
Field Phenotyping

- A Work in Progress: environment is unconstrained, hard to identify methods and assumptions that will allow recovery of plant structure
- Some work on stereo methods, some on segmentation

Conclusion

- Image-based plant phenotyping requires **prior knowledge or assumptions**, like any other computer vision problem
- Lab-based systems are better developed, because its easier to find assumptions that match the task when you can **control the environment**
- Some image analysis operations (segmentation, space carving, stereo) are commonly used, but there are **many variants** on each
- Field phenotyping requires significant further development

International Workshop on Image Analysis Methods for the Plant Sciences



2nd-3rd September 2013, University of Nottingham, Jubilee campus, UK