Segmentation I

Claus.Brenner@ikg.uni-hannover.de

Leibniz Universität Hannover

Segmentation

- Segmentation = elements (e.g. points in scan data, pixels in images) are **grouped according to a homogeneity criterion**
- **Example: in images**
  Connected regions in an image, which have a similar color, e.g.
  - Faces or hands in a RGB image
  - Parcel, street, building, vegetation in a multispectral image
- **Example: in a laser scan**
  Neighboring points, which
  - Are in the same plane
  - Which are on the surface of a geometric body, e.g. a sphere or a cylinder
  - Which exhibit a certain local distribution (“texture”), e.g. points in dense vegetation.

Preliminaries: Estimation of a line

Line estimation

- Given: points

$$\mathbf{p}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad 1 \leq i \leq m$$

- Sought for: line *g*, which minimizes the sum of the quadratic distances
- Often used, since it is simple/linear: $g: \quad y = ax + b$
- Observation equation: $y_i + v_i = ax_i + b$

$$\mathbf{y} + \mathbf{v} = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_m & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \mathbf{A} \begin{bmatrix} a \\ b \end{bmatrix}$$

Line estimation

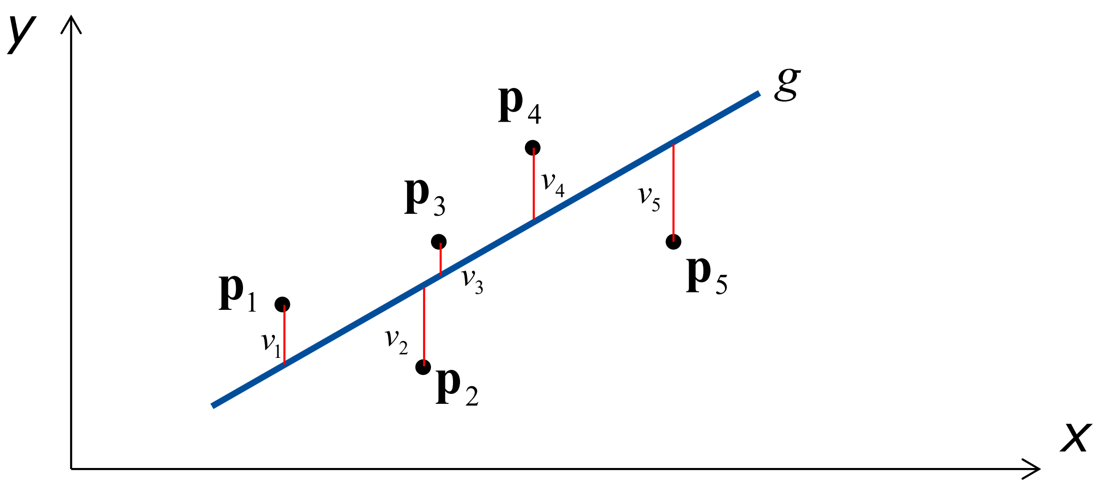
- Figuring out the standard least squares solution for the parameters a und b yields:

$$\mathbf{A} = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_m & 1 \end{bmatrix} \qquad \mathbf{A}^{\mathrm{T}}\mathbf{A} = \begin{bmatrix} \sum x_i^2 & \sum x_i \\ \cdot & m \end{bmatrix}$$

$$\boxed{\begin{bmatrix} a \\ b \end{bmatrix} = \mathbf{x} = \left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{y}}$$

Line estimation

- The observation equation $y_i + v_i = ax_i + b$ assumes the residuals are along the y axis
- This is different from the shortest distance (perpendicular) to the line
- Not useful if a line shall be estimated which has an arbitrary orientation.



Line estimation

- Solution: line equation in Hesse normal form:

$$ax + by + c = 0$$

- This can represent lines in arbitrary orientations, e.g.

$$1 \cdot x + 0 \cdot y + (-1) = 0 \quad \Leftrightarrow \quad x = 1$$

- Note: this uses 3 parameters for 2 degrees of freedom (dof) → normalization necessary
  - typically: $a^2 + b^2 = 1$
- $\mathbf{n} = \begin{bmatrix} a & b \end{bmatrix}^{\mathrm{T}}$ is the normal vector (perpendicular to the line direction)
- If **n** is normalized, then the (signed) distance of an arbitrary point to the line is given by:

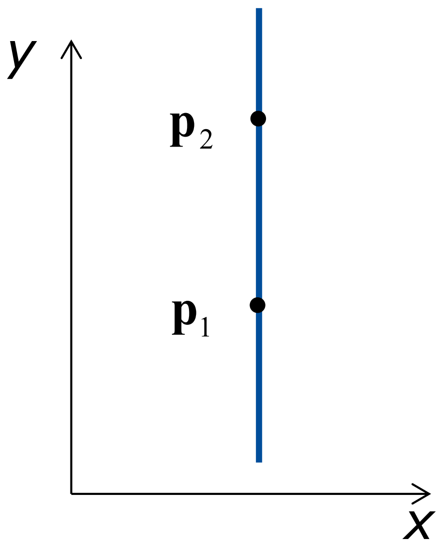
$$dist\left(\begin{bmatrix} x & y \end{bmatrix}^{\mathrm{T}}\right) = ax + by + c$$

Line estimation

- Estimate line through:

$$\mathbf{p}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \mathbf{p}_2 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

?



$$x = 1$$

Cannot be written in the form
$y = ax + b$

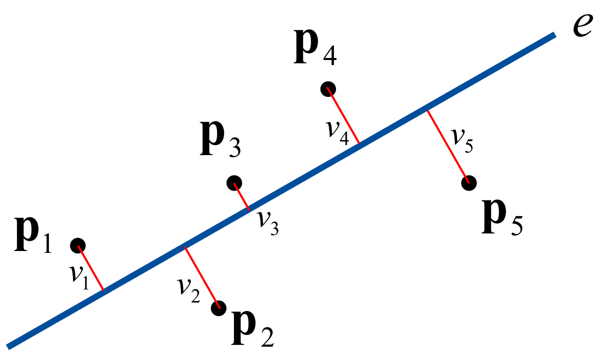
Estimation of planes

Plane estimation

- Given: m points

$$\mathbf{p}_i = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} \quad 1 \le i \le m$$

- Sought for: optimal plane e, such that the sum of the quadratic distances of the points to the plane is minimal.



Plane in Hesse normal form

- Point $[x \quad y \quad z]^{\mathrm{T}}$ is in the plane *e*, given by *a*, *b*, *c*, *d*, if:

$$ax + by + cz + d = 0$$

- Analogous to the case of the line:
  - $\mathbf{n} = [a \quad b \quad c]^{\mathrm{T}}$ is the normal vector of the plane
  - If the normal vector is normalized, then the distance of an arbitrary point **p** from the plane is:

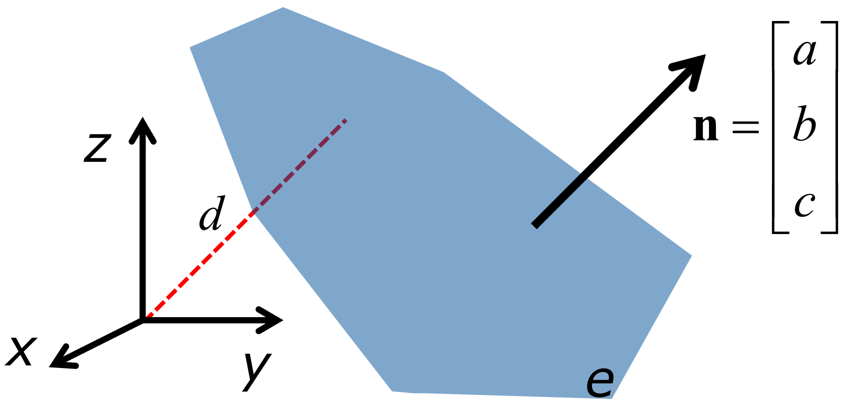
$$\|\mathbf{n}\| = 1 \quad (\text{i.e., } a^2 + b^2 + c^2 = 1) \quad \Rightarrow$$

$$dist(\mathbf{p}) = dist\left([x \quad y \quad z]^{\mathrm{T}}\right) = ax + by + cz + d$$

Plane in Hesse normal form

- This means:
  - The distance is obtained by substituting the point into the plane equation (if and only if the normal vector is normalized)
  - Note: this is the **signed** distance
- It also follows that:
  - d is the distance of the plane to the origin, since:

$$dist(\mathbf{0}) = a \cdot 0 + b \cdot 0 + c \cdot 0 + d = d$$



Observation equation

- Define the residual for point *i*:

$$v_i = ax_i + by_i + cz_i + d$$

- Then, we are looking for the plane *e*:
  - *e* is given by: a, b, c, d
  - where the following shall hold: $a^2 + b^2 + c^2 = 1$
  - and *e* is chosen such that:

$$\sum v_i^2 = \mathbf{v}^{\mathrm{T}} \mathbf{v} \overset{!}{=} \min$$

- Note:
  - If we did not impose the normal vector length constraint, we would obtain the following trivial result:

$$a = b = c = d = 0$$

Observation equation

- Note $v_i = ax_i + by_i + cz_i + d$ i.e., in matrix form, written out:

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_m & y_m & z_m \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \end{bmatrix} + \begin{bmatrix} d \\ d \\ \vdots \\ d \end{bmatrix}$$

- In matrix notation:

$$\mathbf{v} = \mathbf{A}\mathbf{n} + \mathbf{1} \cdot d$$

Closed form computation of the solution

- Assume that the points are reduced by the center of mass (we know this trick!):

$$\mathbf{p}_i \quad 1 \leq i \leq m \quad \text{mit} \ \sum_{i=1}^{m} \mathbf{p}_i = \mathbf{0}$$

- Equivalently:

$$\sum_{i=1}^{m} x_i = \sum_{i=1}^{m} y_i = \sum_{i=1}^{m} z_i = 0$$

Closed form computation of the solution

- Compute the least squares sum:

$$\begin{aligned}\mathbf{v}^{\mathrm{T}}\mathbf{v} &= (\mathbf{An}+\mathbf{1}d)^{\mathrm{T}}(\mathbf{An}+\mathbf{1}d)\\ &= (\mathbf{n}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}+\mathbf{1}^{\mathrm{T}}d)(\mathbf{An}+\mathbf{1}d)\\ &= \mathbf{n}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{An}+\mathbf{n}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{1}d+d\mathbf{1}^{\mathrm{T}}\mathbf{An}+d^2\mathbf{1}^{\mathrm{T}}\mathbf{1}\end{aligned}$$

- Note:

$$\mathbf{1}^{\mathrm{T}}\mathbf{A}=\begin{bmatrix}1 & 1 & \cdots & 1\end{bmatrix}\begin{bmatrix}x_1 & y_1 & z_1\\ x_2 & y_2 & z_2\\ \vdots & \vdots & \vdots\\ x_m & y_m & z_m\end{bmatrix}=\begin{bmatrix}\sum x_i & \sum y_i & \sum z_i\end{bmatrix}=\mathbf{0}$$

since the points were assumed to be reduced by the center of mass

Closed form computation of the solution

- Then, it also follows that:

$$\mathbf{A}^{\mathrm{T}}\mathbf{1} = \left(\mathbf{1}^{\mathrm{T}}\mathbf{A}\right)^{\mathrm{T}} = \mathbf{0}$$

- In addition, the following holds:

$$d^2\mathbf{1}^{\mathrm{T}}\mathbf{1} = d^2\begin{bmatrix}1 & 1 & \cdots & 1\end{bmatrix}\begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix} = d^2 m$$

Closed form computation of the solution

- Overall, we obtain:

$$\begin{aligned}
\mathbf{v}^{\mathrm{T}}\mathbf{v} &= \mathbf{n}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{n} + \underbrace{\mathbf{n}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{1}d}_{0} + \underbrace{d\mathbf{1}^{\mathrm{T}}\mathbf{A}\mathbf{n}}_{0} + d^2\mathbf{1}^{\mathrm{T}}\mathbf{1} \\
&= \underbrace{\mathbf{n}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{n}}_{\text{Part 1}} + \underbrace{d^2 m}_{\text{Part 2}}
\end{aligned}$$

- Look at the parts of this sum:
  - Part 1 depends only on the normal vector
  - Part 2 depends only on *d*
- Both parts can therefore be minimized independently
  - The minimum of $d^2 m$ is obtained for $d = 0$
  - I.e., the center of mass lies in the plane!

Closed form computation of the solution

- The only remaining part is the minimization of:

$$\mathbf{n}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{n} \quad \text{under the condition } \|\mathbf{n}\| = 1$$

- Solution (cf. our solution for the 7-parameter estimation):
  - The solution is the eigenvector belonging to the smallest eigenvalue of

$$\mathbf{A}^{\mathrm{T}}\mathbf{A}$$

  - Note: this is the following, symmetric matrix:

$$\mathbf{A}^{\mathrm{T}}\mathbf{A} = \begin{bmatrix} \sum x_i^2 & \sum x_i y_i & \sum x_i z_i \\ \cdot & \sum y_i^2 & \sum y_i z_i \\ \cdot & \cdot & \sum z_i^2 \end{bmatrix}$$

Cookbook: Plane estimation

- Given: arbitrary points

$$\mathbf{p}_i = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} \quad 1 \le i \le m$$

- Sought for: plane e, which minimizes the quadratic (perpendicular) distances, given by:

$$e: \quad ax + by + cz + d = 0$$

Cookbook: Plane estimation

- 1. Determine center of mass and reduced coordinates

$$\bar{\mathbf{p}} = \frac{1}{m}\sum_{i=1}^{m}\mathbf{p}_i\,, \quad \mathbf{p}'_i = \mathbf{p}_i - \bar{\mathbf{p}}$$

- 2. Compute $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ using reduced coordinates:

$$\mathbf{A}^{\mathrm{T}}\mathbf{A} = \begin{bmatrix} \sum x_i'^2 & \sum x_i' y_i' & \sum x_i' z_i' \\ \cdot & \sum y_i'^2 & \sum y_i' z_i' \\ \cdot & \cdot & \sum z_i'^2 \end{bmatrix}$$

  - The sought-for normal vector $\mathbf{n}$ is the eigenvector belonging to the smallest eigenvalue of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$
  - This determines the parameters *a, b, c*
  - (Note: $\mathbf{n}$ should be normalized to unit length for the steps on the following slide)

Cookbook: Plane estimation

- 3. The center of mass lies on the plane, i.e.:

$$dist(\overline{\mathbf{p}}) = \mathbf{n}^{\mathrm{T}}\overline{\mathbf{p}} + d = 0$$

  - Therefore, the remaining parameter *d* is determined by:

$$d = -\mathbf{n}^{\mathrm{T}}\overline{\mathbf{p}} = -\left(a\bar{x} + b\bar{y} + c\bar{z}\right)$$

Plane estimation: standard deviation estimate

- Sum of the squared residuals of all point distances

$$v_i = ax_i + by_i + cz_i + d \qquad \sum_{i=1}^{m} (v_i)^2 = \mathbf{v}^{\mathrm{T}}\mathbf{v}$$

- Standard deviation of the point distance:

$$\boxed{\sigma = \sqrt{\frac{\mathbf{v}^{\mathrm{T}}\mathbf{v}}{m-3}}}$$

- How to compute this?
  - Loop: substitute each point into the plane equation, compute v_i, compute square, sum up…
  - Simpler solution: note that **n** was obtained as eigenvector belonging to the smallest eigenvalue λ of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$

$$\mathbf{v}^{\mathrm{T}}\mathbf{v} = \mathbf{n}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{n} = \mathbf{n}^{\mathrm{T}}\lambda\mathbf{n} = \lambda$$

Therefore, σ can be directly obtained from λ.

Plane estimation: Using weights

- It is also possible to use the closed-form estimate of a plane if the points are weighted:
  - Function to minimize:

$$\mathbf{v}^{\mathrm{T}}\mathbf{P}\mathbf{v} \overset{!}{=} \min$$

  - Typically, a diagonal matrix is used (one weight per point)

$$\mathbf{P} = \mathrm{diag}(p_1, p_2, \ldots, p_m) = \begin{bmatrix} p_1 & 0 & \cdots & 0 \\ 0 & p_2 & & 0 \\ \vdots & & \ddots & 0 \\ 0 & 0 & 0 & p_m \end{bmatrix}$$

Plane estimation: Using weighted points, modified cookbook

- 1. Reduction by center of mass (**q** are points, *p* weights)

$$\overline{\mathbf{q}} = \frac{1}{\sum_{i=1}^{m} p_i} \cdot \sum_{i=1}^{m} p_i \cdot \mathbf{q}_i \,, \quad \mathbf{q}_i' = \mathbf{q}_i - \overline{\mathbf{q}}$$

- 2. Matrix $\mathbf{A}^{\mathrm{T}}\mathbf{P}\mathbf{A}$:

$$\mathbf{A}^{\mathrm{T}}\mathbf{P}\mathbf{A} = \begin{bmatrix} \sum p_i x_i'^2 & \sum p_i x_i' y_i' & \sum p_i x_i' z_i' \\ \cdot & \sum p_i y_i'^2 & \sum p_i y_i' z_i' \\ \cdot & \cdot & \sum p_i z_i'^2 \end{bmatrix}$$

  - From this, eigenvector for smallest eigenvalue: $\mathbf{n} = \begin{bmatrix} a & b & c \end{bmatrix}^{\mathrm{T}}$

- 3. Plane parameters *d*:

$$d = -\mathbf{n}^{\mathrm{T}} \overline{\mathbf{q}} = -\left(a\overline{x} + b\overline{y} + c\overline{z} \right)$$

Segmentation methods

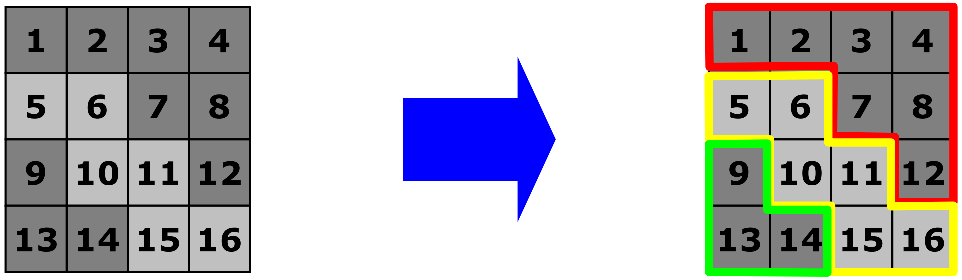
Segmentation - Definition

- Region based segmentation:
  - Partitioning of an overall region *R* into disjunct, connected parts R_i
  - All parts satisfy a given predicate $P(\cdot)$
  - The parts are maximal, i.e. they could not be made larger

- Formal definition:

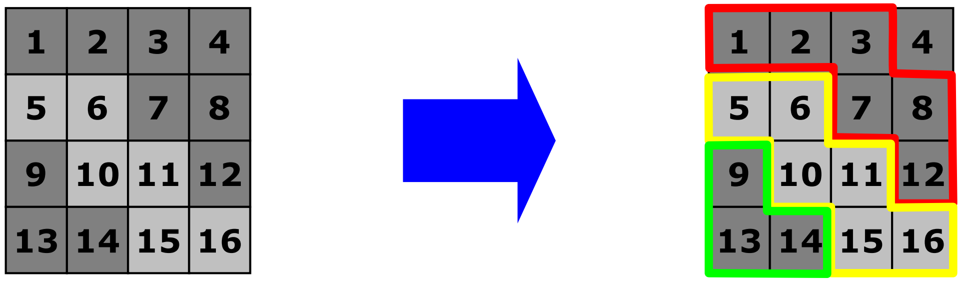
1. $\bigcup_{i=1}^{n} R_i = R$
2. $\forall\, 1 \leq i \leq n : R_i$ is connected
3. $\forall i \neq j : R_i \cap R_j = \varnothing$
4. $P(R_i) = \text{true}$
5. $\forall i \neq j : P(R_i \cup R_j) = \text{false, if } R_i, R_j \text{ are neighbors}$

Segmentation – Example (image)



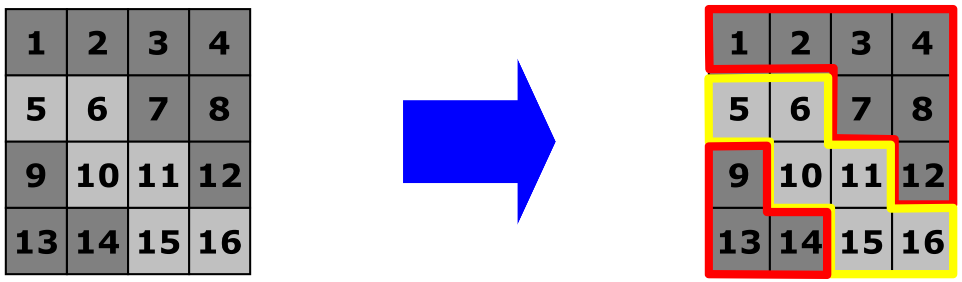
$$R_1 = \{1, 2, 3, 4, 7, 8, 12\}$$
$$R_2 = \{5, 6, 10, 11, 15, 16\}$$
$$R_3 = \{9, 13, 14\}$$

Segmentation – Example (image)



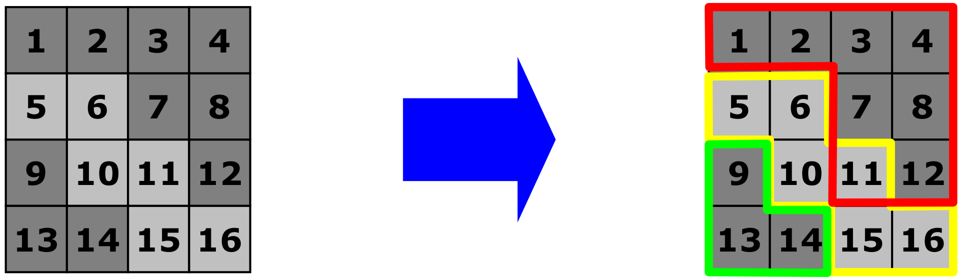
1. $\boxed{\bigcup_{i=1}^{n} R_i = R}$
2. $\forall\, 1 \leq i \leq n : R_i$ is connected
3. $\forall i \neq j : R_i \cap R_j = \varnothing$
4. $P(R_i) = \text{true}$
5. $\forall i \neq j : P(R_i \cup R_j) = \text{false}$, if R_i, R_j are neighbors

Segmentation – Example (image)



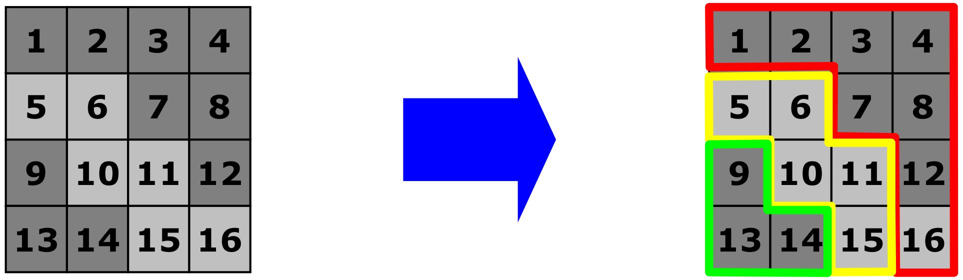
1. $\bigcup_{i=1}^{n} R_i = R$
2. $\boxed{\forall\, 1 \le i \le n : R_i \text{ is connected}}$
3. $\forall i \neq j : R_i \cap R_j = \varnothing$
4. $P(R_i) = \text{true}$
5. $\forall i \neq j : P(R_i \cup R_j) = \text{false, if } R_i, R_j \text{ are neighbors}$

Segmentation – Example (image)



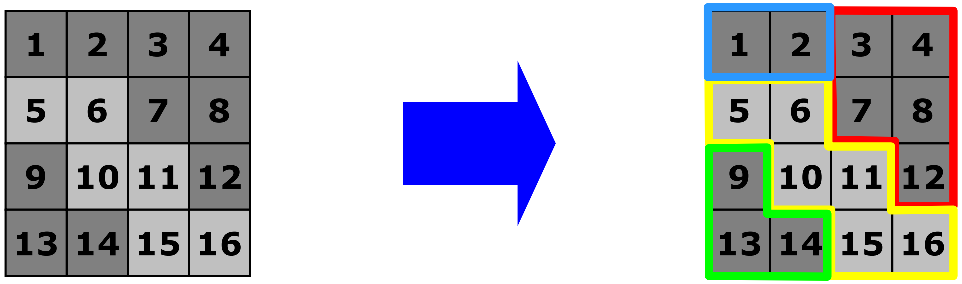
1. $\bigcup_{i=1}^{n} R_i = R$
2. $\forall\, 1 \le i \le n : R_i$ is connected
3. $\boxed{\forall i \ne j : R_i \cap R_j = \varnothing}$
4. $P(R_i) = \text{true}$
5. $\forall i \ne j : P(R_i \cup R_j) = \text{false, if } R_i, R_j$ are neighbors

Segmentation – Example (image)



1. $\bigcup_{i=1}^{n} R_i = R$
2. $\forall\, 1 \le i \le n : R_i$ is connected
3. $\forall i \ne j: R_i \cap R_j = \varnothing$
4. $\boxed{P(R_i) = \text{true}}$
5. $\forall i \ne j : P(R_i \cup R_j) = \text{false, if } R_i, R_j \text{ are neighbors}$

Segmentation – Example (image)

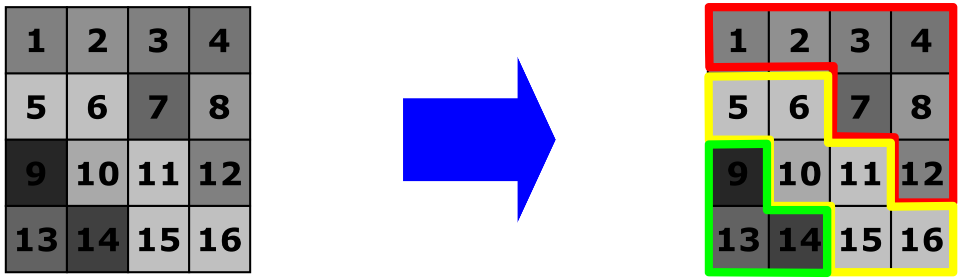


1. $\bigcup_{i=1}^{n} R_i = R$
2. $\forall\, 1 \le i \le n : R_i$ is connected
3. $\forall i \neq j : R_i \cap R_j = \varnothing$
4. $P(R_i) = \text{true}$
5. $\boxed{\forall i \neq j : P(R_i \cup R_j) = \text{false, if } R_i, R_j \text{ are neighbors}}$

Segmentation

- Finding an optimal segmentation is intractable (computational cost)
- Two problems:
  - How many sub-regions are part of the overall region? (Model selection problem)
  - Find the optimal assignment of elements (pixels, points) to regions, which minimize a global error measure

Example: segmentation is intractable



G_1 $R_1 = \{1,2,3,4,7,8,12\}$

G_2 $R_2 = \{5,6,10,11,15,16\}$

G_3 $R_3 = \{9,13,14\}$

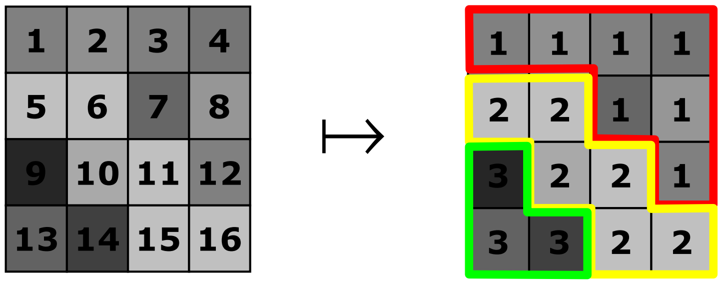
- In reality, gray values of a region are not identical
- Assume for now (for simplicity): we know it is 3 regions
- Task: search for:
  - Three gray values G_1, G_2, G_3 (one per region)
  - And an assignment function, assigning the pixels to the 3 regions, such that the total deviation (over all regions) is minimal

Example: segmentation is intractable

- Formally, define an assignment function
  - $z(i)$ is the assignment of cell *i* to the region $z(i) \in \{1,2,3\}$

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
z(i)	1	1	1	1	2	2	1	1	3	2	2	1	3	3	2	2

$$i \mapsto z(i)$$



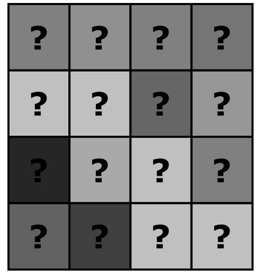
Example: segmentation is intractable

- Formal problem formulation
  - $z(i)$ is the assignment of cell *i* to the region $z(i) \in \{1,2,3\}$
  - Assume, our error measure is the sum of the squared differences
  - let p_i be the gray value of cell *i*
  - let G_j be the (optimal) gray value of region *j*

- Task:
  - Given: $p_i \ , \quad i \in \{1,2,\ldots,16\}$
  - Sought: G_1, G_2, G_3 and assignment table $i \mapsto z(i)$
    such that this error is minimized:

$$\mathrm{Error}(G_1, G_2, G_3; z) = \sum_{i=1}^{16} \left(p_i - G_{z(i)}\right)^2$$

Example: segmentation is intractable



- Searching for the minimal error:
  - There are 3^{16} = 43.046.721 possibilities for the assignment function
  - Each of those has to be checked for the criteria 1-5
  - For each assignment which passes those criteria, G_1, G_2, G_3 can be computed
  - From this, $\mathrm{Error}(G_1, G_2, G_3; z)$ can be computed

- Even for this very small problem this is quite costly!

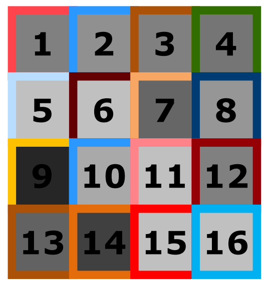
- Approach is not applicable to real data
  - E.g. 5M pixel image, 20 regions → $20^{5000000} = 10^{6505150}$

The "number of regions" – model selection

- In the previous example, the number of regions was known
- This case rarely happens in real life
- Some constraint must be imposed to limit the number of regions
- Because: trivial solution: 16 parts

$$\left.\begin{aligned} G_i &:= p_i \\ z(i) &:= i \end{aligned}\right\} \; i = 1, \ldots, 16$$

$$\Rightarrow \mathrm{Error}(G_1, G_2, \ldots, G_{16}; z) = 0$$



- This is a simple case of "model selection"
  - Which is a highly non-trivial problem, encountered in many places

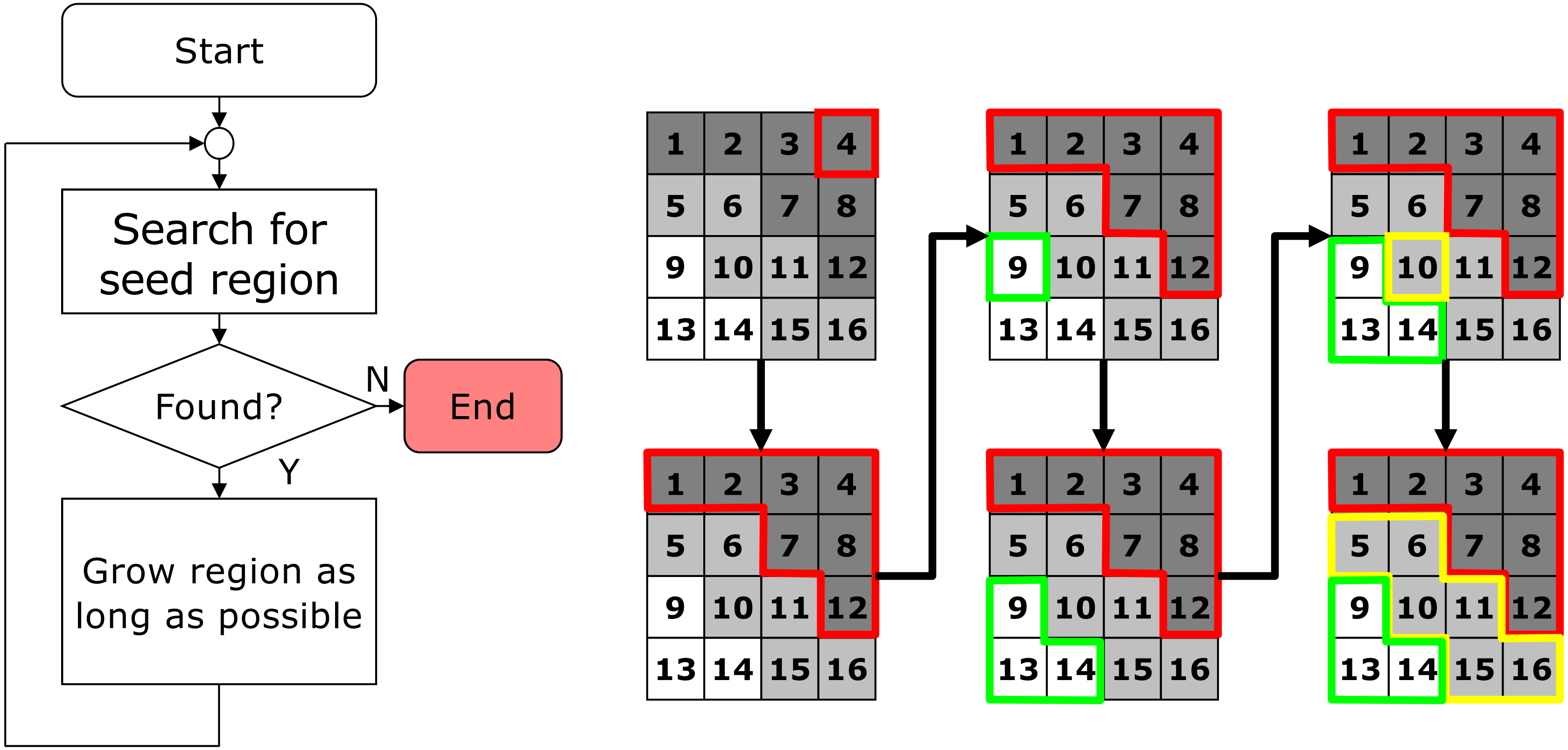
Segmentation: heuristics

- Since the problem is intractable, heuristics have been used for its solution
- Number of regions:
  - E.g. enforcement of a minimum region size
- Assignment of elements to parts:
  - Instead of a global optimum, use approximations. Often: greedy heuristics
  - E.g., accept “large” regions, without checking for a global optimum

- Well-known methods
  - Region Growing
  - RANSAC
  - Hough Transformation

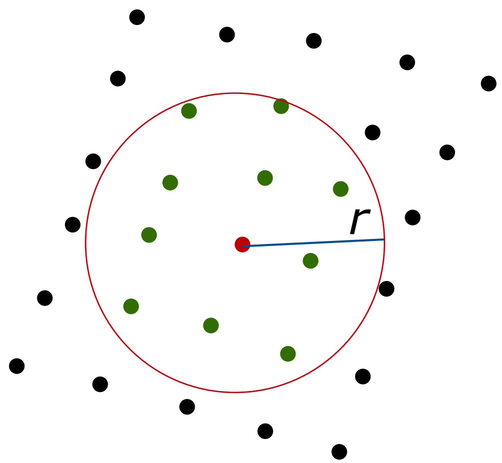
Segmentation: Region Growing (Surface Growing)

Region growing



Step 1: Detection of a seed region

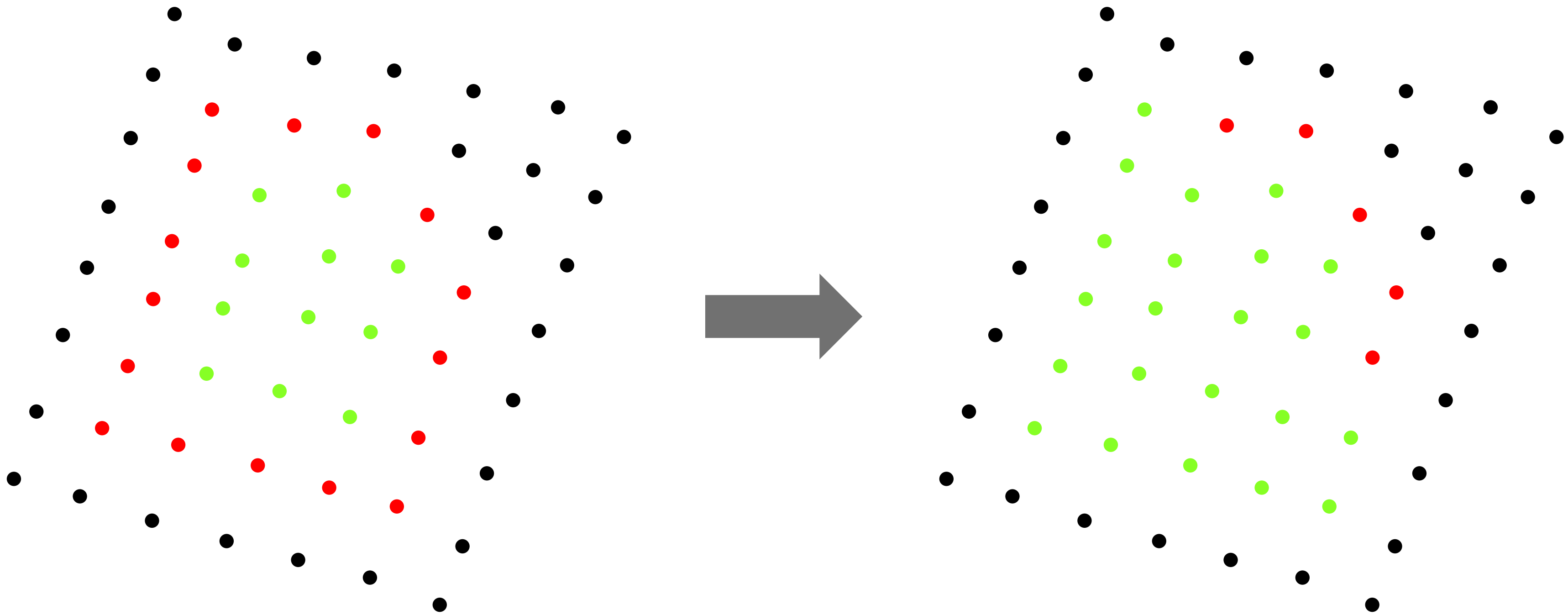
- Example: segmentation into planes
- Idea: a "good" seed region is planar (locally) →
  - For each point:
    - Compute best plane in a local neighborhood (ball with radius r)
    - Compute standard deviation of the distance of the points to the plane (using the equations above)
  - Select point with smallest standard deviation as seed region



Step 2a: Growing the region

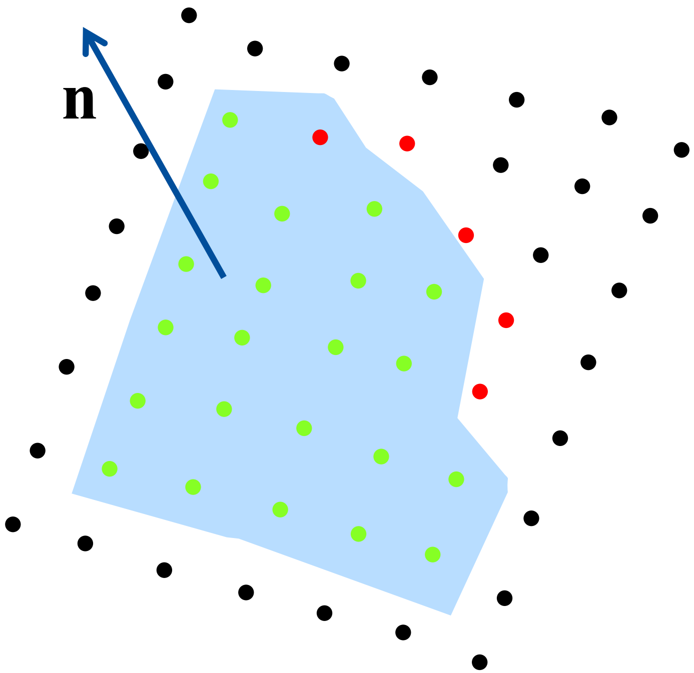
- Consider the neighbors ● of the current region ●
- If these neighbors are close to the plane, add them to the current region:

$$|ax + by + cz + d| < \varepsilon \Rightarrow \text{add the points}$$



Step 2b: Improve the plane equation

- Using the newly added points, the parameters of the plane are improved (by re-estimation of the plane)
- Trade-off:
  - Re-estimation is costly, do not perform this after each addition of a single point
  - → e.g.: re-estimate only after ~20-50% points were added



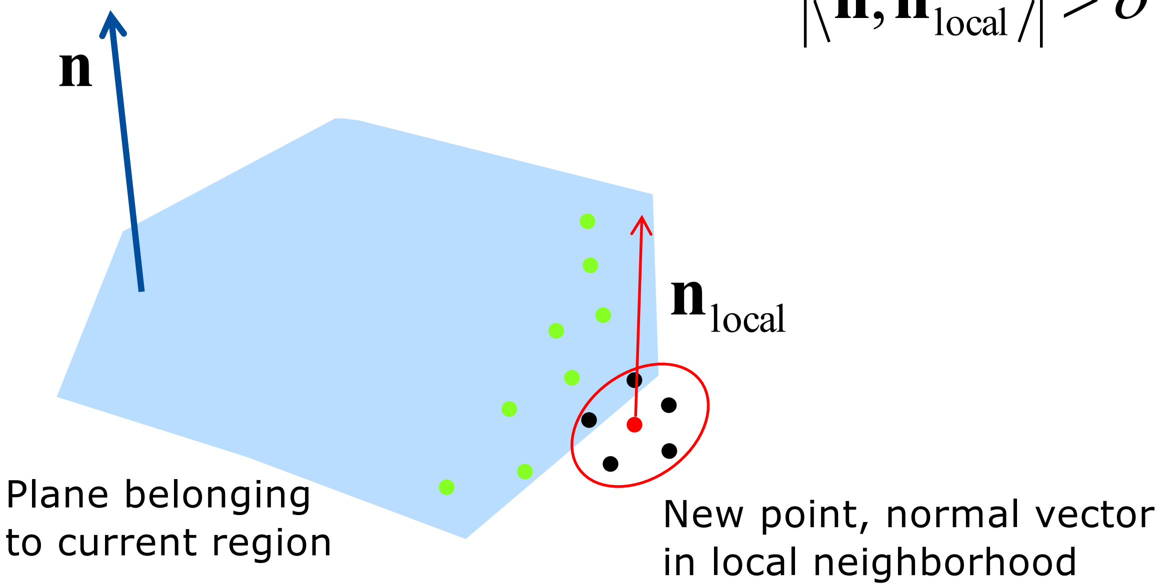
Step 2c: Repeat

- If step 2a did not add new points, growing the current region will terminate
- The result is then the current region and its associated plane
- Then, the next seed region is selected and expanded
  - Note: previously found "best" seed regions may be part of an already expanded region
  - All points which are assigned to a region already cannot be used anymore (all used points must be marked)

Enhancement for step 2a

- Instead of checking only the point-to-plane distance, one may include (in addition) a comparison of the local (point) normal vector with the normal vector of the region's plane

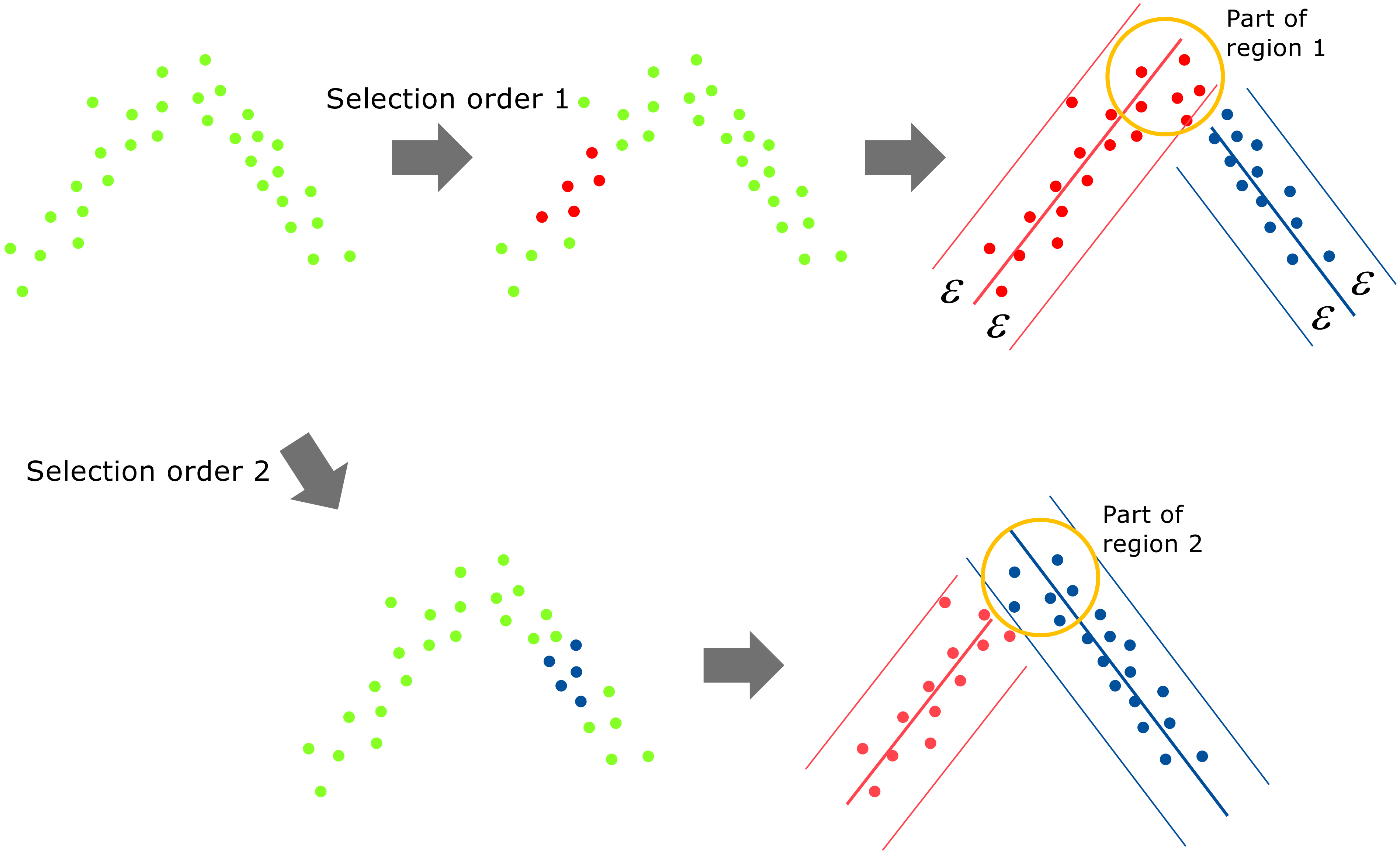
$$\left.\begin{array}{r} |ax+by+cz+d| < \varepsilon \\ |\langle \mathbf{n}, \mathbf{n}_{\mathrm{local}} \rangle| > \delta \end{array}\right\} \Rightarrow \text{add to region}$$



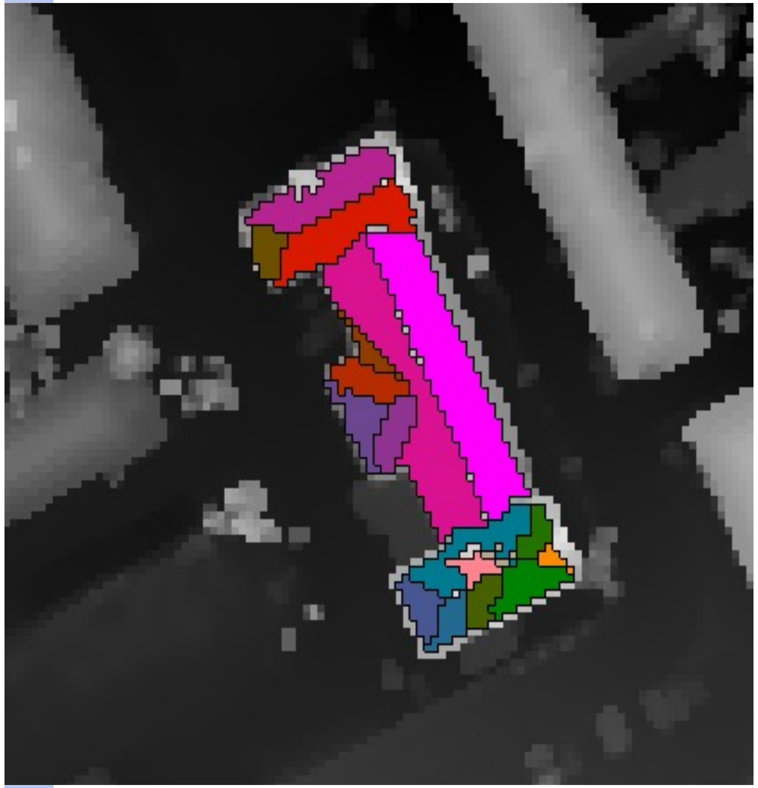
Problems of the region growing approach

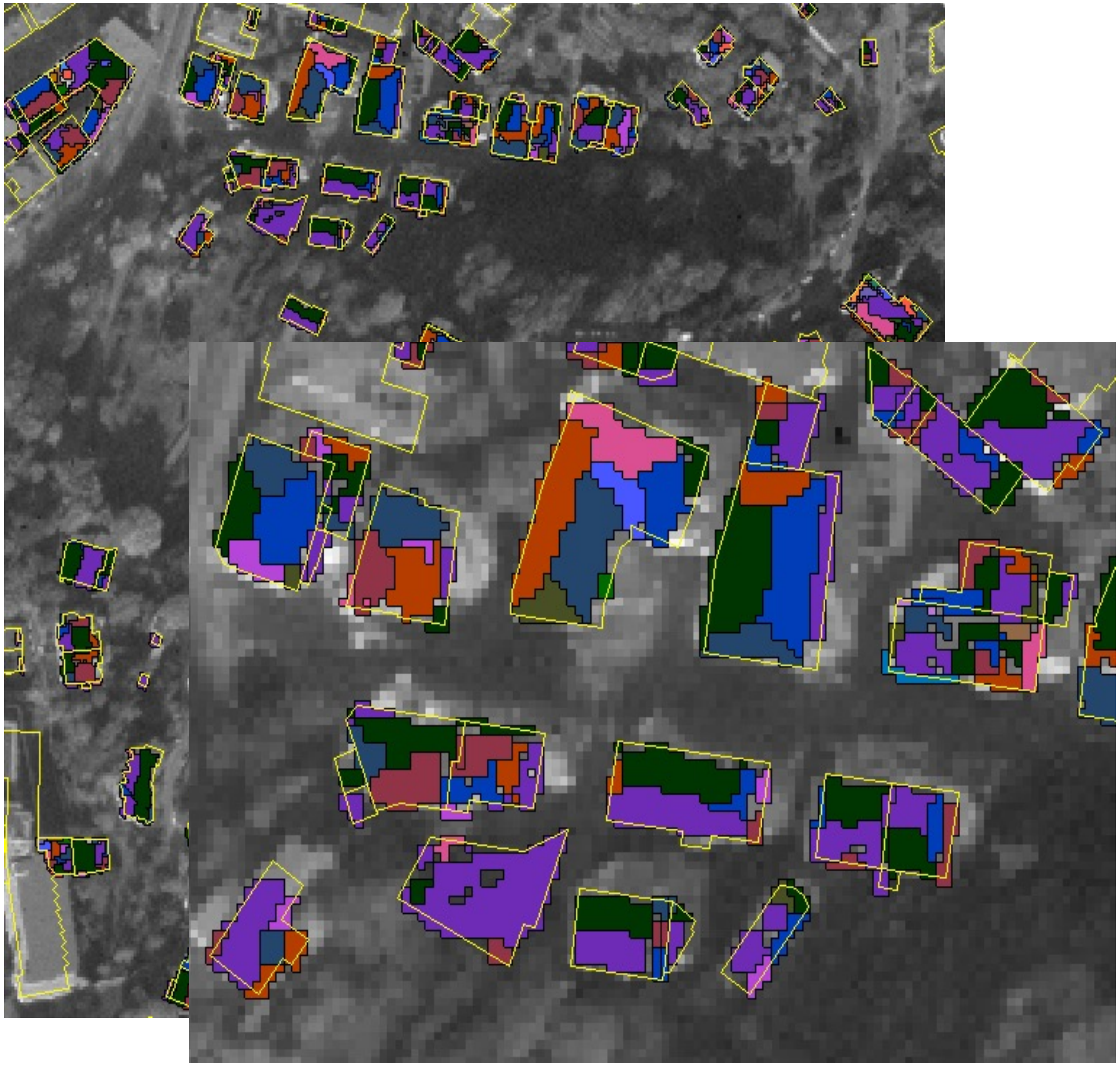
- Iterative addition of points and re-estimation of the plane is costly
- Since the algorithm is greedy, the result depends on the **order of the selection of the seed regions**
  - Region borders may be improved by re-assignment of (already assigned) points along the borders

Result variations depending on the seed region selection order

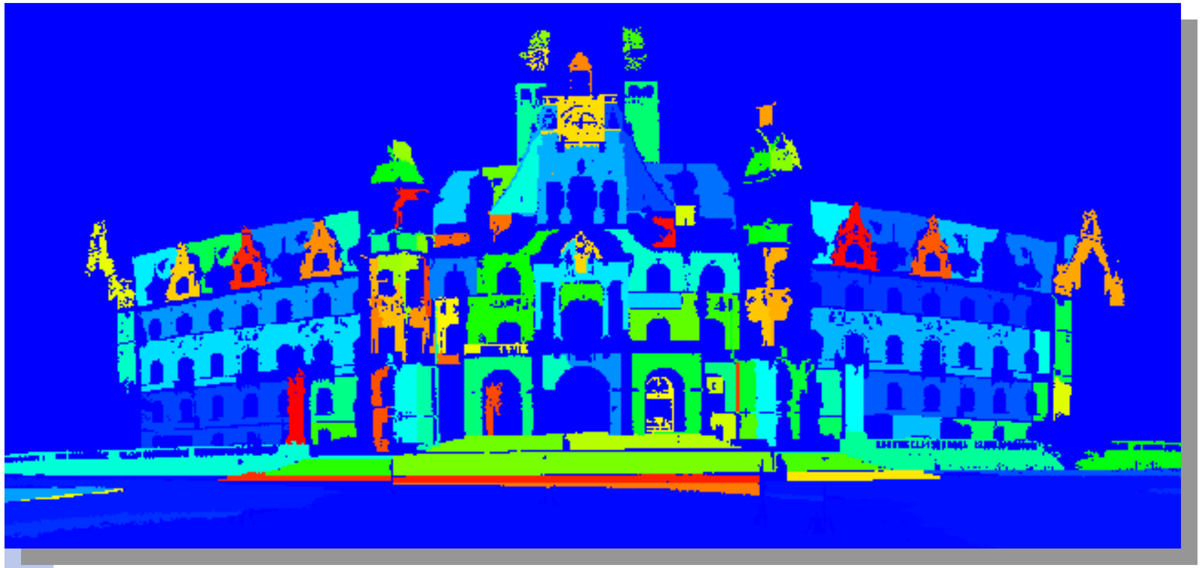
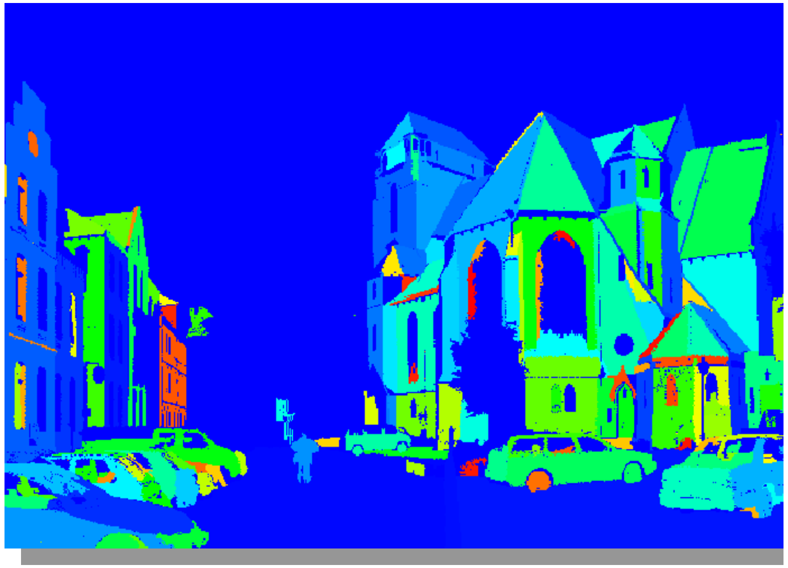
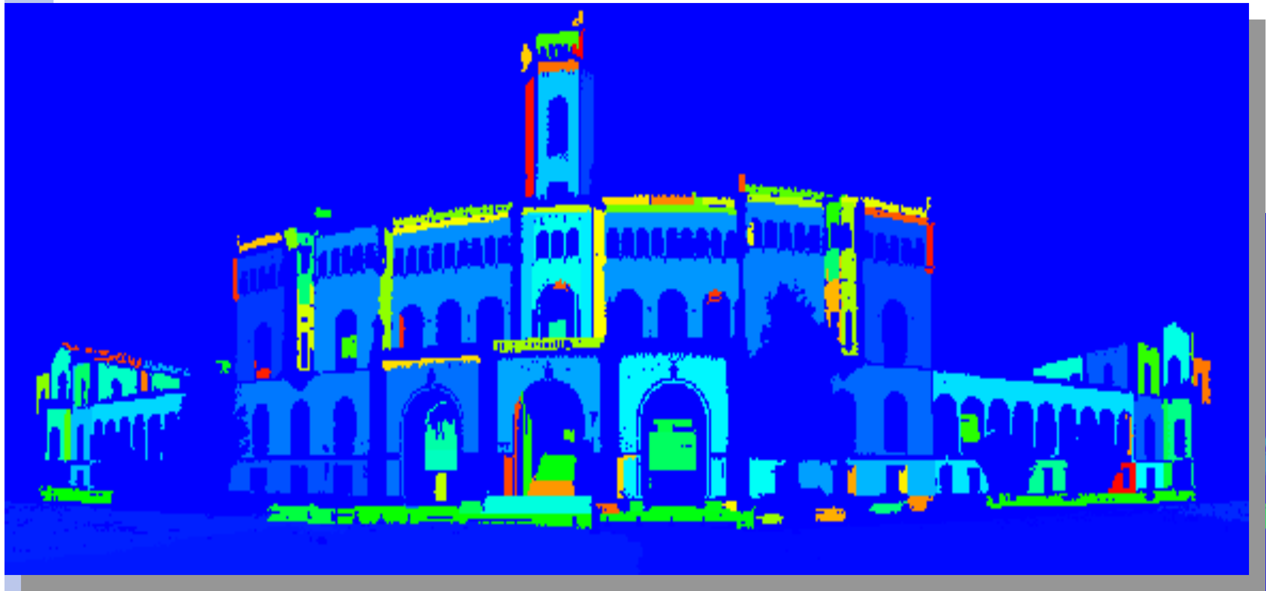
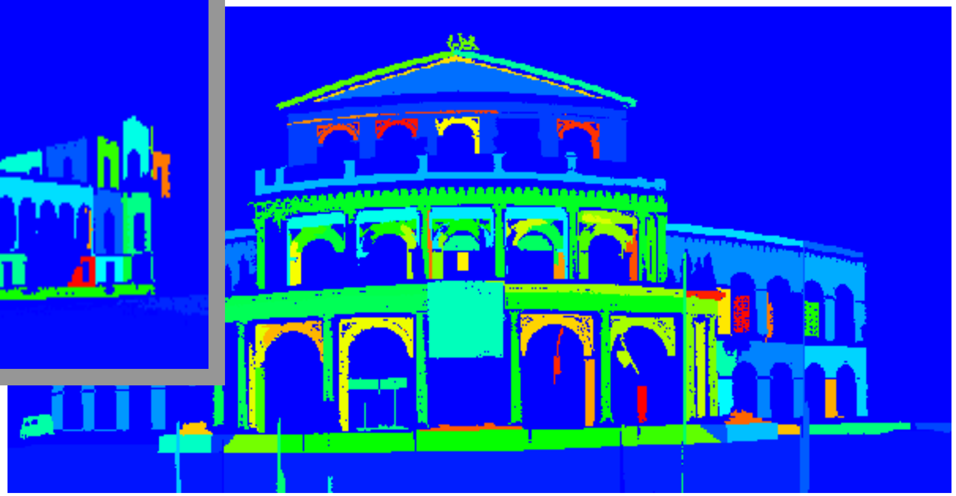


Region growing: Examples





Region growing: examples (terrestrial scans)

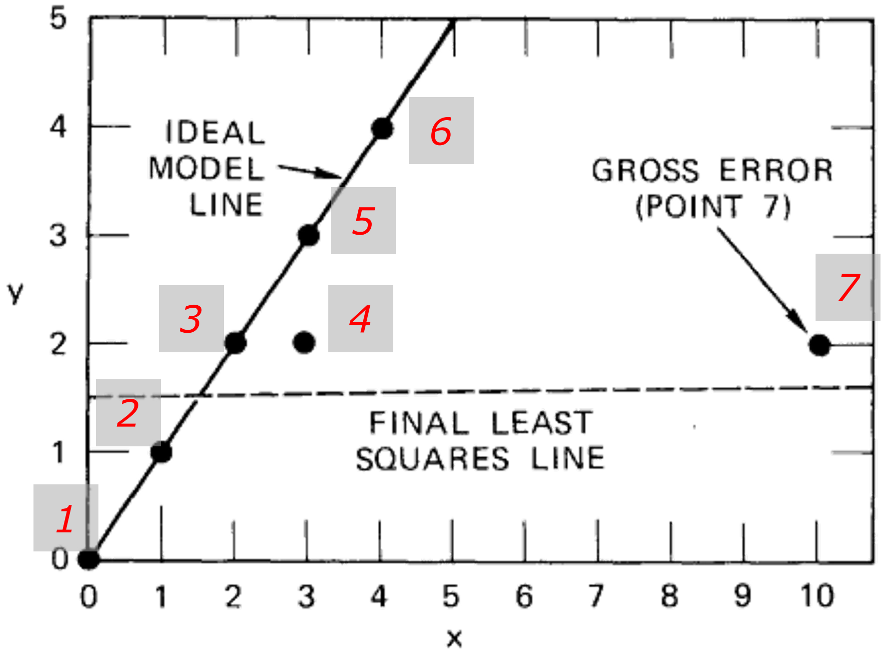





Segmentation: RANSAC

RANSAC (**Ran**dom **Sa**mple **C**onsensus, Fischler & Bolles 1981)

- General method for robust parameter estimation
- Algorithm:
  - Repeat (maximum *k* times):
    - Select the number *n* of necessary observations by randomly picking them (e.g. *n*=3 points for a plane, *n*=2 points for a line) **= random sample**
    - Compute (exactly) the parameters based on the selected observations
    - Compute the number *m* of total observations which are compatible **= consensus**
    - If the number of compatible observations (the consensus set) is larger than a threshold *t*, accept the solution immediately and break the loop [This is often omitted, see later.]
- Very useful algorithm, which is very simple to implement.

RANSAC (**Ran**dom **Sa**mple **C**onsensus, Fischler & Bolles 1981)



POINT	x	y
1	0	0
2	1	1
3	2	2
4	3	2
5	3	3
6	4	4
7	10	2

COMMENT: Six of the seven points are valid data and can be fit by the solid line. Using Least Squares (and the "throwing out the worst residual" heuristic), we terminate after four iterations with four remaining points, including the gross error at (10,2) fit by the dashed line.

SUCCESSIVE LEAST SQUARES APPROXIMATIONS

ITERATION	DATA SET	FITTING LINE
1	1, 2, 3, 4, 5, 6, 7	1.48 + .16x
2	1, 2, 3, 4, 5, 7	1.25 + .13x
3	1, 2, 3, 4, 7	0.96 + .14x
4	2, 3, 4, 7	1.51 + .06x

Source: Fischler & Bolles 1981

Number of draws

- How many draws should be undertaken?
- Fischler & Bolles give 2 methods to determine this, here shown is method 2:
  - Let w be the probability (prob.), that a (single) drawn observation belongs to the model
  - Then:
    - $b=w^n$ is the prob. that all n draws belong to the model
    - $(1-b)$ is the prob. that at least one draw does not belong to the model, i.e., the probability that the overall draw is unusable
    - $(1-b)^k$ is the prob. that k such draws are unusable
    - $1-(1-b)^k$ is the prob. that at least one of the k draws is usable
  - One now specifies the desired probability z with which the model shall be found in the data
    - $1-(1-b)^k > z$

Number of draws

- It follows:

$$1-(1-b)^k > z$$

$$(1-b)^k < 1-z$$

$$k \log(1-b) < \log(1-z)$$

- Therefore*:

$$\boxed{k > \frac{\log(1-z)}{\log(1-b)}}$$

*note that:

$$\log(1-b) < 0$$

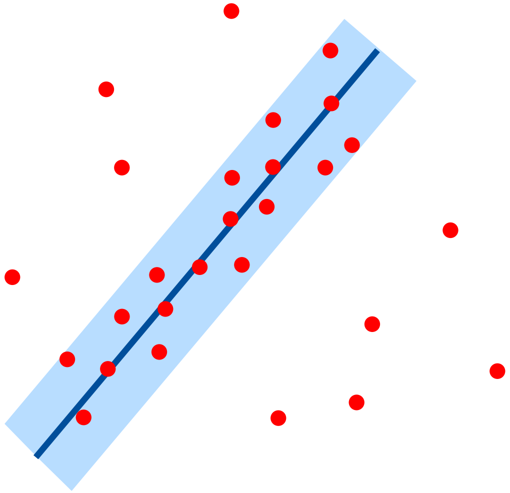
Number of draws – Example

- Estimation of a line from point samples
- It is known that there are 30% outliers → $w=0.7$
- Two points have to be drawn per line → $b=w^2=0.49$
- The correct line shall be found with probability 95% → $z=0.95$

$$k > \frac{\log(1-z)}{\log(1-b)} = \frac{\log(0.05)}{\log(0.51)} = 4.45 \quad \Rightarrow k \geq 5$$

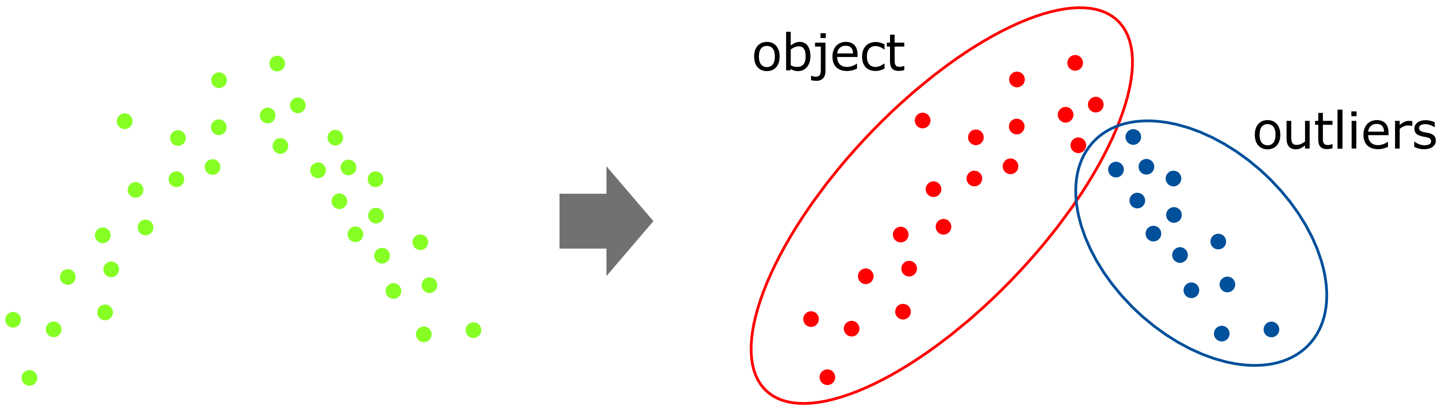
Although there is a large number of outliers (30%), the number of required draws (5) is rather small.

For $z=0.99$ we obtain: $k \geq 7$

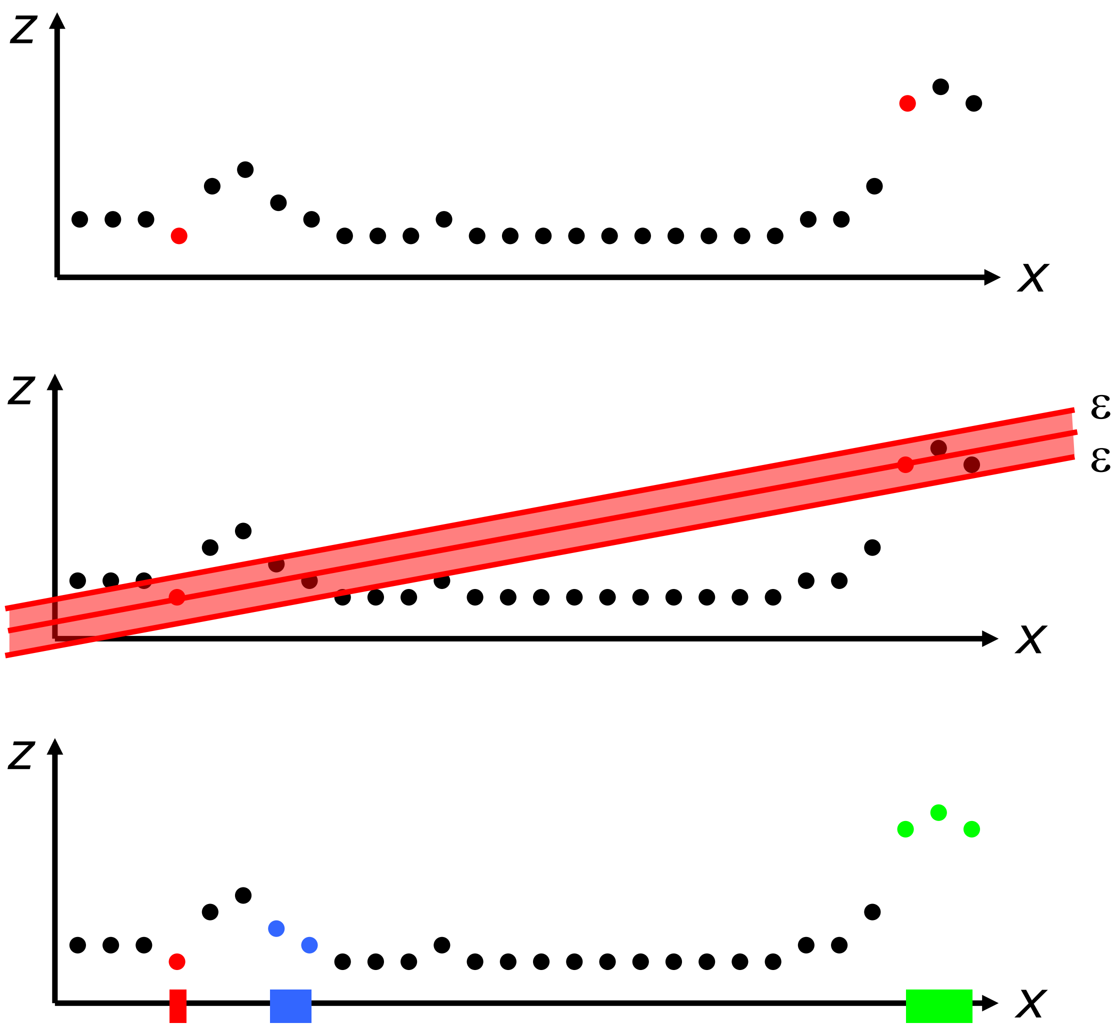


Using RANSAC for region segmentation

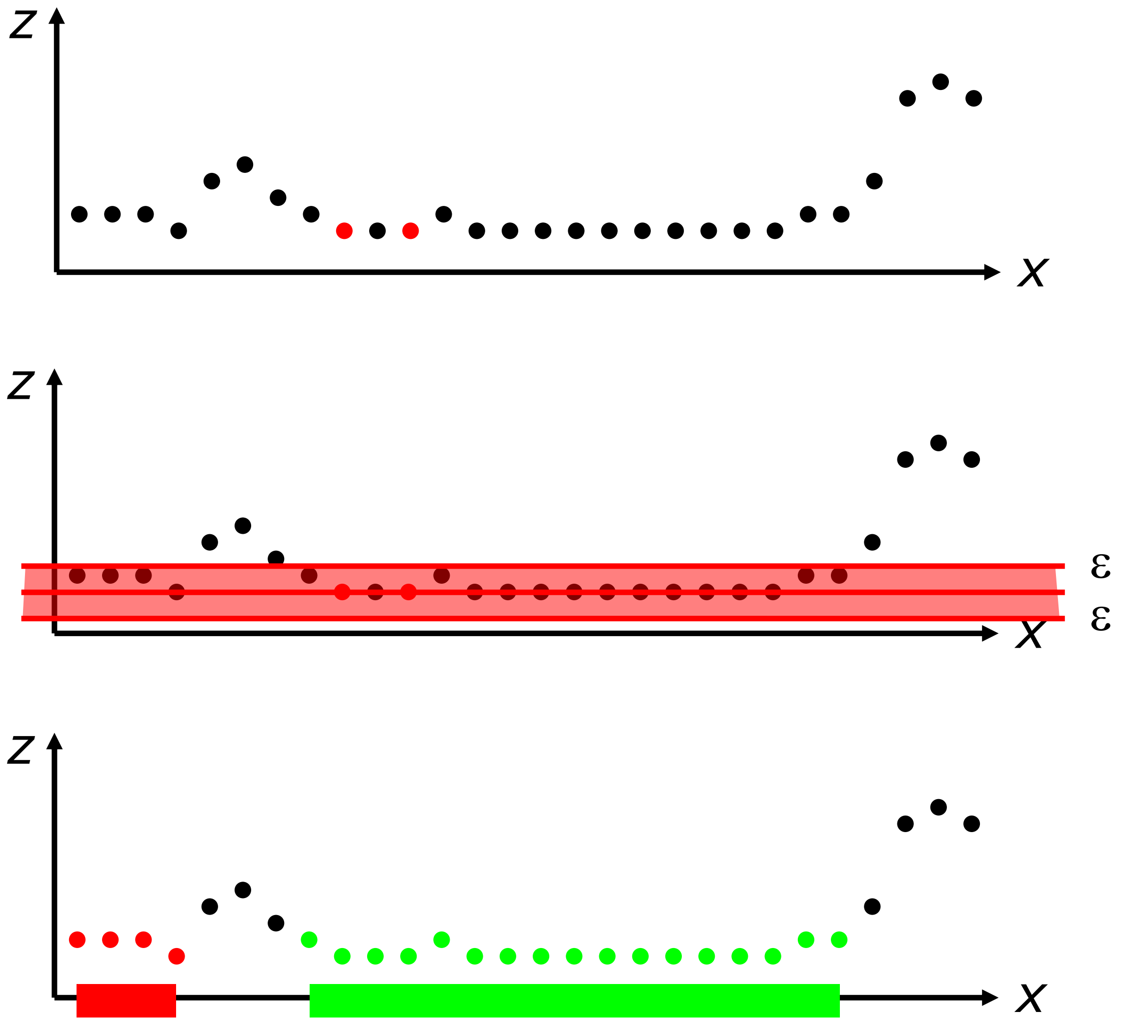
- Consider the number of observations to be separated into:
  - Observations which belong to **one object** (model), and
  - **all other observations**, which are considered to be **outliers**
- Using RANSAC, find this object
- Delete all observations belonging to this object and repeat the search for an object in the remaining point set



RANSAC example: finding linear segments

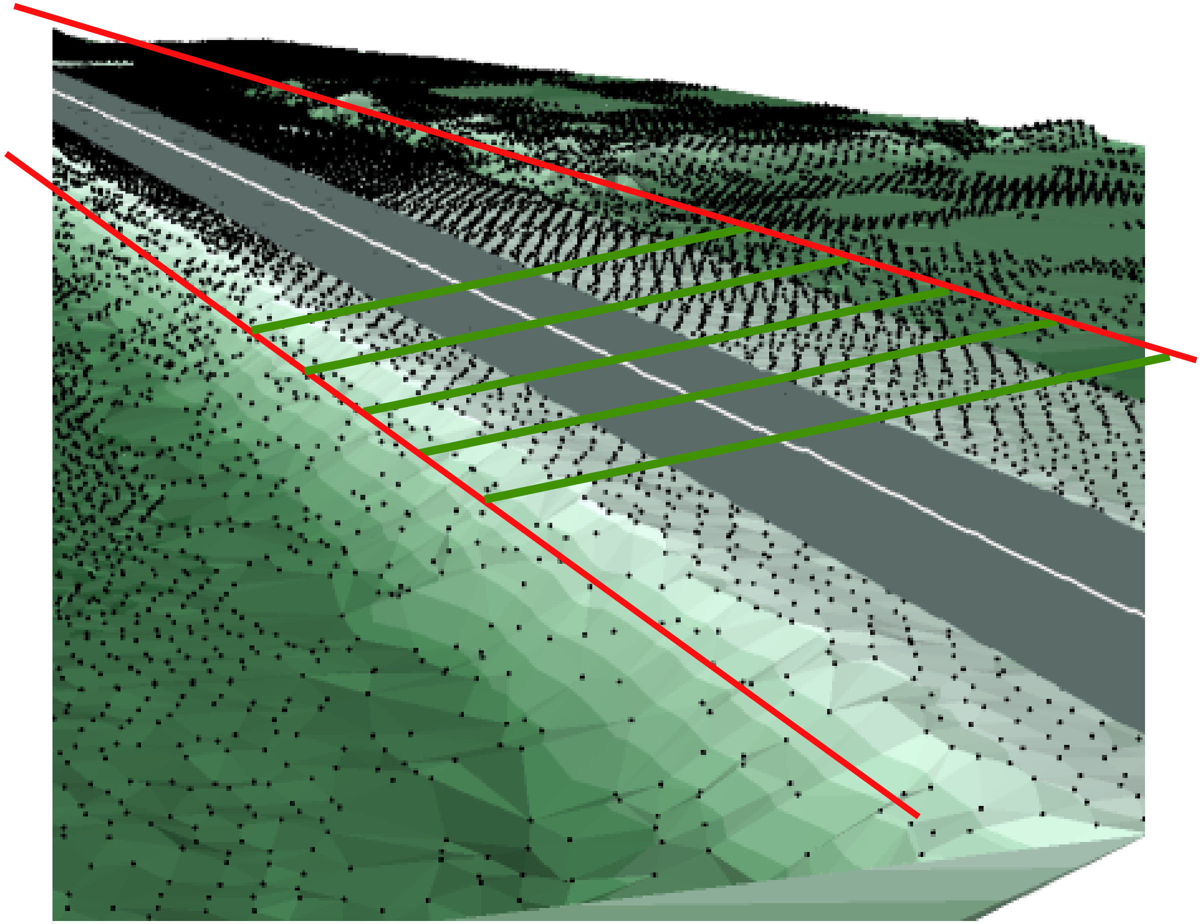


RANSAC example: finding linear segments



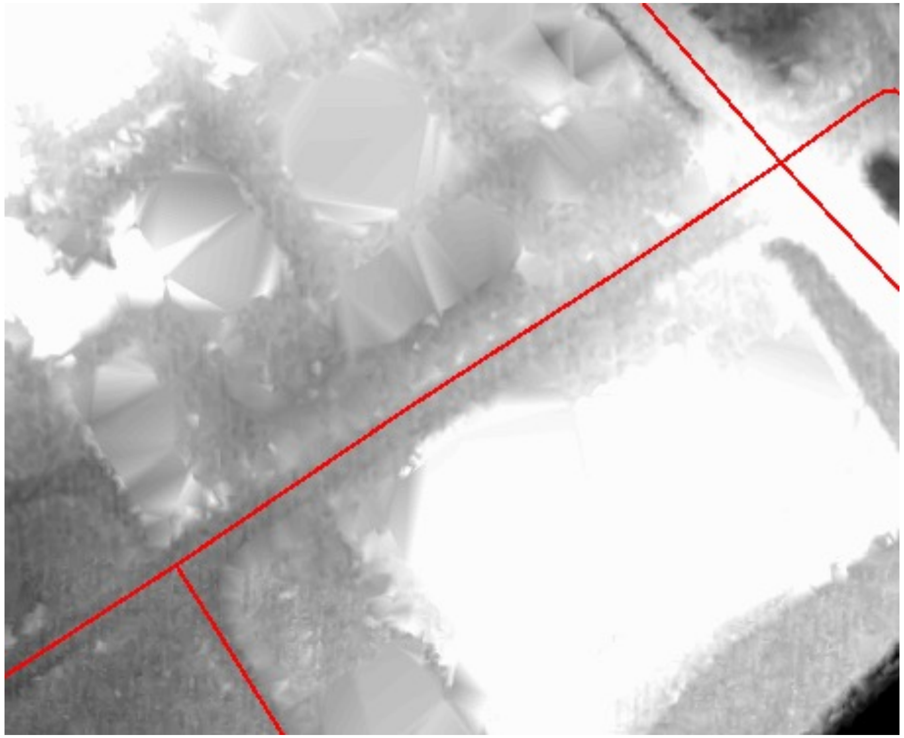
Example application: road segmentation in digital terrain models (DTMs)

- Sampling of DTM profiles orthogonal to street axis

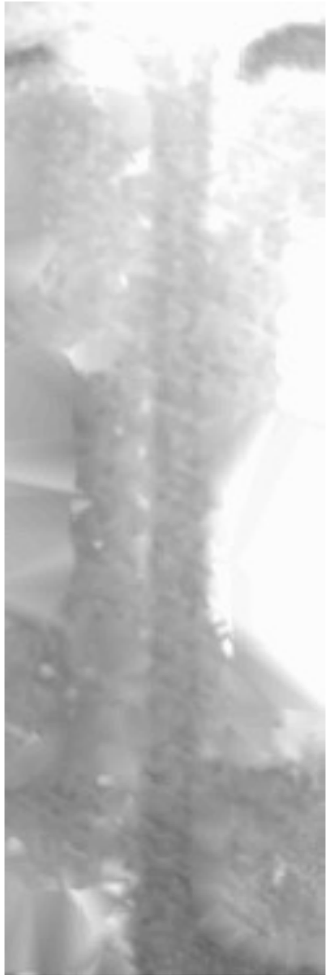


Resampling

- Resampling based on road centerline (given by a GIS)



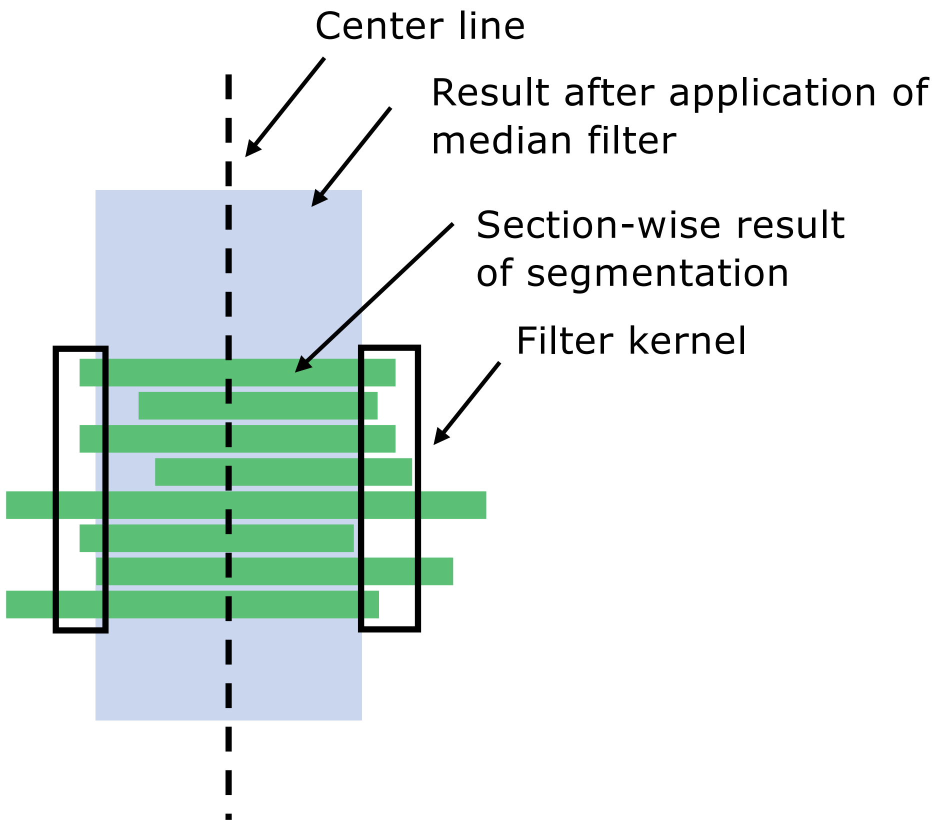
Road centerlines mapped onto DTM, Resolution *0,5 m.*



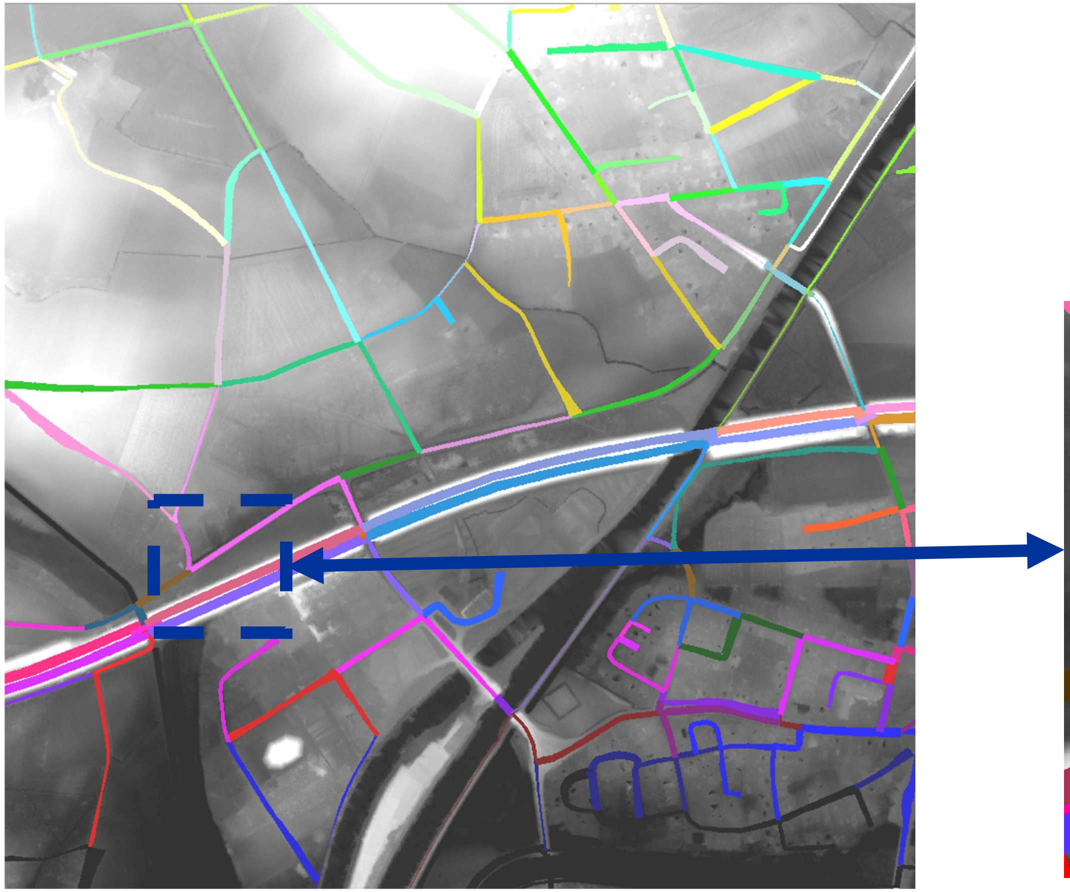
Resampled part of DTM

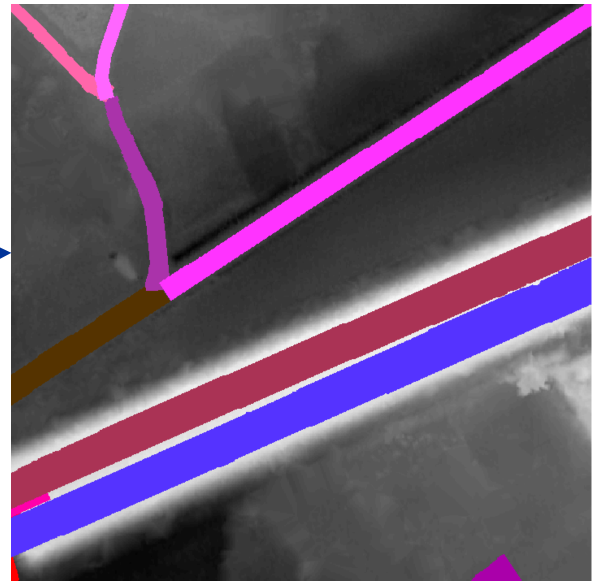
RANSAC performed on each profile individually

- After finding the extents for each line, a median filter is applied to smooth the borders



Results, re-projected onto DTM

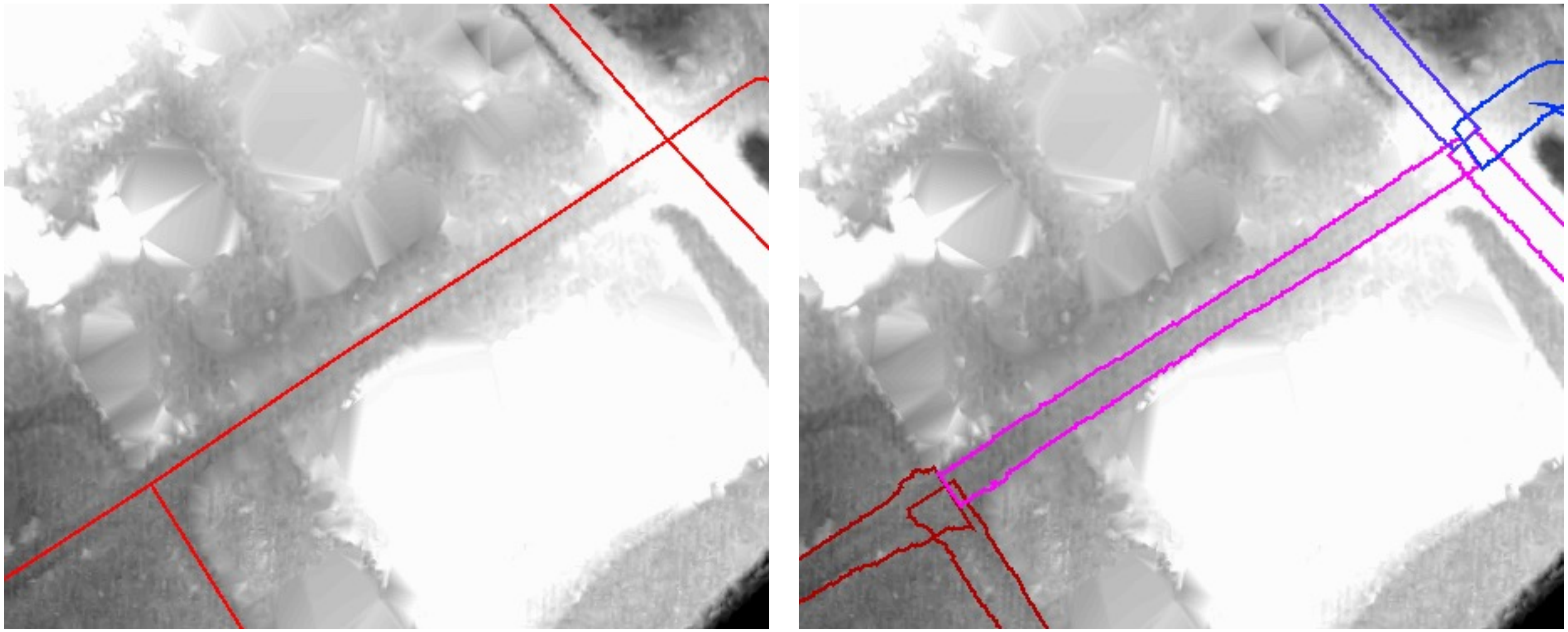




Results re-projected onto DTM,
Resolution *1m*

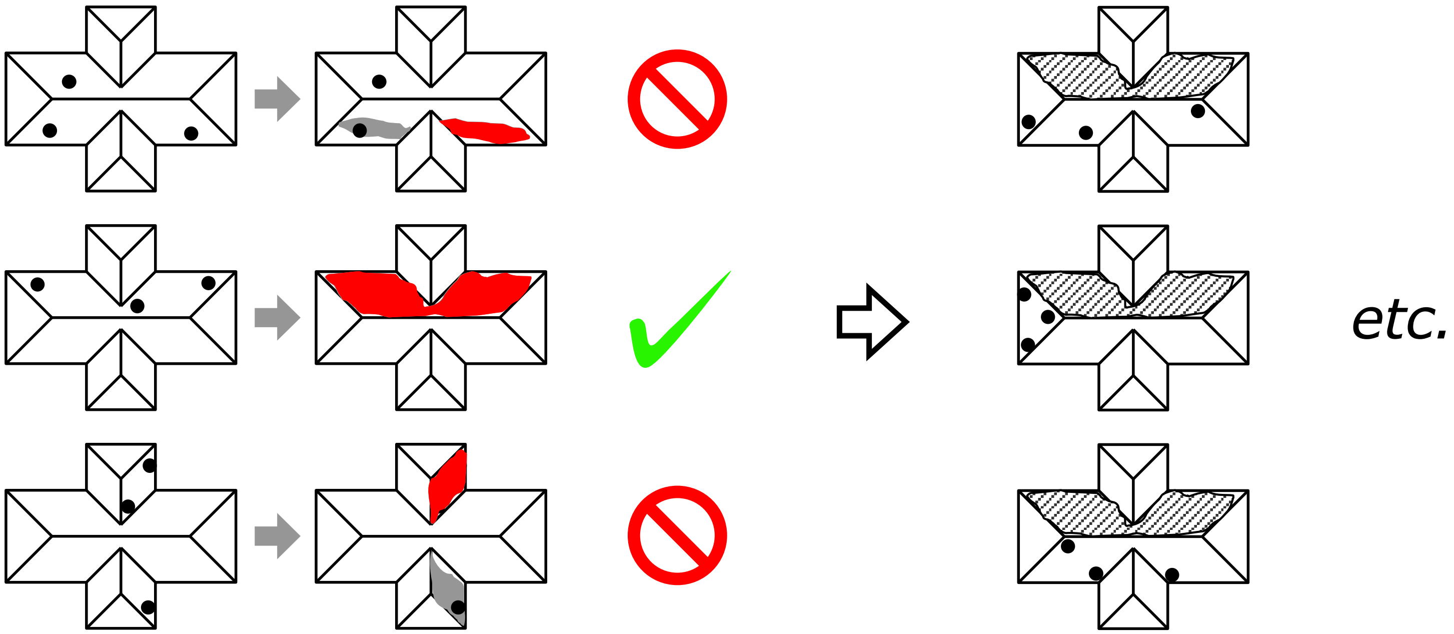
Detailed view

Results, inside built-up region

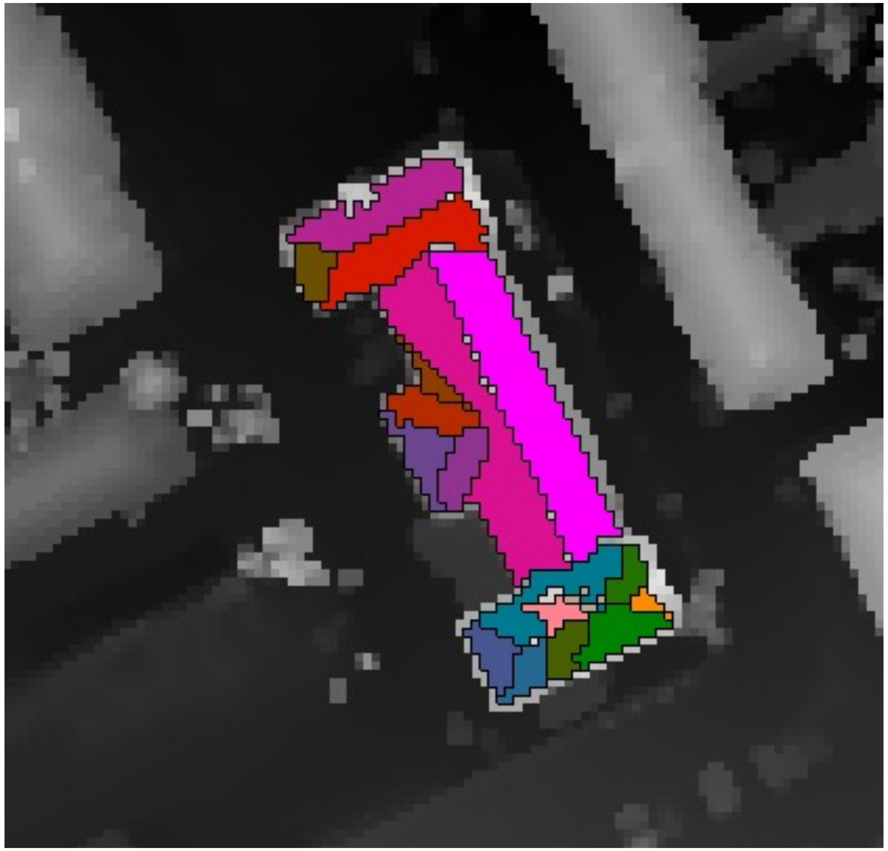


Results, inside built-up region

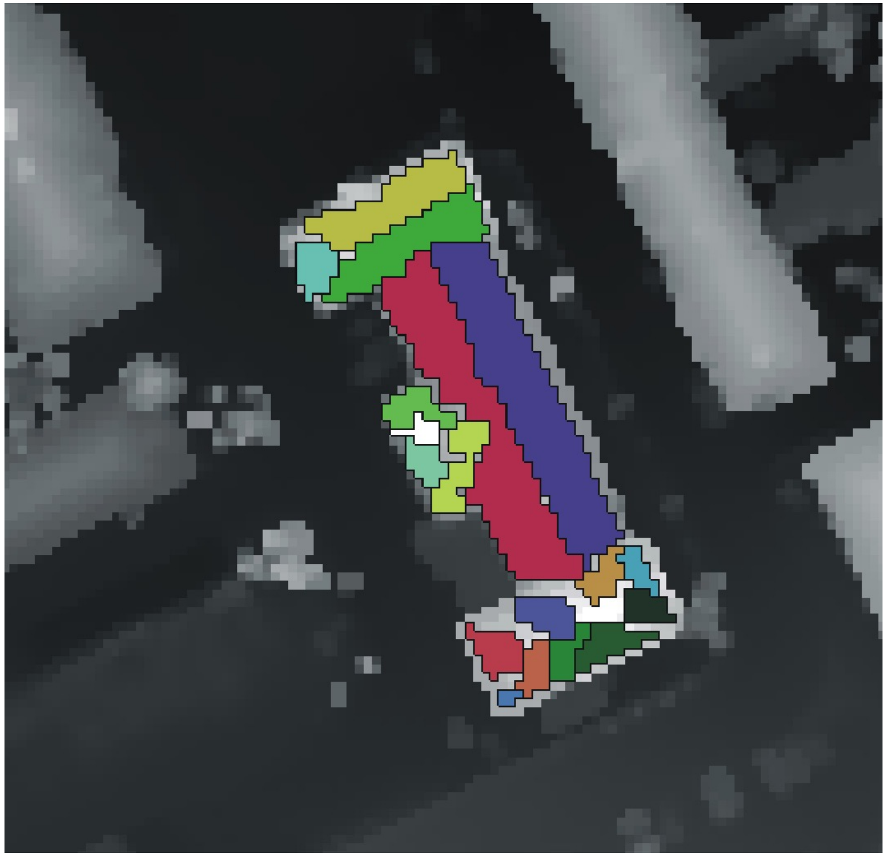
RANSAC example: finding planar segments



Example application: finding roof planes



Region growing



RANSAC

Variants

- Instead of breaking early if the consensus set has at least t compatible observations, always run the loop to the end (i.e., the number of iterations which was computed using the probability calculations). That is, the test with threshold t is simply omitted. In the end, select the solution which has led to the largest number m of compatible observations.

- After RANSAC has terminated, a re-estimation of the parameters is performed (e.g. by a least squares adjustment of the plane, using all points in the consensus set). This will usually improve the RANSAC solution, which was found using the minimal number of observations.

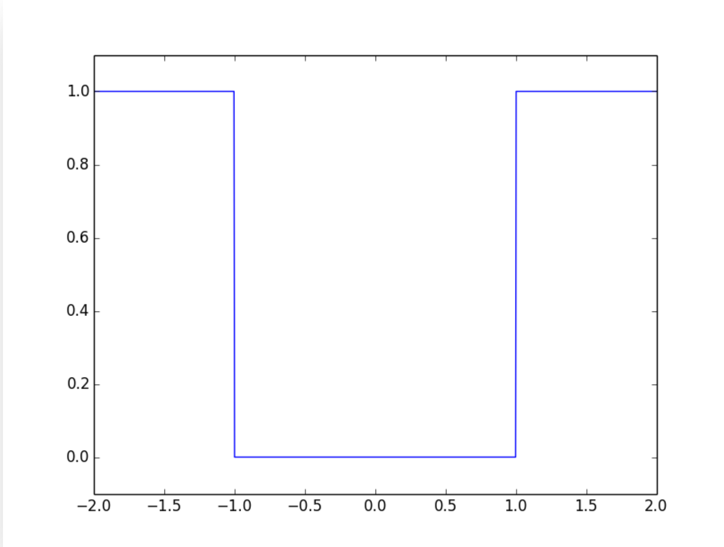
Extension: MSAC

- M-estimator sample consensus
- Recap the “standard” RANSAC approach:
  - It searches for the model which has the largest consensus set
- Problem:
  - Selection of the parameter ε (max. distance to the model)
  - If it is too small, correct models get too few compatible observations
  - If it is too large, erroneous models will get too many compatible observations!
- Modification: “invert” the counting approach, now use a score:
  - +0, if the observation has a distance < ε to the model
  - +1 else.
  - Then, search for the model with the smallest score
- (Think of “counting outliers” instead of “inliers”).

Extension: MSAC

- Formally:

$$\rho(M,\mathbf{p}) = \begin{cases} 0 & \mathrm{dist}(M,\mathbf{p}) < \varepsilon \\ 1 & \mathrm{dist}(M,\mathbf{p}) \geq \varepsilon \end{cases}$$



- Search for the model M*, which has the minimum score:

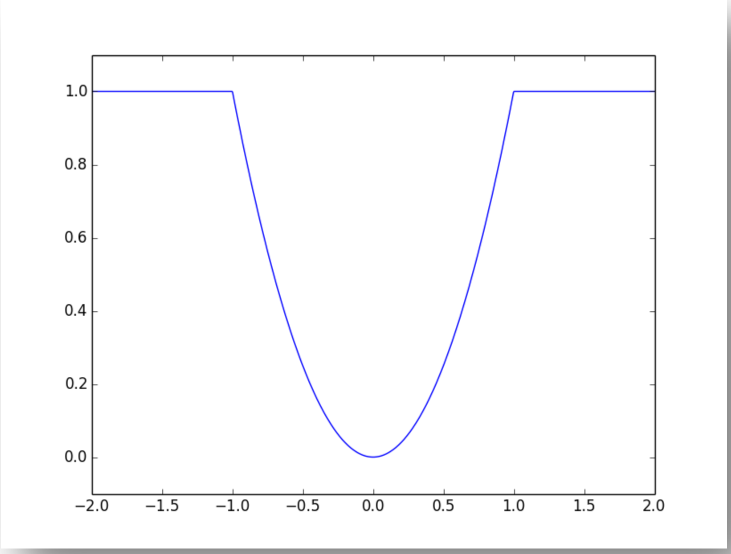
$$M^* = \arg\min_{M} \sum_{\mathbf{p} \in \mathrm{points}} \rho(M,\mathbf{p})$$

- So far, while we changed the formulation, our “new” algorithm leads to exactly the same results. But now… →

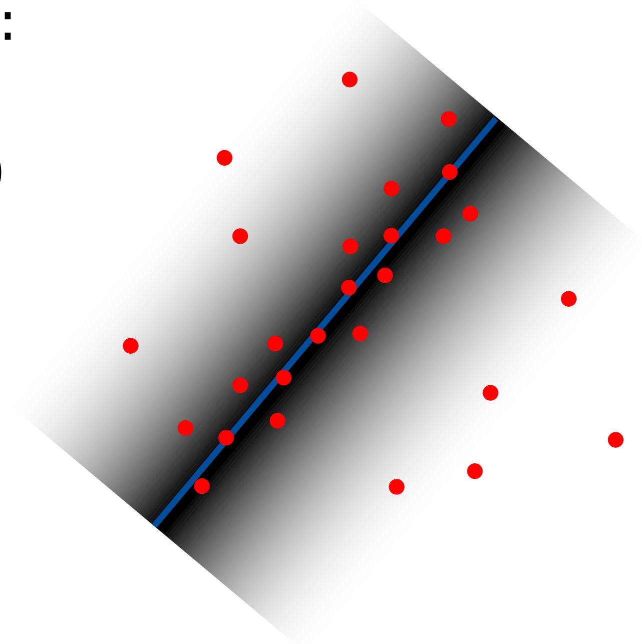
Extension: MSAC

- Modify the score function:

$$\rho(M,\mathbf{p}) = \begin{cases} [\mathrm{dist}(M,\mathbf{p})]^2 & \mathrm{dist}(M,\mathbf{p}) < \varepsilon \\ \varepsilon^2 & \mathrm{dist}(M,\mathbf{p}) \geq \varepsilon \end{cases}$$



- This leads to the following:
  - Outliers are still punished by constant costs: ε^2 (previously, this was 1)
  - Observations which fit to the model (inliers) now contribute to the score depending on their actual (squared) distance
  - Thus, good models can be found even if ε has been set too large. (Or: when facing the same number of compatible points, the closer model is preferred.)



References

- Vosselman, Maas, Airborne and Terrestrial Laser Scanning, Whittles Publishing, 2010. [Overview, Chap. 2]
- Fischler, Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, Comm. of the ACM, 24(6), June 1981. [RANSAC]
- Torr, Zisserman, Robust computation and parameterization of multiple view relations, ICCV'98, 1998. [MSAC]
- C. Hatger and C. Brenner: Extraction of Road Geometry Parameters form Laser Scanning and Existing Databases, Proc. Workshop 3-D reconstruction from airborne laserscanner and InSAR data, International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XXXIV, Maas, H.-G., Vosselman, G., Streilein, A. (eds.), Dresden, 2003 [Road extraction example]