Digital Image - Lecture 07

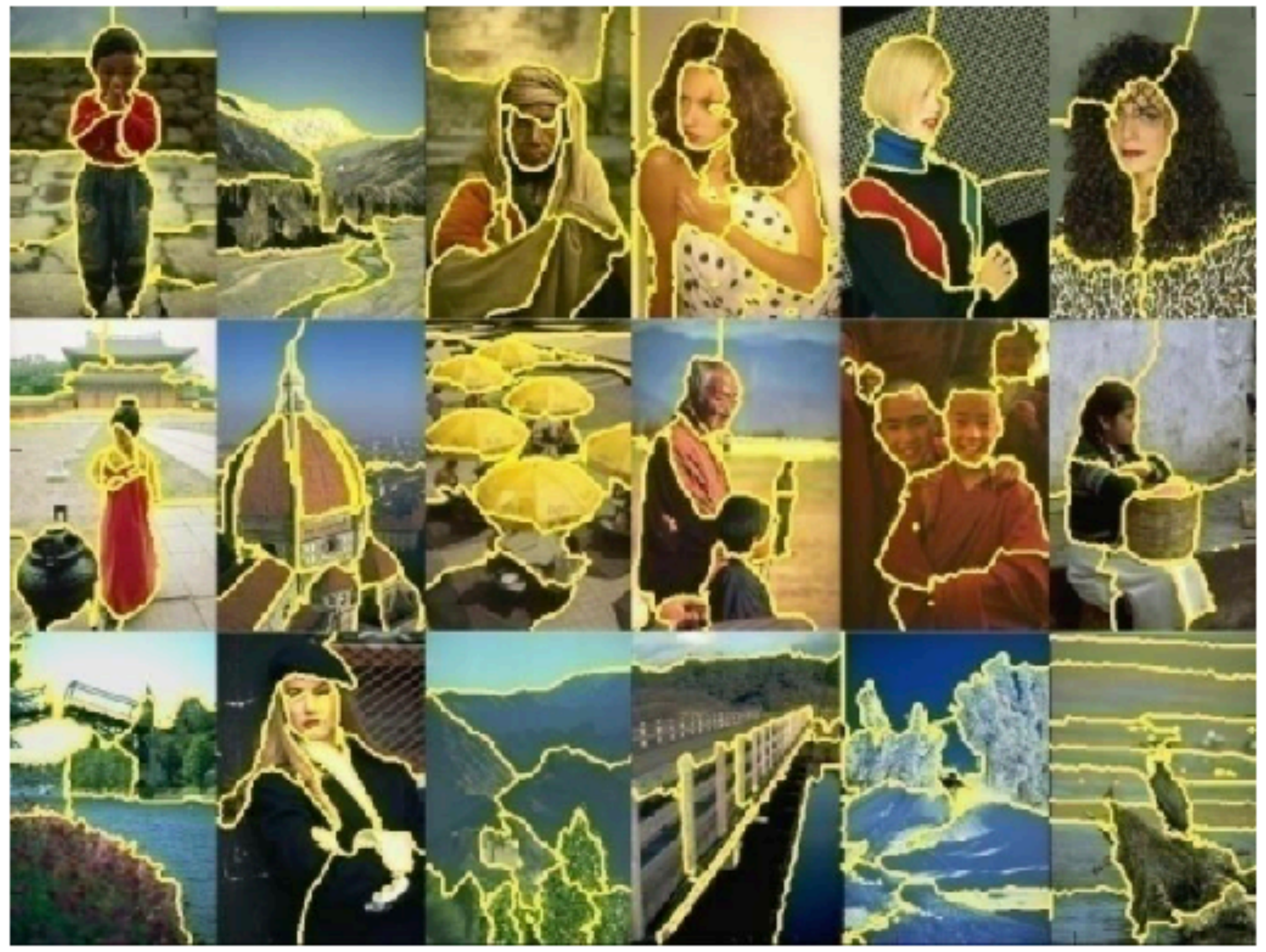
Image Segmentation
Part I

Torsten Sattler

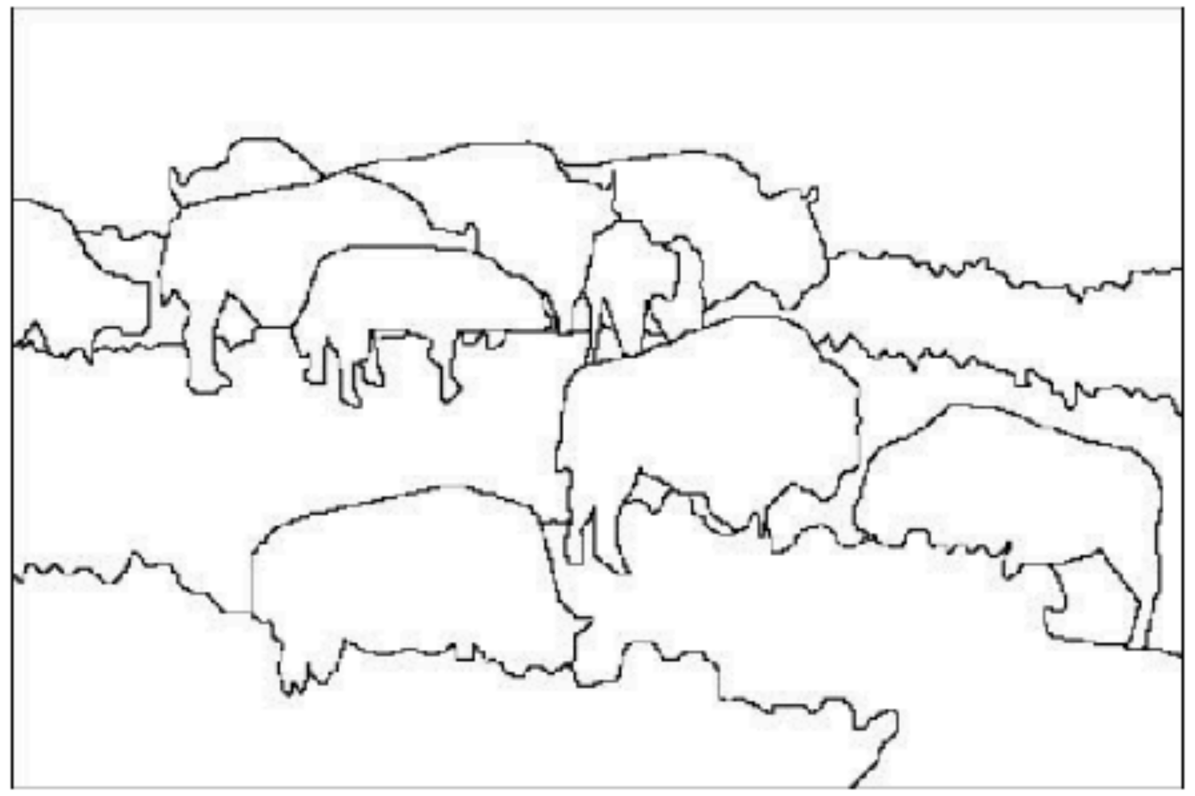
Czech Institute of Informatics, Robotics and Cybernetics
Czech Technical University in Prague

slides adapted from Václav Hlaváč and Bastian Leibe

What Is Image Segmentation?



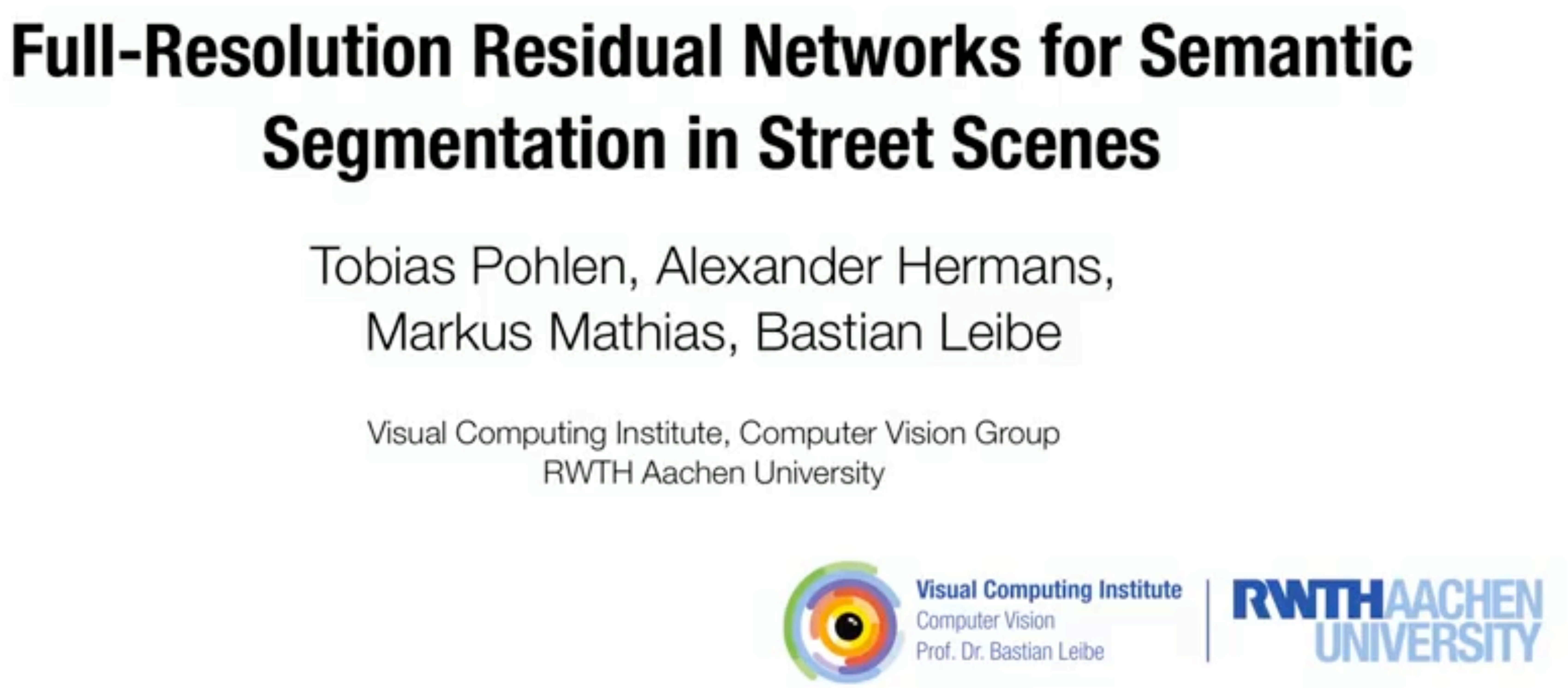
Image, courtesy Ondřej Drbohlav



Goal: **segment** image **into** (semantically) meaningful **regions**

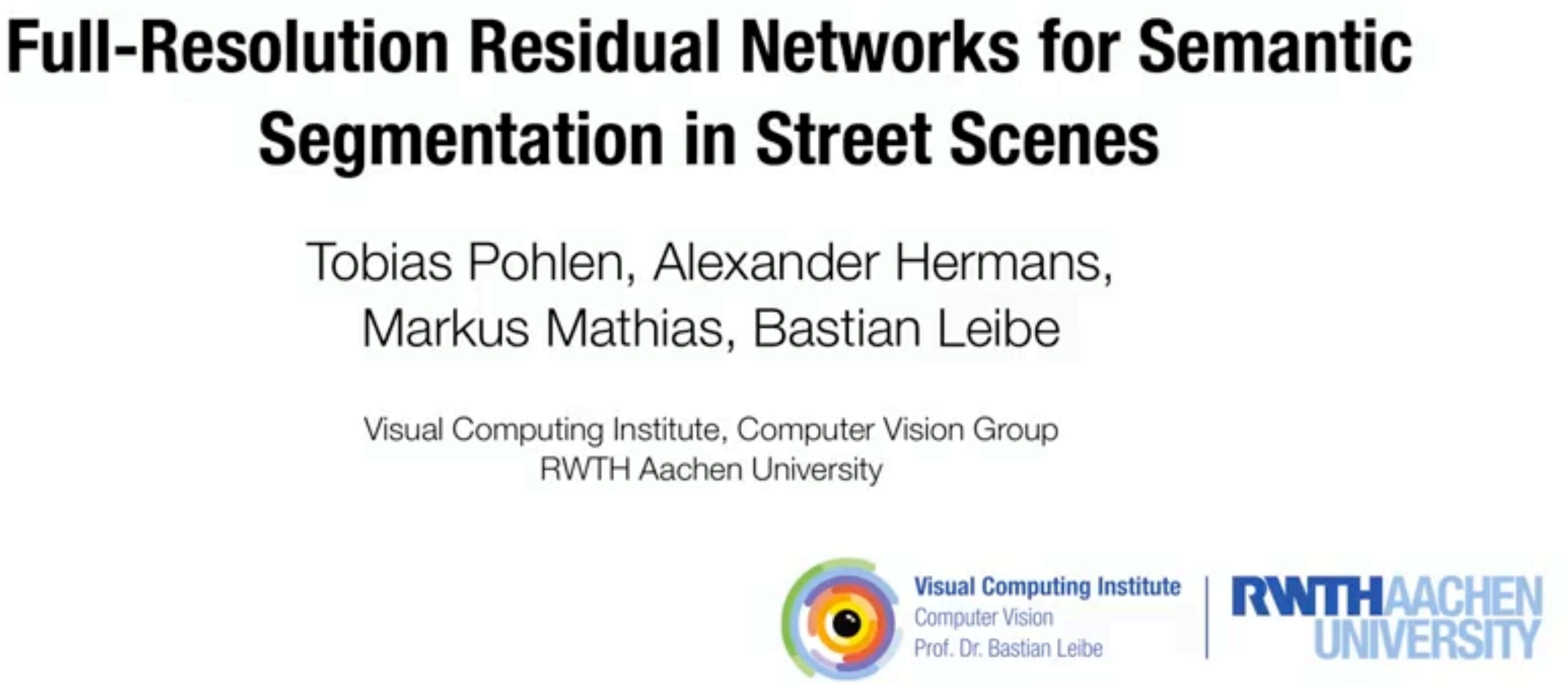
slide credit: Václav Hlaváč, Bastian Leibe, Kristen Grauman, Svetlana Lazebnik

Example: Semantic Segmentation



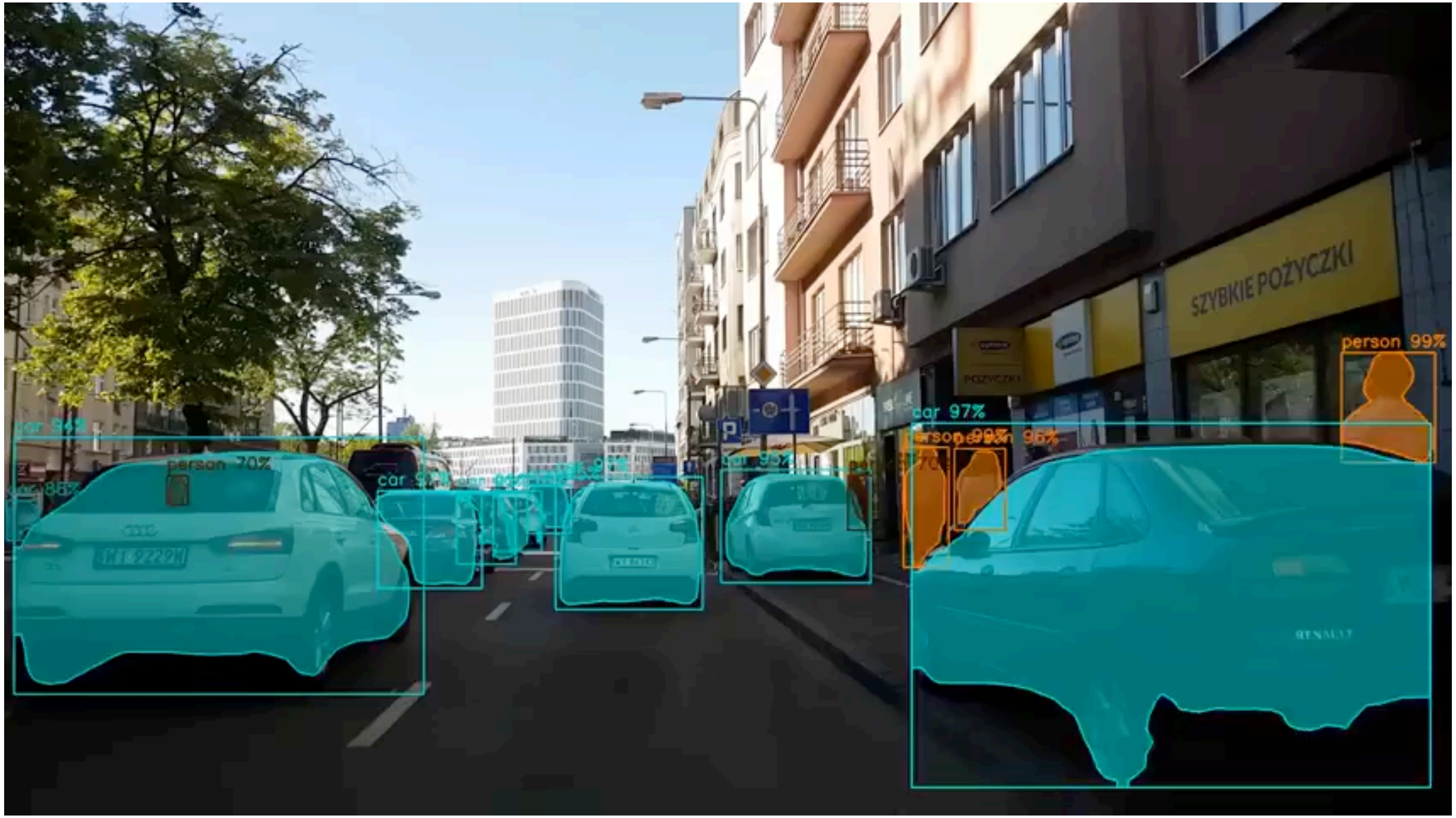
[Pohlen, Hermans, Mathias, Leibe, Full-Resolution Residual Networks for Semantic Segmentation in Street Scenes, CVPR 2017] video link

Example: Semantic Segmentation



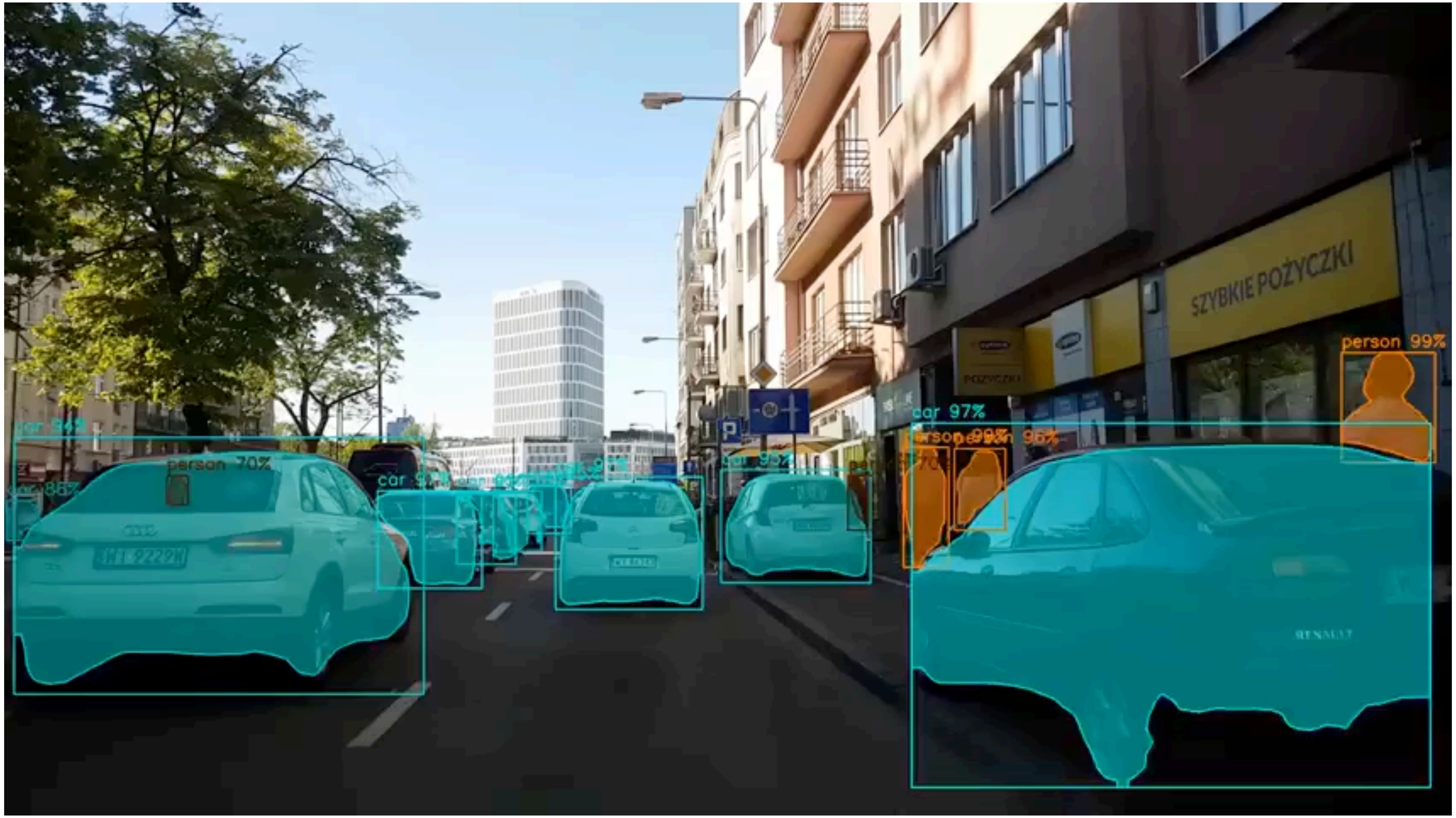
[Pohlen, Hermans, Mathias, Leibe, Full-Resolution Residual Networks for Semantic Segmentation in Street Scenes, CVPR 2017] video link

Example: Instance-Level Segmentation



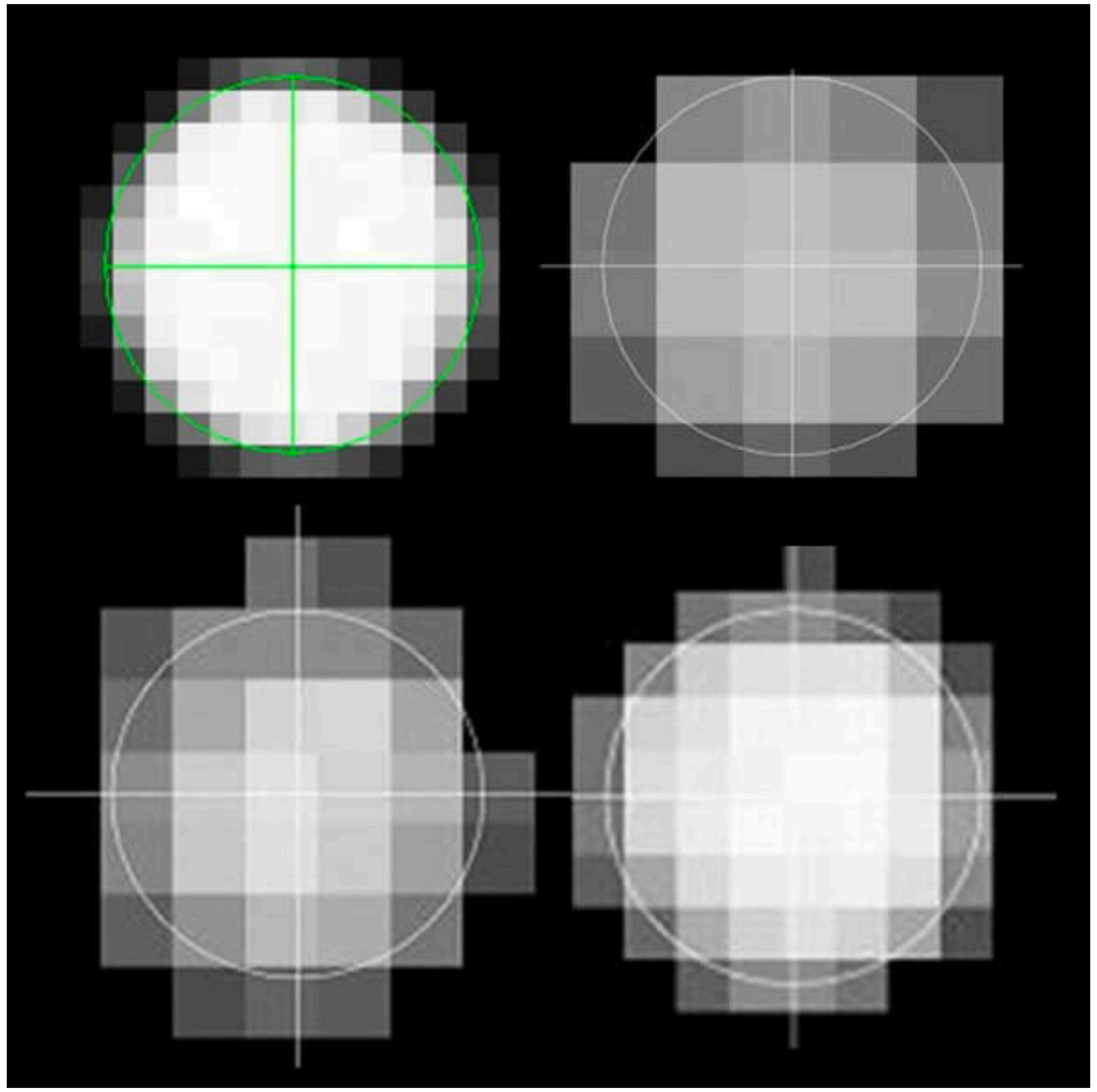
video link

Example: Instance-Level Segmentation



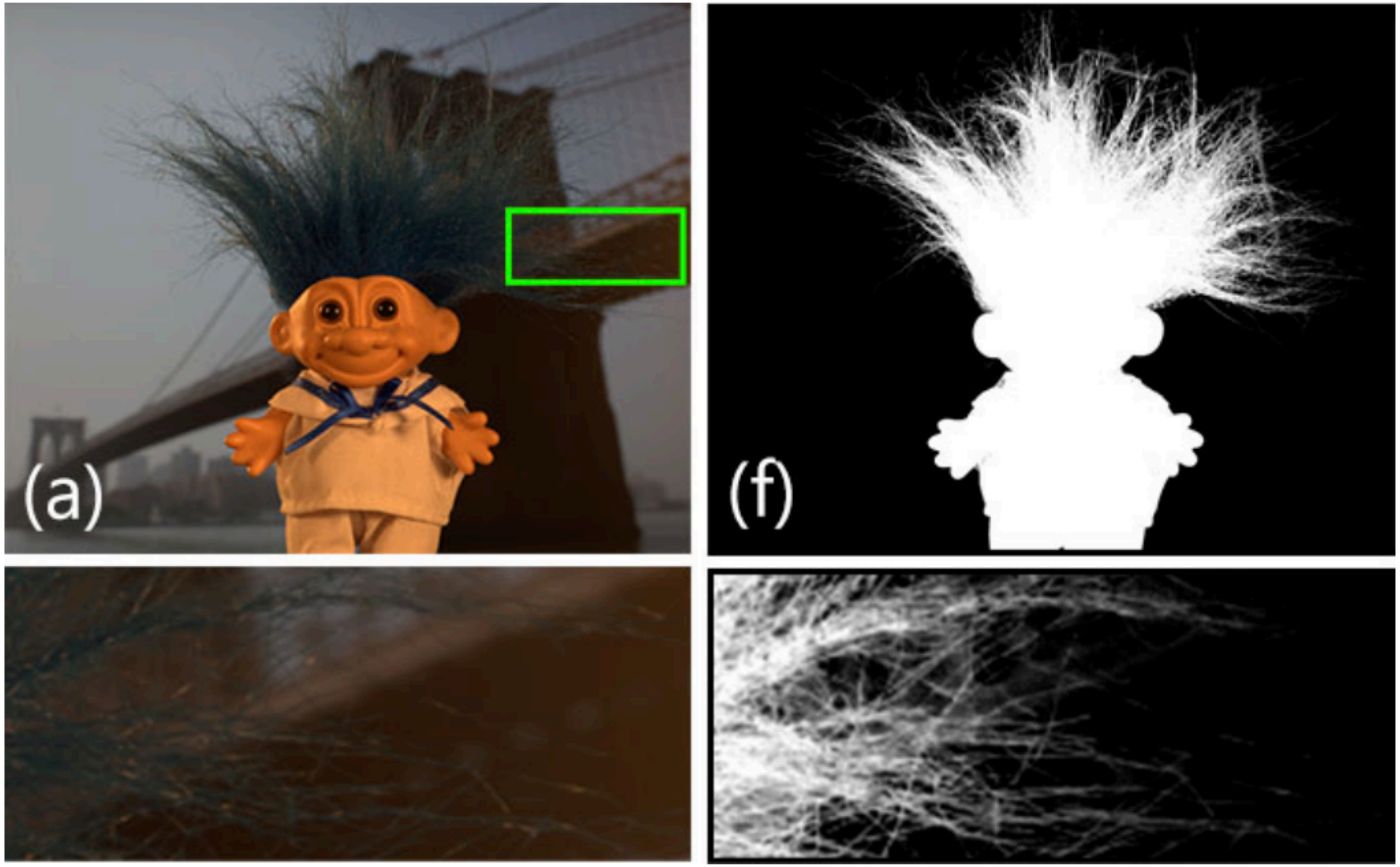
video link

Example: Motion Capture



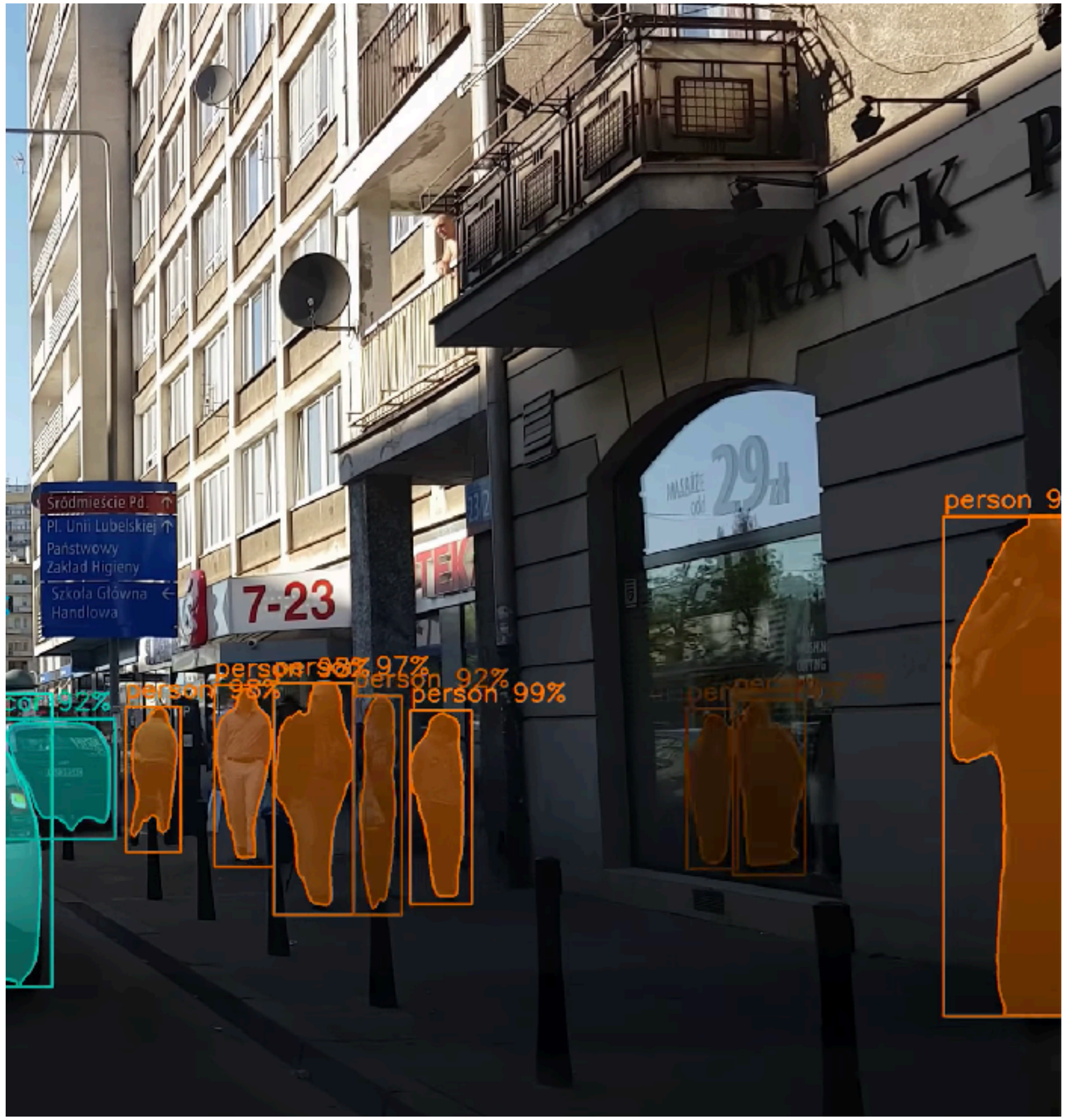
images taken from Vicon website

Example: Foreground Background Segmentation



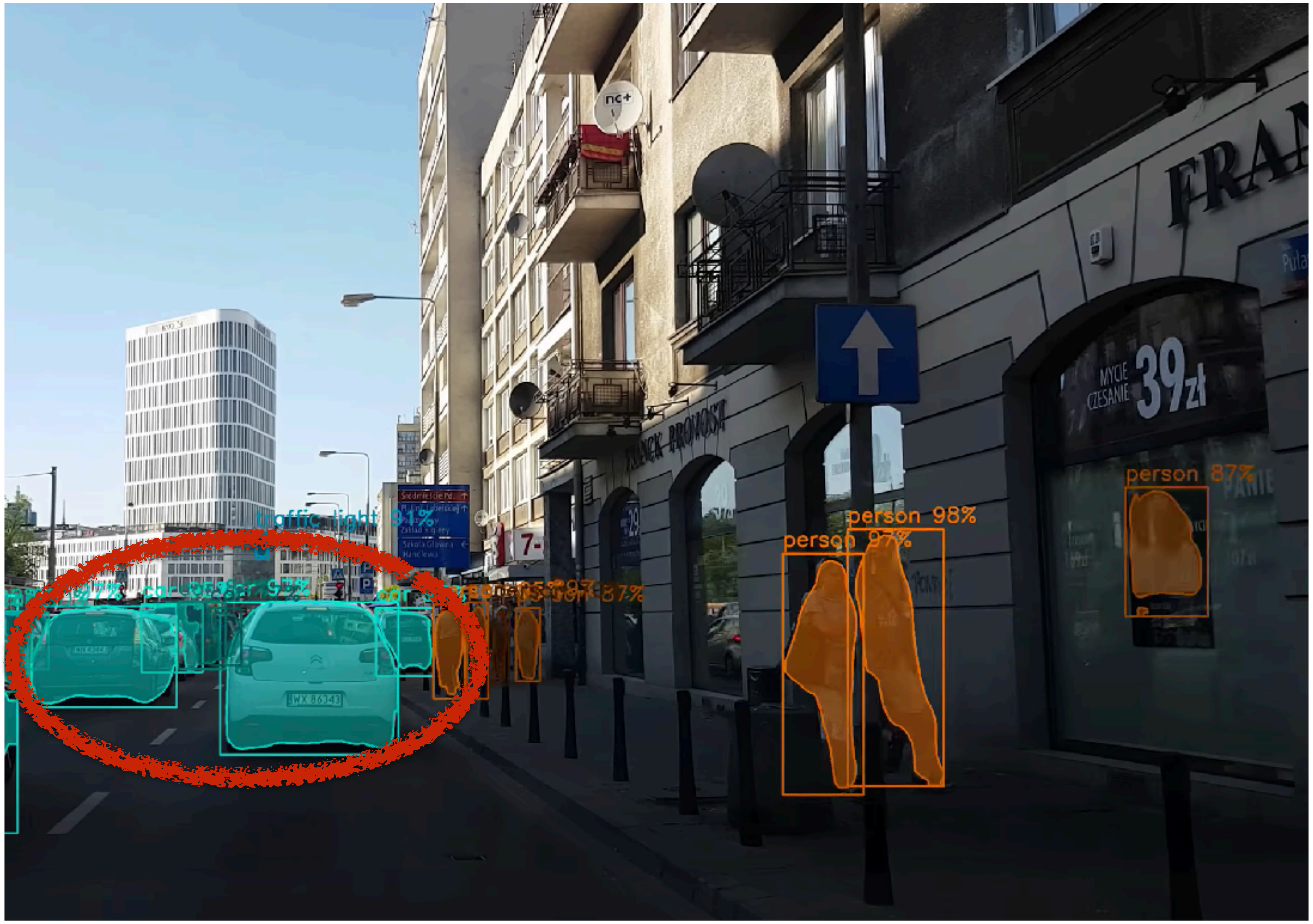
[Yağız Aksoy, Tunç Ozan Aydın, Marc Pollefeys, Designing Effective Inter-Pixel Information Flow for Natural Image Matting, CVPR 2017]

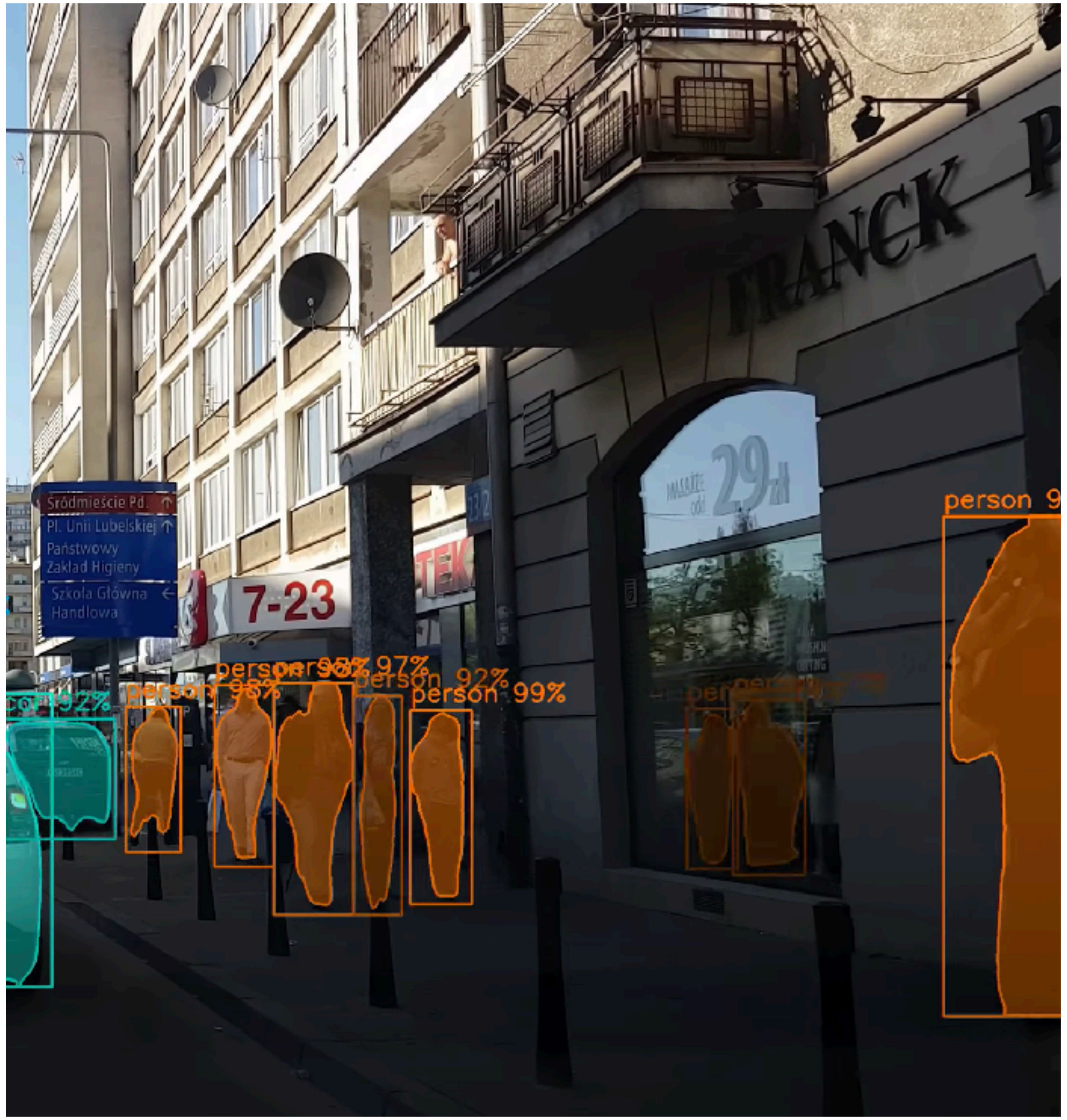
Image Segmentation Can Be Hard



video link

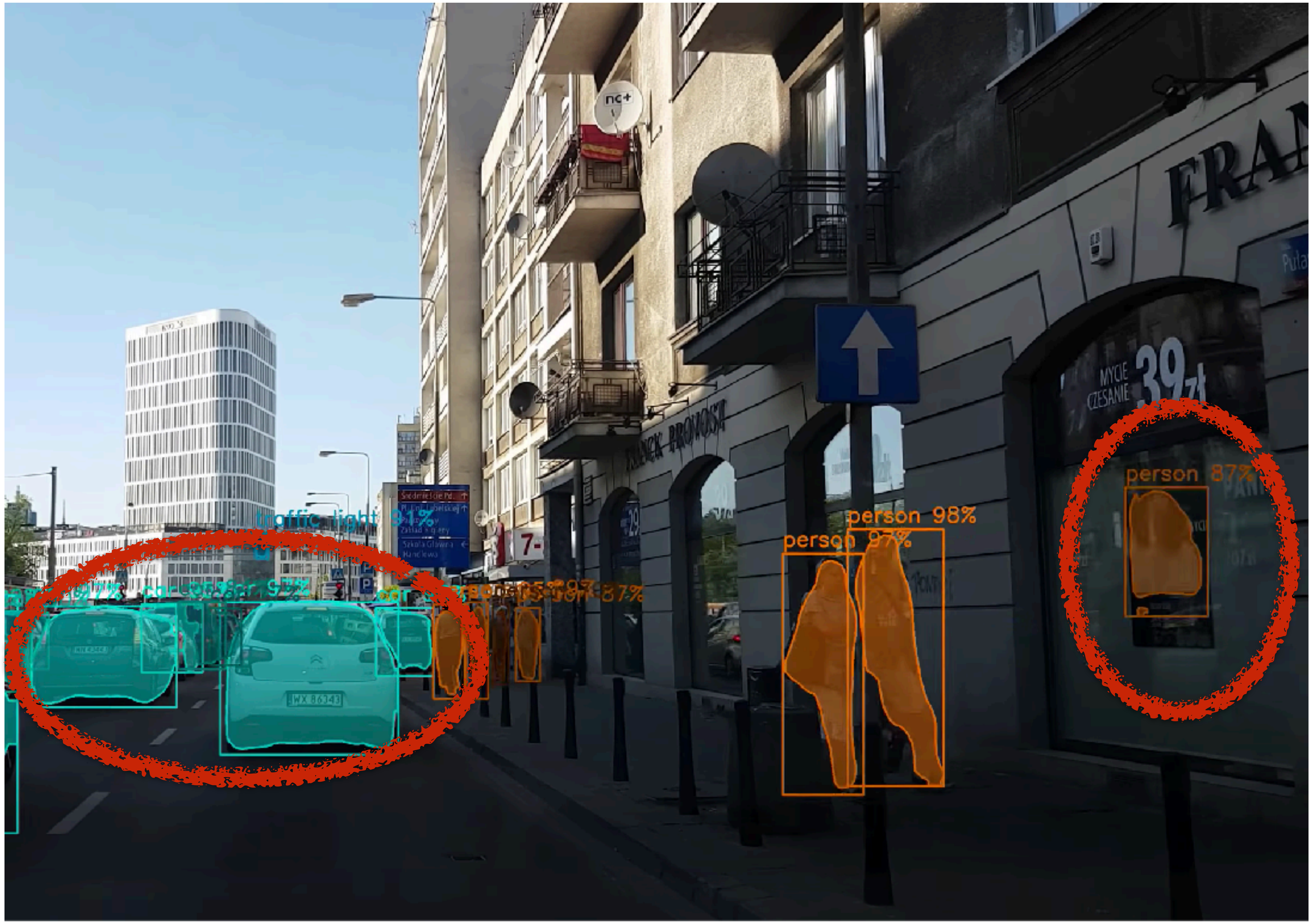
Image Segmentation Can Be Hard

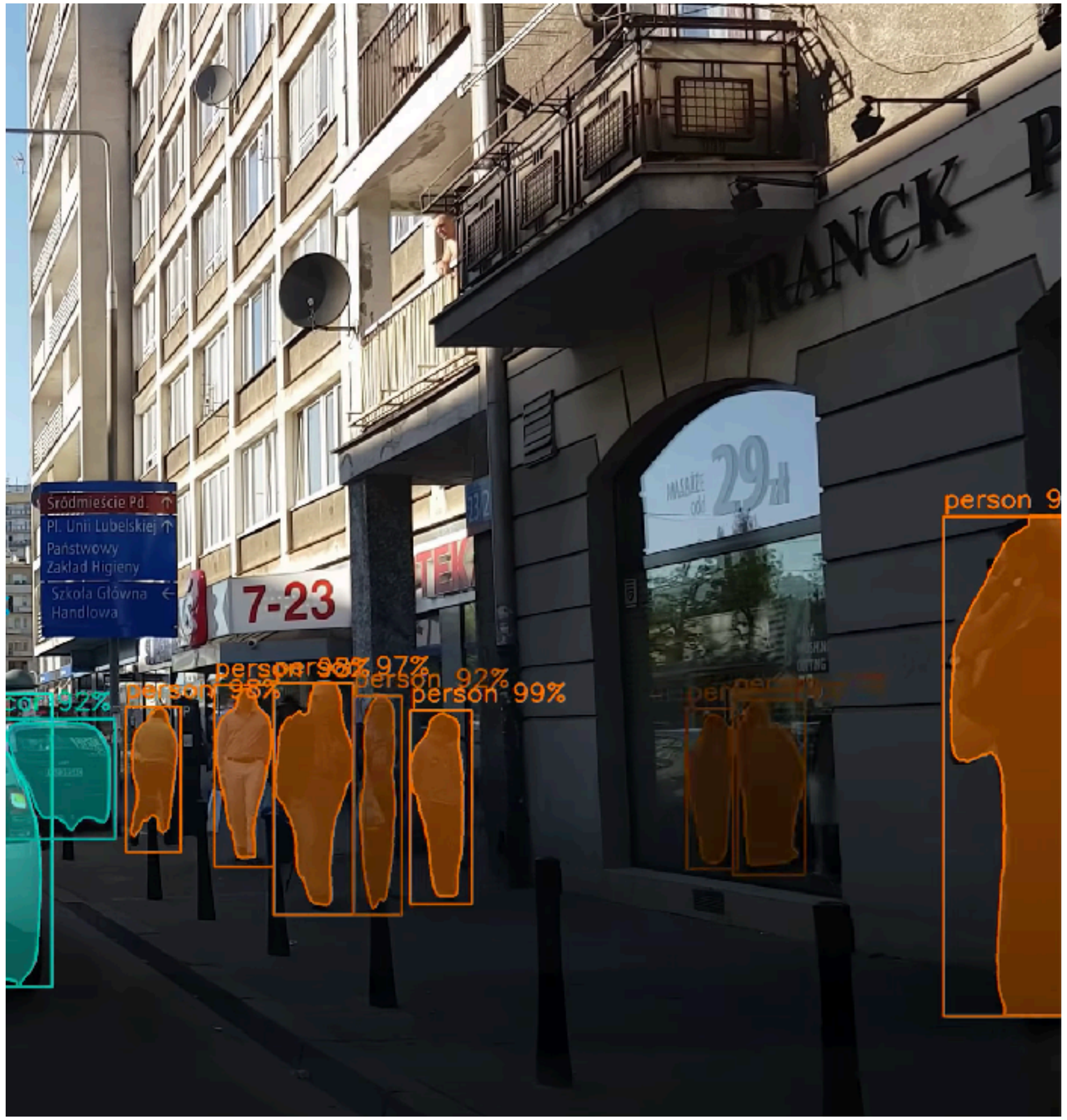




video link

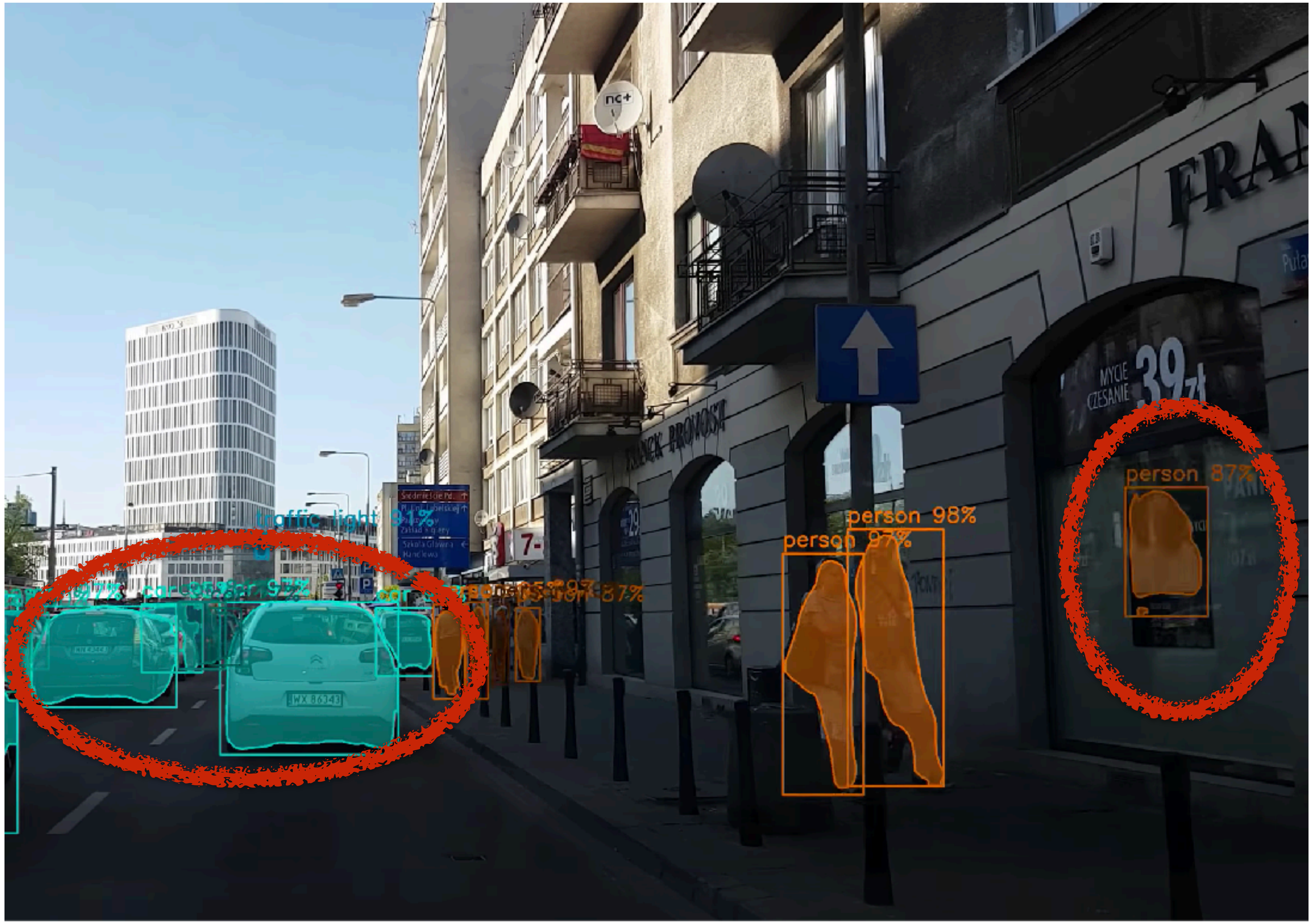
Image Segmentation Can Be Hard





video link

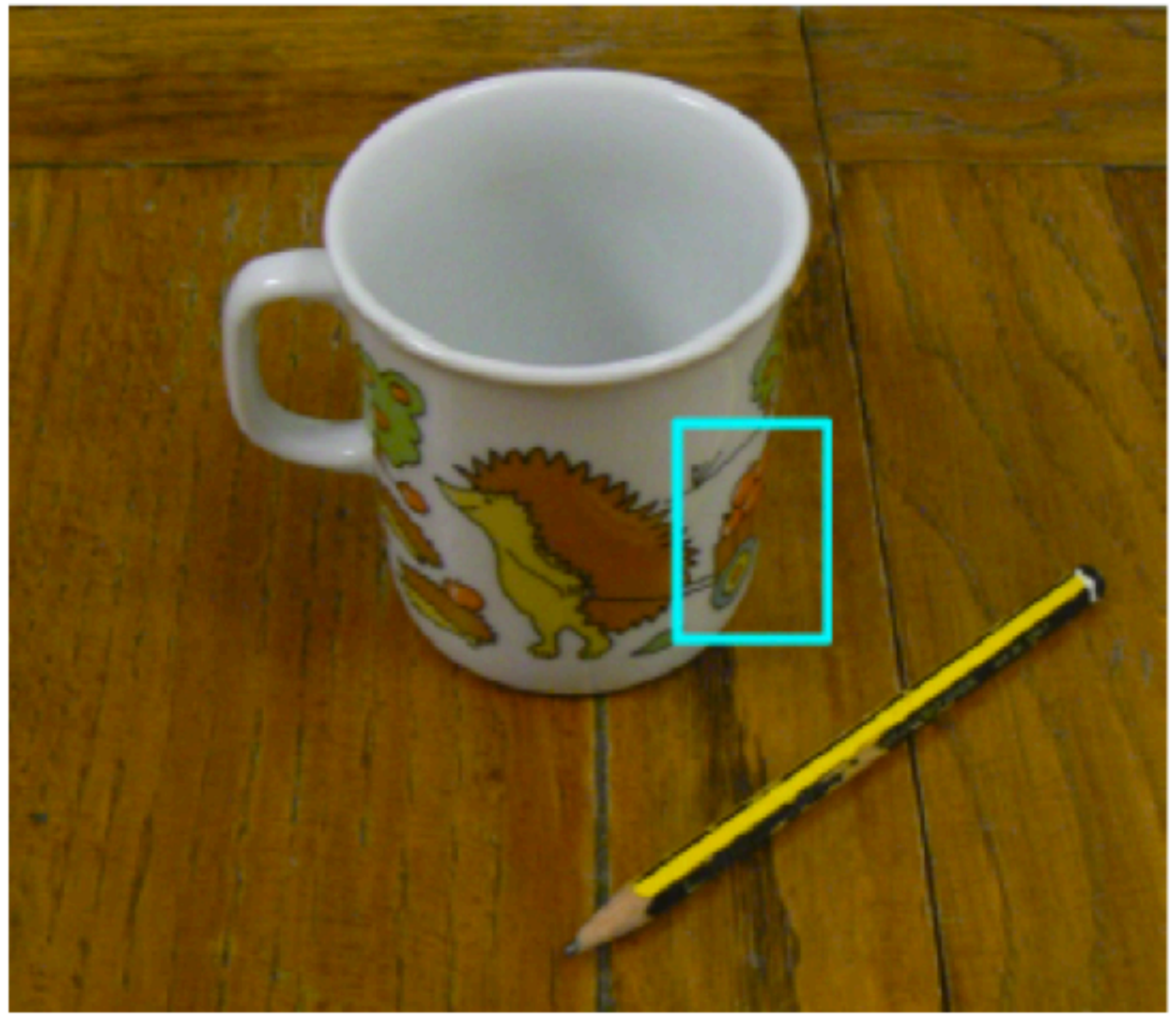
Image Segmentation Can Be Hard

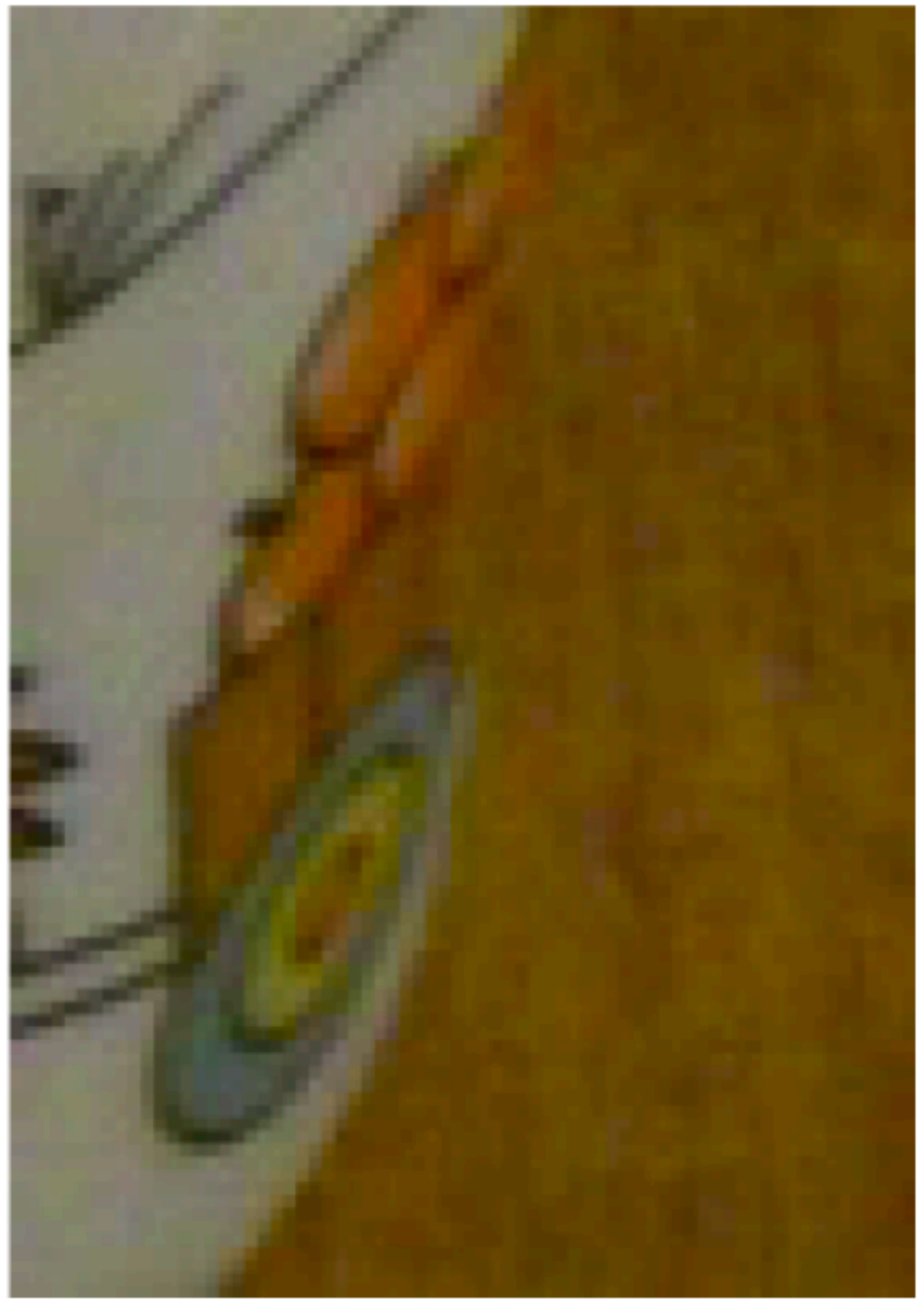


video link

Image Segmentation Can Be Hard

Image, courtesy Ondřej Drbohlav

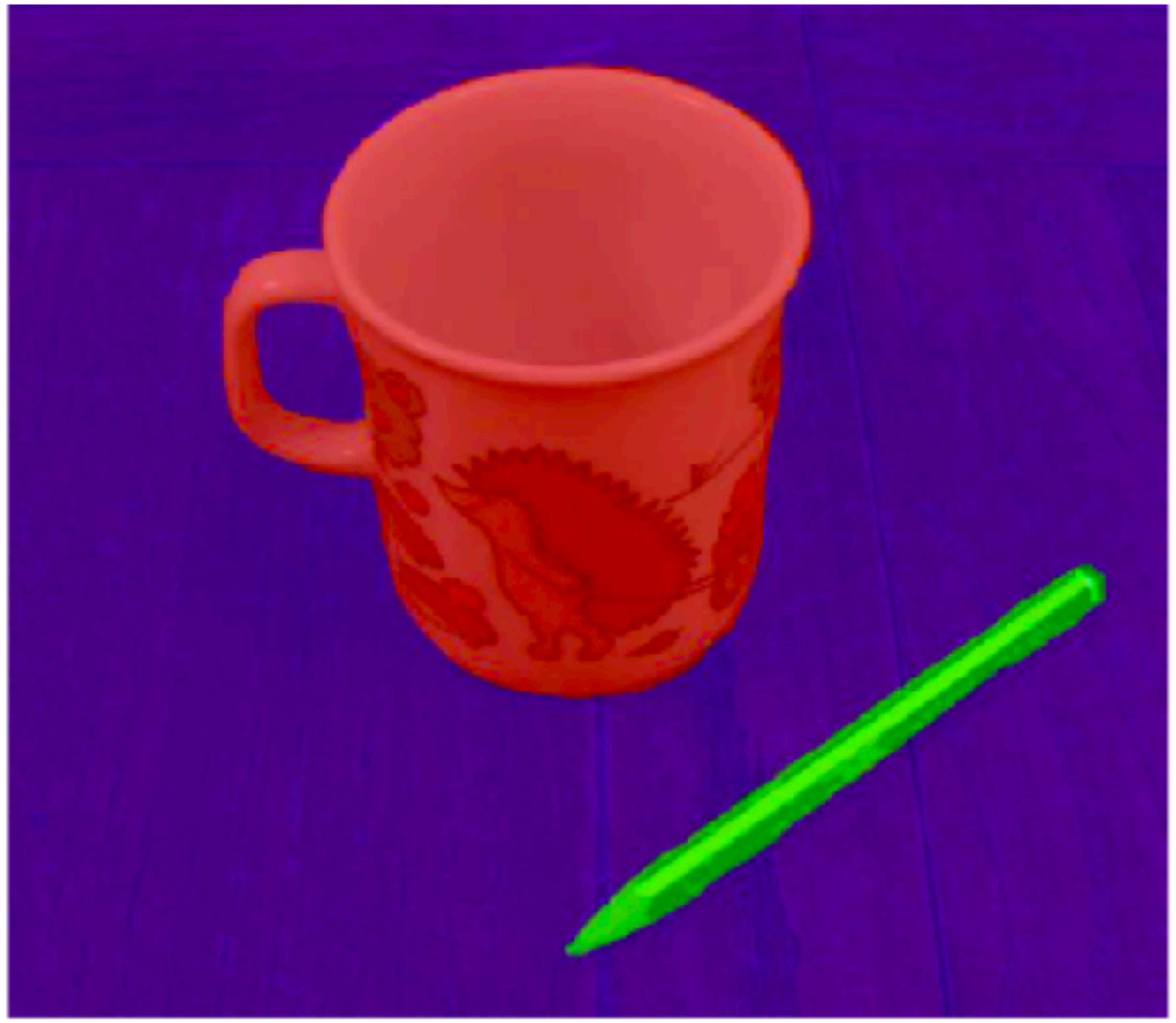




- Cannot use color to distinguish between border of cup and background
- Need some semantic understanding of what a “cup” is

slide credit: Václav Hlaváč

Image Segmentation As Grouping



Image, courtesy Ondřej Drbohlav

Goal: **group pixels** that belong together into **regions**

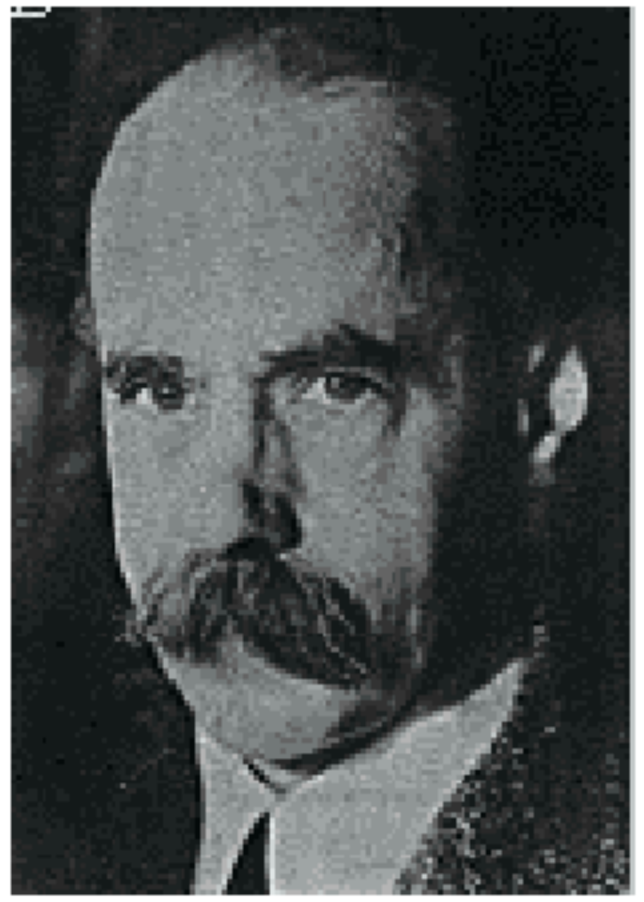
slide credit: Václav Hlaváč

Inspiration from Humans? - The Gestalt School

- **Grouping of elements is key to human visual perception**
- Founding publication by Max Wertheimer (born in Prague) in 1912
- Gestalt theory was meant to be generally applicable, but main tenets almost exclusively derived from observations of visual perception
- Psychologists showed that human visual systems seems predisposed to group elements

"I stand at the window and see a house, trees, sky. Theoretically I might say there were 327 brightnesses and nuances of colour. Do I have "327"? No. I have sky, house, and trees."

Max Wertheimer
(1880-1943)



Untersuchungen zur Lehre von der Gestalt, *Psychologische Forschung*, Vol. 4, pp. 301-350, 1923

slide credit: Václav Hlaváč, Bastian Leibe

Inspiration from Humans? - The Gestalt School

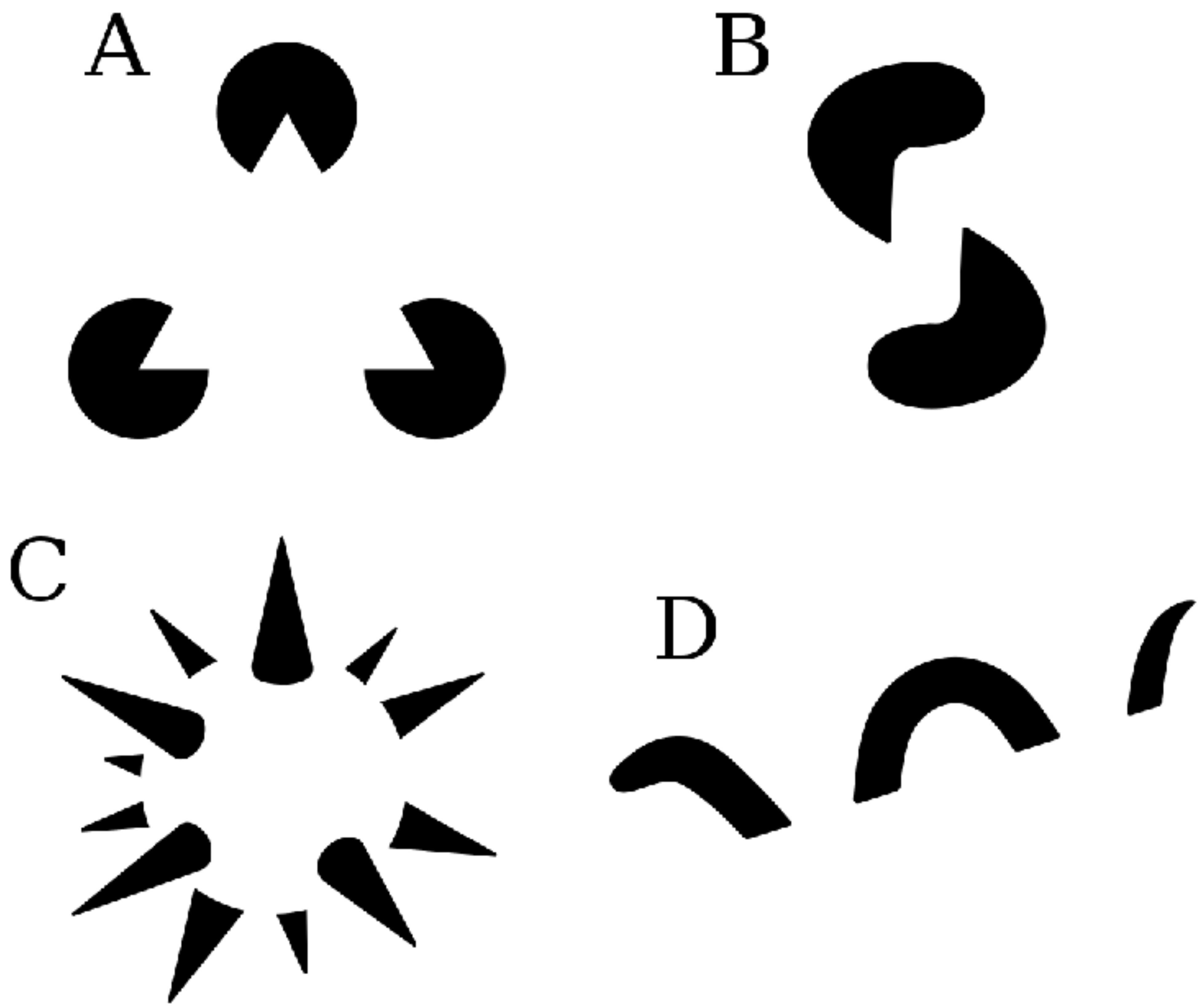
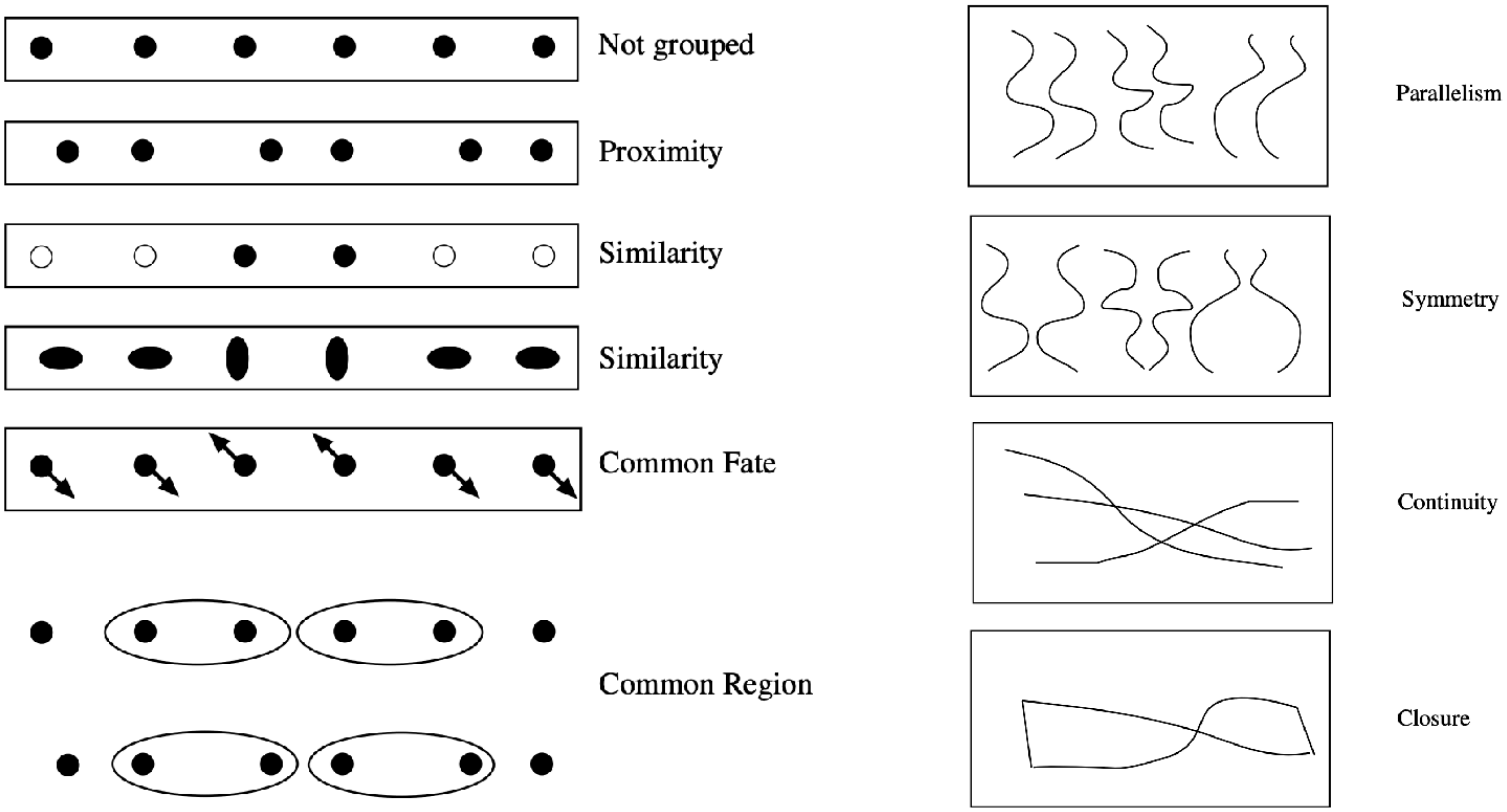


image source: Wikipedia

- **Gestalt**: configuration of elements such that whole is greater than sum of parts
- Properties / features derived from relationship between elements
- https://en.wikipedia.org/wiki/Gestalt_psychology

slide credit: Václav Hlaváč, Bastian Leibe

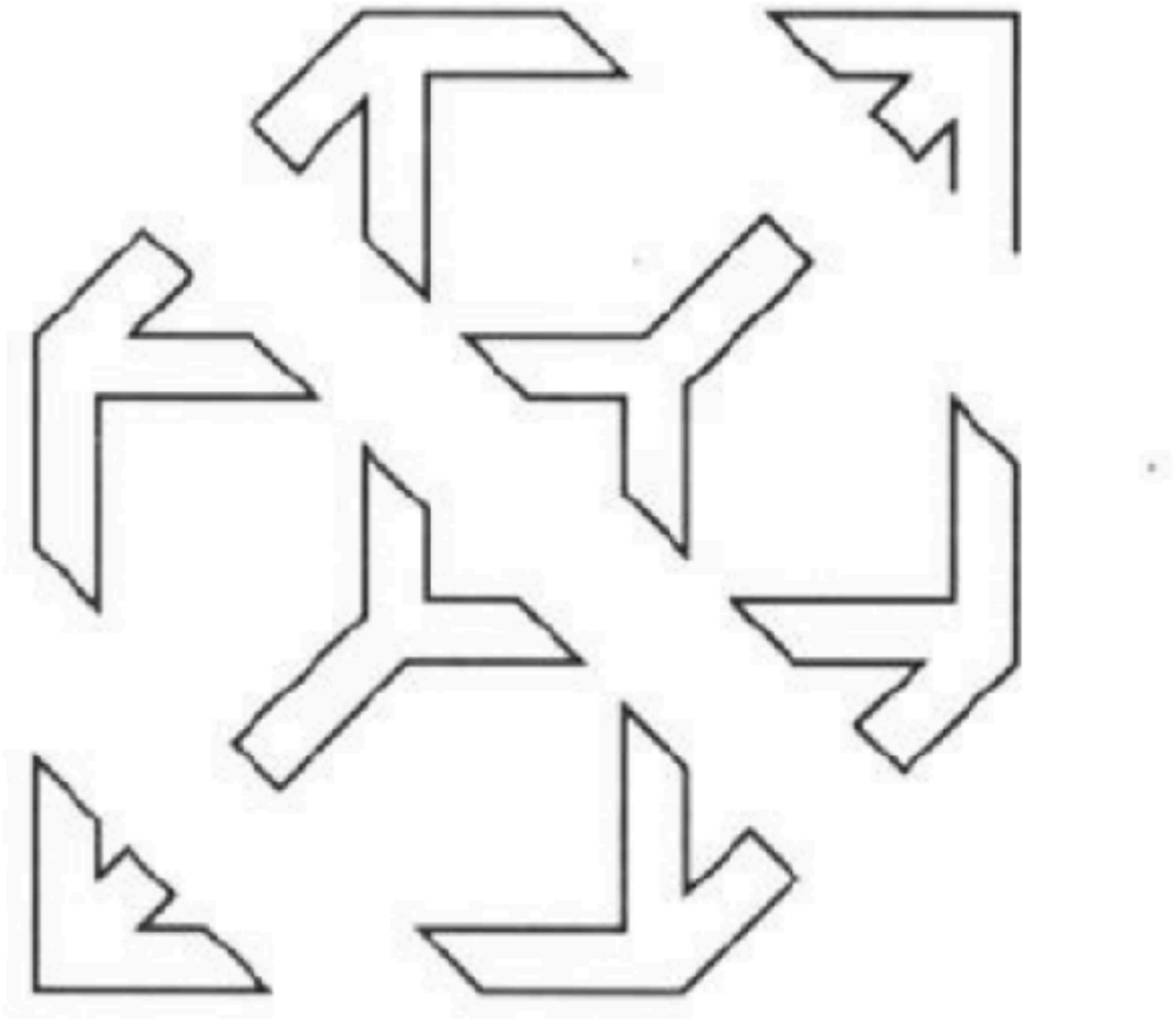
Gestalt Grouping Principles



slide credit: Václav Hlaváč, Bastian Leibe

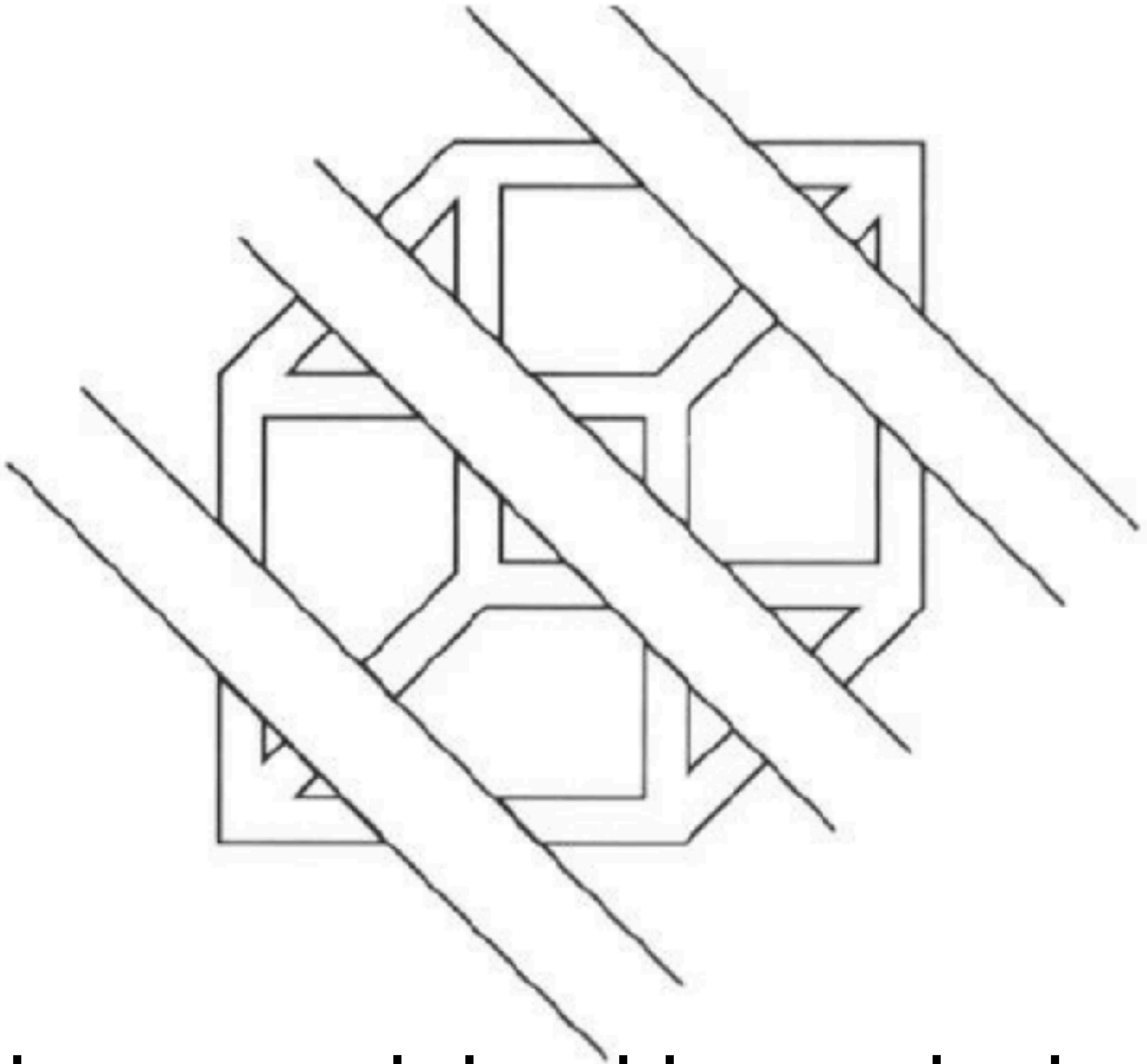
image source: D. Forsyth, J. Ponce, Computer Vision - A Modern Approach, 2nd edition, Pearson, 2011

Continuity Through Occlusions



slide credit: Bastian Leibe

Continuity Through Occlusions

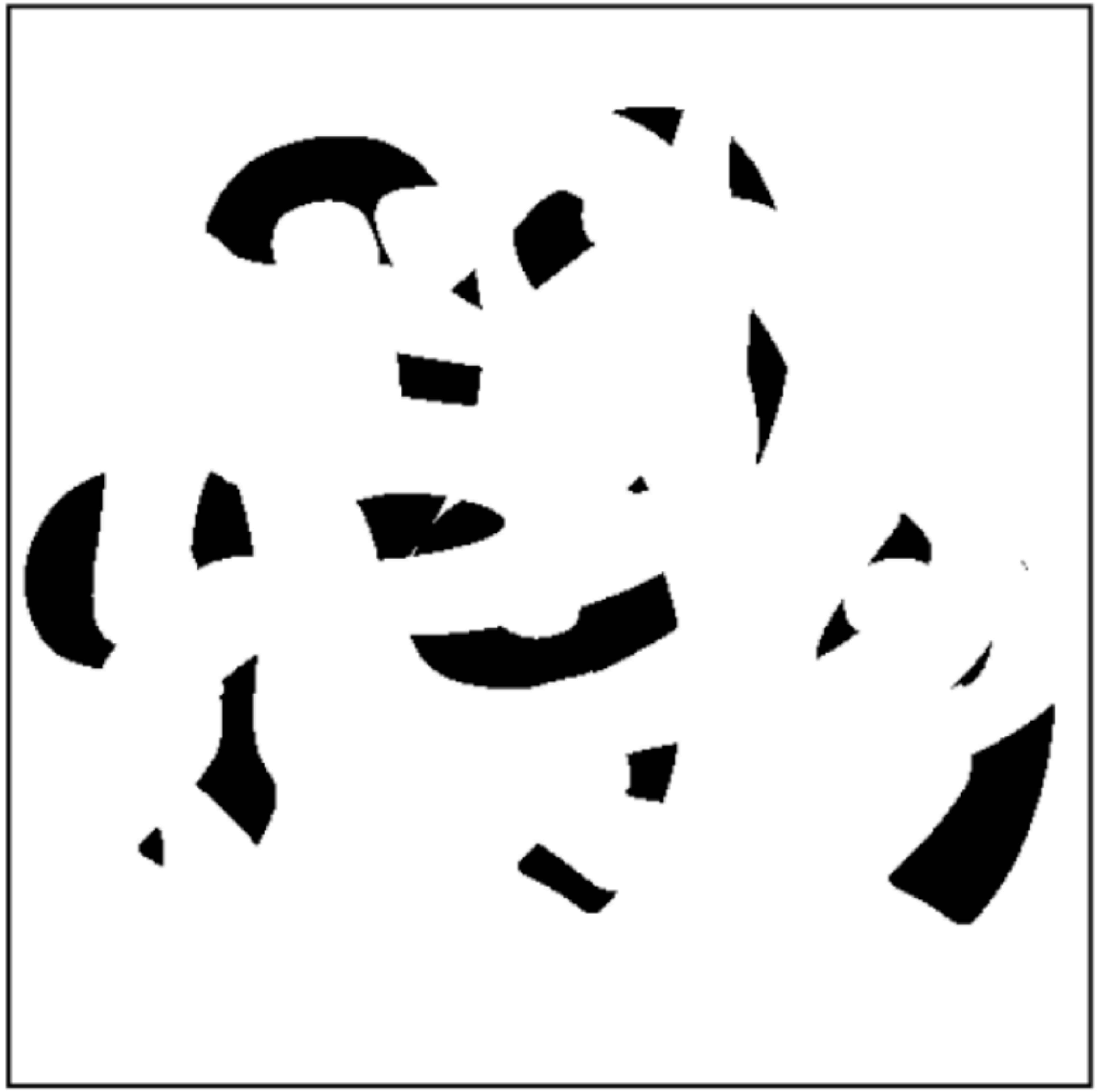


Shape explained by occlusions

slide credit: Bastian Leibe

Continuity Through Occlusions

What do you see?



slide credit: Václav Hlaváč, Bastian Leibe

image source: D. Forsyth, J. Ponce, Computer Vision - A Modern Approach, 2nd edition, Pearson, 2011

Continuity Through Occlusions

What do you see?

slide credit: Václav Hlaváč, Bastian Leibe

image source: D. Forsyth, J. Ponce, Computer Vision - A Modern Approach, 2nd edition, Pearson, 2011

Grouping Can Be Very Hard

What do you see?

Picture by R. C. James

slide credit: Václav Hlaváč, Bastian Leibe

Grouping Can Be Very Hard

What do you see?

Picture by R. C. James

How to teach Gestalt principles to a machine?

slide credit: Václav Hlaváč, Bastian Leibe

Lecture Overview

- A simple approach to segmentation: **(intensity) thresholding**

slide credit: Václav Hlaváč

Lecture Overview

- A simple approach to segmentation: **(intensity) thresholding**
- Segmentation based on spatial coherence: **edge-based segmentation**, **region growing**

slide credit: Václav Hlaváč

Lecture Overview

- A simple approach to segmentation: **(intensity) thresholding**
- Segmentation based on spatial coherence: **edge-based segmentation**, **region growing**
- Segmentation as a **clustering** problem: **k-means clustering**, **mean-shift clustering**

slide credit: Václav Hlaváč

Lecture Overview

- A simple approach to segmentation: **(intensity) thresholding**
- Segmentation based on spatial coherence: **edge-based segmentation**, **region growing**
- Segmentation as a **clustering** problem: **k-means clustering**, **mean-shift clustering**
- Segmentation as a statistical (unsupervised) learning problem: **expectation maximization (EM) algorithm**

slide credit: Václav Hlaváč

Lecture Overview

- A simple approach to segmentation: **(intensity) thresholding**
- Segmentation based on spatial coherence: **edge-based segmentation**, **region growing**
- Segmentation as a **clustering** problem: **k-means clustering**, **mean-shift clustering**
- Segmentation as a statistical (unsupervised) learning problem: **expectation maximization (EM) algorithm**
- Next lecture: **graph-based segmentation**, **supervised learning with neural networks** (if time and interest)

slide credit: Václav Hlaváč

Lecture Overview

simple & heuristic

complex & principled

- A simple approach to segmentation: **(intensity) thresholding**
- Segmentation based on spatial coherence: **edge-based segmentation**, **region growing**
- Segmentation as a **clustering** problem: **k-means clustering**, **mean-shift clustering**
- Segmentation as a statistical (unsupervised) learning problem: **expectation maximization (EM) algorithm**
- Next lecture: **graph-based segmentation**, **supervised learning with neural networks** (if time and interest)

slide credit: Václav Hlaváč

Other Segmentation Approaches Not Covered

- **Template matching**: detect regions in image by comparing with templates, fitting structures in the image
  - Object detection based on templates
  - Parametric model detection, e.g., straight lines, circles, ellipses, …

slide credit: Václav Hlaváč

Other Segmentation Approaches Not Covered

- **Template matching**: detect regions in image by comparing with templates, fitting structures in the image
  - Object detection based on templates
  - Parametric model detection, e.g., straight lines, circles, ellipses, …
- **Based on unusual phenomena**: segmentation by detecting unusual structures
  - Camouflage detection based on unusual texture
  - Image compression: large regions as unusual occurrences that can be heavily compressed (e.g., regions of same color)

slide credit: Václav Hlaváč

A Few Words of Advice

- There is **no general purpose** segmentation algorithm for all cases

slide credit: Václav Hlaváč

A Few Words of Advice

- There is **no general purpose** segmentation algorithm for all cases
- Algorithm to use depends on circumstances:
  - Lots of labelled training data → supervised learning with CNNs
  - Simple structure & large color differences → thresholding
  - Little to no training data → unsupervised learning via clustering

slide credit: Václav Hlaváč

A Few Words of Advice

- There is **no general purpose** segmentation algorithm for all cases
- Algorithm to use depends on circumstances:
  - Lots of labelled training data → supervised learning with CNNs
  - Simple structure & large color differences → thresholding
  - Little to no training data → unsupervised learning via clustering
- Use **prior knowledge** whenever available:
  - Knowledge about shape or color of an object
  - Priors on position of object or region in image (e.g., images centered on object)
  - Relation between objects or regions (e.g., car always on top of road)

slide credit: Václav Hlaváč

Not Covered: Feature Design

- We directly observe **primary features**: pixel intensities, colors, depth (range cameras, e.g., LiDAR, Kinect), temperature (thermal cameras)

slide credit: Václav Hlaváč

Not Covered: Feature Design

- We directly observe **primary features**: pixel intensities, colors, depth (range cameras, e.g., LiDAR, Kinect), temperature (thermal cameras)
- We want to identify regions in the input that belong together

slide credit: Václav Hlaváč

Not Covered: Feature Design

- We directly observe **primary features**: pixel intensities, colors, depth (range cameras, e.g., LiDAR, Kinect), temperature (thermal cameras)
- We want to identify regions in the input that belong together
- How do we compare pixels / structures / regions? Extract **features** from direct observations:
  - Primary features (typically not very robust, e.g., to illumination changes)
  - **Secondary features**: information extracted from observations, e.g., shape parameters, texture parameters, relations between regions, motion parameters in video, stereo disparity / depth, …

slide credit: Václav Hlaváč

Not Covered: Feature Design

- We directly observe **primary features**: pixel intensities, colors, depth (range cameras, e.g., LiDAR, Kinect), temperature (thermal cameras)
- We want to identify regions in the input that belong together
- How do we compare pixels / structures / regions? Extract **features** from direct observations:
  - Primary features (typically not very robust, e.g., to illumination changes)
  - **Secondary features**: information extracted from observations, e.g., shape parameters, texture parameters, relations between regions, motion parameters in video, stereo disparity / depth, …
- **Choice of features is very important**, but not covered here

slide credit: Václav Hlaváč

Not Covered: Feature Design

- We directly observe **primary features**: pixel intensities, colors, depth (range cameras, e.g., LiDAR, Kinect), temperature (thermal cameras)
- We want to identify regions in the input that belong together
- How do we compare pixels / structures / regions? Extract **features** from direct observations:
  - Primary features (typically not very robust, e.g., to illumination changes)
  - **Secondary features**: information extracted from observations, e.g., shape parameters, texture parameters, relations between regions, motion parameters in video, stereo disparity / depth, …
- **Choice of features is very important**, but not covered here
- **Modern choice**: learn features from data → deep learning / machine learning

slide credit: Václav Hlaváč

Image Segmentation

- Goal: compute **complete segmentation** of image
  - Subdivide the image $\mathcal{I}$ into S disjoint regions R_1, R_2, …, R_s, i.e.,

$$\mathcal{I} = \bigcup_{i=1}^{S} R_i, \quad R_i \cap R_j = \emptyset, i \neq j$$

slide credit: Václav Hlaváč

Image Segmentation

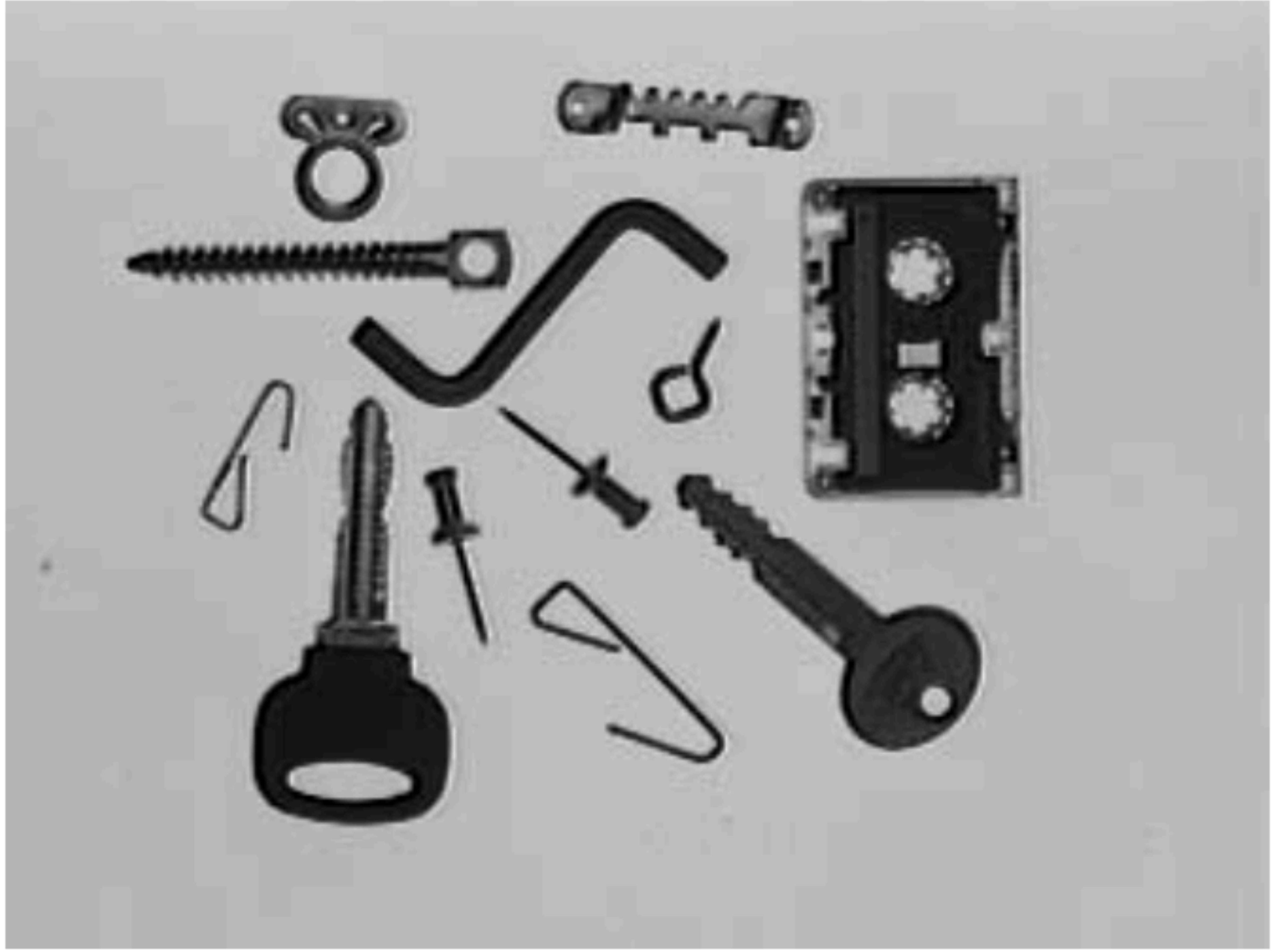
- Goal: compute **complete segmentation** of image
  - Subdivide the image $\mathscr{I}$ into S disjoint regions $R_1, R_2, \ldots, R_s$, i.e.,

$$\mathscr{I} = \bigcup_{i=1}^{S} R_i, \quad R_i \cap R_j = \emptyset, i \neq j$$

- Simplest case: **binary segmentation** into **foreground** (objects) and **background**
  - Surprisingly often a valid assumption as we often do not care about background

slide credit: Václav Hlaváč

Image Segmentation via Thresholding



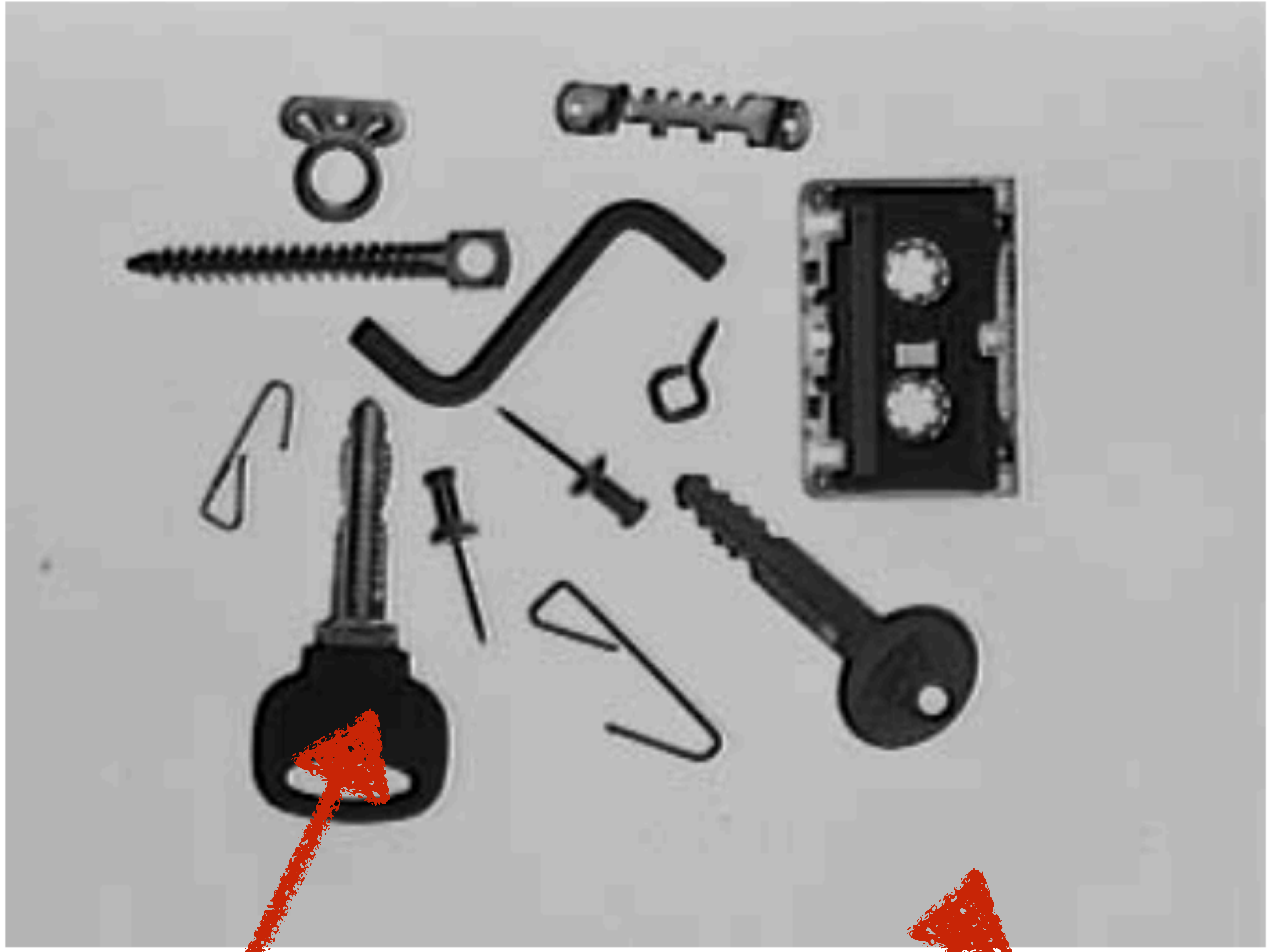
slide credit: Václav Hlaváč

Image Segmentation via Thresholding

simple background

slide credit: Václav Hlaváč

Image Segmentation via Thresholding

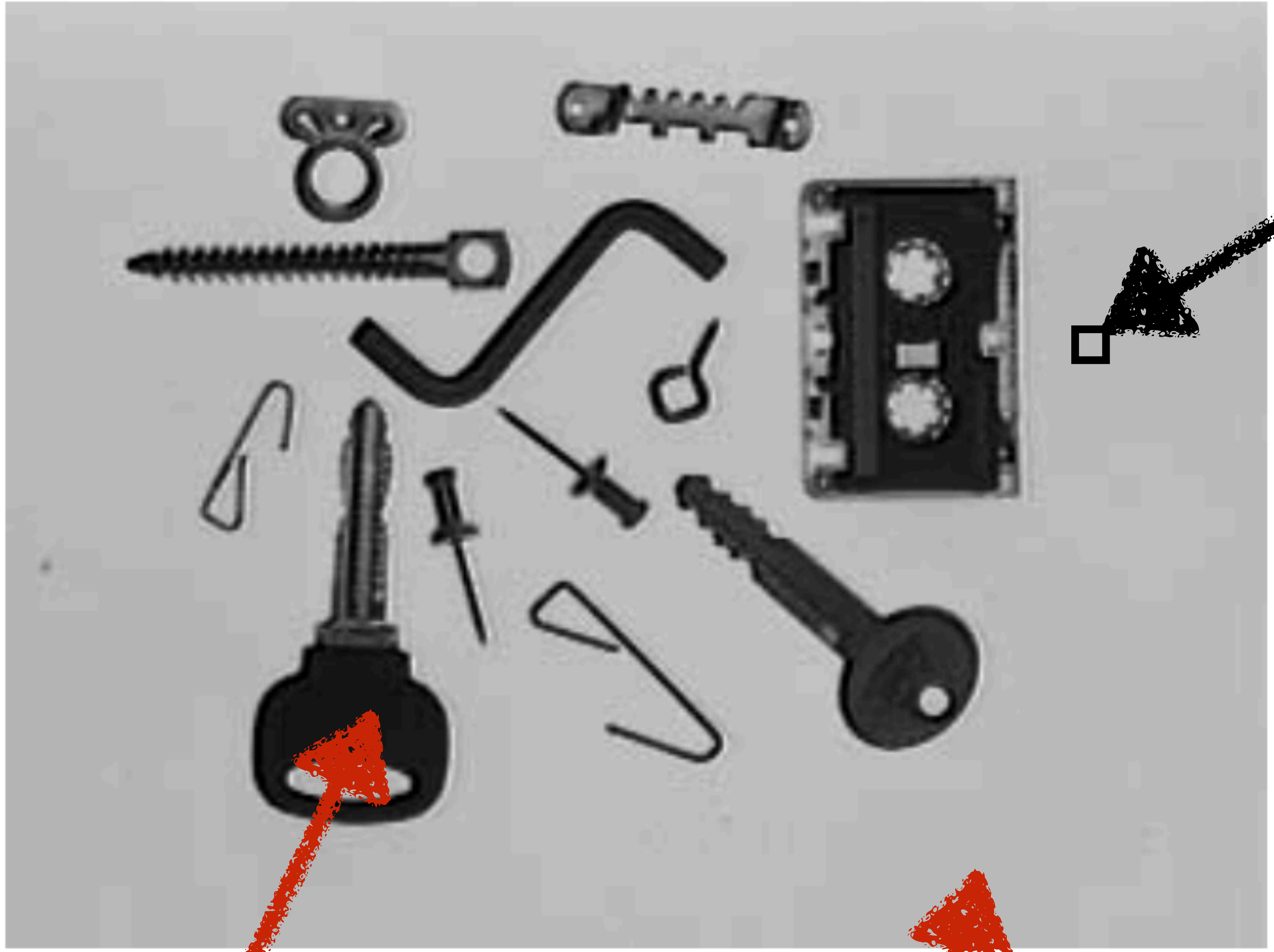


distinctly colored objects

simple background

slide credit: Václav Hlaváč

Image Segmentation via Thresholding



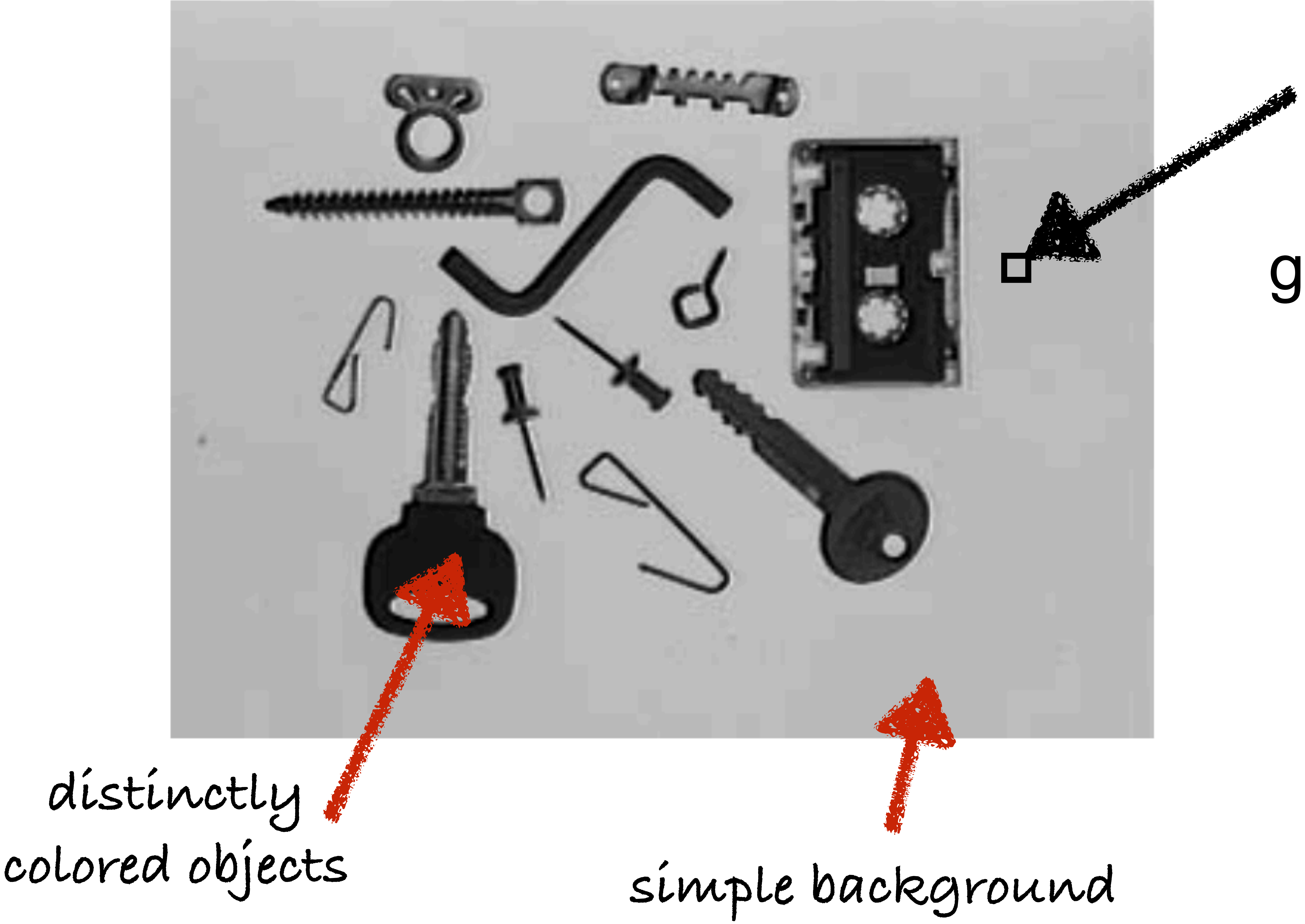
pixel (i, j) with intensity $f(i, j)$

distinctly colored objects

simple background

slide credit: Václav Hlaváč

Image Segmentation via Thresholding



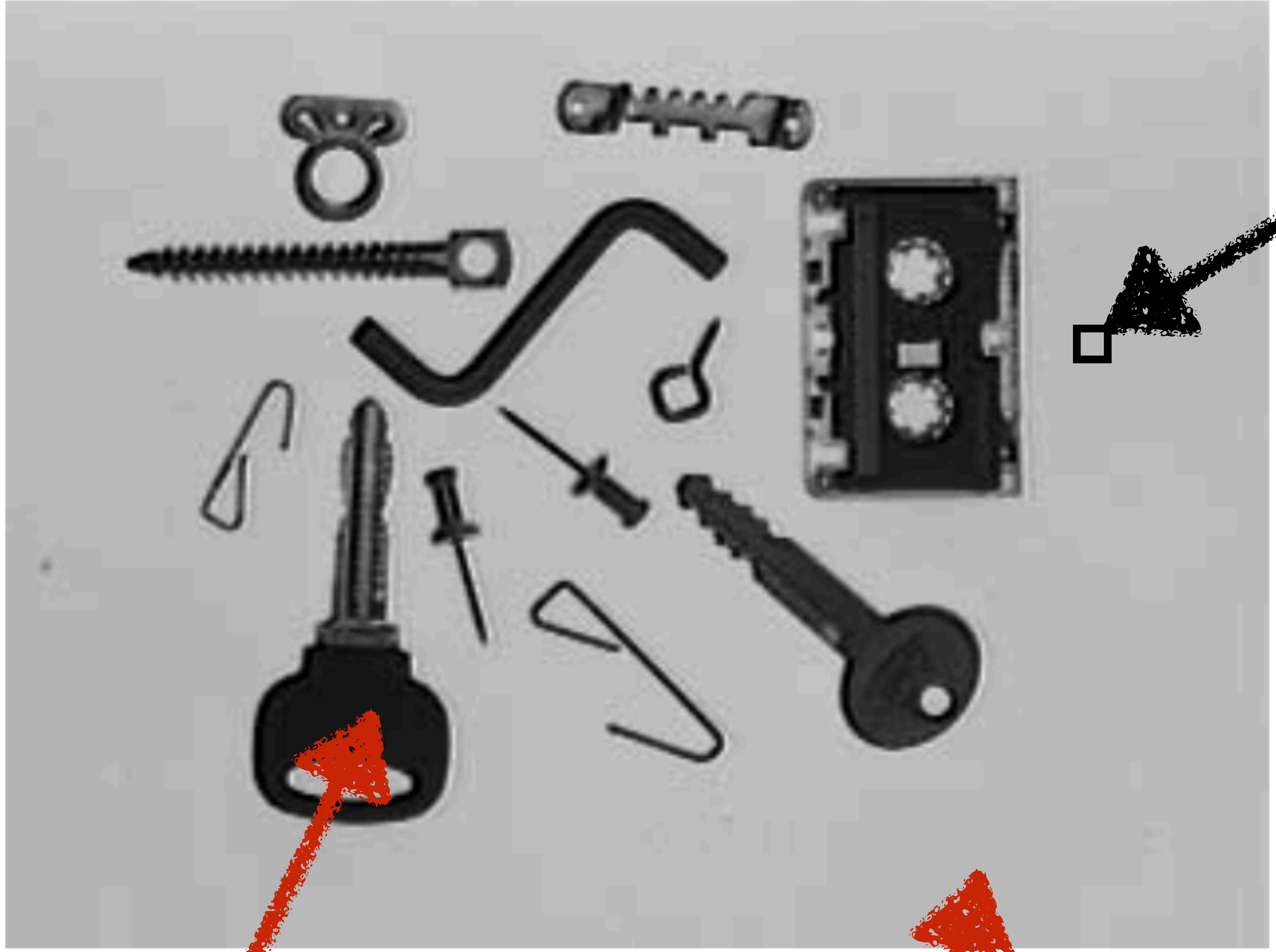
pixel (i,j) with intensity $f(i,j)$

generate binary image b with

$$g(i,j) = \begin{cases} 1 & \text{if} \quad f(i,j) \geq T \\ 0 & \text{if} \quad f(i,j) < T \end{cases}$$

slide credit: Václav Hlaváč

Image Segmentation via Thresholding



pixel (i,j) with intensity $f(i,j)$

generate binary image b with

$$g(i,j) = \begin{cases} 1 & \text{if} \quad f(i,j) \geq T \\ 0 & \text{if} \quad f(i,j) < T \end{cases}$$

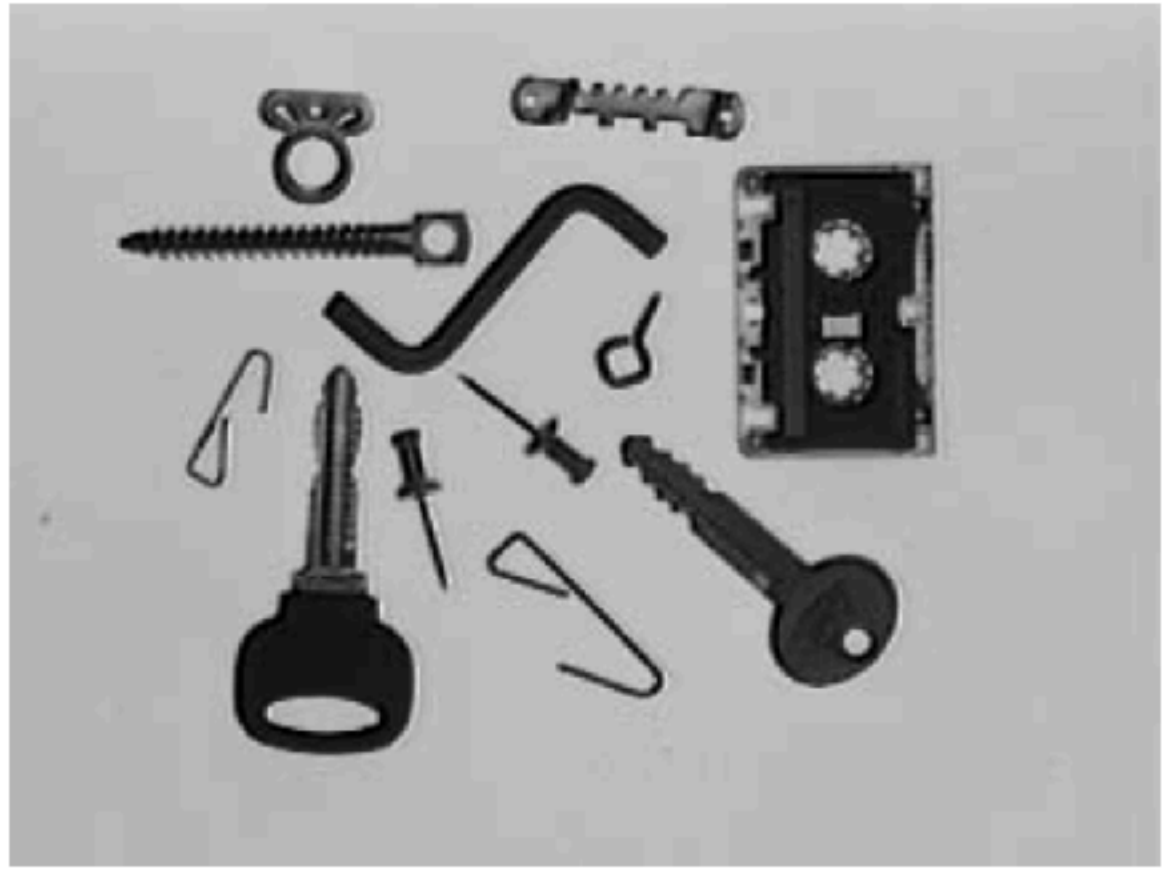
threshold

distinctly colored objects

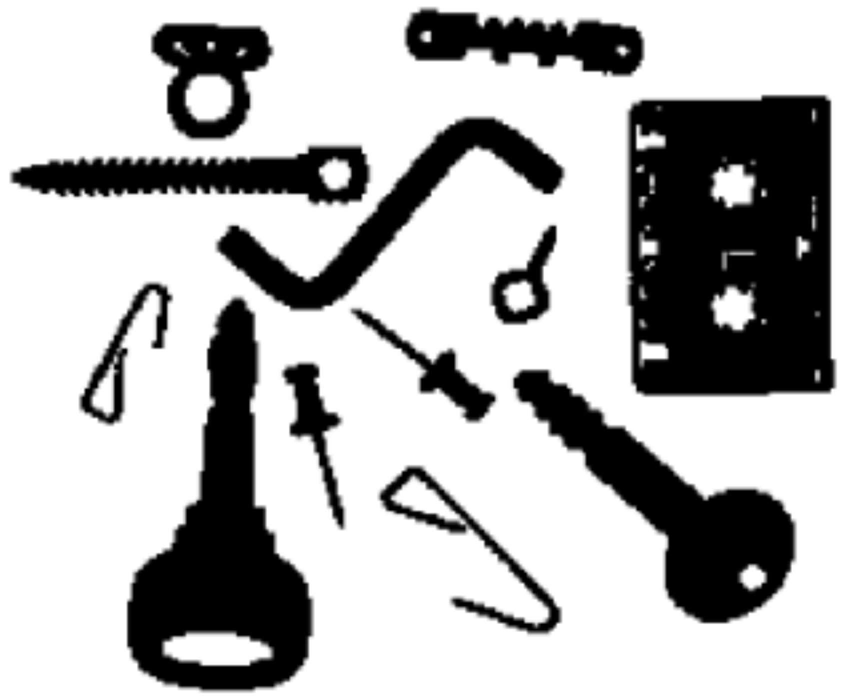
simple background

slide credit: Václav Hlaváč

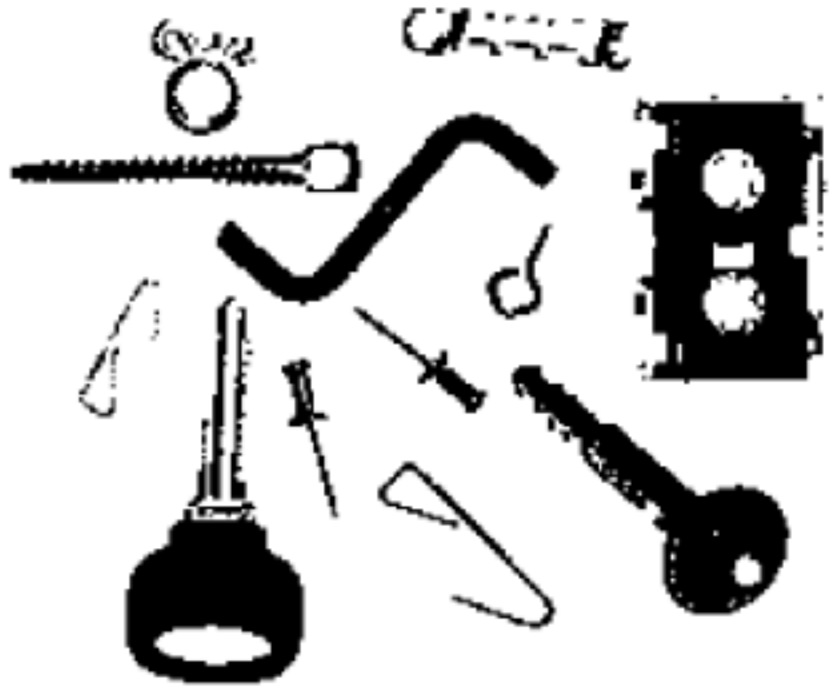
Example



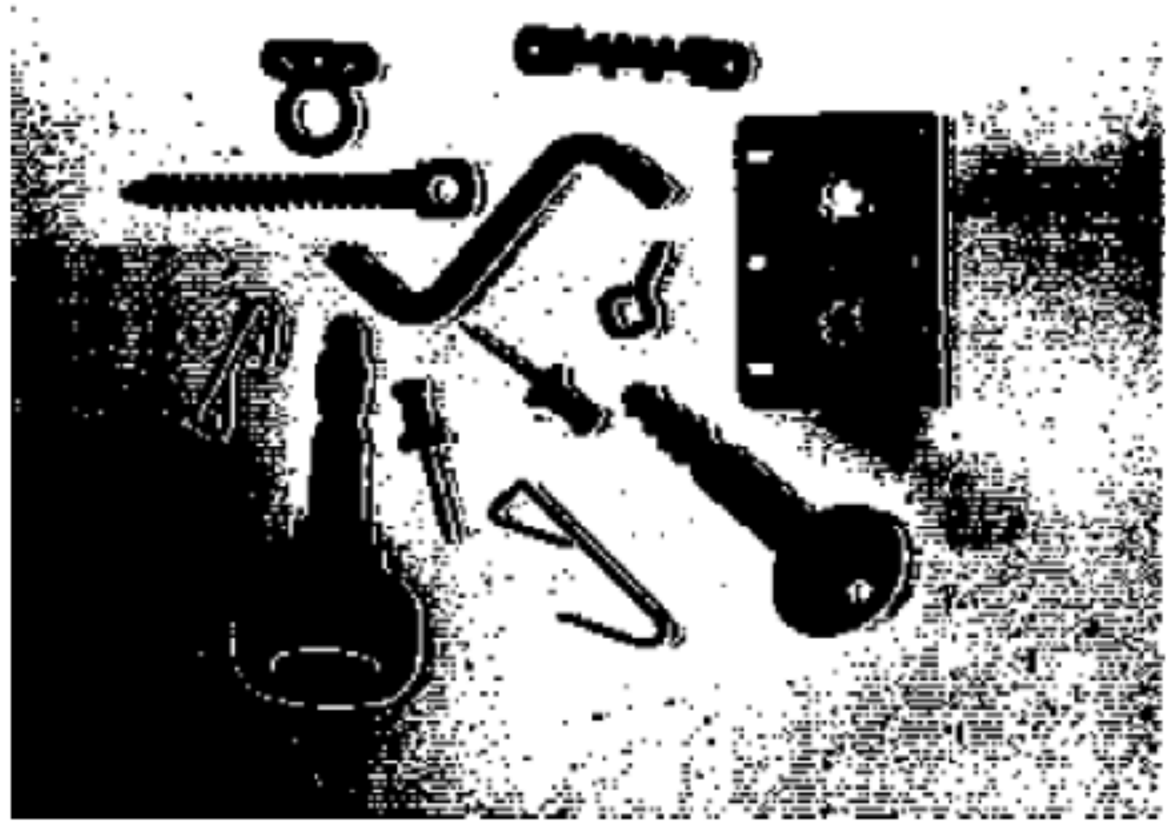
Original image.



Threshold segmentation.



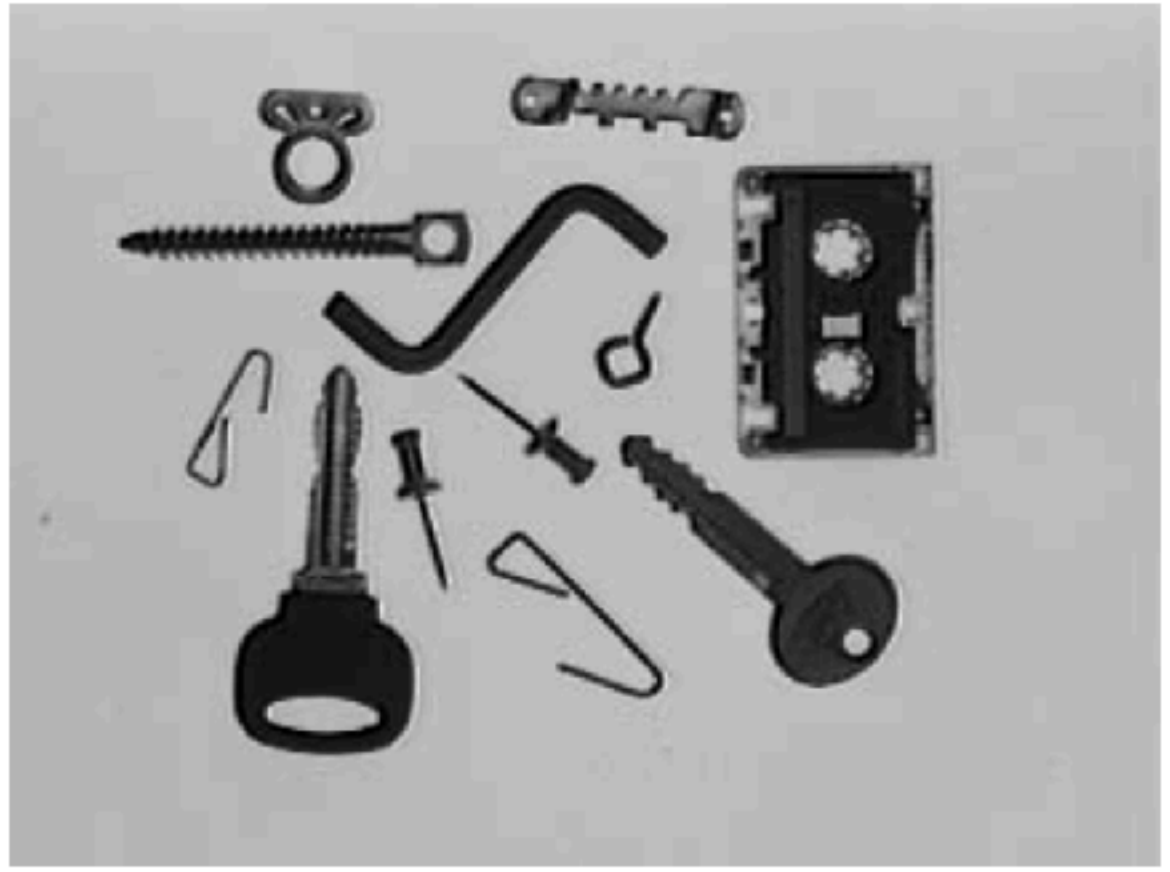
Threshold too low.



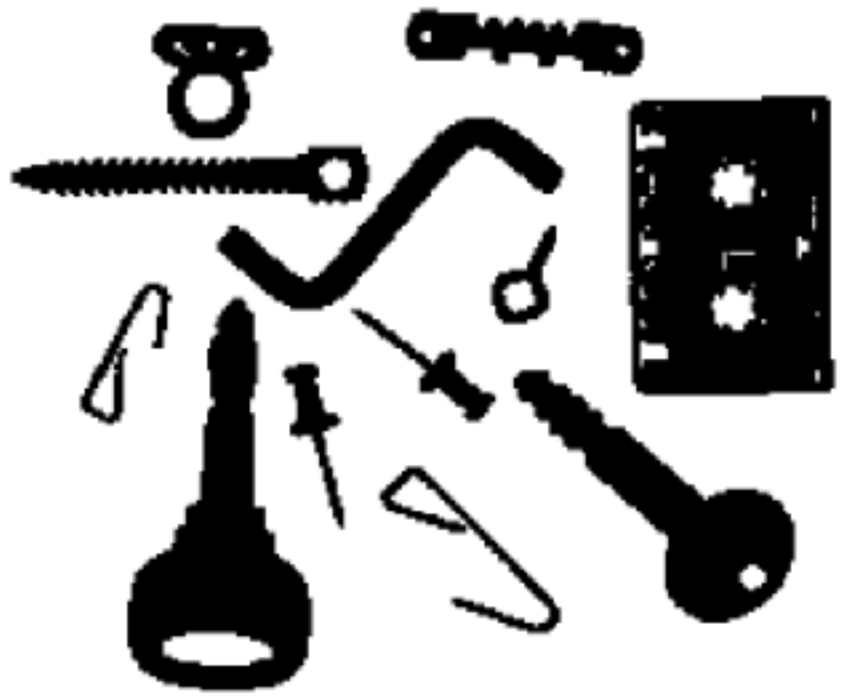
Threshold too high.

slide credit: Václav Hlaváč

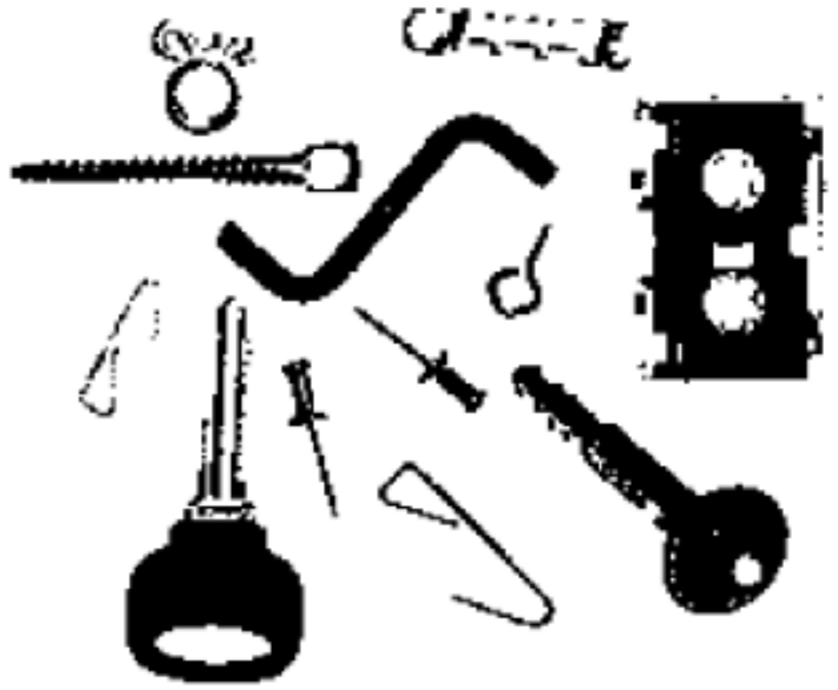
Example



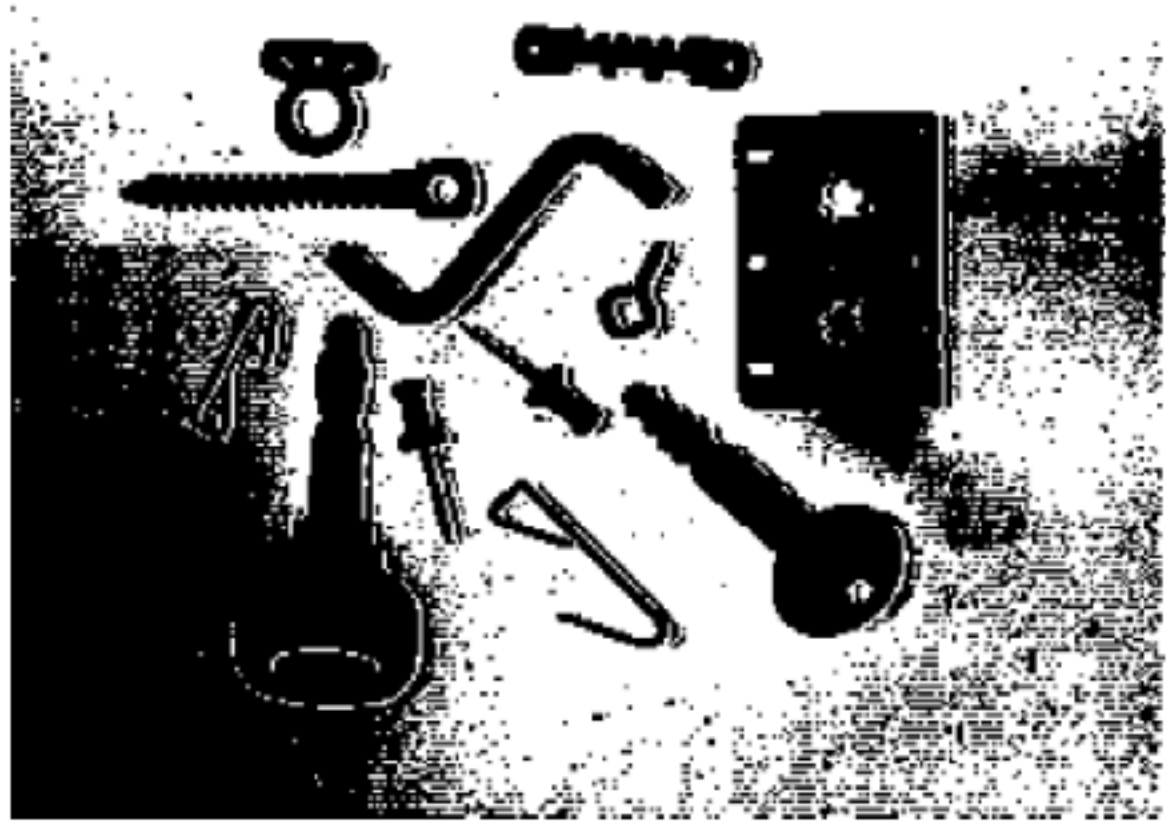
Original image.



Threshold segmentation.



Threshold too low.



Threshold too high.

How to choose the threshold?

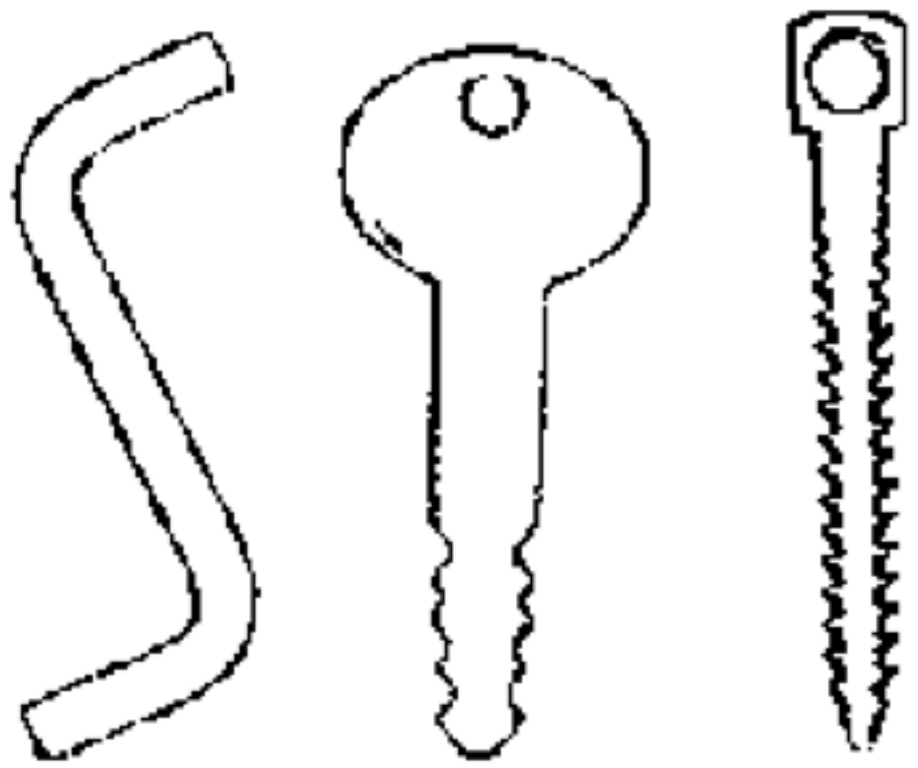
slide credit: Václav Hlaváč

Thresholding Choices

- **Band thresholding**: consider range D of intensities

$$g(i,j) = \begin{cases} 1 & \text{if} \quad f(i,j) \in D \\ 0 & \text{otherwise} \end{cases}$$

Original image.



Border regions detected.

slide credit: Václav Hlaváč

Thresholding Choices

- **Band thresholding**: consider range D of intensities

$$g(i,j) = \begin{cases} 1 & \text{if} \quad f(i,j) \in D \\ 0 & \text{otherwise} \end{cases}$$

- **Locally adaptive thresholding**: divide images into regions (e.g., regular grid) and find a threshold for each region

slide credit: Václav Hlaváč

Thresholding Choices

- **Band thresholding**: consider range D of intensities

$$g(i,j) = \begin{cases} 1 & \text{if} \quad f(i,j) \in D \\ 0 & \text{otherwise} \end{cases}$$

- **Locally adaptive thresholding**: divide images into regions (e.g., regular grid) and find a threshold for each region

- **Multiple thresholds**: use multiple thresholds for $S > 2$ classes

$$g(i,j) = \begin{cases} 2 & \text{if} \quad f(i,j) \geq T_2 \\ 1 & \text{if} \quad T_1 \leq f(i,j) < T_2 \\ 0 & \text{if} \quad f(i,j) < T_1 \end{cases}$$

slide credit: Václav Hlaváč

Thresholding Choices

- **Semi-thresholding**: only segment out the background, let human / other algorithm deal with foreground

$$g(i,j) = \begin{cases} f(i,j) & \text{if} \quad f(i,j) \geq T \\ 0 & \text{if} \quad f(i,j) < T \end{cases}$$

slide credit: Václav Hlaváč

Thresholding Choices

- **Semi-thresholding**: only segment out the background, let human / other algorithm deal with foreground

$$g(i,j) = \begin{cases} f(i,j) & \text{if} \quad f(i,j) \geq T \\ 0 & \text{if} \quad f(i,j) < T \end{cases}$$

- ***p*-tile thresholding**: if object covers 1/*p* of image, find the corresponding 1/*p* of histogram (e.g., when we know size of printed characters)

slide credit: Václav Hlaváč

Thresholding Choices

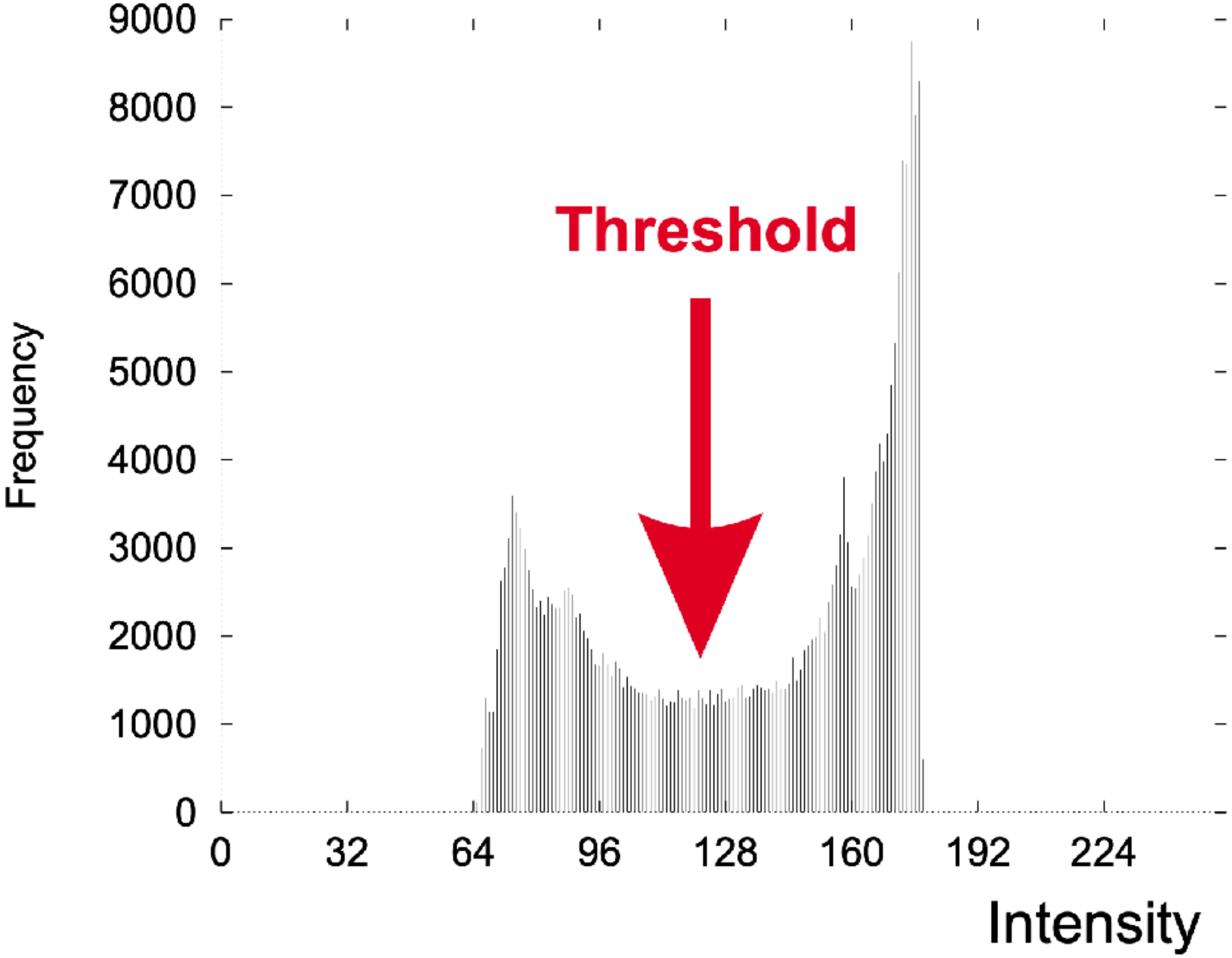
- **Semi-thresholding**: only segment out the background, let human / other algorithm deal with foreground

$$g(i,j) = \begin{cases} f(i,j) & \text{if} \quad f(i,j) \geq T \\ 0 & \text{if} \quad f(i,j) < T \end{cases}$$

- ***p*-tile thresholding**: if object covers 1/*p* of image, find the corresponding 1/*p* of histogram (e.g., when we know size of printed characters)
- **Automatic thresholding based on histograms**: compute histogram of intensities, objects and background should correspond to distinct modes, find threshold(s) separating the modes

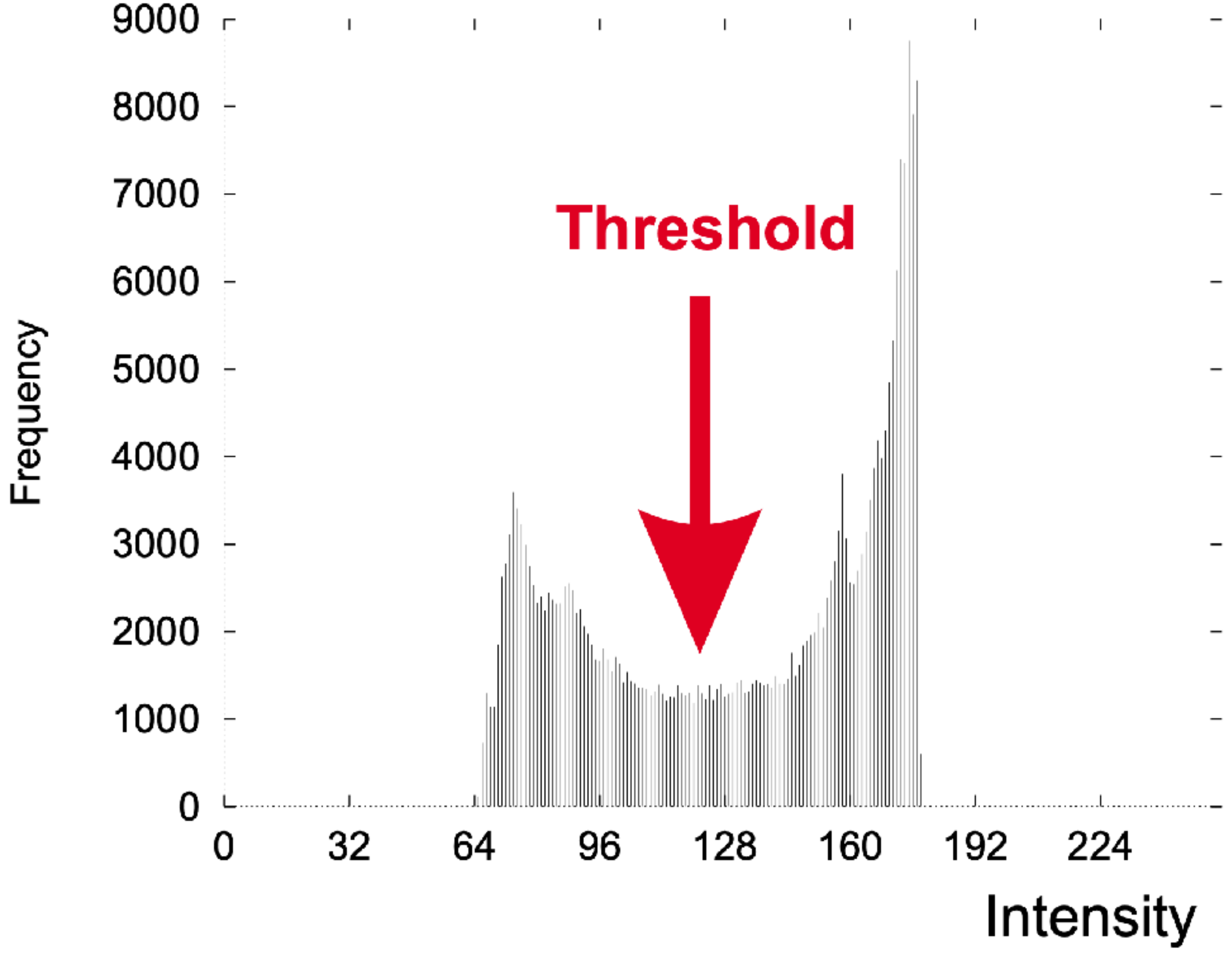
slide credit: Václav Hlaváč

Automatic Thresholding Based On Histograms



slide credit: Václav Hlaváč

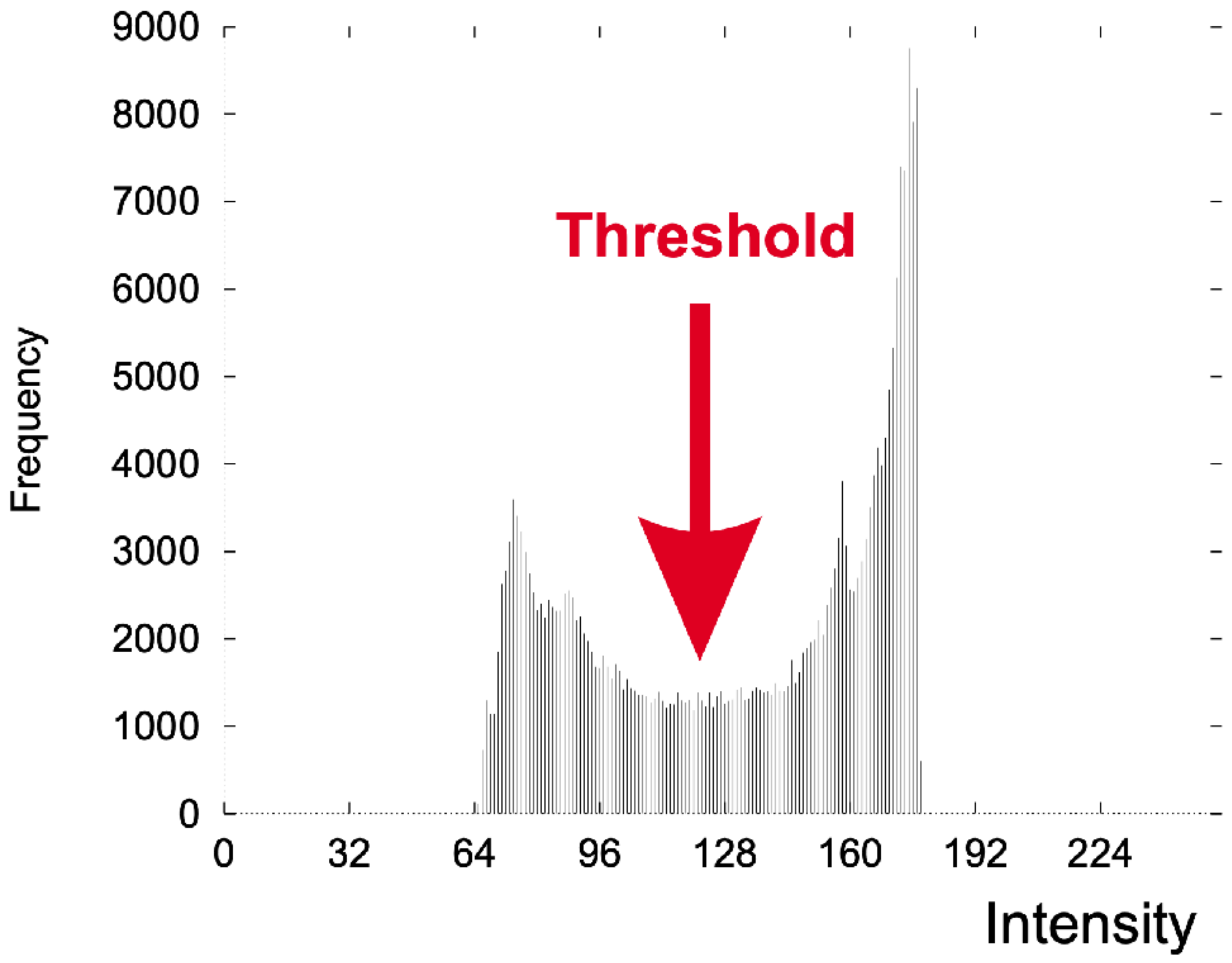
Automatic Thresholding Based On Histograms



- How to handle noisy histograms?

slide credit: Václav Hlaváč

Automatic Thresholding Based On Histograms



- How to handle noisy histograms?
- How to find optimal threshold(s)?

slide credit: Václav Hlaváč

Smoothing Histograms

- Noise in observations (e.g., noisy pixel intensities) → noisy / ragged histograms

slide credit: Václav Hlaváč

Smoothing Histograms

- Noise in observations (e.g., noisy pixel intensities) → noisy / ragged histograms
- Leads to **multiple local extrema**, makes analysis harder

slide credit: Václav Hlaváč

Smoothing Histograms

- Noise in observations (e.g., noisy pixel intensities) → noisy / ragged histograms
- Leads to **multiple local extrema**, makes analysis harder
- Smooth histogram before further processing, e.g., using **1D sliding average filter**:

slide credit: Václav Hlaváč

Smoothing Histograms

- Noise in observations (e.g., noisy pixel intensities) → noisy / ragged histograms
- Leads to **multiple local extrema**, makes analysis harder
- Smooth histogram before further processing, e.g., using **1D sliding average filter**:
  - Input histogram $h(i)$ over intensities $i = 0, \ldots, i_{\max}$

slide credit: Václav Hlaváč

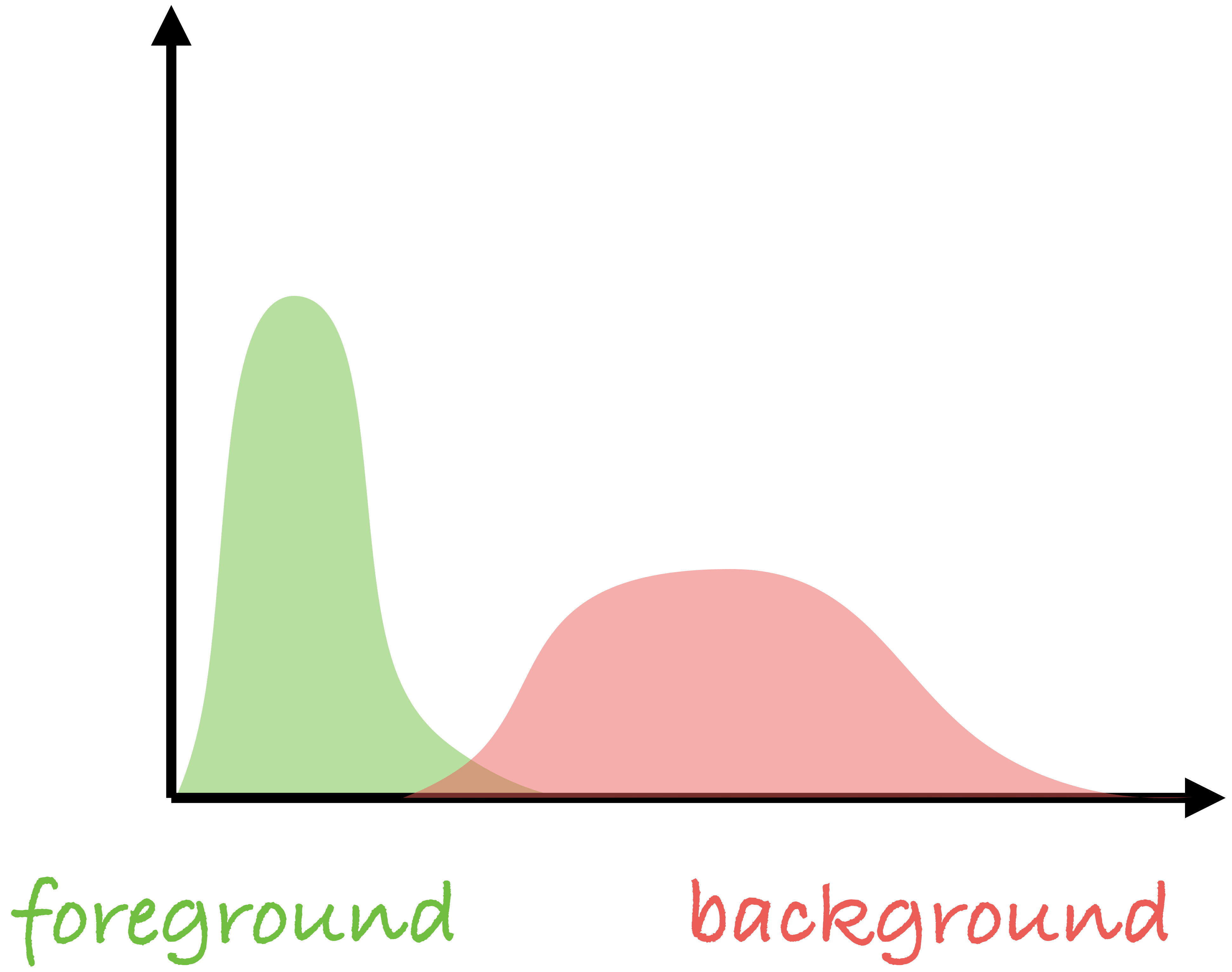
Smoothing Histograms

- Noise in observations (e.g., noisy pixel intensities) → noisy / ragged histograms
- Leads to **multiple local extrema**, makes analysis harder
- Smooth histogram before further processing, e.g., using **1D sliding average filter**:
  - Input histogram $h(i)$ over intensities $i = 0, \ldots, i_{\max}$
  - New histogram $h'(i)$ after applying sliding average with window size $2K+1$

$$h(i) = \frac{1}{2K+1} \sum_{j=-K}^{K} h(i+j), \quad i = K, \ldots, i_{\max} - K$$

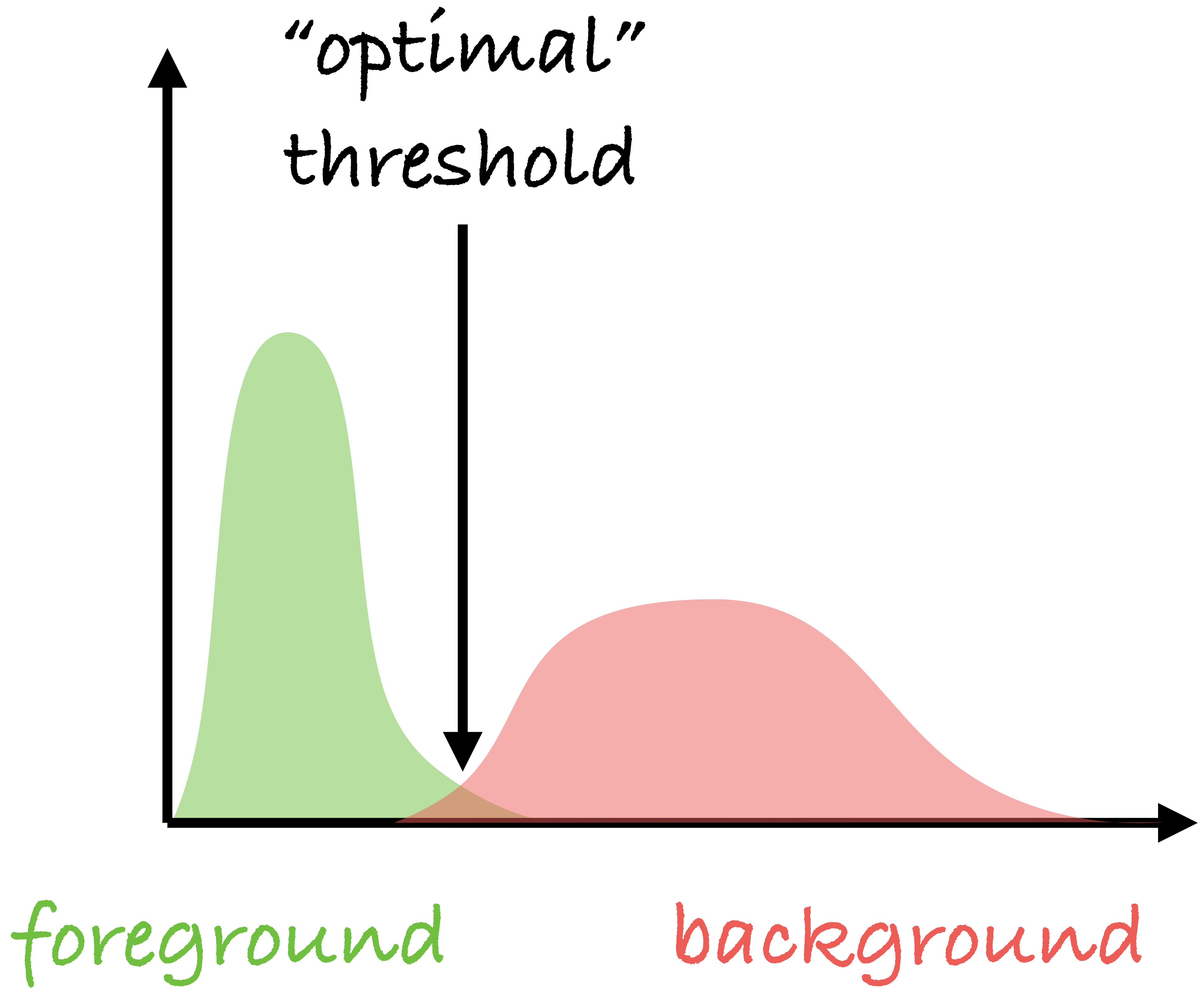
slide credit: Václav Hlaváč

“Optimal” Thresholding Via Mixture of Gaussians



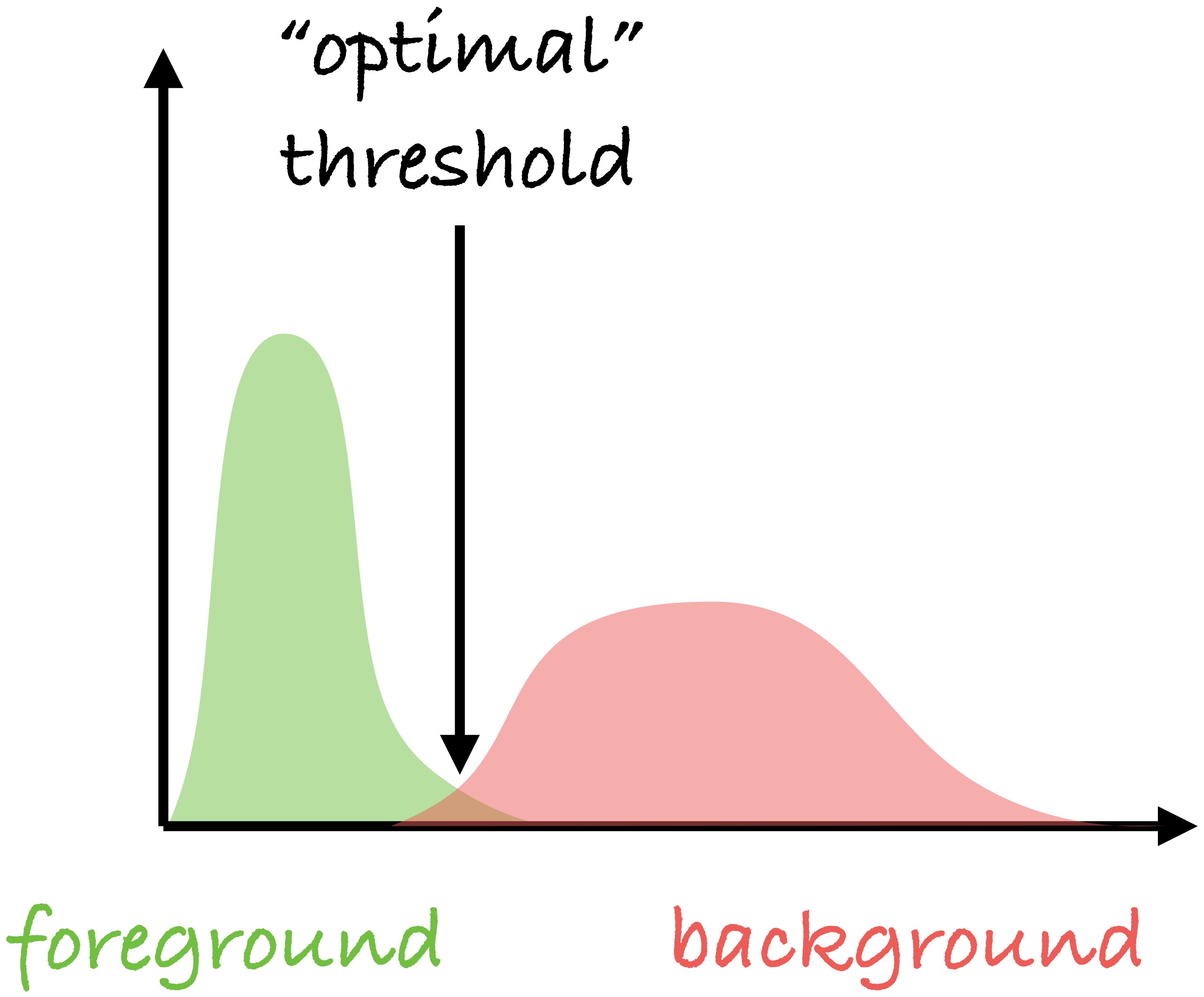
slide credit: Václav Hlaváč

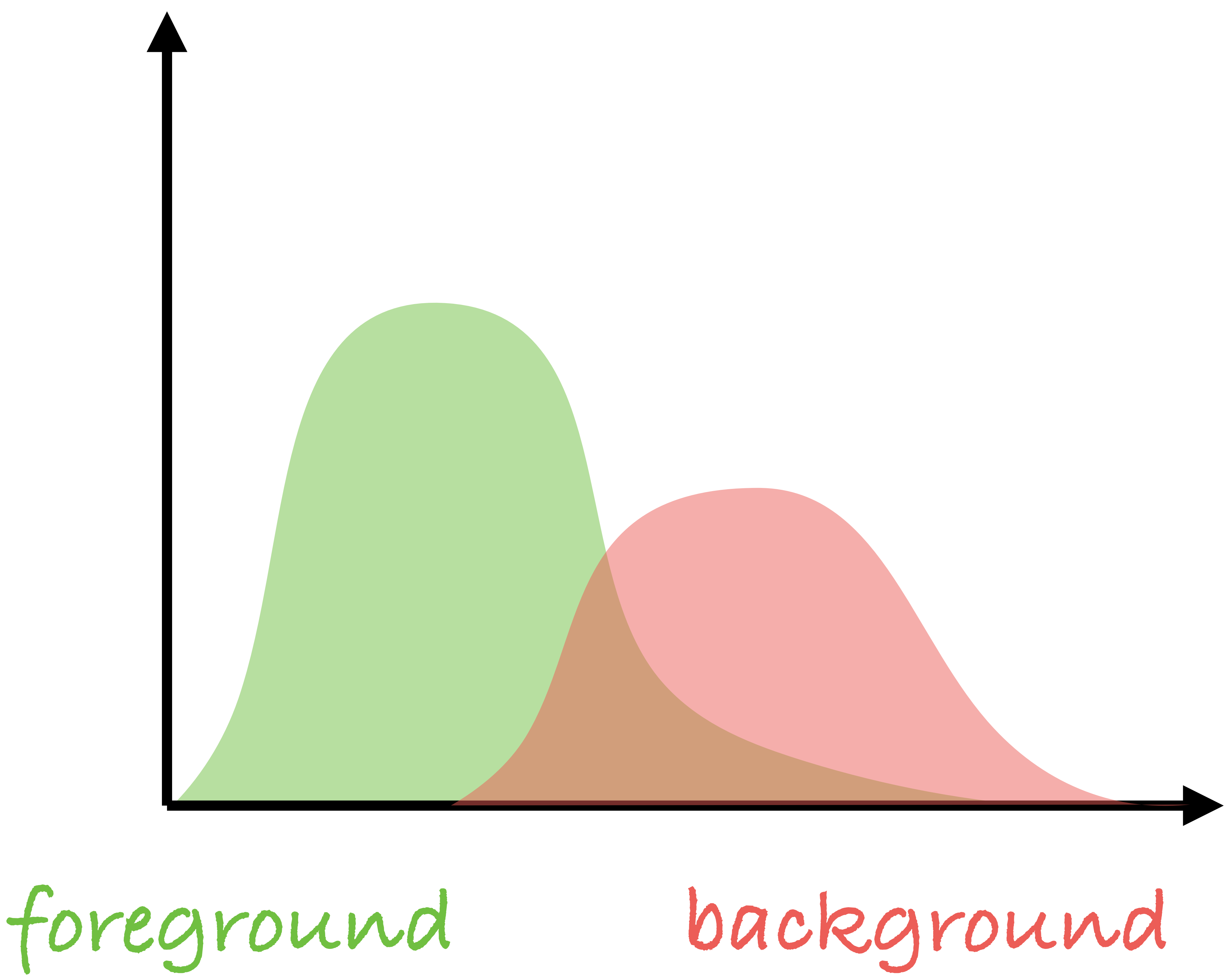
“Optimal” Thresholding Via Mixture of Gaussians



slide credit: Václav Hlaváč

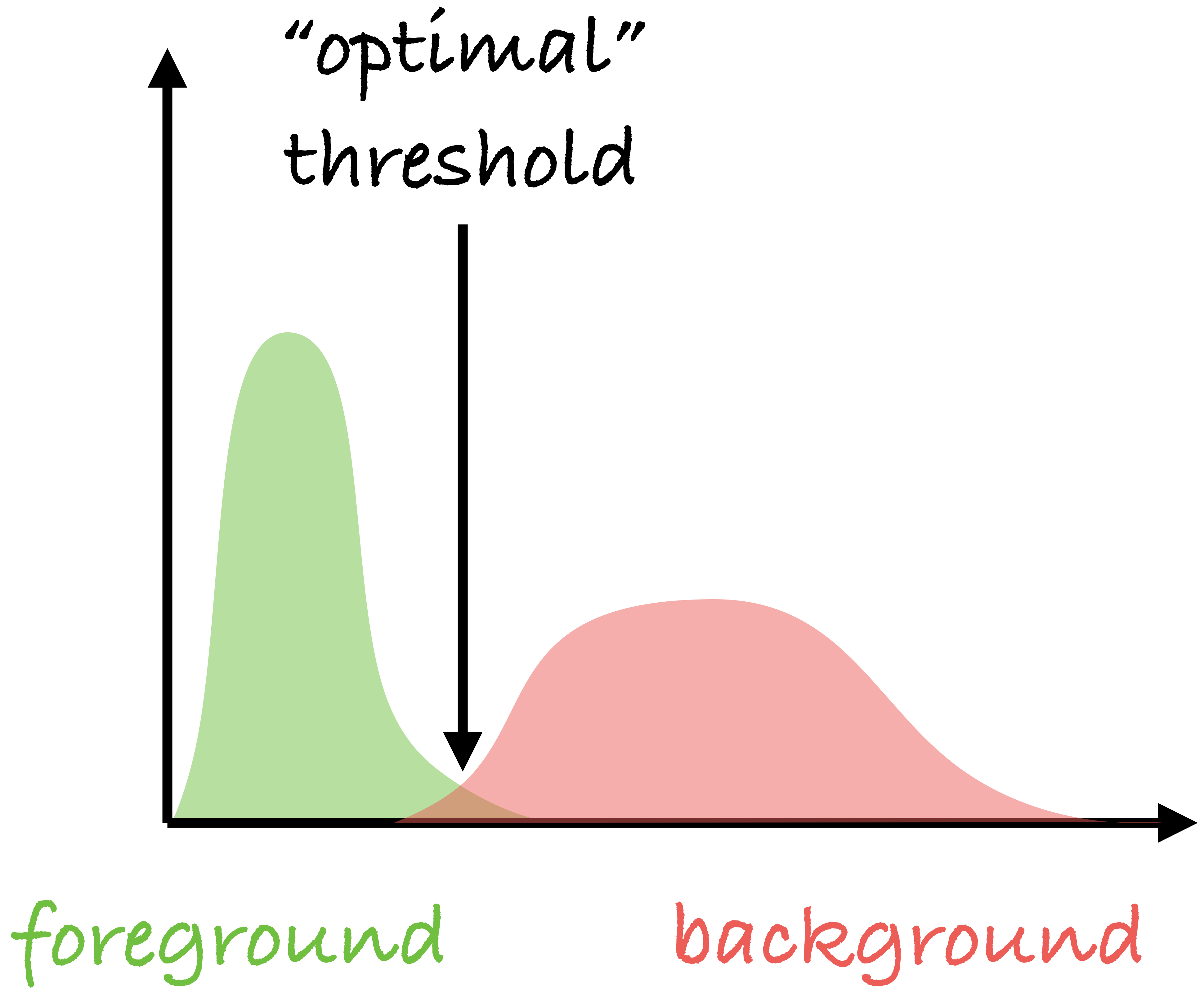
“Optimal” Thresholding Via Mixture of Gaussians

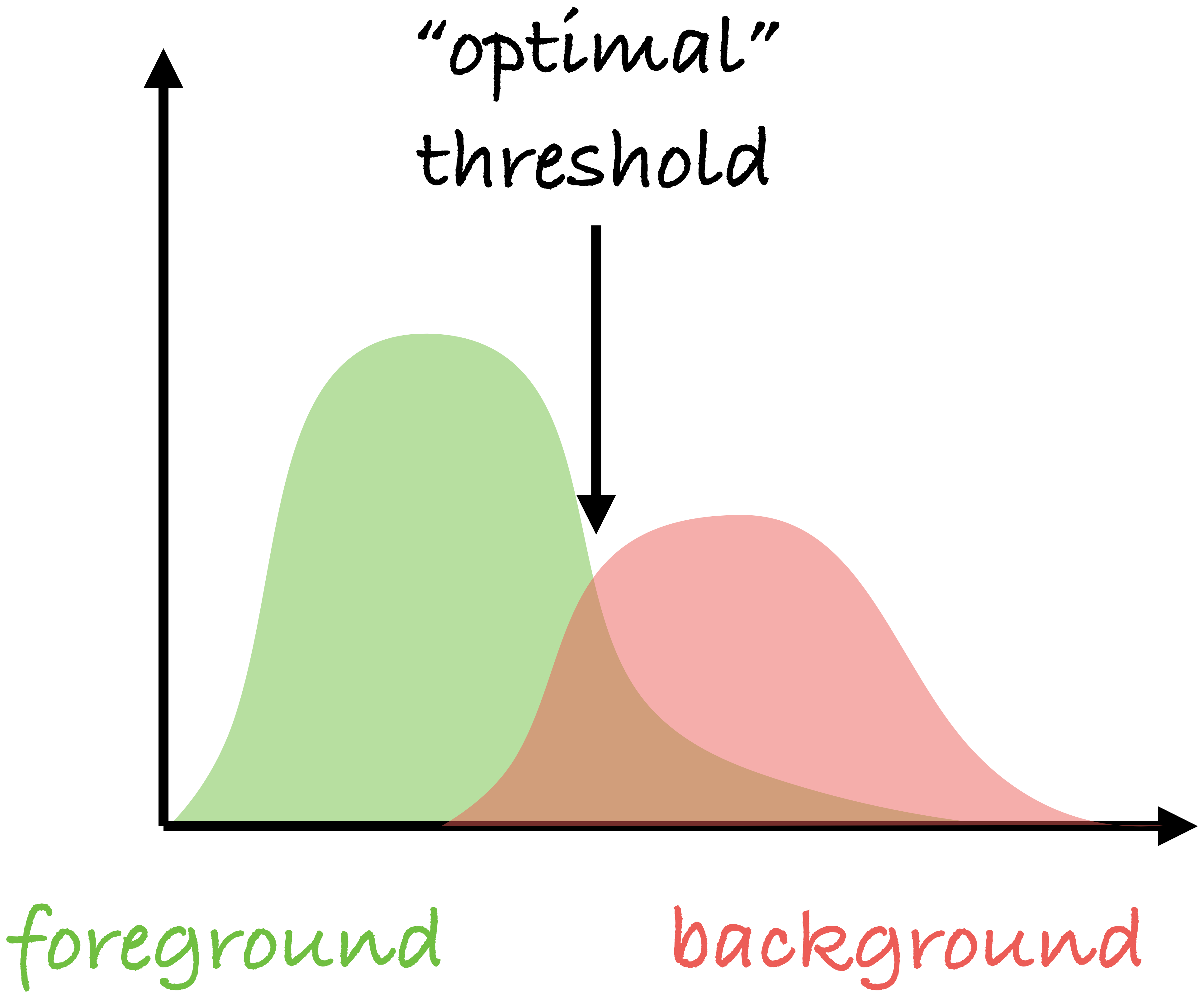




slide credit: Václav Hlaváč

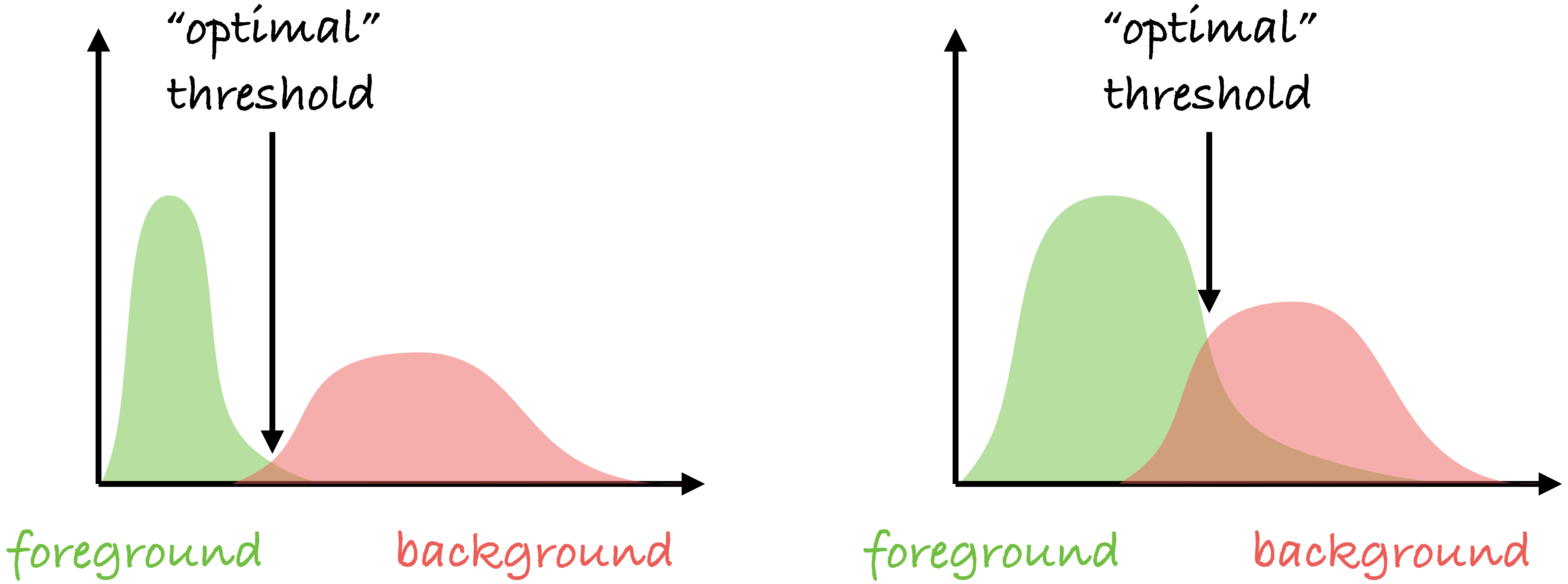
“Optimal” Thresholding Via Mixture of Gaussians





slide credit: Václav Hlaváč

“Optimal” Thresholding Via Mixture of Gaussians



Choose thresholds based on decision boundaries:

$$p(\text{foreground} \mid x) > p(\text{background} \mid x)$$

slide credit: Václav Hlaváč

Fitting A Mixture of Gaussians

- Input: observed histogram $h(g)$

slide credit: Václav Hlaváč

Fitting A Mixture of Gaussians

- Input: observed histogram $h(g)$
- Estimate: approximate histogram $h_{\text{model}}(g)$ modeled by *n* Gaussians

$$h_{\text{model}}(g) = \sum_{i=1}^{n} a_i e^{-\frac{(g-\mu_i)^2}{2\sigma_i^2}}$$

slide credit: Václav Hlaváč

Fitting A Mixture of Gaussians

- Input: observed histogram $h(g)$
- Estimate: approximate histogram $h_{\text{model}}(g)$ modeled by *n* Gaussians

$$h_{\text{model}}(g) = \sum_{i=1}^{n} a_i e^{-\frac{(g-\mu_i)^2}{2\sigma_i^2}}$$

- Fit by minimizing $\sum_{g \in G} \left(h(g) - h_{\text{model}}(h) \right)^2$

slide credit: Václav Hlaváč

Fitting A Mixture of Gaussians

- Input: observed histogram $h(g)$
- Estimate: approximate histogram $h_{\text{model}}(g)$ modeled by *n* Gaussians

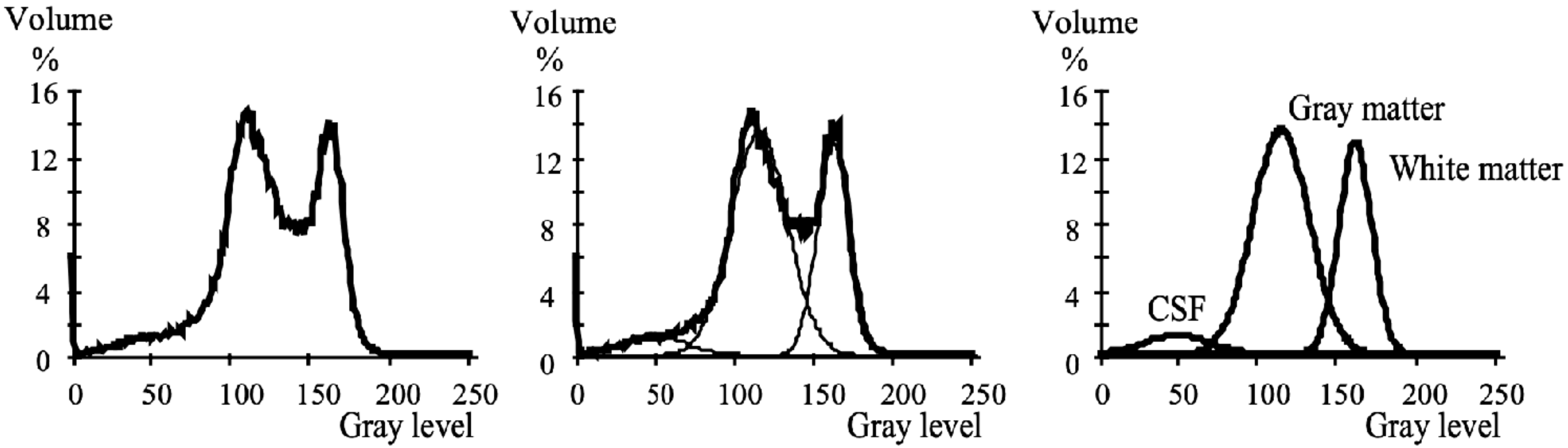
$$h_{\text{model}}(g) = \sum_{i=1}^{n} a_i e^{-\frac{(g-\mu_i)^2}{2\sigma_i^2}}$$

- Fit by minimizing $\sum_{g \in G} \left(h(g) - h_{\text{model}}(h) \right)^2$
- See part on **expectation maximization**

slide credit: Václav Hlaváč

Example: Brain MRI Segmentation

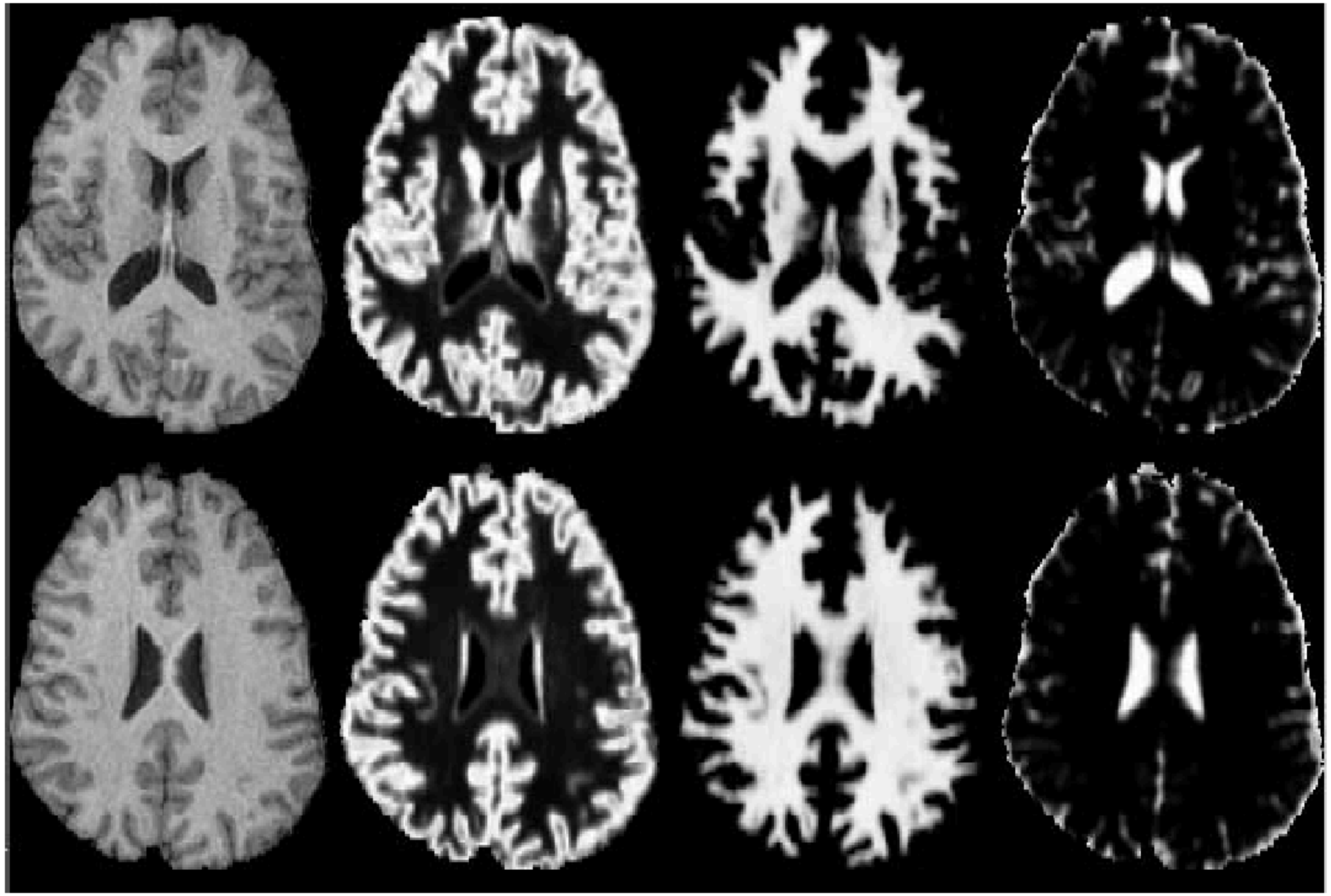
- Input: NMR images
- Classes: white matter, grey matter, celebrities-spinal fluid (CSF)



Courtesy: Milan Šonka, University of Iowa.

slide credit: Václav Hlaváč

Example: Brain MRI Segmentation

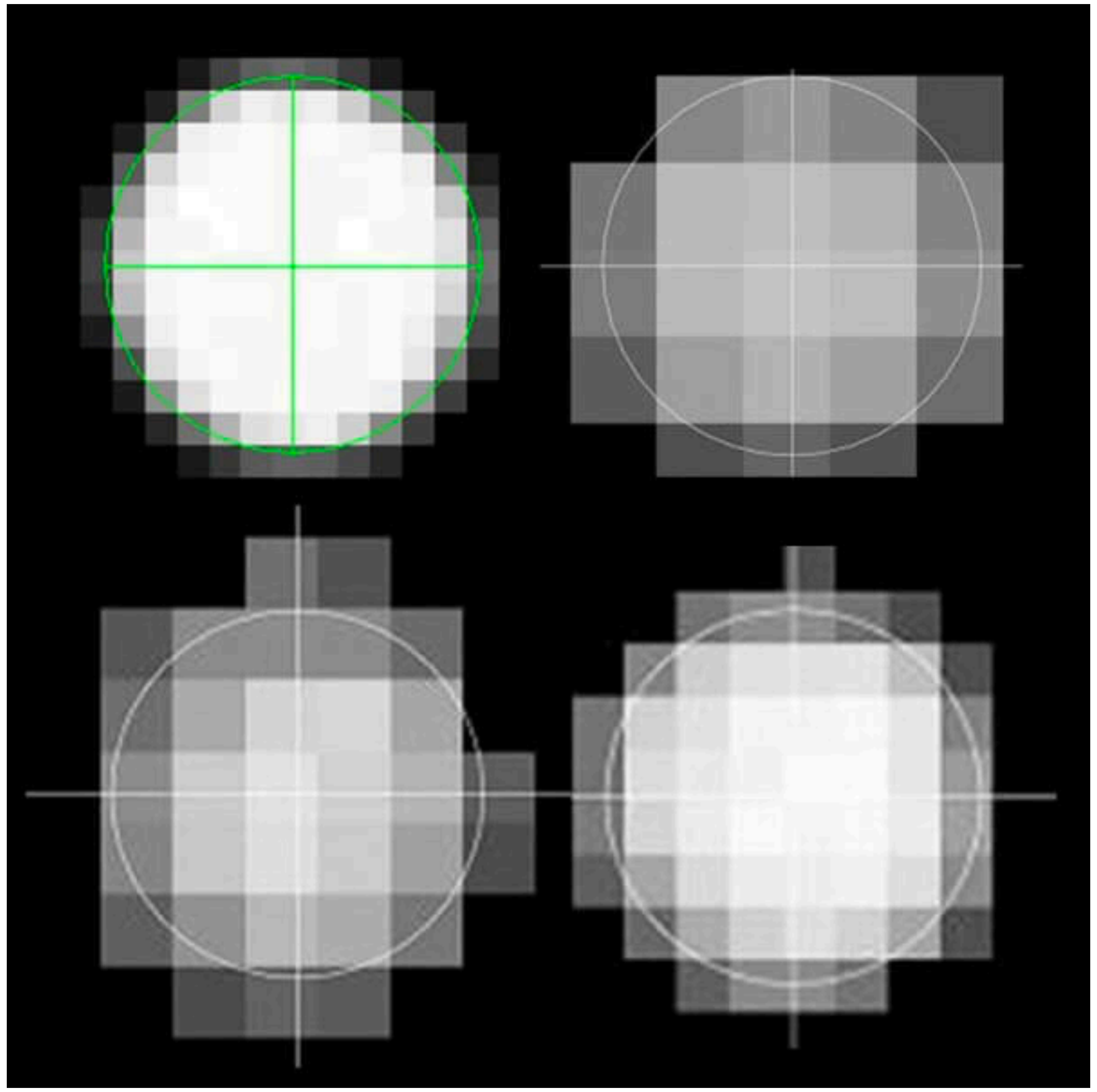


original gray matter white matter CSF

Courtesy: Milan Šonka, University of Iowa.

slide credit: Václav Hlaváč

Isn’t This Outdated?



Depending on application, we can ensure that thresholding works well!

images taken from Vicon website

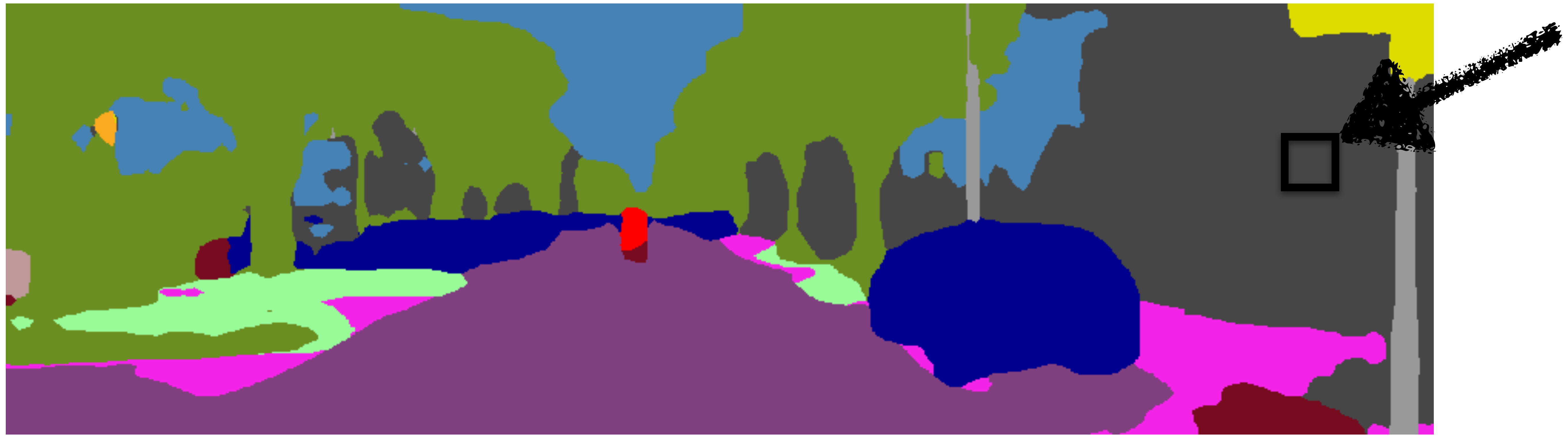
Isn't This Outdated?

Compute representations where thresholding works

images from [Lianos, Schönberger, Pollefeys, Sattler, VSO: Visual Semantic Odometry]

Isn't This Outdated?

Compute representations where thresholding works



set of probabilities per pixel x

$p(x = \text{car}) = \ldots$

$p(x = \text{building}) = \ldots$

$p(x = \text{road}) = \ldots$

$\vdots$

images from [Lianos, Schönberger, Pollefeys, Sattler, VSO: Visual Semantic Odometry]

Isn't This Outdated?

Compute representations where thresholding works

set of probabilities per pixel x

$p(x = \text{car}) = \ldots$

$p(x = \text{building}) = \ldots$

$p(x = \text{road}) = \ldots$

$\vdots$

result of thresholding on $p(x = \text{car})$

images from [Lianos, Schönberger, Pollefeys, Sattler, VSO: Visual Semantic Odometry]

Lecture Overview

simple & heuristic

complex & principled

- A simple approach to segmentation: **(intensity) thresholding**
- Segmentation based on spatial coherence: **edge-based segmentation**, **region growing**
- Segmentation as a **clustering** problem: **k-means clustering**, **mean-shift clustering**
- Segmentation as a statistical (unsupervised) learning problem: **expectation maximization (EM) algorithm**
- Next lecture: **graph-based segmentation**, **supervised learning with neural networks** (if time and interest)

slide credit: Václav Hlaváč

Thresholding Summarized

- **Pro**: very easy to implement

slide credit: Václav Hlaváč

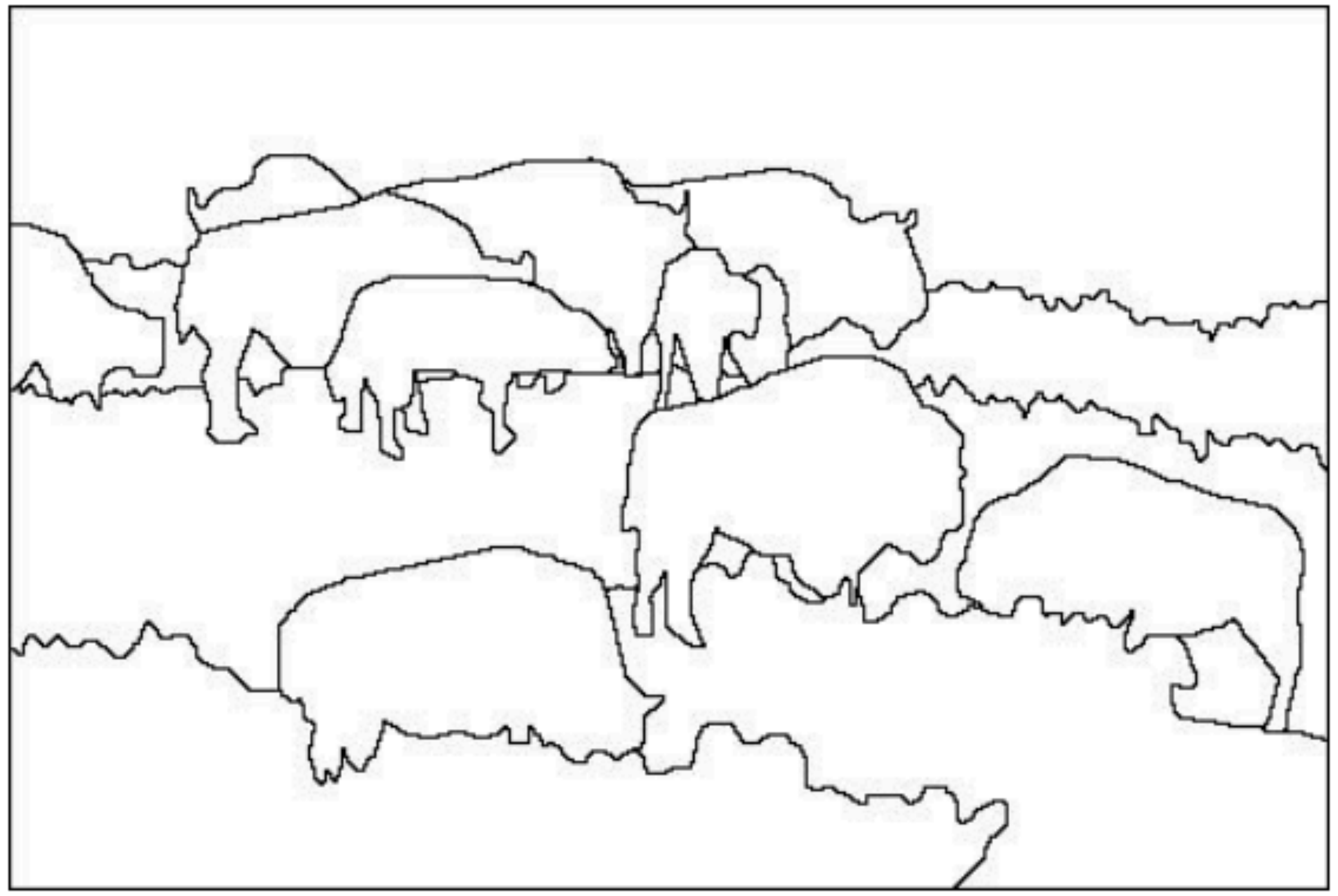
Thresholding Summarized

- **Pro**: very easy to implement
- **Pro**: very easy to parallelize

slide credit: Václav Hlaváč

Thresholding Summarized

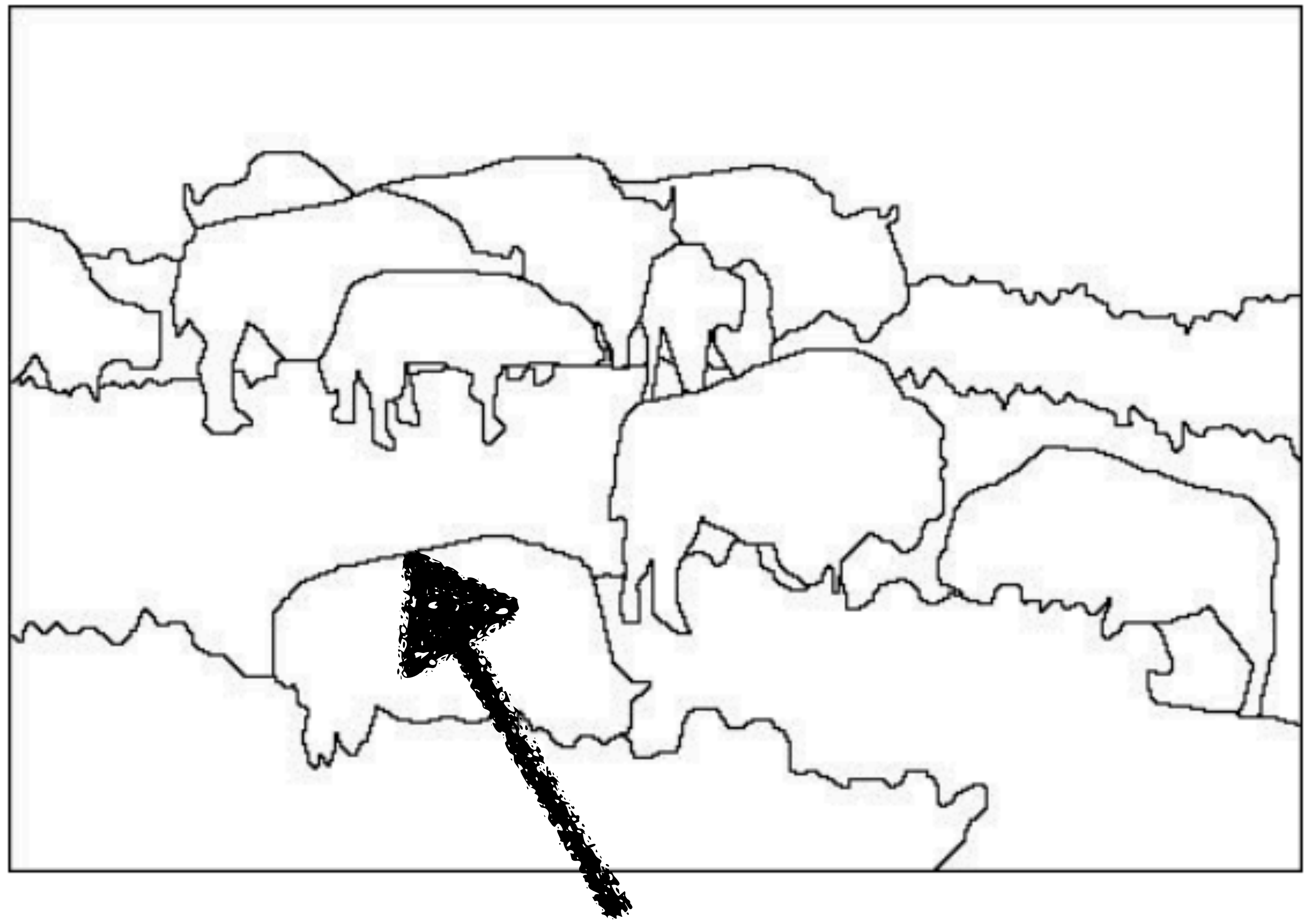
- **Pro**: very easy to implement
- **Pro**: very easy to parallelize
- **Con**: we are ignoring that we are looking for **regions** of pixels that belong together → ignoring **spatial consistency**



slide credit: Václav Hlaváč

Thresholding Summarized

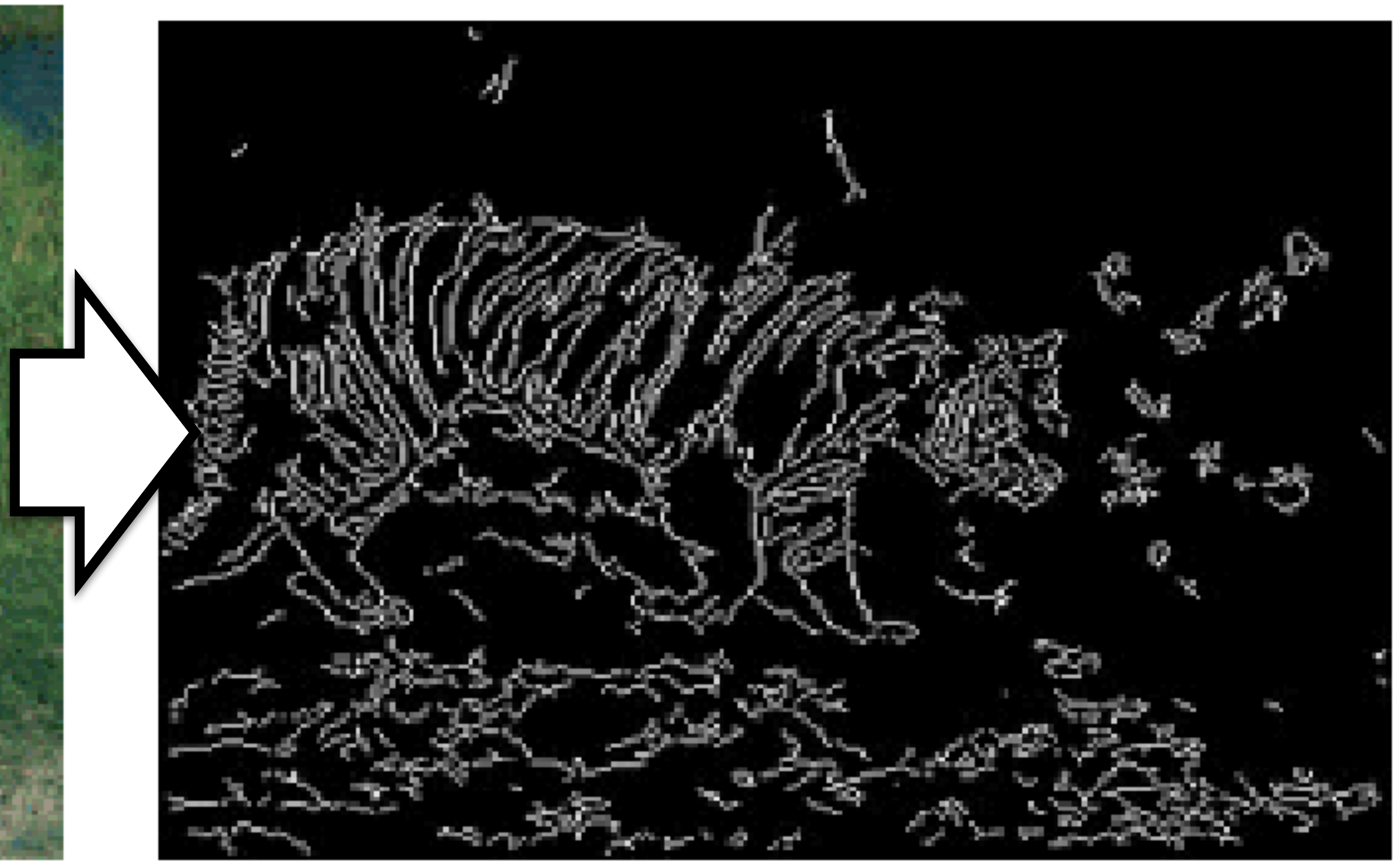
- **Pro**: very easy to implement
- **Pro**: very easy to parallelize
- **Con**: we are ignoring that we are looking for **regions** of pixels that belong together → ignoring **spatial consistency**



edges = region boundaries

slide credit: Václav Hlaváč

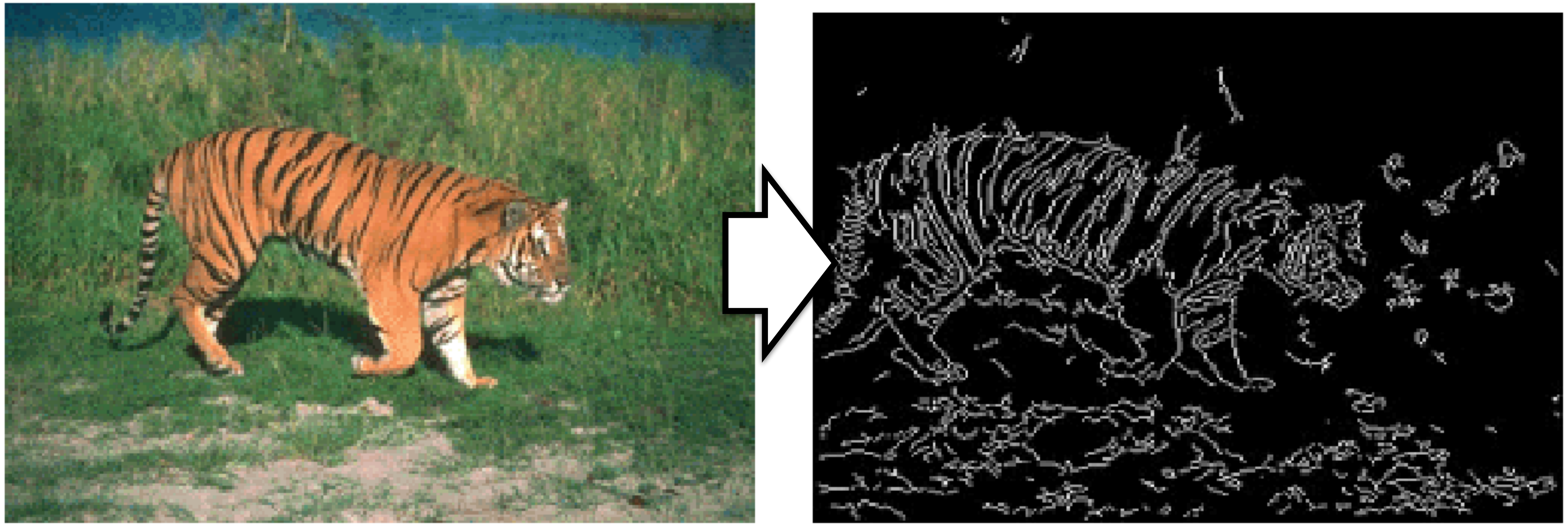
Edge-Based Image Segmentation



- **Edgels**: significant edges from edge detector (e.g., Canny)

slide credit: Václav Hlaváč

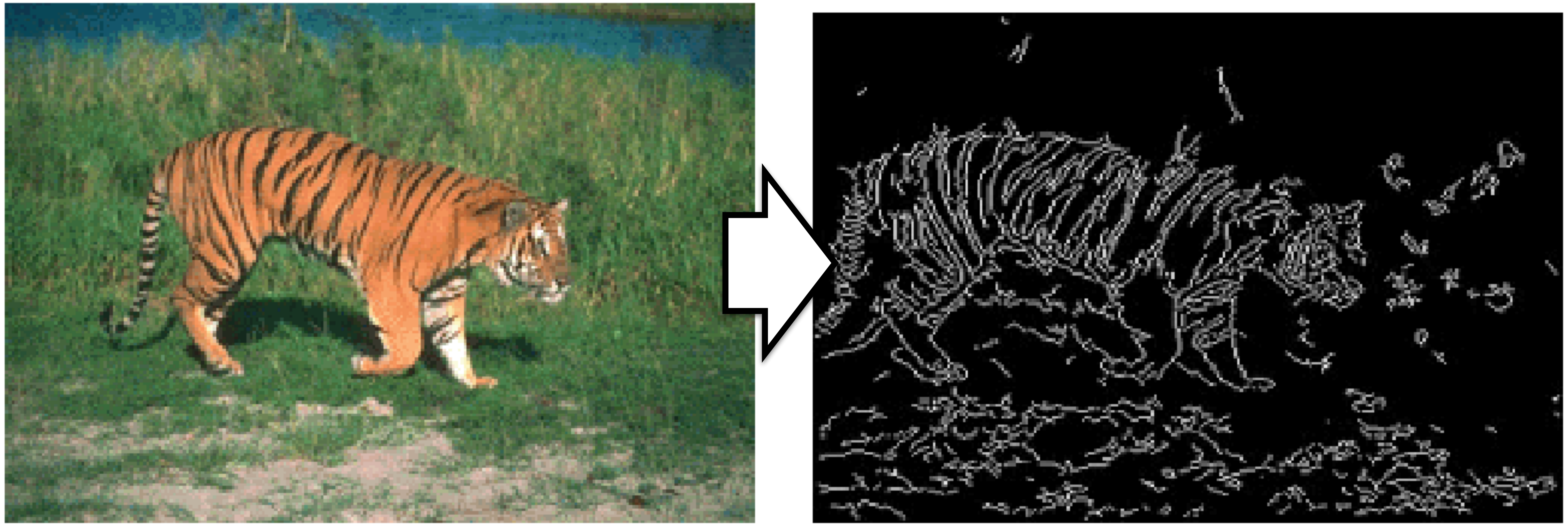
Edge-Based Image Segmentation



- **Edgels**: significant edges from edge detector (e.g., Canny)
- **Processing**: link edges, followed by relaxation, voting, dynamic programming, etc. to obtain region boundaries

slide credit: Václav Hlaváč

Edge-Based Image Segmentation



- **Edgels**: significant edges from edge detector (e.g., Canny)
- **Processing**: link edges, followed by relaxation, voting, dynamic programming, etc. to obtain region boundaries
- Leads to **partial segmentation** that requires post-processing, some regions might not be segmented at all

slide credit: Václav Hlaváč

Edge-Based Image Segmentation

original image

edge image
(enhanced for display)

slide credit: Václav Hlaváč

Edge-Based Image Segmentation

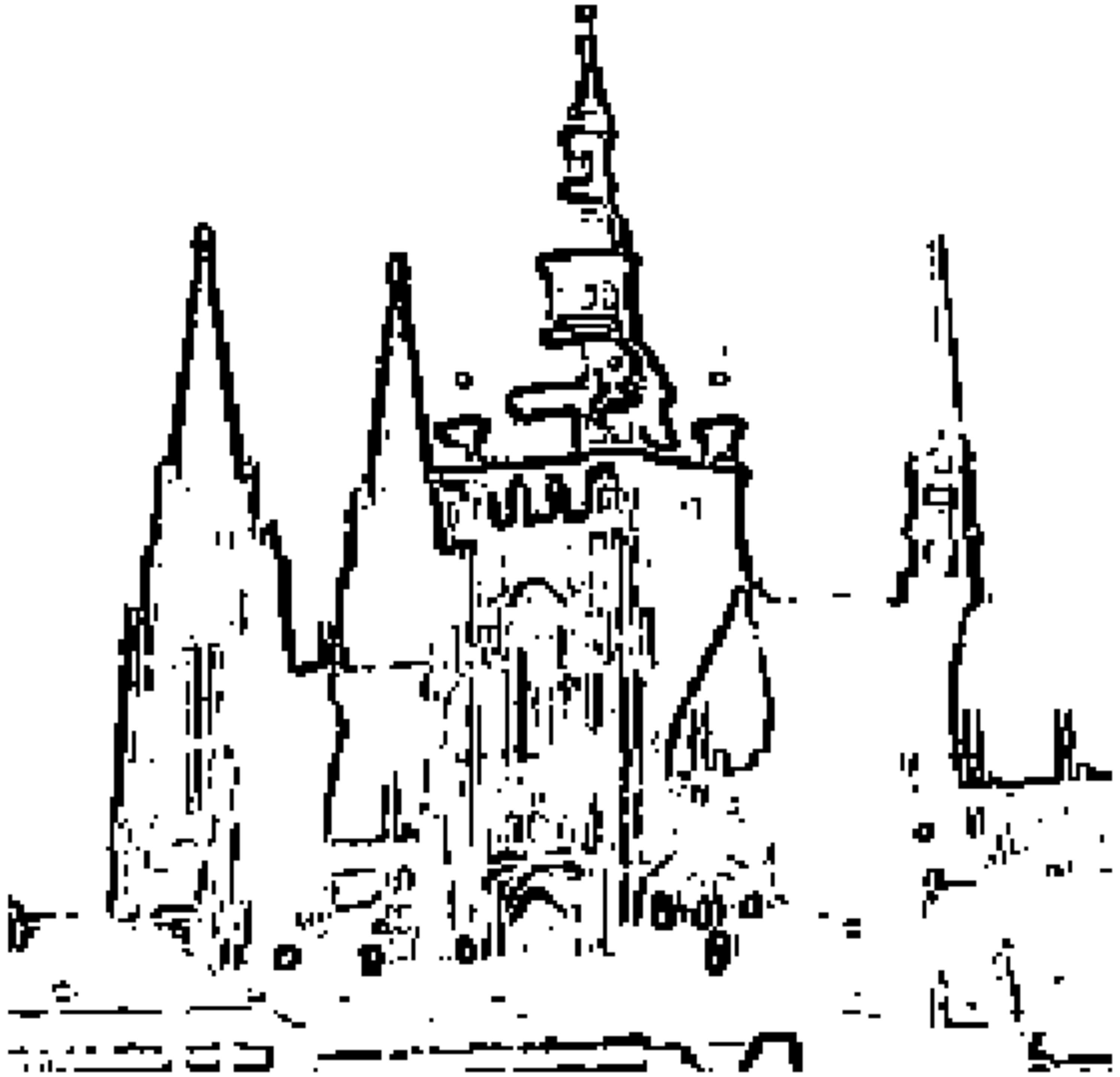
original image

edge image
(enhanced for display)

How to get from edge(l)s to region boundaries?

slide credit: Václav Hlaváč

Thresholding Edge Responses



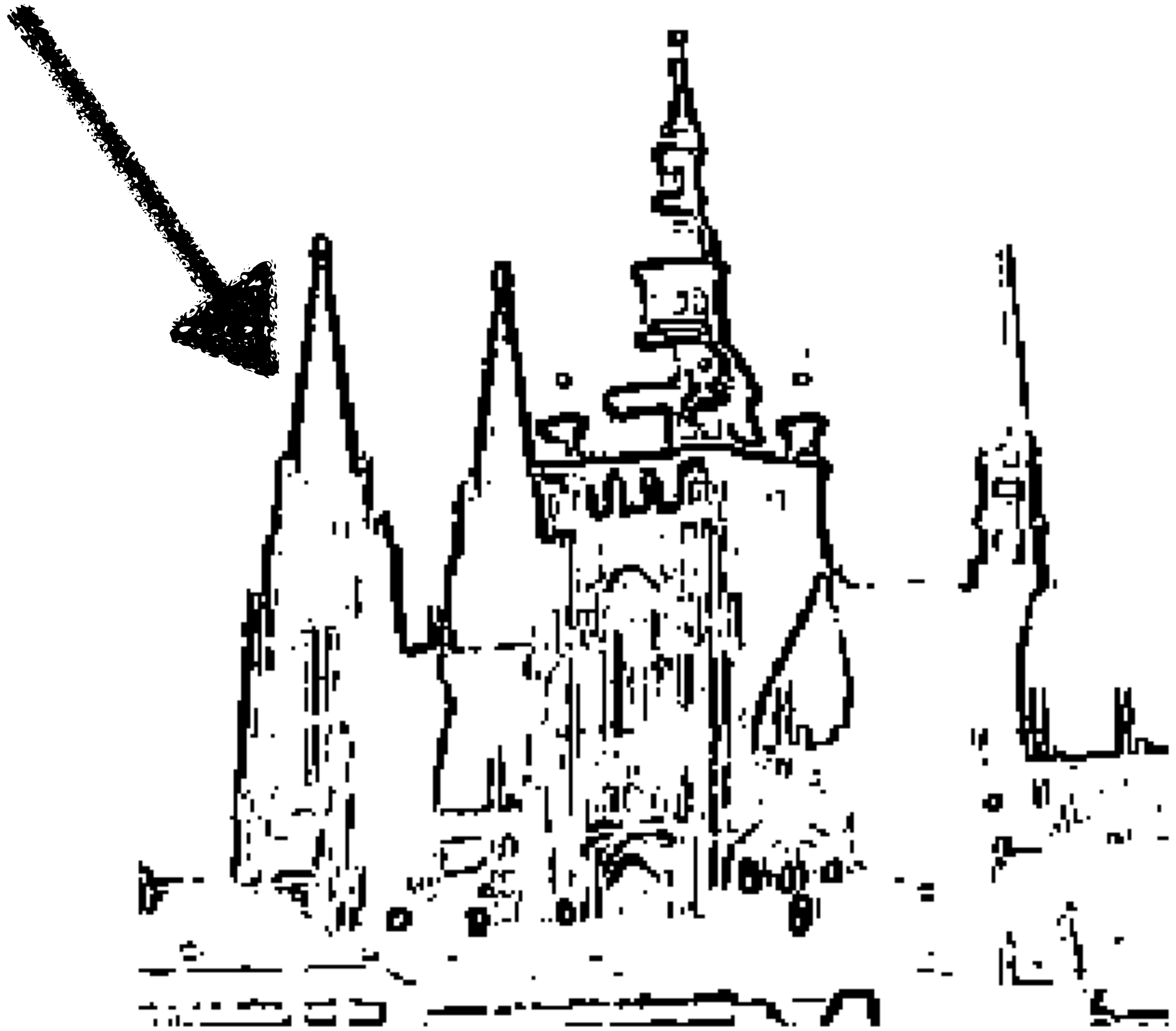
threshold 30

threshold 10 (too small)

slide credit: Václav Hlaváč

Thresholding Edge Responses

thick edges, multiple pixels wide



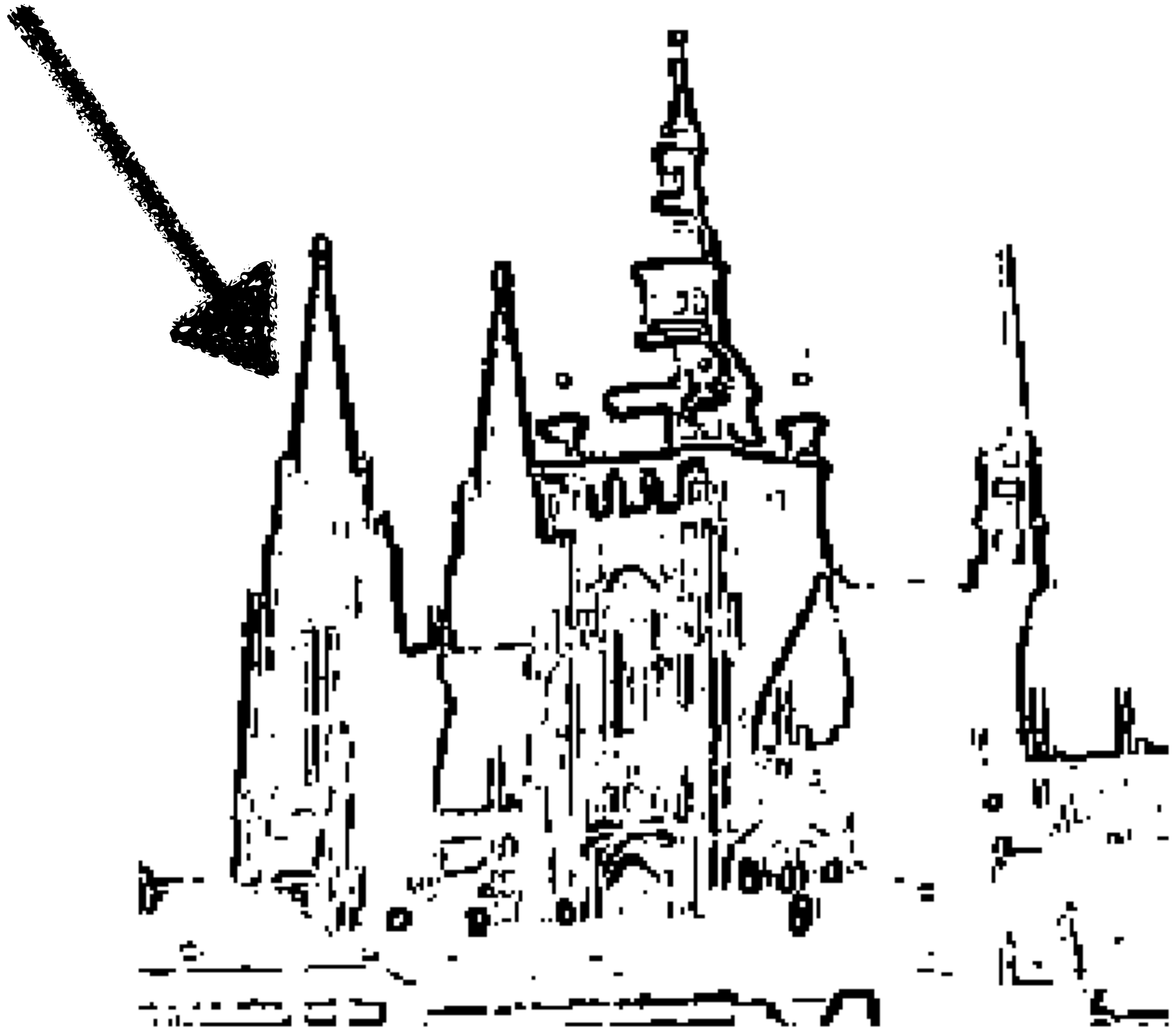
threshold 30

threshold 10 (too small)

slide credit: Václav Hlaváč

Thresholding Edge Responses

thick edges, multiple pixels wide



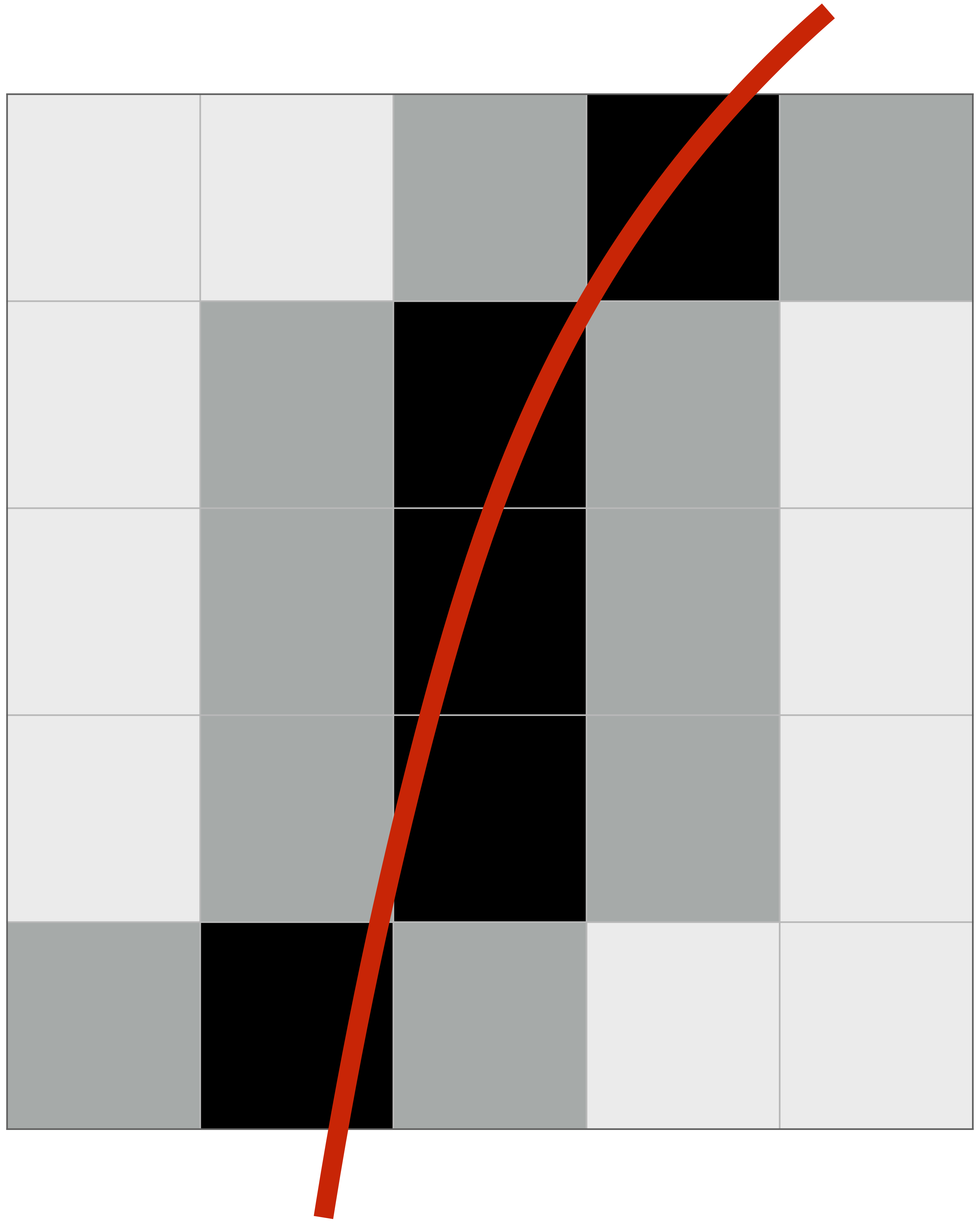
threshold 30

threshold 10 (too small)

How to get thin edges?

slide credit: Václav Hlaváč

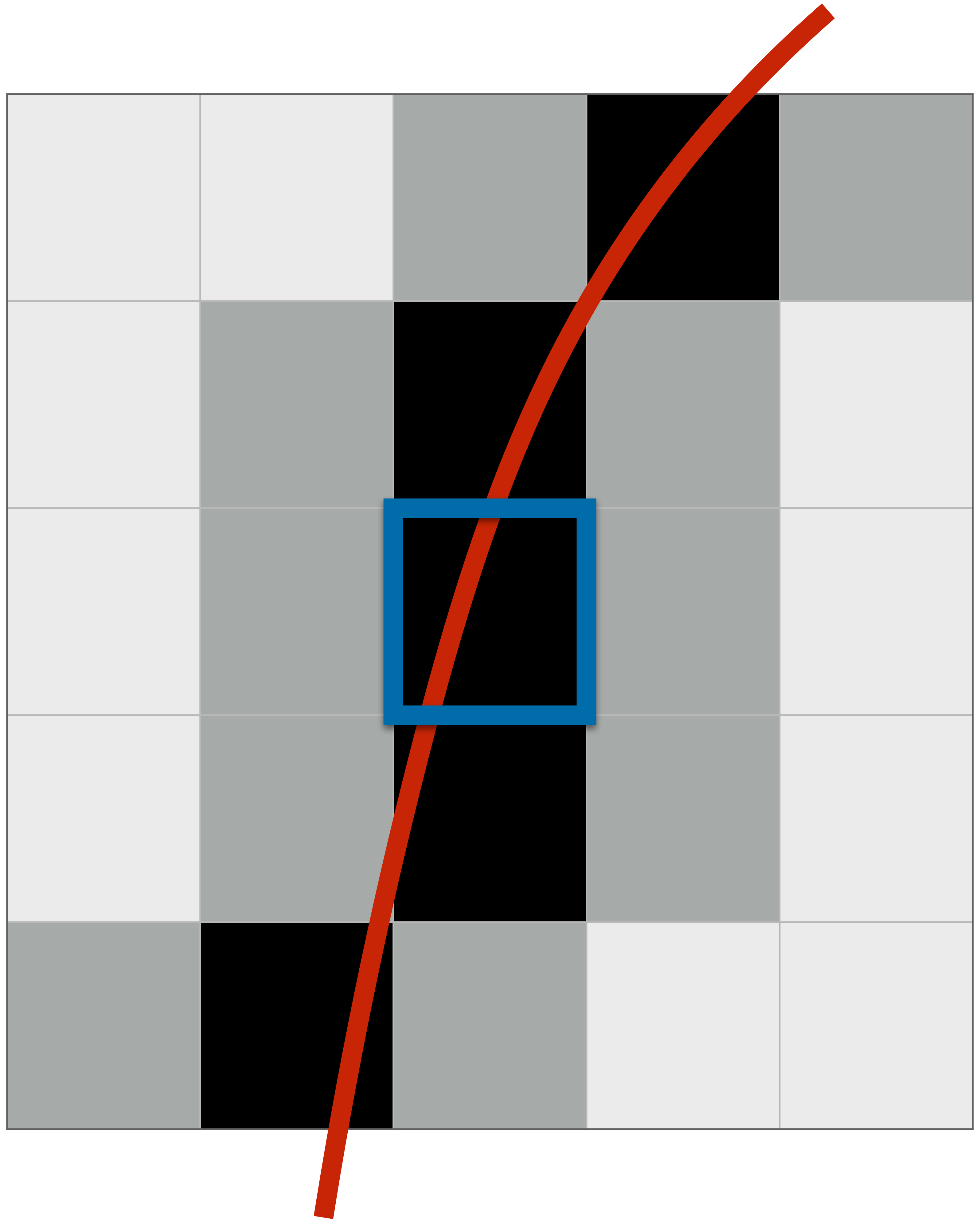
Non-Maximum Suppression For Thin Edges



- Compare pixel intensity to neighbors
- Keep only local maxima

slide credit: Václav Hlaváč

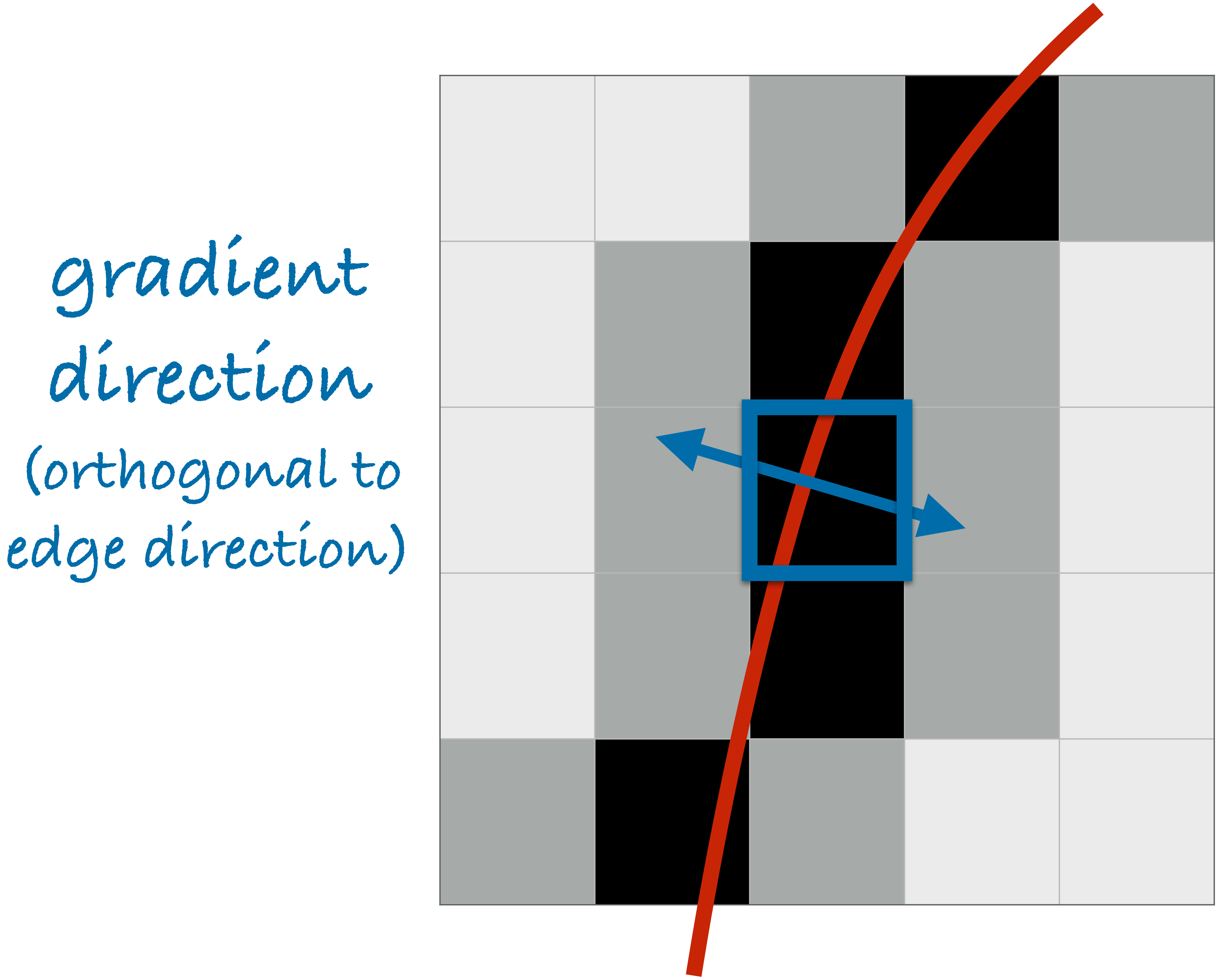
Non-Maximum Suppression For Thin Edges



- Compare pixel intensity to neighbors
- Keep only local maxima

slide credit: Václav Hlaváč

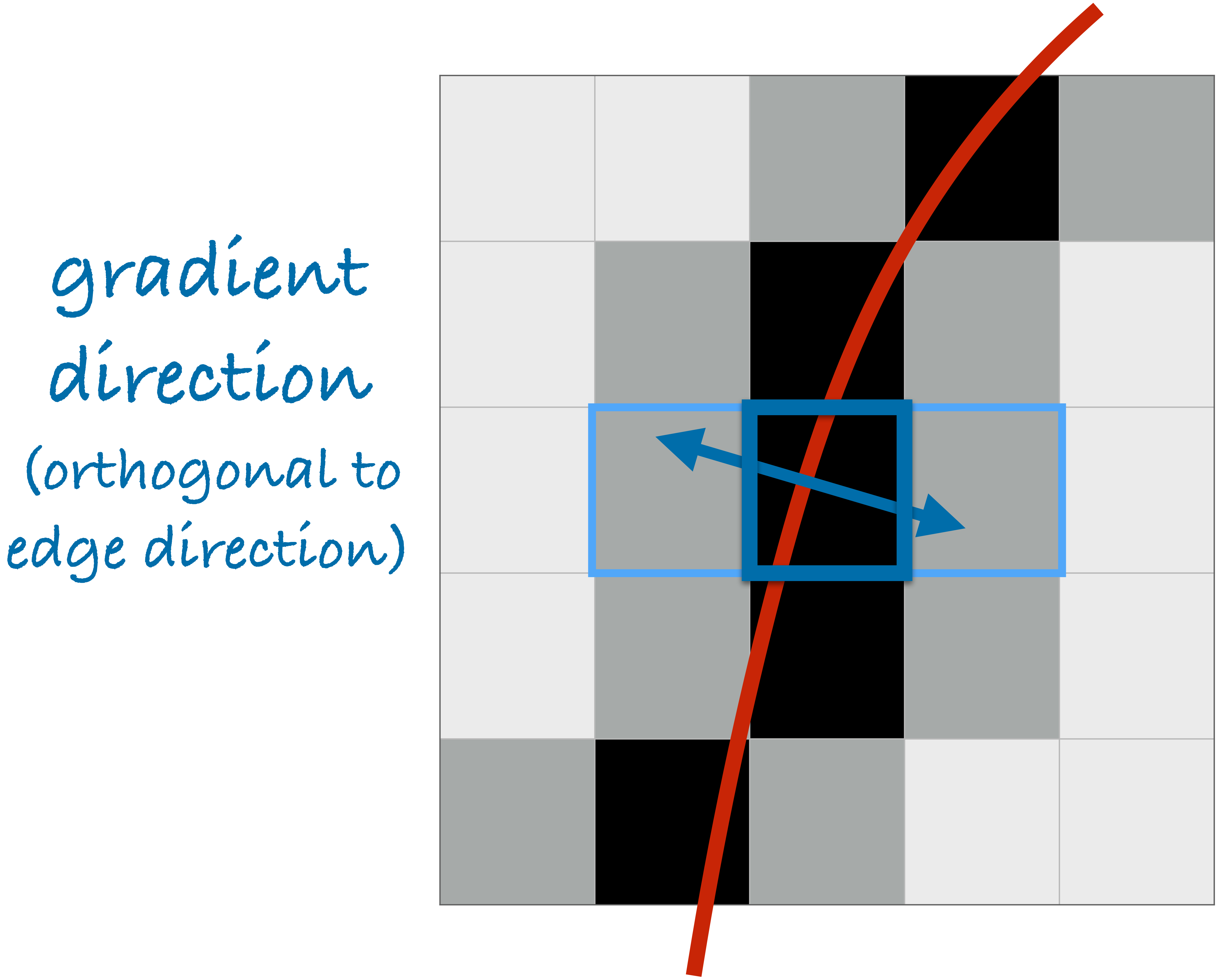
Non-Maximum Suppression For Thin Edges



- Compare pixel intensity to neighbors
- Keep only local maxima

slide credit: Václav Hlaváč

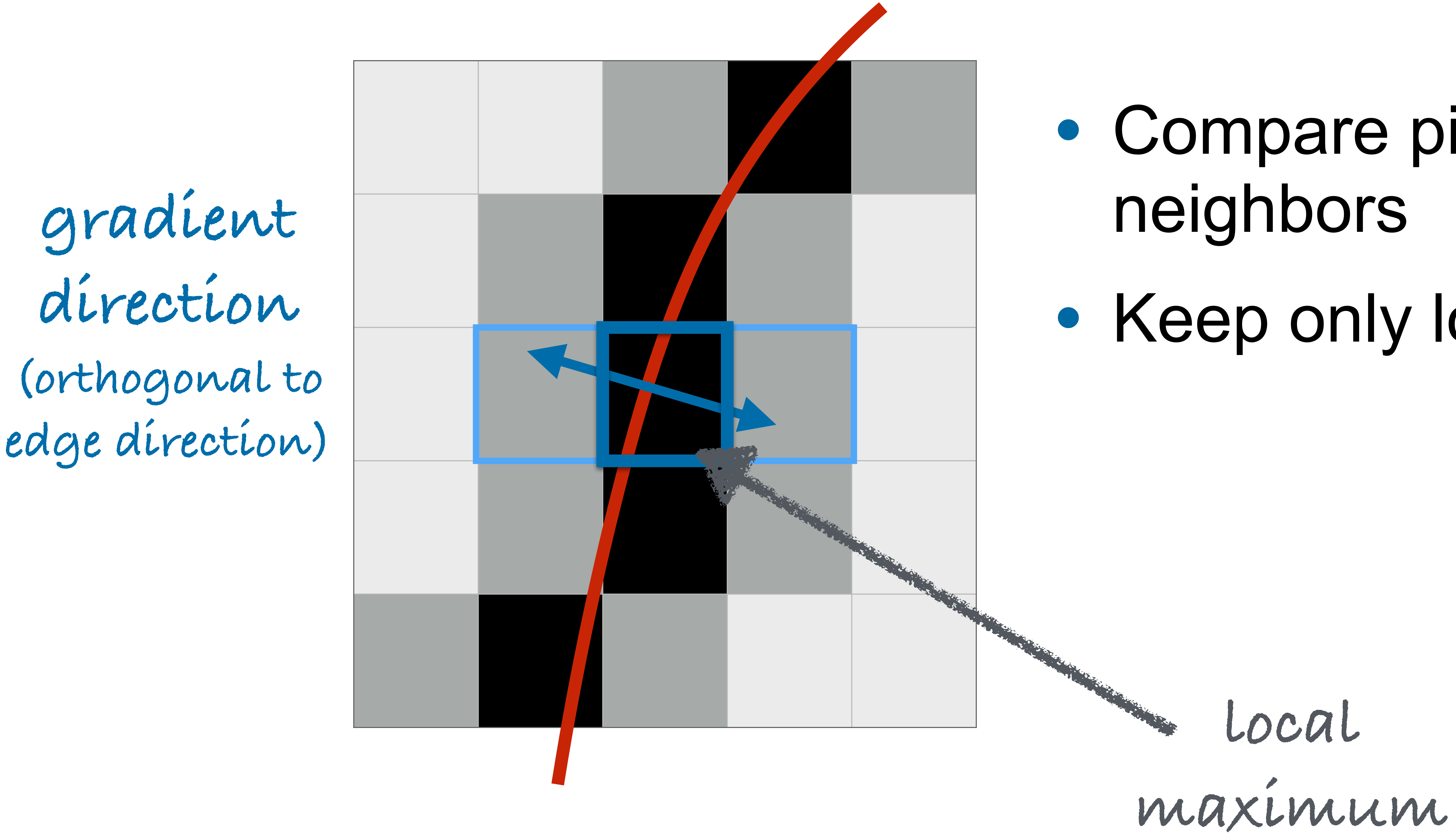
Non-Maximum Suppression For Thin Edges



- Compare pixel intensity to neighbors
- Keep only local maxima

slide credit: Václav Hlaváč

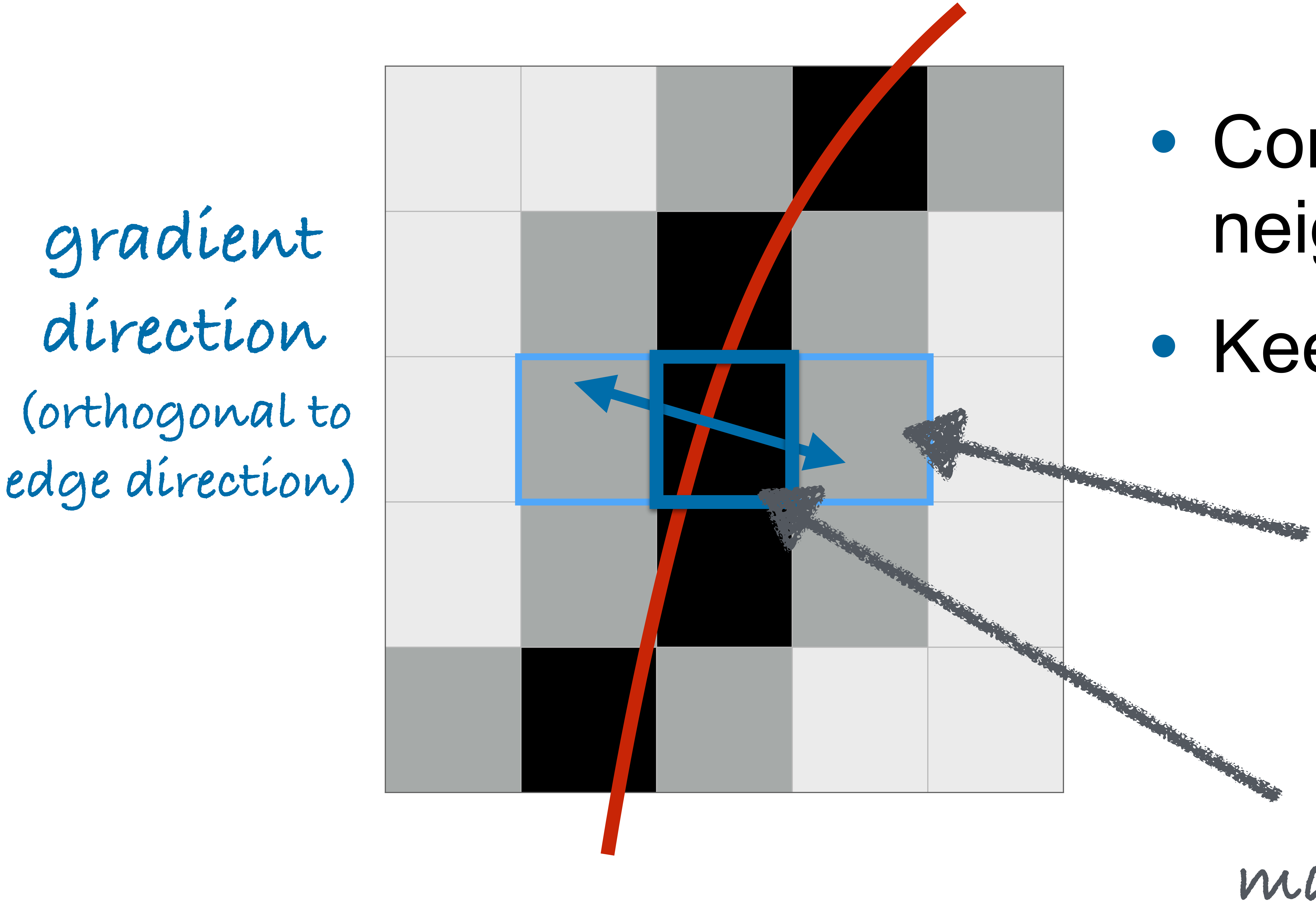
Non-Maximum Suppression For Thin Edges



- Compare pixel intensity to neighbors
- Keep only local maxima

slide credit: Václav Hlaváč

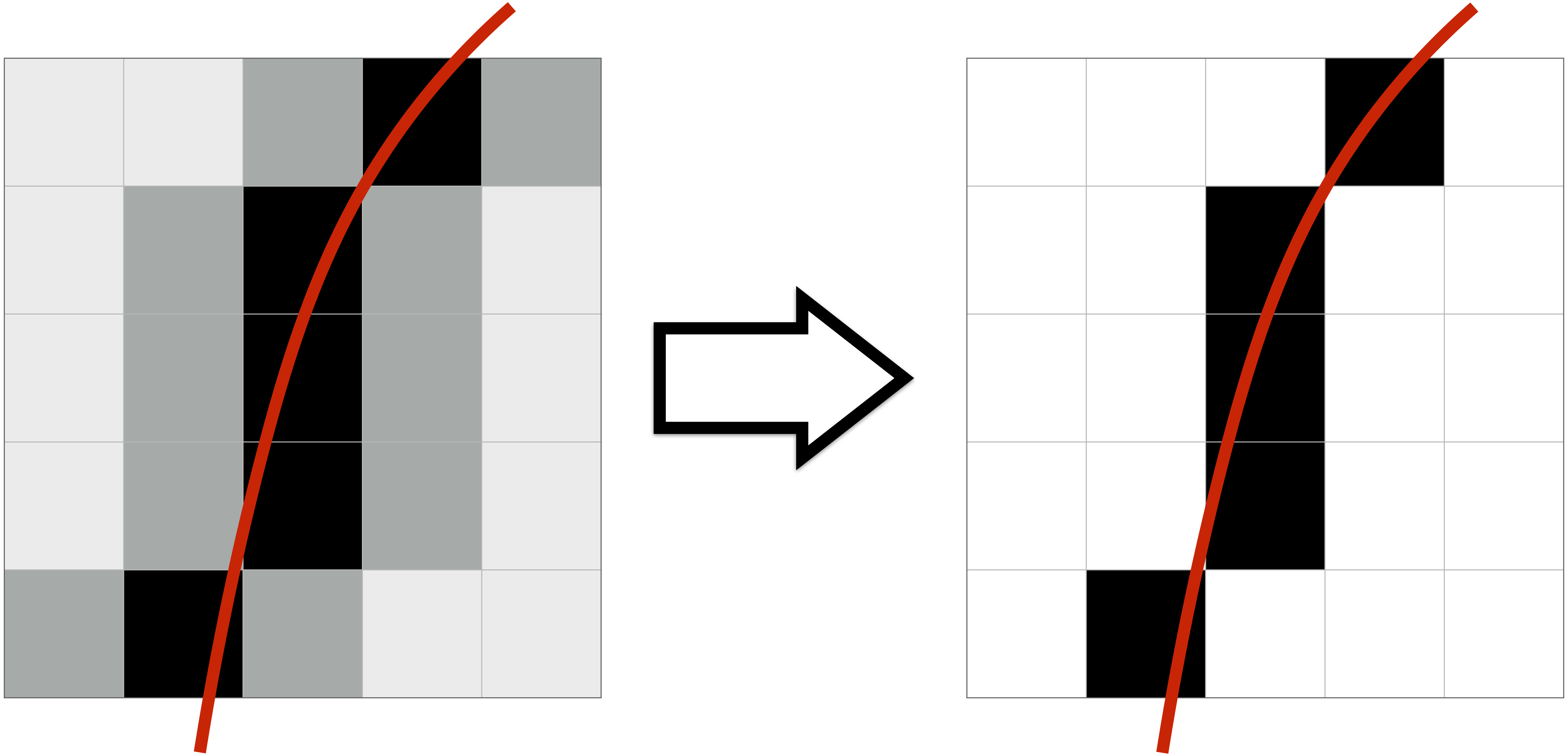
Non-Maximum Suppression For Thin Edges



- Compare pixel intensity to neighbors
- Keep only local maxima

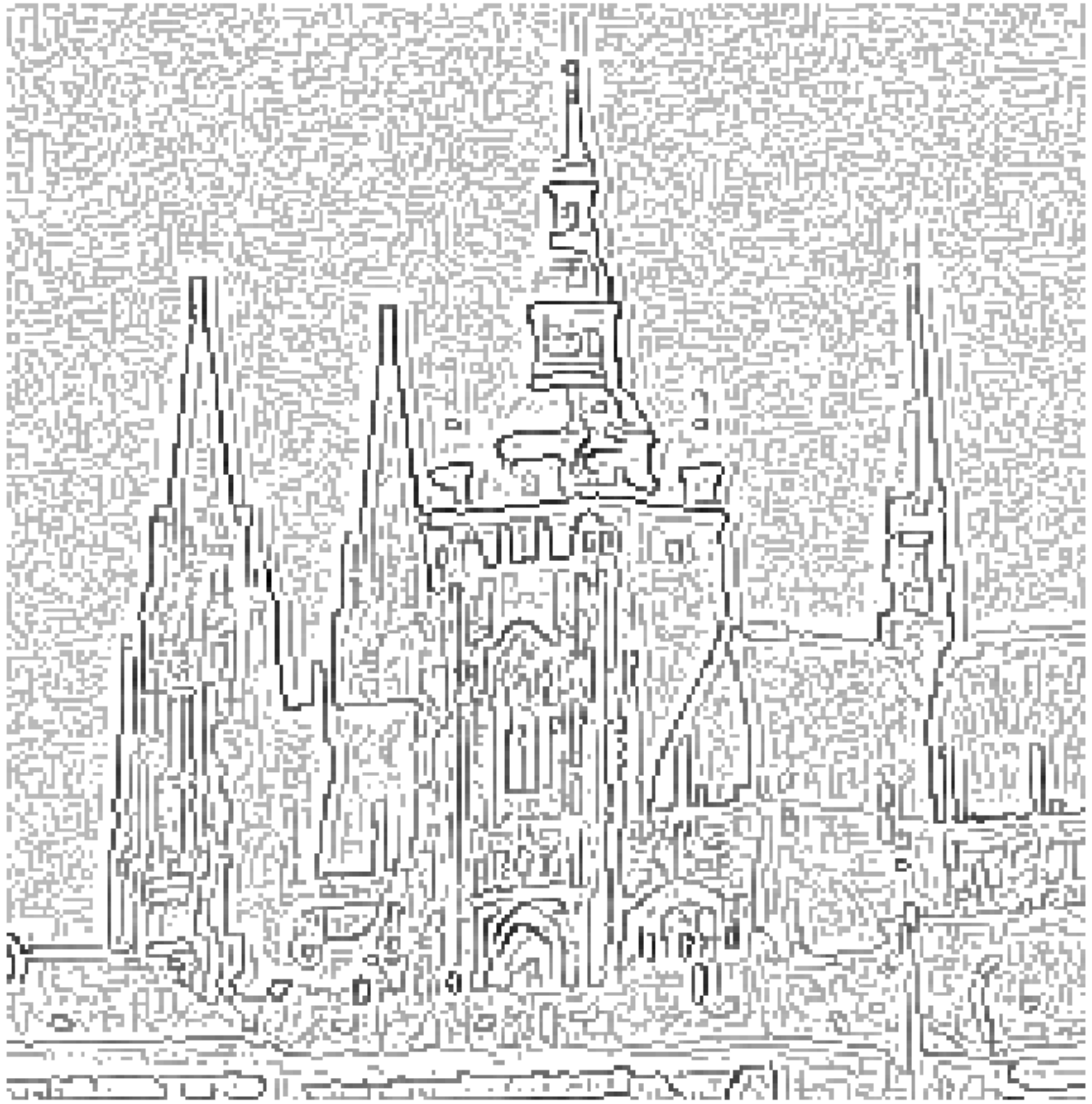
slide credit: Václav Hlaváč

Non-Maximum Suppression For Thin Edges



slide credit: Václav Hlaváč

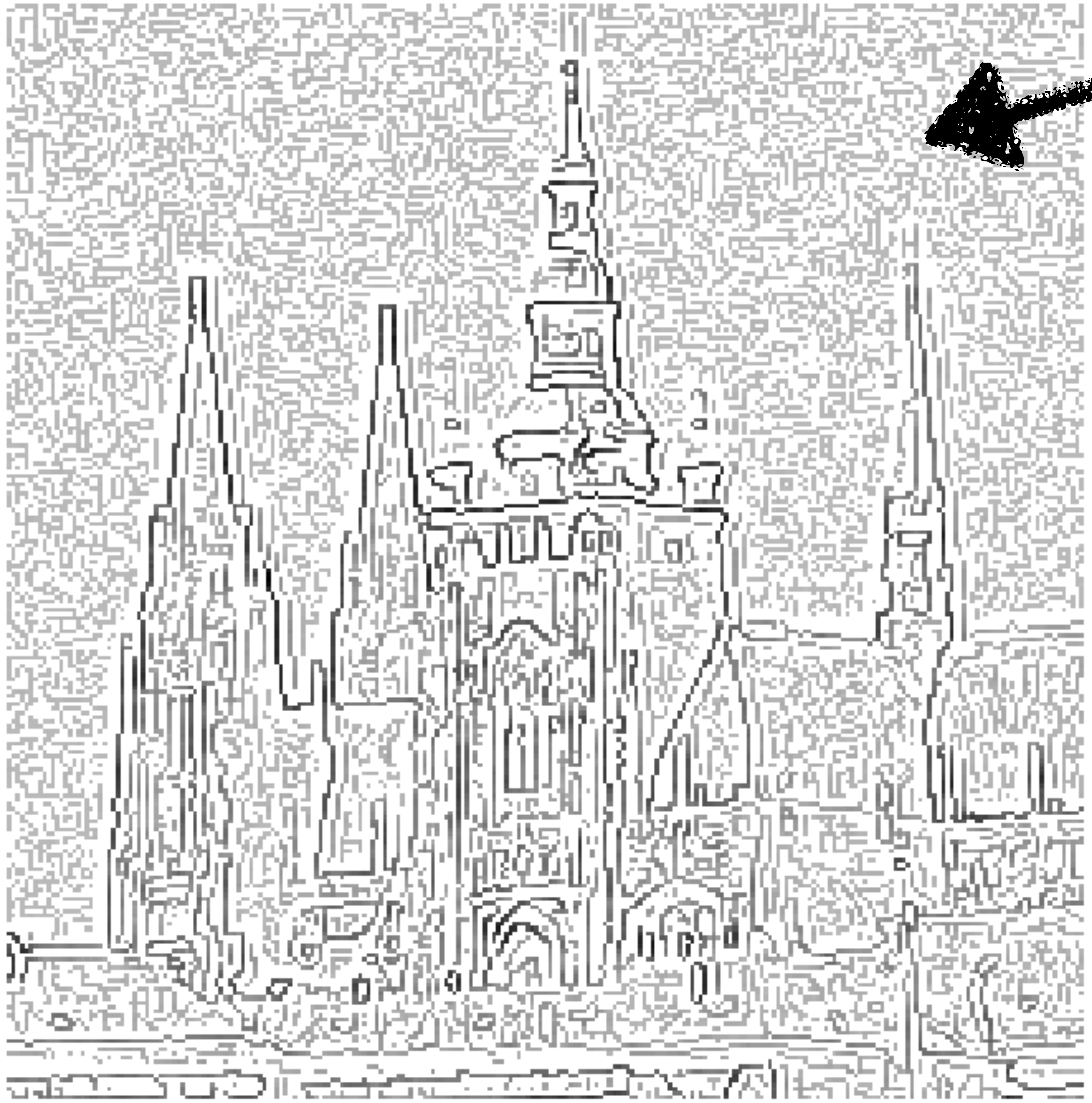
Non-Maximum Suppression & Hysteresis



Non-maximal suppression

slide credit: Václav Hlaváč

Non-Maximum Suppression & Hysteresis

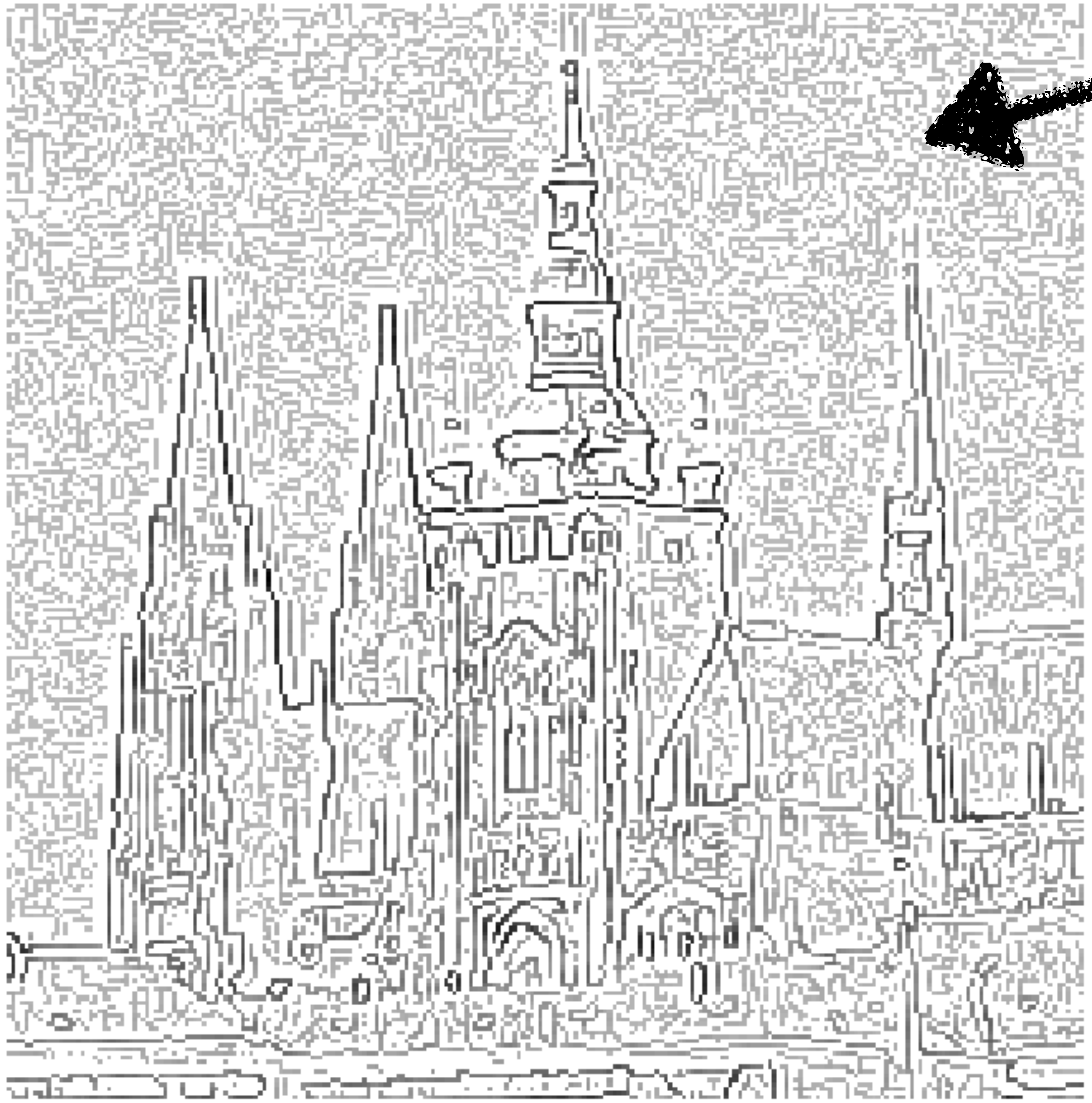


Non-maximal suppression

lots of noisy edges

slide credit: Václav Hlaváč

Non-Maximum Suppression & Hysteresis



Non-maximal suppression

lots of noisy edges

Hysteresis:

slide credit: Václav Hlaváč

Non-Maximum Suppression & Hysteresis

Non-maximal suppression

lots of noisy edges

Hysteresis:

- Start with edge pixel with edge score above higher threshold t_{high}

slide credit: Václav Hlaváč

Non-Maximum Suppression & Hysteresis

Non-maximal suppression

lots of noisy edges

Hysteresis:

- Start with edge pixel with edge score above higher threshold t_{high}
- Follow edge as long as pixels have edge score above lower threshold t_{low}

slide credit: Václav Hlaváč

Non-Maximum Suppression & Hysteresis

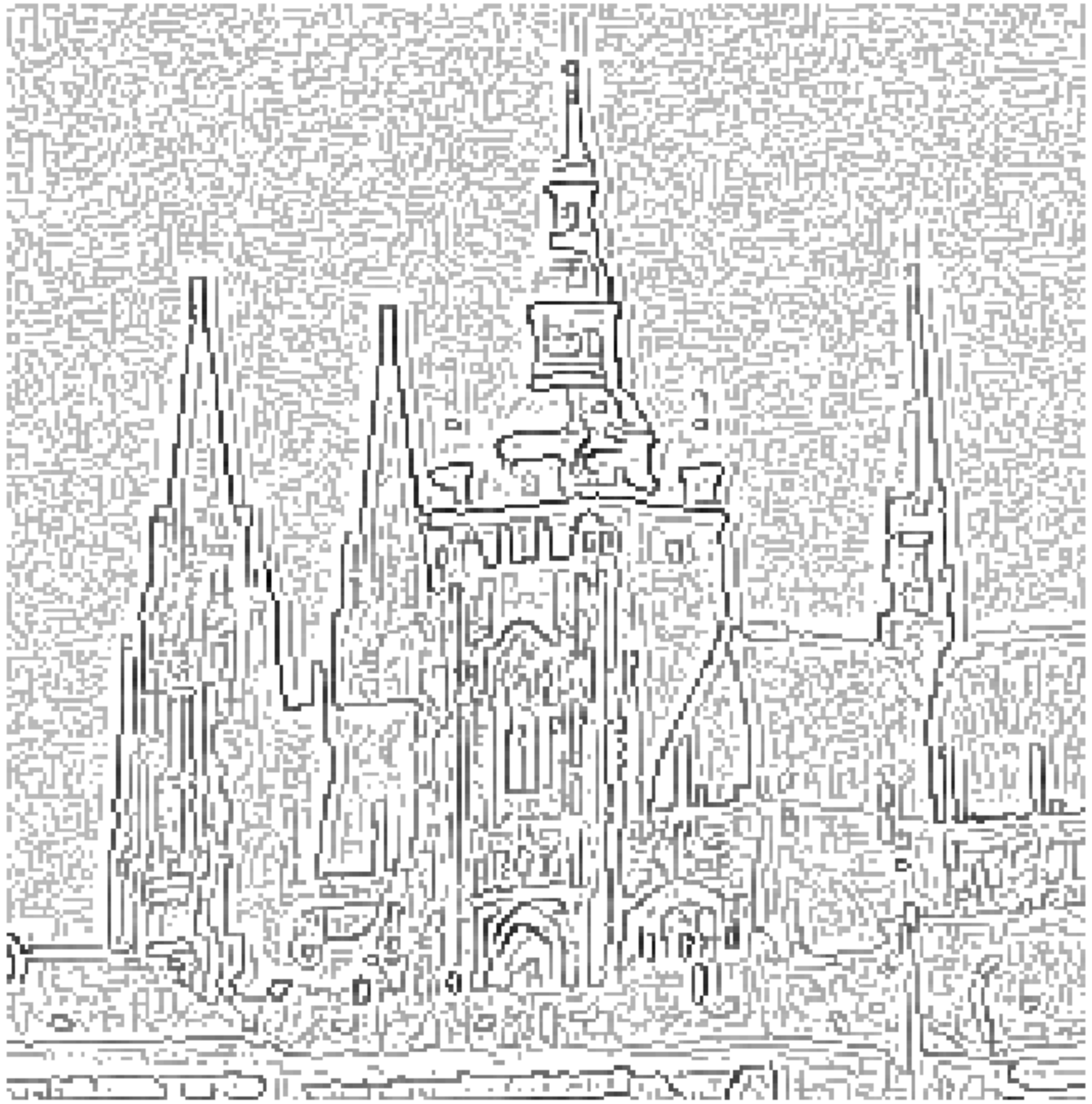
Non-maximal suppression

Hysteresis:

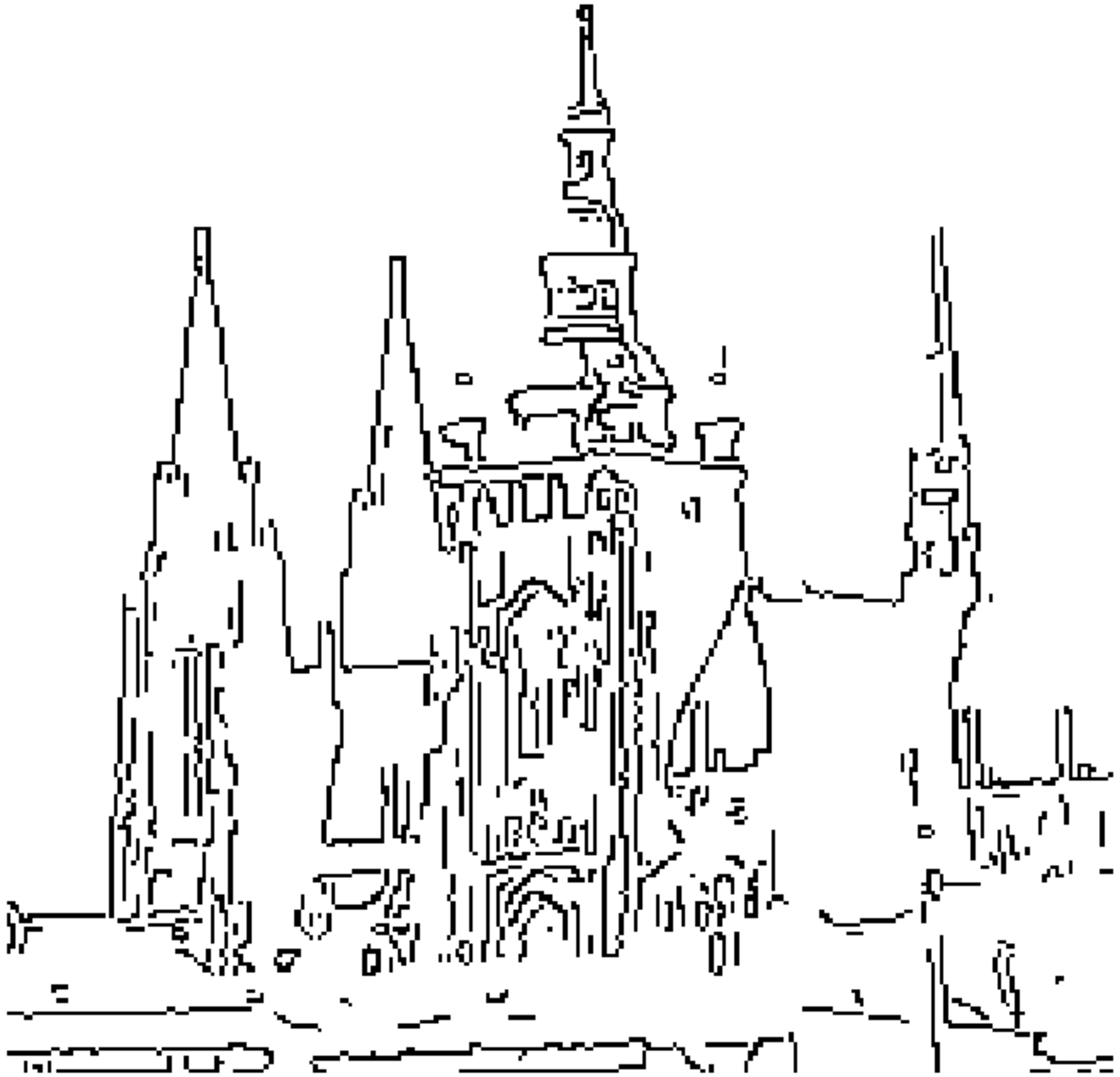
- Start with edge pixel with edge score above higher threshold t_{high}
- Follow edge as long as pixels have edge score above lower threshold t_{low}
- Iterate until all pixels considered

slide credit: Václav Hlaváč

Edge-Based Image Segmentation



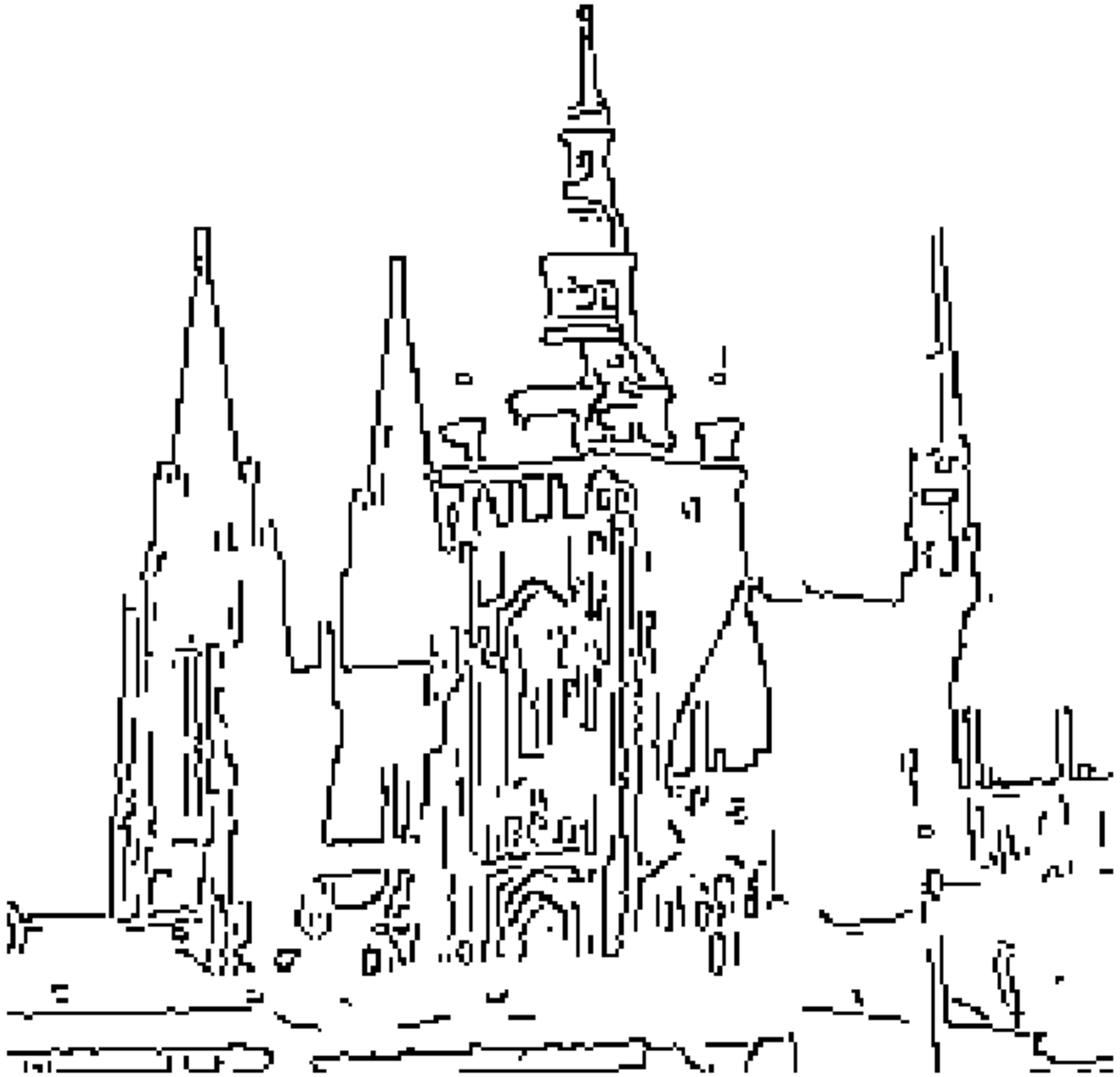
Non-maximal suppression



hysteresis
high threshold 70, lower 10

slide credit: Václav Hlaváč

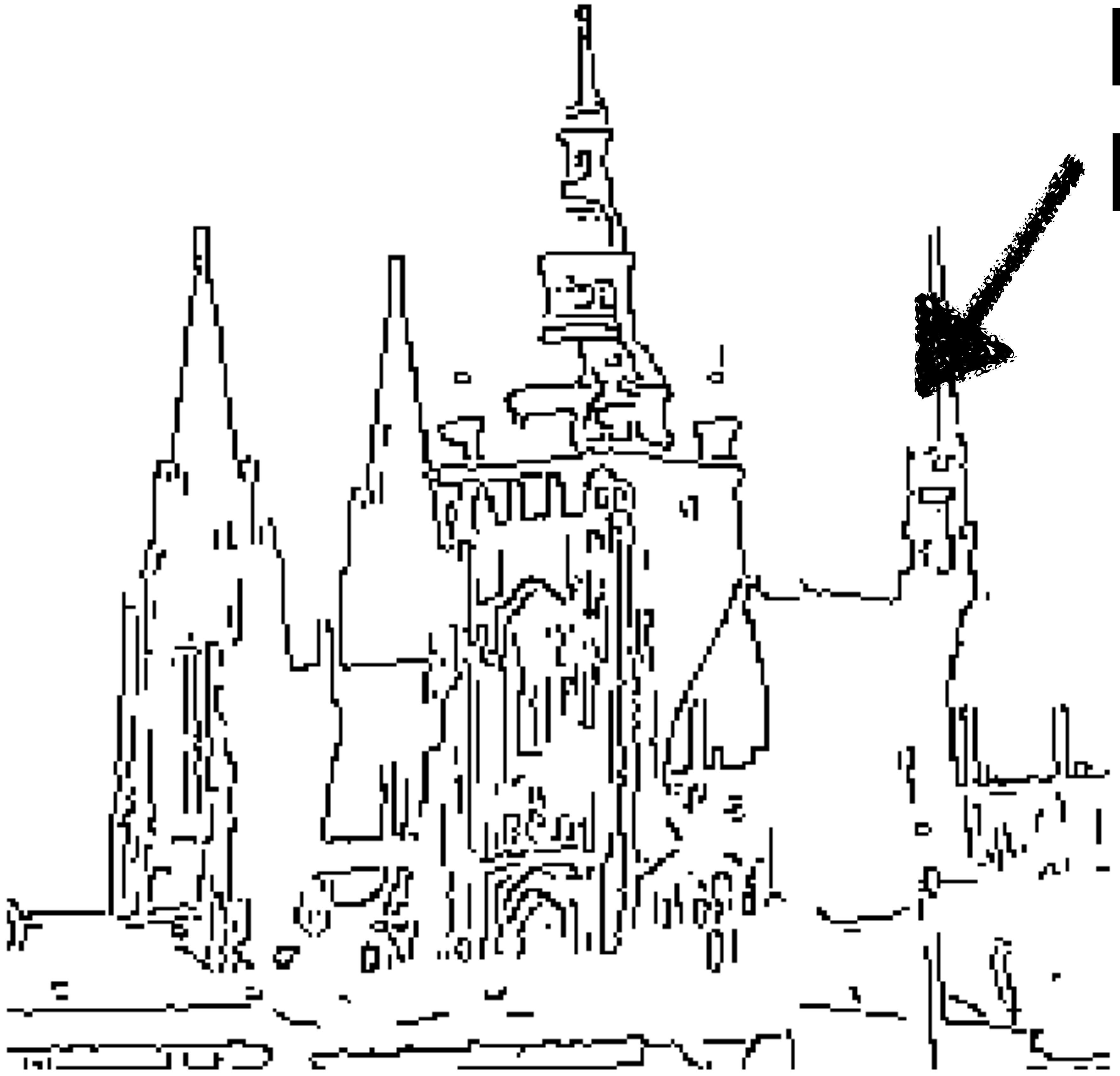
Edge Relaxation



hysteresis
high threshold 70, lower 10

slide credit: Václav Hlaváč

Edge Relaxation

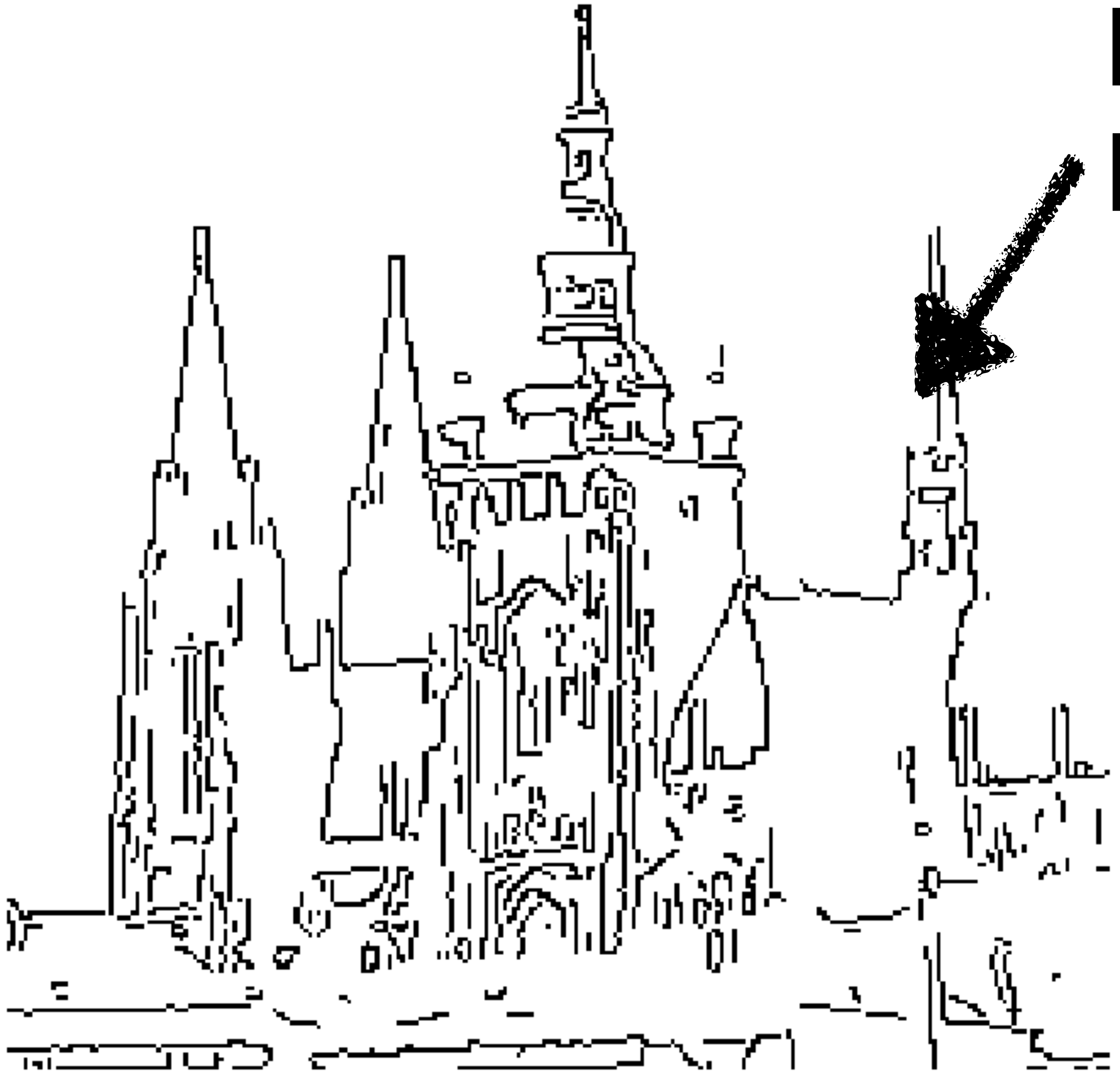


hysteresis
high threshold 70, lower 10

No closed boundaries
Parts missing

slide credit: Václav Hlaváč

Edge Relaxation



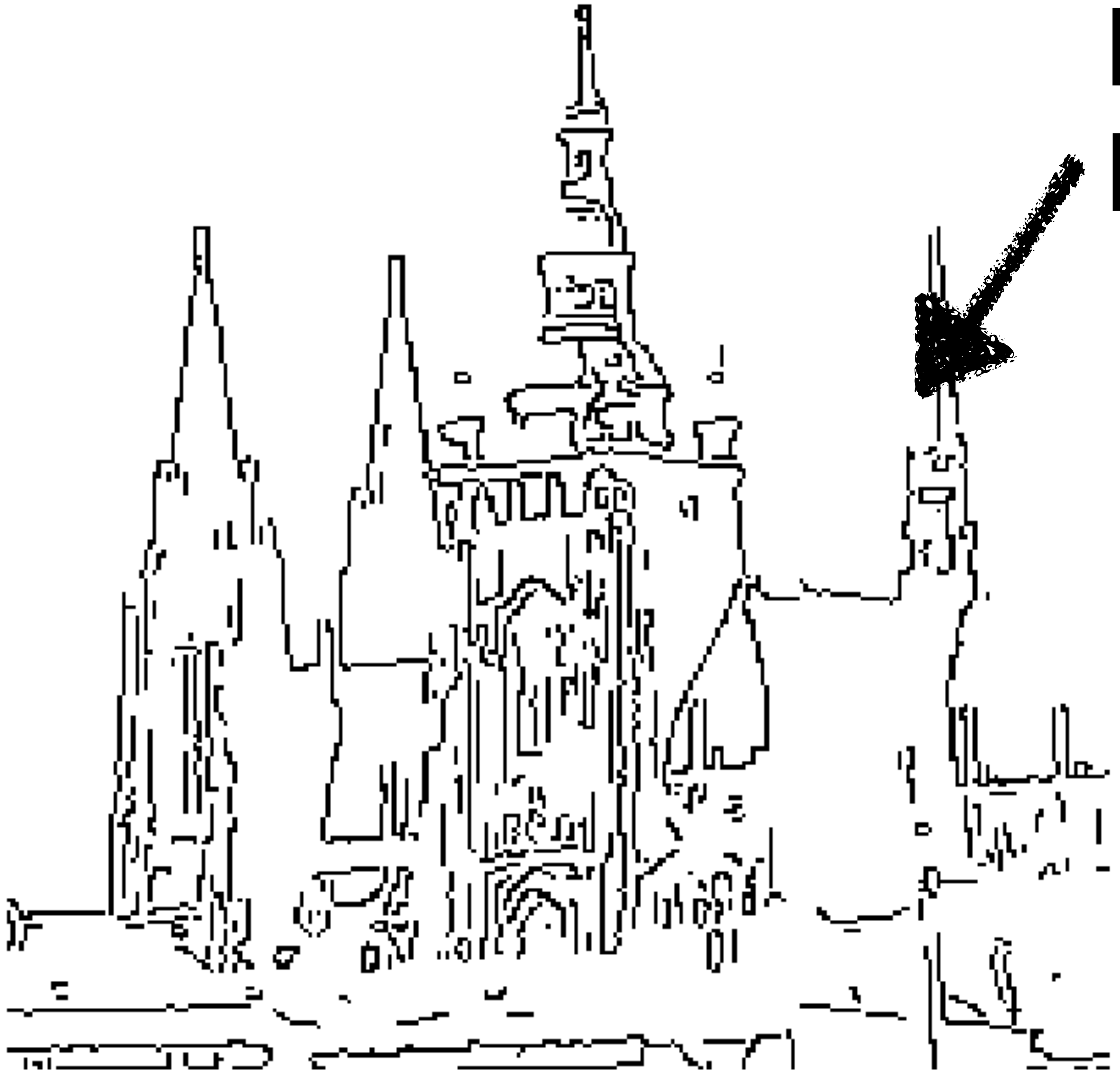
hysteresis
high threshold 70, lower 10

No closed boundaries
Parts missing

Edge Relaxation:

slide credit: Václav Hlaváč

Edge Relaxation



hysteresis
high threshold 70, lower 10

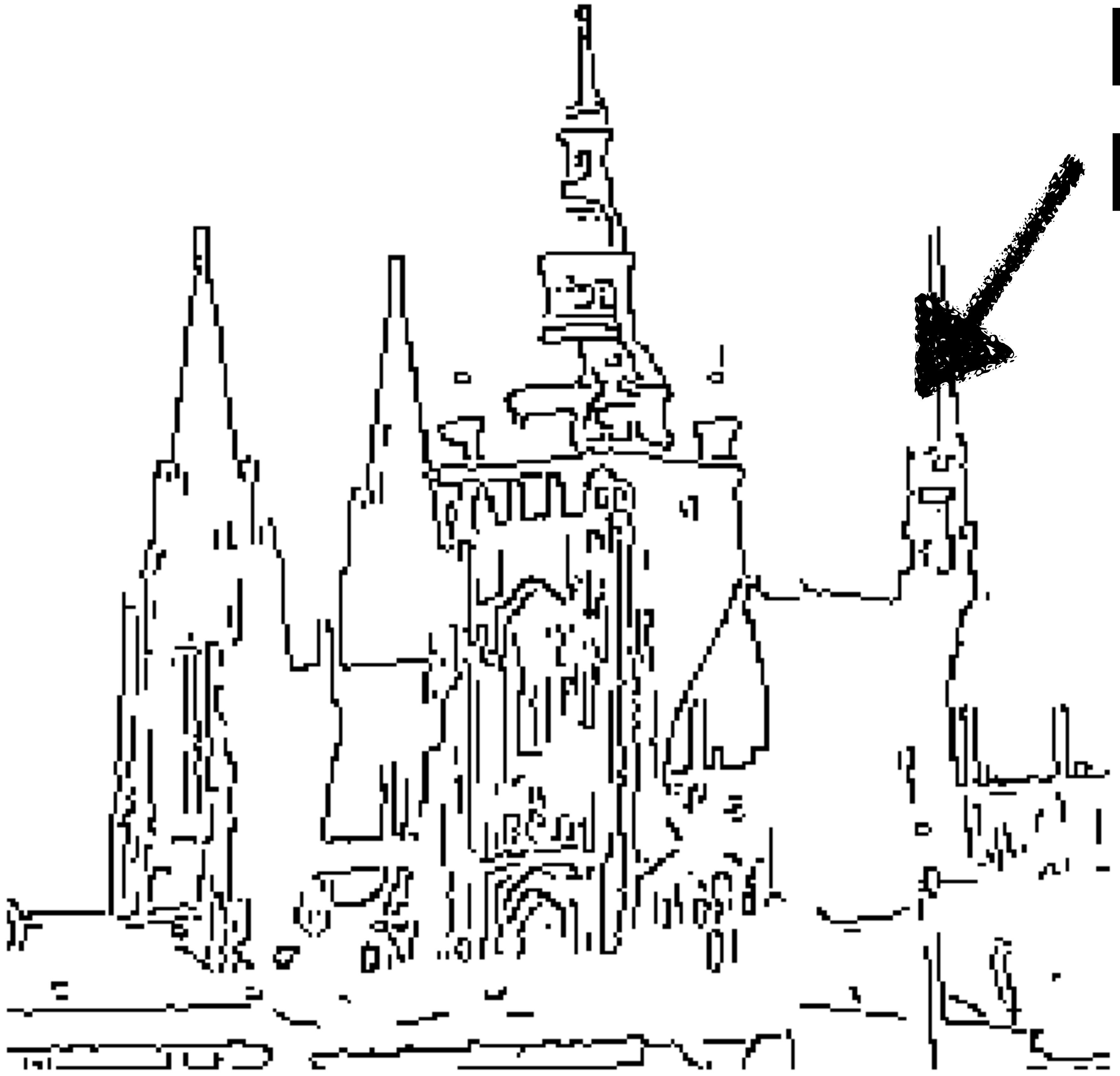
No closed boundaries
Parts missing

Edge Relaxation:

- Attempt to close gaps in post-processing

slide credit: Václav Hlaváč

Edge Relaxation



hysteresis
high threshold 70, lower 10

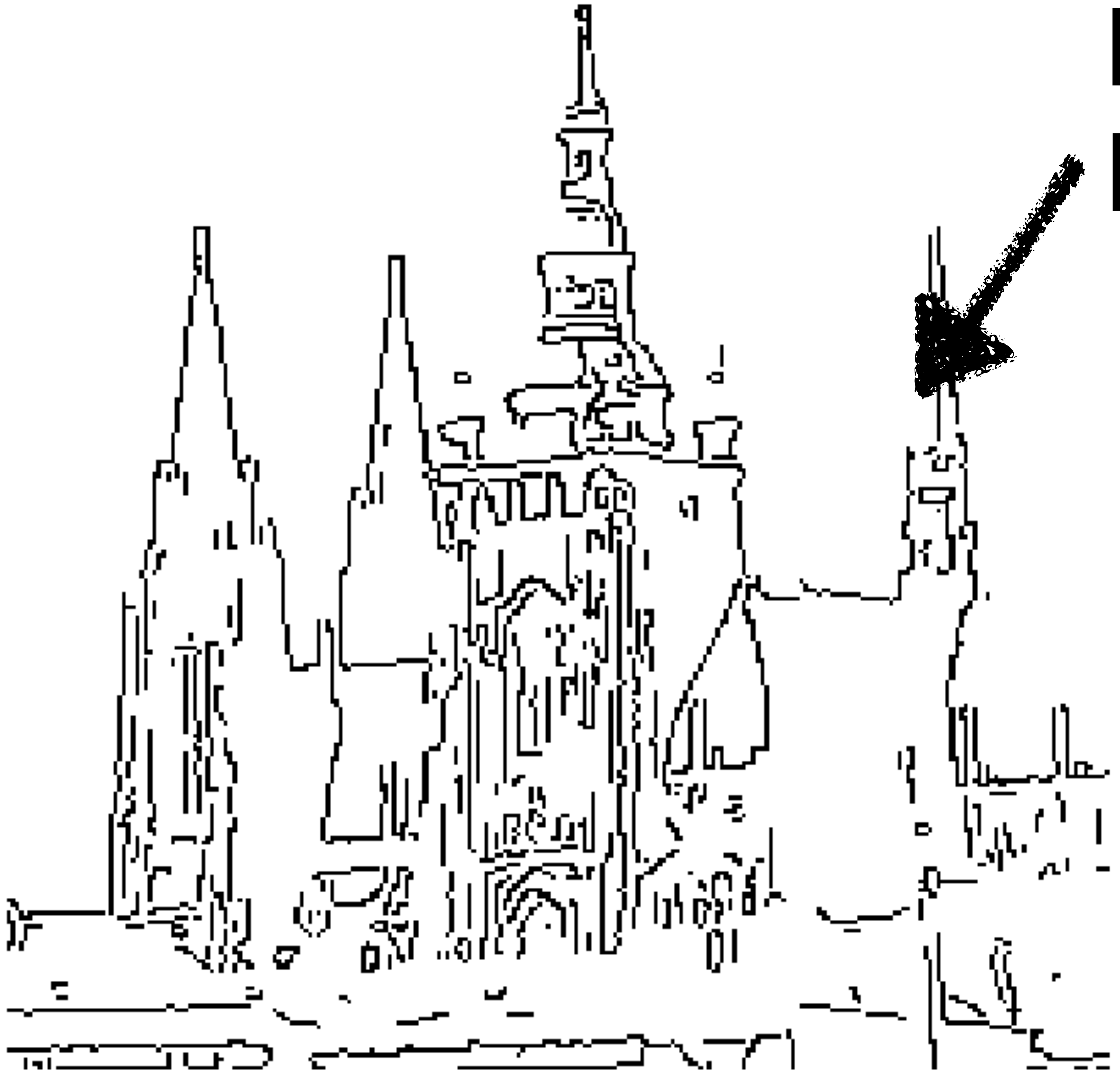
No closed boundaries
Parts missing

Edge Relaxation:

- Attempt to close gaps in post-processing
- Iteratively improve edge properties based on neighboring edges

slide credit: Václav Hlaváč

Edge Relaxation



No closed boundaries
Parts missing

hysteresis
high threshold 70, lower 10

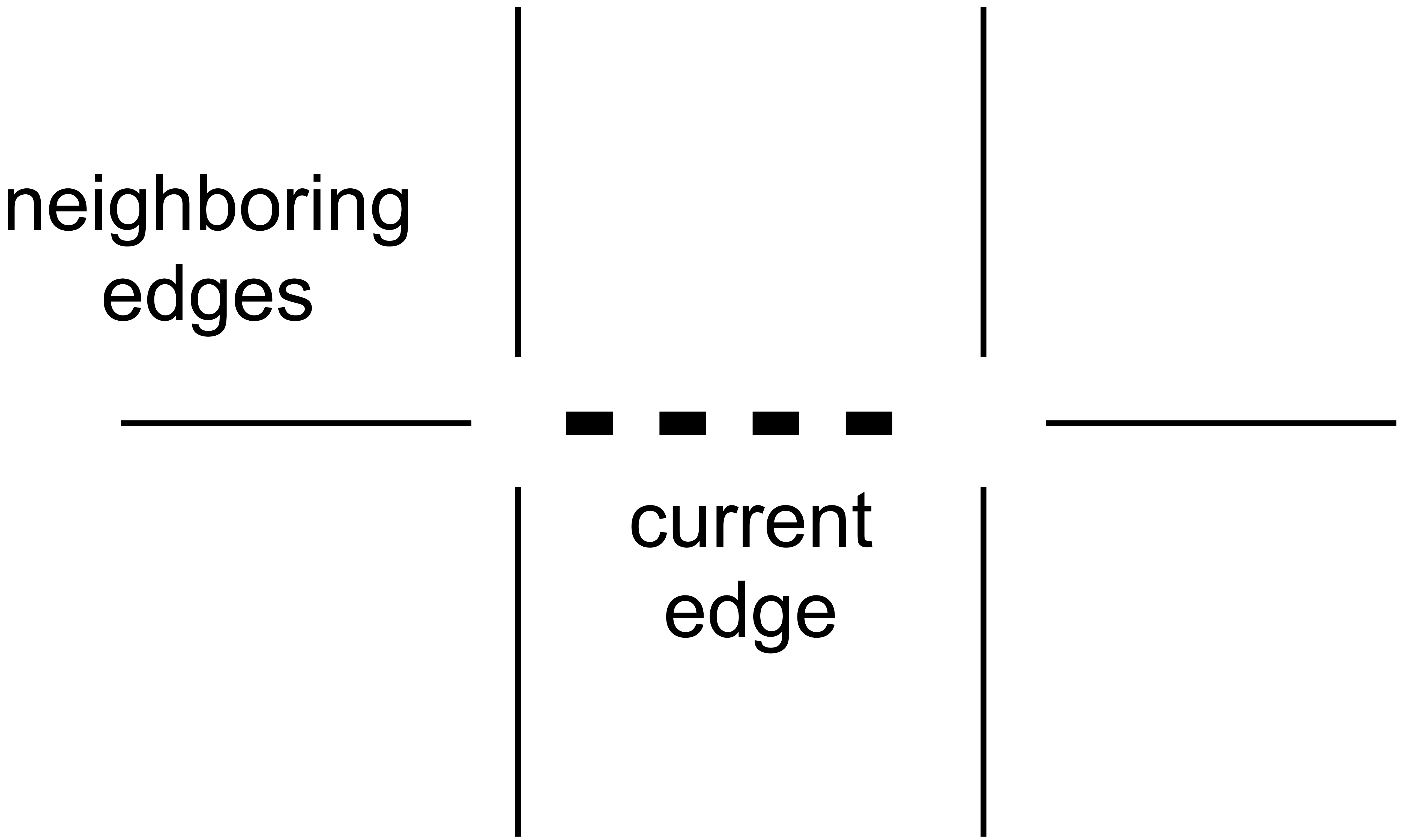
Edge Relaxation:

- Attempt to close gaps in post-processing
- Iteratively improve edge properties based on neighboring edges
- Instance of a general algorithm "relaxation labelling"

slide credit: Václav Hlaváč

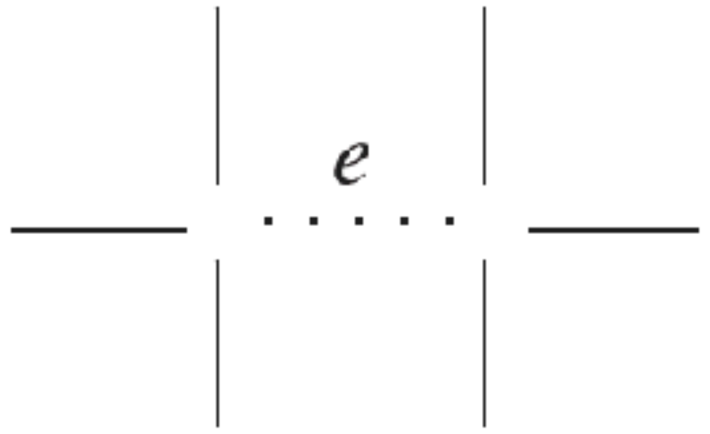
Edge Relaxation For Crack Edges

Example illustration here based on crack edges (Hanson, Rieseman, 1978)



slide credit: Václav Hlaváč

Crack Edges Patterns

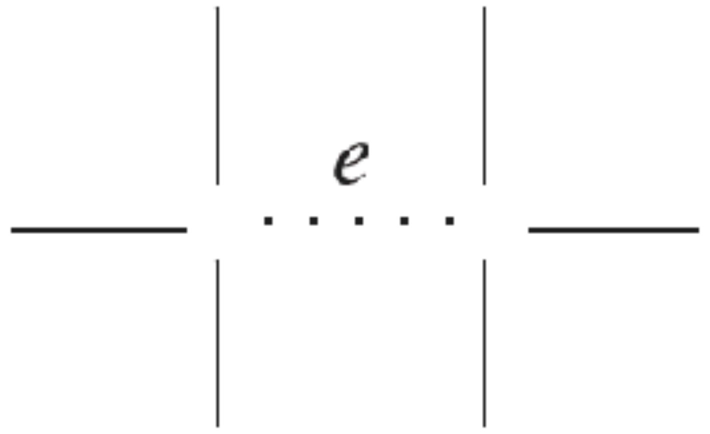


0–0

- **0-0** isolated edge → decrease edge confidence

slide credit: Václav Hlaváč

Crack Edges Patterns

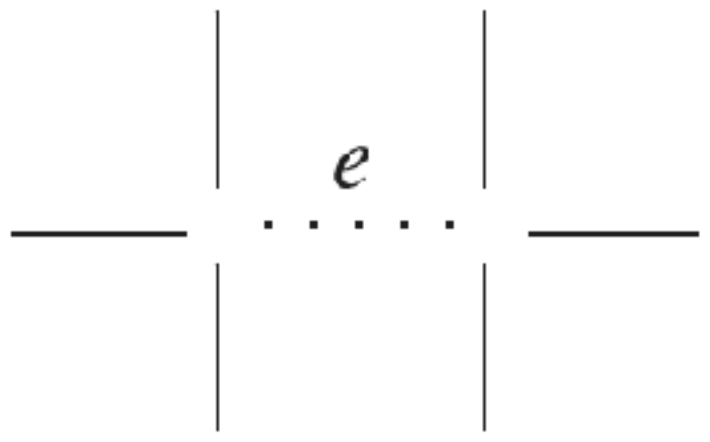


0–0

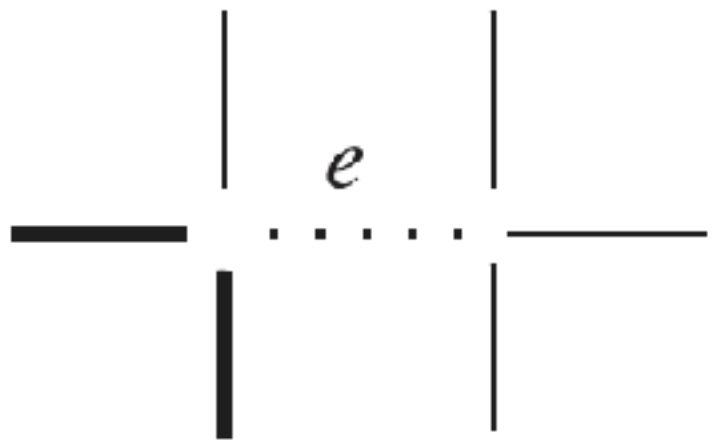
- **0-0** isolated edge → decrease edge confidence
- **0-1** uncertain → weak increase, or no influence

slide credit: Václav Hlaváč

Crack Edges Patterns



0–0

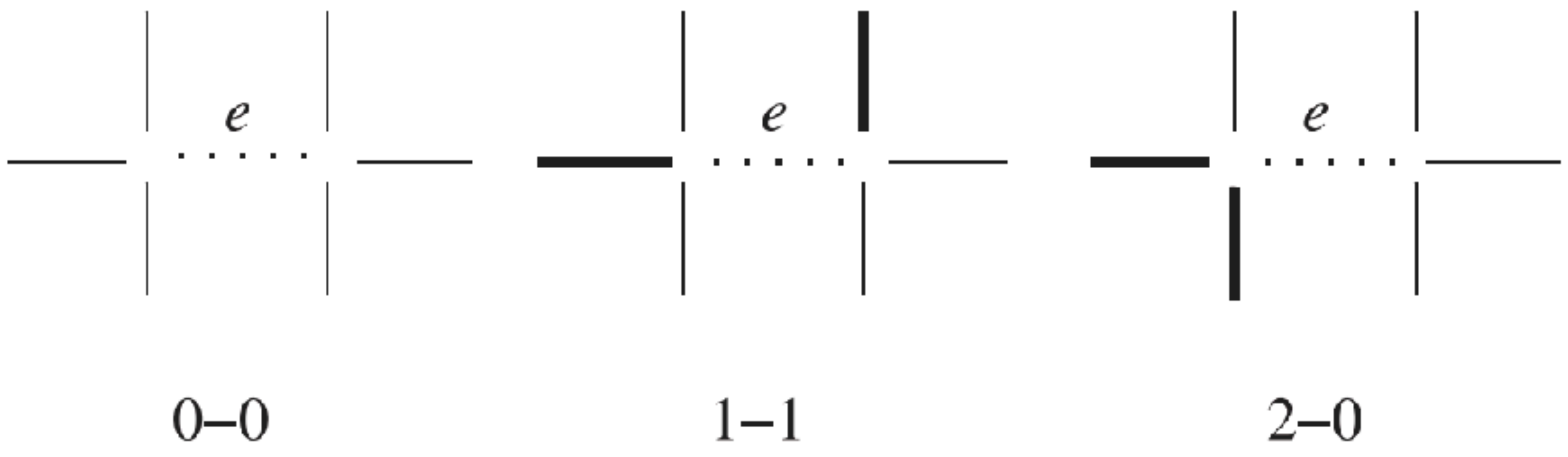


2–0

- **0-0** isolated edge → decrease edge confidence
- **0-1** uncertain → weak increase, or no influence
- **0-2**, **0-3** dead end → decrease edge confidence

slide credit: Václav Hlaváč

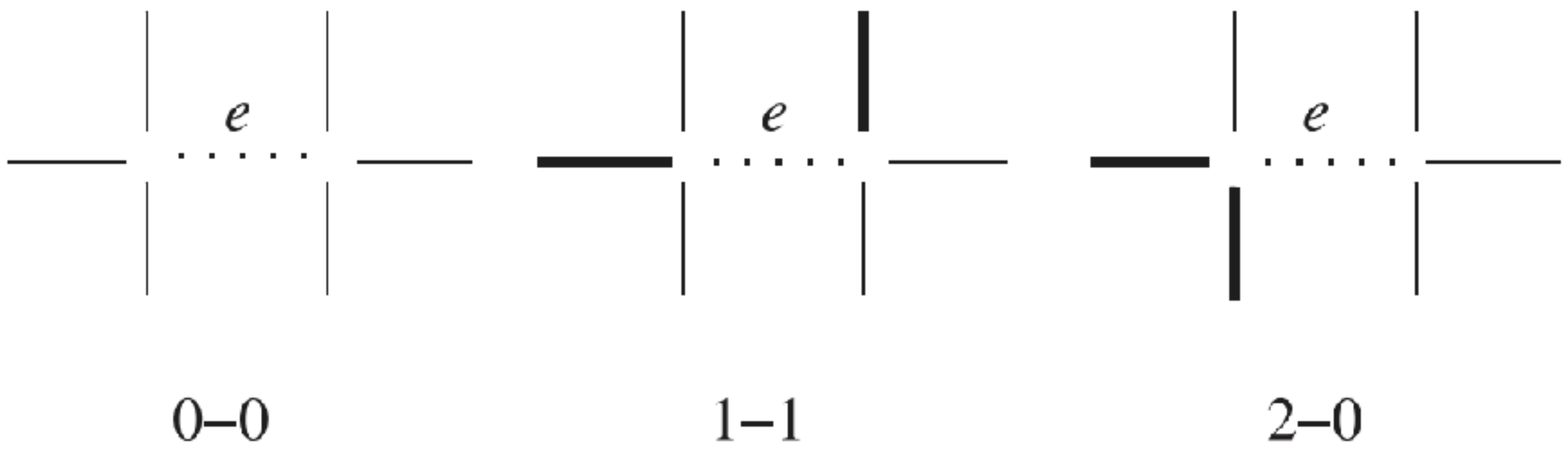
Crack Edges Patterns



- **0-0** isolated edge → decrease edge confidence
- **0-1** uncertain → weak increase, or no influence
- **0-2**, **0-3** dead end → decrease edge confidence
- **1-1** continuation → strong positive influence on edge confidence

slide credit: Václav Hlaváč

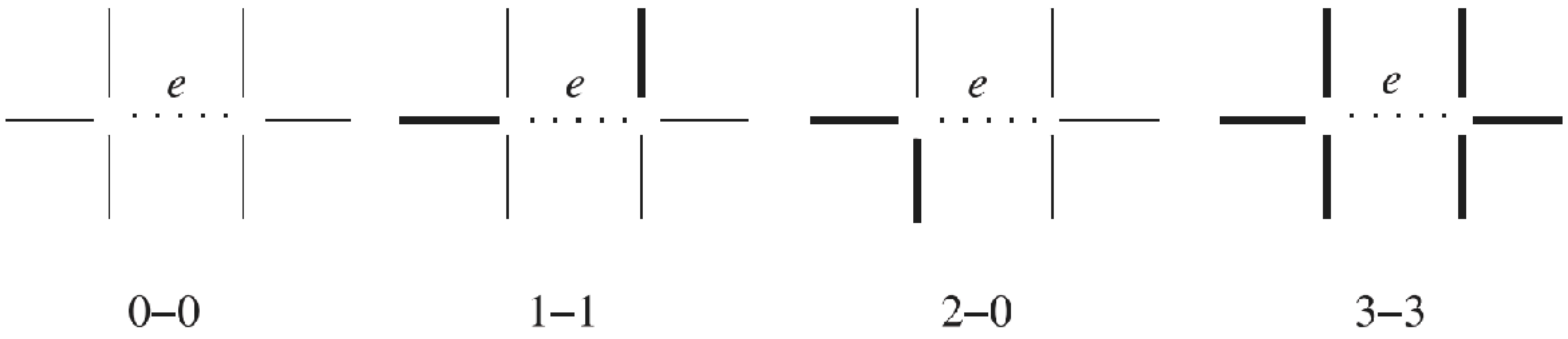
Crack Edges Patterns



- **0-0** isolated edge → decrease edge confidence
- **0-1** uncertain → weak increase, or no influence
- **0-2**, **0-3** dead end → decrease edge confidence
- **1-1** continuation → strong positive influence on edge confidence
- **1-2**, **1-3** continuation to border intersection → medium positive influence on edge confidence

slide credit: Václav Hlaváč

Crack Edges Patterns



- **0-0** isolated edge → decrease edge confidence
- **0-1** uncertain → weak increase, or no influence
- **0-2**, **0-3** dead end → decrease edge confidence
- **1-1** continuation → strong positive influence on edge confidence
- **1-2**, **1-3** continuation to border intersection → medium positive influence on edge confidence
- **2-2**, **2-3**, **3-3** bridge between borders → not necessary for segmentation, no influence on edge confidence

slide credit: Václav Hlaváč

Edge Relaxation

Evaluate confidence $c^1(e)$ for each crack edge e

Set k = 1

Repeat

Determine type of edge based on confidences $c^k(e)$ in neighborhood

Compute confidence $c^{k+1}(e)$ of e based on type and $c^k(e)$

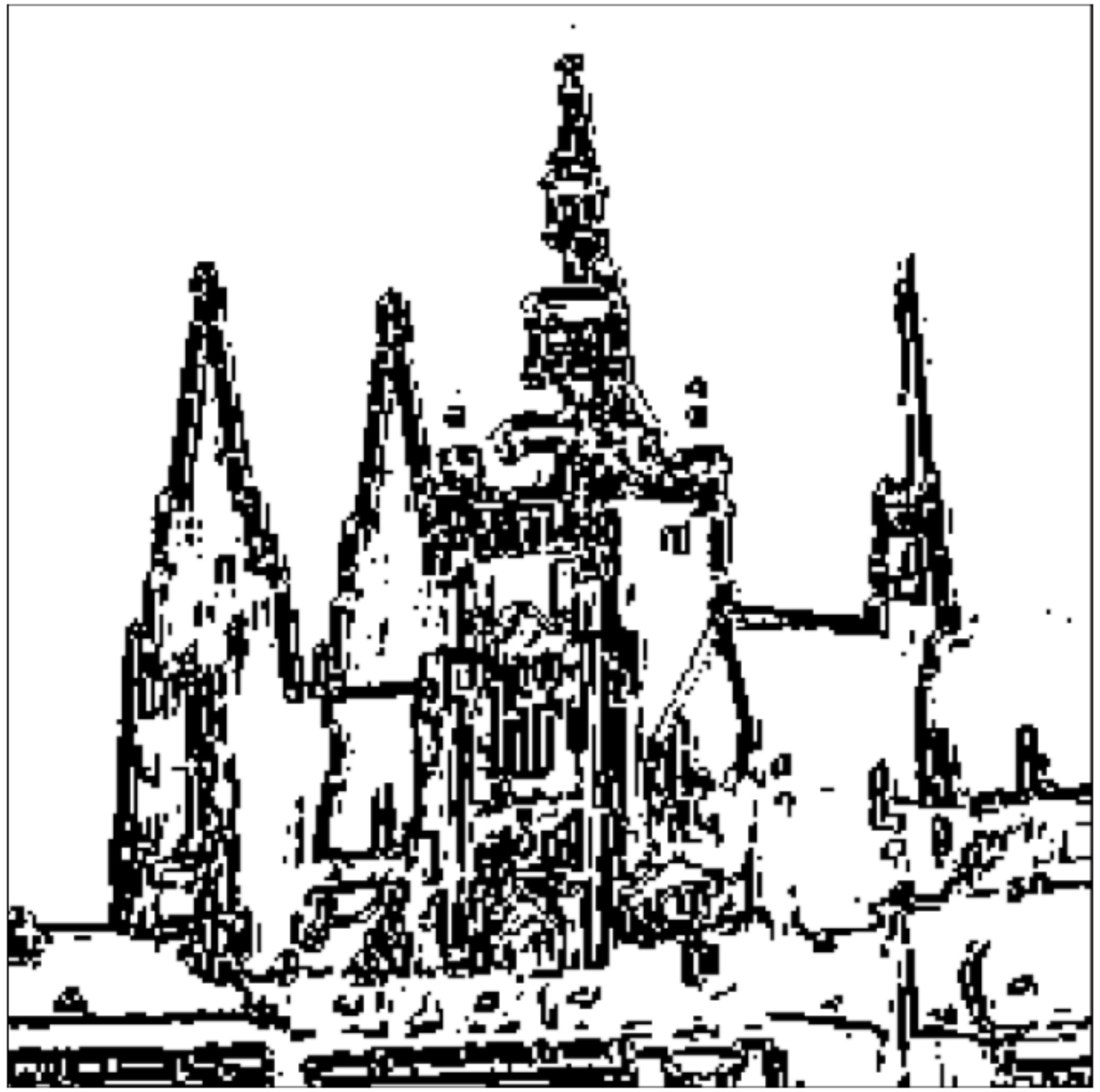
Evaluate model on all 2D-2D and 2D-3D matches

k = k + 1

Stop if all edge confidences have converged to 0 or 1

slide credit: Václav Hlaváč

Edge Relaxation



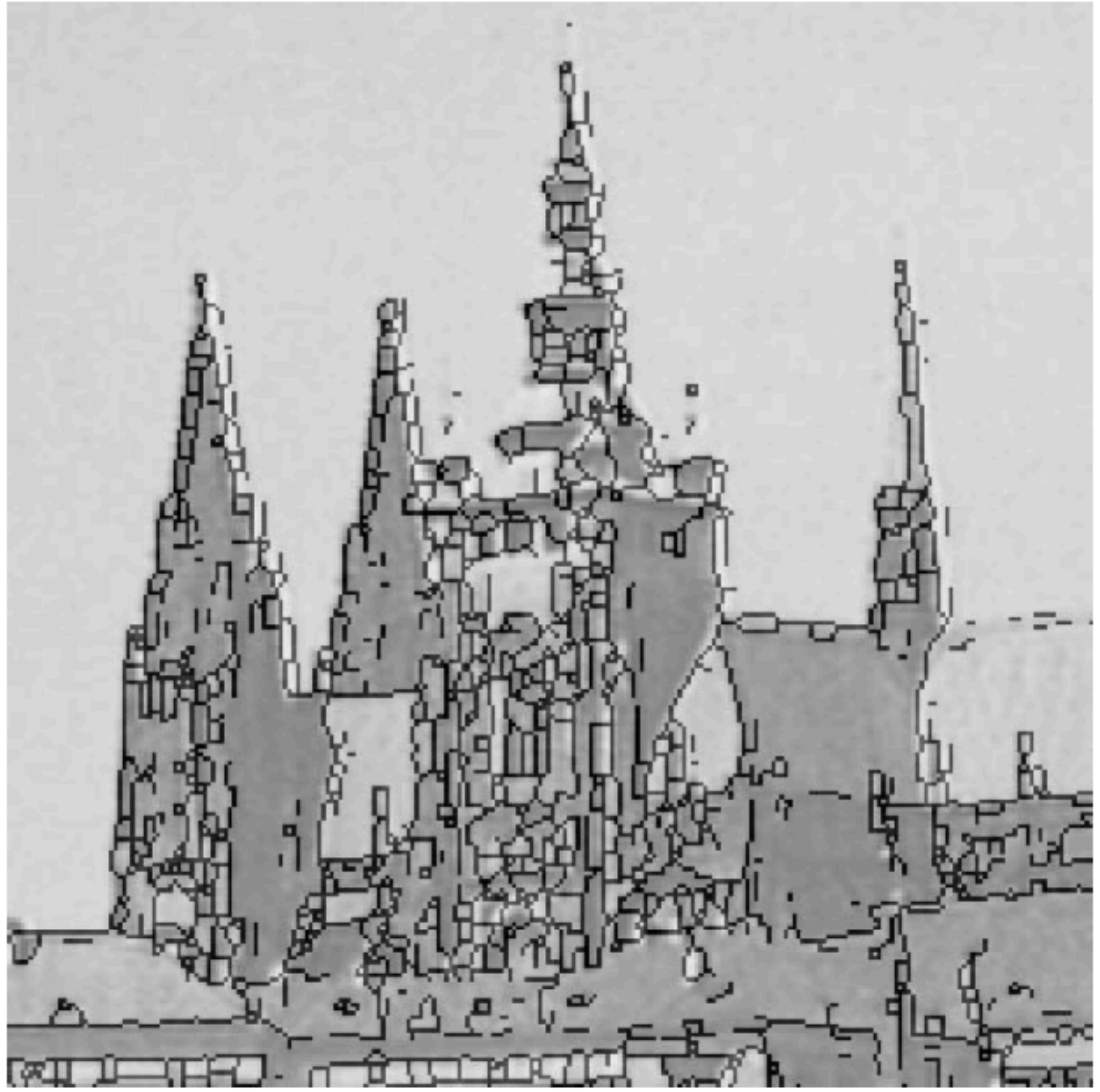
borders after 10 iterations

borders after thinning

slide credit: Václav Hlaváč

Edge Relaxation

borders after 100 iterations
thinned



overlaid over original

slide credit: Václav Hlaváč

Region Growing

Repeat until no more seeds

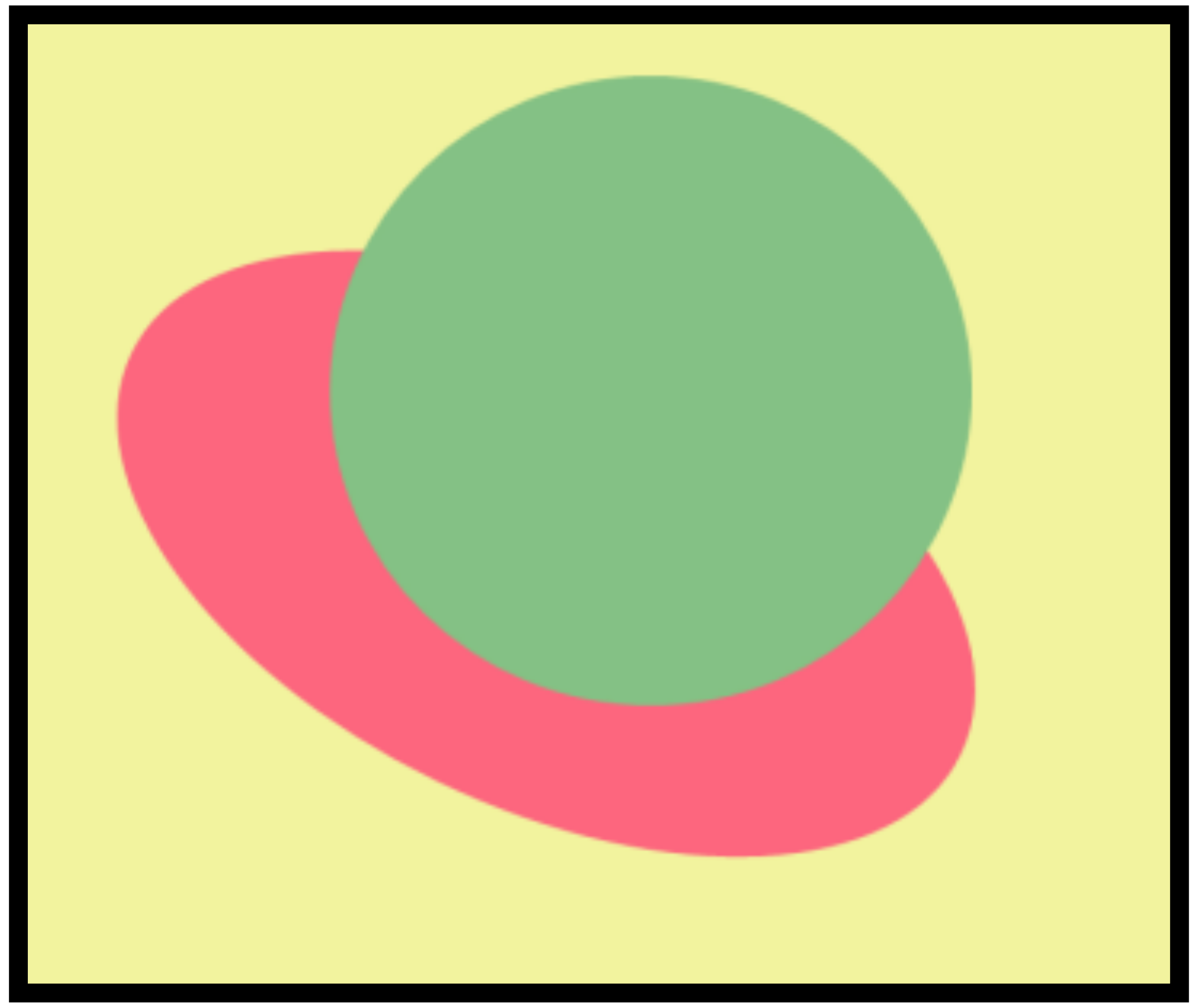
Select set K of seed pixels to start new region R

Repeat

For each pixel $p \in K$:

Add neighbor q to K and R if similar enough to pixels in R

remove p from K



slide credit: Václav Hlaváč

Region Growing

Repeat until no more seeds

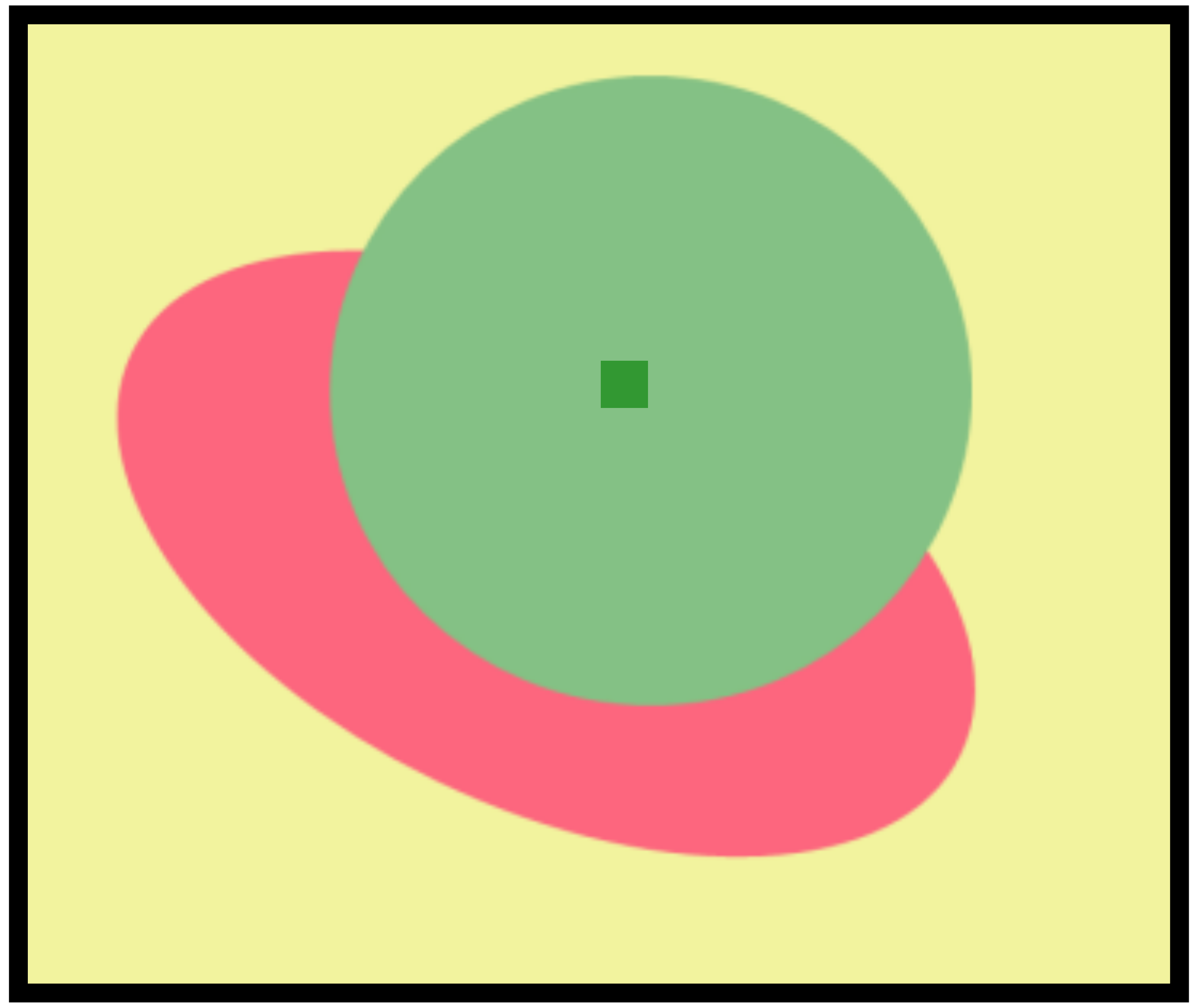
Select set K of seed pixels to start new region R

Repeat

For each pixel $p \in K$:

Add neighbor q to K and R if similar enough to pixels in R

remove p from K



slide credit: Václav Hlaváč

Region Growing

Repeat until no more seeds

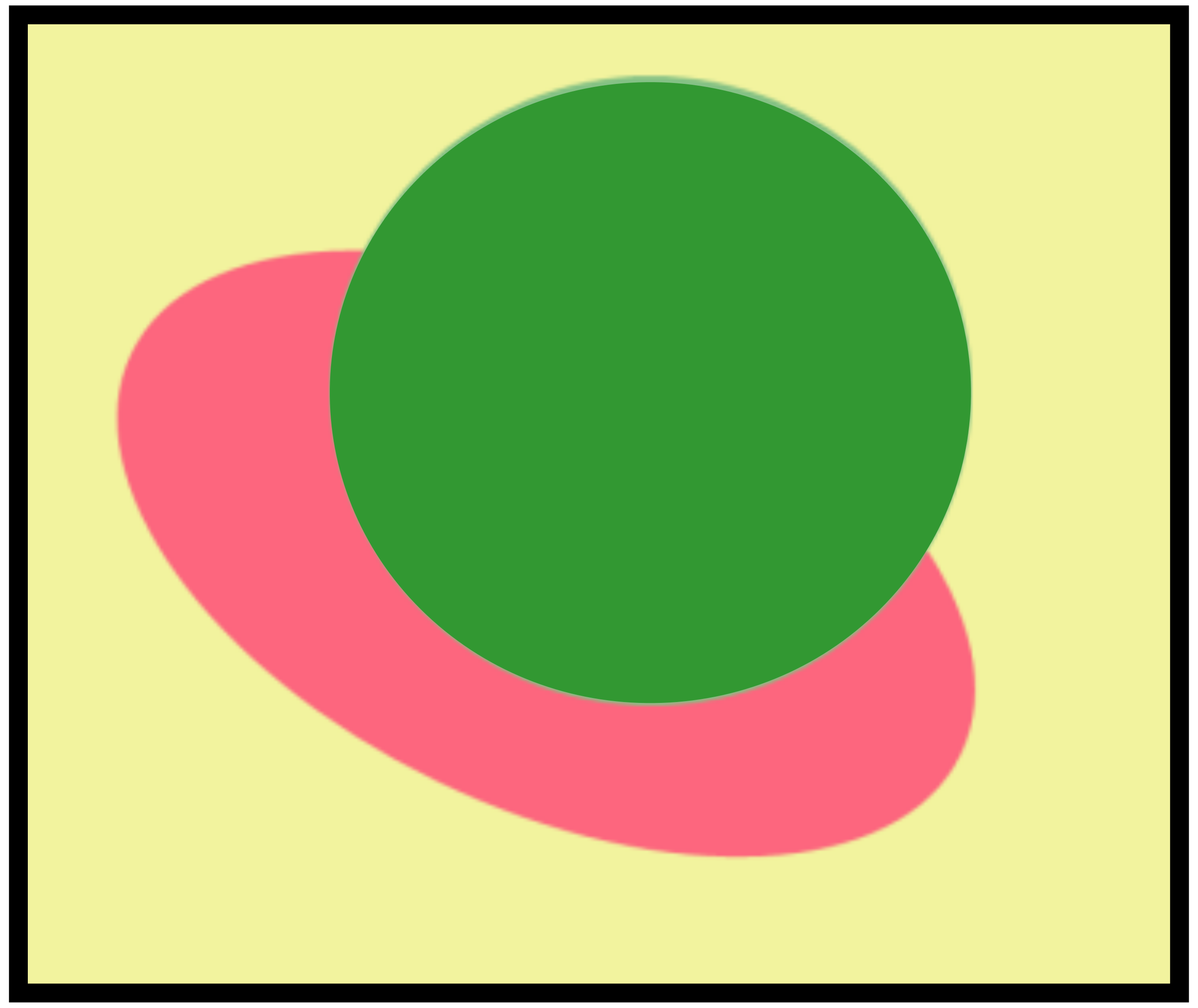
Select set K of seed pixels to start new region R

Repeat

For each pixel $p \in K$:

Add neighbor q to K and R if similar enough to pixels in R

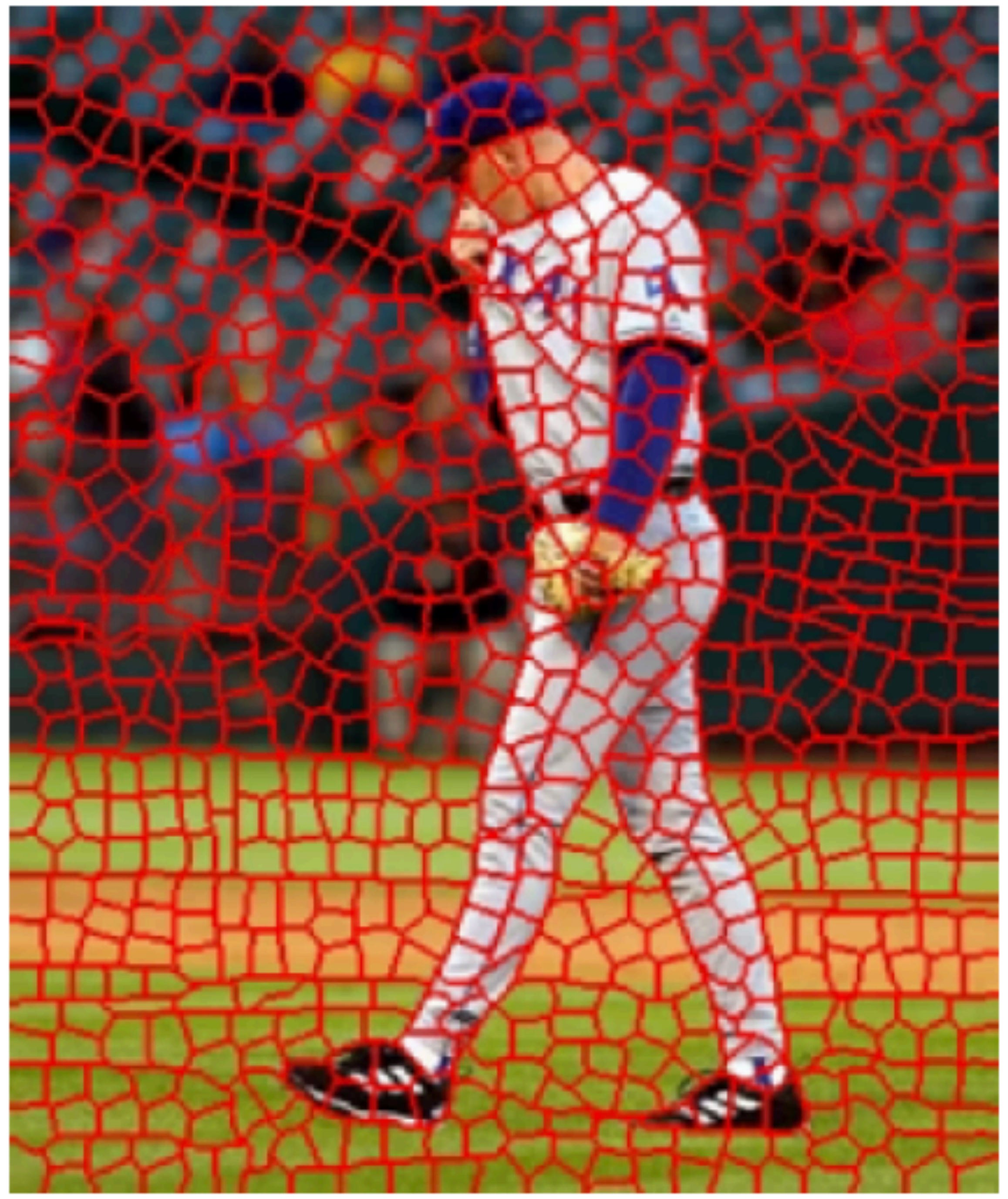
remove p from K



slide credit: Václav Hlaváč

Superpixels

- Group together similar-looking pixels
- Increases efficiency of further processing: use superpixels rather than pixels



[Ren & Malik, Learning a classification model for segmentation, ICCV 2003]

slide credit: Václav Hlaváč

Discussion

- Edge-based segmentation and region growing are examples for **bottom-up approaches** to segmentation:
  - Obtain initial segmentation, then process it (merge close-by segments to semantically meaningful regions, etc.)
  - Risk: early mistakes cannot be recovered (premature hard decisions)

slide credit: Václav Hlaváč

Discussion

- Edge-based segmentation and region growing are examples for **bottom-up approaches** to segmentation:
  - Obtain initial segmentation, then process it (merge close-by segments to semantically meaningful regions, etc.)
  - Risk: early mistakes cannot be recovered (premature hard decisions)
- **Top-down segmentation**: start with larger regions and split them into semantically meaningful parts

slide credit: Václav Hlaváč

Discussion

- Edge-based segmentation and region growing are examples for **bottom-up approaches** to segmentation:
  - Obtain initial segmentation, then process it (merge close-by segments to semantically meaningful regions, etc.)
  - Risk: early mistakes cannot be recovered (premature hard decisions)
- **Top-down segmentation**: start with larger regions and split them into semantically meaningful parts
- Bottom-up and top-down approaches can work together, are not mutually exclusive

slide credit: Václav Hlaváč

Lecture Overview

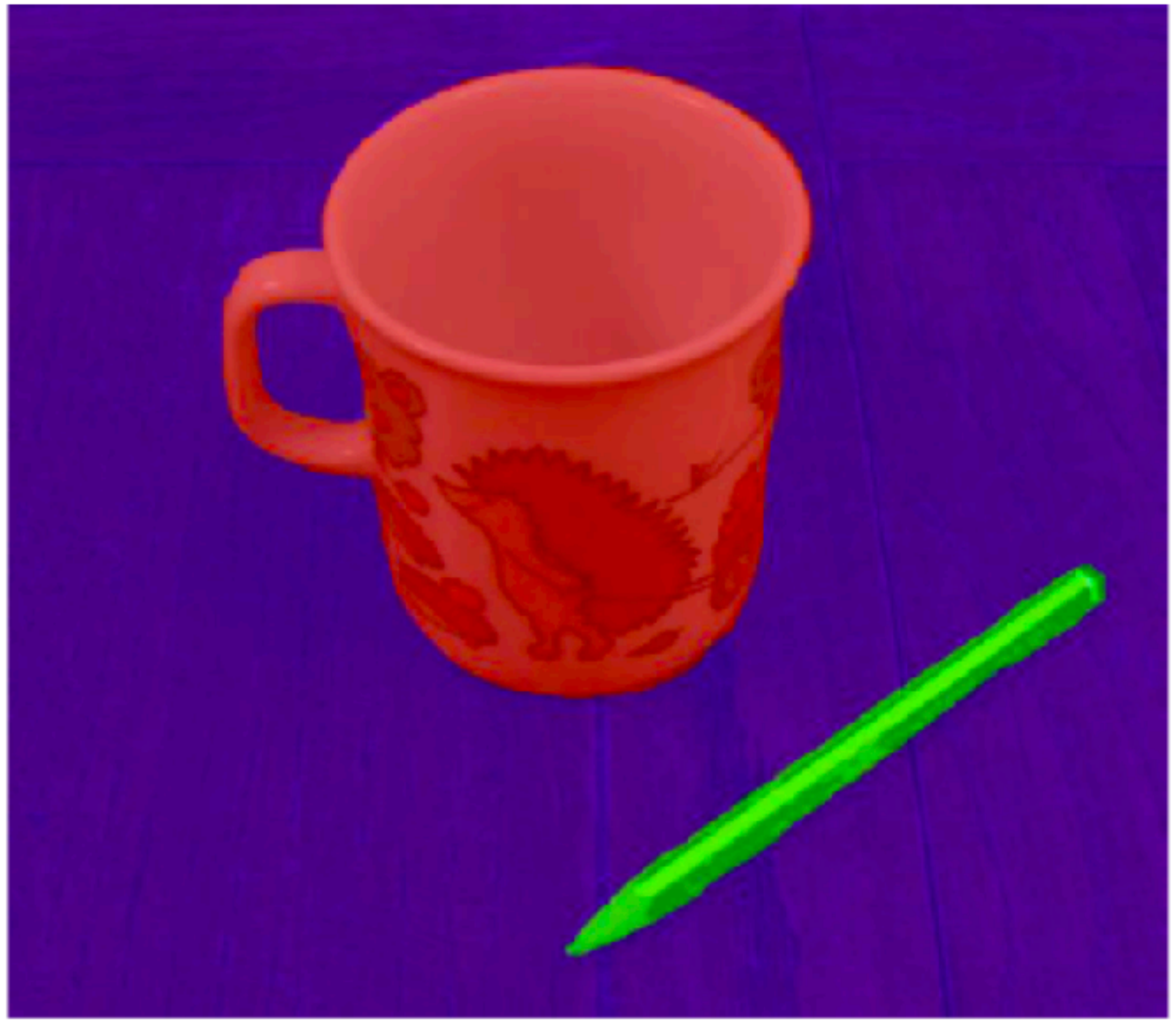
simple & heuristic

- A simple approach to segmentation: **(intensity) thresholding**
- Segmentation based on spatial coherence: **edge-based segmentation**, **region growing**
- Segmentation as a **clustering** problem: **k-means clustering**, **mean-shift clustering**
- Segmentation as a statistical (unsupervised) learning problem: **expectation maximization (EM) algorithm**
- Next lecture: **graph-based segmentation**, **supervised learning with neural networks** (if time and interest)

complex & principled

slide credit: Václav Hlaváč

Segmentation As Clustering



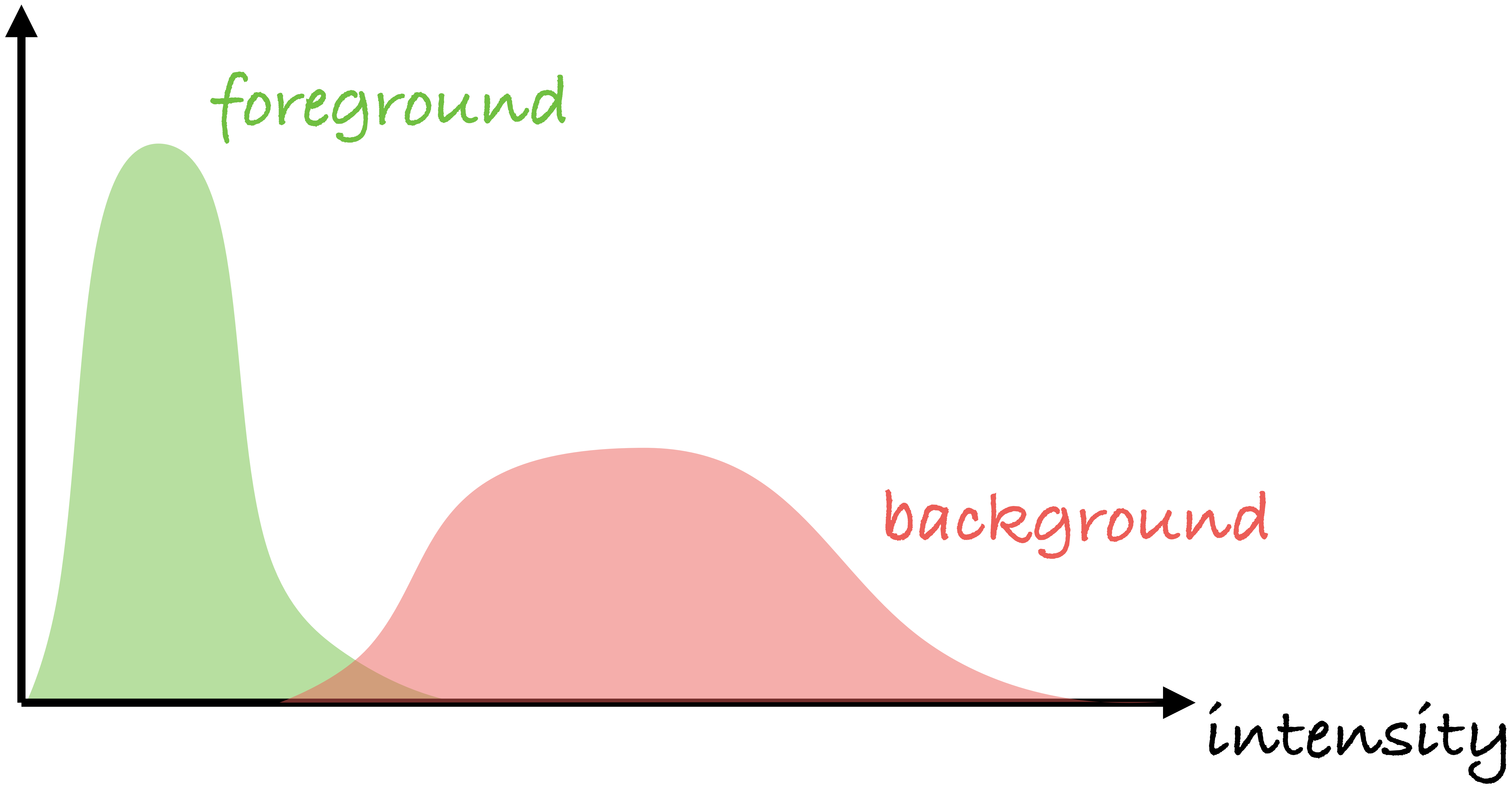
Image, courtesy Ondřej Drbohlav

Goal: **cluster pixels** into **regions** based on spatial and appearance similarity

slide credit: Václav Hlaváč

Thresholding As Clustering

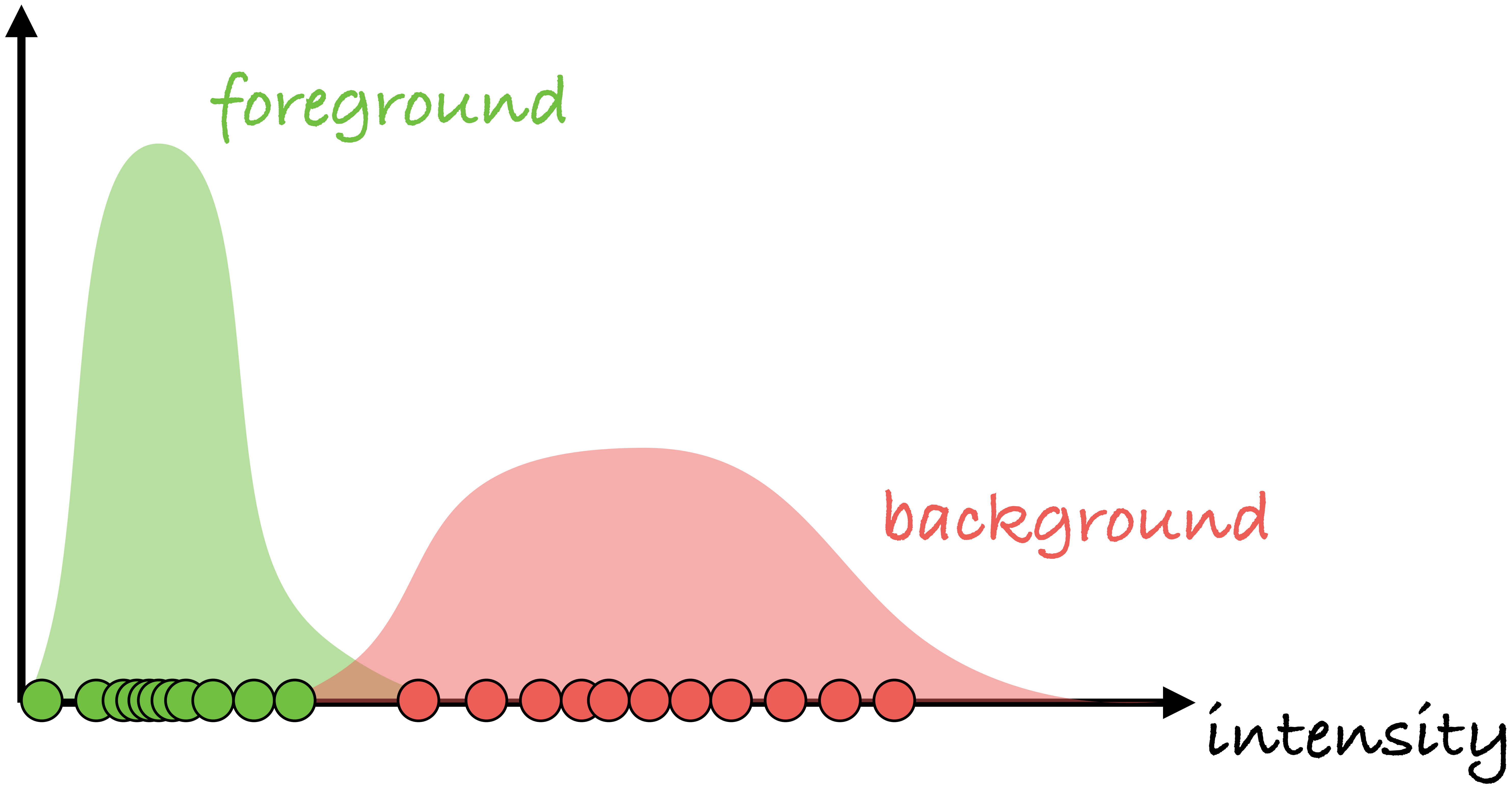
Goal: find **clusters of pixels that belong together** based on pixel intensities



slide credit: Bastian Leibe

Thresholding As Clustering

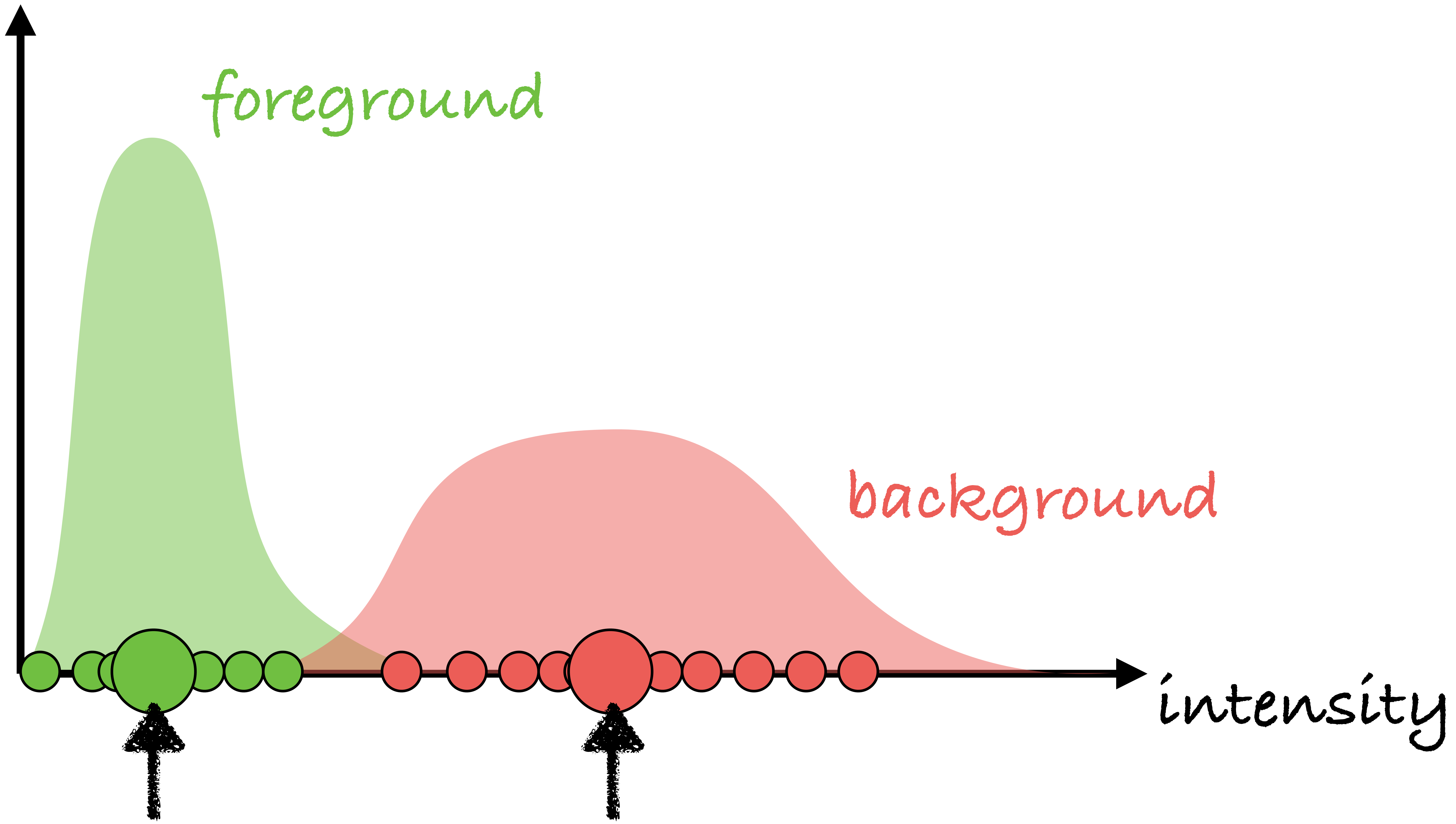
Goal: find **clusters of pixels that belong together** based on pixel intensities



slide credit: Bastian Leibe

Thresholding As Clustering

Goal: find **clusters of pixels that belong together** based on pixel intensities



Represent clusters by “centers” assign pixel to cluster of nearest center

slide credit: Bastian Leibe

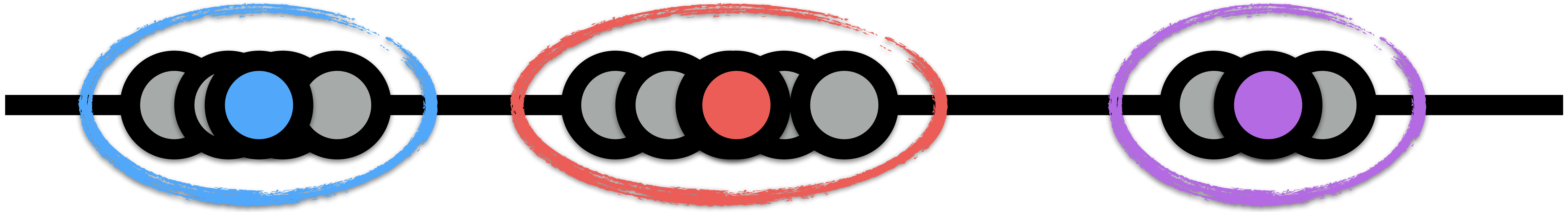
k-Means Clustering

- **Chicken-and-egg problem**:
  - Given cluster membership, we can compute cluster centers (e.g., as mean)

slide credit: Bastian Leibe

k-Means Clustering

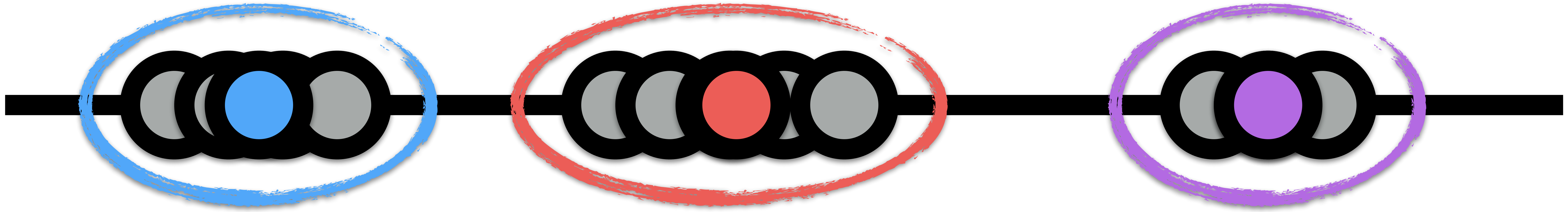
- **Chicken-and-egg problem**:
  - Given cluster membership, we can compute cluster centers (e.g., as mean)



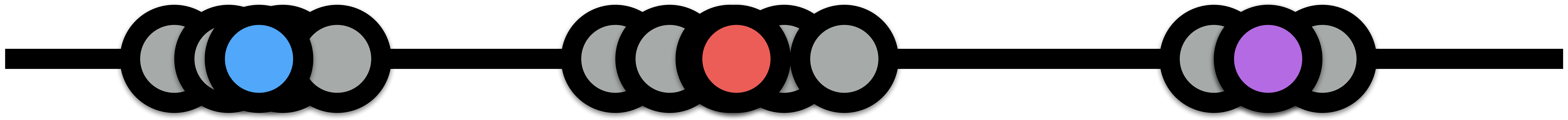
slide credit: Bastian Leibe

k-Means Clustering

- **Chicken-and-egg problem**:
  - Given cluster membership, we can compute cluster centers (e.g., as mean)



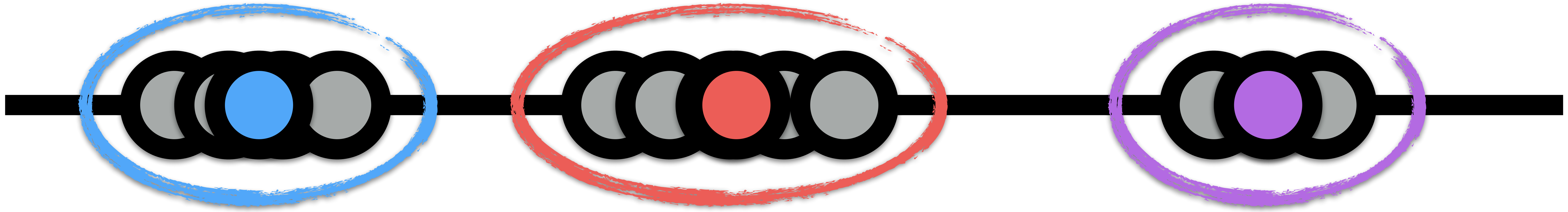
  - Given cluster centers, we can compute cluster membership (find nearest cluster center)



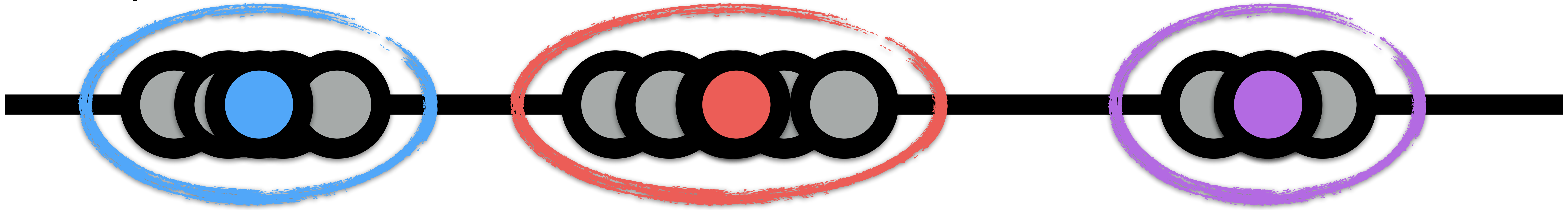
slide credit: Bastian Leibe

k-Means Clustering

- **Chicken-and-egg problem**:
  - Given cluster membership, we can compute cluster centers (e.g., as mean)



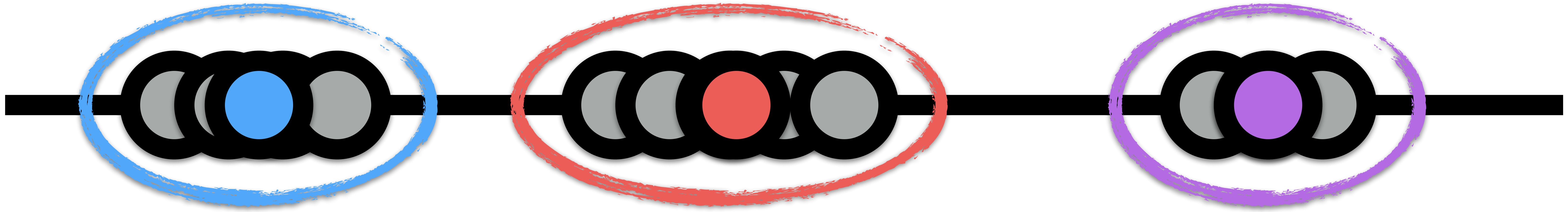
  - Given cluster centers, we can compute cluster membership (find nearest cluster center)



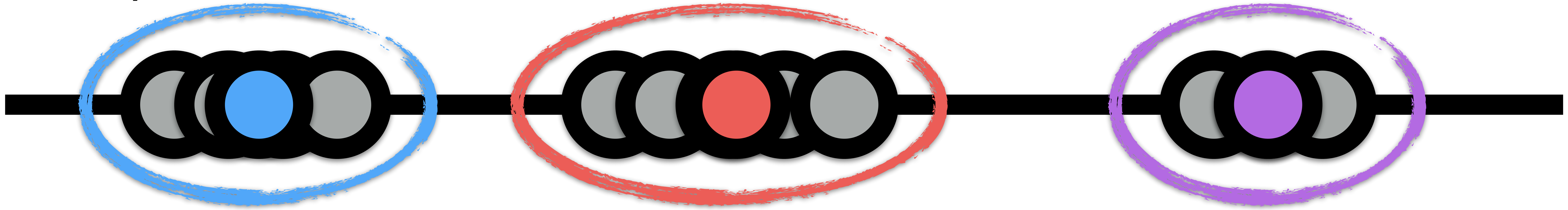
slide credit: Bastian Leibe

k-Means Clustering

- **Chicken-and-egg problem**:
  - Given cluster membership, we can compute cluster centers (e.g., as mean)



  - Given cluster centers, we can compute cluster membership (find nearest cluster center)



- **k-means clustering**: alternate between computing cluster membership and computing cluster centers

slide credit: Bastian Leibe

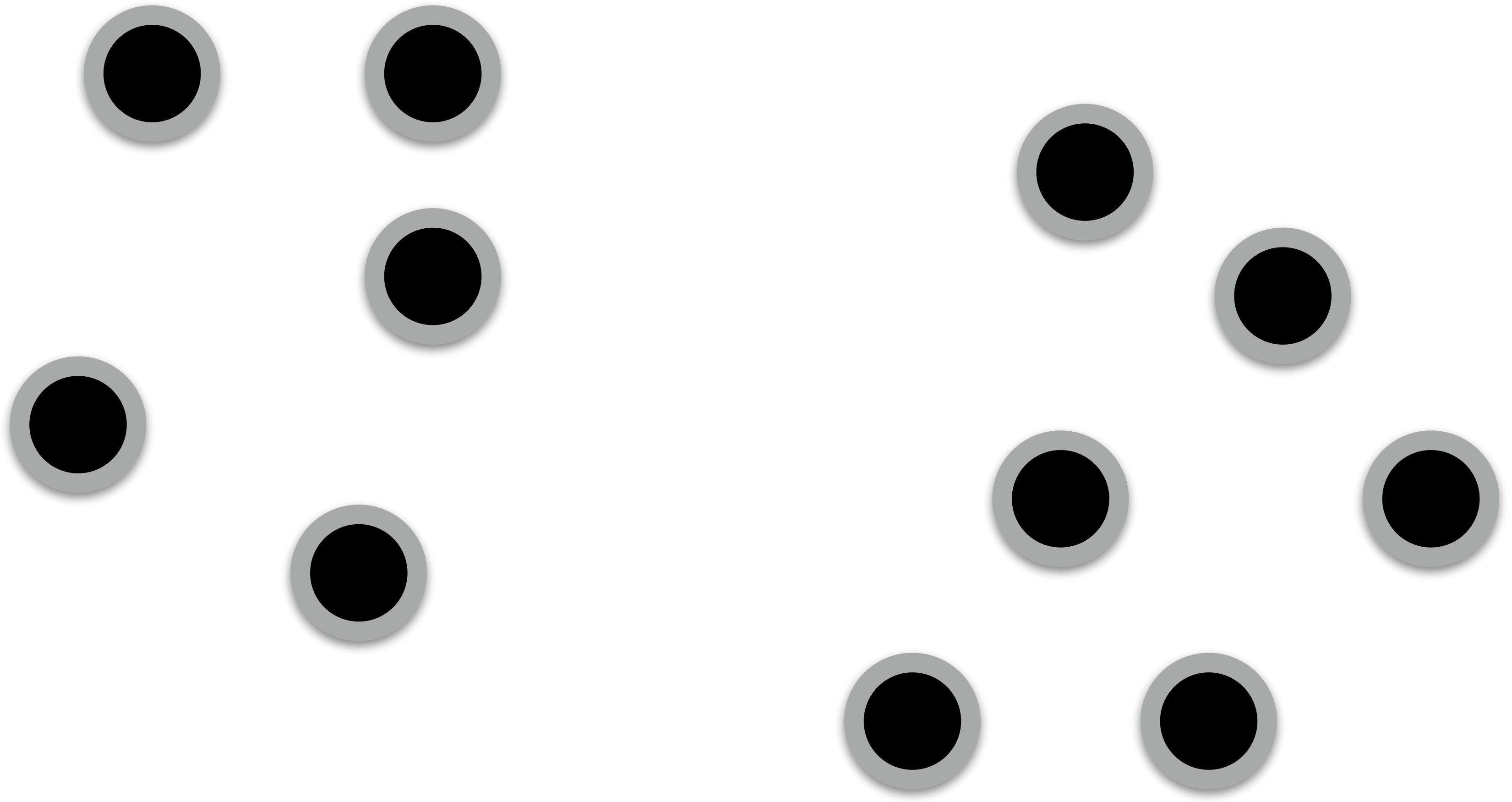
k-Means Clustering / Lloyd's Algorithm

Randomly initialize k cluster centers

Repeat until assignments / centers do not change:

Assign each point to the closest center

Recompute centers as mean of all points assigned to centers



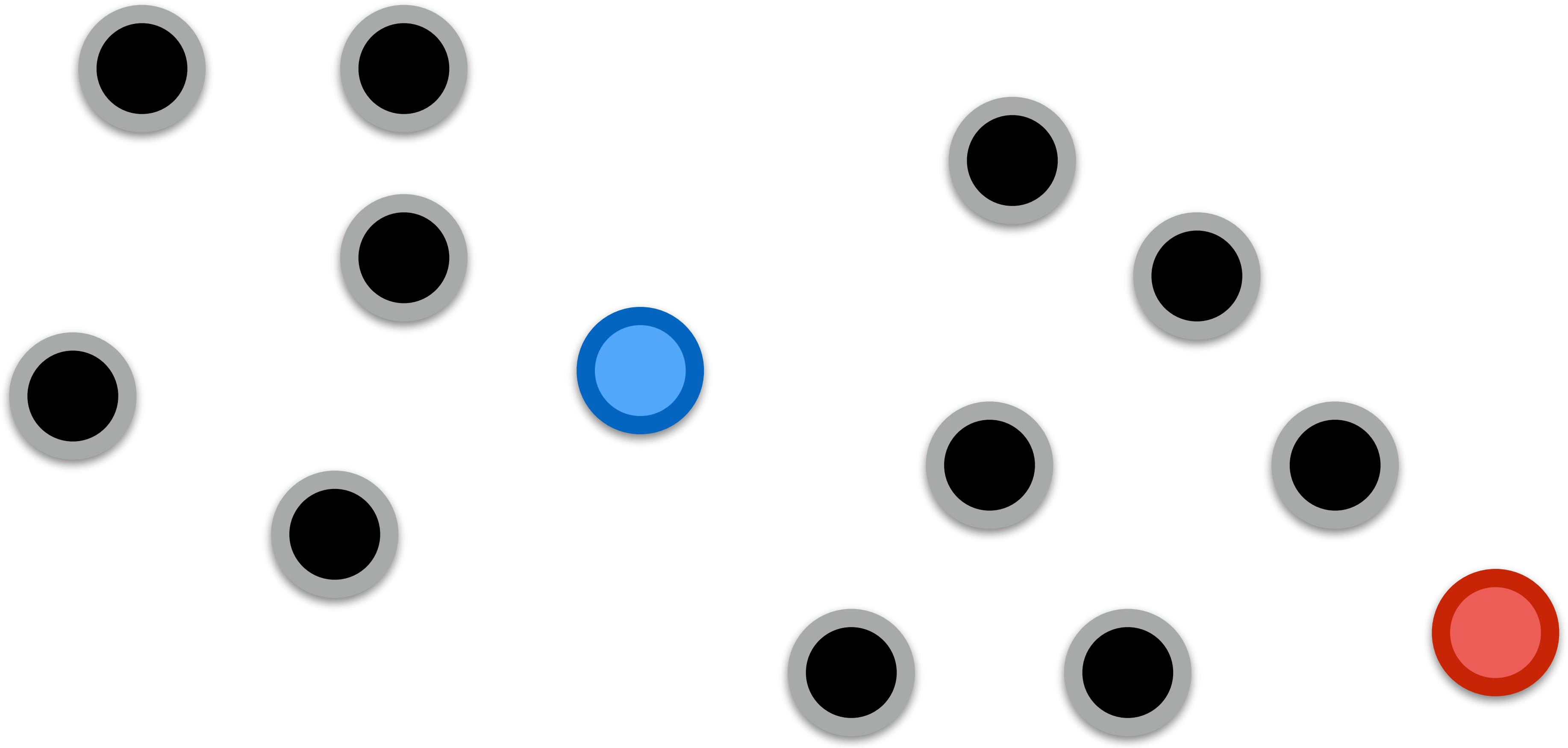
[Lloyd, Least square quantization in PCM's, Bell Telephone Laboratories Paper 1957]
[Lloyd, Least squares quantization in PCM, Spec. issue on quantiz., IEEE Trans. Inform. Theory, 28:129–137, 1982]

k-Means Clustering / Lloyd's Algorithm

Randomly initialize k cluster centers

Repeat until assignments / centers do not change:

- Assign each point to the closest center
- Recompute centers as mean of all points assigned to centers



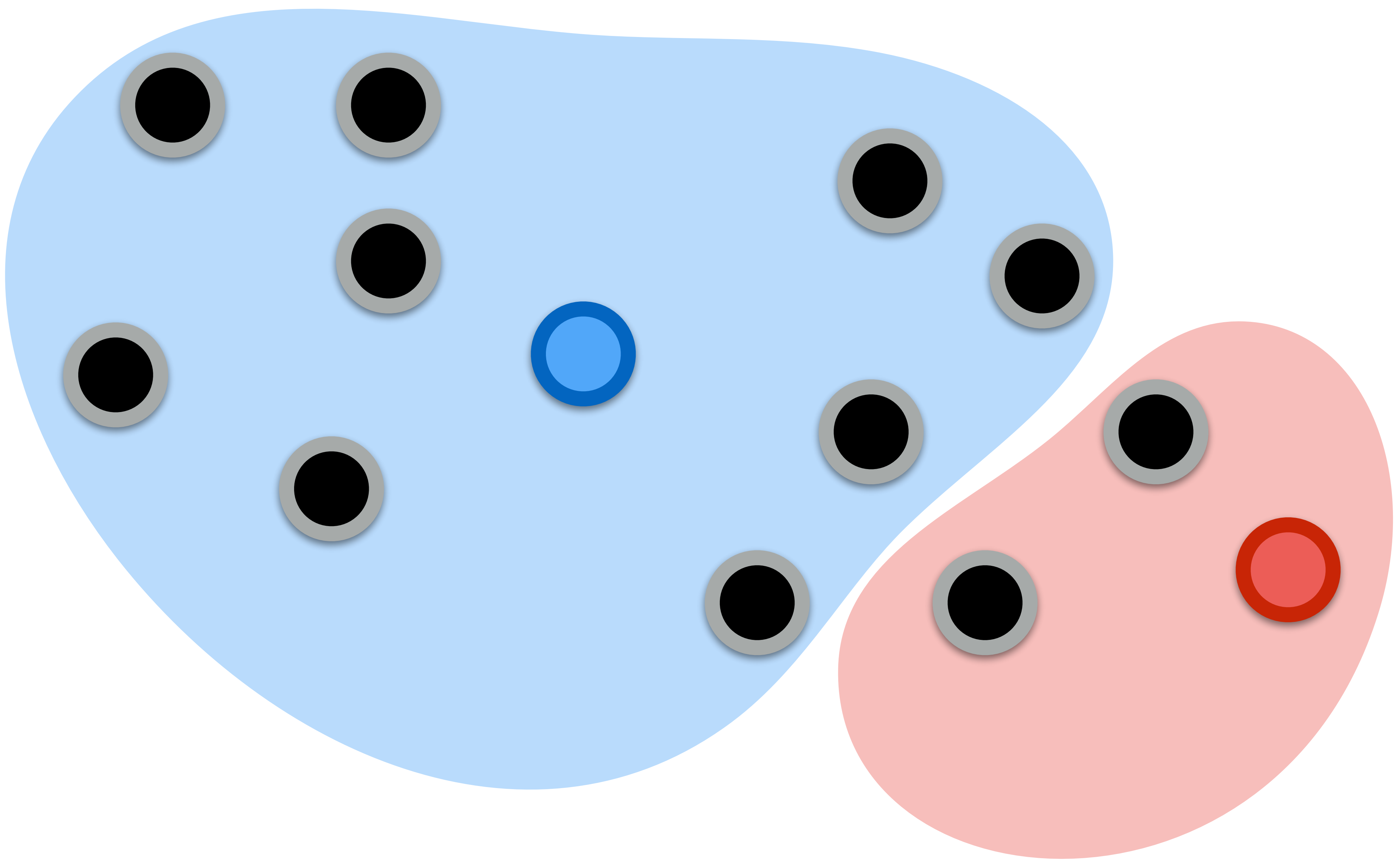
[Lloyd, Least square quantization in PCM's, Bell Telephone Laboratories Paper 1957]
[Lloyd, Least squares quantization in PCM, Spec. issue on quantiz., IEEE Trans. Inform. Theory, 28:129–137, 1982]

k-Means Clustering / Lloyd's Algorithm

Randomly initialize k cluster centers

Repeat until assignments / centers do not change:

- Assign each point to the closest center
- Recompute centers as mean of all points assigned to centers



[Lloyd, Least square quantization in PCM's, Bell Telephone Laboratories Paper 1957]
[Lloyd, Least squares quantization in PCM, Spec. issue on quantiz., IEEE Trans. Inform. Theory, 28:129–137, 1982]

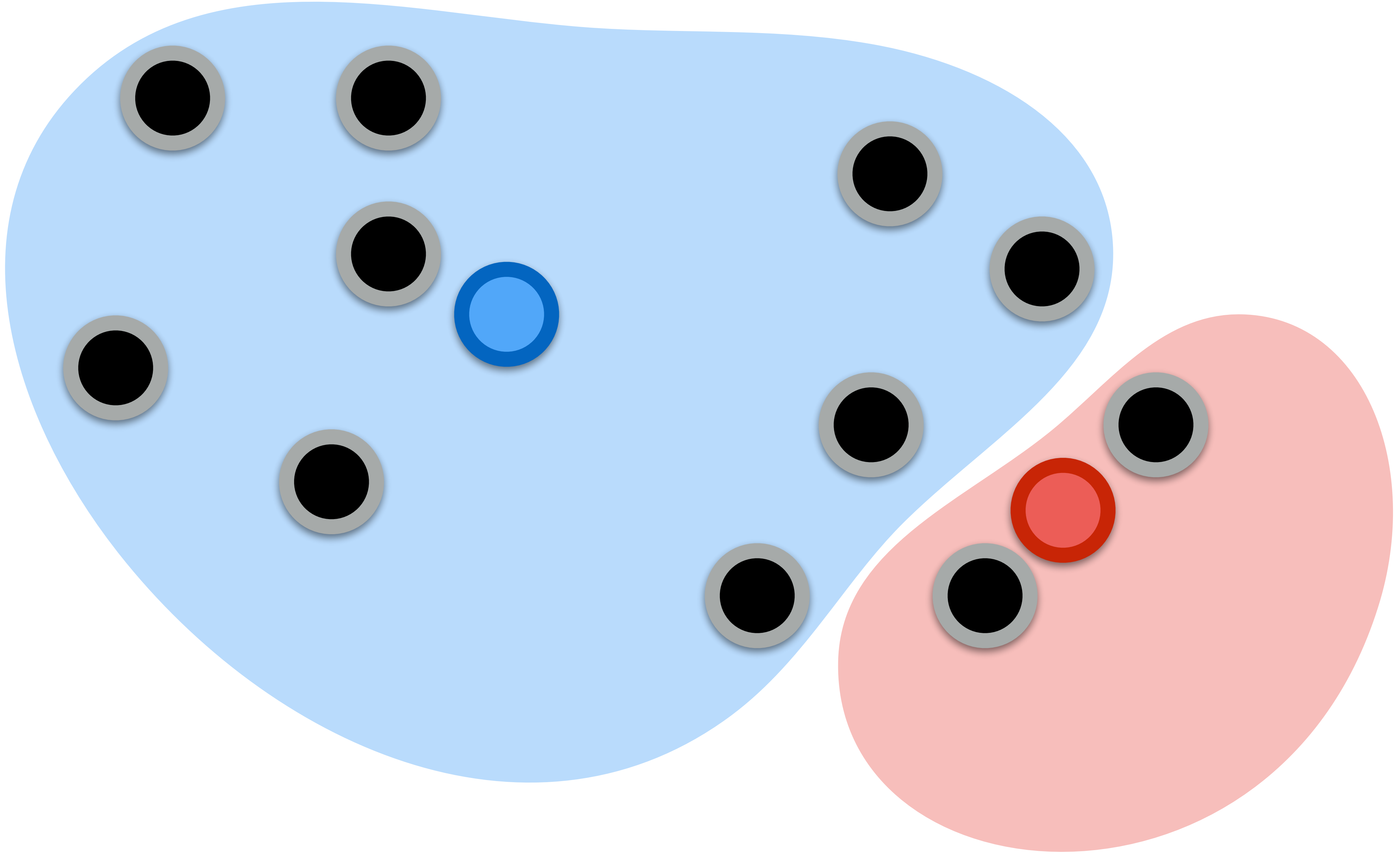
k-Means Clustering / Lloyd's Algorithm

Randomly initialize k cluster centers

Repeat until assignments / centers do not change:

Assign each point to the closest center

Recompute centers as mean of all points assigned to centers



[Lloyd, Least square quantization in PCM's, Bell Telephone Laboratories Paper 1957]
[Lloyd, Least squares quantization in PCM, Spec. issue on quantiz., IEEE Trans. Inform. Theory, 28:129–137, 1982]

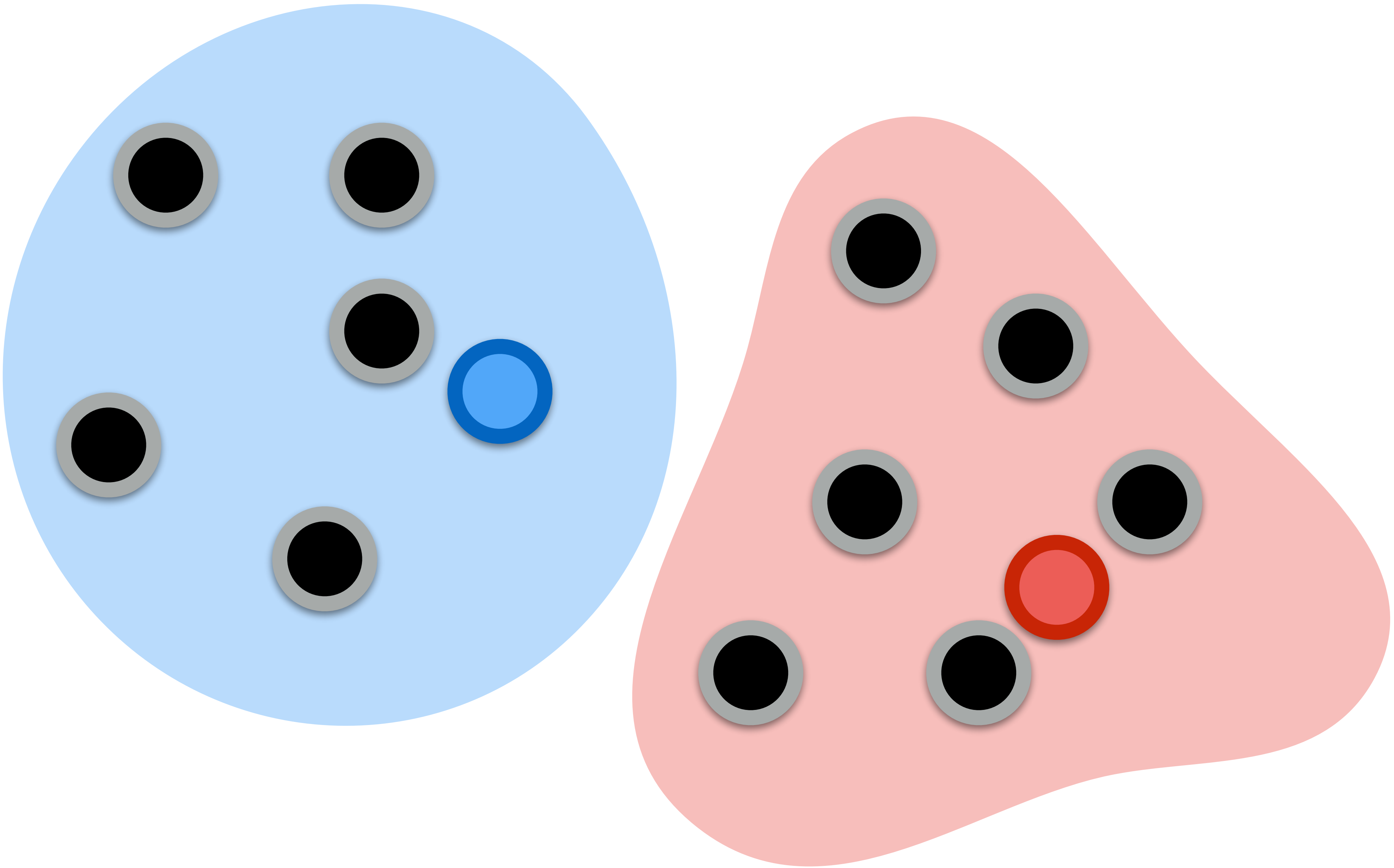
k-Means Clustering / Lloyd's Algorithm

Randomly initialize k cluster centers

Repeat until assignments / centers do not change:

Assign each point to the closest center

Recompute centers as mean of all points assigned to centers



[Lloyd, Least square quantization in PCM's, Bell Telephone Laboratories Paper 1957]
[Lloyd, Least squares quantization in PCM, Spec. issue on quantiz., IEEE Trans. Inform. Theory, 28:129–137, 1982]

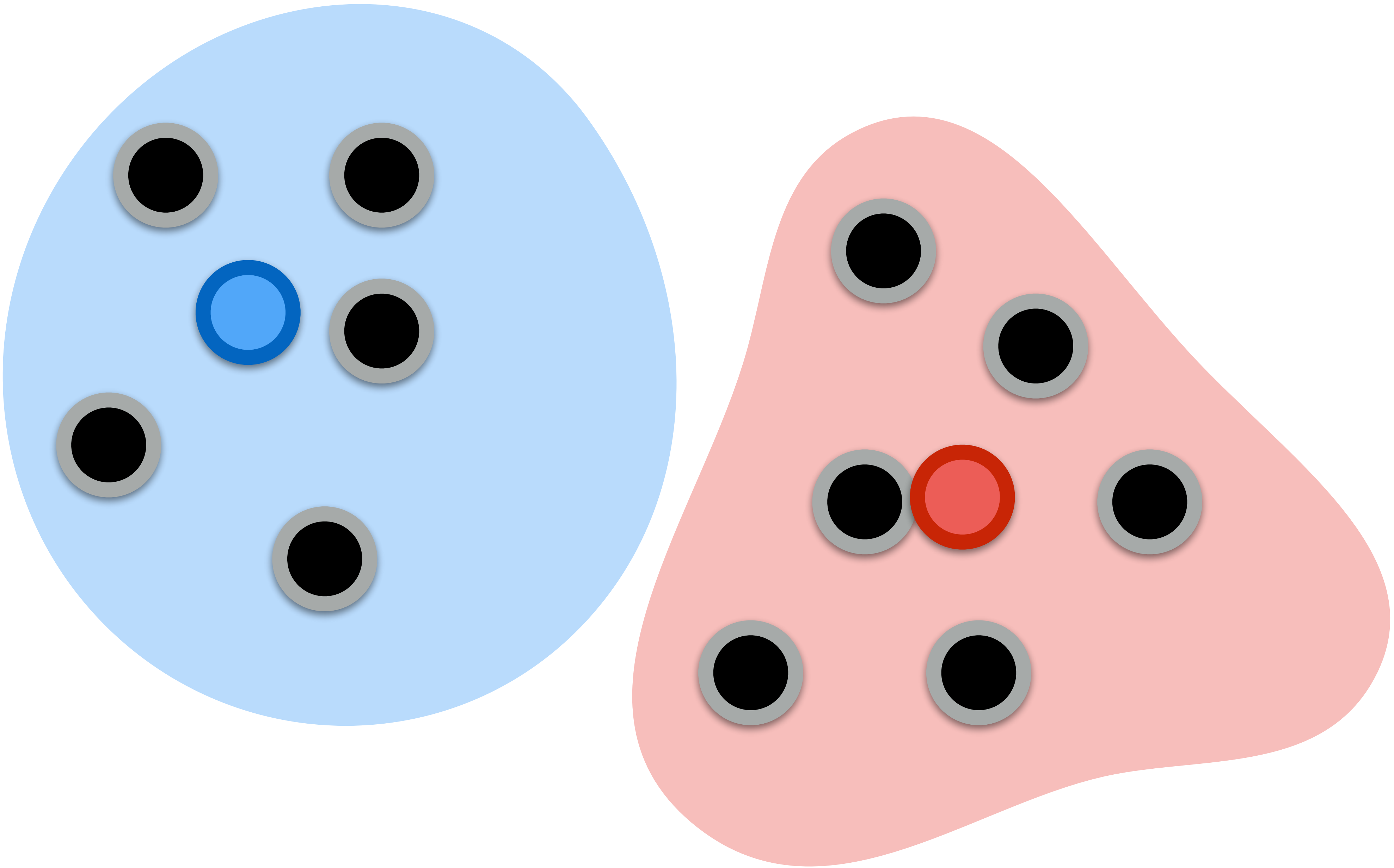
k-Means Clustering / Lloyd's Algorithm

Randomly initialize k cluster centers

Repeat until assignments / centers do not change:

Assign each point to the closest center

Recompute centers as mean of all points assigned to centers



[Lloyd, Least square quantization in PCM's, Bell Telephone Laboratories Paper 1957]
[Lloyd, Least squares quantization in PCM, Spec. issue on quantiz., IEEE Trans. Inform. Theory, 28:129–137, 1982]

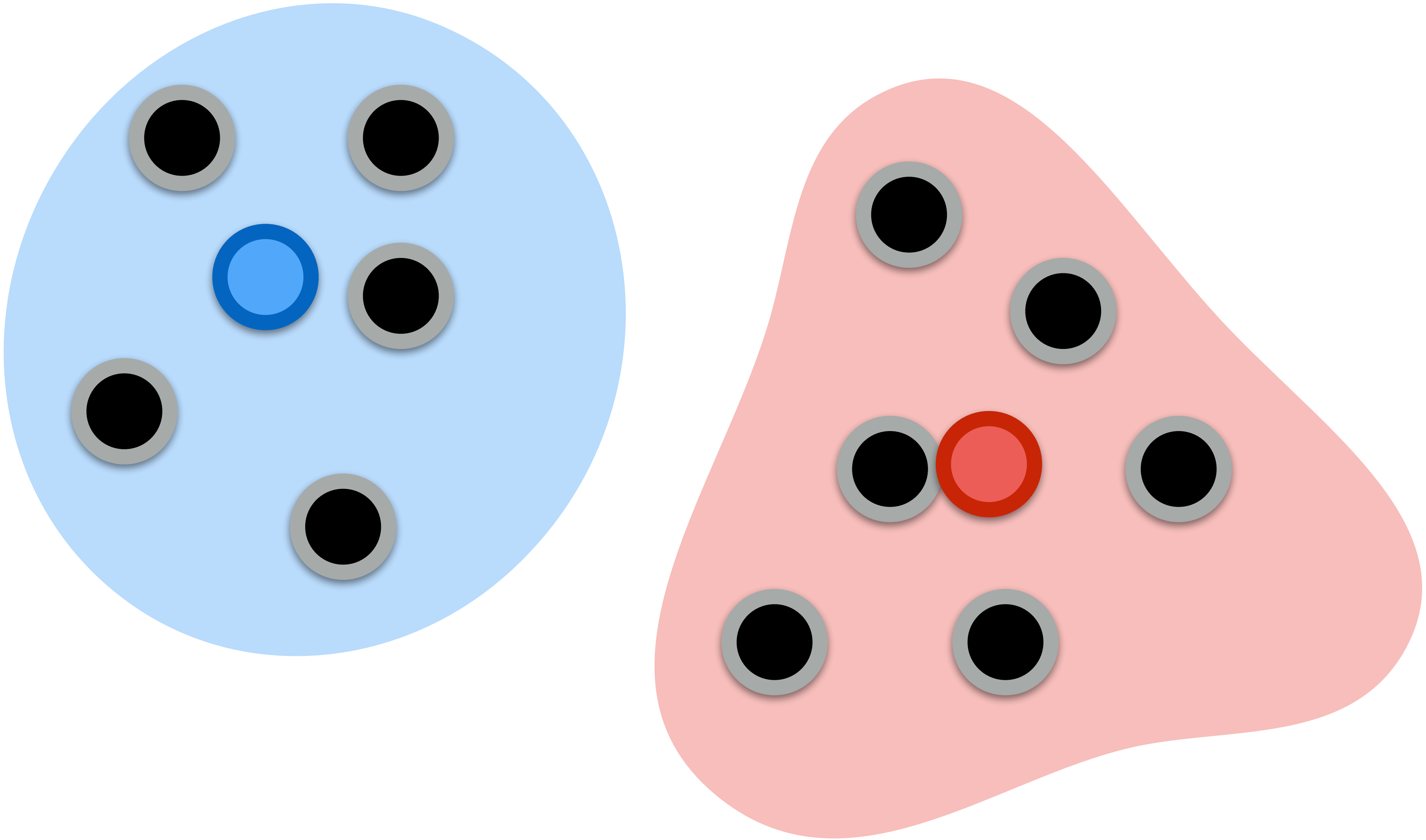
k-Means Clustering / Lloyd's Algorithm

Randomly initialize k cluster centers

Repeat until assignments / centers do not change:

  Assign each point to the closest center

  Recompute centers as mean of all points assigned to centers



[Lloyd, Least square quantization in PCM's, Bell Telephone Laboratories Paper 1957]
[Lloyd, Least squares quantization in PCM, Spec. issue on quantiz., IEEE Trans. Inform. Theory, 28:129–137, 1982]

k-Means Clustering / Lloyd's Algorithm

slide credit: Václav Hlaváč, Bastian Leibe

k-Means Clustering / Lloyd's Algorithm

- Does k-means clustering converge?

slide credit: Václav Hlaváč, Bastian Leibe

k-Means Clustering / Lloyd's Algorithm

- Does k-means clustering converge?
- k-means cluster optimizes the cost:

$$\sum_{\text{clusters } i} \; \sum_{\text{points } p \text{ in cluster } i} \|p - c_i\|^2$$

slide credit: Václav Hlaváč, Bastian Leibe

k-Means Clustering / Lloyd's Algorithm

- Does k-means clustering converge?
- k-means cluster optimizes the cost:

$$\sum_{\text{clusters } i} \; \sum_{\text{points } p \text{ in cluster } i} \|p - c_i\|^2$$

  - Updating cluster centers via the mean reduces cost for same assignments

slide credit: Václav Hlaváč, Bastian Leibe

k-Means Clustering / Lloyd's Algorithm

- Does k-means clustering converge?
- k-means cluster optimizes the cost:

$$\sum_{\text{clusters } i} \; \sum_{\text{points } p \text{ in cluster } i} \|p - c_i\|^2$$

  - Updating cluster centers via the mean reduces cost for same assignments
  - Updating an assignment reduces cost as well

slide credit: Václav Hlaváč, Bastian Leibe

k-Means Clustering / Lloyd's Algorithm

- Does k-means clustering converge?
- k-means cluster optimizes the cost:

$$\sum_{\text{clusters } i} \quad \sum_{\text{points } p \text{ in cluster } i} \|p - c_i\|^2$$

  - Updating cluster centers via the mean reduces cost for same assignments
  - Updating an assignment reduces cost as well
  - Will thus converge to some **local minimum**

slide credit: Václav Hlaváč, Bastian Leibe

k-Means Clustering / Lloyd's Algorithm

- Does k-means clustering converge?
- k-means cluster optimizes the cost:

$$\sum_{\text{clusters } i} \sum_{\text{points } p \text{ in cluster } i} \|p - c_i\|^2$$

  - Updating cluster centers via the mean reduces cost for same assignments
  - Updating an assignment reduces cost as well
  - Will thus converge to some **local minimum**
  - In practice: stop algorithm if cost change is too small

slide credit: Václav Hlaváč, Bastian Leibe

k-Means Clustering / Lloyd's Algorithm

- Does k-means clustering converge?
- k-means cluster optimizes the cost:

$$\sum_{\text{clusters } i} \sum_{\text{points } p \text{ in cluster } i} \|p - c_i\|^2$$

  - Updating cluster centers via the mean reduces cost for same assignments
  - Updating an assignment reduces cost as well
  - Will thus converge to some **local minimum**
  - In practice: stop algorithm if cost change is too small
- Convergence can be superpolynomial $\mathcal{O}\left(2^{\sqrt{2}}\right)$ [Arthur, Vassilvitskii, How Slow is the k-means Method? Proceedings of the 2006 Symposium on Computational Geometry]

slide credit: Václav Hlaváč, Bastian Leibe

k-Means Clustering / Lloyd's Algorithm

- Does k-means clustering converge?
- k-means cluster optimizes the cost:

$$\sum_{\text{clusters } i} \sum_{\text{points } p \text{ in cluster } i} \|p - c_i\|^2$$

  - Updating cluster centers via the mean reduces cost for same assignments
  - Updating an assignment reduces cost as well
  - Will thus converge to some **local minimum**
  - In practice: stop algorithm if cost change is too small
- Convergence can be superpolynomial $\mathcal{O}\left(2^{\sqrt{2}}\right)$ [Arthur, Vassilvitskii, How Slow is the k-means Method? Proceedings of the 2006 Symposium on Computational Geometry]
- Finding global optimum is NP-hard even for $k = 2$

slide credit: Václav Hlaváč, Bastian Leibe

k-Means++

- Can we avoid arbitrarily bad local minima?

slide credit: Bastian Leibe, Steve Seitz

k-Means++

- Can we avoid arbitrarily bad local minima?
- **k-means++ initialization**:
  - Randomly choose first center
  - Repeat until k centers chosen: randomly choose data point with probability proportional to distance to closest cluster center

slide credit: Bastian Leibe, Steve Seitz

k-Means++

- Can we avoid arbitrarily bad local minima?
- **k-means++ initialization**:
  - Randomly choose first center
  - Repeat until k centers chosen: randomly choose data point with probability proportional to distance to closest cluster center
- Intuition: other clusters likely to be “far" away from existing ones

slide credit: Bastian Leibe, Steve Seitz

k-Means++

- Can we avoid arbitrarily bad local minima?
- **k-means++ initialization**:
  - Randomly choose first center
  - Repeat until k centers chosen: randomly choose data point with probability proportional to distance to closest cluster center
- Intuition: other clusters likely to be “far" away from existing ones
- Randomization to reduce influence of outliers (see later slide)

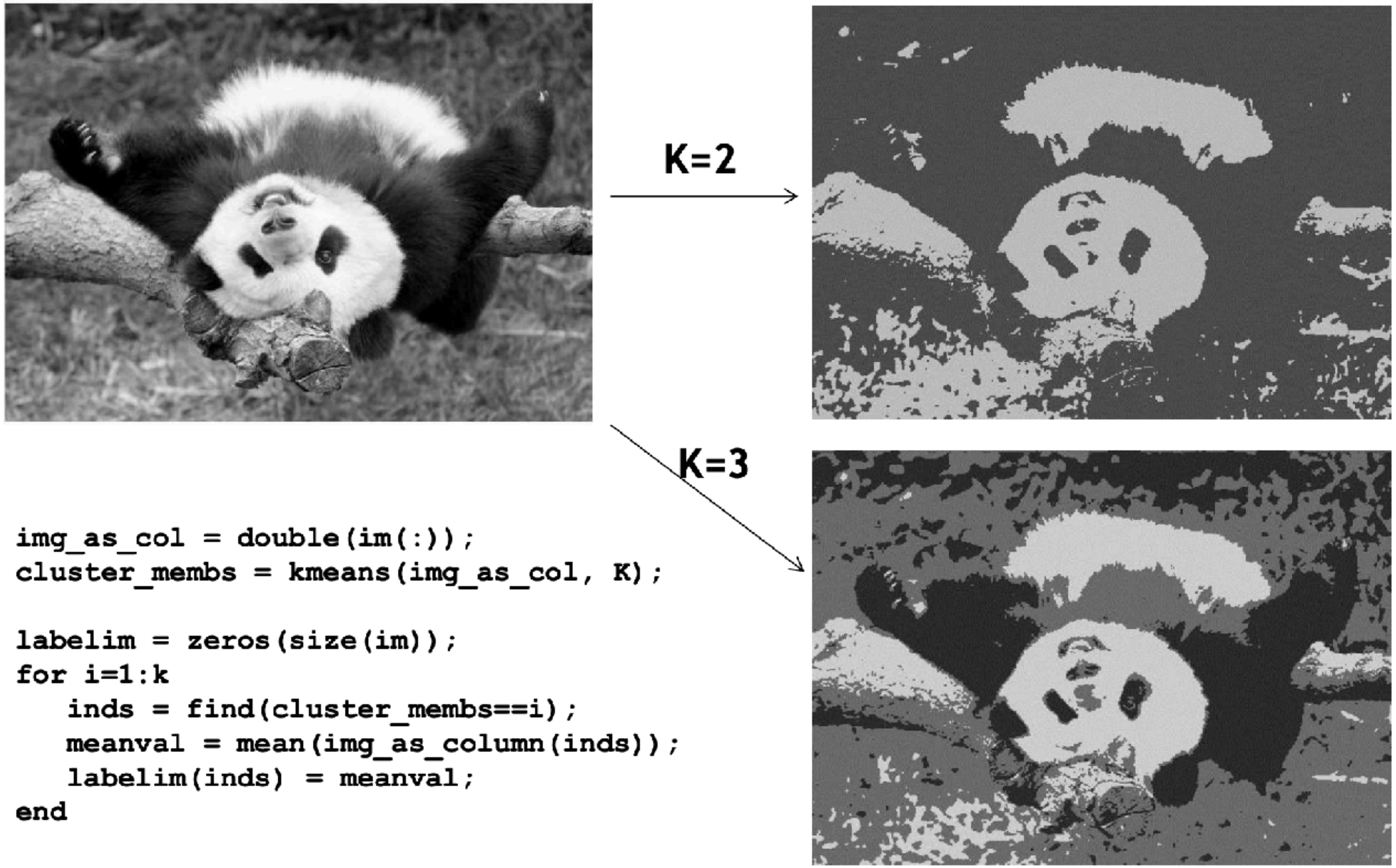
slide credit: Bastian Leibe, Steve Seitz

k-Means++

- Can we avoid arbitrarily bad local minima?
- **k-means++ initialization**:
  - Randomly choose first center
  - Repeat until k centers chosen: randomly choose data point with probability proportional to distance to closest cluster center
- Intuition: other clusters likely to be “far" away from existing ones
- Randomization to reduce influence of outliers (see later slide)
- Arthur & Vassilvitskii 2007: Expected error: $\Theta(\log k) \cdot$ optimum

slide credit: Bastian Leibe, Steve Seitz

Example




```
img_as_col = double(im(:));
cluster_membs = kmeans(img_as_col, K);

labelim = zeros(size(im));
for i=1:k
   inds = find(cluster_membs==i);
   meanval = mean(img_as_column(inds));
   labelim(inds) = meanval;
end
```

slide credit: Bastian Leibe

Example

Image, courtesy Ondoej Drbohlav

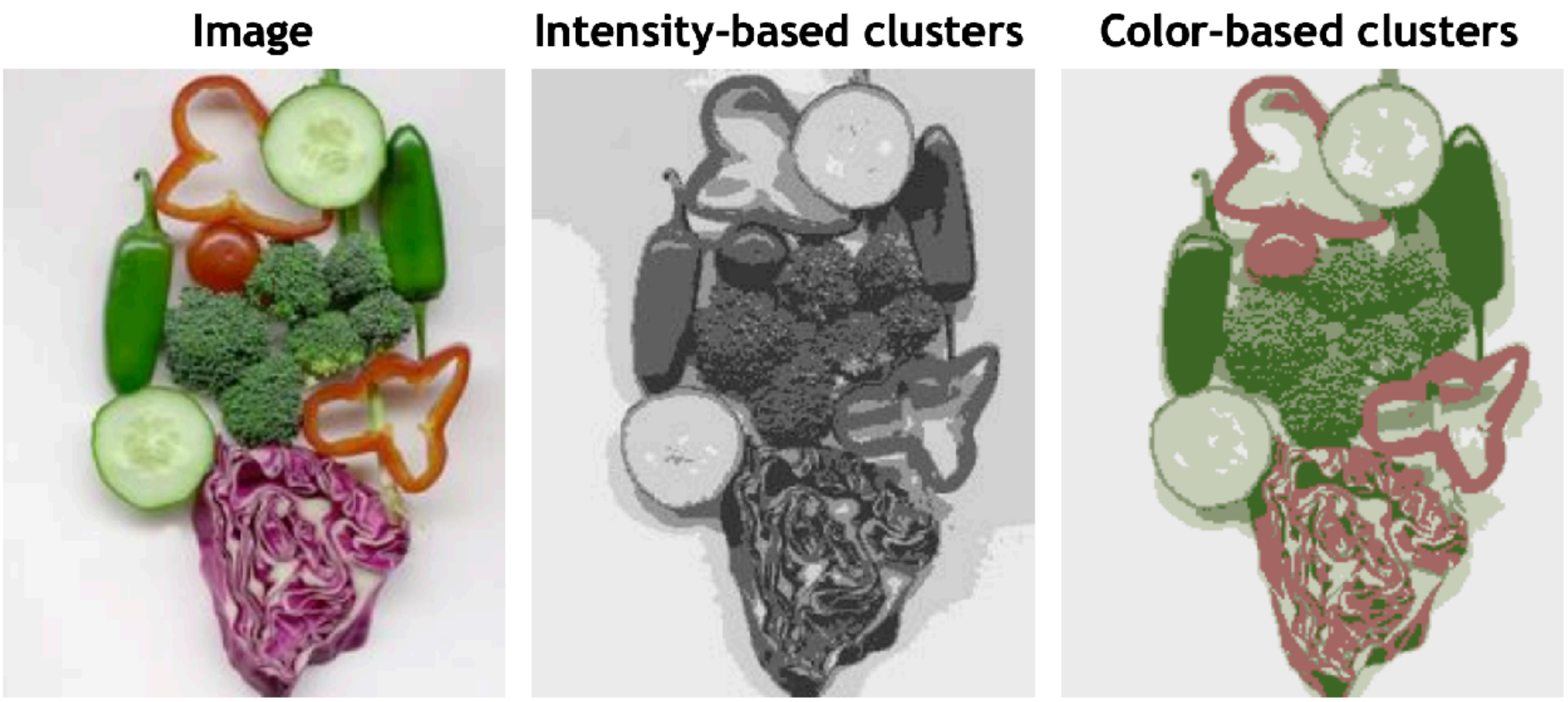
- Clustering into $k = 2$ regions based on absolute values of 1st derivative

$$\left(\left| \frac{\partial I(x,y)}{\partial x} \right|, \left| \frac{\partial I(x,y)}{\partial y} \right| \right)$$

slide credit: Václav Hlaváč

Accounting For Spatial Coherence

- Clustering based on image intensities, color, texture similarity, etc. does not take spatial coherence into account



slide credit: Bastian Leibe, Svetlana Lazebnik

image source: D. Forsyth, J. Ponce, Computer Vision - A Modern Approach, 2nd edition, Pearson, 2011

Accounting For Spatial Coherence

- Clustering based on image intensities, color, texture similarity, etc. does not take spatial coherence into account
- Cluster based on, e.g., color, and pixel position: $(\mathtt{r}, \mathtt{g}, \mathtt{b}, \mathtt{x}, \mathtt{y})$

slide credit: Bastian Leibe, Svetlana Lazebnik

image source: D. Forsyth, J. Ponce, Computer Vision - A Modern Approach, 2nd edition, Pearson, 2011

k-Means Clustering - Discussion

- **Pros**:
  - Simple to implement, fast to compute, many good implementations available (Matlab, Python, …)
  - Converges to local minimum

slide credit: Václav Hlaváč, Bastian Leibe, Kristen Grauman

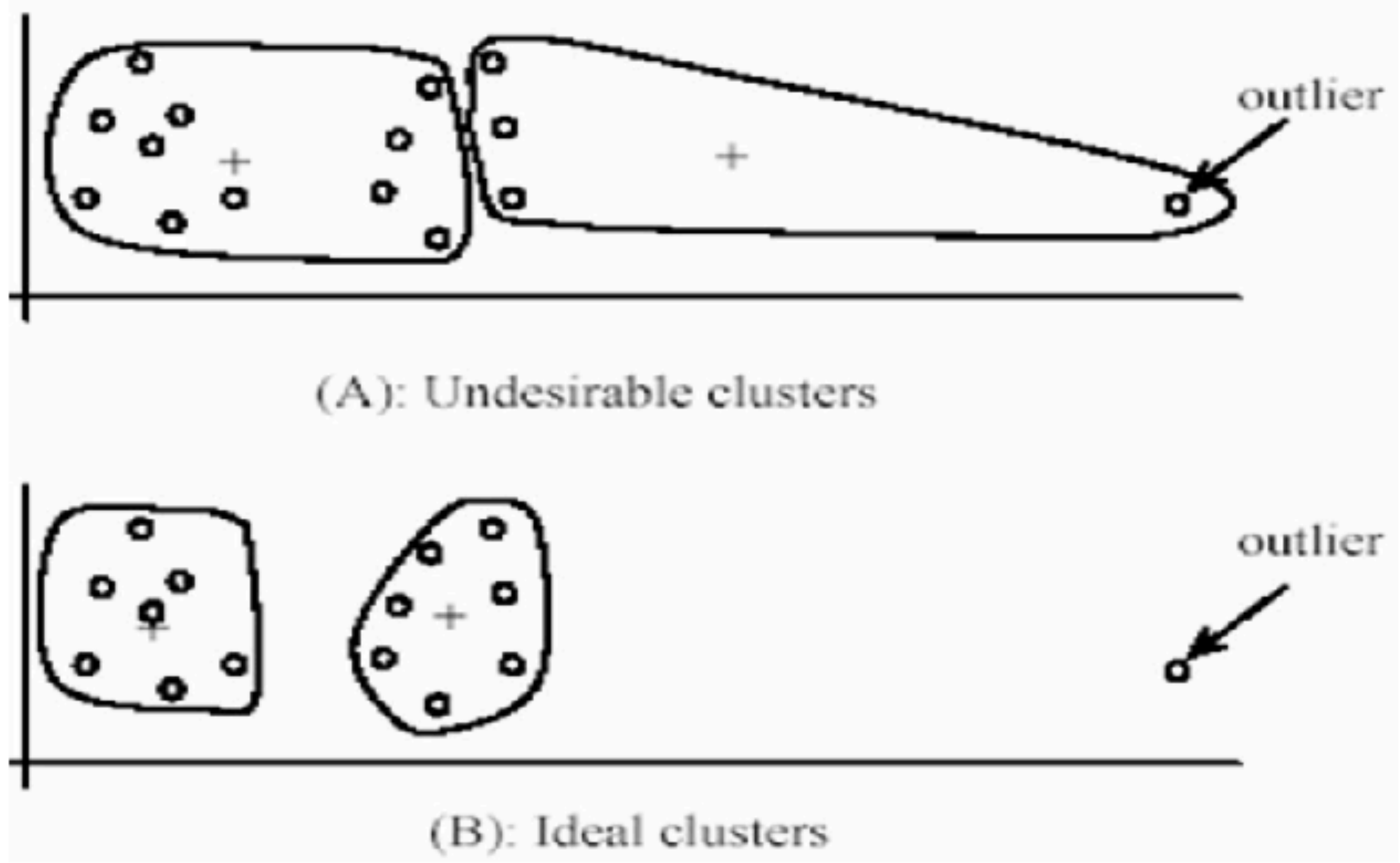
k-Means Clustering - Discussion

- **Pros**:
  - Simple to implement, fast to compute, many good implementations available (Matlab, Python, …)
  - Converges to local minimum
- **Cons**:
  - How to choose good value for k?
  - Sensitive to outliers (randomization can help)
  - Sensitive to initialization
  - Clusters are spherical
  - Mean needs to be defined

slide credit: Václav Hlaváč, Bastian Leibe, Kristen Grauman

k-Means Clustering - Discussion

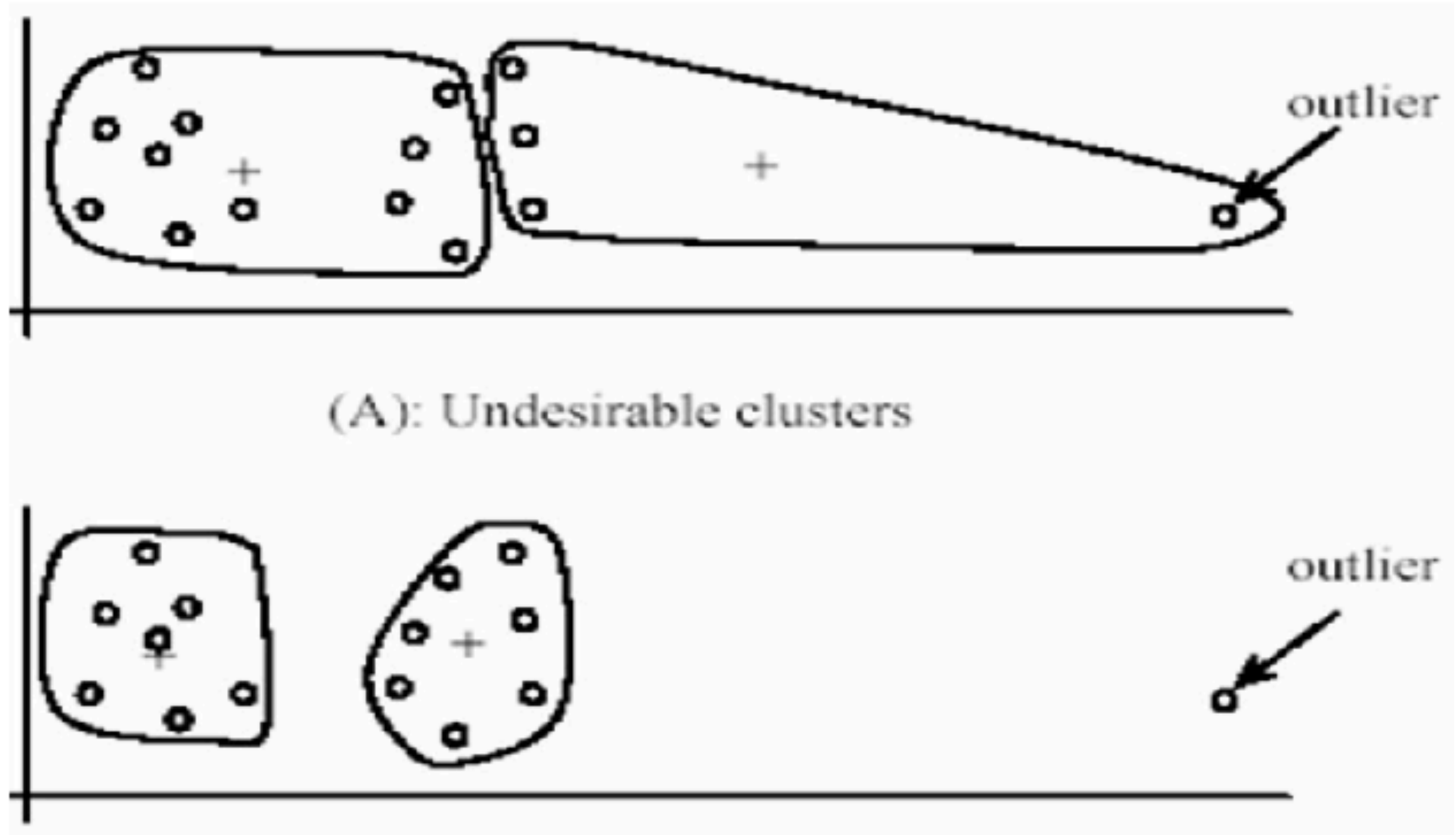
- **Pros**:
  - Simple to implement, fast to compute, many good implementations available (Matlab, Python, …)
  - Converges to local minimum
- **Cons**:
  - How to choose good value for k?
  - Sensitive to outliers (randomization can help)
  - Sensitive to initialization
  - Clusters are spherical
  - Mean needs to be defined



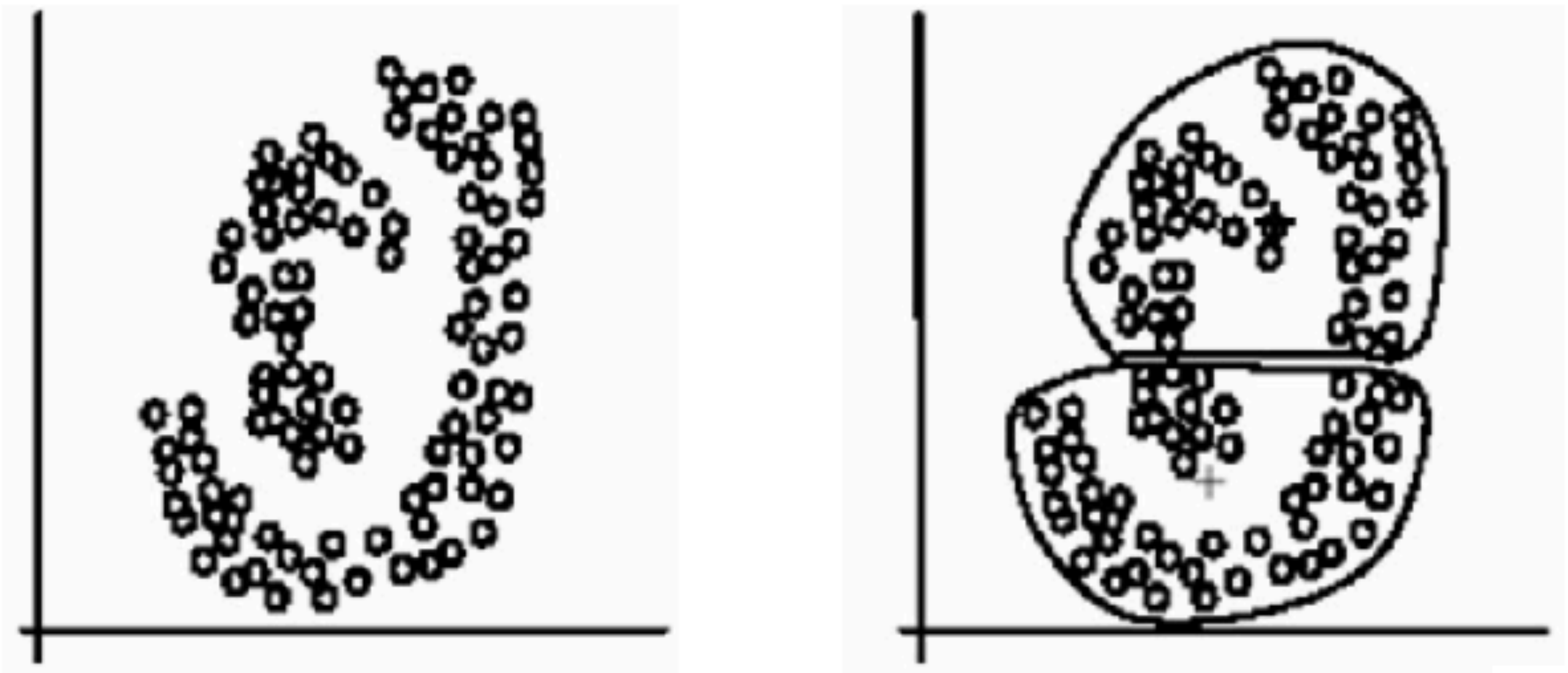
slide credit: Václav Hlaváč, Bastian Leibe, Kristen Grauman

k-Means Clustering - Discussion

- **Pros**:
  - Simple to implement, fast to compute, many good implementations available (Matlab, Python, …)
  - Converges to local minimum
- **Cons**:
  - How to choose good value for k?
  - Sensitive to outliers (randomization can help)
  - Sensitive to initialization
  - Clusters are spherical
  - Mean needs to be defined

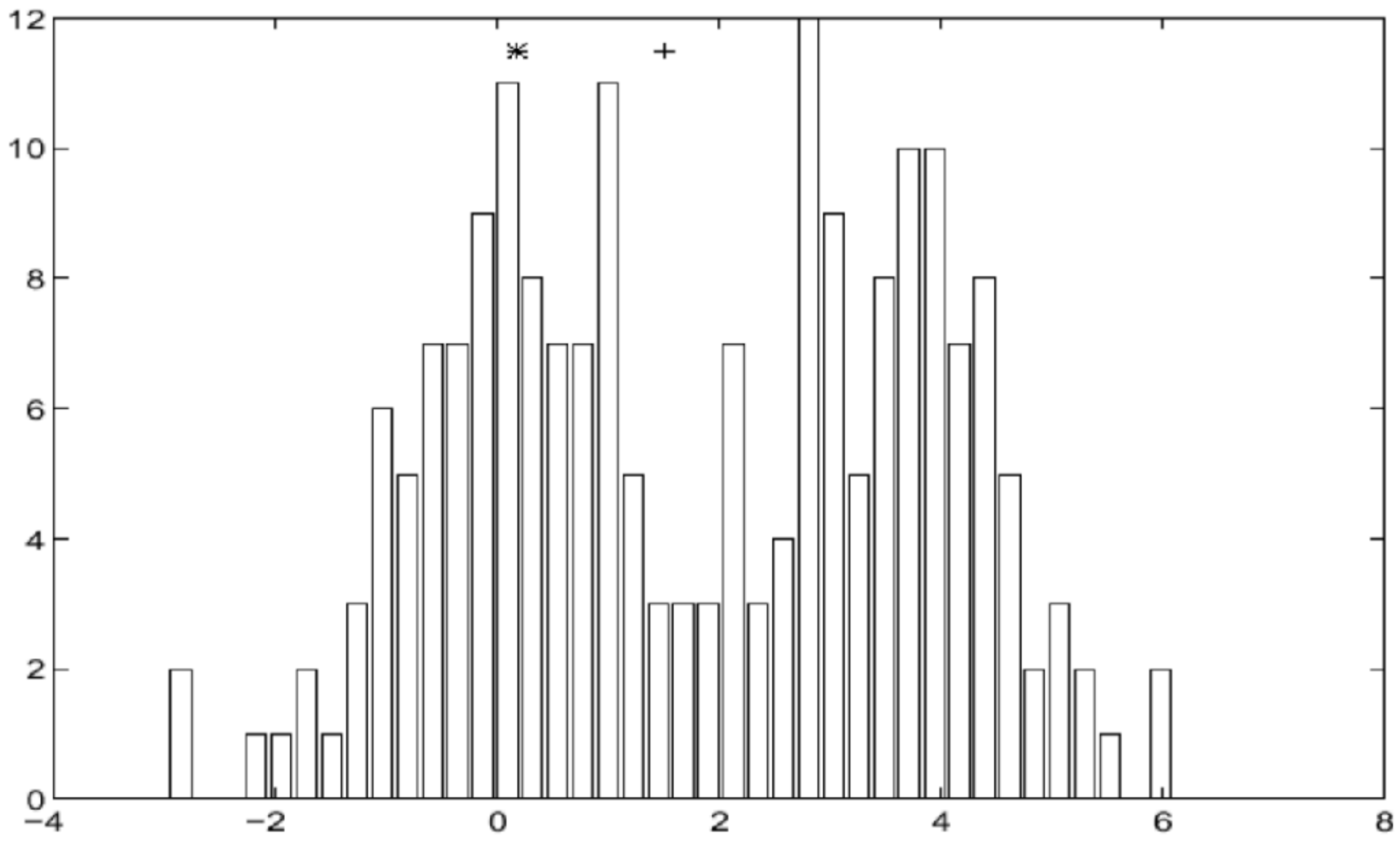


(A): Undesirable clusters
(B): Ideal clusters


(A): Two natural clusters
(B): k-means clusters

slide credit: Václav Hlaváč, Bastian Leibe, Kristen Grauman

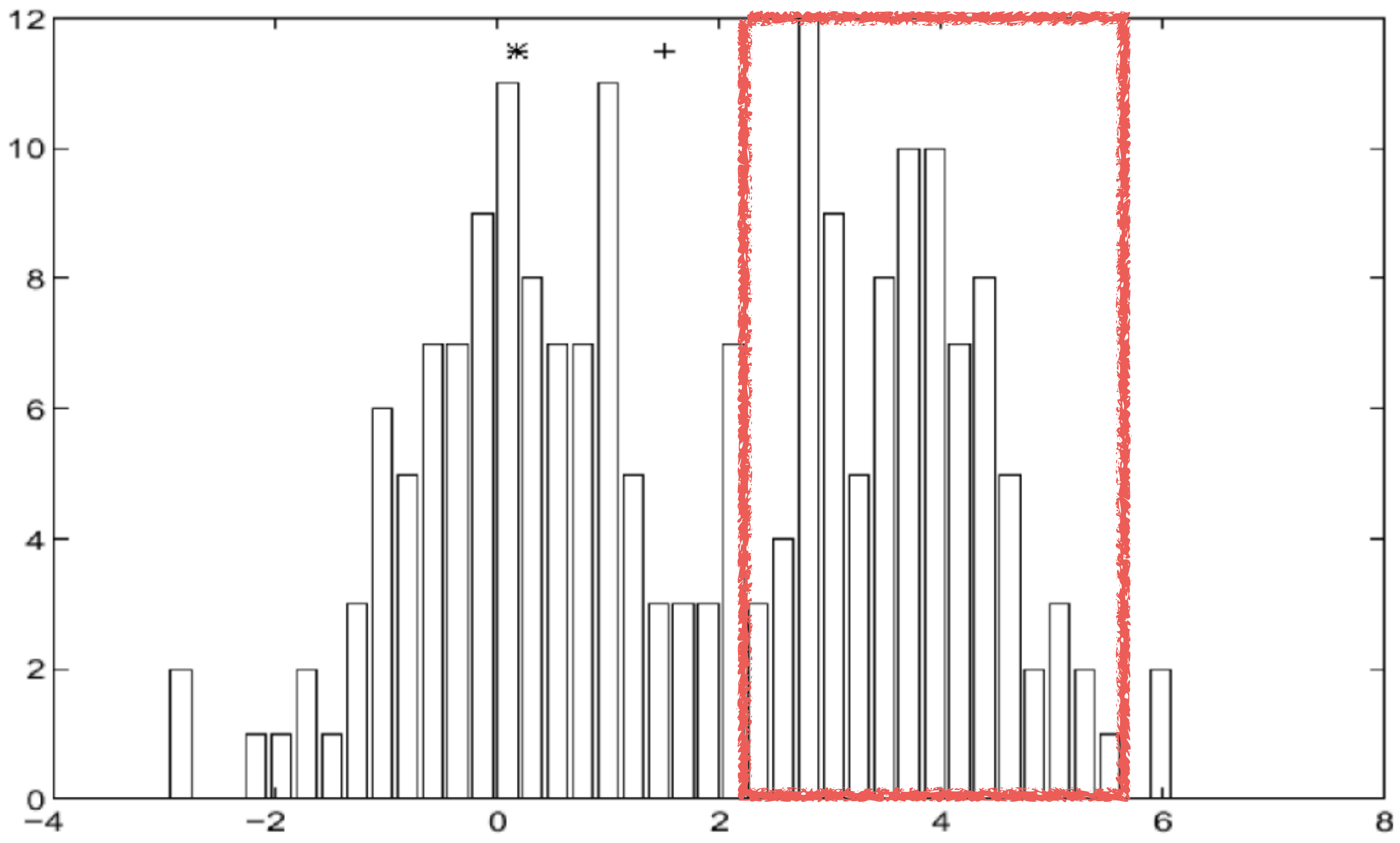
Mean Shift Clustering



- How many modes (= cluster centers) are there?

slide credit: Steve Seitz, Bastian Leibe

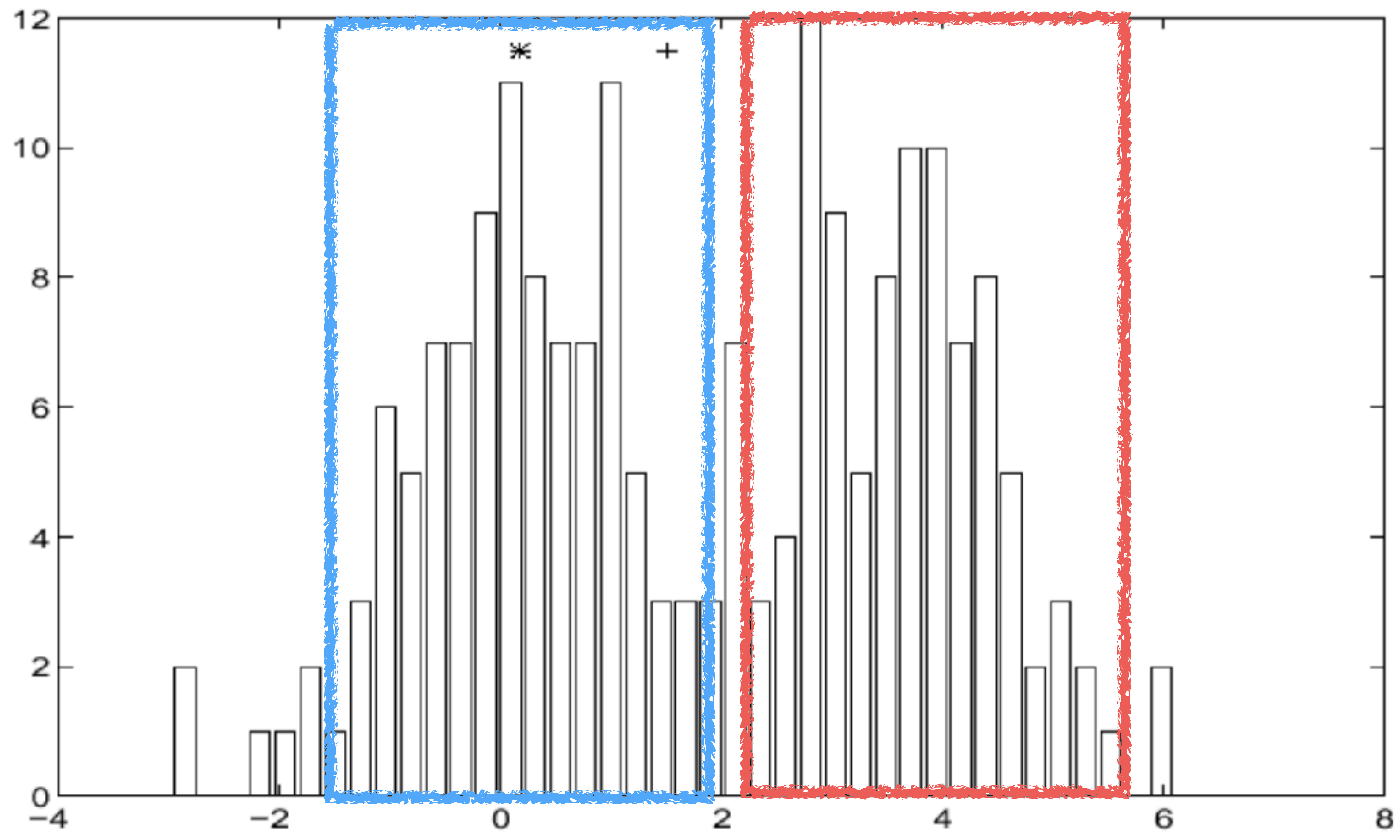
Mean Shift Clustering



- How many modes (= cluster centers) are there?

slide credit: Steve Seitz, Bastian Leibe

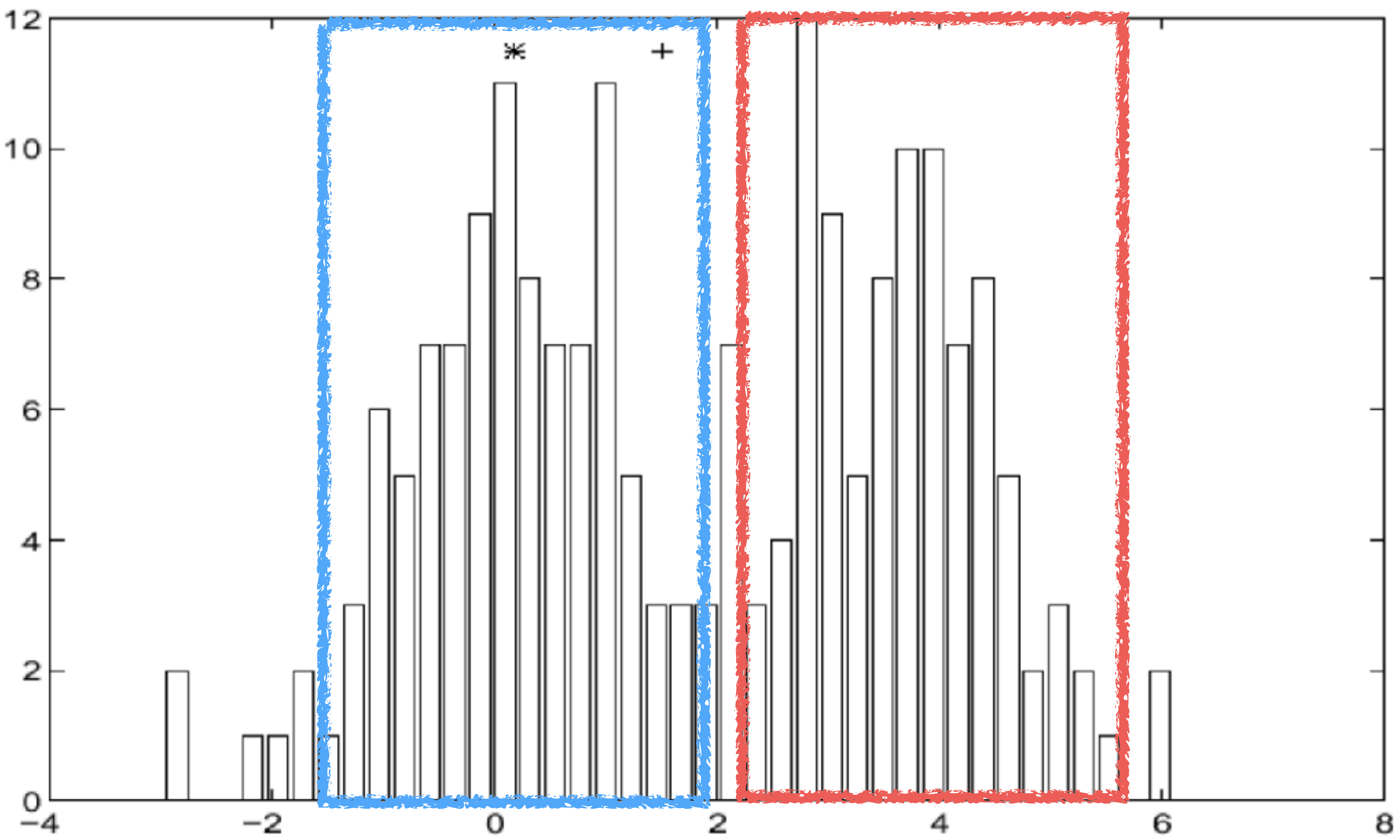
Mean Shift Clustering



- How many modes (= cluster centers) are there?

slide credit: Steve Seitz, Bastian Leibe

Mean Shift Clustering



- How many modes (= cluster centers) are there?
- How to detect automatically?

slide credit: Steve Seitz, Bastian Leibe

Mean Shift Clustering

- **Modes** = local maxima of density of given distribution

slide credit: Václav Hlaváč, Bastian Leibe

Mean Shift Clustering

- **Modes** = local maxima of density of given distribution
- Assumption: density increases towards the modes

slide credit: Václav Hlaváč, Bastian Leibe

Mean Shift Clustering

- **Modes** = local maxima of density of given distribution
- Assumption: density increases towards the modes
- Estimation of the **density gradient** [Fukunaga K.: Introduction to Statistical Pattern Recognition, Academic Press, New York, 1972]:

slide credit: Václav Hlaváč, Bastian Leibe

Mean Shift Clustering

- **Modes** = local maxima of density of given distribution
- Assumption: density increases towards the modes
- Estimation of the **density gradient** [Fukunaga K.: Introduction to Statistical Pattern Recognition, Academic Press, New York, 1972]:
  - Sample **mean of local samples** points in the direction of higher density, provides estimate of the gradient

slide credit: Václav Hlaváč, Bastian Leibe

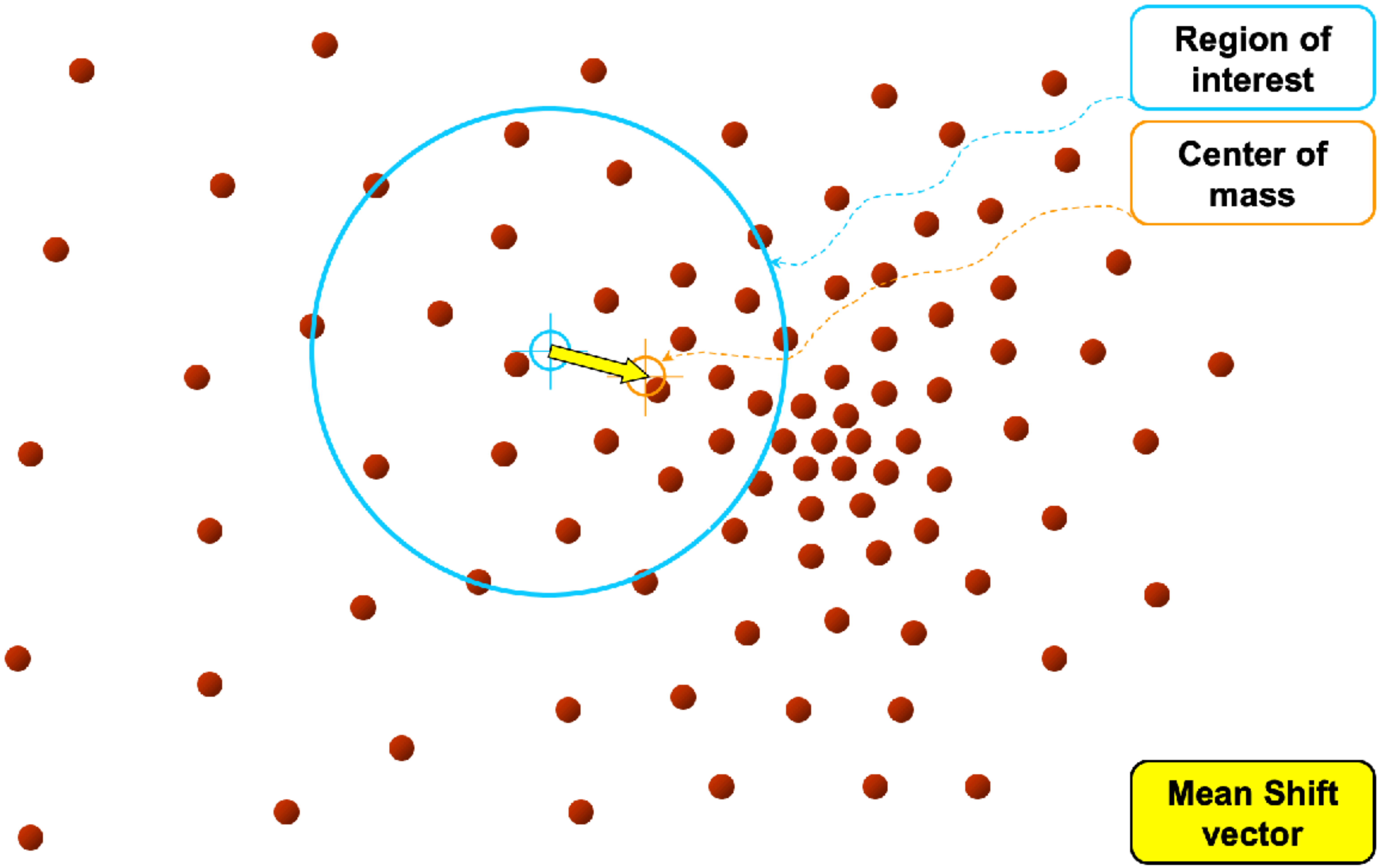
Mean Shift Clustering

- **Modes** = local maxima of density of given distribution
- Assumption: density increases towards the modes
- Estimation of the **density gradient** [Fukunaga K.: Introduction to Statistical Pattern Recognition, Academic Press, New York, 1972]:
  - Sample **mean of local samples** points in the direction of higher density, provides estimate of the gradient
  - Mean shift vector m for point p:

$$m = \sum_{i \in \text{window}} w_i \left(p_i - p\right), \quad w_i = \text{dist}(p, p_i)$$

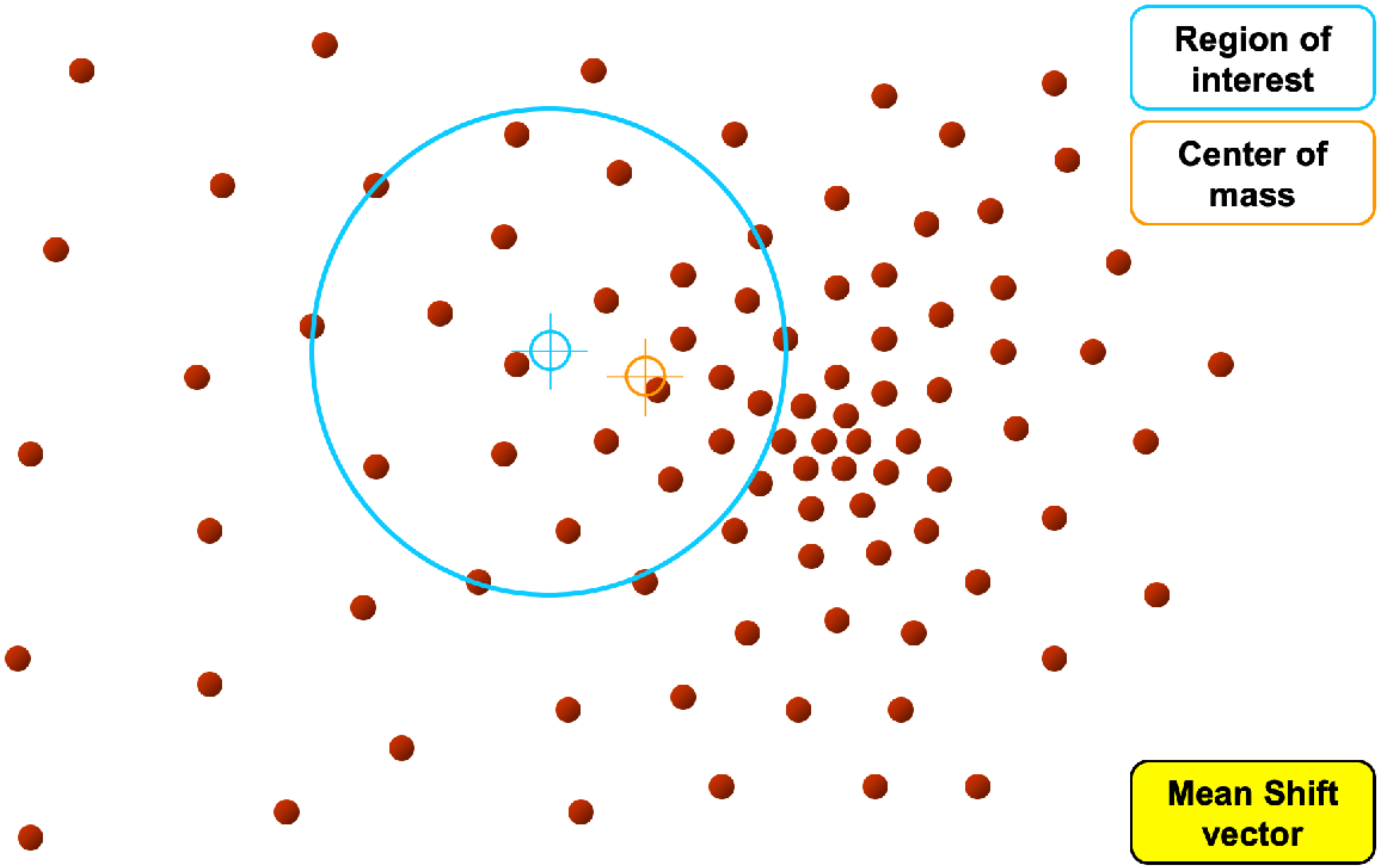
slide credit: Václav Hlaváč, Bastian Leibe

Mean Shift Algorithm



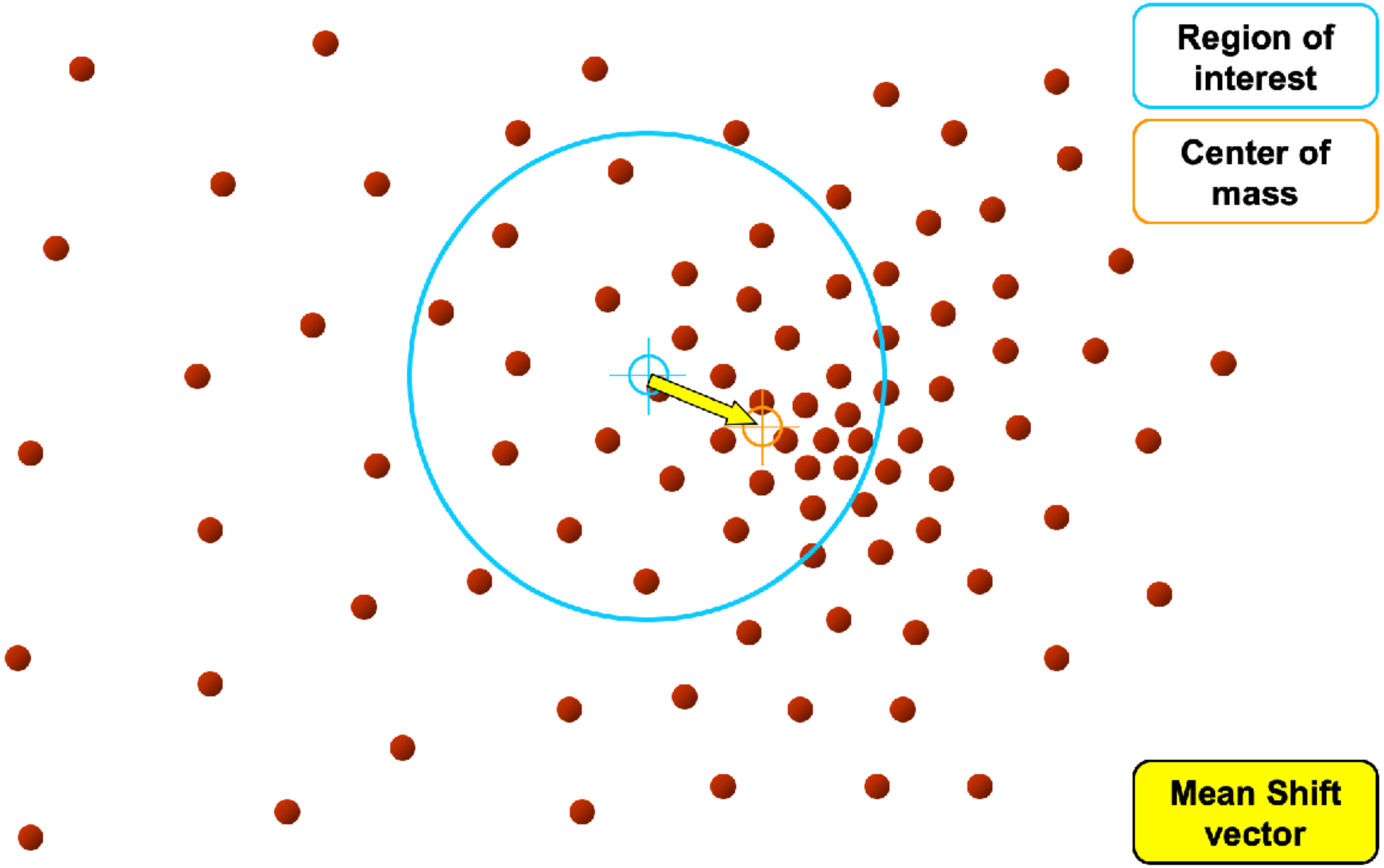
slide credit: Y. Ukrainitz & B. Sarel

Mean Shift Algorithm



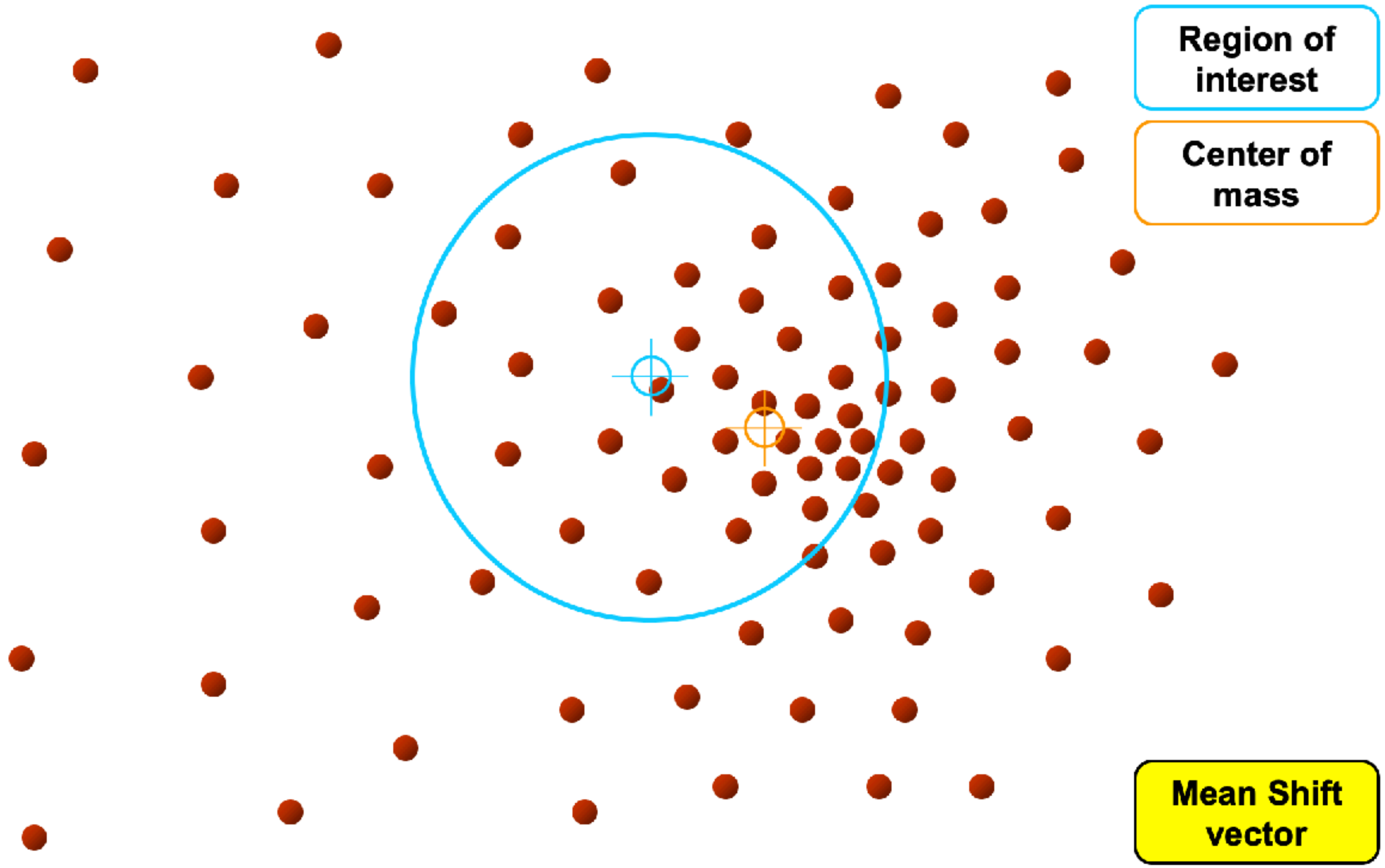
slide credit: Y. Ukrainitz & B. Sarel

Mean Shift Algorithm



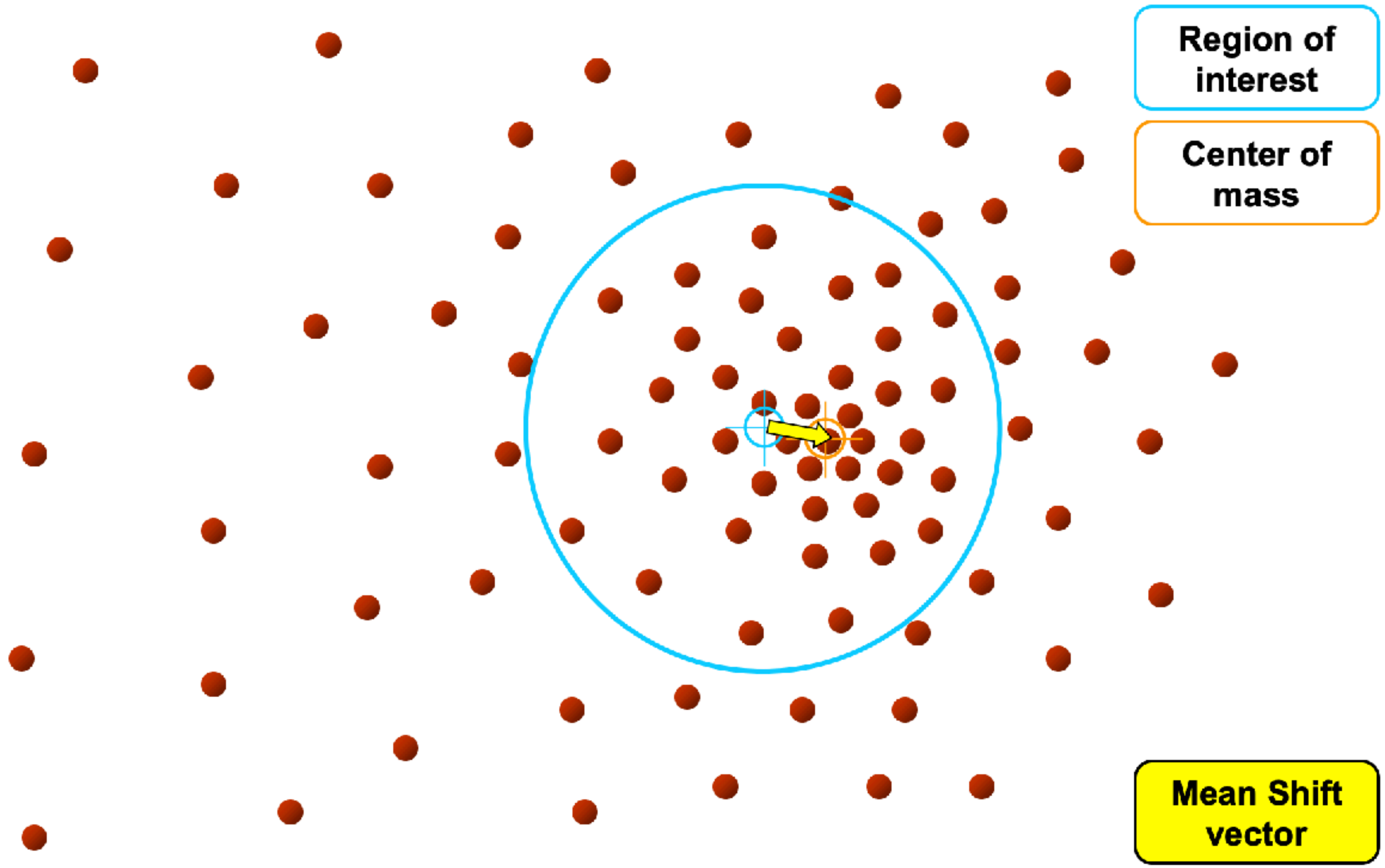
slide credit: Y. Ukrainitz & B. Sarel

Mean Shift Algorithm



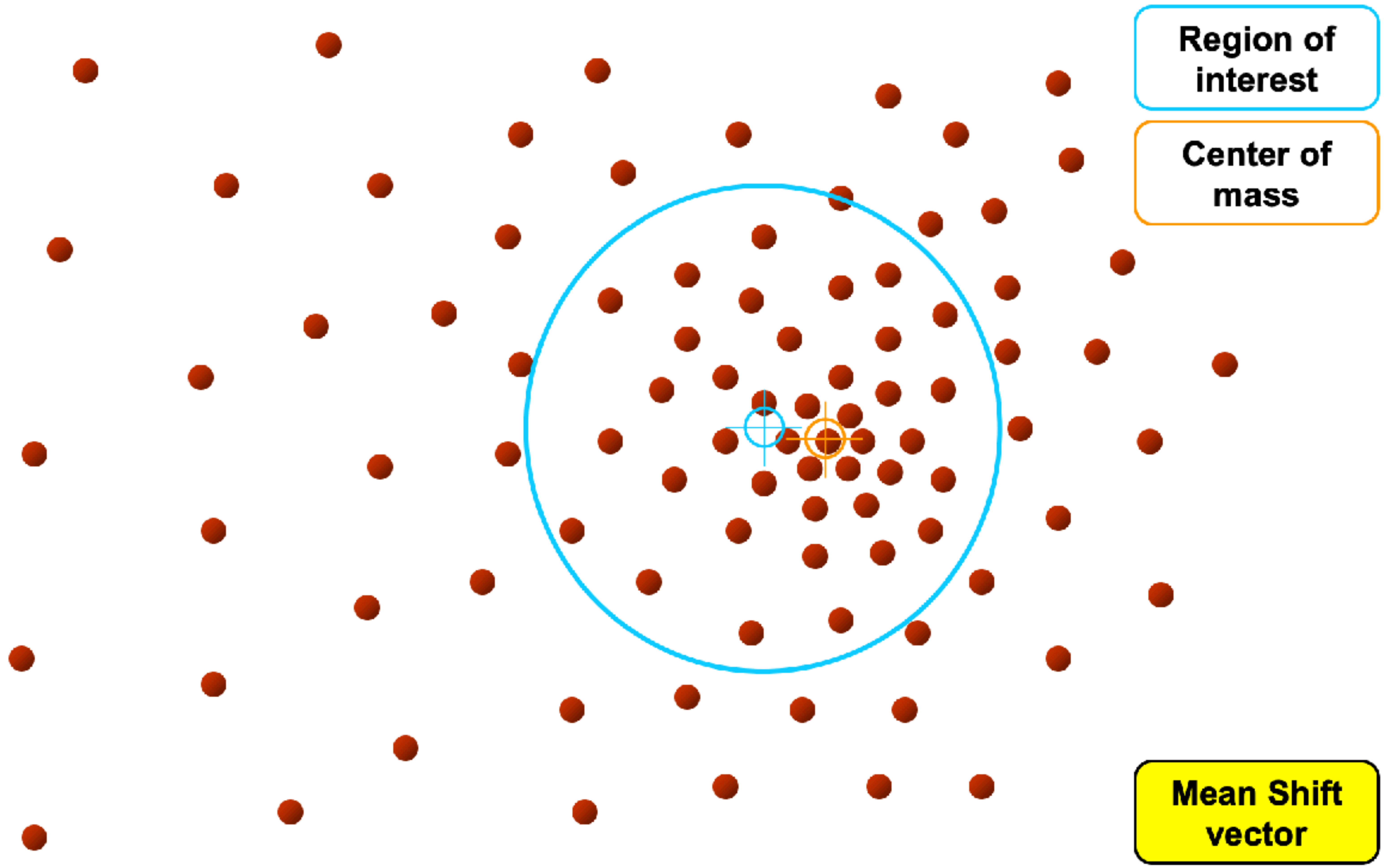
slide credit: Y. Ukrainitz & B. Sarel

Mean Shift Algorithm



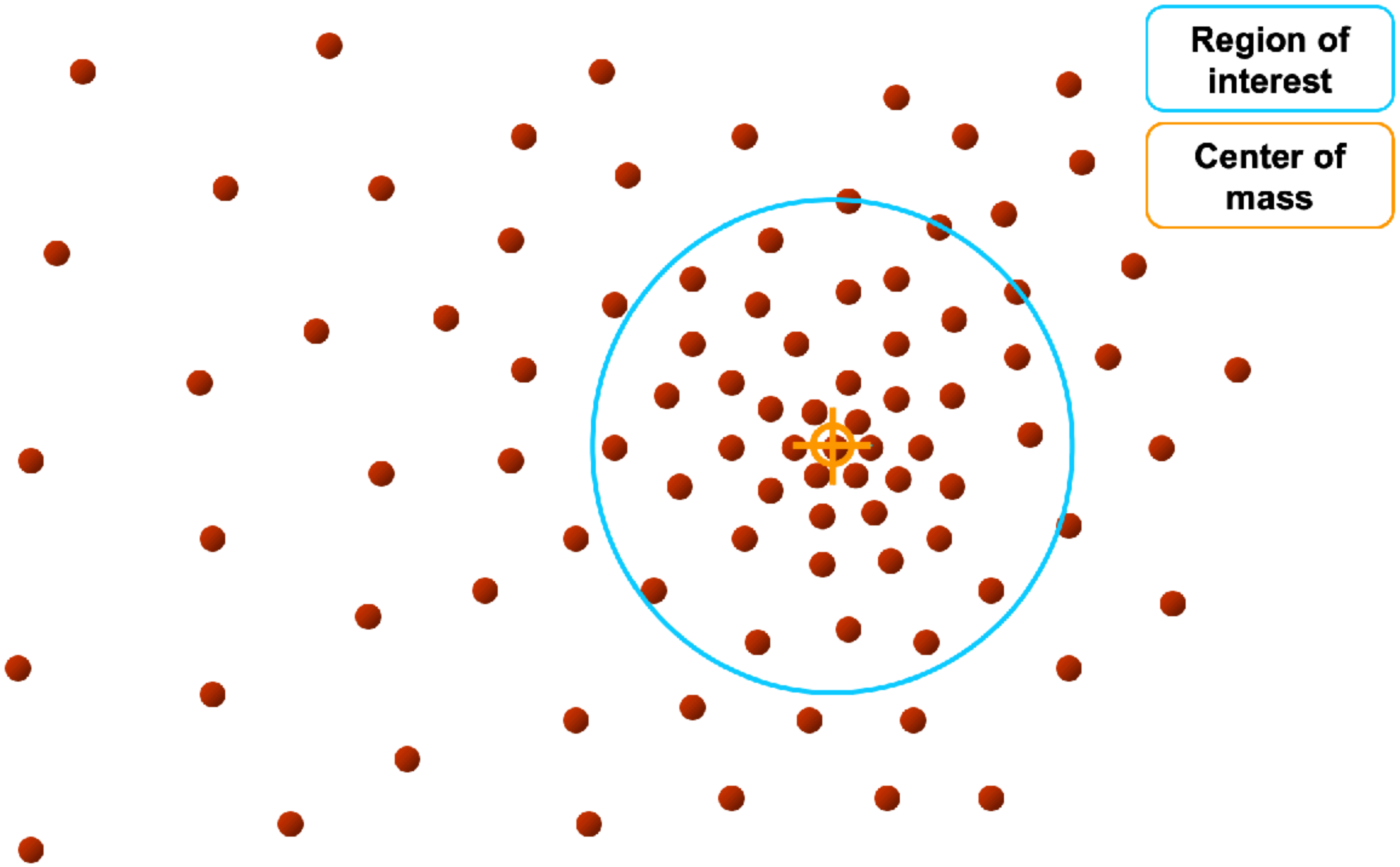
slide credit: Y. Ukrainitz & B. Sarel

Mean Shift Algorithm



slide credit: Y. Ukrainitz & B. Sarel

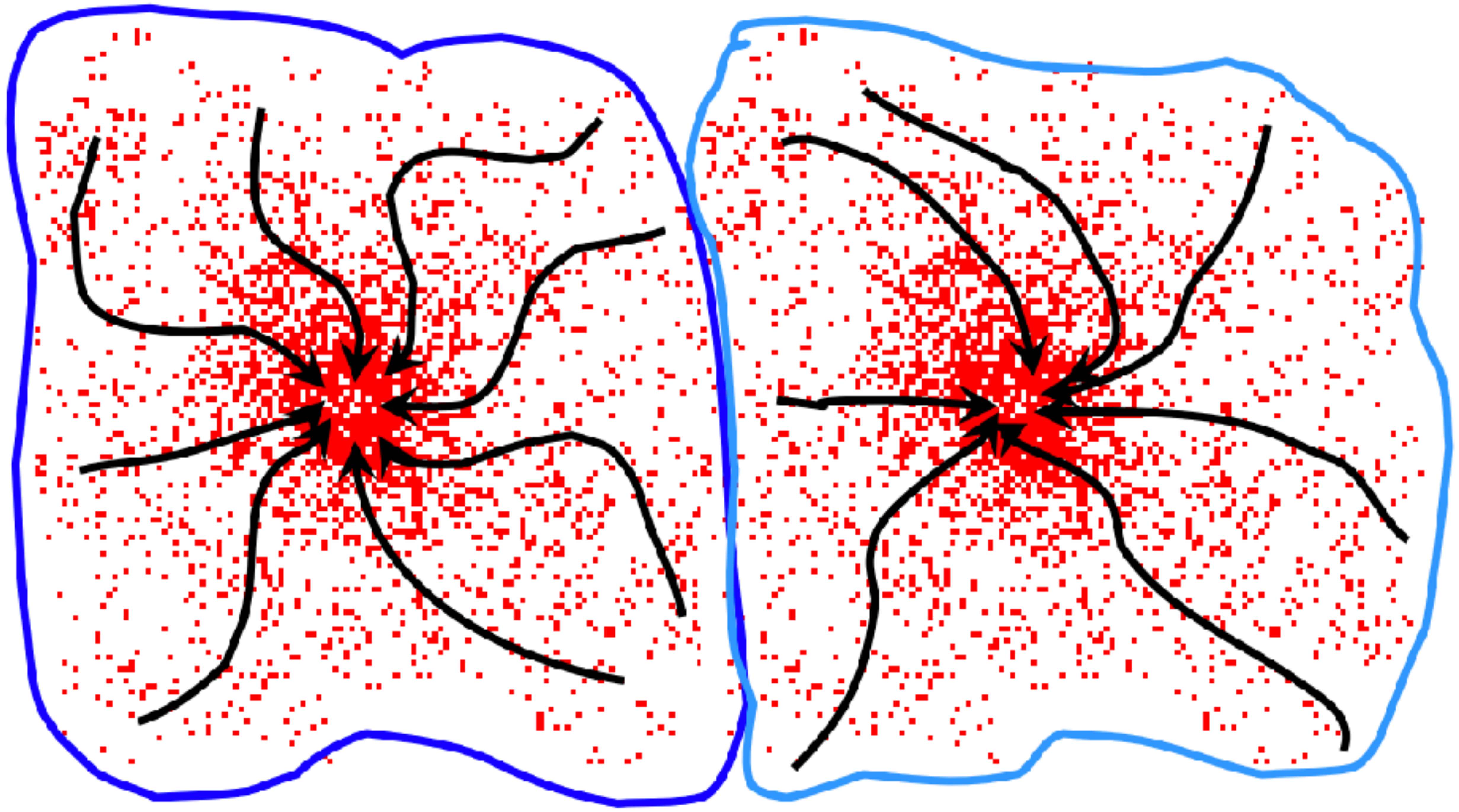
Mean Shift Algorithm



slide credit: Y. Ukrainitz & B. Sarel

Mean Shift Clustering

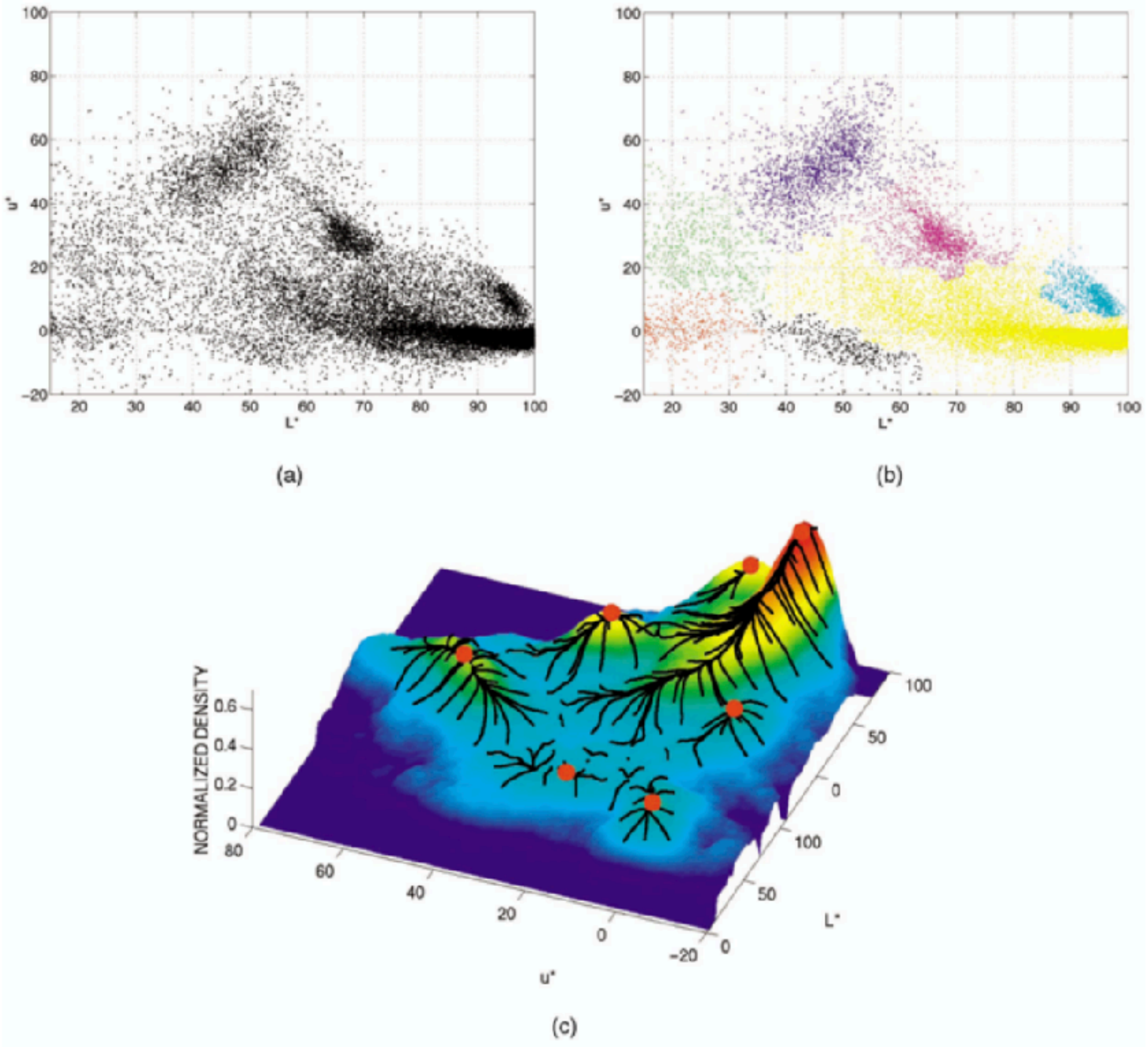
- **Clusters**: all data points in attraction basin of mode
- **Attraction basin**: regions where mean shift leads to same mode



slide credit: Bastian Leibe, Y. Ukrainitz & B. Sarel

Mean Shift Segmentation

- Extract features (colors, intensities, gradients, etc.)
- Initialize search windows at pixel positions / uniformly distributed over image
- Run mean shift for each window
- Merge windows that end up on the same “peak” or mode



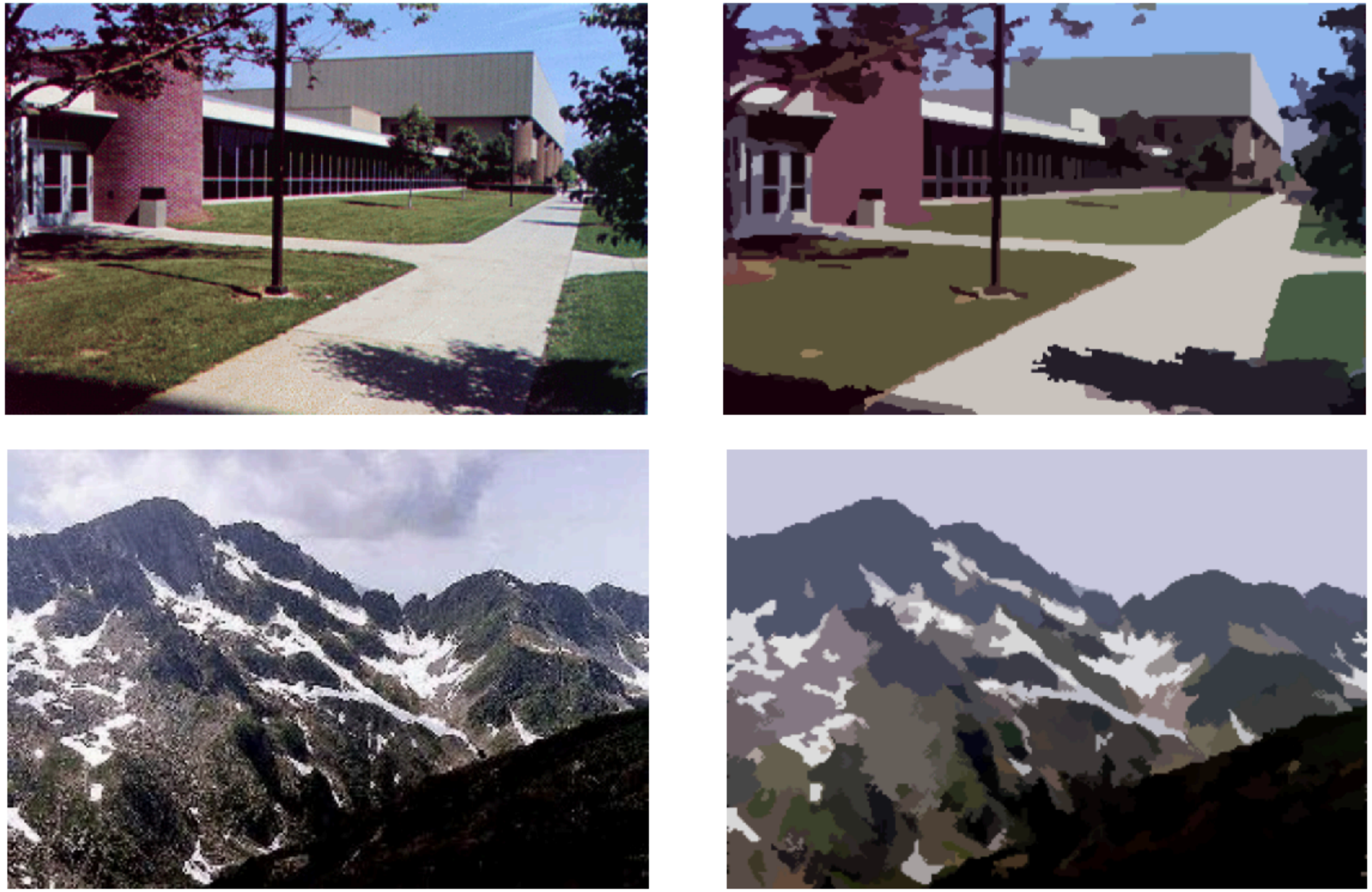
slide credit: Václav Hlaváč, Bastian Leibe

Examples

[D. Comaniciu and P. Meer, Mean Shift: A Robust Approach toward Feature Space Analysis, PAMI 2002]

slide credit: Václav Hlaváč, Bastian Leibe, Svetlana Lazebnik

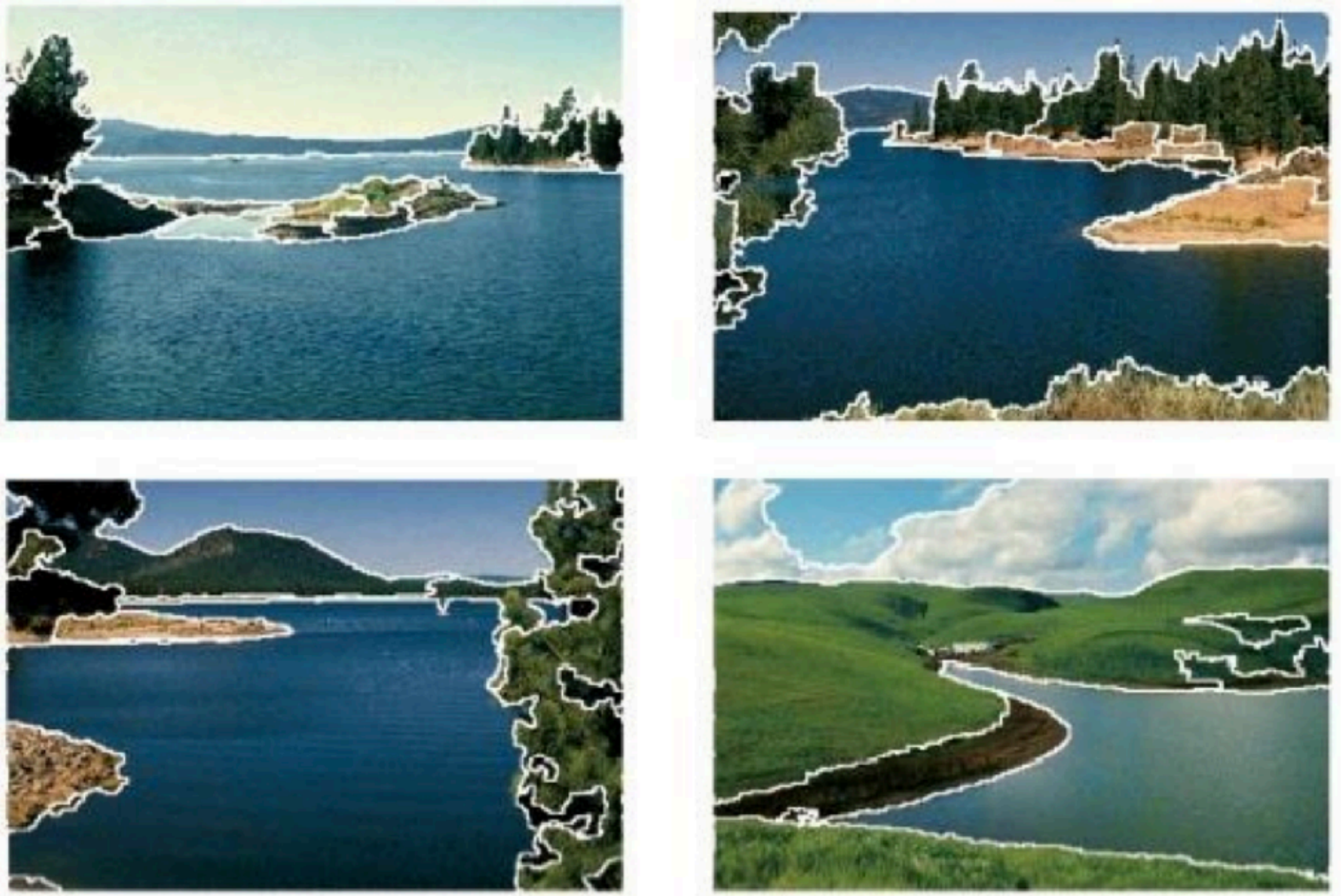
Examples



[D. Comaniciu and P. Meer, Mean Shift: A Robust Approach toward Feature Space Analysis, PAMI 2002]

slide credit: Václav Hlaváč, Bastian Leibe, Svetlana Lazebnik

Examples



[D. Comaniciu and P. Meer, Mean Shift: A Robust Approach toward Feature Space Analysis, PAMI 2002]

slide credit: Václav Hlaváč, Bastian Leibe, Svetlana Lazebnik

Mean Shift Clustering - Discussion

- **Pros**:
  - Model-free, does not assume any prior shape (spherical, elliptical, etc.)
  - Single parameter with physical meaning (window size h)
  - No need to specify number of modes, finds variable number of modes
  - Robust to outliers

slide credit: Václav Hlaváč, Bastian Leibe, Svetlana Lazebnik

Mean Shift Clustering - Discussion

- **Pros**:
  - Model-free, does not assume any prior shape (spherical, elliptical, etc.)
  - Single parameter with physical meaning (window size h)
  - No need to specify number of modes, finds variable number of modes
  - Robust to outliers
- **Cons**:
  - Sensitive to window size h
  - Selecting right window size h not trivial
  - Computationally (relatively) expensive
  - Does not scale well with respect to feature space dimension

slide credit: Václav Hlaváč, Bastian Leibe, Svetlana Lazebnik

Lecture Overview

simple & heuristic

- A simple approach to segmentation: **(intensity) thresholding**
- Segmentation based on spatial coherence: **edge-based segmentation**, **region growing**
- Segmentation as a **clustering** problem: **k-means clustering**, **mean-shift clustering**
- Segmentation as a statistical (unsupervised) learning problem: **expectation maximization (EM) algorithm**
- Next lecture: **graph-based segmentation**, **supervised learning with neural networks** (if time and interest)

complex & principled

slide credit: Václav Hlaváč

A Statistical Learning Perspective on Clustering

- Basic questions of practical relevance:
  - What is the shape of each cluster?
  - What is the probability a point *p* belongs to cluster *c*?

slide credit: Bastian Leibe

A Statistical Learning Perspective on Clustering

- Basic questions of practical relevance:
  - What is the shape of each cluster?
  - What is the probability a point *p* belongs to cluster *c*?
- k-means clustering cannot answer these questions

slide credit: Bastian Leibe

A Statistical Learning Perspective on Clustering

- Basic questions of practical relevance:
  - What is the shape of each cluster?
  - What is the probability a point *p* belongs to cluster *c*?
- k-means clustering cannot answer these questions

- **Statistical approach**:
  - There is a **generative model**: function relating observations $x \in X$ and their hidden state (class label) $y \in Y$
  - Described via the joint probability measure $p(x, y \mid \Theta)$ defined by parameters Θ
  - Want to **learn the parameters from data**

slide credit: Bastian Leibe

Supervised Learning

- If we have $p(x, y \,|\, \Theta)$, we can define classifier / decision function $y = q(x \,|\, \Theta) = \text{argmax}_y p(x, y \,|\, \Theta)$

slide credit: Václav Hlaváč, Bastian Leibe

Supervised Learning

- If we have $p(x, y \,|\, \Theta)$, we can define classifier / decision function $y = q(x \,|\, \Theta) = \text{argmax}_y p(x, y \,|\, \Theta)$
- **Supervised learning**: given labelled training data $((x_1, y_1), \ldots, (x_n, y_n))$
  - Examples: random forests, deep neural networks

slide credit: Václav Hlaváč, Bastian Leibe

Supervised Learning

- If we have $p(x, y \mid \Theta)$, we can define classifier / decision function $y = q(x \mid \Theta) = \text{argmax}_y p(x, y \mid \Theta)$
- **Supervised learning**: given labelled training data $((x_1, y_1), \ldots, (x_n, y_n))$
  - Examples: random forests, deep neural networks

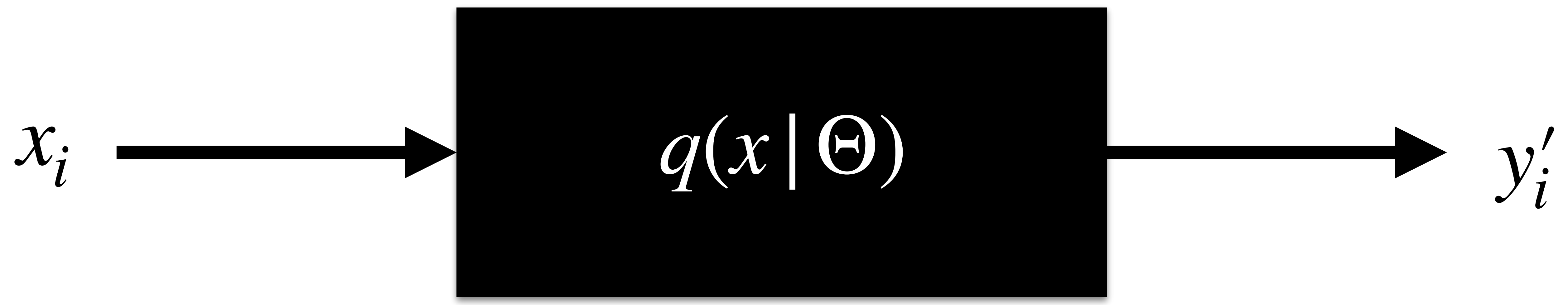
x_i

y_i

slide credit: Václav Hlaváč, Bastian Leibe

Supervised Learning

- If we have $p(x, y \mid \Theta)$, we can define classifier / decision function
  $y = q(x \mid \Theta) = \text{argmax}_y p(x, y \mid \Theta)$
- **Supervised learning**: given labelled training data $((x_1, y_1), \ldots, (x_n, y_n))$
  - Examples: random forests, deep neural networks

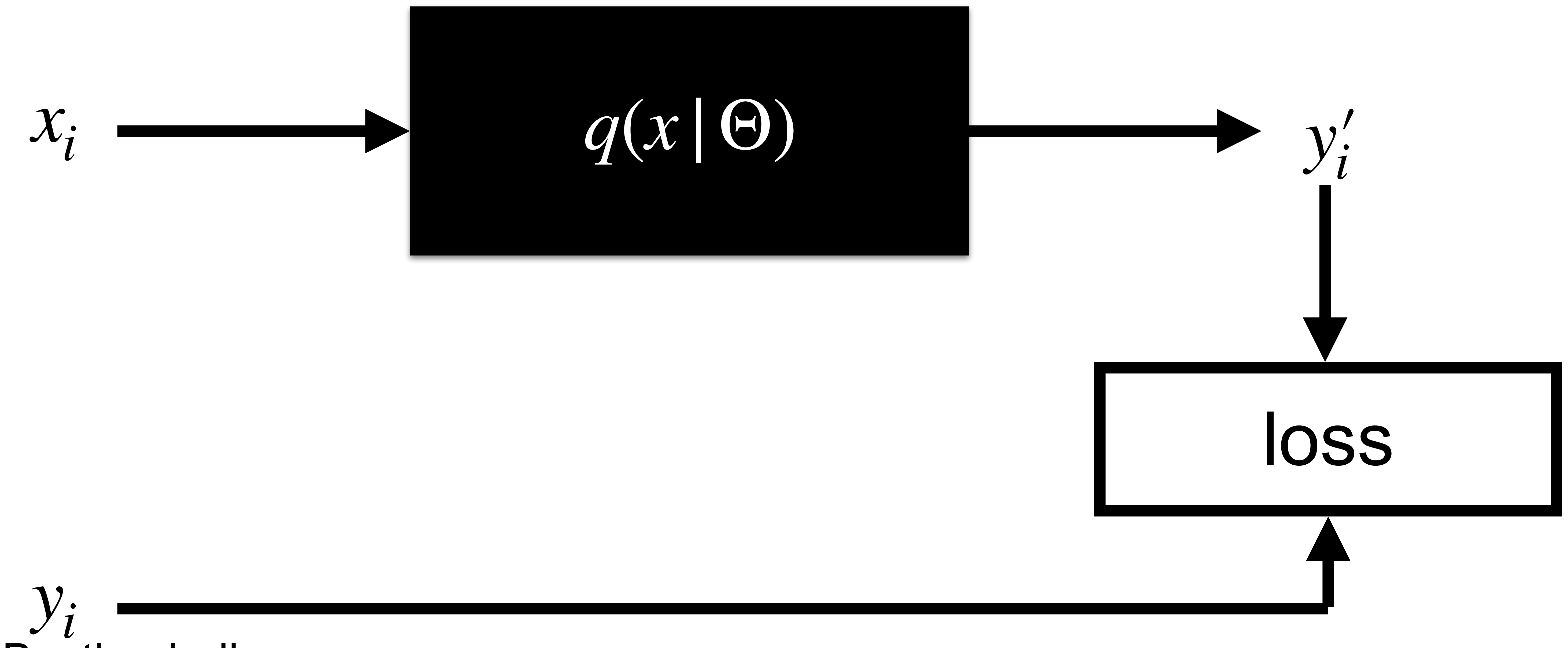


y_i

slide credit: Václav Hlaváč, Bastian Leibe

Supervised Learning

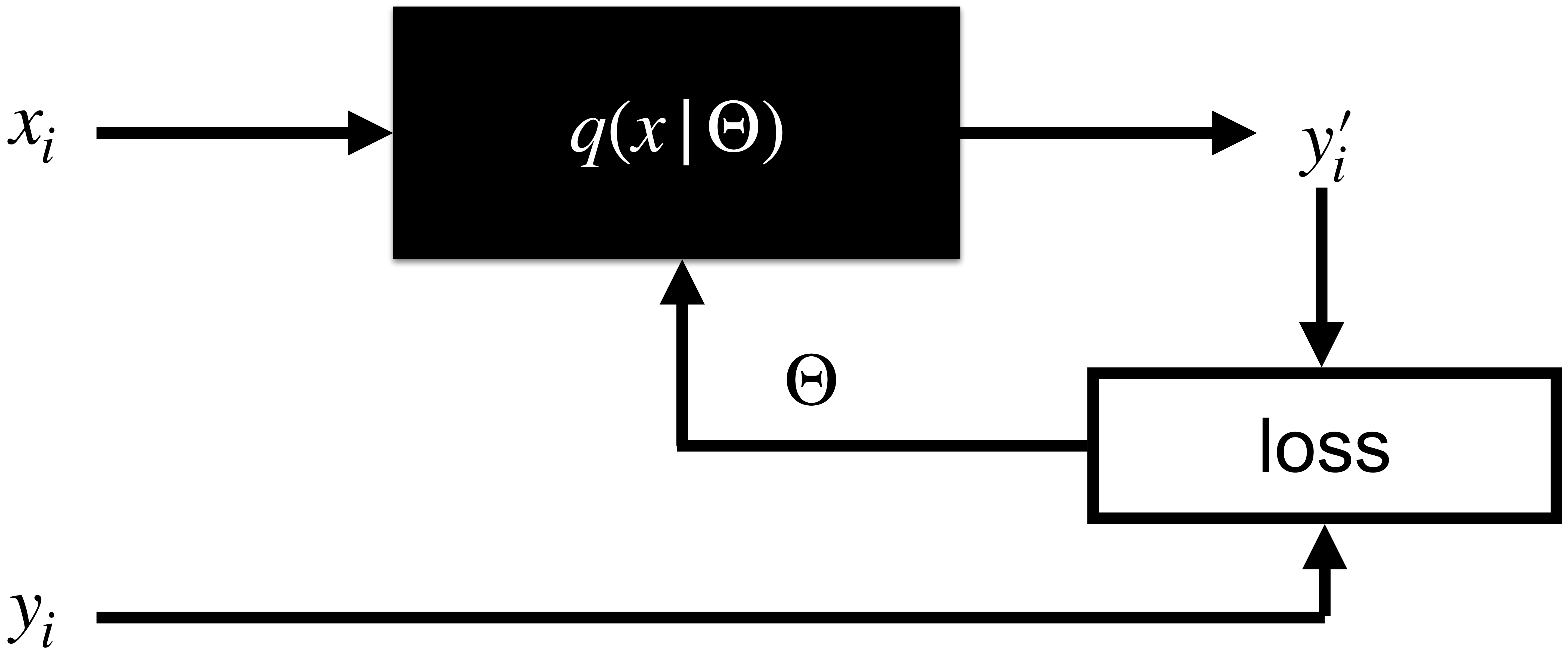
- If we have $p(x, y \mid \Theta)$, we can define classifier / decision function $y = q(x \mid \Theta) = \text{argmax}_y p(x, y \mid \Theta)$
- **Supervised learning**: given labelled training data $((x_1, y_1), \ldots, (x_n, y_n))$
  - Examples: random forests, deep neural networks



slide credit: Václav Hlaváč, Bastian Leibe

Supervised Learning

- If we have $p(x, y \mid \Theta)$, we can define classifier / decision function $y = q(x \mid \Theta) = \text{argmax}_y p(x, y \mid \Theta)$
- **Supervised learning**: given labelled training data $((x_1, y_1), \ldots, (x_n, y_n))$
  - Examples: random forests, deep neural networks



slide credit: Václav Hlaváč, Bastian Leibe

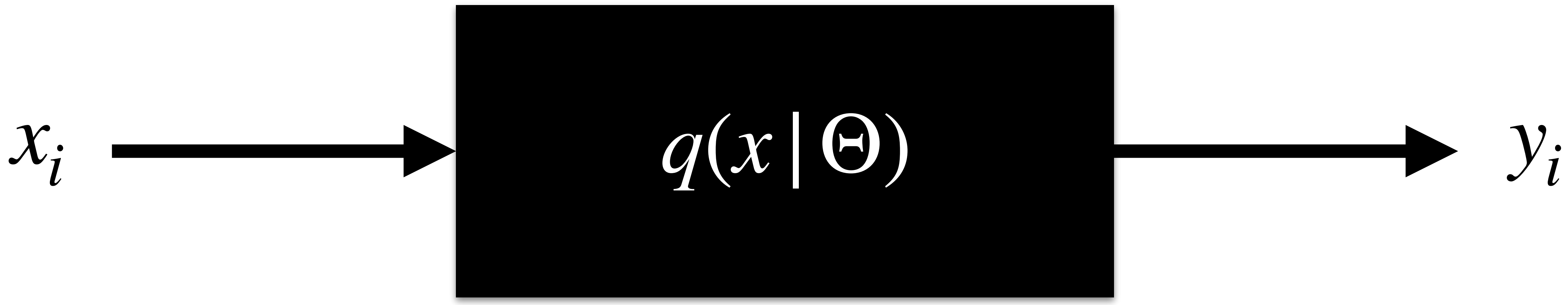
Unsupervised Learning

- If we have $p(x, y \mid \Theta)$, we can define classifier / decision function $y = q(x \mid \Theta) = \text{argmax}_y p(x, y \mid \Theta)$
- **Unsupervised learning**: given unlabelled training data $(x_1, \ldots, x_n)$

slide credit: Václav Hlaváč

Unsupervised Learning

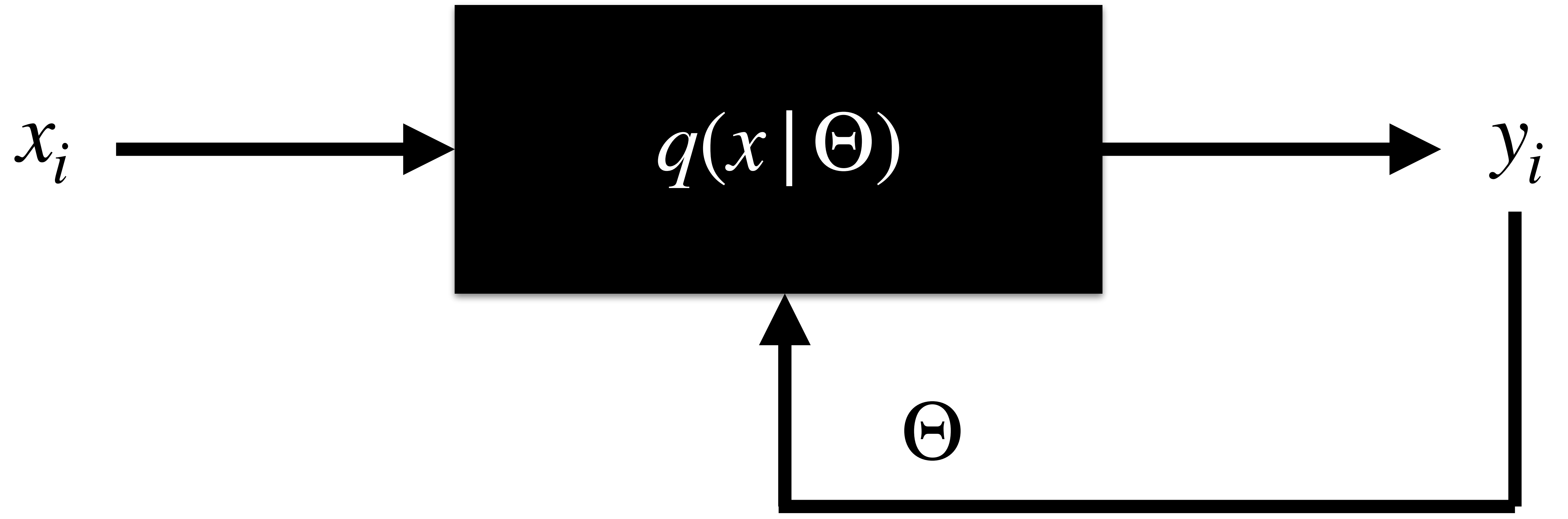
- If we have $p(x, y \,|\, \Theta)$, we can define classifier / decision function $y = q(x \,|\, \Theta) = \text{argmax}_y p(x, y \,|\, \Theta)$
- **Unsupervised learning**: given unlabelled training data $(x_1, \ldots, x_n)$



slide credit: Václav Hlaváč

Unsupervised Learning

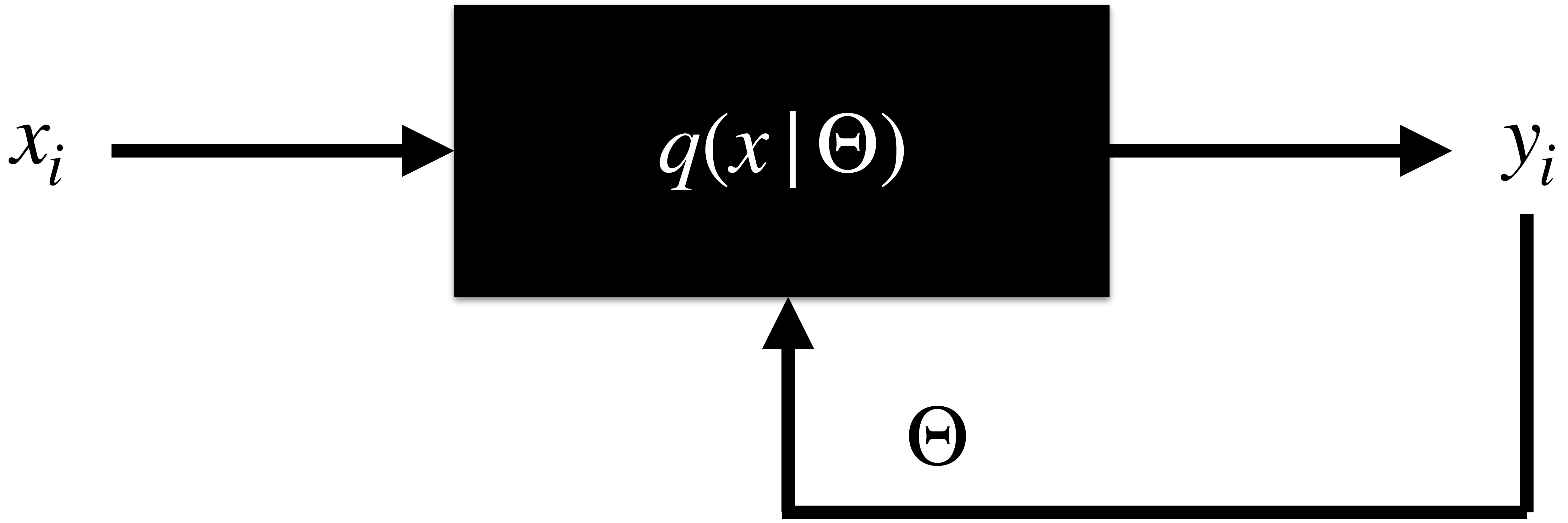
- If we have $p(x, y \mid \Theta)$, we can define classifier / decision function $y = q(x \mid \Theta) = \text{argmax}_y p(x, y \mid \Theta)$
- **Unsupervised learning**: given unlabelled training data $(x_1, \ldots, x_n)$



slide credit: Václav Hlaváč

Unsupervised Learning

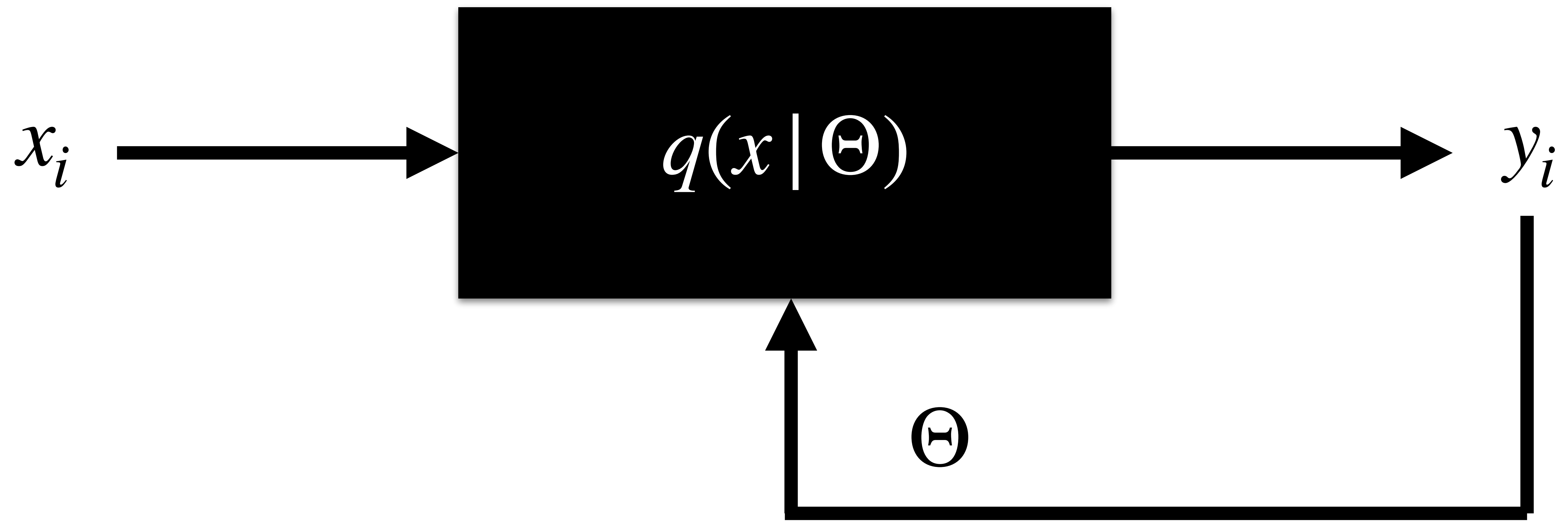
- **Unsupervised learning**: given unlabelled training data $(x_1, \ldots, x_n)$



slide credit: Václav Hlaváč, Bastian Leibe

Unsupervised Learning

- **Unsupervised learning**: given unlabelled training data $(x_1, \ldots, x_n)$

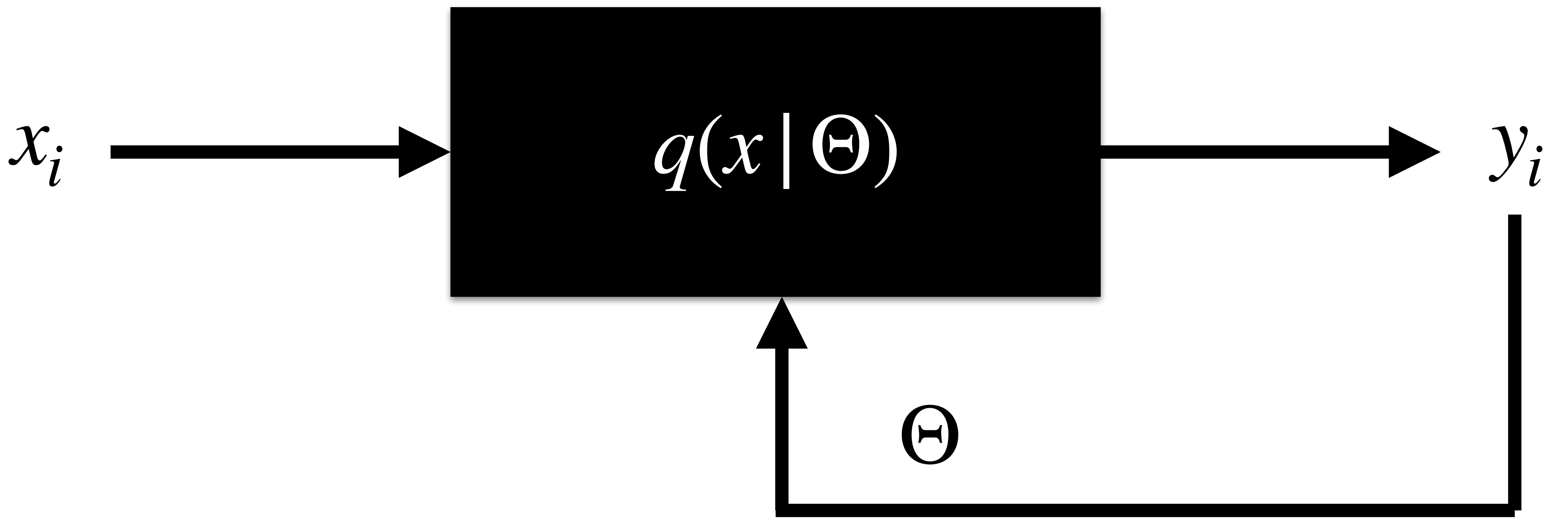


- **Chicken-and-egg problem**: if we have Θ, we can compute $y = q(x \mid \Theta)$; if we have y, we can compute Θ

slide credit: Václav Hlaváč, Bastian Leibe

Unsupervised Learning

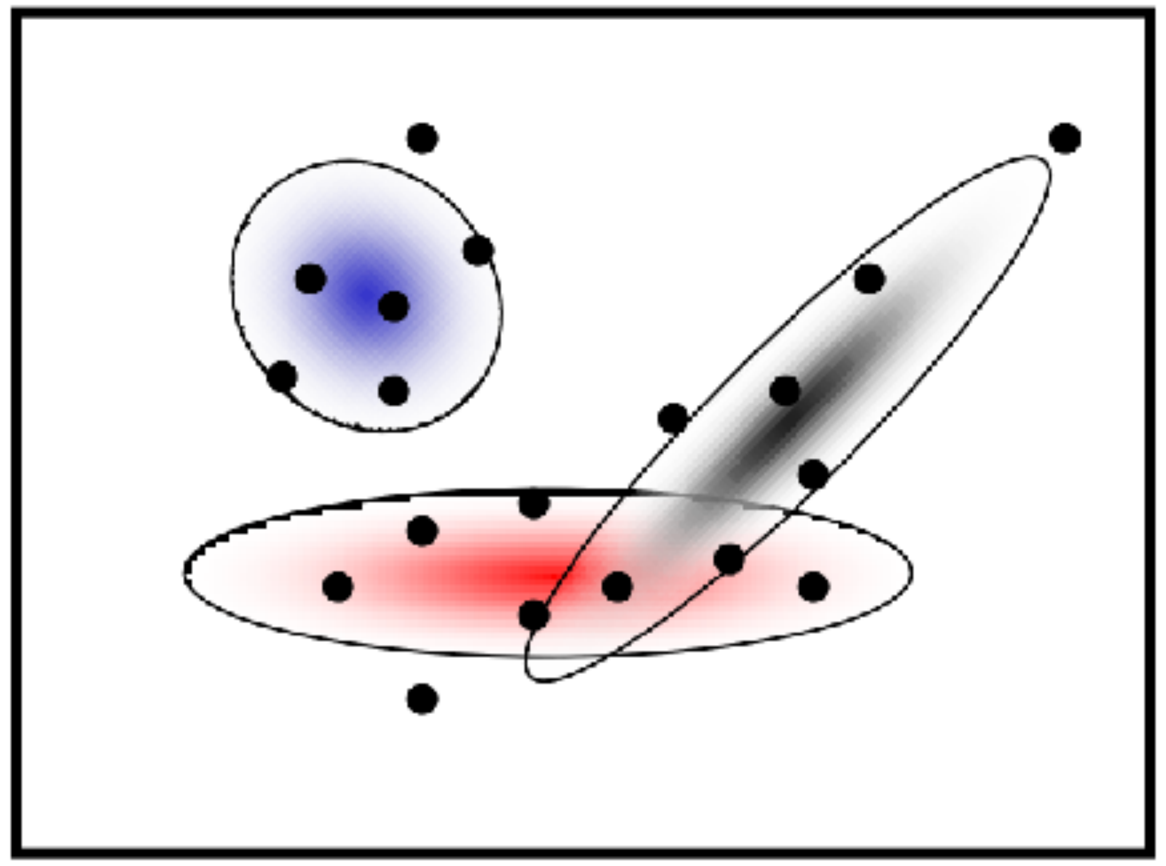
- **Unsupervised learning**: given unlabelled training data $(x_1, \ldots, x_n)$



- **Chicken-and-egg problem**: if we have Θ, we can compute $y = q(x \mid \Theta)$; if we have y, we can compute Θ
- Sounds familiar?

slide credit: Václav Hlaváč, Bastian Leibe

Mixture of Gaussians



- Mixture of Gaussians is one generative model:
  - K Gaussian blobs with means μ_j, cov. matrices Σ_j, dimensionality D

$$p(x \mid \Theta_j) = \frac{1}{(2\pi)^{D/2} \mid \Sigma_j \mid^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_j)^T \Sigma_j^{-1}(x-\mu_j)\right)$$

  - Gaussian j selected with probability π_j
  - Likelihood of observing data point **x** is weighted mixture of Gaussians:

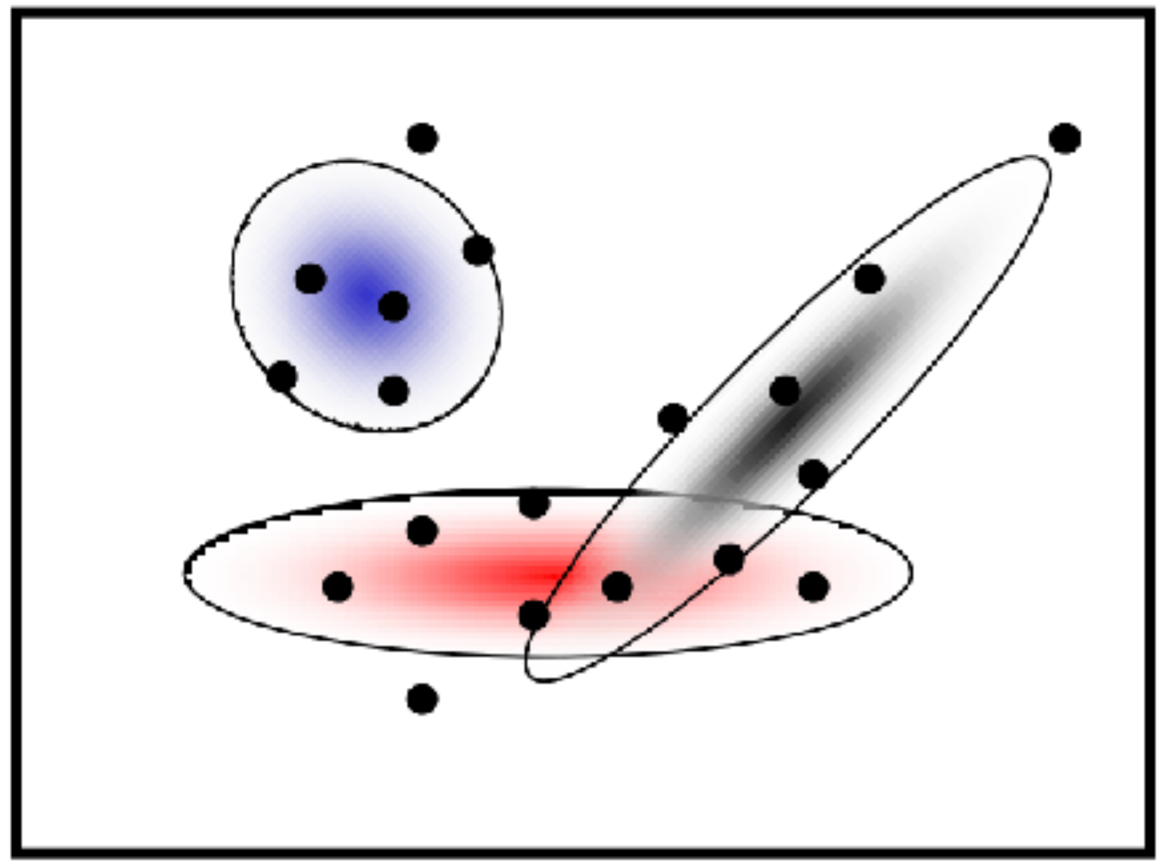
$$p(x \mid \Theta) = \sum_{j=1}^{K} \pi_j p(x \mid \Theta_j) \quad \Theta = \left(\pi_1, \mu_1, \Sigma_1, \ldots, \pi_K, \mu_K, \Sigma_K\right)$$

slide credit: Bastian Leibe

Expectation Maximization (EM) Algorithm

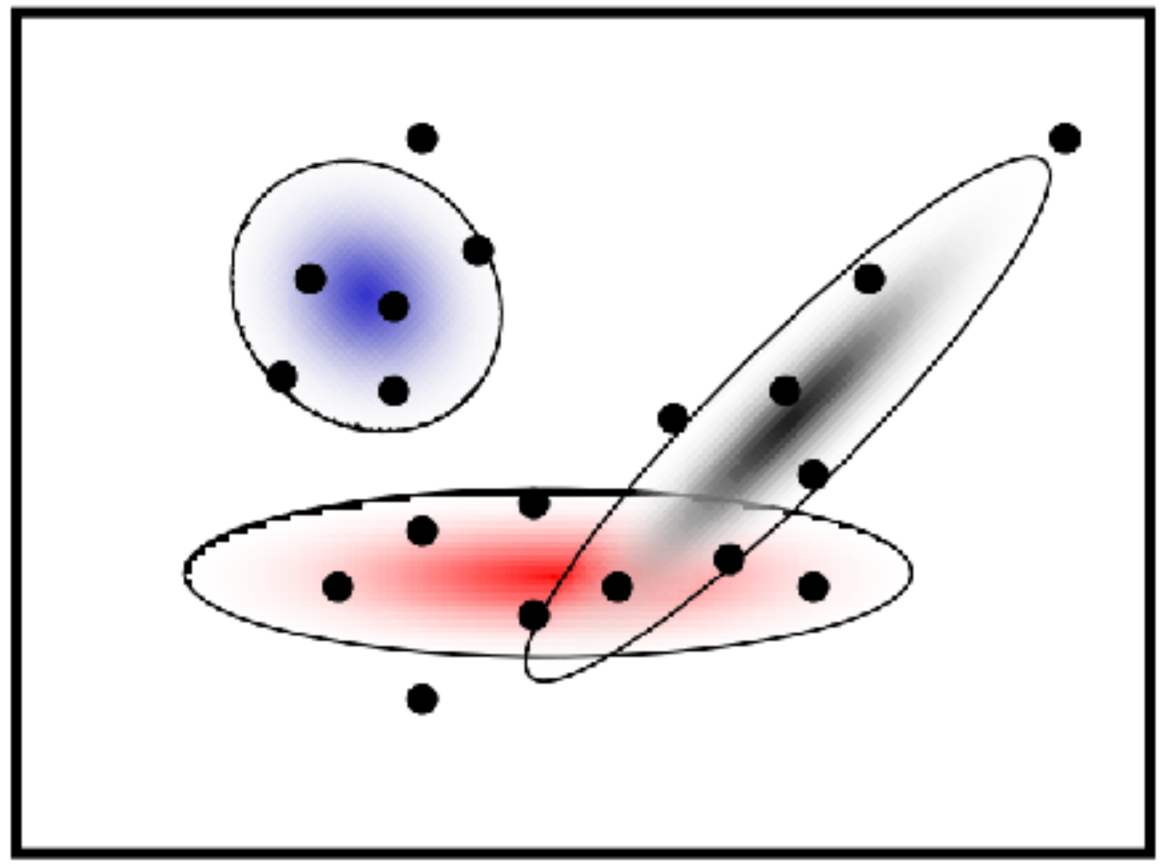
- Goal: find parameters Θ that maximize likelihood function:

$$p(\text{data} \mid \Theta) = \prod_{i=1}^{n} p(x_i \mid \Theta)$$



slide credit: Bastian Leibe, Steve Seitz

Expectation Maximization (EM) Algorithm



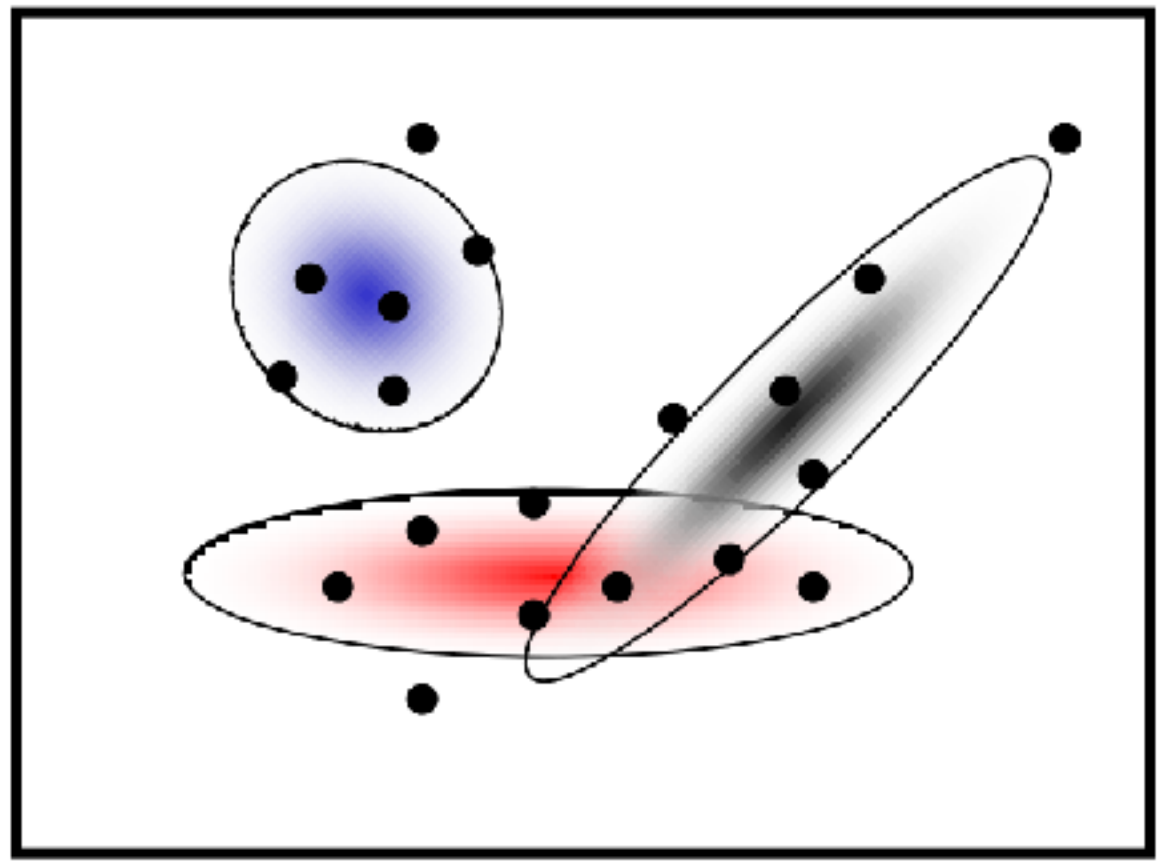
- Goal: find parameters Θ that maximize likelihood function:

$$p(\text{data} \mid \Theta) = \prod_{i=1}^{n} p(x_i \mid \Theta)$$

- **Expectation Maximization** (EM) approach:

slide credit: Bastian Leibe, Steve Seitz

Expectation Maximization (EM) Algorithm



- Goal: find parameters Θ that maximize likelihood function:

$$p(\text{data} \mid \Theta) = \prod_{i=1}^{n} p(x_i \mid \Theta)$$

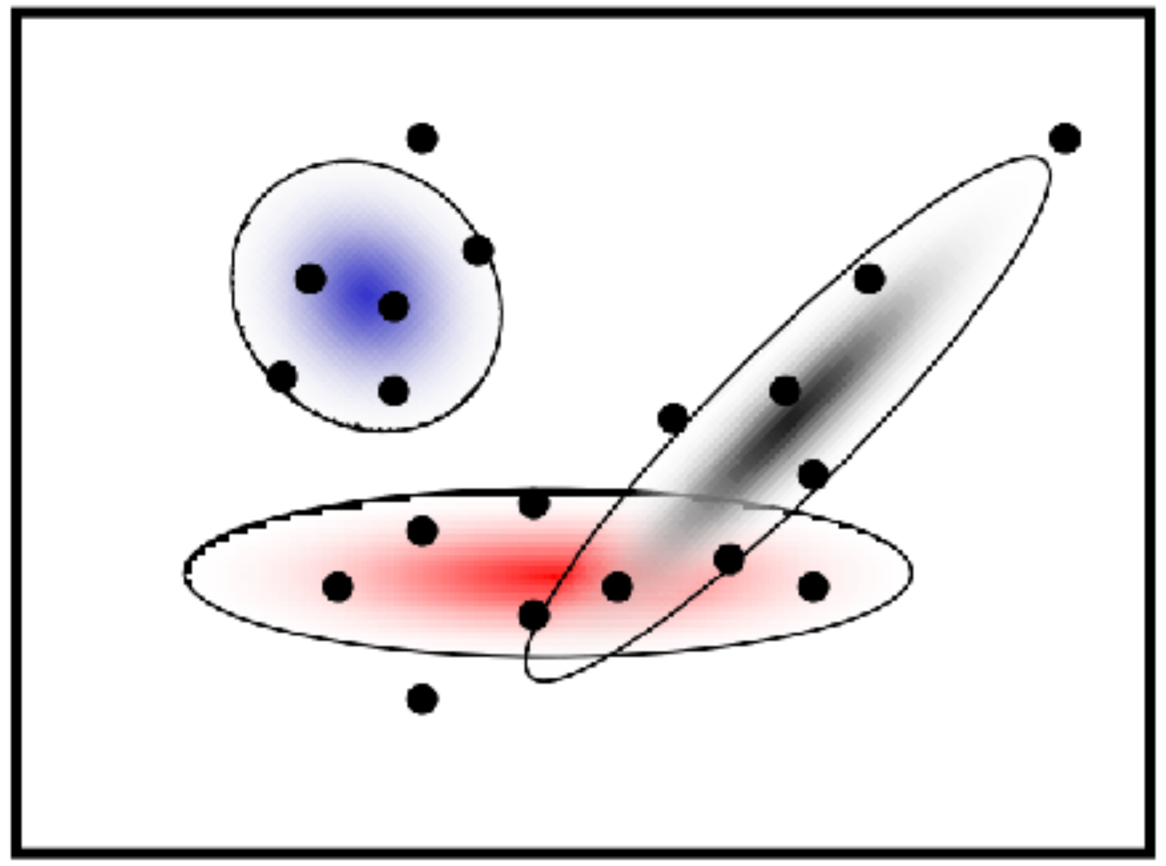
- **Expectation Maximization** (EM) approach:
  - Obtain initial estimate Θ^0 for Θ

slide credit: Bastian Leibe, Steve Seitz

Expectation Maximization (EM) Algorithm

- Goal: find parameters Θ that maximize likelihood function:

$$p(\text{data} \mid \Theta) = \prod_{i=1}^{n} p(x_i \mid \Theta)$$



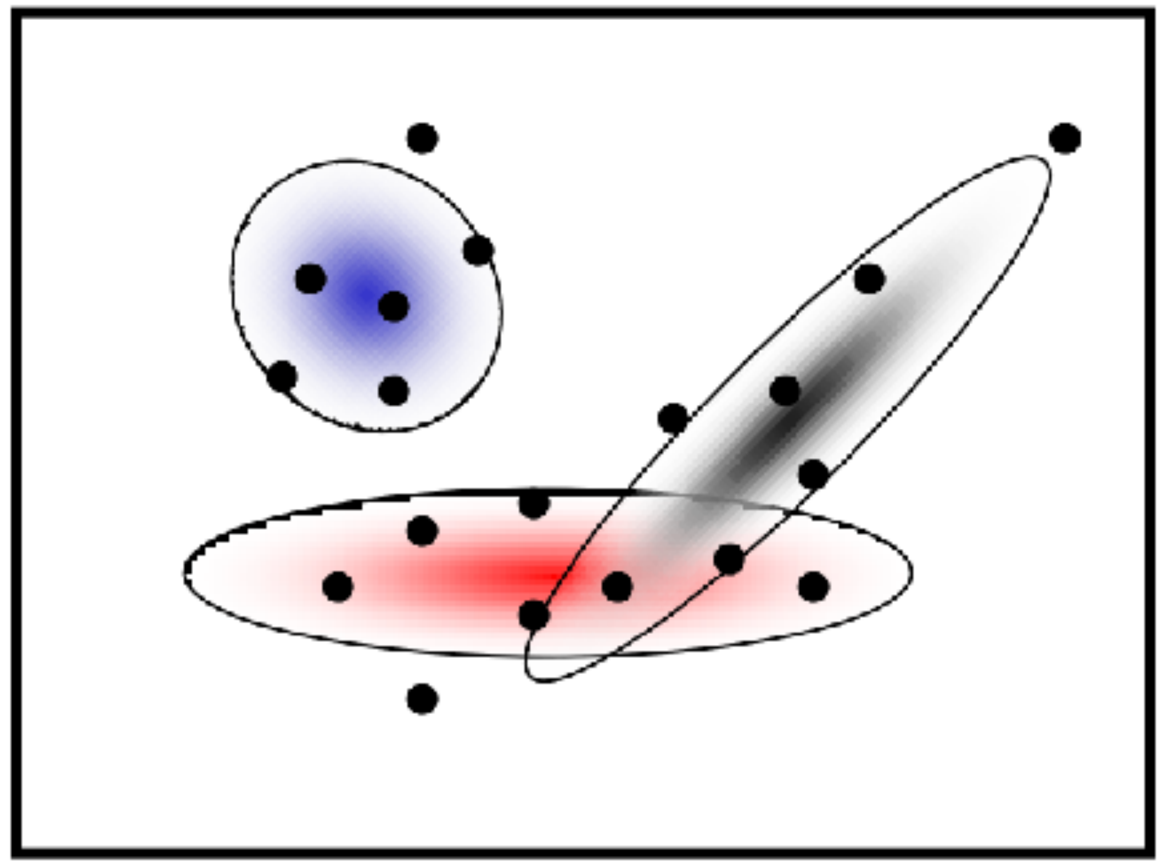
- **Expectation Maximization** (EM) approach:
  - Obtain initial estimate Θ^0 for Θ
  - Repeat:

slide credit: Bastian Leibe, Steve Seitz

Expectation Maximization (EM) Algorithm

- Goal: find parameters Θ that maximize likelihood function:

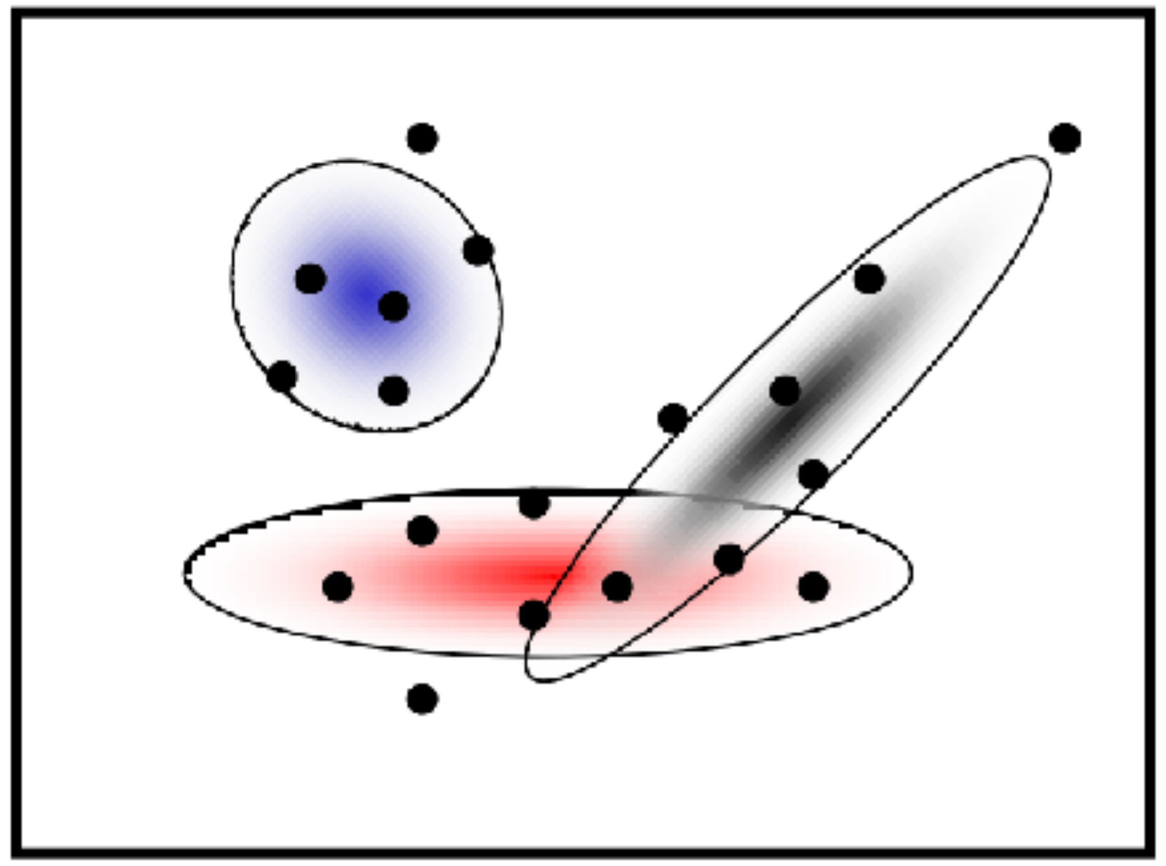
$$p(\text{data} \mid \Theta) = \prod_{i=1}^{n} p(x_i \mid \Theta)$$



- **Expectation Maximization** (EM) approach:
  - Obtain initial estimate Θ^0 for Θ
  - Repeat:
    - **E-step**: given Θ^i assign data points to Gaussians

slide credit: Bastian Leibe, Steve Seitz

Expectation Maximization (EM) Algorithm



- Goal: find parameters Θ that maximize likelihood function:

$$p(\text{data} \mid \Theta) = \prod_{i=1}^{n} p(x_i \mid \Theta)$$

- **Expectation Maximization** (EM) approach:
  - Obtain initial estimate Θ^0 for Θ
  - Repeat:
    - **E-step**: given Θ^i assign data points to Gaussians
    - **M-step**: given assignments, estimate Θ^{i+1} by maximizing likelihood function

slide credit: Bastian Leibe, Steve Seitz

Expectation Maximization (EM) Algorithm

- **E-step**: compute soft assignment of data points to mixture components:

$$\gamma_j(x_i) = \frac{\pi_j \mathcal{N}(x_i \mid \mu_j, \Sigma_j)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i \mid \mu_k, \Sigma_k)}$$

slide credit: Bastian Leibe

Expectation Maximization (EM) Algorithm

- **E-step**: compute soft assignment of data points to mixture components:

$$\gamma_j(x_i) = \frac{\pi_j \mathcal{N}(x_i \mid \mu_j, \Sigma_j)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i \mid \mu_k, \Sigma_k)}$$

- **M-step**: use soft assignments to re-estimate parameters Θ_j per component:

slide credit: Bastian Leibe

Expectation Maximization (EM) Algorithm

- **E-step**: compute soft assignment of data points to mixture components:

$$\gamma_j(x_i) = \frac{\pi_j \mathcal{N}(x_i \mid \mu_j, \Sigma_j)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i \mid \mu_k, \Sigma_k)}$$

- **M-step**: use soft assignments to re-estimate parameters Θ_j per component:

$N_j = \sum_{i=1}^{n} \gamma_j(x_i)$ = soft number of points assigned to cluster j

slide credit: Bastian Leibe

Expectation Maximization (EM) Algorithm

- **E-step**: compute soft assignment of data points to mixture components:

$$\gamma_j(x_i) = \frac{\pi_j \mathcal{N}(x_i \,|\, \mu_j, \Sigma_j)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i \,|\, \mu_k, \Sigma_k)}$$

- **M-step**: use soft assignments to re-estimate parameters Θ_j per component:

$$N_j = \sum_{i=1}^{n} \gamma_j(x_i) = \text{soft number of points assigned to cluster } j$$

$$\pi_j^{\text{new}} = N_j / n = \text{probability of component } j$$

slide credit: Bastian Leibe

Expectation Maximization (EM) Algorithm

- **E-step**: compute soft assignment of data points to mixture components:

$$\gamma_j(x_i) = \frac{\pi_j \mathcal{N}(x_i \mid \mu_j, \Sigma_j)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i \mid \mu_k, \Sigma_k)}$$

- **M-step**: use soft assignments to re-estimate parameters Θ_j per component:

$N_j = \sum_{i=1}^{n} \gamma_j(x_i)$ = soft number of points assigned to cluster j

$\pi_j^{\text{new}} = N_j / n$ = probability of component j

$\mu_j^{\text{new}} = \frac{1}{N_j} \sum_{i=1}^{n} \gamma_j(x_i) x_i$ = new cluster center

slide credit: Bastian Leibe

Expectation Maximization (EM) Algorithm

- **E-step**: compute soft assignment of data points to mixture components:

$$\gamma_j(x_i) = \frac{\pi_j \mathcal{N}(x_i \mid \mu_j, \Sigma_j)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i \mid \mu_k, \Sigma_k)}$$

- **M-step**: use soft assignments to re-estimate parameters Θ_j per component:

$$N_j = \sum_{i=1}^{n} \gamma_j(x_i)$$ = soft number of points assigned to cluster j

$$\pi_j^{\text{new}} = N_j / n$$ = probability of component j

$$\mu_j^{\text{new}} = \frac{1}{N_j} \sum_{i=1}^{n} \gamma_j(x_i) x_i$$ = new cluster center

$$\Sigma_j^{\text{new}} = \frac{1}{N_j} \sum_{i=1}^{n} \gamma_j(x_i)(x_i - \mu_j^{\text{new}})^T (x_i - \mu_j^{\text{new}})$$

slide credit: Bastian Leibe

k-Means as Expectation Maximization

- **k-means clustering** is a special case of EM algorithm

slide credit: Václav Hlaváč

k-Means as Expectation Maximization

- **k-means clustering** is a special case of EM algorithm
- Gaussian mixture model with unit covariances

slide credit: Václav Hlaváč

k-Means as Expectation Maximization

- **k-means clustering** is a special case of EM algorithm
- Gaussian mixture model with unit covariances

- **E-step**: hard assignments to component:

$$k_i = \mathrm{argmax}_{k'} p(x_i \,|\, \Theta_{k'}) = \mathrm{argmax}_{k'} \log\left(\exp\left(-\frac{1}{2}(x_i - \mu_{k'})^T (x_i - \mu_{k'})\right)\right) = \mathrm{argmin}_{k'} \|x_i - \mu_{k'}\|^2$$

slide credit: Václav Hlaváč

k-Means as Expectation Maximization

- **k-means clustering** is a special case of EM algorithm
- Gaussian mixture model with unit covariances
- **E-step**: hard assignments to component:

$$k_i = \text{argmax}_{k'} p(x_i \mid \Theta_{k'}) = \text{argmax}_{k'} \log \left(\exp \left(-\frac{1}{2}(x_i - \mu_{k'})^T (x_i - \mu_{k'}) \right) \right) = \text{argmin}_{k'} \|x_i - \mu_{k'}\|^2$$

- **M-step**: update cluster centers $\Theta = (\mu_1, \ldots, \mu_K)$:

slide credit: Václav Hlaváč

k-Means as Expectation Maximization

- **k-means clustering** is a special case of EM algorithm
- Gaussian mixture model with unit covariances
- **E-step**: hard assignments to component:

$$k_i = \text{argmax}_{k'} p(x_i \mid \Theta_{k'}) = \text{argmax}_{k'} \log \left(\exp \left(-\frac{1}{2}(x_i - \mu_{k'})^T (x_i - \mu_{k'}) \right) \right) = \text{argmin}_{k'} \|x_i - \mu_{k'}\|^2$$

- **M-step**: update cluster centers $\Theta = (\mu_1, \ldots, \mu_K)$:

$$\Theta^* = \text{argmax}_{\Theta} \sum_{i=1}^{n} \log(p(x_i \mid \Theta_{k_i})$$

slide credit: Václav Hlaváč

k-Means as Expectation Maximization

- **k-means clustering** is a special case of EM algorithm
- Gaussian mixture model with unit covariances
- **E-step**: hard assignments to component:

$$k_i = \text{argmax}_{k'} p(x_i \mid \Theta_{k'}) = \text{argmax}_{k'} \log \left(\exp \left(-\frac{1}{2}(x_i - \mu_{k'})^T (x_i - \mu_{k'}) \right) \right) = \text{argmin}_{k'} \|x_i - \mu_{k'}\|^2$$

- **M-step**: update cluster centers $\Theta = (\mu_1, \ldots, \mu_K)$:

$$\Theta^* = \text{argmax}_{\Theta} \sum_{i=1}^{n} \log(p(x_i \mid \Theta_{k_i})) = \text{argmin}_{\mu_1,\ldots,\mu_k} \sum_{i=1}^{n} \|x_i - \mu_{k_i}\|^2$$

slide credit: Václav Hlaváč

k-Means as Expectation Maximization

- **k-means clustering** is a special case of EM algorithm
- Gaussian mixture model with unit covariances
- **E-step**: hard assignments to component:

$$k_i = \text{argmax}_{k'} p(x_i \mid \Theta_{k'}) = \text{argmax}_{k'} \log \left(\exp \left(-\frac{1}{2}(x_i - \mu_{k'})^T (x_i - \mu_{k'}) \right) \right) = \text{argmin}_{k'} \|x_i - \mu_{k'}\|^2$$

- **M-step**: update cluster centers $\Theta = (\mu_1, \ldots, \mu_K)$:

$$\Theta^* = \text{argmax}_{\Theta} \sum_{i=1}^{n} \log(p(x_i \mid \Theta_{k_i}) = \text{argmin}_{\mu_1,\ldots,\mu_k} \sum_{i=1}^{n} \|x_i - \mu_{k_i}\|^2$$

$$= \text{argmin}_{\mu_1} \sum_{\{i \mid k_i = 1\}} \|x_i - \mu_1\|^2, \ldots, \text{argmin}_{\mu_K} \sum_{\{i \mid k_i = K\}} \|x_i - \mu_K\|^2$$

slide credit: Václav Hlaváč

k-Means as Expectation Maximization

- **k-means clustering** is a special case of EM algorithm
- Gaussian mixture model with unit covariances
- **E-step**: hard assignments to component:

$$k_i = \text{argmax}_{k'} p(x_i \mid \Theta_{k'}) = \text{argmax}_{k'} \log\left(\exp\left(-\frac{1}{2}(x_i - \mu_{k'})^T(x_i - \mu_{k'})\right)\right) = \text{argmin}_{k'} \|x_i - \mu_{k'}\|^2$$

- **M-step**: update cluster centers $\Theta = (\mu_1, \ldots, \mu_K)$:

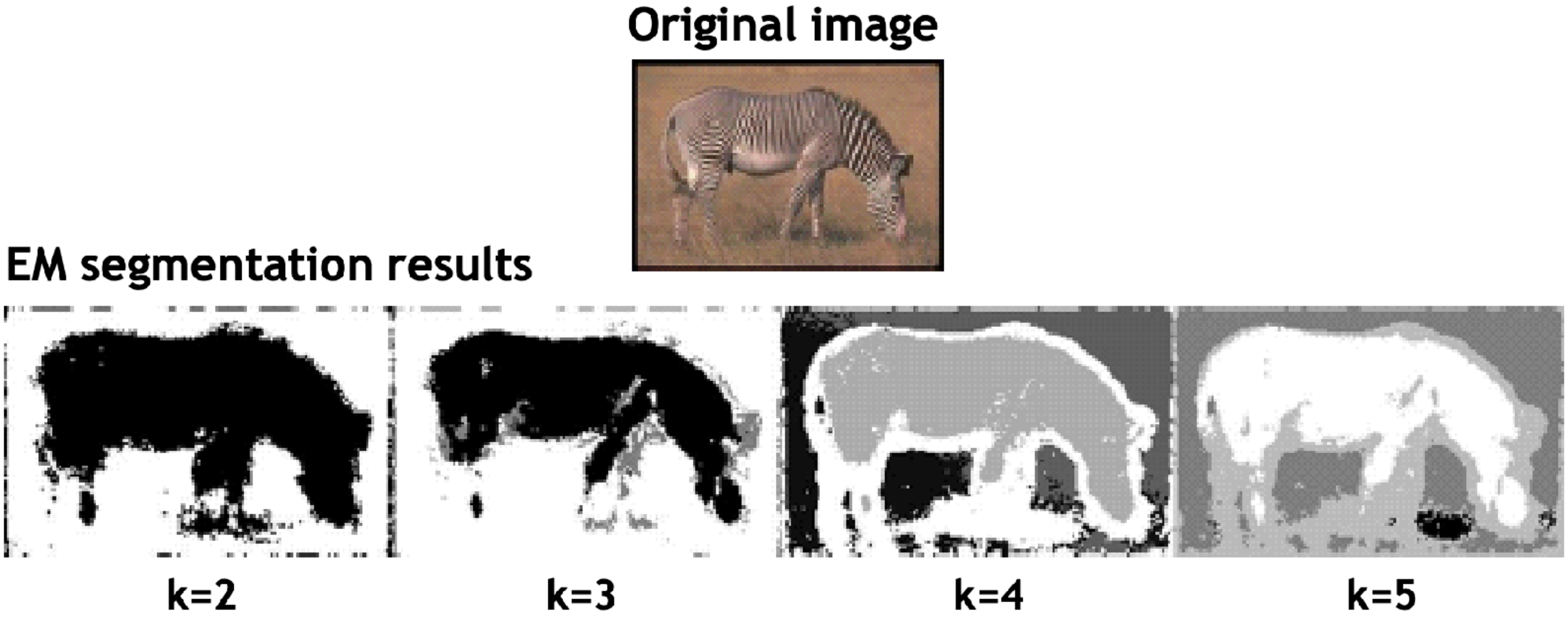
$$\Theta^* = \text{argmax}_\Theta \sum_{i=1}^{n} \log(p(x_i \mid \Theta_{k_i}) = \text{argmin}_{\mu_1,\ldots,\mu_k} \sum_{i=1}^{n} \|x_i - \mu_{k_i}\|^2$$

$$= \text{argmin}_{\mu_1} \sum_{\{i|k_i=1\}} \|x_i - \mu_1\|^2, \ldots, \text{argmin}_{\mu_K} \sum_{\{i|k_i=K\}} \|x_i - \mu_K\|^2$$

$$\Rightarrow \mu_j = \frac{1}{|\{i \mid k_i = j\}|} \sum_{\{i|k_i=j\}} x_i, \quad j = 1,\ldots,K$$

slide credit: Václav Hlaváč

EM Algorithm for Segmentation



slide credit: Bastian Leibe

image credit: Serge Belongie

The EM Algorithm

- General statistical approach for missing data / data with hidden states

slide credit: Václav Hlaváč

The EM Algorithm

- General statistical approach for missing data / data with hidden states
- Can be used to obtain **Maximum Likelihood Estimate** (MLE) even if MLE cannot be computed directly

slide credit: Václav Hlaváč

The EM Algorithm

- General statistical approach for missing data / data with hidden states
- Can be used to obtain **Maximum Likelihood Estimate** (MLE) even if MLE cannot be computed directly
- **General concept**:
  - Marginalize over hidden states $y \in Y$: $p(x \mid \Theta) = \sum_y p(x, y \mid \Theta)$
  - Simplify estimation of $p(x \mid \Theta)$ by inferring hidden states

slide credit: Václav Hlaváč

The EM Algorithm

- General statistical approach for missing data / data with hidden states
- Can be used to obtain **Maximum Likelihood Estimate** (MLE) even if MLE cannot be computed directly
- **General concept**:
  - Marginalize over hidden states $y \in Y$: $p(x \mid \Theta) = \sum_y p(x, y \mid \Theta)$
  - Simplify estimation of $p(x \mid \Theta)$ by inferring hidden states
- **Guaranteed to converge** as cost function decreases monotonically (proof via special form of Jensen's inequality)

slide credit: Václav Hlaváč

The EM Algorithm

- General statistical approach for missing data / data with hidden states
- Can be used to obtain **Maximum Likelihood Estimate** (MLE) even if MLE cannot be computed directly
- **General concept**:
  - Marginalize over hidden states $y \in Y$: $p(x \mid \Theta) = \sum_y p(x, y \mid \Theta)$
  - Simplify estimation of $p(x \mid \Theta)$ by inferring hidden states
- **Guaranteed to converge** as cost function decreases monotonically (proof via special form of Jensen's inequality)
- **No general guarantees about global optimality**, EM is essentially gradient ascent

slide credit: Václav Hlaváč

EM Algorithm Maximizes Lower Bound on Likelihood

- Starting from initial estimate Θ^0, iterate:

slide credit: Václav Hlaváč

EM Algorithm Maximizes Lower Bound on Likelihood

- Starting from initial estimate Θ^0, iterate:
  - **E-step**: estimate lower bound of the likelihood function $L(\Theta)$ at point Θ^t

$$L(\Theta) = p(\text{data} \mid \Theta) = \prod_i p(x_i \mid Theta) = \prod_i \sum_y p(x_i, y \mid \Theta)$$

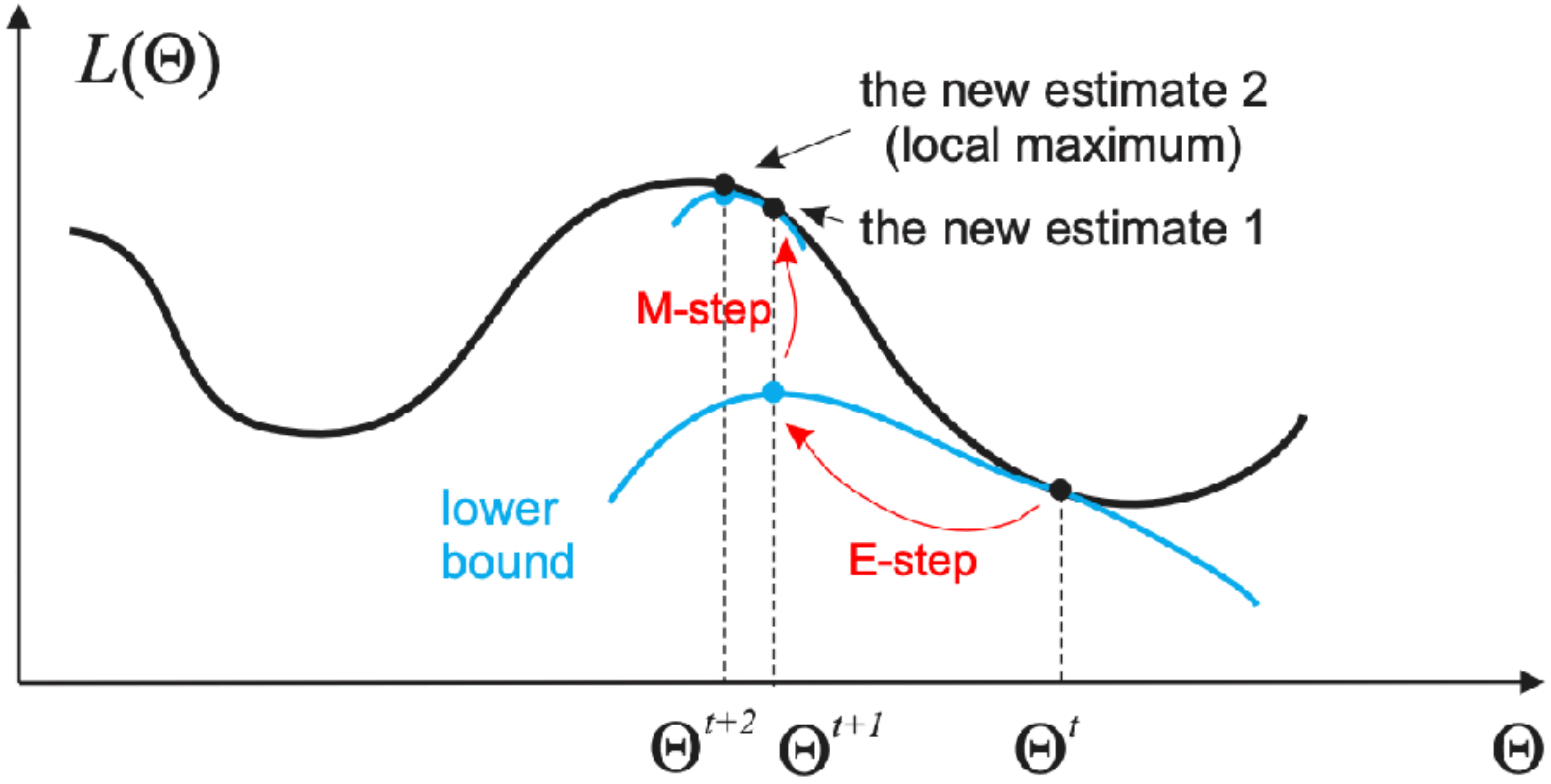
slide credit: Václav Hlaváč

EM Algorithm Maximizes Lower Bound on Likelihood

- Starting from initial estimate Θ^0, iterate:
  - **E-step**: estimate lower bound of the likelihood function $L(\Theta)$ at point Θ^t

$$L(\Theta) = p(\text{data} \mid \Theta) = \prod_i p(x_i \mid Theta) = \prod_i \sum_y p(x_i, y \mid \Theta)$$

  - **M-step**: estimate Θ^{t+1} that maximizes lower bound



slide credit: Václav Hlaváč

The EM Algorithm

- **Pros**:
  - Probabilistic interpretation of data
  - Soft assignments instead of hard assignments
  - Generative model: can predict new datapoint

slide credit: Bastian Leibe

image credit: Serge Belongie

The EM Algorithm

- **Pros**:
  - Probabilistic interpretation of data
  - Soft assignments instead of hard assignments
  - Generative model: can predict new datapoint
- **Cons**:
  - Local optimization will lead to local minima
  - Initialization is thus important (e.g., use k-means for initialization)
  - Similar to k-means clustering: need estimate for K
  - Need to choose proper generative model
  - Numerical instabilities can be an issue

slide credit: Bastian Leibe

image credit: Serge Belongie