Computer Vision

(for Autonomous Driving)

Raoul de Charette

INVENTEURS DU MONDE NUMÉRIQUE

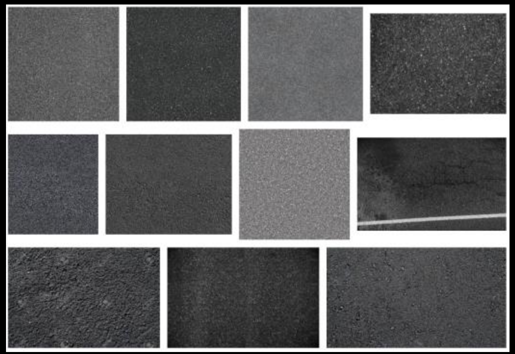
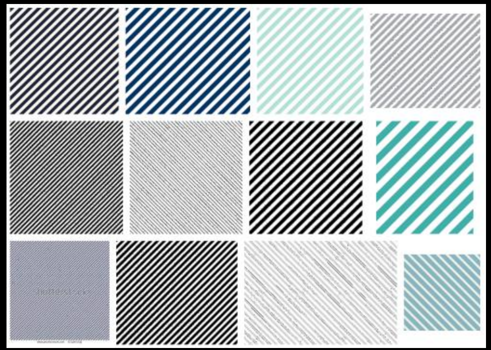
UNIVERSIDAD SIMÓN BOLÍVAR

Texture segmentation

Data representation

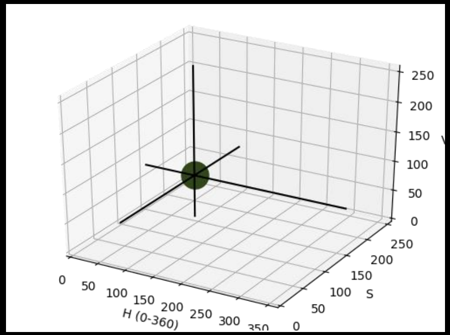
- Find the richest most compact representation
    - Texture, Chromacity, Motion, Frequency, Entropy, etc.

- Solution is data-dependent

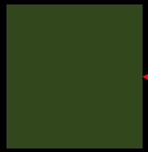




- Usually dealing with low saliency data

Single point model



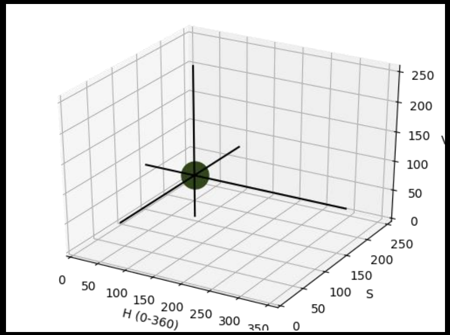
Single point model



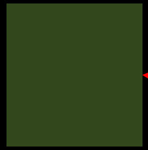
RGB [50 71 28]
HSV [90 154 71]

- Naïve solution: pixel-wise Euclidean distance (obviously fails)

Single point model



Single point model

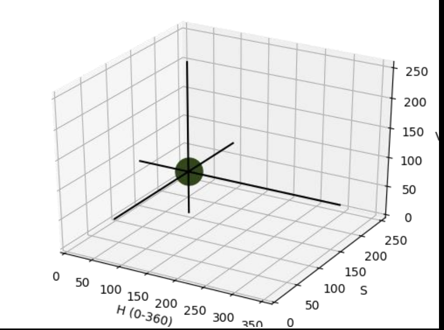


RGB [50 71 28]
HSV [90 154 71]

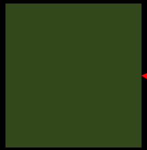
- Naïve solution: pixel-wise Euclidean distance (obviously fails)

1. - RGBEuclideanDistance

Single point model



Single point model



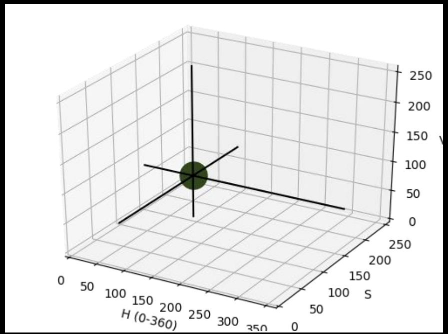
RGB [50 71 28]
HSV [90 154 71]

- Naïve solution: pixel-wise Euclidean distance (obviously fails)

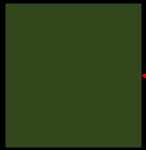
1. - RGBEuclideanDistance

1. - HSVEuclideanDistance

Single point model



Single point model



RGB [50 71 28]
HSV [90 154 71]

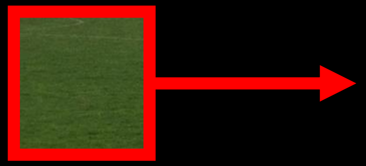
- Naïve solution: pixel-wise Euclidean distance (obviously fails)

1. - RGBEuclideanDistance

1. - HSVEuclideanDistance

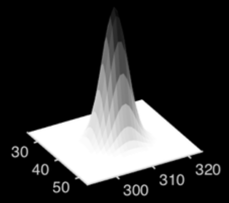
1. - HSEuclideanDistance

Gaussian model



Grass patch

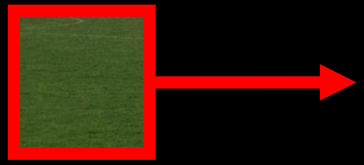
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$



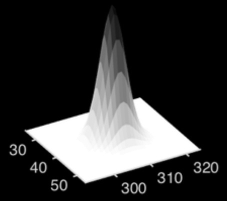
- Models the texture as a Multivariate Normal (3D Gaussian)
  - Advantage: models the signal variance
- **P**robability **D**ensity **F**unction (PDF) allows computation of pixel-wise probability

PDF: $f(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^k|\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$

Gaussian model



$\mathcal{N}(\mu, \sigma)$



- Models the texture as a Multivariate Normal (3D Gaussian)
  - Advantage: models the signal variance
- **P**robability **D**ensity **F**unction (PDF) allows computation of pixel-wise probability

PDF: $f(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^k|\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$

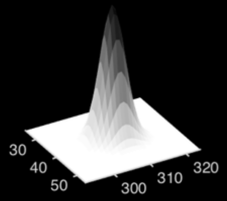
PDF of RGB Normal

PDF of HSV Normal

Gaussian model

Grass patch

$\mathcal{N}(\mu, \sigma)$

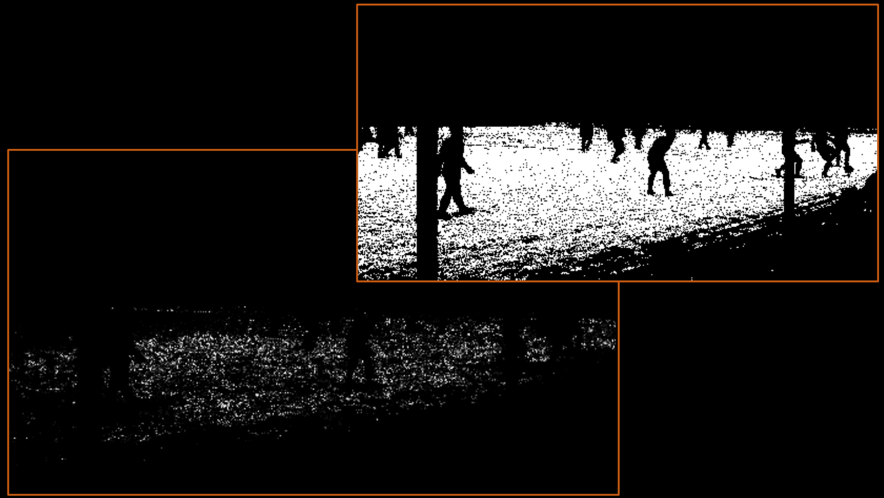


- Models the texture as a Multivariate Normal (3D Gaussian)
  - Advantage: models the signal variance
- **P**robability **D**ensity **F**unction (PDF) allows computation of pixel-wise probability

  PDF: $f(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^k|\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$

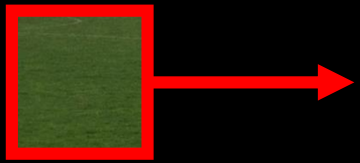
- Why does it work so well ?

PDF of RGB Normal



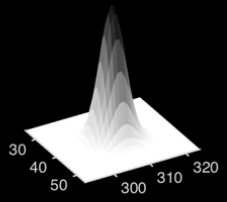
PDF of HSV Normal

Gaussian model



Grass patch

$\mathcal{N}(\mu, \sigma)$

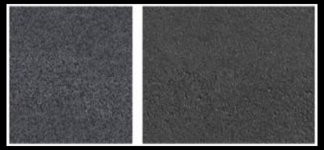


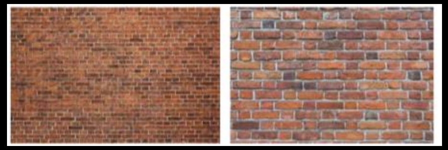
- Models the texture as a Multivariate Normal (3D Gaussian)
  - Advantage: models the signal variance
- **P**robability **D**ensity **F**unction (PDF) allows computation of pixel-wise probability

  PDF: $f(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^k |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})\right)$

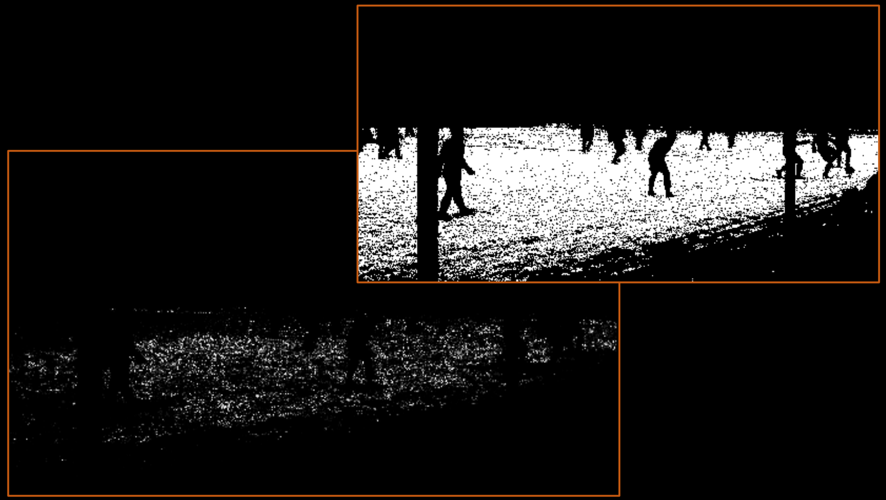
- Why does it work so well ?

- How will it perform to find these:



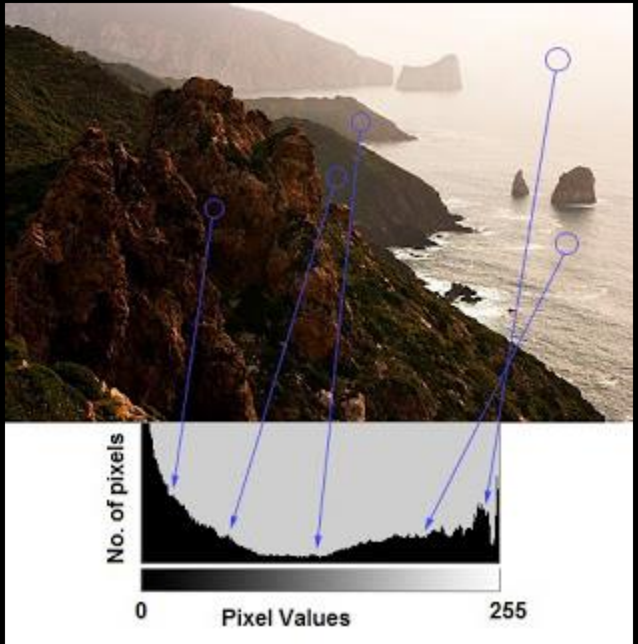


PDF of RGB Normal



PDF of HSV Normal

Histograms

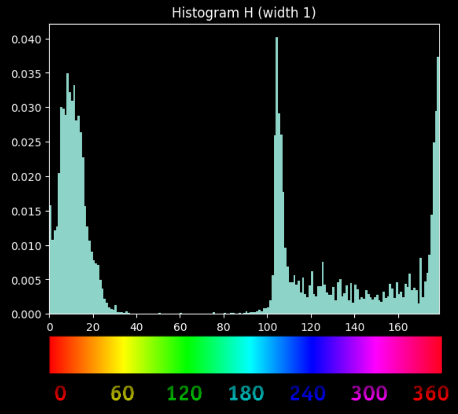


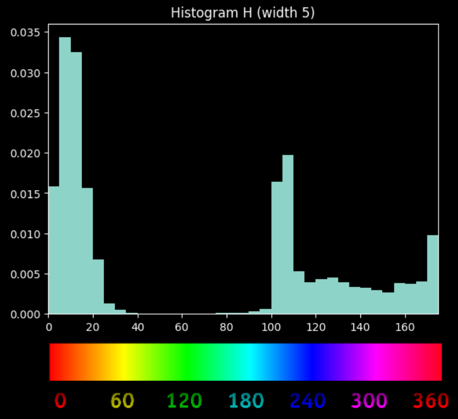
- Building histograms
  - Choose the right dimension-space
  - Usually no more than 2D for histograms

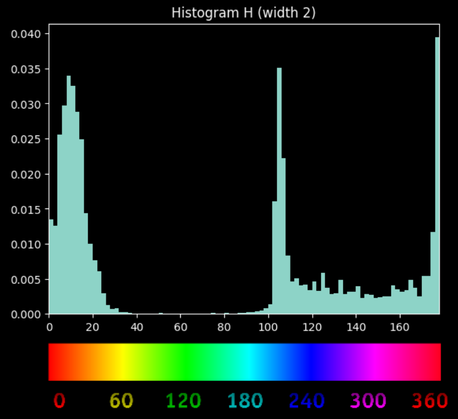
- Let’s consider Histogram $\mathcal{H}(\ldots)$ with h bins
- Bin stores number of occurrences

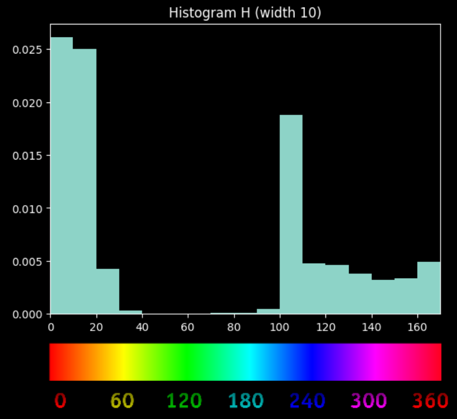
$$\mathcal{H}(\mathbf{x}) = |\{p \; \forall \; p = \mathbf{x}, p \in I\}|$$

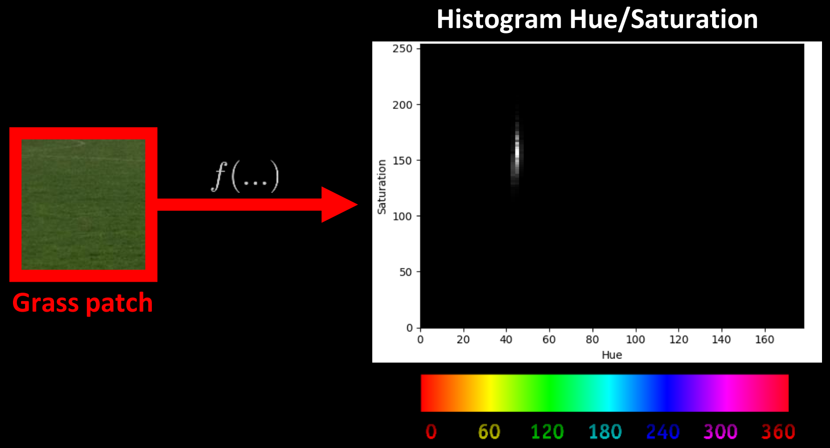
- Size matters for histograms.











- Back projection: likelihood of pixels to image histogram $\mathcal{H}$

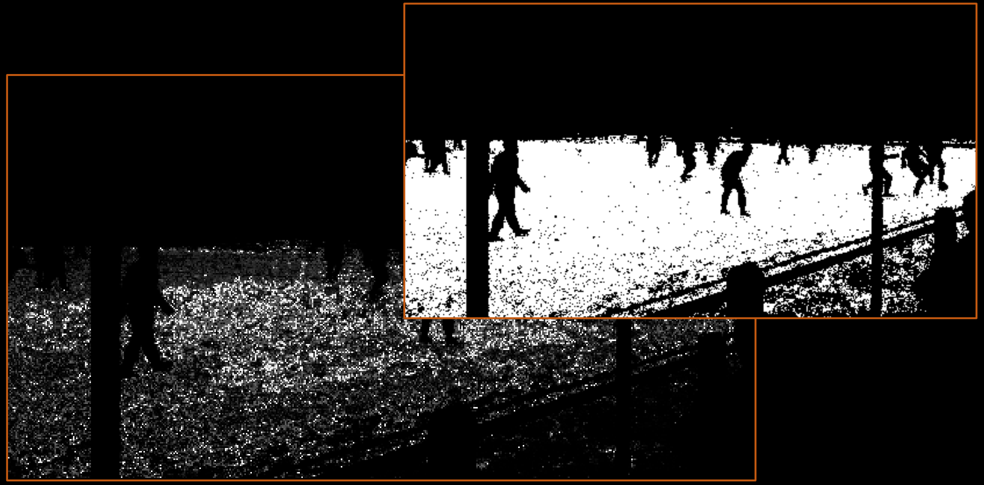
- Assume a normalized histogram: $\sum_{\mathbf{x}} \mathcal{H}(\mathbf{x}) = 1$

  The normalized bins can be considered likelihood

  Hence likelihood of pixel p to belong to histogram is:

  $$\mathcal{H}(\underbrace{f(I(p))}_{\textbf{Mapped value}})$$

- Referred as: probability histogram

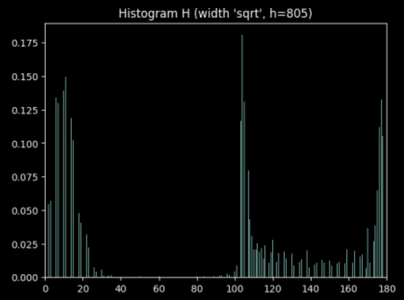


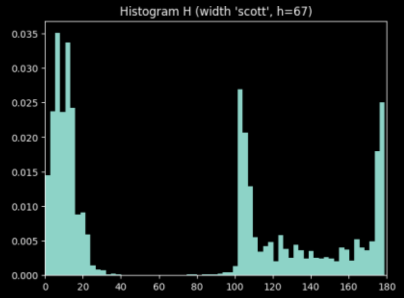
Back projection of grass Histogram

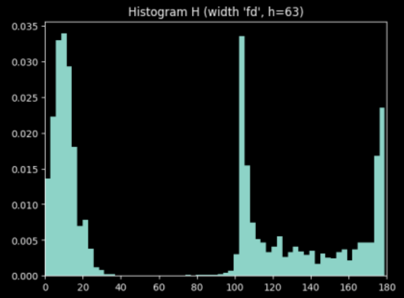
A words on histograms

- Bins can be asymmetrical => generally a bad idea

- Optimal bin size (h)
  - Square root
    $$n_h = \sqrt{n}$$
    Naïve estimator
  - Scott rule
    $$h = \sigma \sqrt[3]{\frac{24 * \sqrt{\pi}}{n}}$$
    Usually considered for large datasets. Not robust to outliers
  - Freedman Diaconis Estimator (FD estimator)
    $$h = 2\frac{IQR}{n^{1/3}}$$
    Using Interquartile Range (IQR). Optimal for large datasets. Robust to outliers.



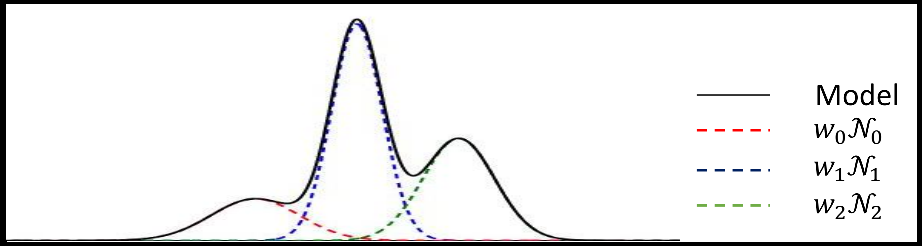




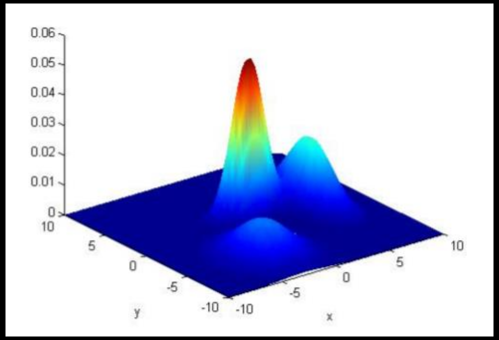
Gaussian Mixture Models (GMM)

- GMM are really useful to represent mix of signals

$$\text{GMM} = \sum_i w_i \mathcal{N}(\mu_i, \sigma_i)$$



1D



2D

- Where the set of data X is supposed to be a partial image of the GMM
  - Models more complex signal

- Problem how to estimate the set of parameters $\{w_i, \mu_i, \sigma_i\}$?
  - Expectation Maximization, k-Means

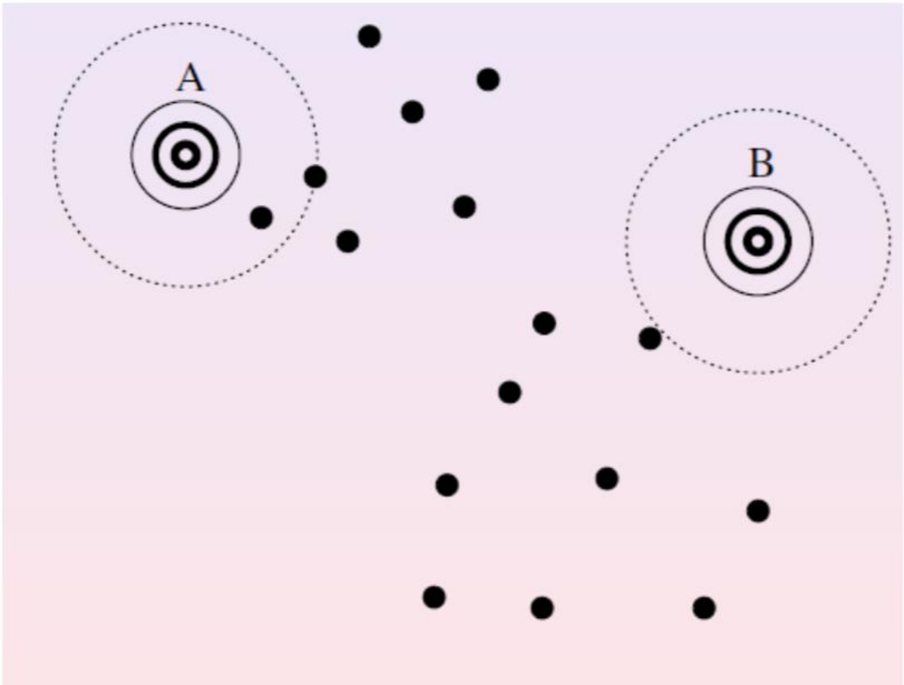
Expectation-Maximization (Arthur P. Dempster)

$\{w_i, \mu_i, \sigma_i\}$ Gaussian Mixture Models (inc. noise)

- Estimates parameters of a statistical models partially observed

- How does it work ?
  - Tries to map a data to N Gaussians through maximization of likelihood
  - Randomly initialize a set of parameters (random Gaussians)
  - E-Step: For each point compute the probability to be generated by Gaussians
  - M-Step: Update parameters to maximize the (log) likelihood of the data

  - Iterate E and M until convergence



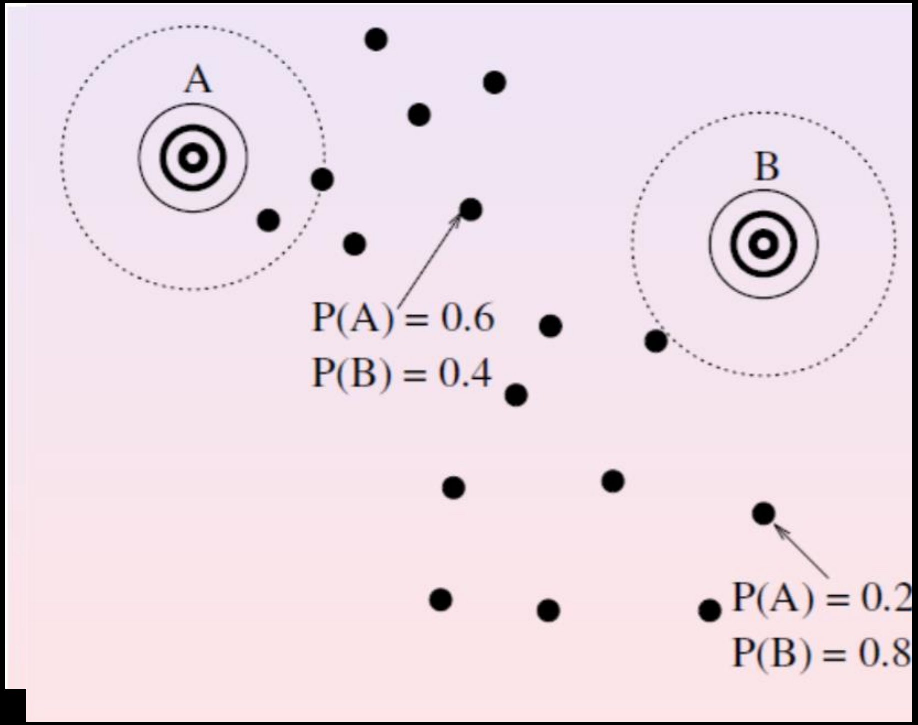
(Mohand Saïd Allili, Tutorial, 2010)

Expectation-Maximization (Arthur P. Dempster)

$\{w_i, \mu_i, \sigma_i\}$ Gaussian Mixture Models (inc. noise)

- Estimates parameters of a statistical models partially observed

- How does it work ?
  - Tries to map a data to N Gaussians through maximization of likelihood
  - Randomly initialize a set of parameters (random Gaussians)
  - E-Step: For each point compute the probability to be generated by Gaussians
  - M-Step: Update parameters to maximize the (log) likelihood of the data
  - Iterate E and M until convergence



(Mohand Saïd Allili, Tutorial, 2010)

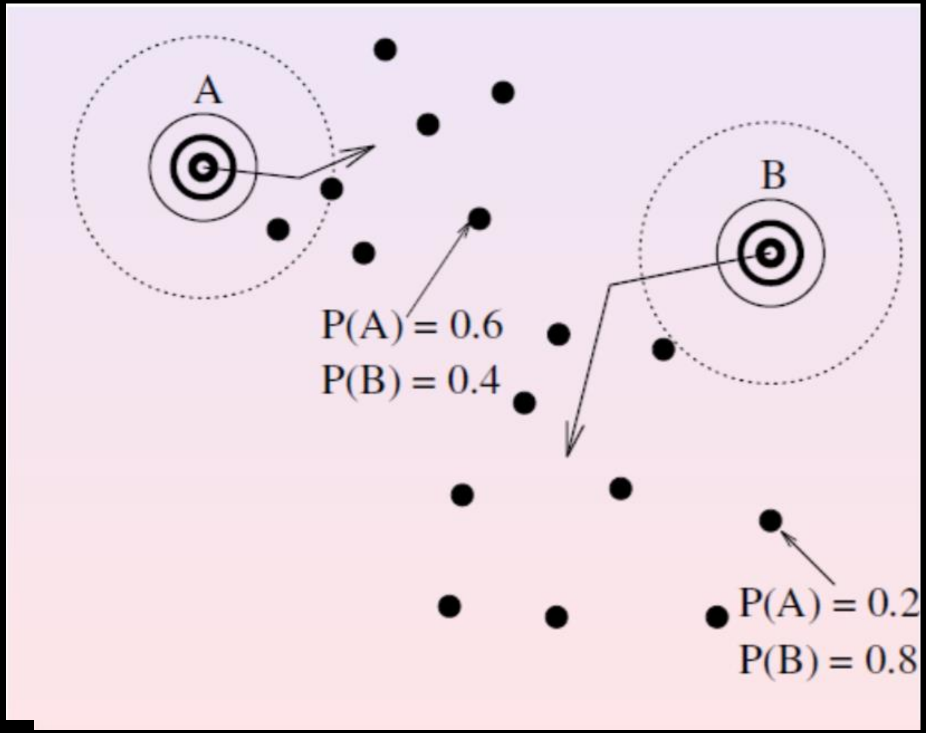
Expectation-Maximization (Arthur P. Dempster)

$\{w_i, \mu_i, \sigma_i\}$ Gaussian Mixture Models (inc. noise)

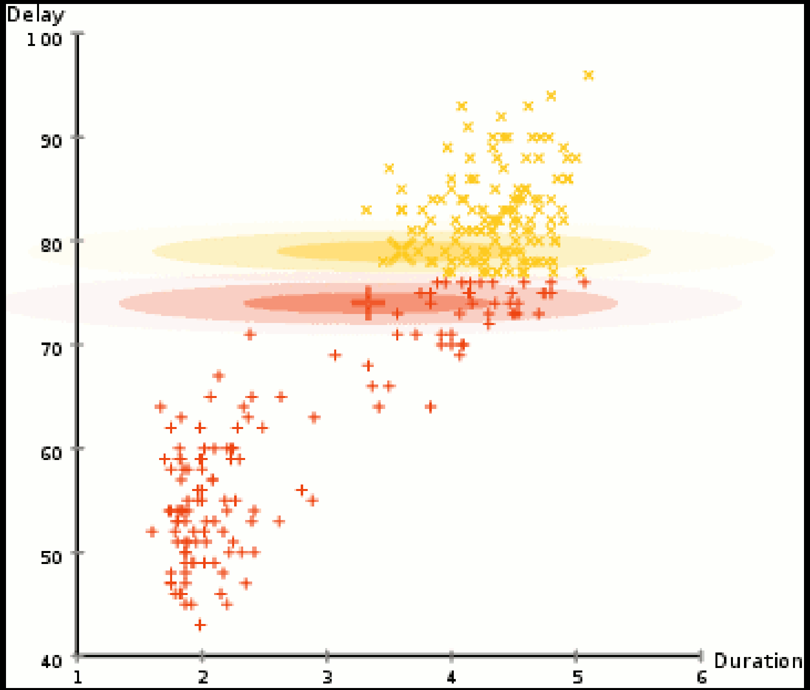
- Estimates parameters of a statistical models partially observed

- How does it work ?
  - Tries to map a data to N Gaussians through maximization of likelihood
  - Randomly initialize a set of parameters (random Gaussians)
  - E-Step: For each point compute the probability to be generated by Gaussians
  - M-Step: Update parameters to maximize the (log) likelihood of the data

  - Iterate E and M until convergence

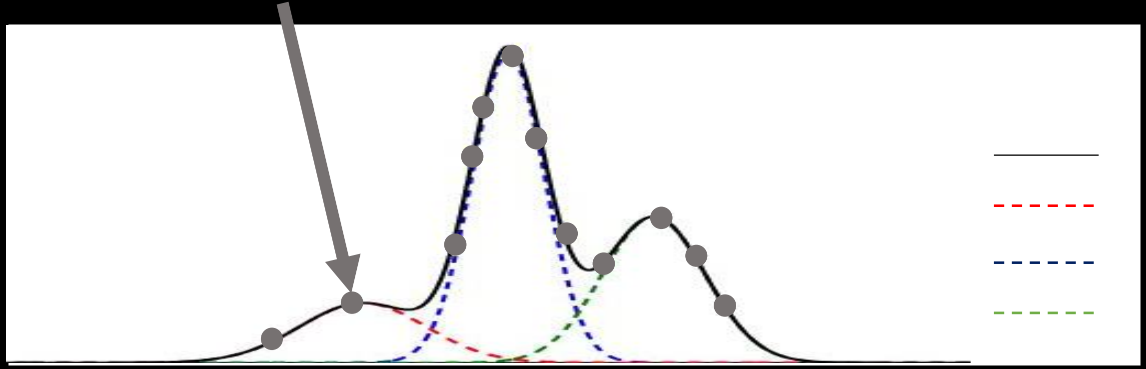


(Mohand Saïd Allili, Tutorial, 2010)



Expectation Maximization (EM)

Which Gaussian does it belong ?



1D

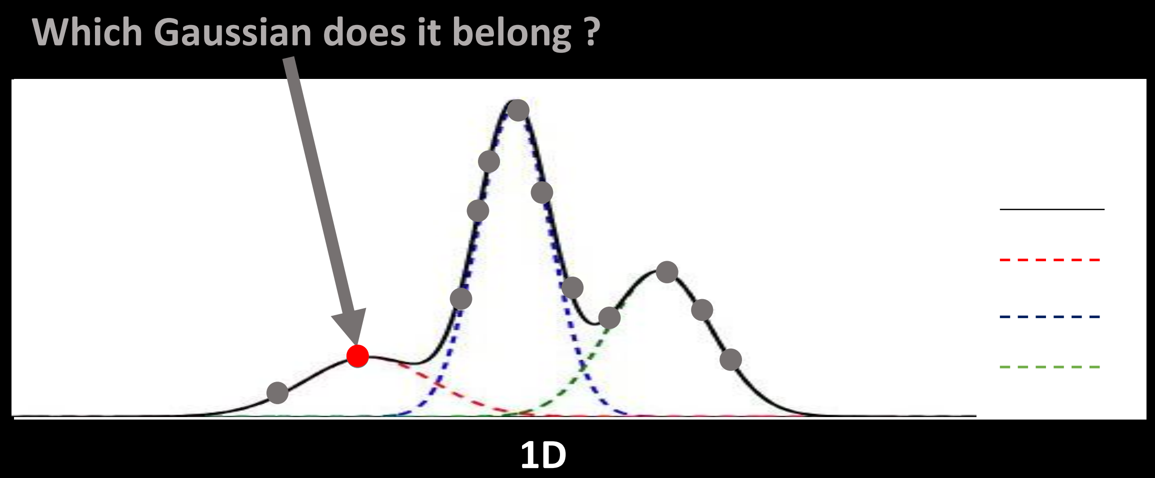
Suppose all parameters estimated $\{w_i, \mu_i, \sigma_i\}$

Assigning a label (“Gaussian”) to each data points is straightforward

- The probability that a point belong to a Gaussian $\mathcal{N}$ is: $\mathrm{p}(\mathrm{x}|\mathcal{N}) = \mathcal{N}_{PDF}(x)$

- Suppose n Gaussians, the class is the Gaussian with highest probability

$$\mathrm{label}(\boldsymbol{x}) = \mathrm{argmax}_i\, p(\boldsymbol{x}|\mathcal{N}_i)$$



Suppose all parameters estimated $\{w_i, \mu_i, \sigma_i\}$

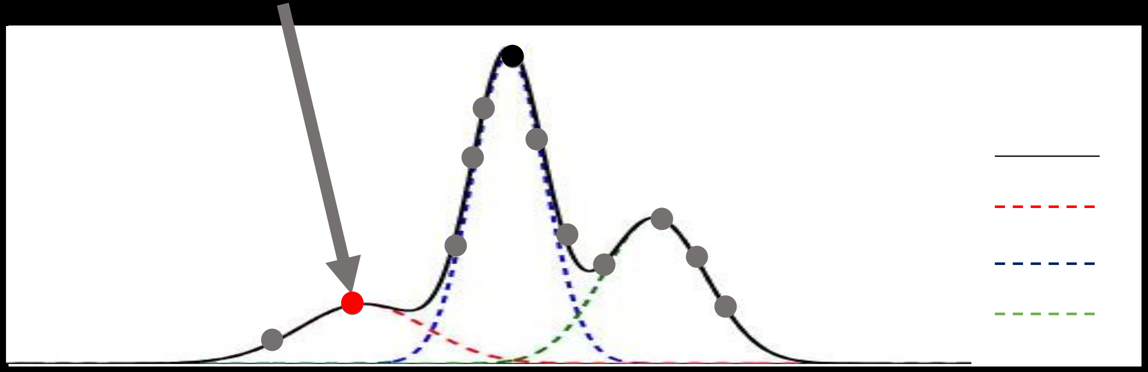
Assigning a label ("Gaussian") to each data points is straightforward

- The probability that a point belong to a Gaussian $\mathcal{N}$ is: $\mathrm{p}(\mathrm{x}|\mathcal{N}) = \mathcal{N}_{PDF}(x)$

- Suppose n Gaussians, the class is the Gaussian with highest probability

$$\mathrm{label}(\boldsymbol{x}) = \mathrm{argmax}_i \, p(\boldsymbol{x}|\mathcal{N}_i)$$

Which Gaussian does it belong ?



1D

Suppose all parameters estimated $\{w_i, \mu_i, \sigma_i\}$

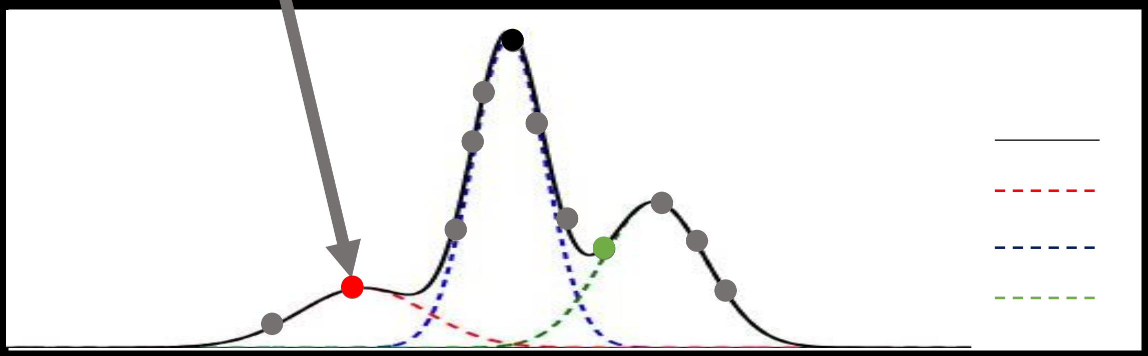
Assigning a label ("Gaussian") to each data points is straightforward

- The probability that a point belong to a Gaussian $\mathcal{N}$ is: $\mathrm{p}(\mathrm{x}|\mathcal{N}) = \mathcal{N}_{PDF}(x)$

- Suppose n Gaussians, the class is the Gaussian with highest probability

$$\mathrm{label}(\boldsymbol{x}) = \mathrm{argmax}_i \, p(\boldsymbol{x}|\mathcal{N}_i)$$

Which Gaussian does it belong ?



1D

Suppose all parameters estimated $\{w_i, \mu_i, \sigma_i\}$

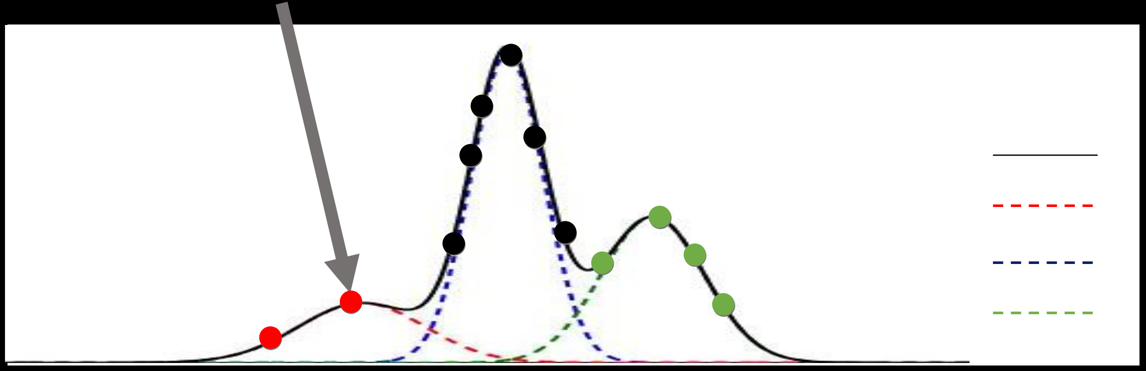
Assigning a label ("Gaussian") to each data points is straightforward

- The probability that a point belong to a Gaussian $\mathcal{N}$ is: $\mathrm{p(x|\mathcal{N})} = \mathcal{N}_{PDF}(x)$

- Suppose n Gaussians, the class is the Gaussian with highest probability

$$\text{label}(\boldsymbol{x}) = \text{argmax}_i\, p(\boldsymbol{x}|\mathcal{N}_i)$$

Which Gaussian does it belong ?



1D

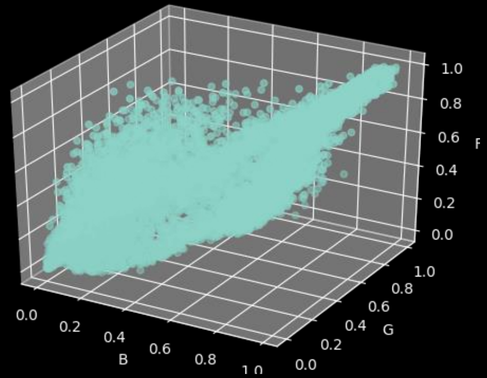
Suppose all parameters estimated $\{w_i, \mu_i, \sigma_i\}$

Assigning a label (“Gaussian”) to each data points is straightforward

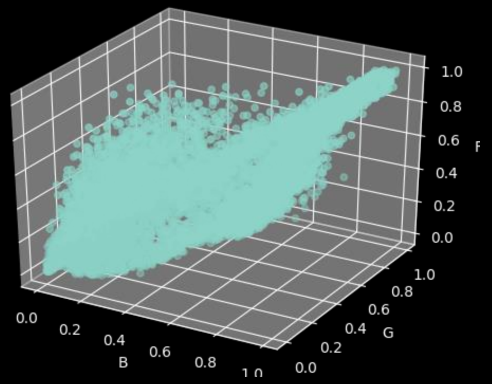
- The probability that a point belong to a Gaussian $\mathcal{N}$ is: $\mathrm{p}(\mathrm{x}|\mathcal{N}) = \mathcal{N}_{PDF}(x)$

- Suppose n Gaussians, the class is the Gaussian with highest probability

$$\mathrm{label}(\boldsymbol{x}) = \mathrm{argmax}_i\, p(\boldsymbol{x}|\mathcal{N}_i)$$

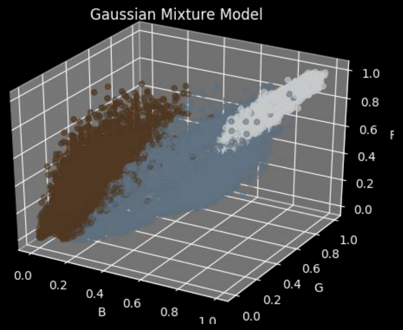


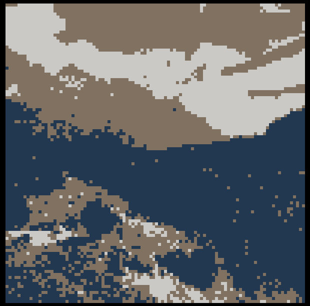
GMM with EM

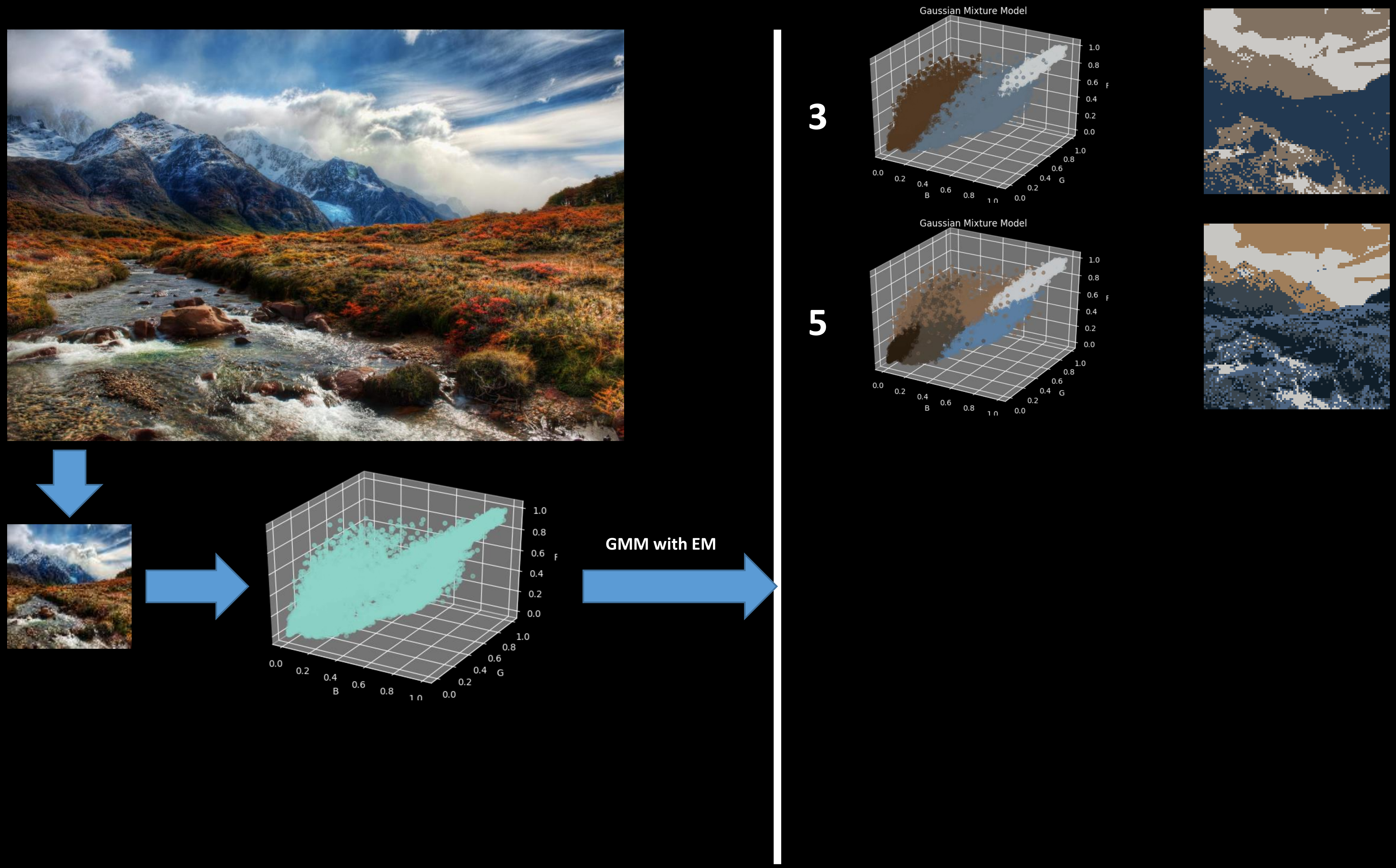


GMM with EM

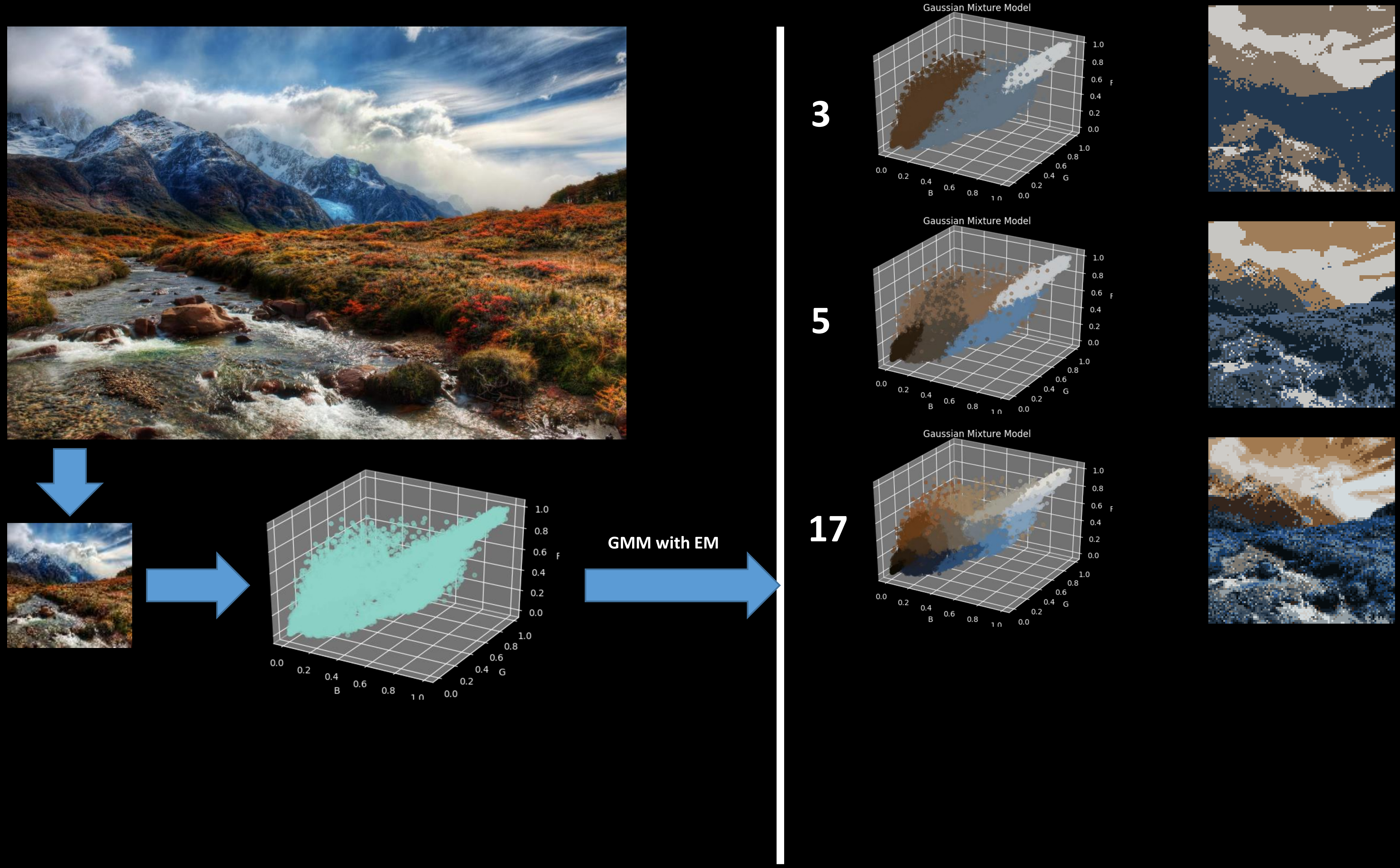
3

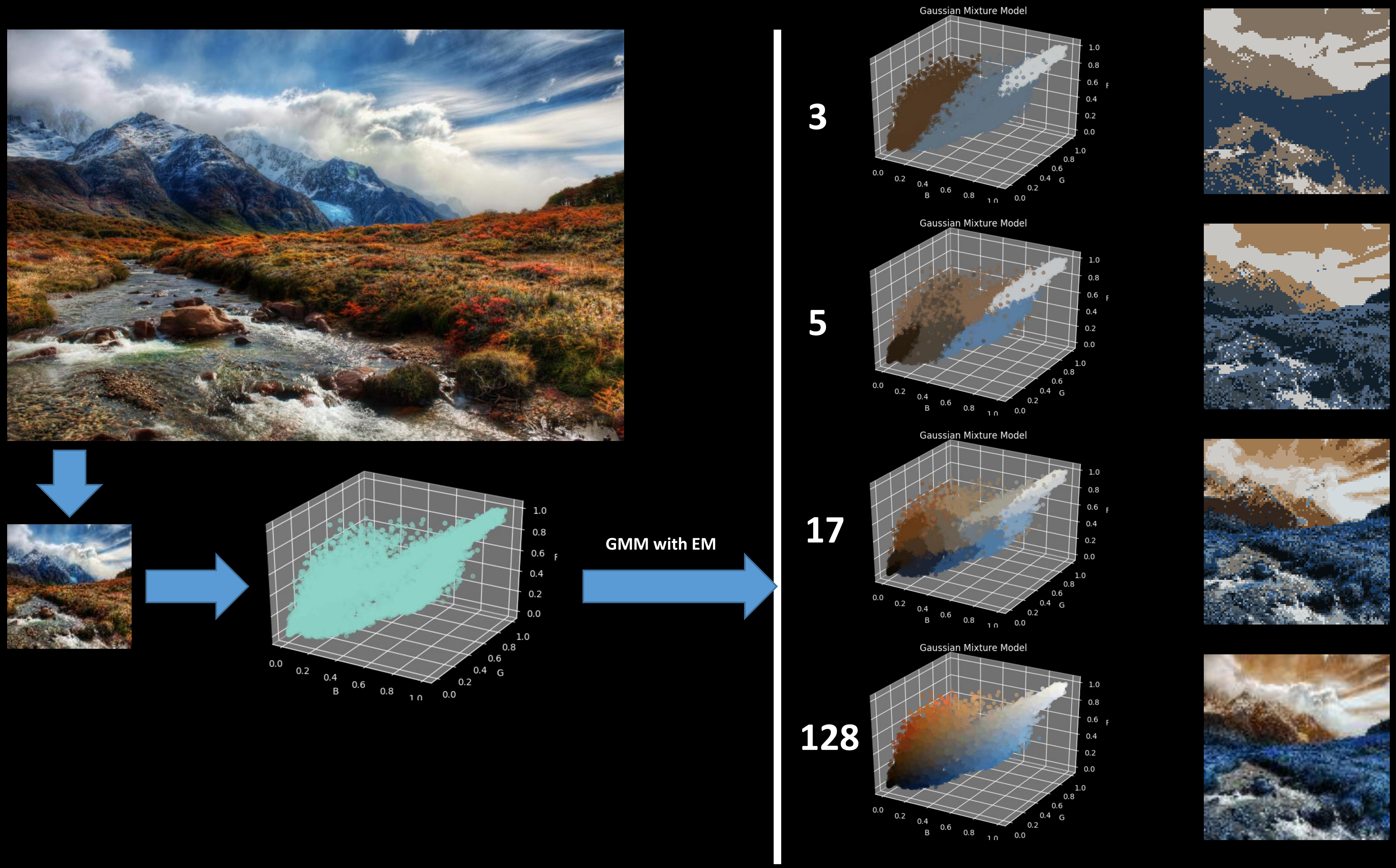






GMM with EM
3
5
Gaussian Mixture Model
Gaussian Mixture Model


GMM with EM
3
Gaussian Mixture Model
5
Gaussian Mixture Model
17
Gaussian Mixture Model
B
G


GMM with EM
Gaussian Mixture Model
3
Gaussian Mixture Model
5
Gaussian Mixture Model
17
Gaussian Mixture Model
128
B
G

Summarizing

- Point Model are very naïve
- Gaussian Model can work well (e.g. keying)
- Histograms works well for texture with a few modes
- Gaussian Mixture Models are very powerful with distinct modes

Summarizing

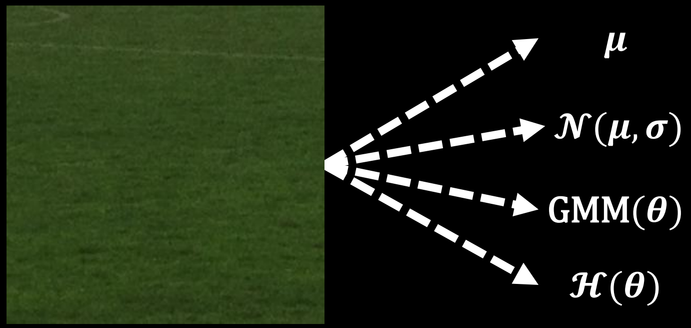
- Point Model are very naïve
- Gaussian Model can work well (e.g. keying)
- Histograms works well for texture with a few modes
- Gaussian Mixture Models are very powerful with distinct modes

- But… Can you see a problem with all these methods ?
  - They all think at a pixel level
  - Spatial information is lost

Summarizing

- Point Model are very naïve
- Gaussian Model can work well (e.g. keying)
- Histograms works well for texture with a few modes
- Gaussian Mixture Models are very powerful with distinct modes

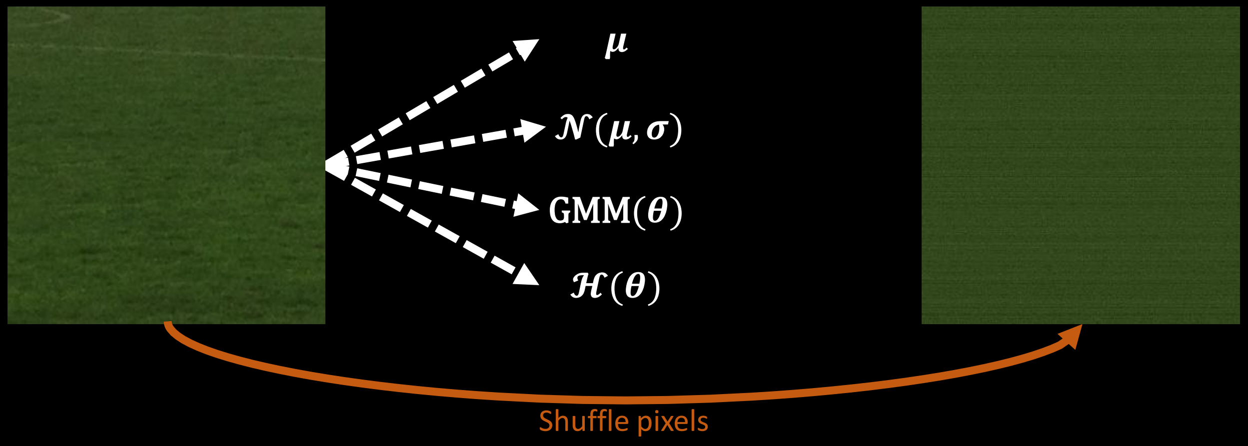
- But… Can you see a problem with all these methods ?
  - They all think at a pixel level
  - Spatial information is lost



Summarizing

- Point Model are very naïve
- Gaussian Model can work well (e.g. keying)
- Histograms works well for texture with a few modes
- Gaussian Mixture Models are very powerful with distinct modes

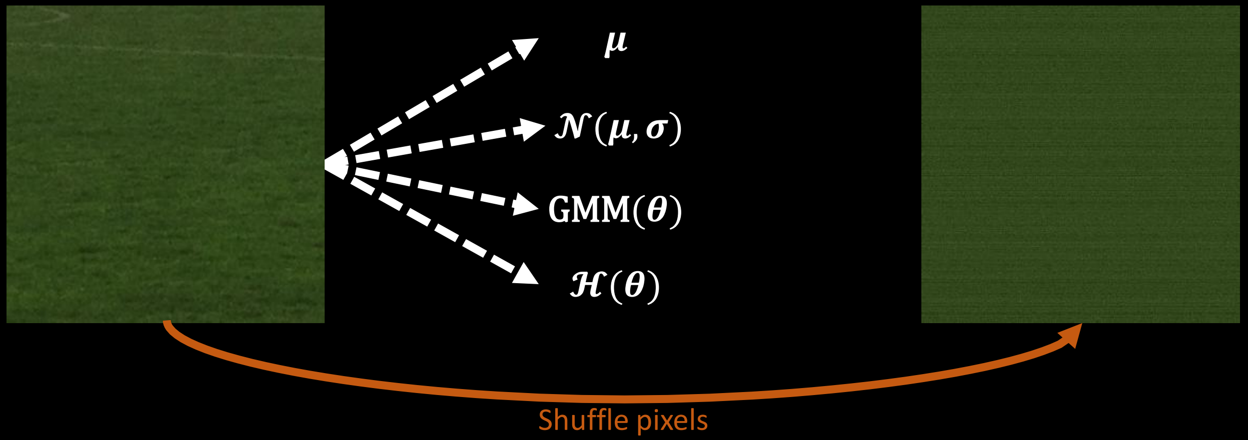
- But… Can you see a problem with all these methods ?
  - They all think at a pixel level
  - Spatial information is lost



Summarizing

- Point Model are very naïve
- Gaussian Model can work well (e.g. keying)
- Histograms works well for texture with a few modes
- Gaussian Mixture Models are very powerful with distinct modes

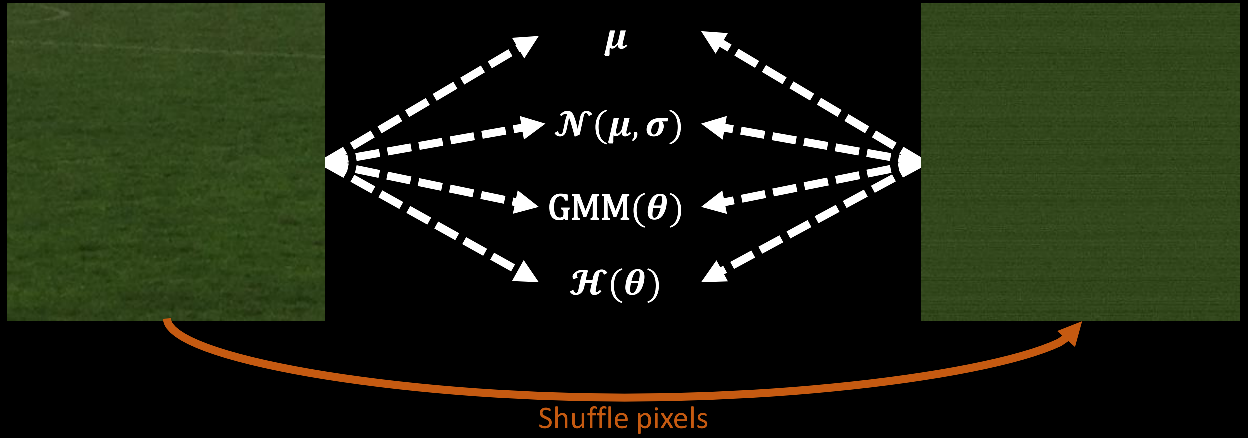
- But… Can you see a problem with all these methods ?
    - They all think at a pixel level
    - Spatial information is lost



Summarizing

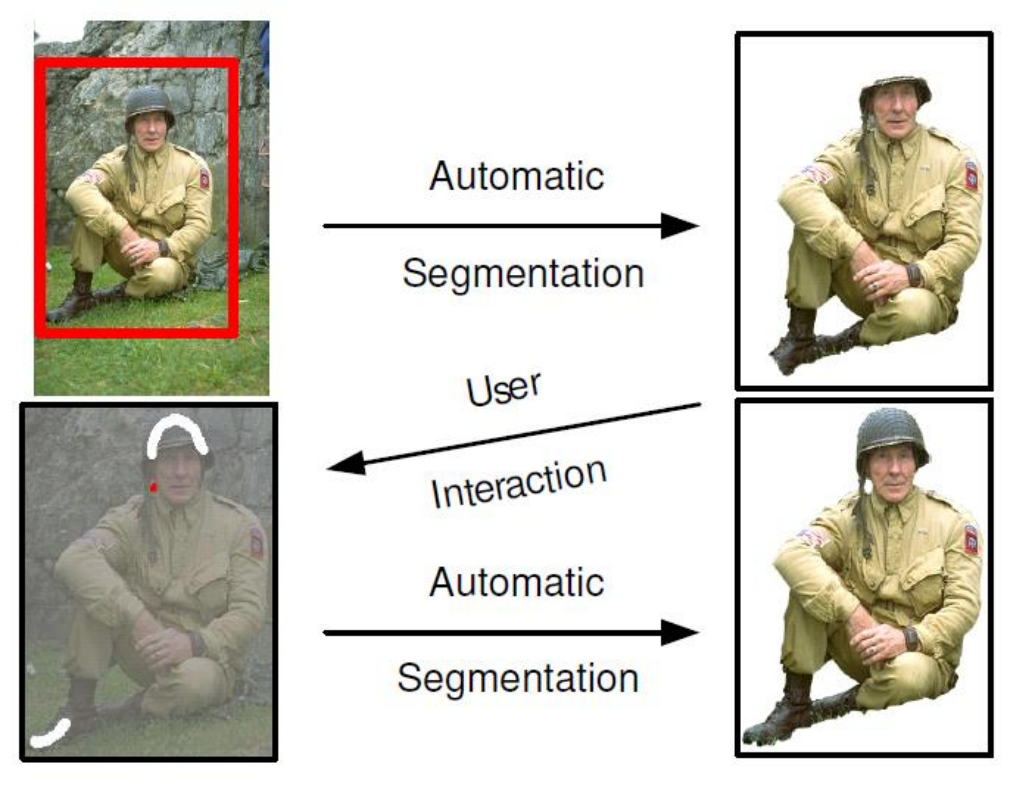
- Point Model are very naïve
- Gaussian Model can work well (e.g. keying)
- Histograms works well for texture with a few modes
- Gaussian Mixture Models are very powerful with distinct modes

- But… Can you see a problem with all these methods ?
  - They all think at a pixel level
  - Spatial information is lost



Interactive segmentation

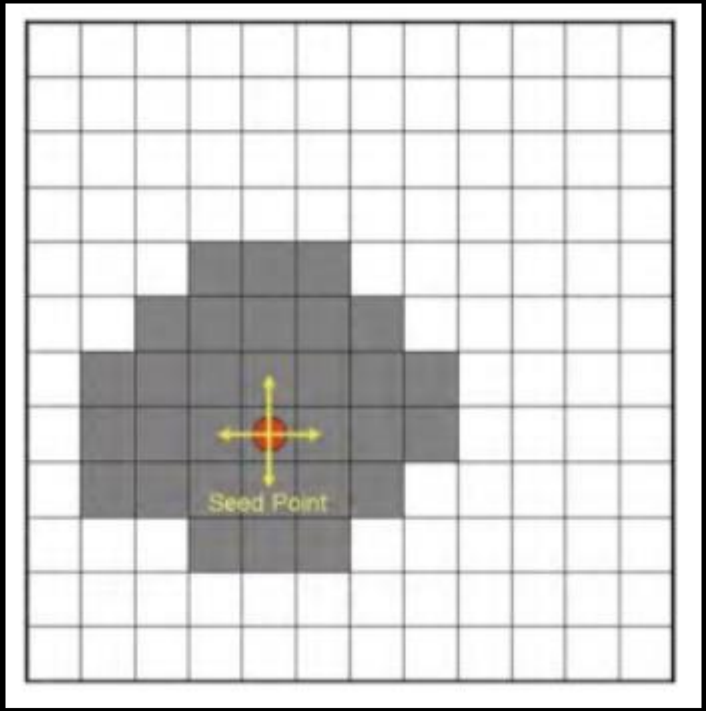
- What is it ?
- When to use interactive segmentation ?



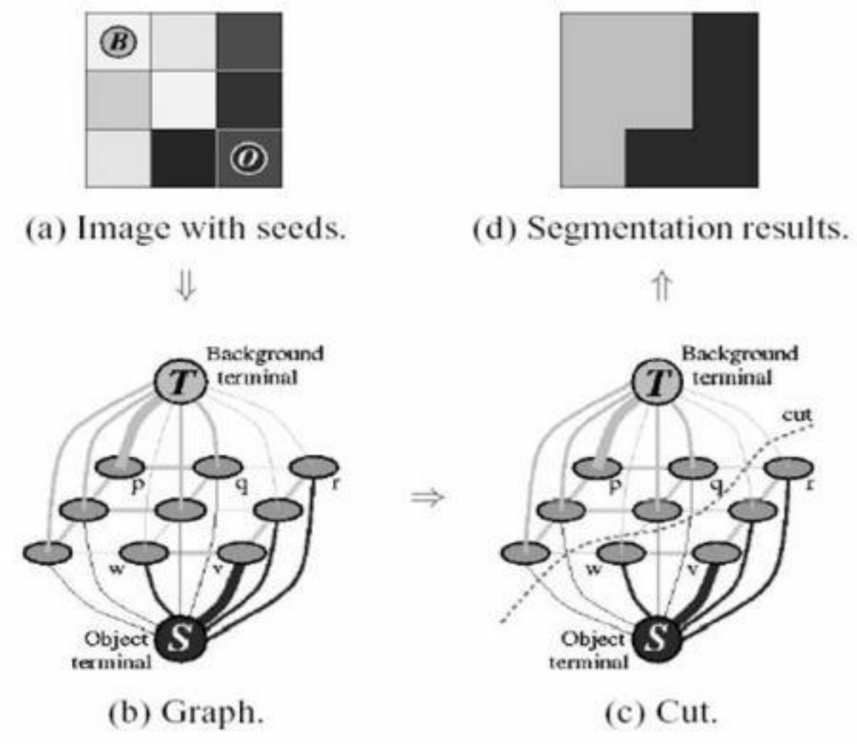
- Region growing
  - Start from a seed
  - And iteratively grow if criteria is still valid

- Usual criteria:
  - Distance to seed < threshold
  - Distance to neighbor < threshold
  - Region (max-min) < threshold

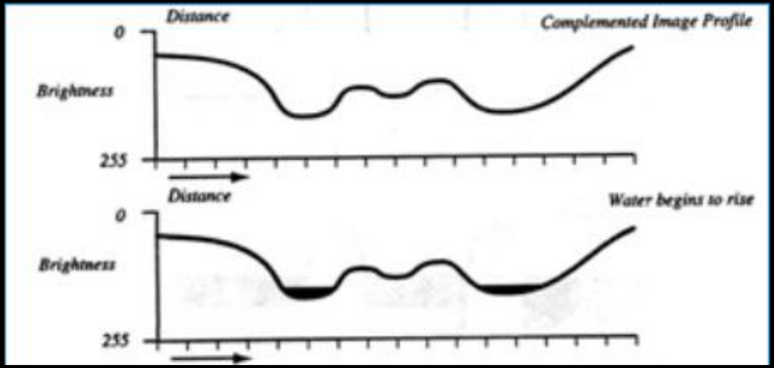
- Very sensitive to parameters
- Still used in 3D

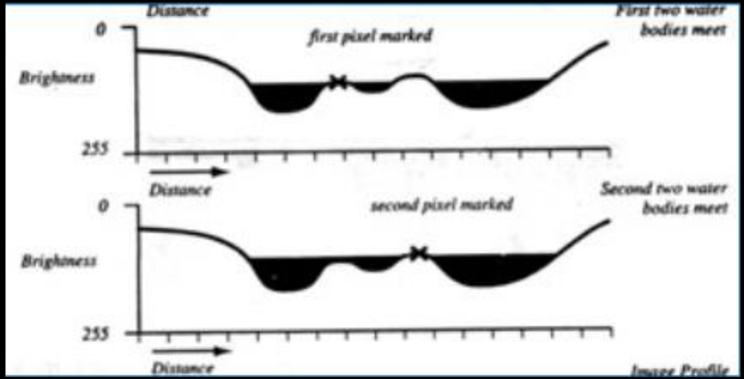


- Grab cut


(a) Image with seeds.
(d) Segmentation results.
Background terminal
cut
Object terminal
(b) Graph.
(c) Cut.

- Watershed


Distance
Complemented Image Profile
0
Brightness
255
Distance
Water begins to rise
0
Brightness
255


Distance
First two water bodies meet
0
first pixel marked
Brightness
255
Distance
Second two water bodies meet
0
second pixel marked
Brightness
255
Distance
Image Profile

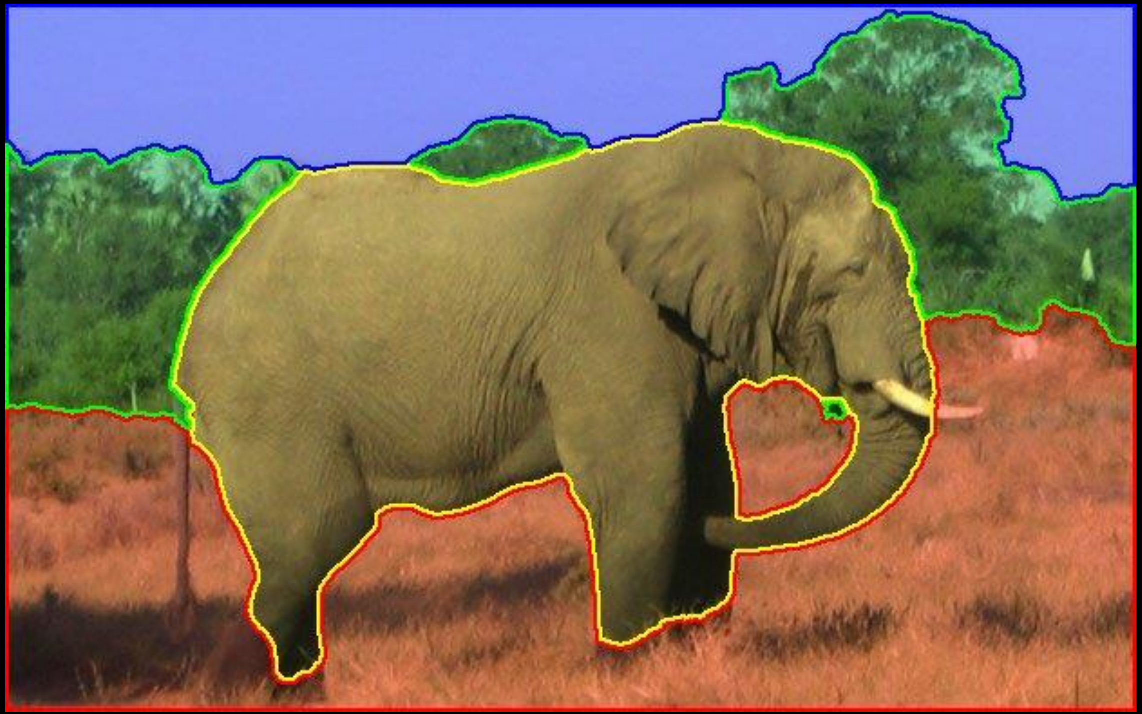
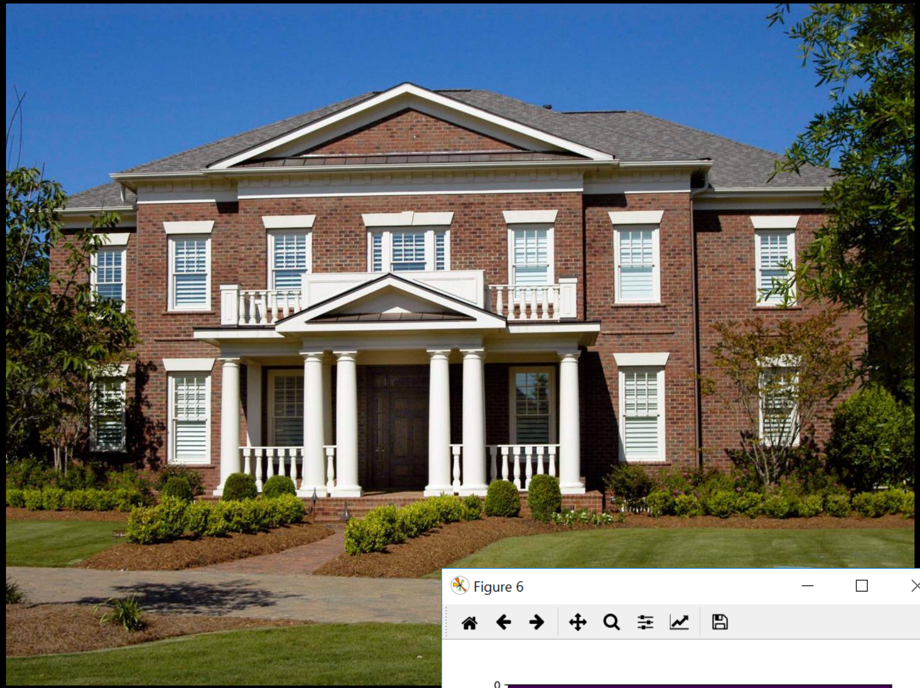
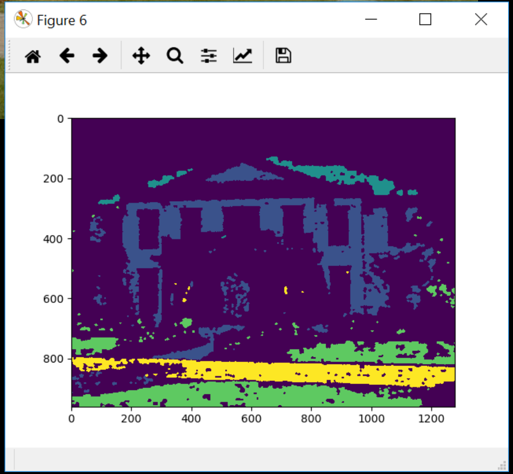


Image segmentation

Segment the **brick walls** (and else) with a histogram back projection





- Guidelines

  Read image and display it

  ```
  imBGR = cv2.imread("house.jpg")
  plt.imshow(imBGR[…,::-1])
  ```

  Select a sub region of the image to be the template and display it

  ```
  patchBGR = imBGR[y1:y2, x1:x2]
  plt.imshow(patchBGR[…,::-1])
  ```

  Convert BGR patch to HSV. (use `cv2.cvtColor` and `cv2.COLOR_BGR2HSV`)
  Build the histogram:

  ```
  hist = cv2.calcHist([patchHSV], channels=[0], mask=None,
                     histSize=[180], ranges=[0, 180])
  ```

  and normalize it (divide by max value)

  Compute the back projection of the histogram

  ```
  cv2.calcBackProject([imHSV], channels=[0], hist=hist, ranges=[0, 180], scale=255)
  ```

- Once walls are segmented continue with: **grass, roof, pathway**

- **Finished ?** 1) Build a real segmentation map where 0=void, 1=brick, 2=roof, 3=grass, 4=path. 2) Apply Gaussian model segmentation for sky.
- **You're really good ?** 1) Code your own back projection, start using calcHist. 2) Use a FD estimator for hist (yes, you need to recode calcHist). Helps others ☺
- **You're even better ?** Cluster the image with a GMM model (say: 20 Gaussians) and assign grass/roof/wall/pathway/sky label to each.

Legend: hints, values to adjust