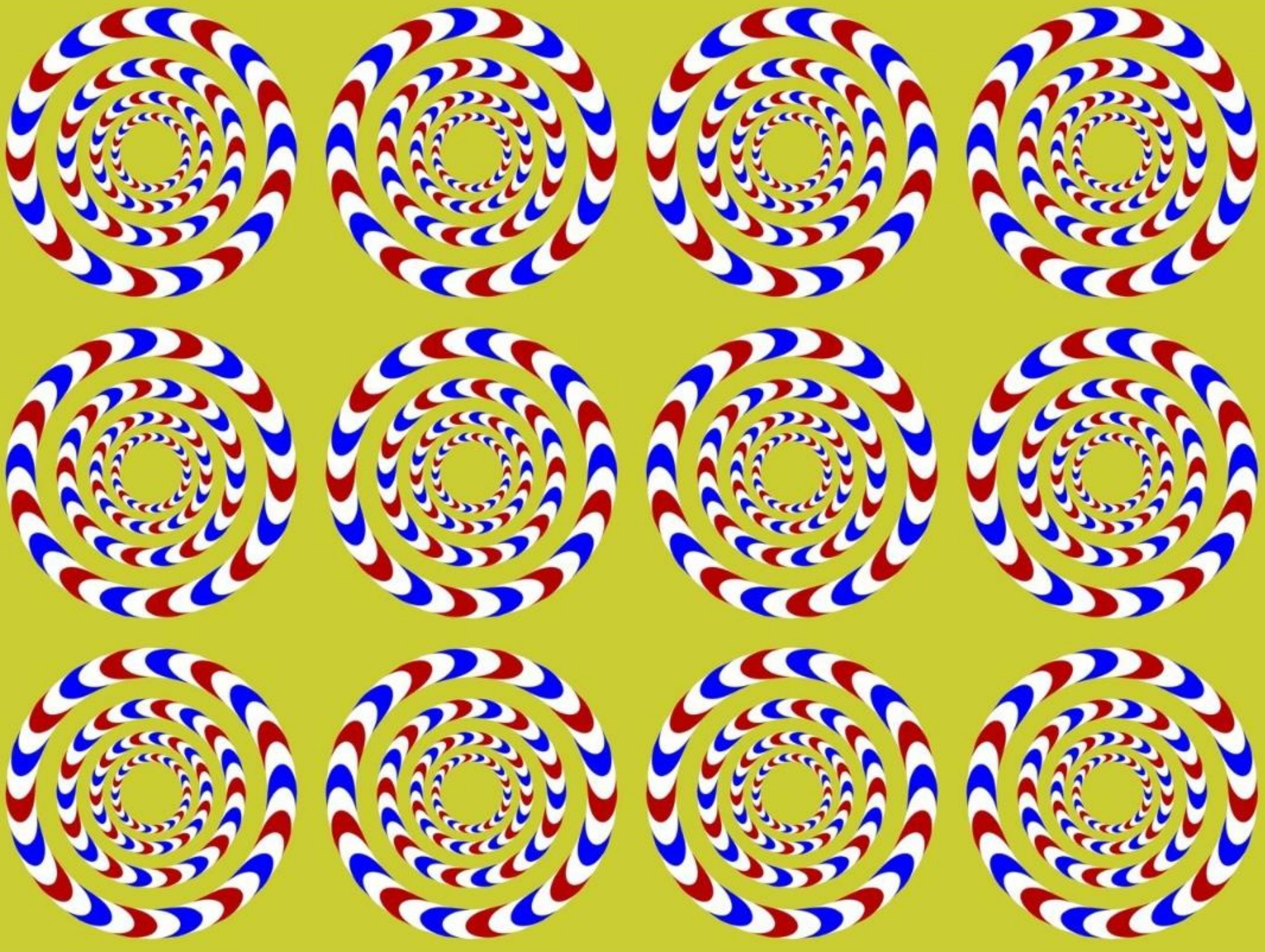


Recap: Optical Flow

Brightness constancy constraint equation

$$I_x u + I_y v + I_t = 0$$

- What do the static image gradients have to do with motion estimation?

If I told you
I_t is -5
I_x is 2
I_y is 1

What was the pixel shift (u,v)?

Recap: Conditions for solvability

Optimal (u, v) satisfies Lucas-Kanade equation

$$\underbrace{\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix}}_{A^T A} \begin{bmatrix} u \\ v \end{bmatrix} = - \underbrace{\begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}}_{A^T b}$$

When is this solvable? I.e., what are good points to track?

- $\mathbf{A^TA}$ should be invertible
- $\mathbf{A^TA}$ should not be too small due to noise
  - eigenvalues λ_1 and λ_2 of $\mathbf{A^TA}$ should not be too small
- $\mathbf{A^TA}$ should be well-conditioned
  - λ_1 / λ_2 should not be too large (λ_1 = larger eigenvalue)

Does this remind you of anything?

Criteria for Harris corner detector

Machine Learning Crash Course

Computer Vision
James Hays

Photo: CMU Machine Learning Department protests G20

Slides: Isabelle Guyon, Erik Sudderth, Mark Johnson, Derek Hoiem

Machine Learning Problems

	Supervised Learning	***Unsupervised Learning***
Discrete	classification or categorization	clustering
Continuous	regression	$\boxed{\text{dimensionality reduction}}$

Dimensionality Reduction

Simplest dimensionality reduction: drop a dimension

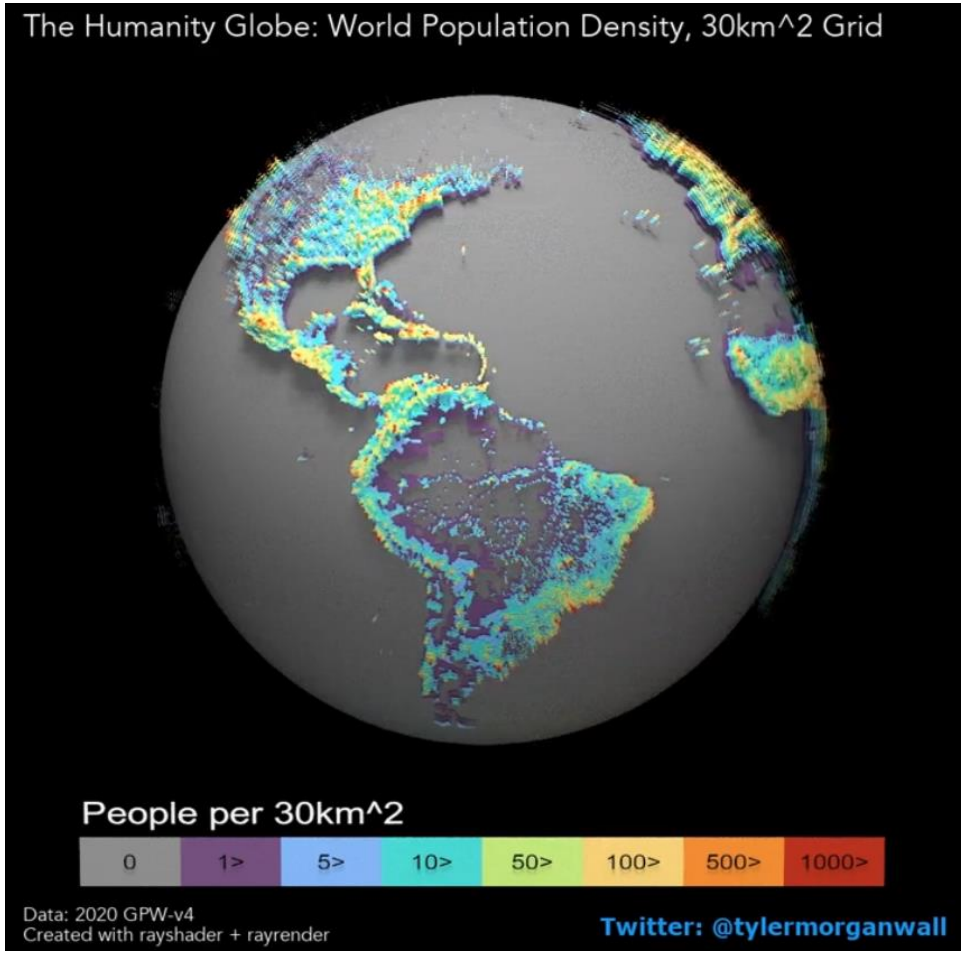
Credit: xkcd

Dimensionality Reduction

The Earth is pretty smooth

Credit: xkcd

Non-linear Dimensionality Reduction



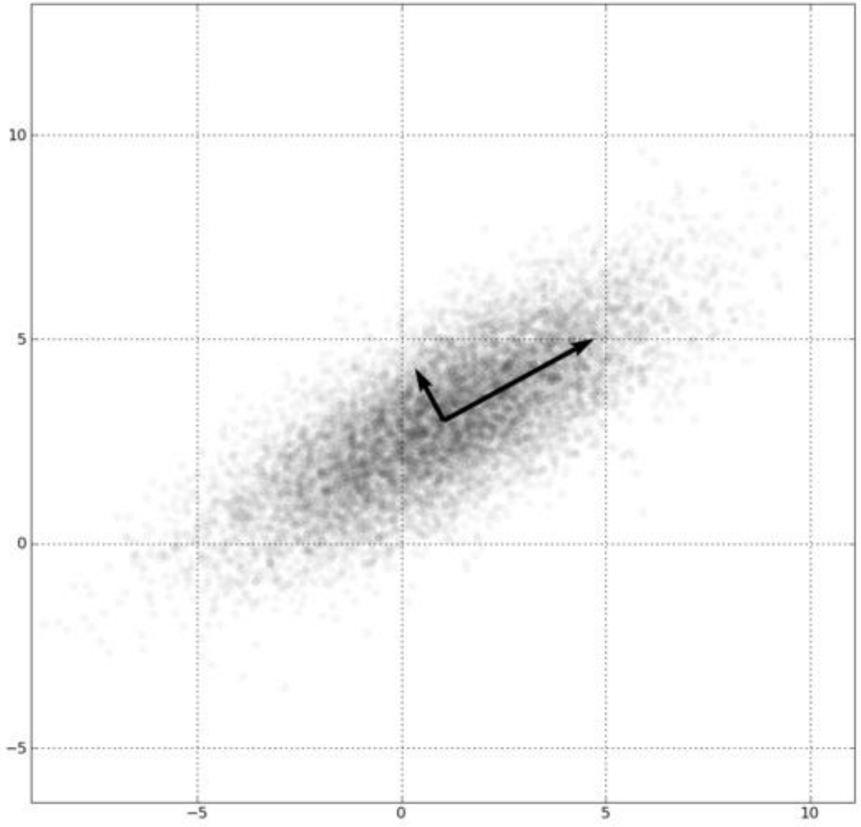
Credit: xkcd

Dimensionality Reduction

- **PCA**, ICA, LLE, Isomap, *Autoencoder*

- PCA is the most important technique to know. It takes advantage of correlations in data dimensions to produce the best possible lower dimensional representation based on linear projections (minimizes reconstruction error).

- Be wary of trying to assign meaning to the discovered bases.



Eigenfaces for Recognition

Matthew Turk and Alex Pentland
Vision and Modeling Group
The Media Laboratory
Massachusetts Institute of Technology

Training data
16 256x256 images

Figure 1. (b) The average face Ψ.

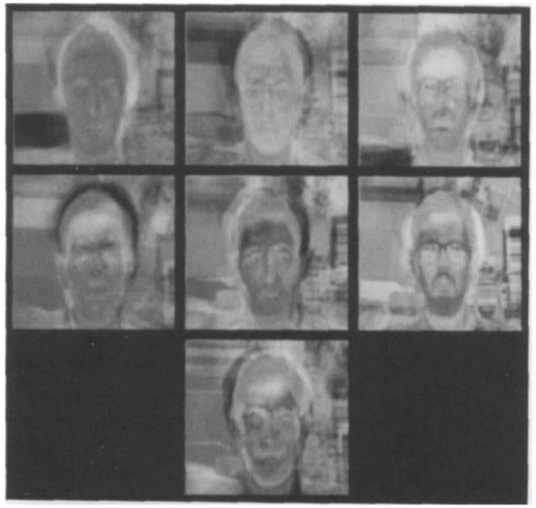
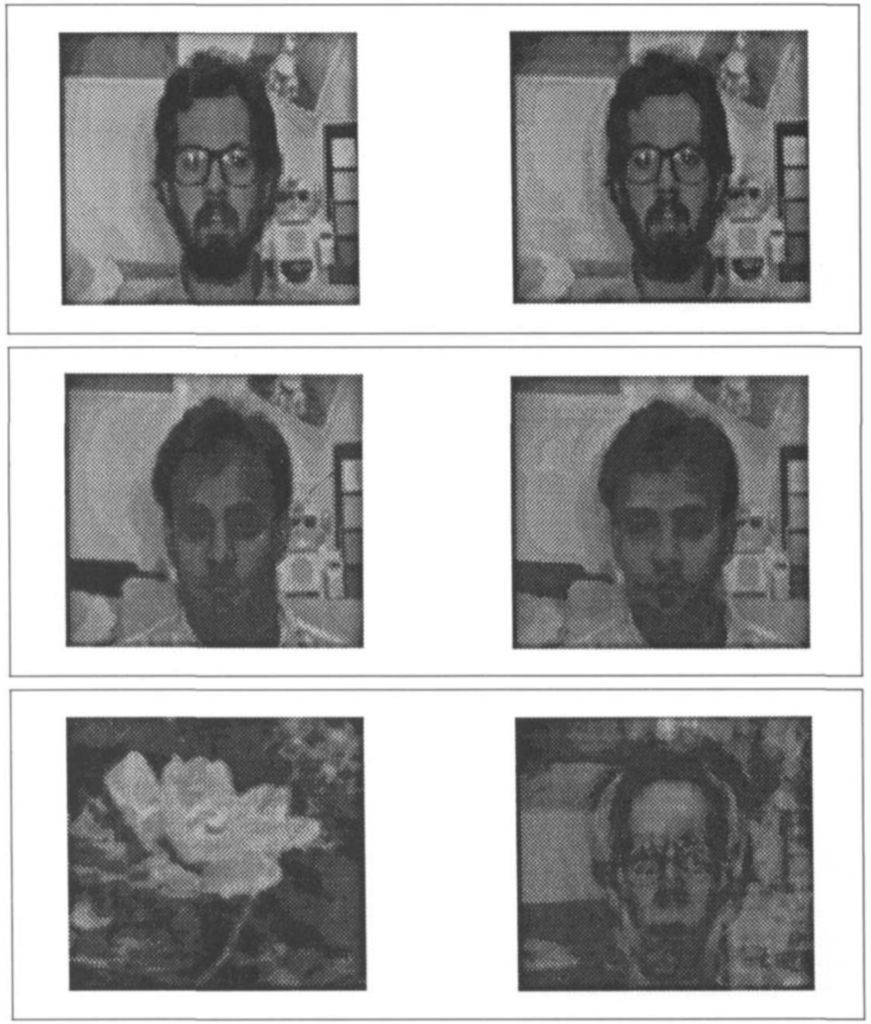


Figure 2. Seven of the eigenfaces calculated from the input images of Figure 1.

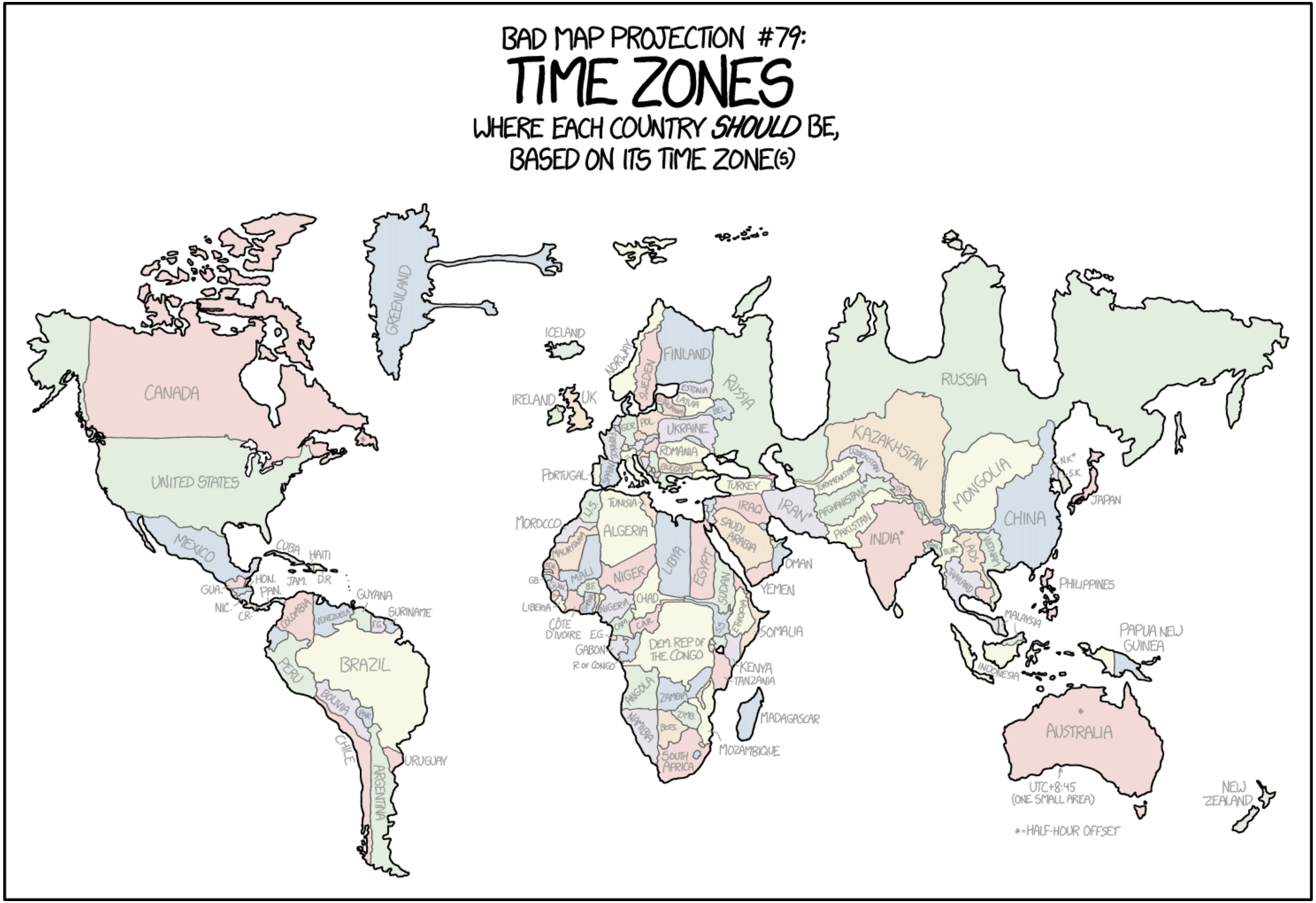
The "Eigenfaces"



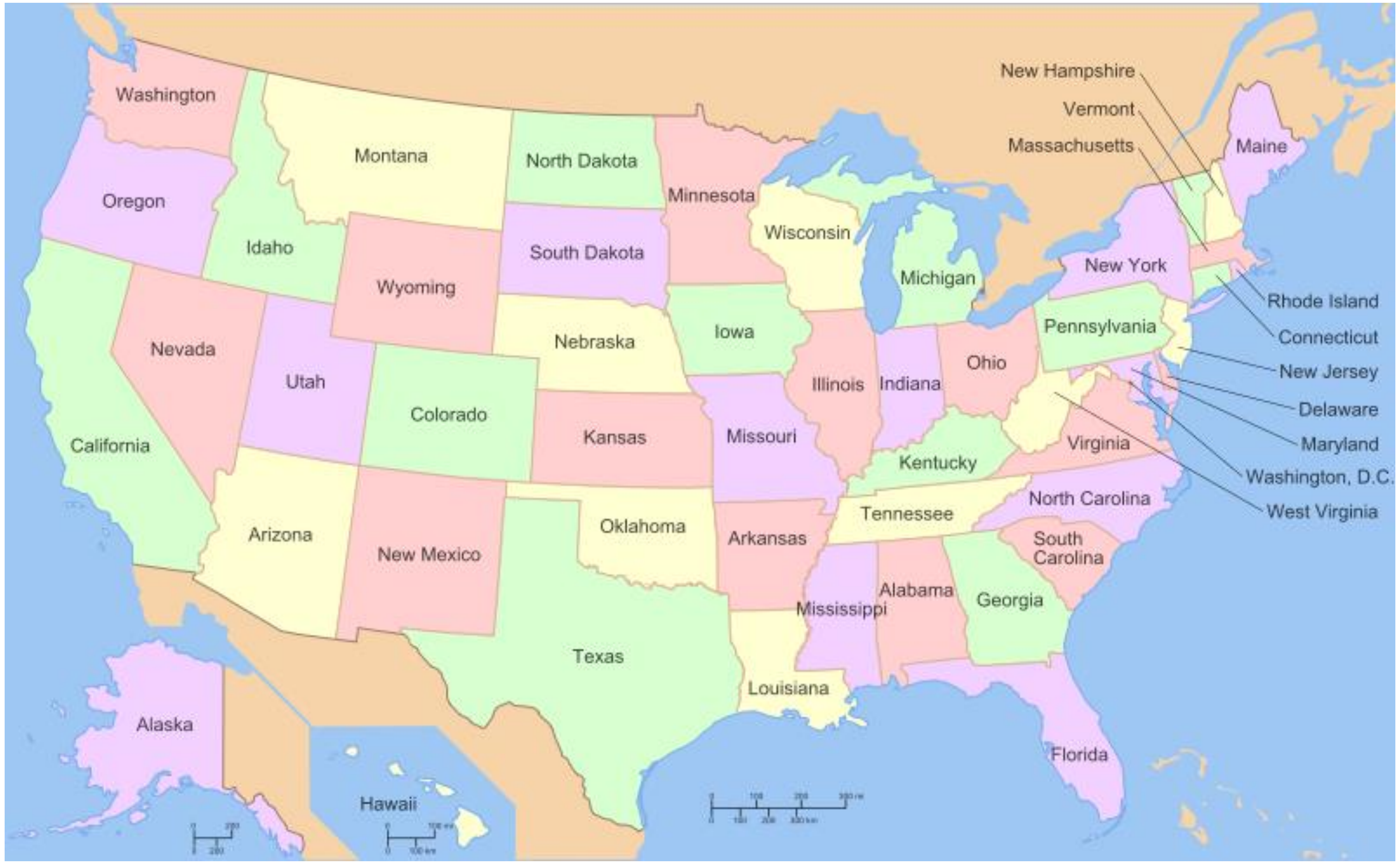
Reconstruction of in-domain and out-of-domain images

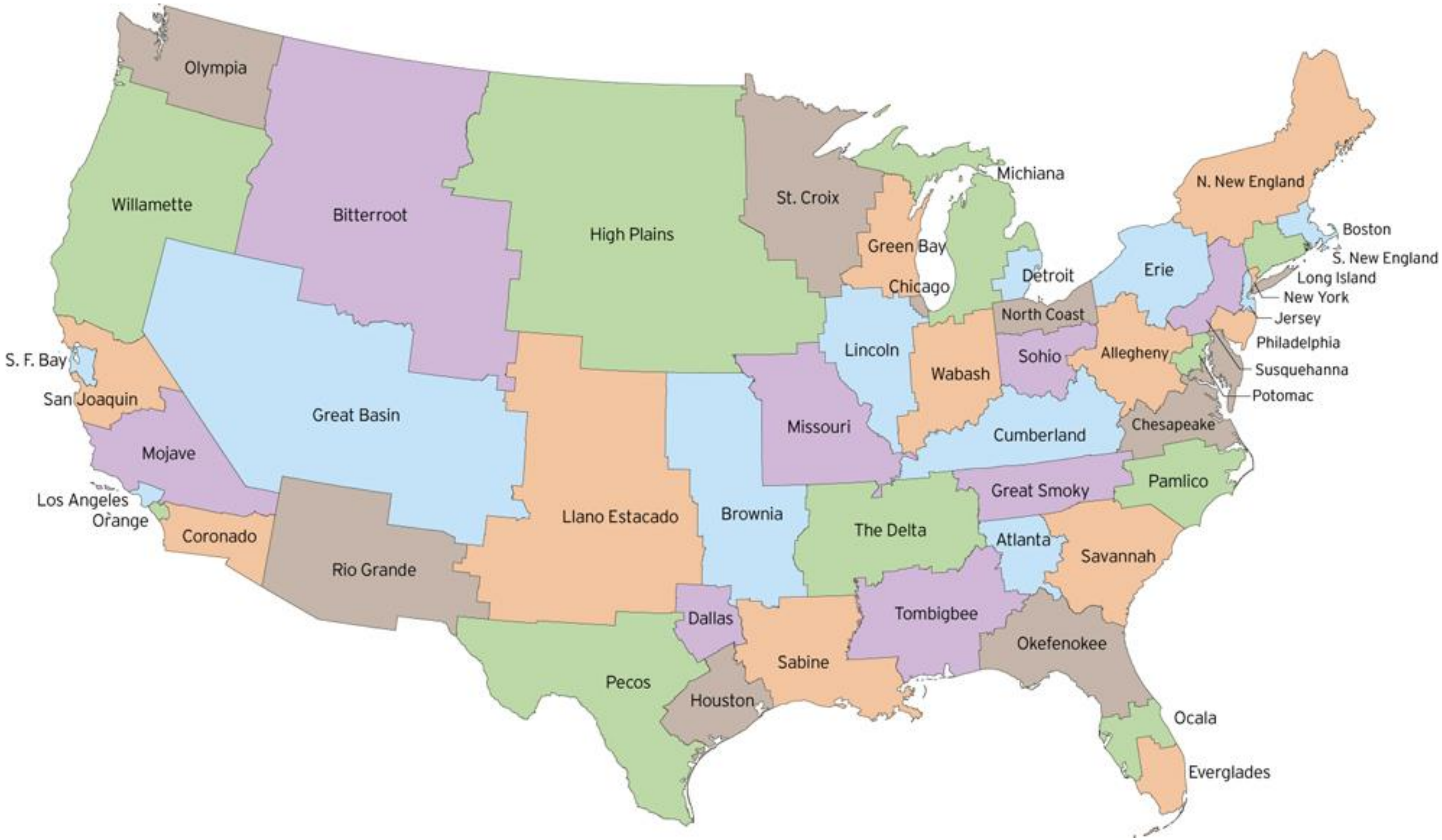
Machine Learning Problems

	Supervised Learning	***Unsupervised Learning***
Discrete	classification or categorization	clustering
Continuous	regression	dimensionality reduction

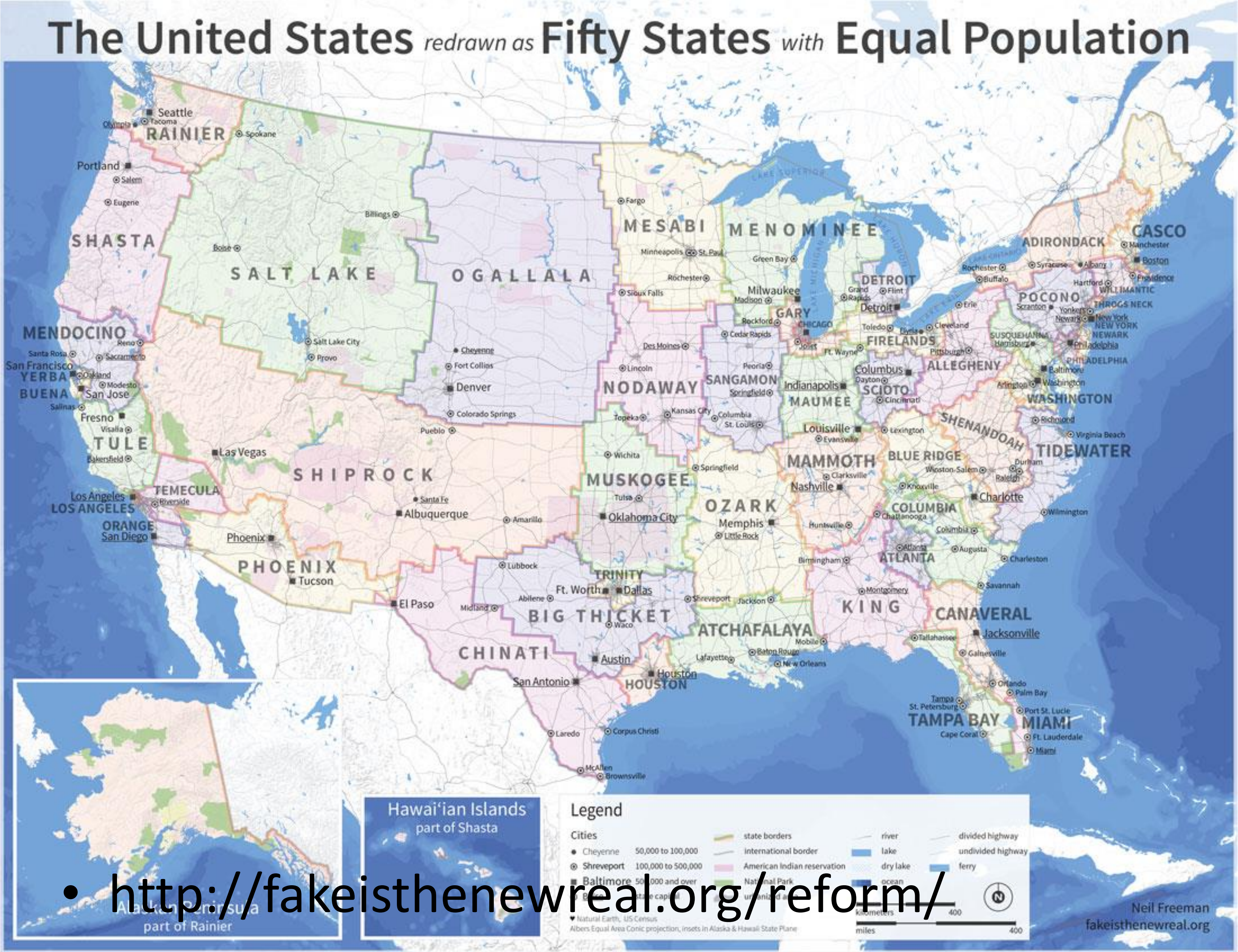

BAD MAP PROJECTION #79:
TIME ZONES
WHERE EACH COUNTRY *SHOULD* BE, BASED ON ITS TIME ZONE(S)
GREENLAND
CANADA
UNITED STATES
MEXICO
CUBA
HAITI
HON.
JAM.
D.R.
PAN.
GUA.
NIC.
CR.
GUYANA
SURINAME
COLOMBIA
VENEZUELA
BRAZIL
PERU
BOLIVIA
PAR.
CHILE
ARGENTINA
URUGUAY
ICELAND
IRELAND
UK
NORWAY
SWEDEN
FINLAND
ESTONIA
LATVIA
BEL.
RUSSIA
GER.
POL.
UKRAINE
FRANCE
ROMANIA
BULGARIA
SPAIN
PORTUGAL
TURKEY
MOROCCO
TUNISIA
ALGERIA
LIBYA
EGYPT
MAURITANIA
MALI
NIGER
CHAD
SUDAN
ETHIOPIA
GB.
LIBERIA
CÔTE D'IVOIRE
NIGERIA
CAM.
C.A.R.
E.G.
GABON
R. OF CONGO
DEM. REP. OF THE CONGO
KENYA
TANZANIA
ANGOLA
ZAMBIA
ZIMB.
NAMIBIA
BOTS.
SOUTH AFRICA
MOZAMBIQUE
MADAGASCAR
SOMALIA
YEMEN
OMAN
SAUDI ARABIA
IRAQ
IRAN*
TURKMENISTAN
UZBEKISTAN
AFGHANISTAN*
PAKISTAN
TAJ.
KAZAKHSTAN
INDIA*
MONGOLIA
CHINA
N.K.*
S.K.
JAPAN
BUR.*
LAOS
VIETNAM
THAILAND
PHILIPPINES
MALAYSIA
INDONESIA
PAPUA NEW GUINEA
AUSTRALIA
UTC+8:45 (ONE SMALL AREA)
NEW ZEALAND
*=HALF-HOUR OFFSET

Credit: xkcd


New Hampshire
Vermont
Massachusetts
Maine
Washington
Montana
North Dakota
Minnesota
Oregon
Idaho
Wisconsin
South Dakota
Wyoming
Michigan
New York
Rhode Island
Connecticut
Nevada
Nebraska
Iowa
Pennsylvania
Ohio
New Jersey
Utah
Illinois
Indiana
Colorado
Delaware
California
Kansas
Missouri
Virginia
Maryland
Kentucky
Washington, D.C.
North Carolina
Tennessee
West Virginia
Oklahoma
Arizona
Arkansas
South Carolina
New Mexico
Alabama
Georgia
Mississippi
Texas
Louisiana
Alaska
Florida
Hawaii



- http://fakeisthenewreal.org/reform/



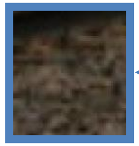
- http://fakeisthenewreal.org/reform/

Clustering example: image segmentation

Goal: Break up the image into meaningful or perceptually similar regions

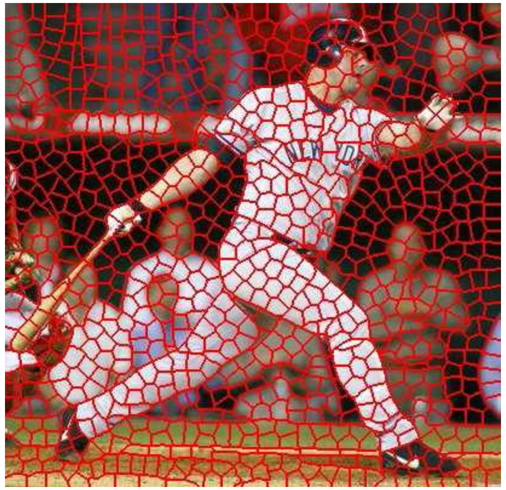
Segmentation for feature support or efficiency

50x50
Patch



50x50
Patch

[Felzenszwalb and Huttenlocher 2004]



[Shi and Malik 2001]

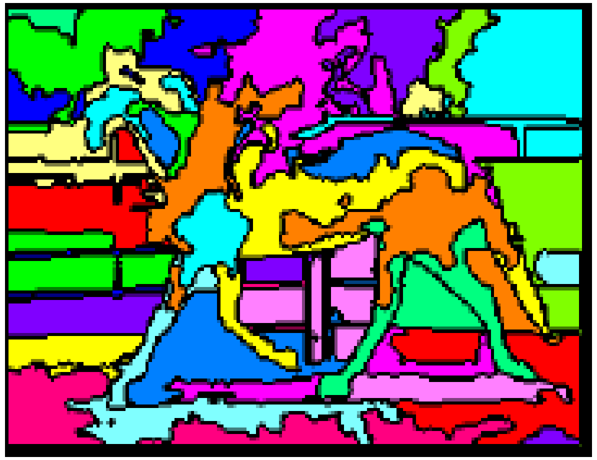
[Hoiem et al. 2005, Mori 2005]

Slide: Derek Hoiem

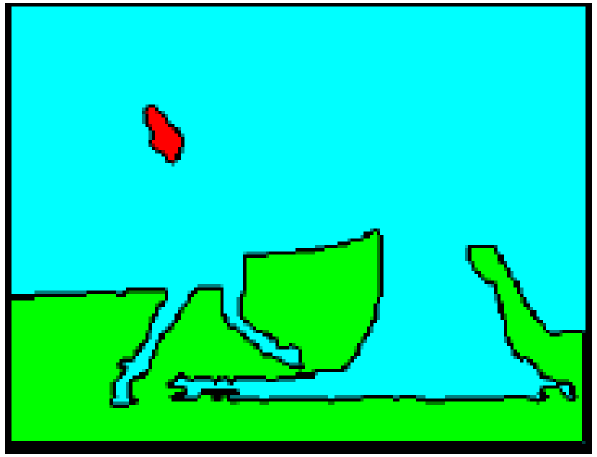
Segmentation as a result

Rother et al. 2004

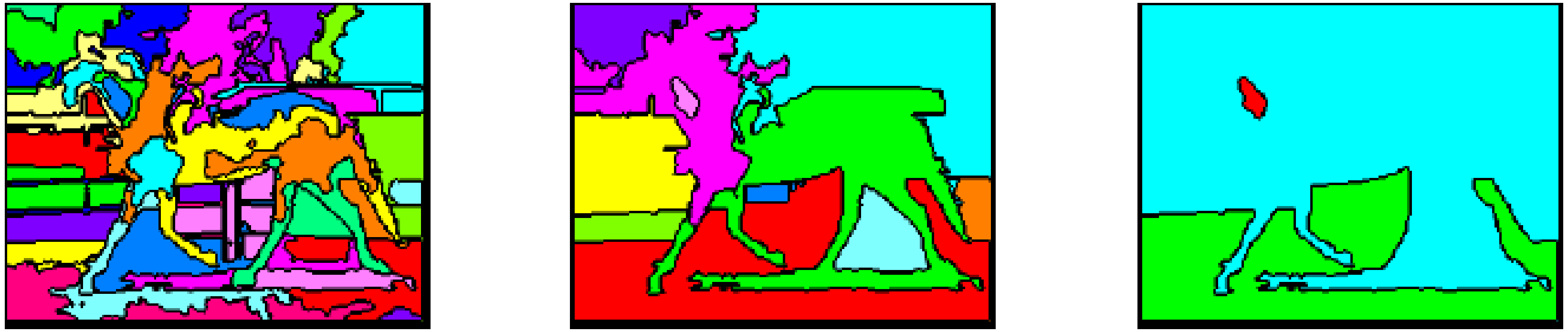
Types of segmentations



Oversegmentation



Undersegmentation



Multiple Segmentations

Clustering: group together similar points and represent them with a single token

Key Challenges:

1) What makes two points/images/patches similar?

2) How do we compute an overall grouping from pairwise similarities?

Slide: Derek Hoiem

How do we cluster?

- K-means
  - Iteratively re-assign points to the nearest cluster center
- Agglomerative clustering
  - Start with each point as its own cluster and iteratively merge the closest clusters
- Mean-shift clustering
  - Estimate modes of pdf
- Spectral clustering
  - Split the nodes in a graph based on assigned links with similarity weights

Clustering for Summarization

Goal: cluster to minimize variance in data given clusters

– Preserve information

Cluster center

Data

$$\mathbf{c}^*, \boldsymbol{\delta}^* = \underset{\mathbf{c},\boldsymbol{\delta}}{\operatorname{argmin}} \tfrac{1}{N} \sum_{j}^{N} \sum_{i}^{K} \delta_{ij} \left(\mathbf{c}_i - \mathbf{x}_j \right)^2$$

Whether x_j is assigned to c_i

K-means algorithm

1. Randomly
select K centers

2. Assign each
point to nearest
center

3. Compute new
center (mean)
for each cluster

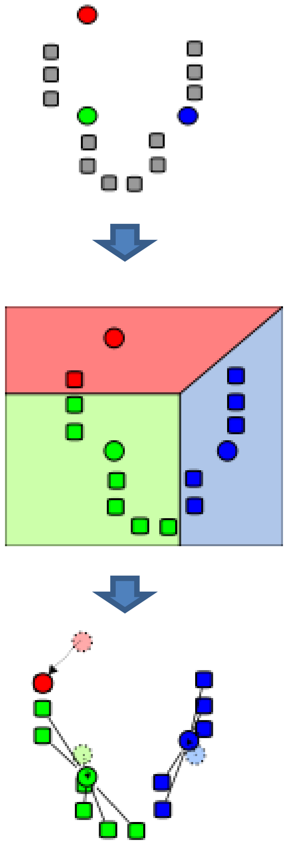


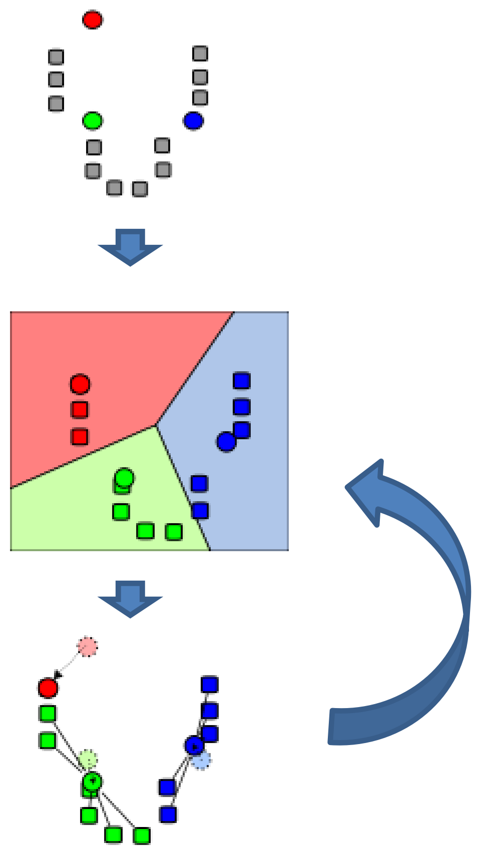
Illustration: http://en.wikipedia.org/wiki/K-means_clustering

K-means algorithm

1. Randomly select K centers

2. Assign each point to nearest center

3. Compute new center (mean) for each cluster



Back to 2

Illustration: http://en.wikipedia.org/wiki/K-means_clustering

K-means

1. Initialize cluster centers: $\mathbf{c}^0$; t=0

2. Assign each point to the closest center

$$\boldsymbol{\delta}^t = \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \tfrac{1}{N} \sum_j^N \sum_i^K \delta_{ij} \left(\mathbf{c}_i^{t-1} - \mathbf{x}_j\right)^2$$

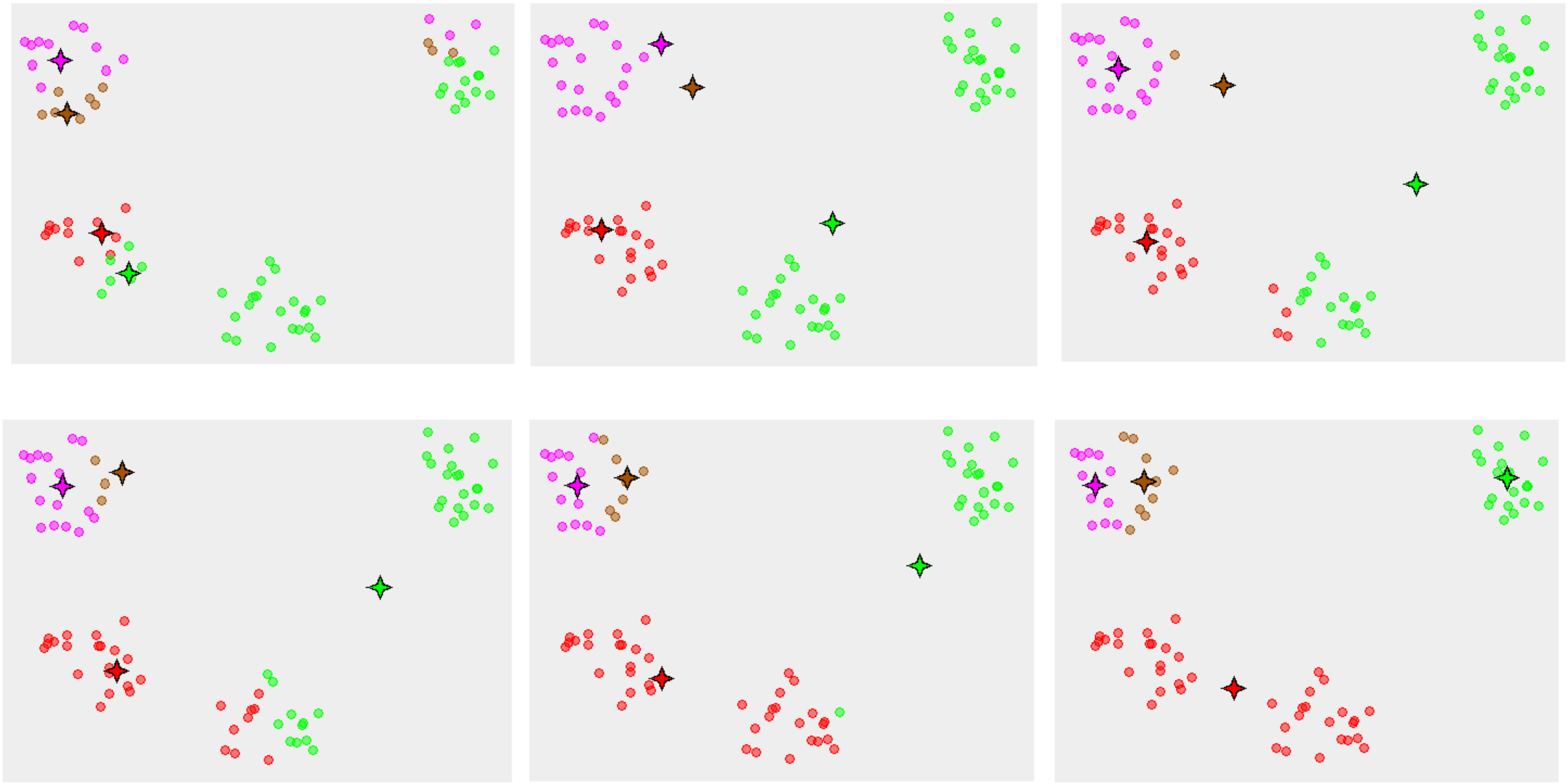
3. Update cluster centers as the mean of the points

$$\mathbf{c}^t = \underset{\mathbf{c}}{\operatorname{argmin}} \tfrac{1}{N} \sum_j^N \sum_i^K \delta_{ij}^t \left(\mathbf{c}_i - \mathbf{x}_j\right)^2$$

4. Repeat 2-3 until no points are re-assigned (t=t+1)

Slide: Derek Hoiem

K-means converges to a local minimum



K-means: design choices

- Initialization
  – Randomly select K points as initial cluster center
  – Or greedily choose K points to minimize residual
- Distance measures
  – Traditionally Euclidean, could be others
- Optimization
  – Will converge to a *local minimum*
  – May want to perform multiple restarts

K-means clustering using intensity or color

Image

Clusters on intensity

Clusters on color

How to evaluate clusters?

- Generative
  - How well are points reconstructed from the clusters?
- Discriminative
  - How well do the clusters correspond to labels?
    - Purity
  - Note: unsupervised clustering does not aim to be discriminative

Slide: Derek Hoiem

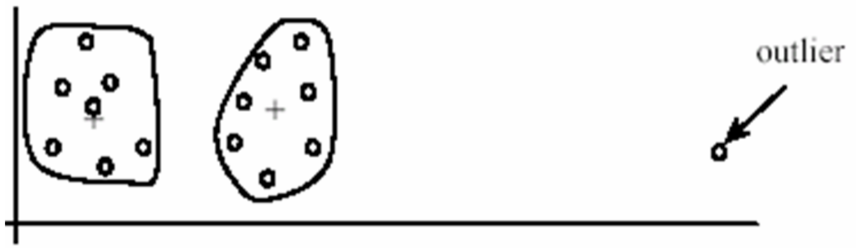
How to choose the number of clusters?

- Validation set
  - Try different numbers of clusters and look at performance on some downstream task
    - When building dictionaries (discussed later), more clusters typically work better

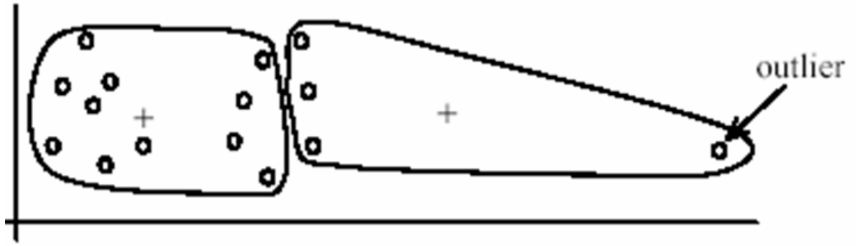
Slide: Derek Hoiem

K-Means pros and cons

- Pros
  - Finds cluster centers that minimize conditional variance (good representation of data)
  - Simple and fast*
  - Easy to implement
- Cons
  - Need to choose K
  - Sensitive to outliers
  - Prone to local minima
  - All clusters have the same parameters (e.g., distance measure is non-adaptive)
  - *Can be slow: each iteration is O(KNd) for N d-dimensional points
- Usage
  - Rarely used for pixel segmentation



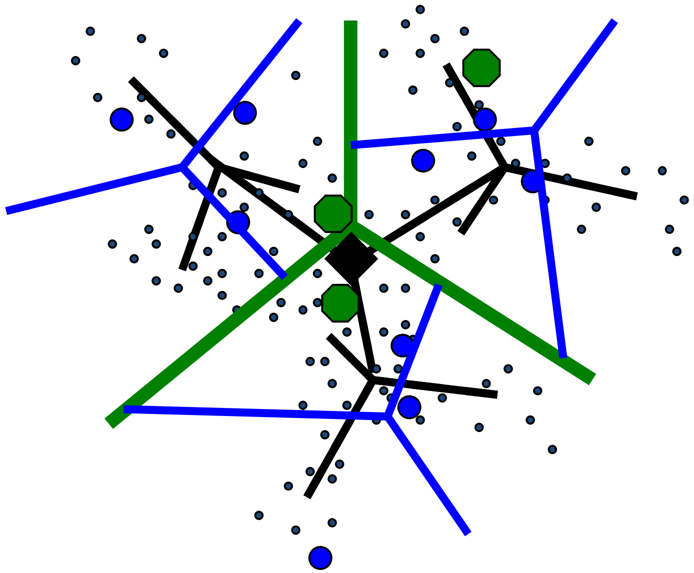
(B): Ideal clusters



Building Visual Dictionaries

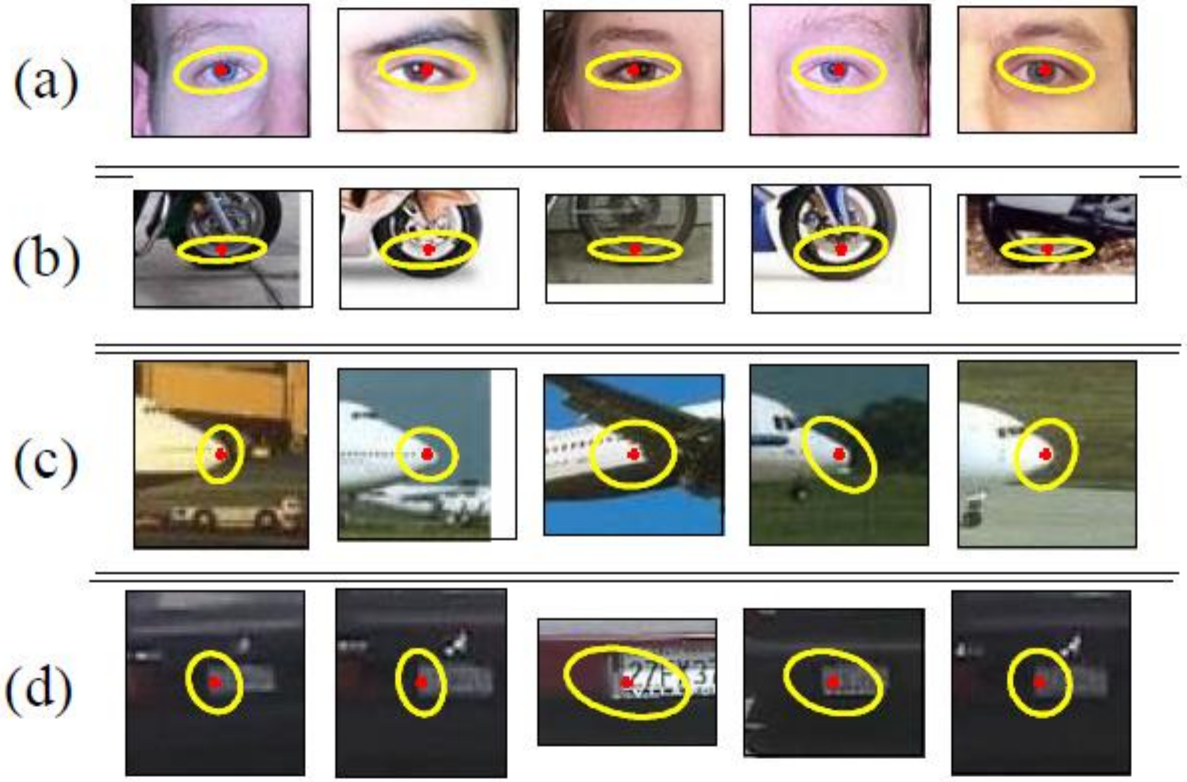
1. Sample patches from a database
   – E.g., 128 dimensional SIFT vectors

2. Cluster the patches
   – Cluster centers are the dictionary



3. Assign a codeword (number) to each new patch, according to the nearest cluster

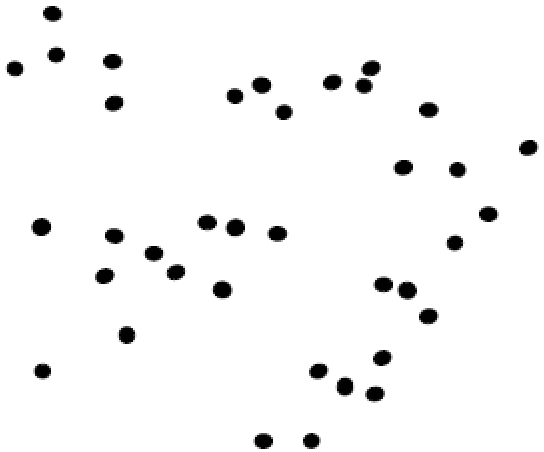
Examples of learned codewords



Most likely codewords for 4 learned "topics"

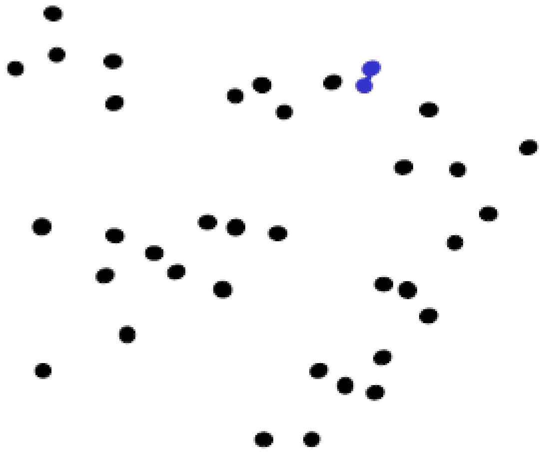
http://www.robots.ox.ac.uk/~vgg/publications/papers/sivic05b.pdf Sivic et al. ICCV 2005

Agglomerative clustering



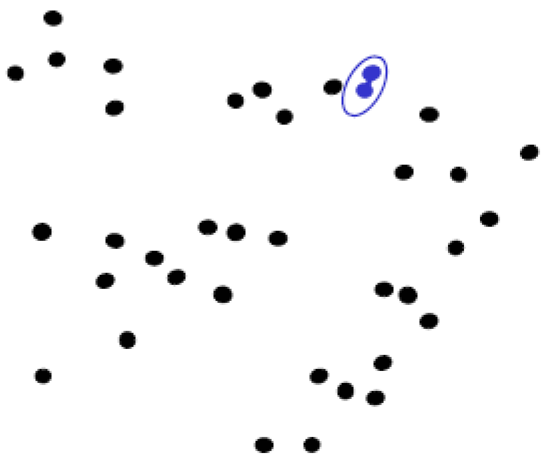
1. Say "Every point is its own cluster"

Agglomerative clustering



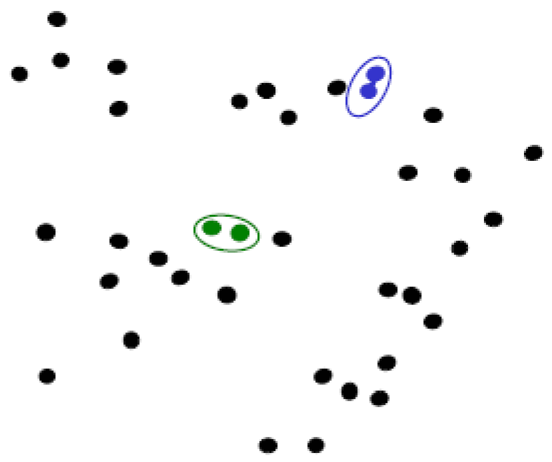
1. Say "Every point is its own cluster"
2. Find "most similar" pair of clusters

Agglomerative clustering

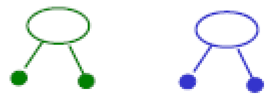


1. Say "Every point is its own cluster"
2. Find "most similar" pair of clusters
3. Merge it into a parent cluster

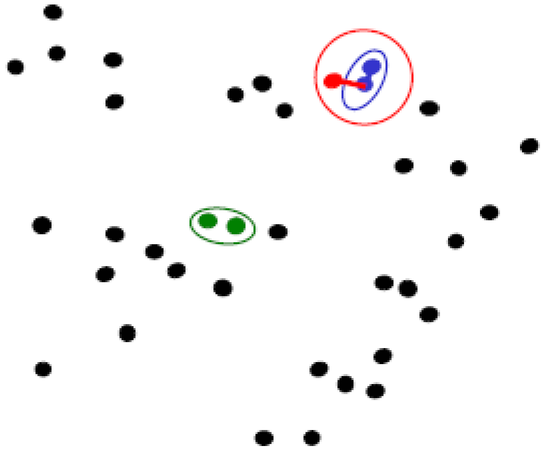
Agglomerative clustering



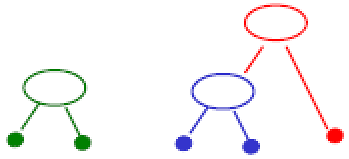
1. Say "Every point is its own cluster"
2. Find "most similar" pair of clusters
3. Merge it into a parent cluster
4. Repeat



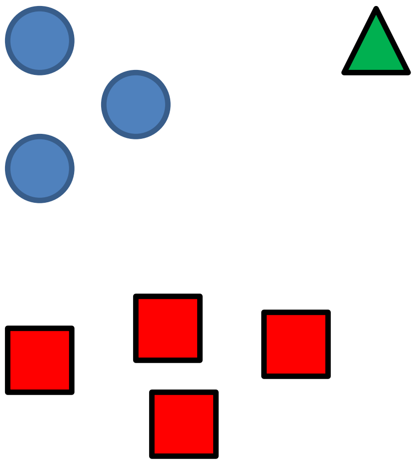
Agglomerative clustering



1. Say "Every point is its own cluster"
2. Find "most similar" pair of clusters
3. Merge it into a parent cluster
4. Repeat



Agglomerative clustering

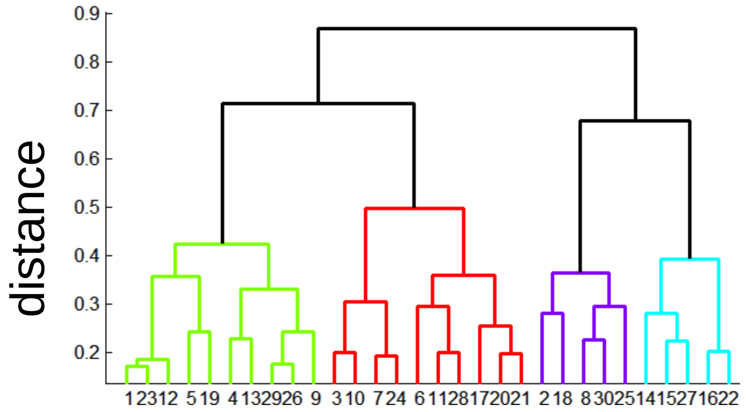


How to define cluster similarity?

- Average distance between points, maximum distance, minimum distance
- Distance between means or medoids

How many clusters?

- Clustering creates a dendrogram (a tree)
- Threshold based on max number of clusters or based on distance between merges



Conclusions: Agglomerative Clustering

Good

- Simple to implement, widespread application
- Clusters have adaptive shapes
- Provides a hierarchy of clusters

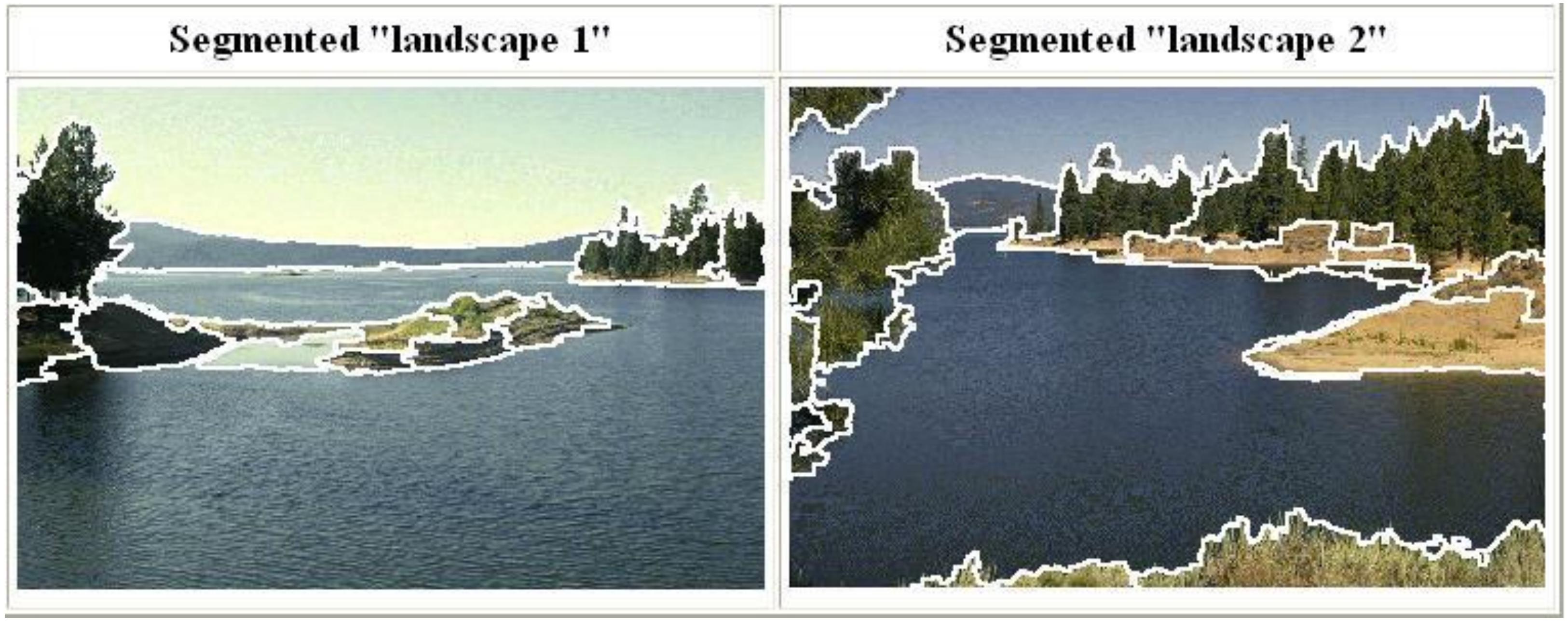
Bad

- May have imbalanced clusters
- Still have to choose number of clusters or threshold
- Need to use an “ultrametric” to get a meaningful hierarchy

Mean shift segmentation

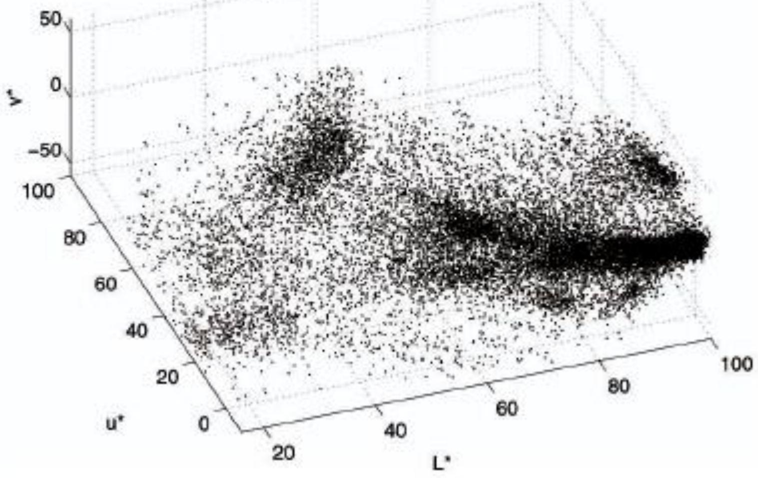
D. Comaniciu and P. Meer, Mean Shift: A Robust Approach toward Feature Space Analysis, PAMI 2002.

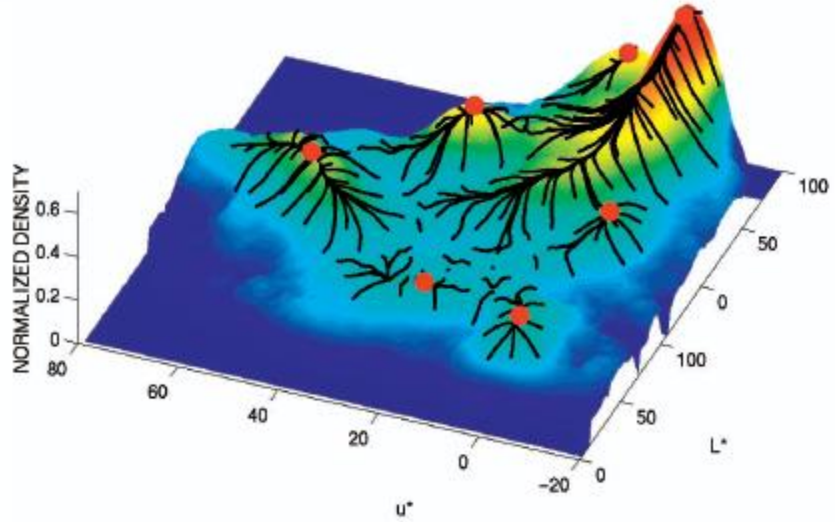
- Versatile technique for clustering-based segmentation

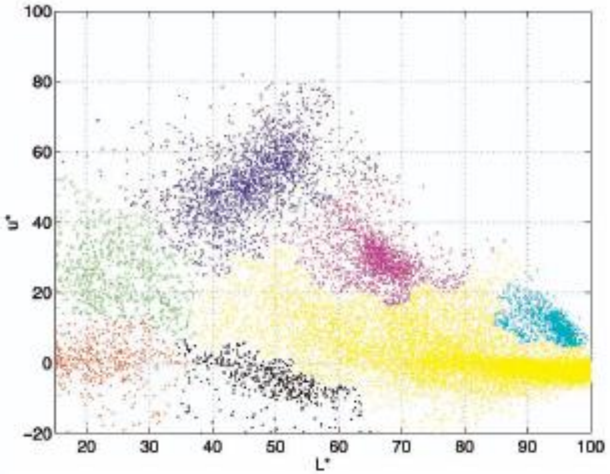


Mean shift algorithm

- Try to find *modes* of this non-parametric density







Kernel density estimation

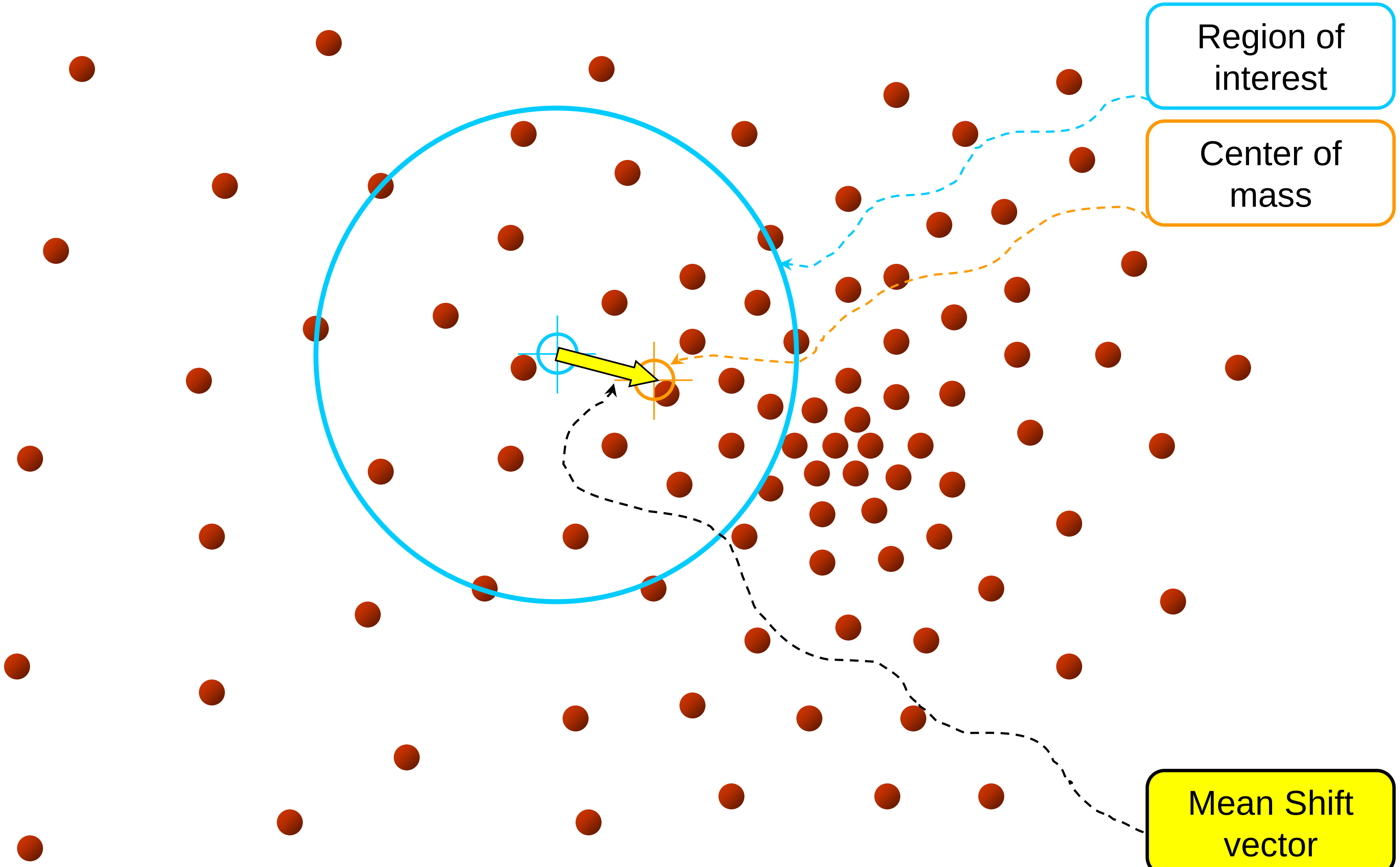
Kernel density estimation function

$$\widehat{f}_h(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

Gaussian kernel

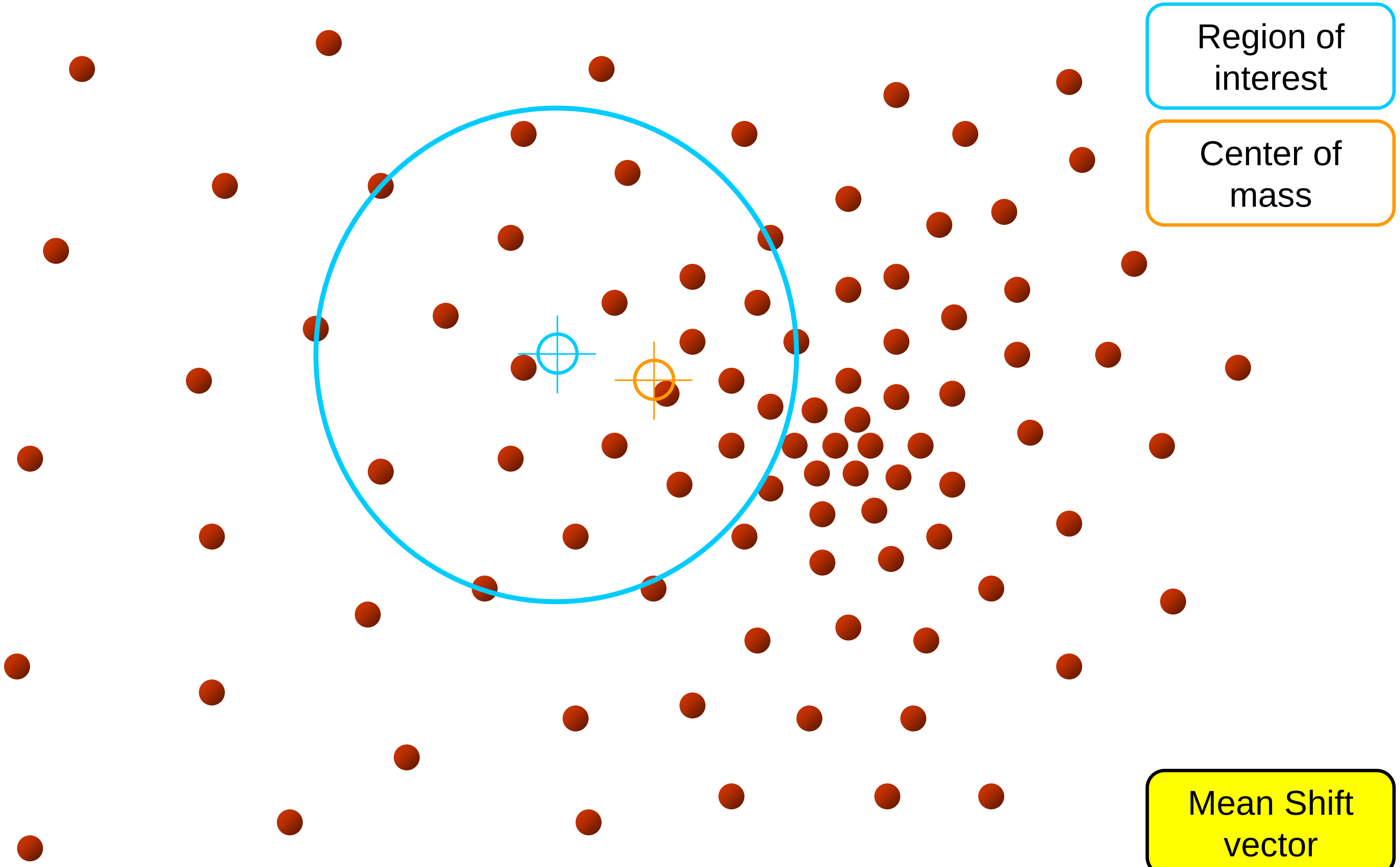
$$K\left(\frac{x - x_i}{h}\right) = \frac{1}{\sqrt{2\pi}}\, e^{-\frac{(x-x_i)^2}{2h^2}}.$$

Mean shift



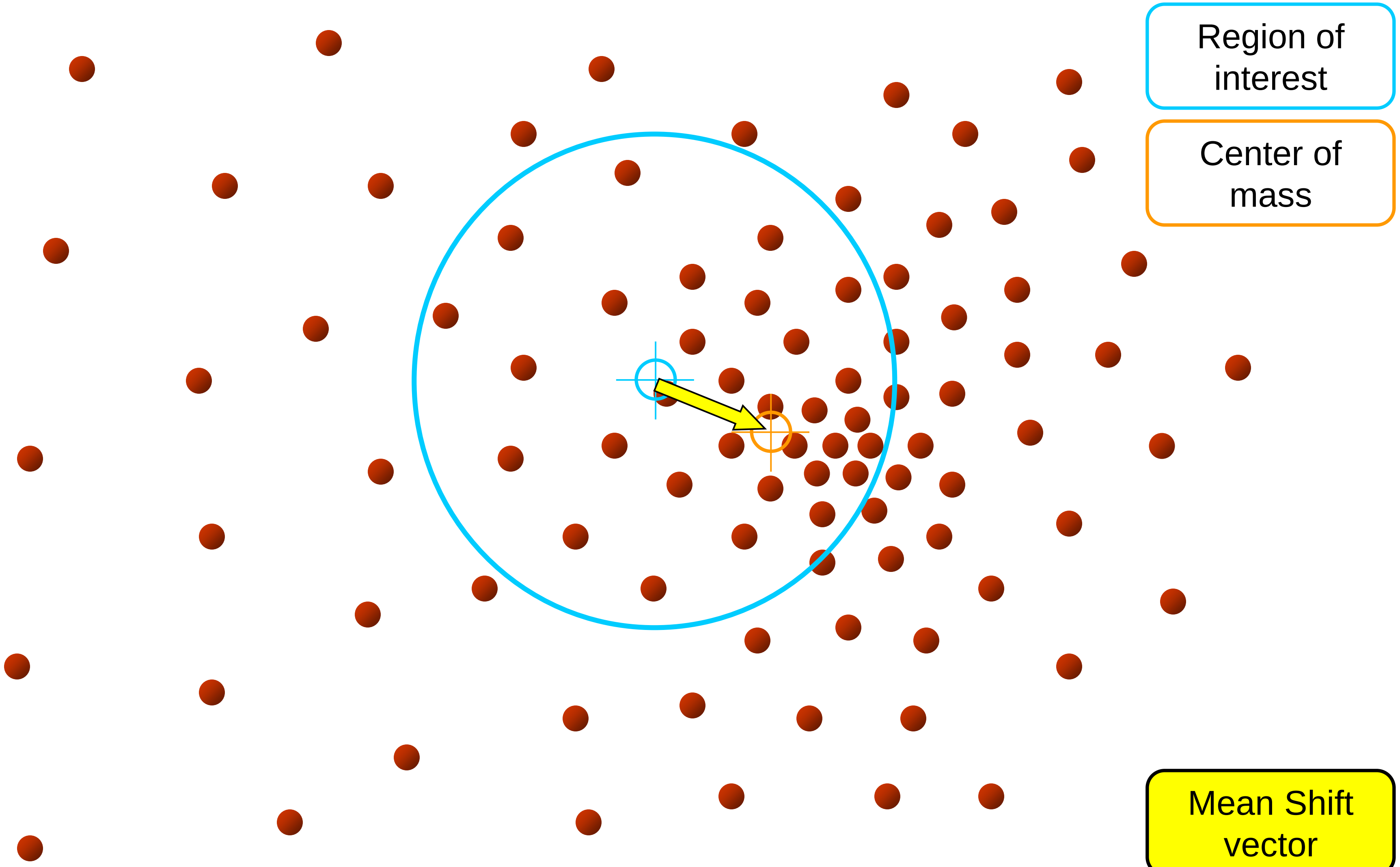
Slide by Y. Ukrainitz & B. Sarel

Mean shift



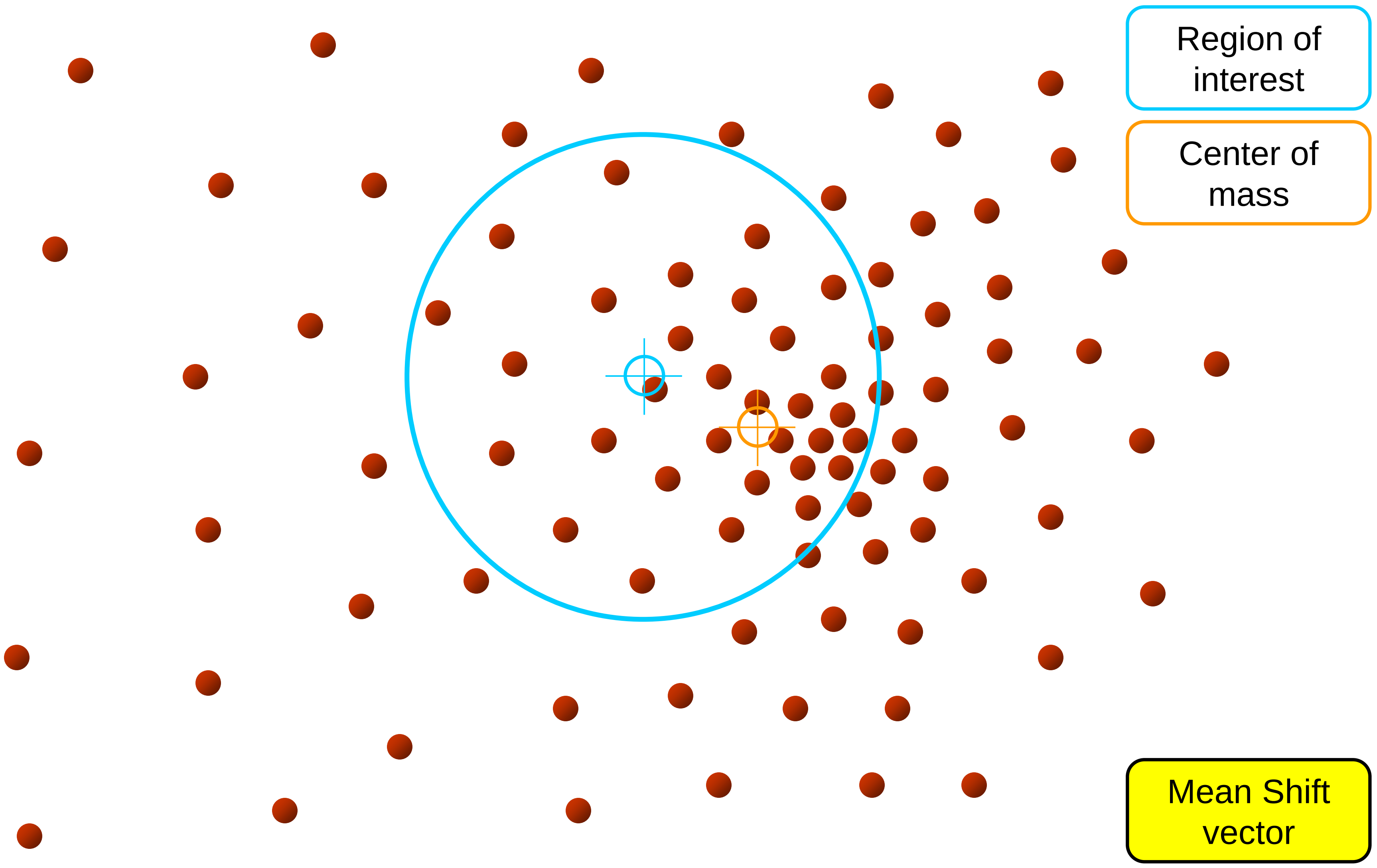
Slide by Y. Ukrainitz & B. Sarel

Mean shift



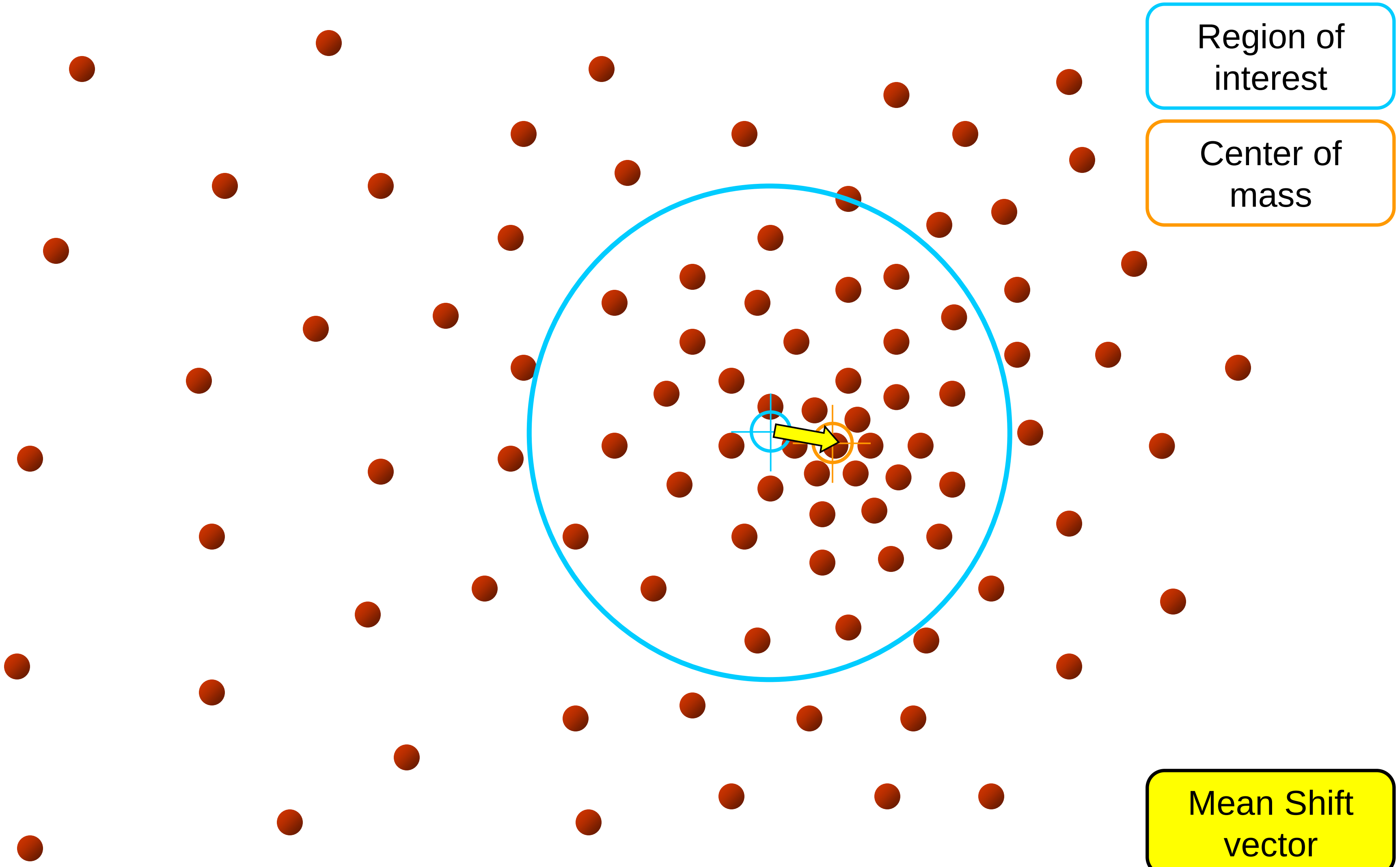
Slide by Y. Ukrainitz & B. Sarel

Mean shift



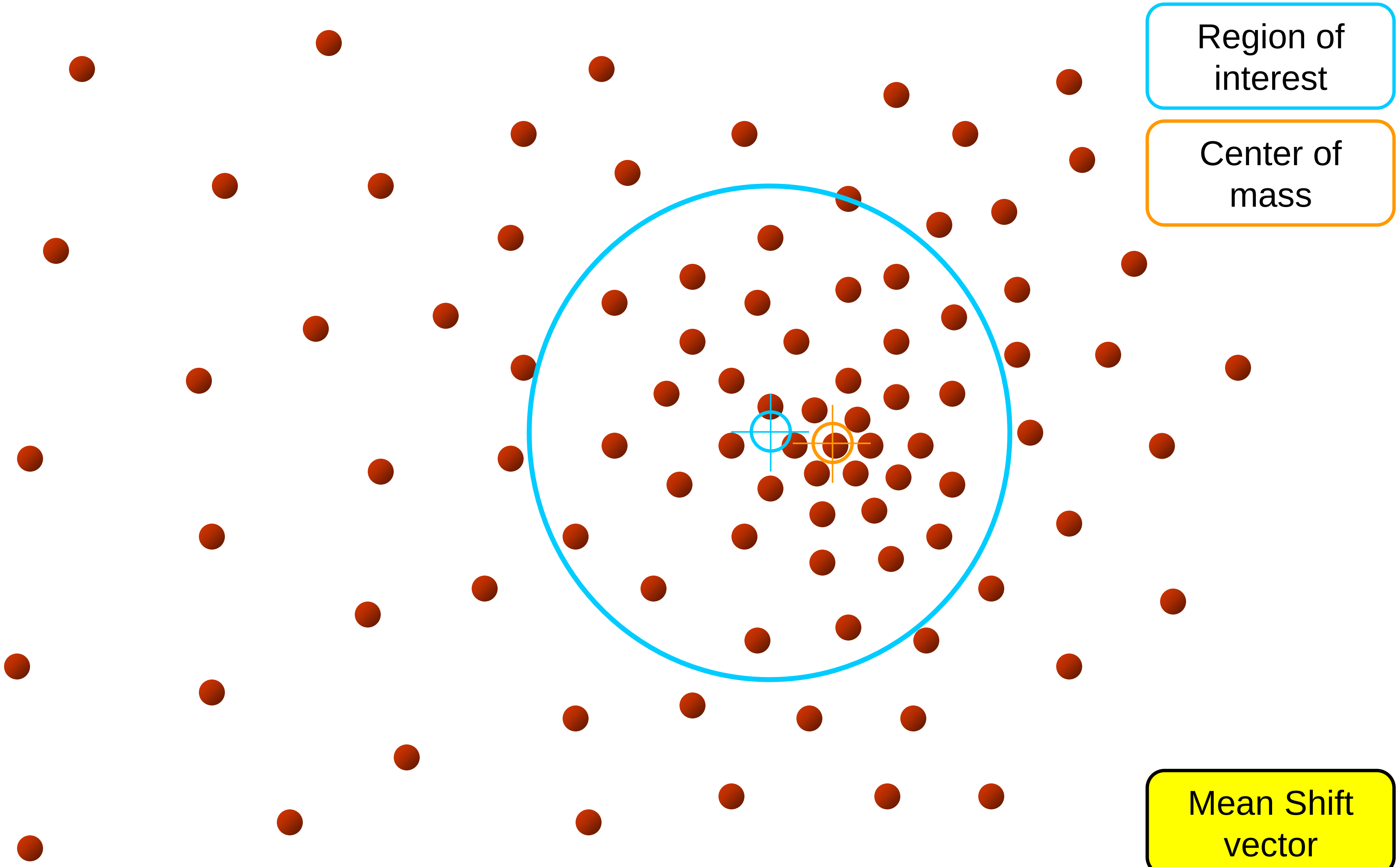
Slide by Y. Ukrainitz & B. Sarel

Mean shift



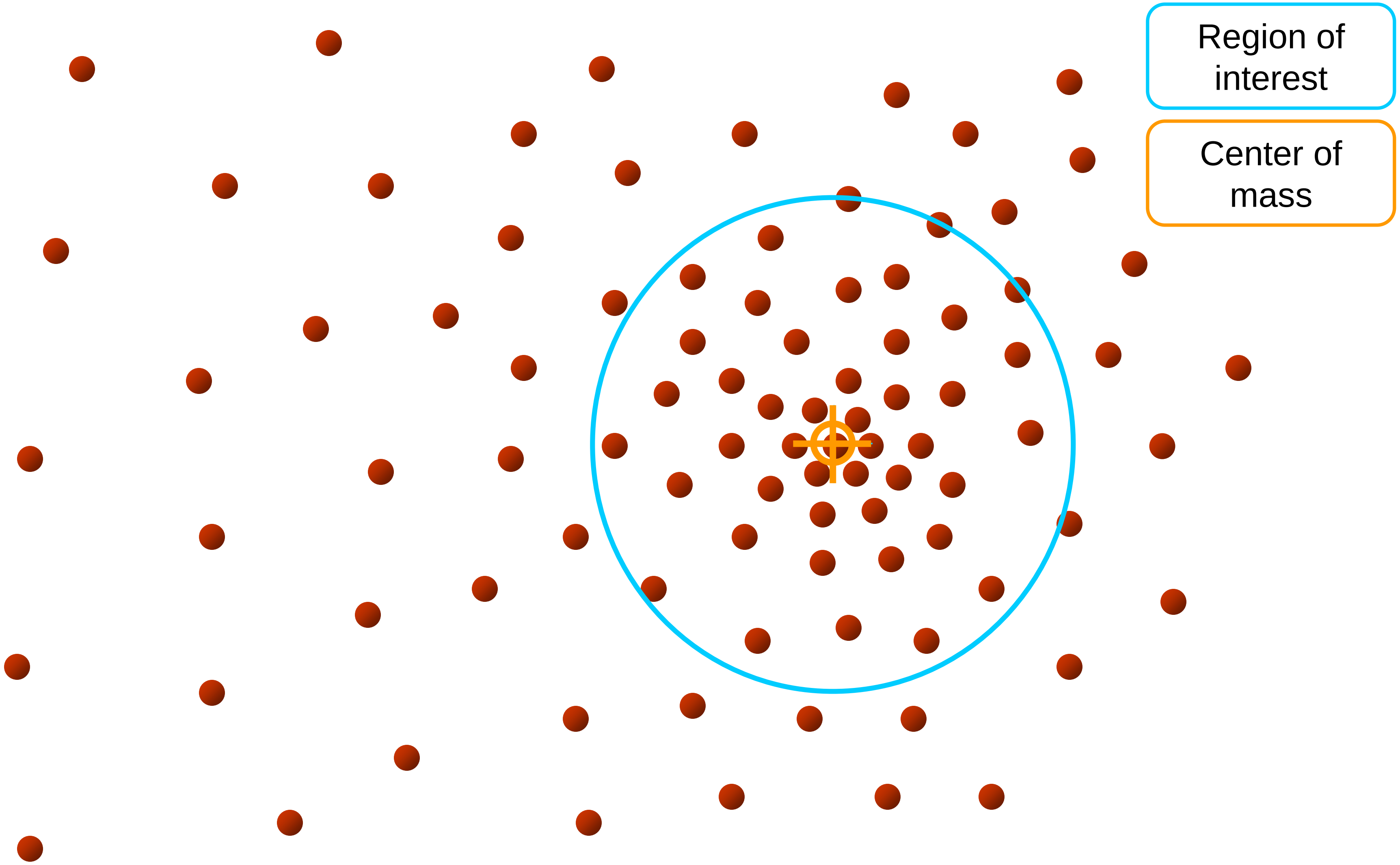
Slide by Y. Ukrainitz & B. Sarel

Mean shift



Slide by Y. Ukrainitz & B. Sarel

Mean shift



Slide by Y. Ukrainitz & B. Sarel

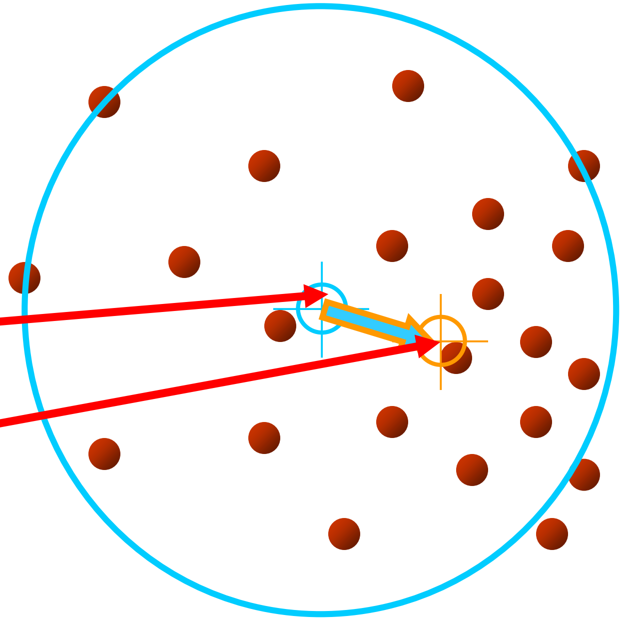
Computing the Mean Shift

Simple Mean Shift procedure:

- Compute mean shift vector

- Translate the Kernel window by **m(x)**

$$\mathbf{m}(\mathbf{x}) = \left[\frac{\sum_{i=1}^{n} \mathbf{x}_i g\left(\frac{\left\| \mathbf{x} - \mathbf{x}_i \right\|^2}{h} \right)}{\sum_{i=1}^{n} g\left(\frac{\left\| \mathbf{x} - \mathbf{x}_i \right\|^2}{h} \right)} - \boxed{\mathbf{x}} \right]$$

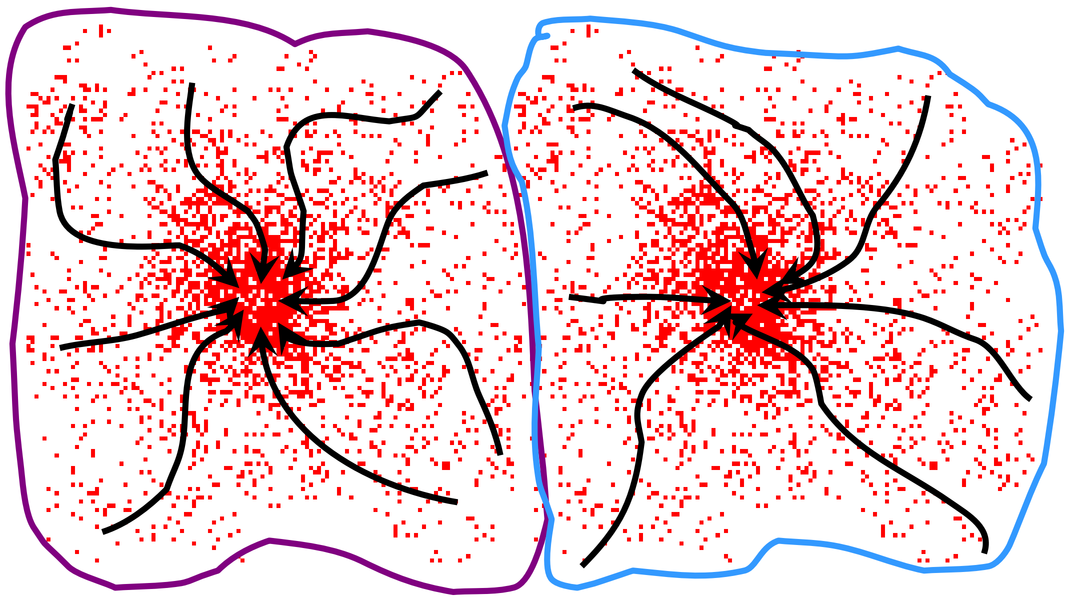


$$g(\mathbf{x}) = -k'(\mathbf{x})$$

Slide by Y. Ukrainitz & B. Sarel

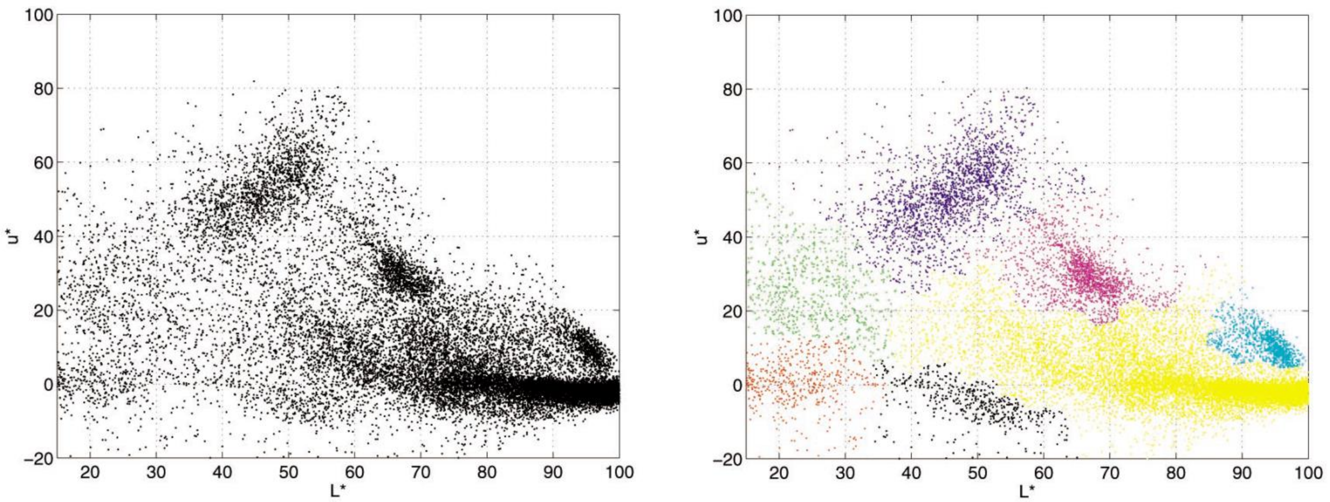
Attraction basin

- Attraction basin: the region for which all trajectories lead to the same mode
- Cluster: all data points in the attraction basin of a mode

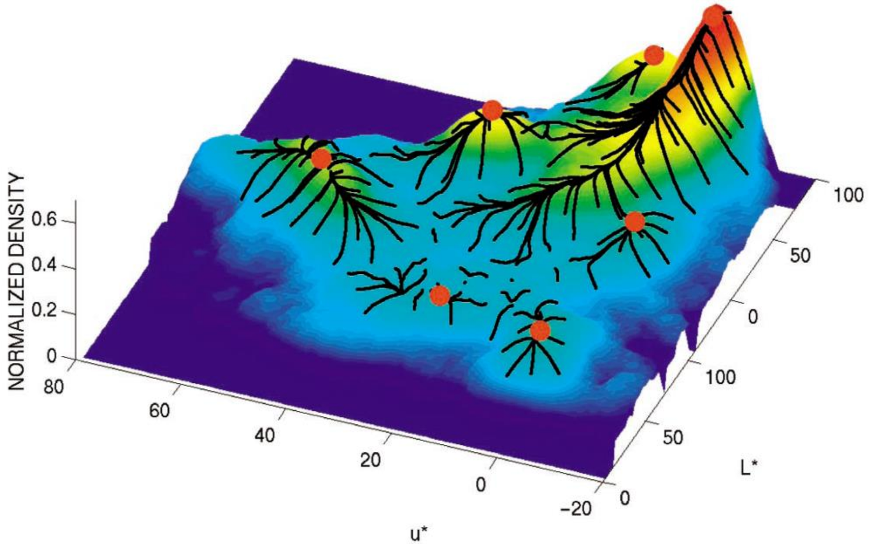


Slide by Y. Ukrainitz & B. Sarel

Attraction basin



(a) (b)

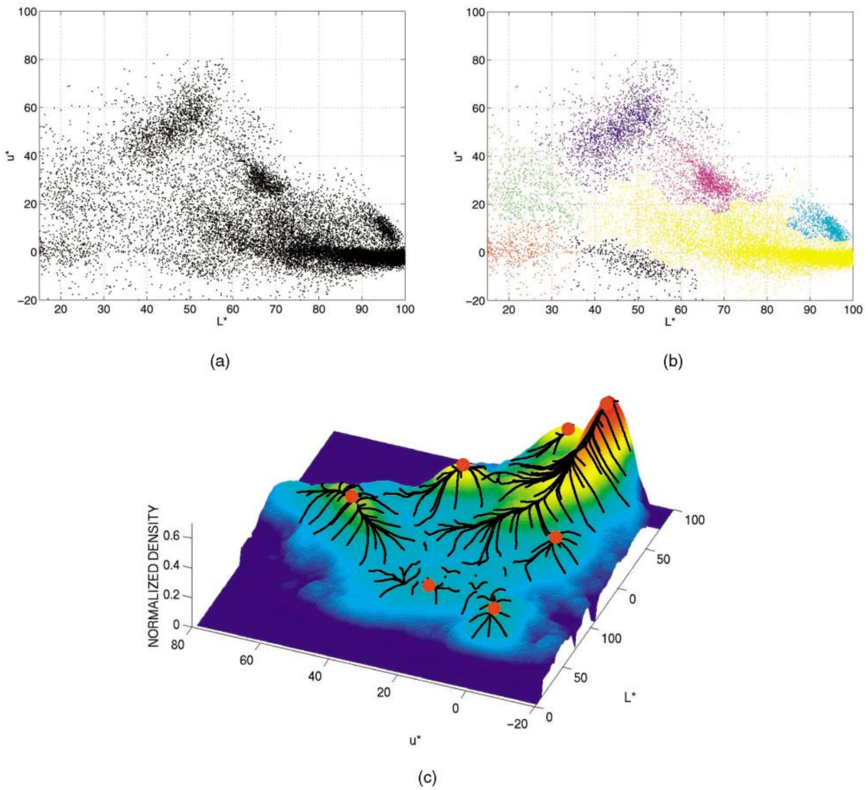


Mean shift clustering

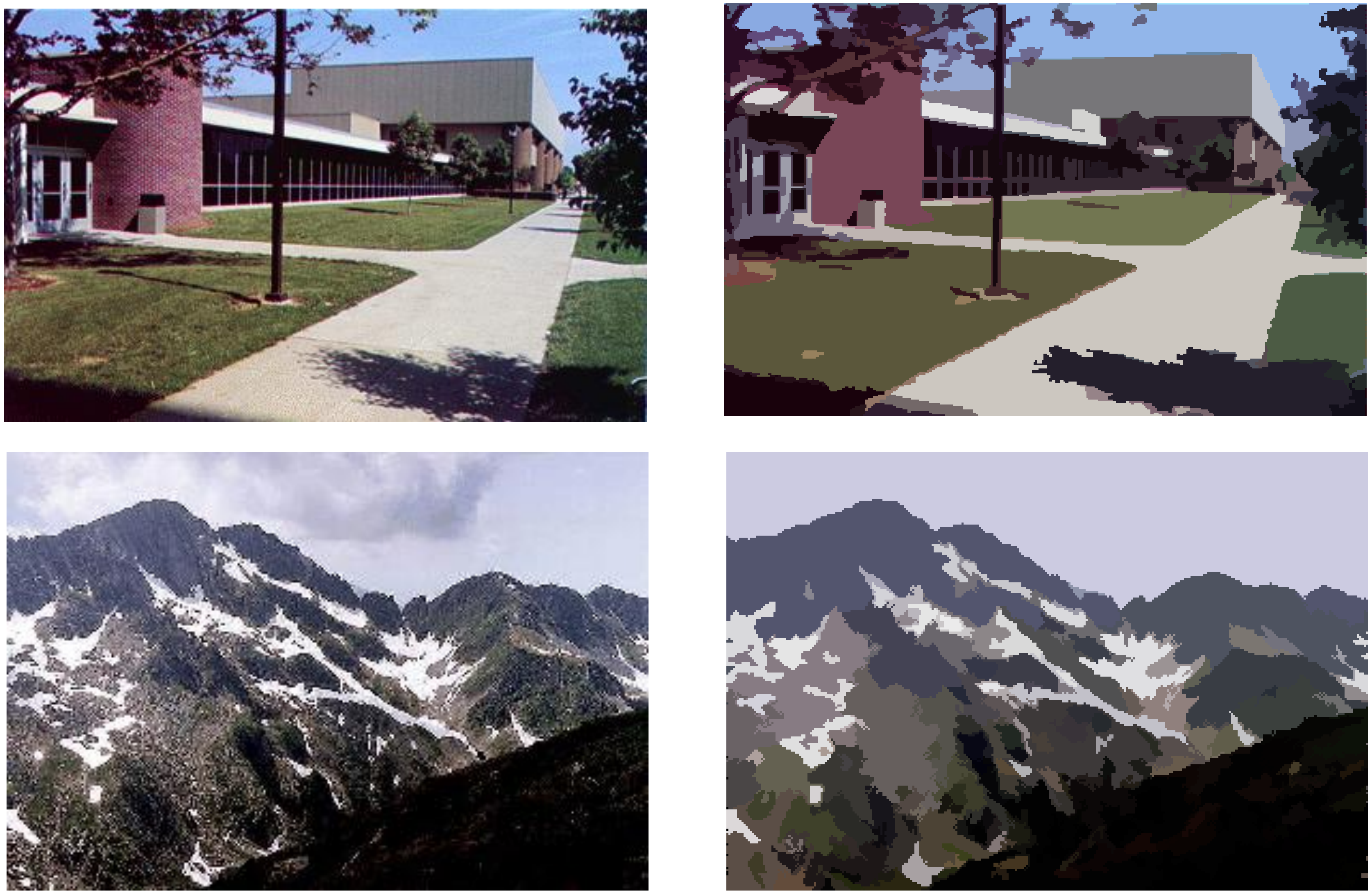
- The mean shift algorithm seeks *modes* of the given set of points
  1. Choose kernel and bandwidth
  2. For each point:
     a) Center a window on that point
     b) Compute the mean of the data in the search window
     c) Center the search window at the new mean location
     d) Repeat (b,c) until convergence
  3. Assign points that lead to nearby modes to the same cluster

Segmentation by Mean Shift

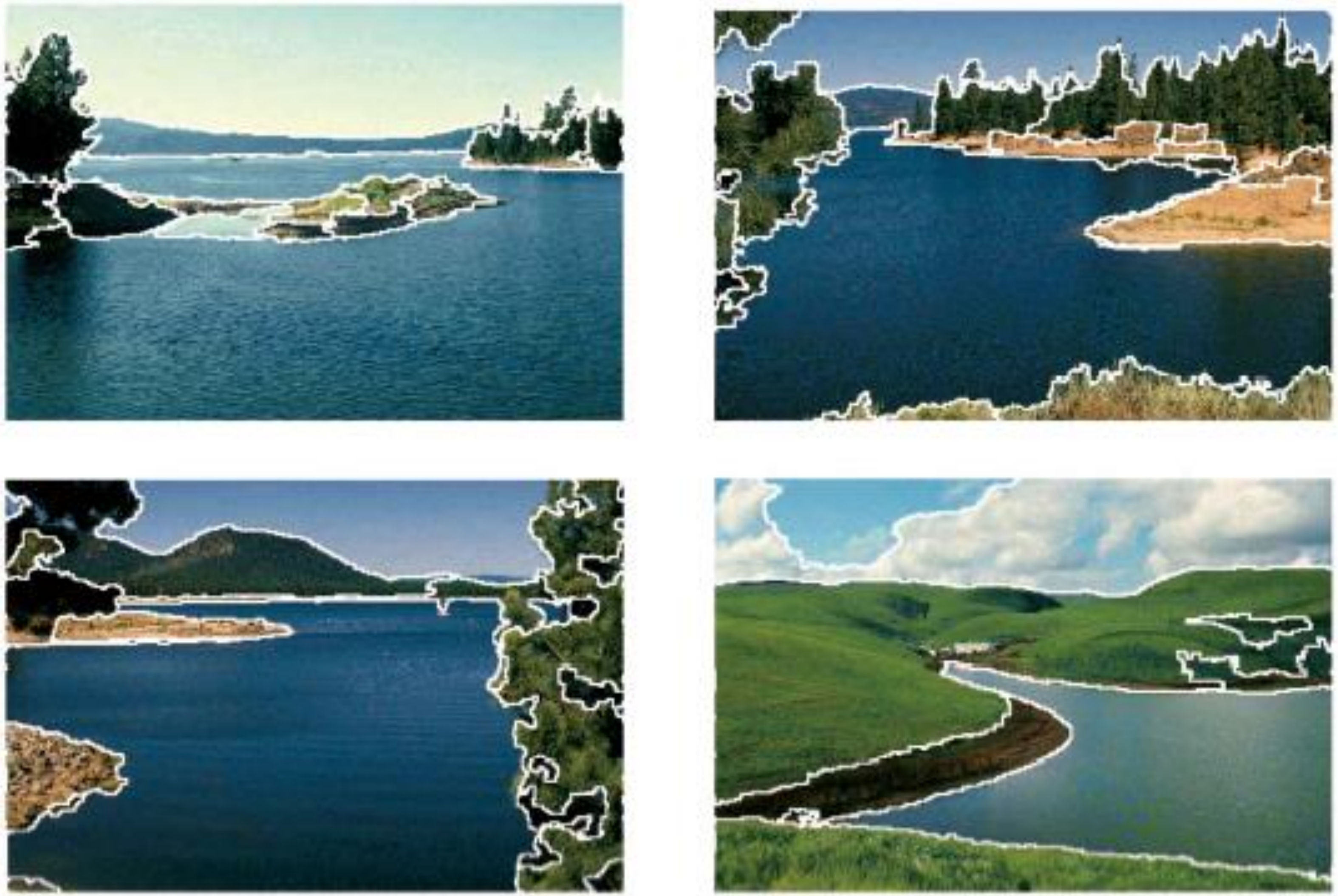
- Compute features for each pixel (color, gradients, texture, etc)
- Set kernel size for features K_f and position K_s
- Initialize windows at individual pixel locations
- Perform mean shift for each window until convergence
- Merge windows that are within width of K_f and K_s



Mean shift segmentation results



Comaniciu and Meer 2002



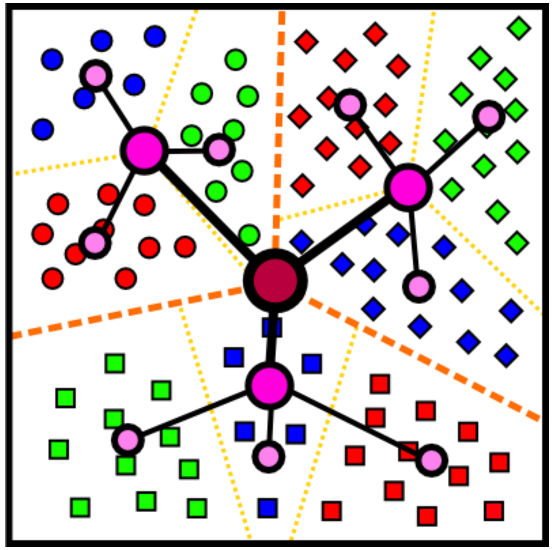
Comaniciu and Meer 2002

Mean shift pros and cons

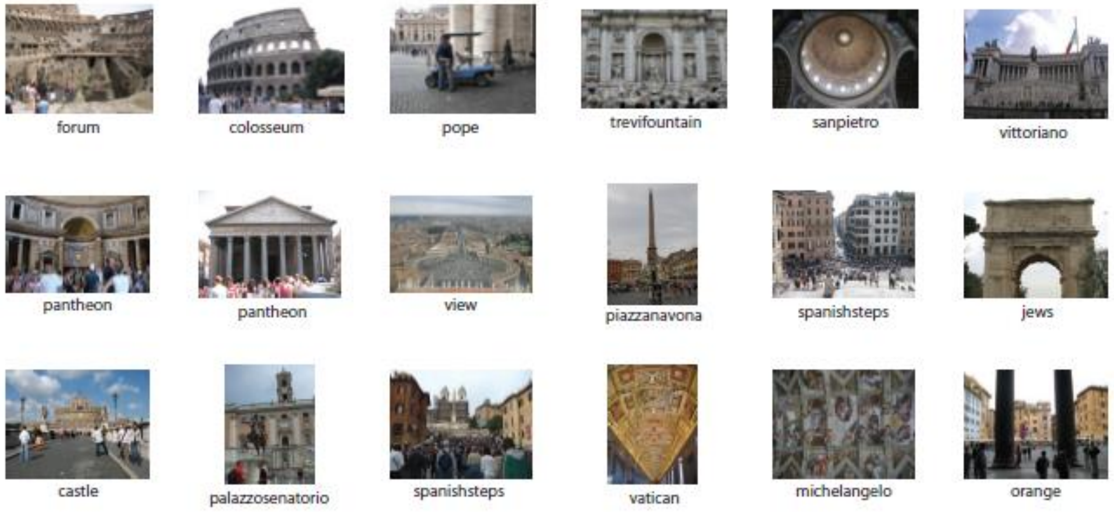
- Pros
  - Good general-practice segmentation
  - Flexible in number and shape of regions
  - Robust to outliers
- Cons
  - Have to choose kernel size in advance
  - Not suitable for high-dimensional features
- When to use it
  - Oversegmentatoin
  - Multiple segmentations
  - Tracking, clustering, filtering applications

Which algorithm to try first?

- Quantization/Summarization: K-means
  - Aims to preserve variance of original data
  - Can easily assign new point to a cluster



Quantization for computing histograms



Summary of 20,000 photos of Rome using "greedy k-means"
http://grail.cs.washington.edu/projects/canonview/

Machine Learning Problems

	Supervised Learning	***Unsupervised Learning***
Discrete	classification or categorization	clustering
Continuous	regression	dimensionality reduction

The machine learning framework

- Apply a prediction function to a feature representation of the image to get the desired output:

f() = “apple”

f() = “tomato”

f() = “cow”

Slide credit: L. Lazebnik

The machine learning framework

$$y = f(\mathbf{x})$$

output

prediction function

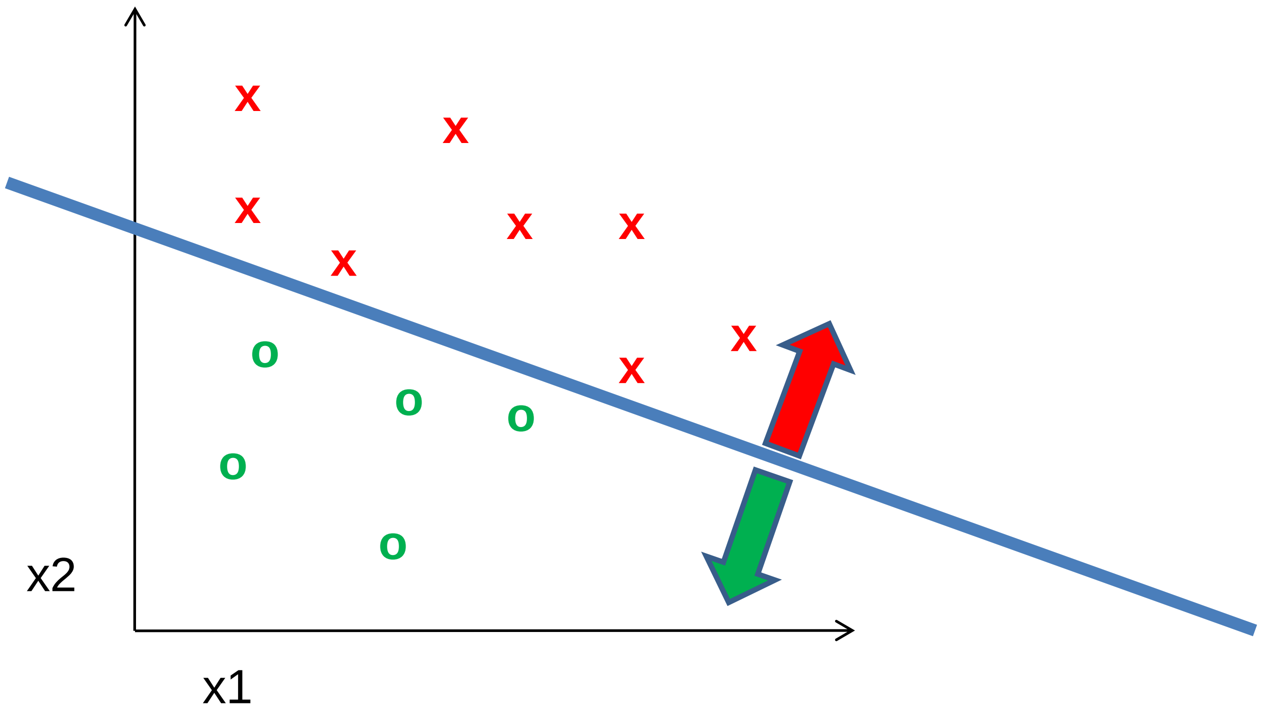
Image or feature

- **Training:** given a *training set* of labeled examples $\{(\mathbf{x}_1,y_1), \ldots, (\mathbf{x}_N,y_N)\}$, estimate the prediction function f by minimizing the prediction error on the training set
- **Testing:** apply f to a never before seen *test example* $\mathbf{x}$ and output the predicted value $y = f(\mathbf{x})$

Slide credit: L. Lazebnik

Learning a classifier

Given some set of features with corresponding labels, learn a function to predict the labels from the features



Steps

Training

Training Images

Image Features → Training → Learned model

Training Labels → Training

Testing

Test Image

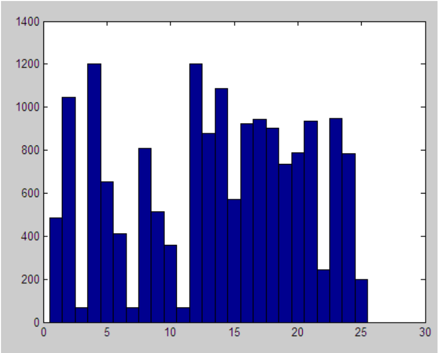
Image Features → Learned model → Prediction

Slide credit: D. Hoiem and L. Lazebnik

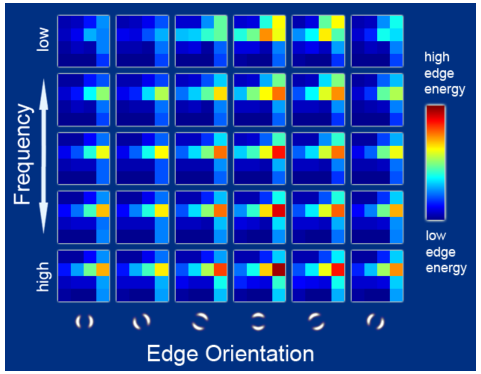
Features

- Raw pixels

- Histograms



- GIST descriptors



- …

Slide credit: L. Lazebnik

Many classifiers to choose from

- SVM
- Neural networks
- Naïve Bayes
- Bayesian network
- Logistic regression
- Randomized Forests
- Boosted Decision Trees
- K-nearest neighbor
- RBMs
- Deep Convolutional Network
- Attentional models or “Transformers”
- Etc.

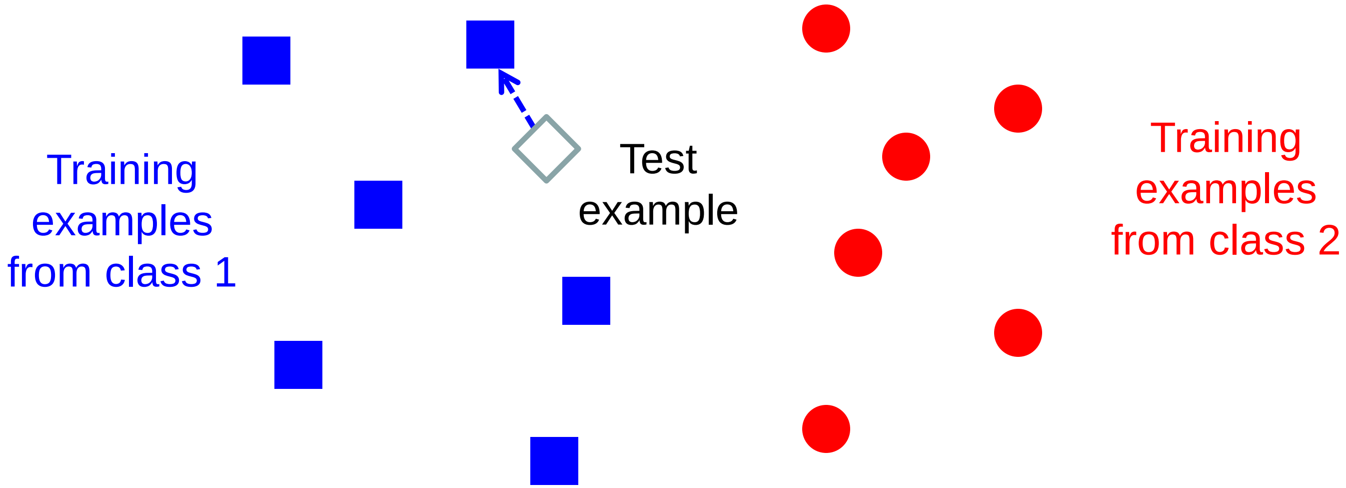
Which is the best one?

Claim:

The decision to *use* machine learning is more important than the choice of a *particular* learning method.

*Deep learning seems to be an exception to this, at the moment, probably because it is learning the feature representation.

Classifiers: Nearest neighbor

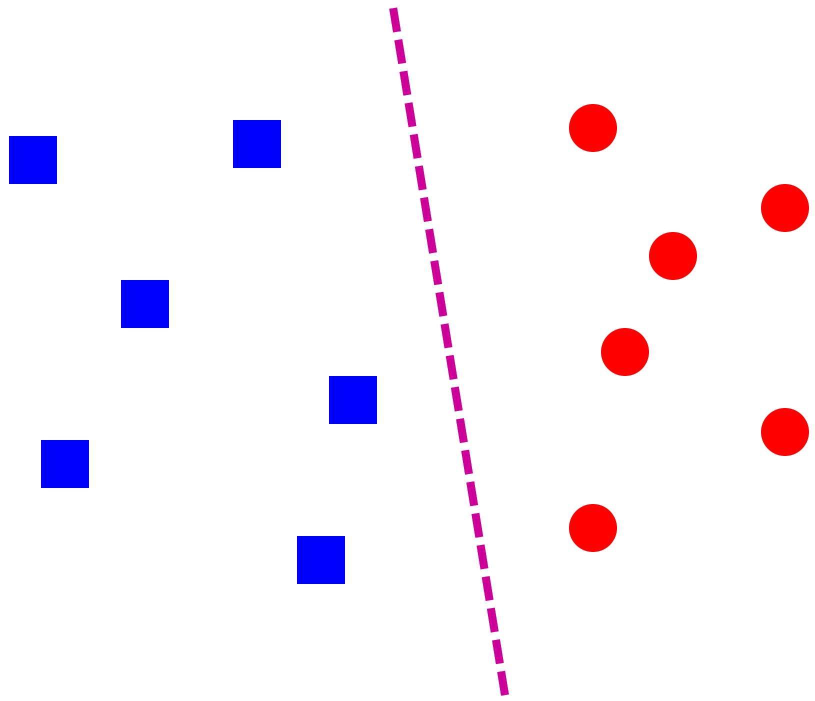


f(**x**) = label of the training example nearest to **x**

- All we need is a distance function for our inputs
- No training required!

Slide credit: L. Lazebnik

Classifiers: Linear



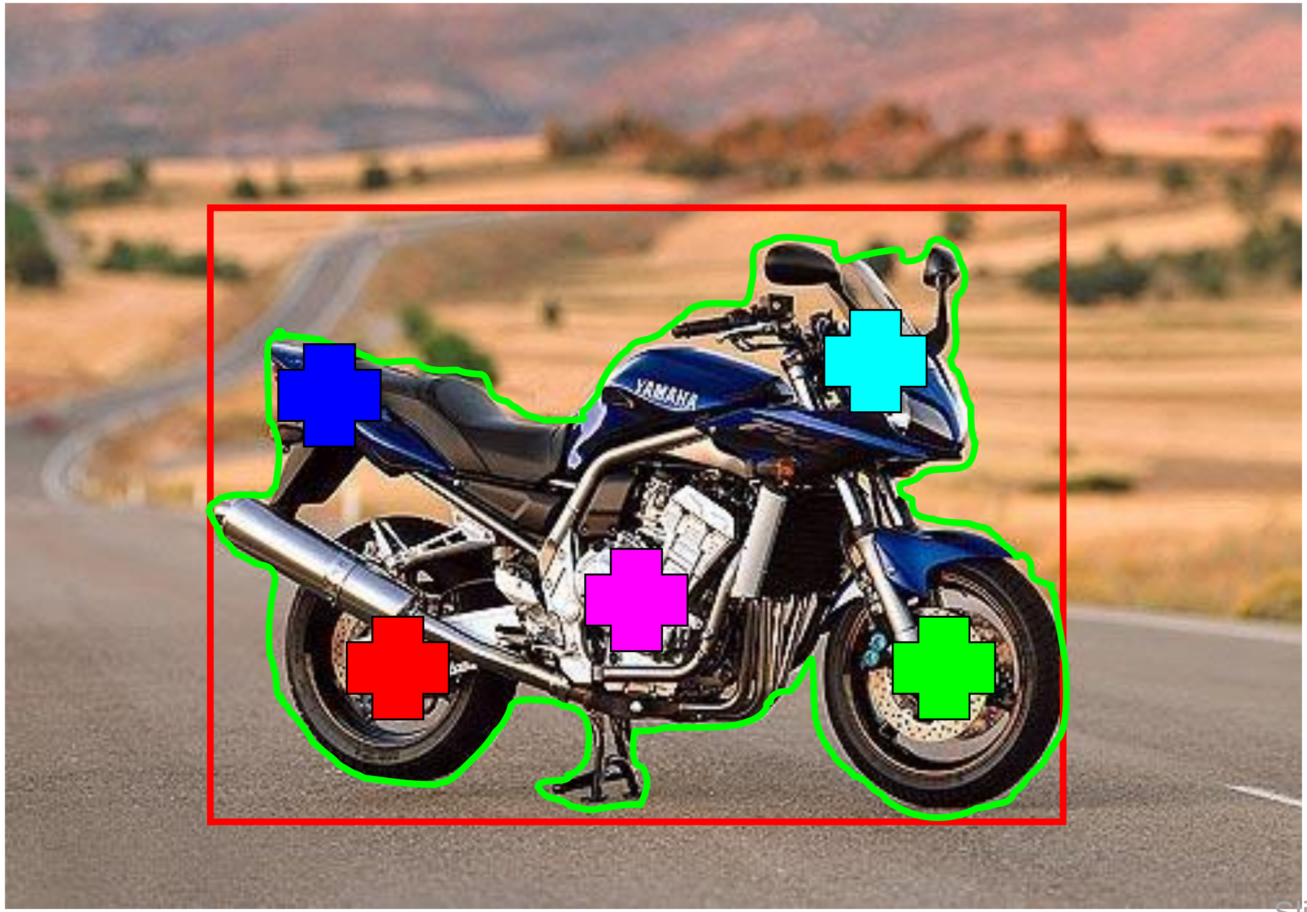
- Find a *linear function* to separate the classes:

$$f(\mathbf{x}) = \mathrm{sgn}(\mathbf{w} \cdot \mathbf{x} + b)$$

Recognition task and supervision

- Images in the training set must be annotated with the “correct answer” that the model is expected to produce

Contains a motorbike



Slide credit: L. Lazebnik

Spectrum of supervision

Less **More**

Unsupervised

"Weakly" supervised

Fully supervised

Definition depends on task

Slide credit: L. Lazebnik

Generalization

Training set (labels known)

Test set (labels unknown)

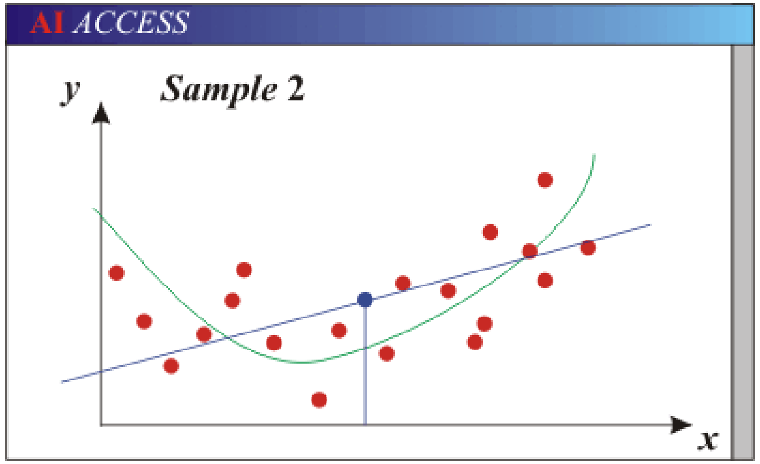
- How well does a learned model generalize from the data it was trained on to a new test set?

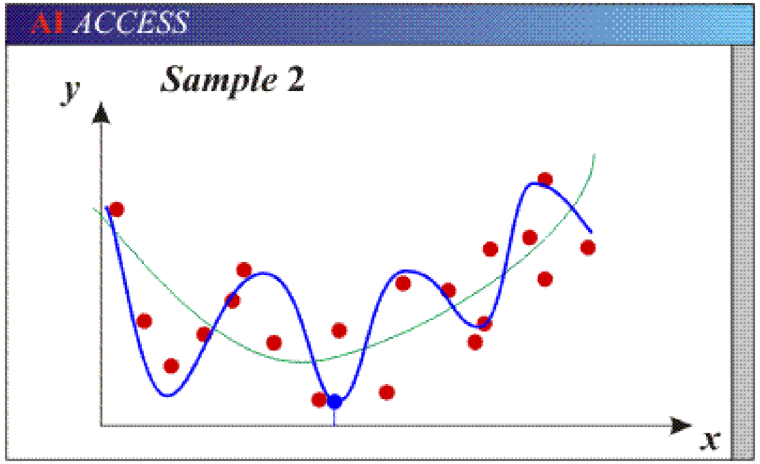
Generalization

- Components of generalization error
  - **Bias:** how much the average model over all training sets differ from the true model?
    - Error due to inaccurate assumptions/simplifications made by the model. “Bias” sounds negative. “Regularization” sounds nicer.
  - **Variance:** how much models estimated from different training sets differ from each other.
- **Underfitting:** model is too “simple” to represent all the relevant class characteristics
  - High bias (few degrees of freedom) and low variance
  - High training error and high test error
- **Overfitting:** model is too “complex” and fits irrelevant characteristics (noise) in the data
  - Low bias (many degrees of freedom) and high variance
  - Low training error and high test error

Slide credit: L. Lazebnik

Bias-Variance Trade-off

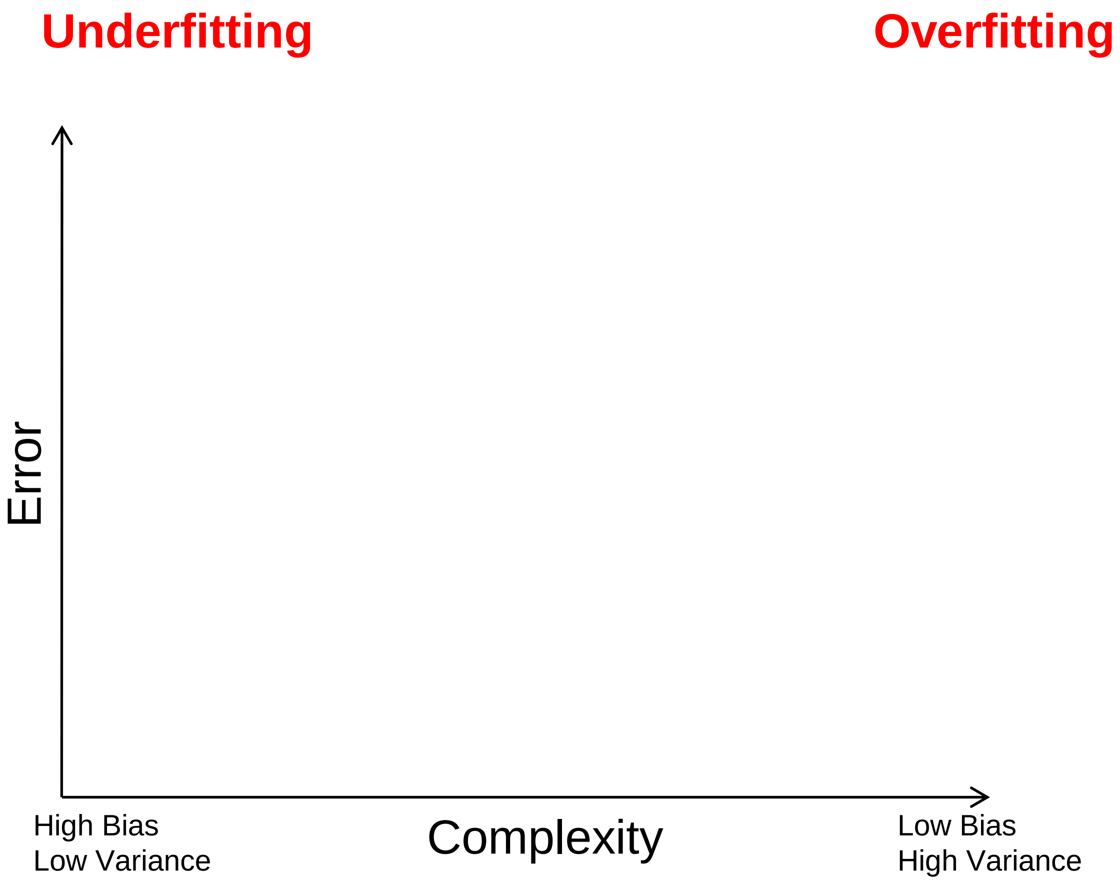




- Models with too few parameters are inaccurate because of a large bias (not enough flexibility).
- Models with too many parameters are inaccurate because of a large variance (too much sensitivity to the sample).

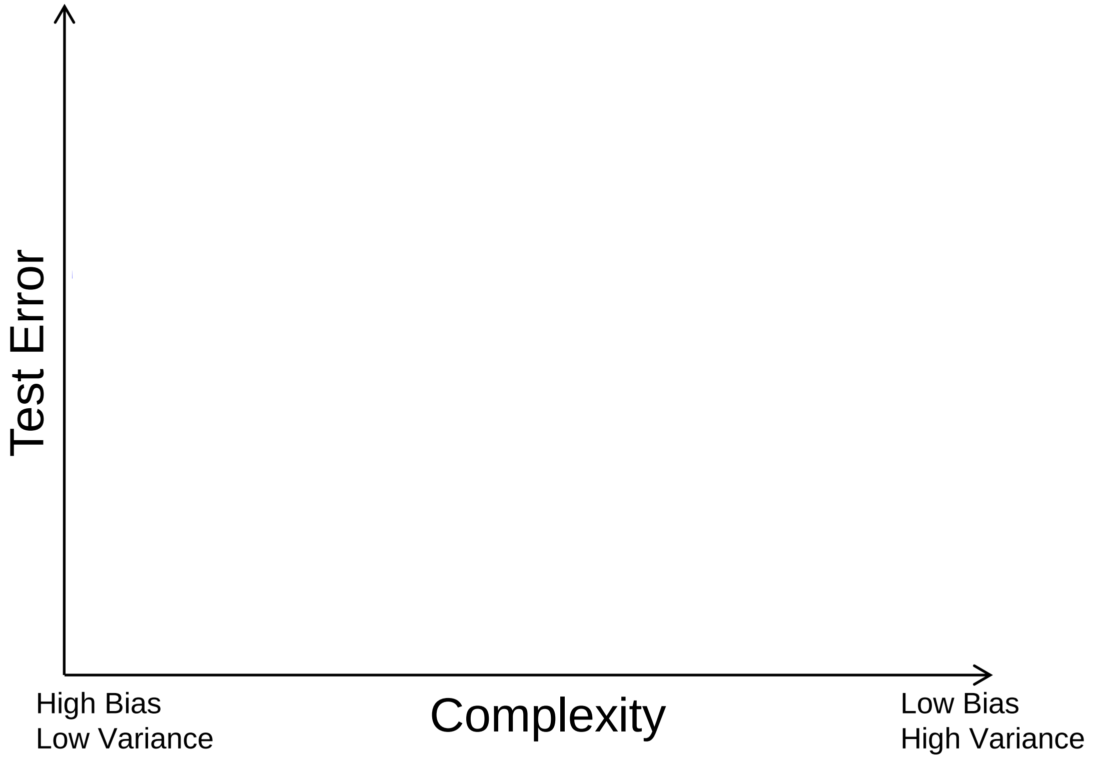
Slide credit: D. Hoiem

Bias-variance tradeoff



Slide credit: D. Hoiem

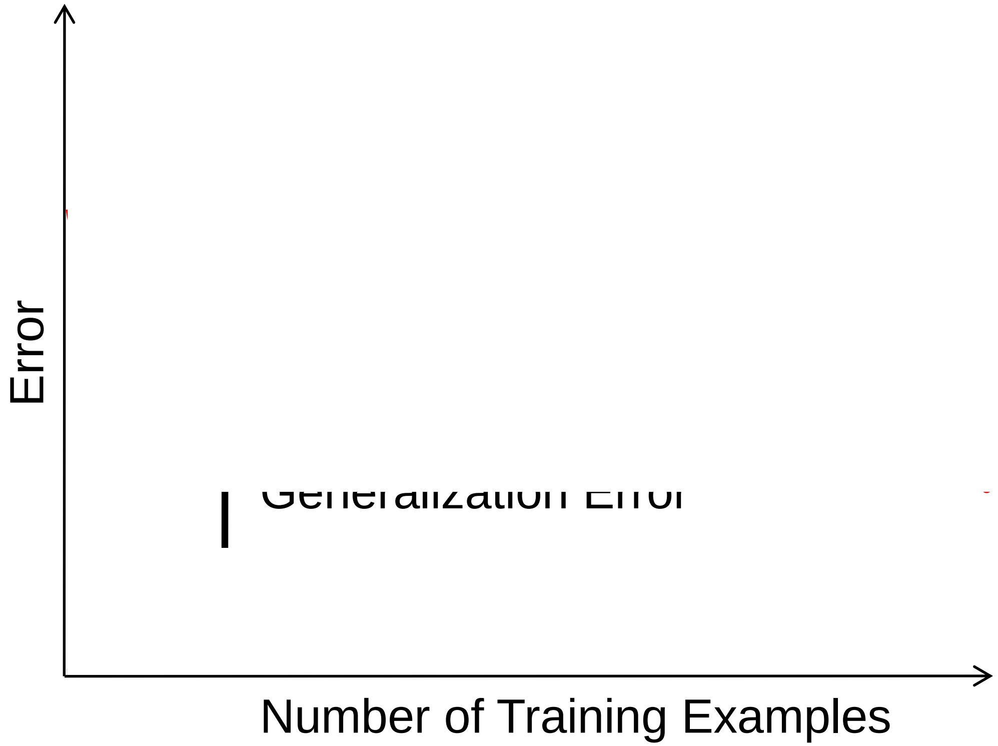
Bias-variance tradeoff



Slide credit: D. Hoiem

Effect of Training Size

Fixed prediction model



Remember…

- No classifier is inherently better than any other: you need to make assumptions to generalize

- Three kinds of error
  - Inherent: unavoidable
  - Bias: due to over-simplifications
  - Variance: due to inability to perfectly estimate parameters from limited data

- How to reduce variance?
  - Choose a simpler classifier
  - Regularize the parameters
  - Get more training data
- How to reduce bias?
  - Choose a more complex, more expressive classifier
  - Remove regularization
  - (These might not be safe to do unless you get more training data)

Slide credit: D. Hoiem

To be continued…