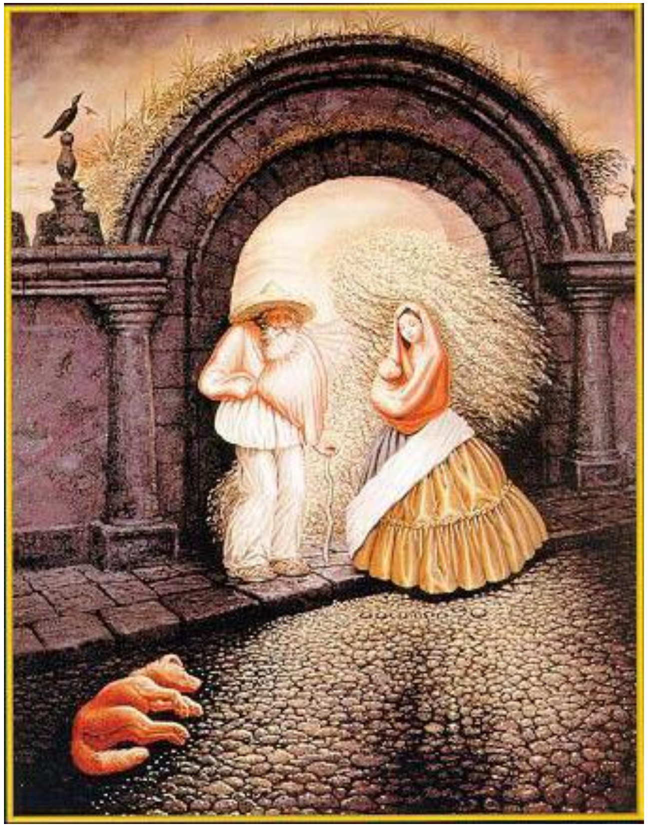
Image Segmentation



Computer Vision

Jia-Bin Huang, Virginia Tech

Many slides from D. Hoiem

Administrative stuffs

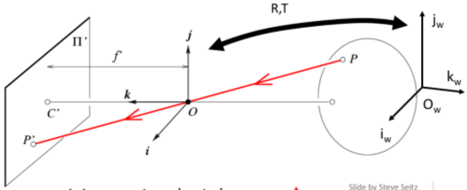
- HW 3 due 11:59 PM, Oct 17 (Monday)
- Final project proposal due Oct 27 (Thursday)
- Will hold office hour this Friday

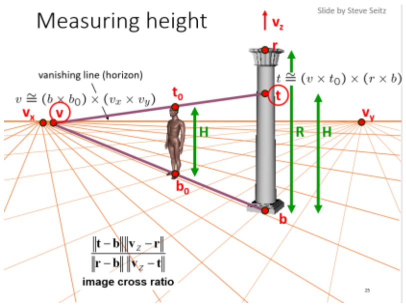
Today's class

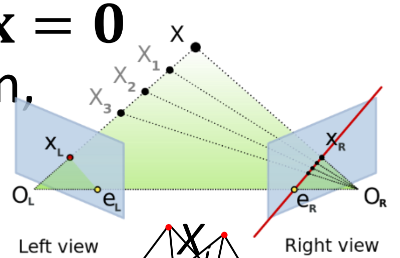
- Review/finish Structure from motion
  - Multi-view stereo
- Segmentation and grouping
  - Gestalt cues
  - By clustering (k-means, mean-shift)
  - By boundaries (watershed)
  - By graph (merging , graph cuts)
  - By labeling (MRF) <- Next Thursday
- Superpixels and multiple segmentations

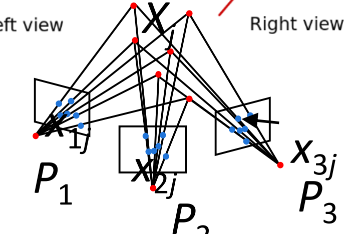
Perspective and 3D Geometry

- **Projective geometry and camera models**
  - Vanishing points/lines
  - $\mathrm{x} = \mathbf{K}[\mathbf{R}\ \ \mathbf{t}]\mathbf{X}$
- **Single-view metrology and camera calibration**
  - Calibration using known 3D object or vanishing points
  - Measuring size using perspective cues
- **Photo stitching**
  - Homography relates rotating cameras $\mathbf{x}' = \mathbf{H}\mathbf{x}$
  - Recover homography using RANSAC + normalized DLT
- **Epipolar Geometry and Stereo Vision**
  - Fundamental/essential matrix relates two cameras $\mathbf{x}'\mathbf{F}\mathbf{x} = \mathbf{0}$
  - Recover $\mathbf{F}$ using RANSAC + normalized 8-point algorithm, enforce rank 2 using SVD
- **Structure from motion**
  - Perspective SfM: triangulation, bundle adjustment
  - Affine SfM: factorization using SVD, enforce rank 3 constraints, resolve affine ambiguity







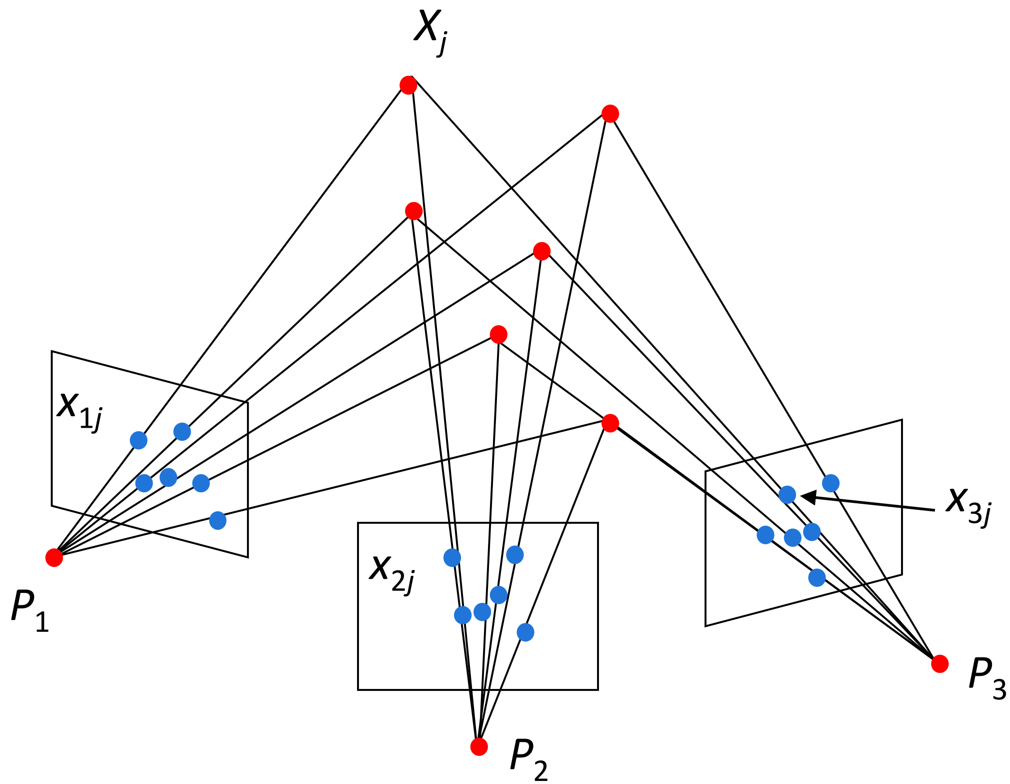


Review: Projective structure from motion

- Given: *m* images of *n* fixed 3D points

$$\mathbf{x}_{ij} = \mathbf{P}_i \mathbf{X}_j, \qquad i = 1, \ldots, m, \quad j = 1, \ldots, n$$

- Problem: estimate *m* projection matrices $\mathbf{P}_i$ and *n* 3D points $\mathbf{X}_j$ from the *mn* corresponding 2D points $\mathbf{x}_{ij}$



Slides: Lana Lazebnik

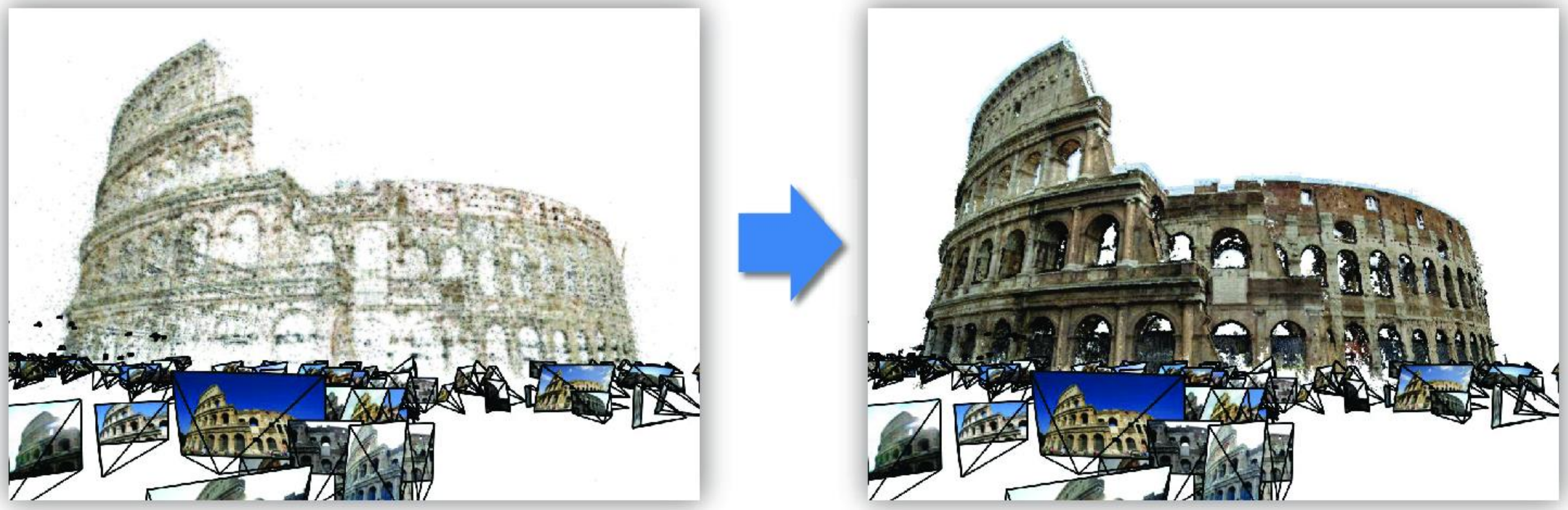
Review: Affine structure from motion

- Given: *m* images and *n* tracked features $\mathbf{x}_{ij}$
- For each image *i, c*enter the feature coordinates
- Construct a $2m \times n$ measurement matrix **D**:
  - Column *j* contains the projection of point *j* in all views
  - Row *i* contains one coordinate of the projections of all the *n* points in image *i*
- Factorize **D**:
  - Compute SVD: $\mathbf{D} = \mathbf{U}\ \mathbf{W}\ \mathbf{V}^{\mathbf{T}}$
  - Create $\mathbf{U}_3$ by taking the first 3 columns of **U**
  - Create $\mathbf{V}_3$ by taking the first 3 columns of **V**
  - Create $\mathbf{W}_3$ by taking the upper left 3 × 3 block of **W**
- Create the motion (affine) and shape (3D) matrices:

  $\mathbf{A} = \mathbf{U}_3\mathbf{W}_3^{½}$ and $\mathbf{S} = \mathbf{W}_3^{½}\ \mathbf{V}_3^{T}$
- Eliminate affine ambiguity
  - Solve $\mathbf{L} = \mathbf{C}\mathbf{C}^{\mathbf{T}}$ using metric constraints
  - Solve **C** using Cholesky decomposition
  - Update **A** and **X: A = AC, S = C^{-1}S**

Source: M. Hebert

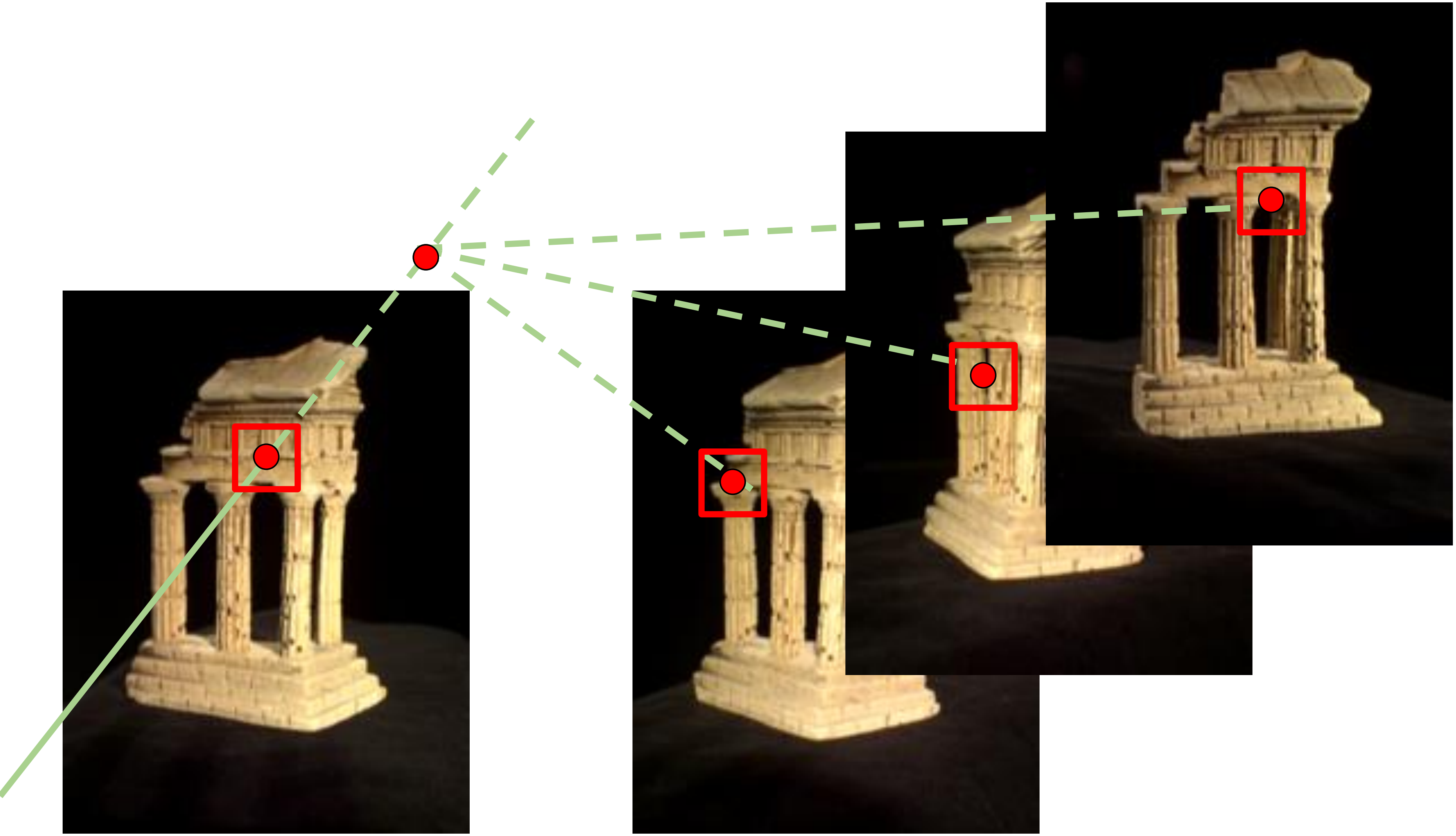
Multi-view stereo



Multi-view stereo

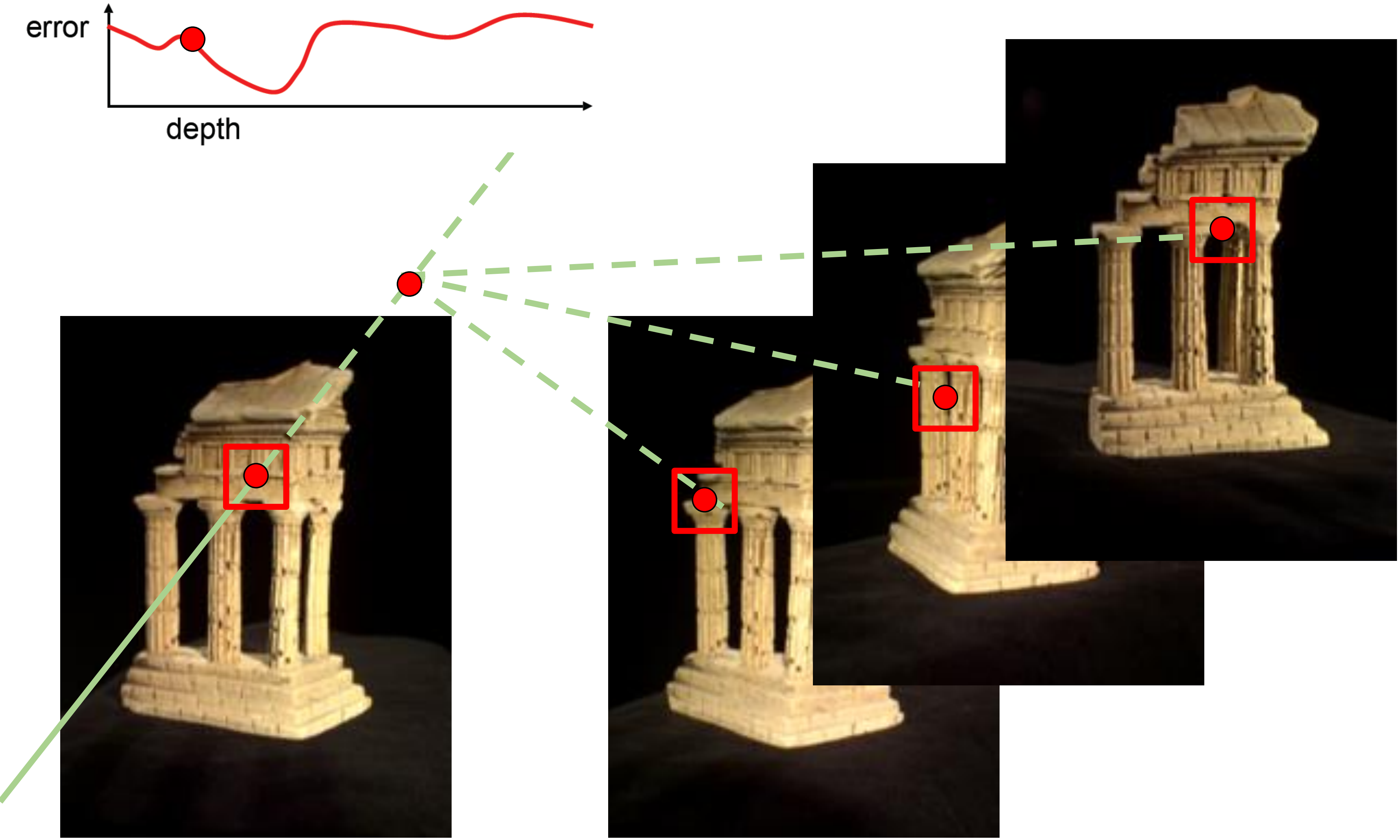
- Generic problem formulation: given several images of the same object or scene, compute a representation of its 3D shape
- “Images of the same object or scene”
  - Arbitrary number of images (from two to thousands)
  - Arbitrary camera positions (special rig, camera network or video sequence)
  - Calibration may be known or unknown
- “Representation of 3D shape”
  - Depth maps
  - Meshes
  - Point clouds
  - Patch clouds
  - Volumetric models
  - ….

Multi-view stereo: Basic idea



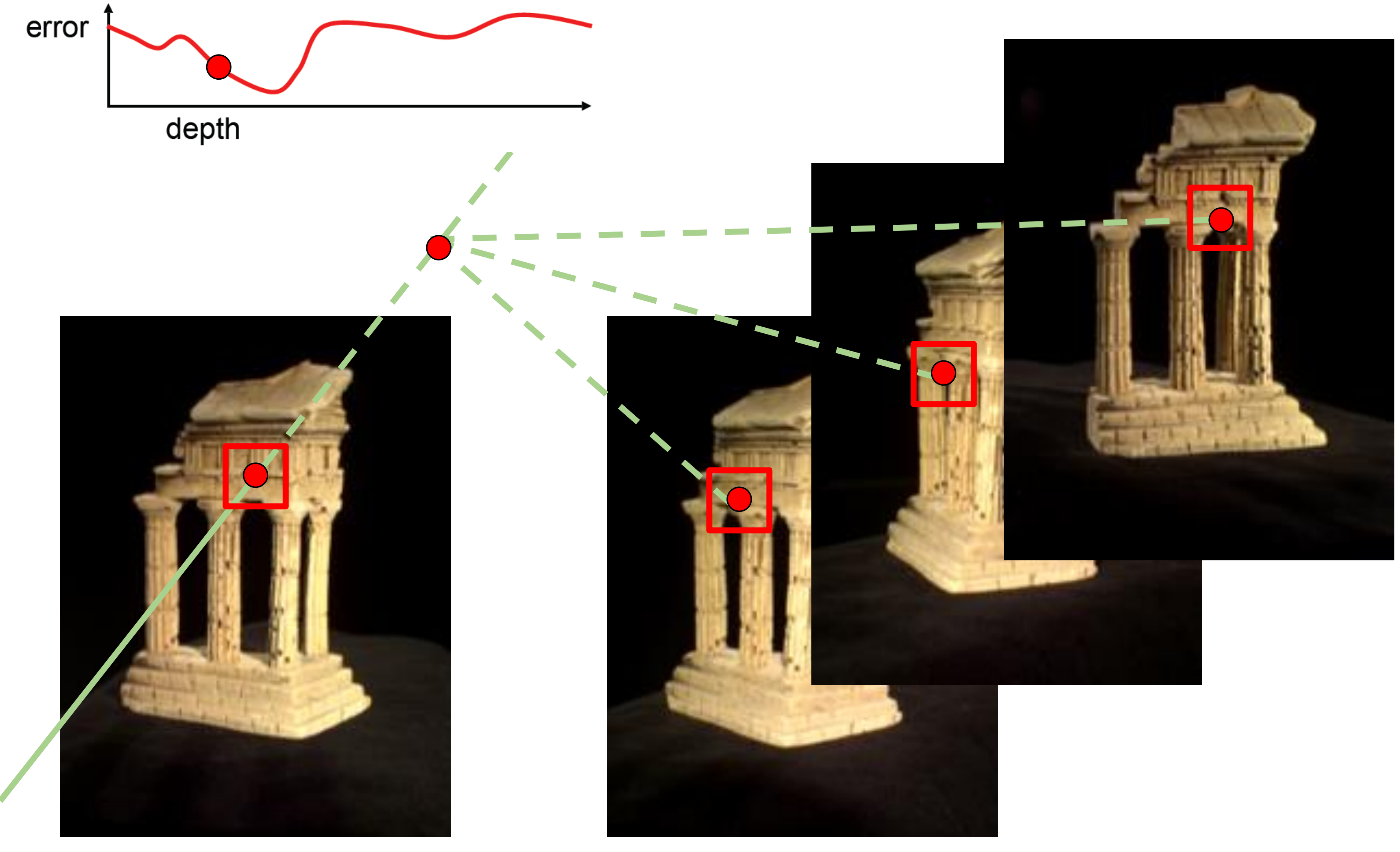
Source: Y. Furukawa

Multi-view stereo: Basic idea



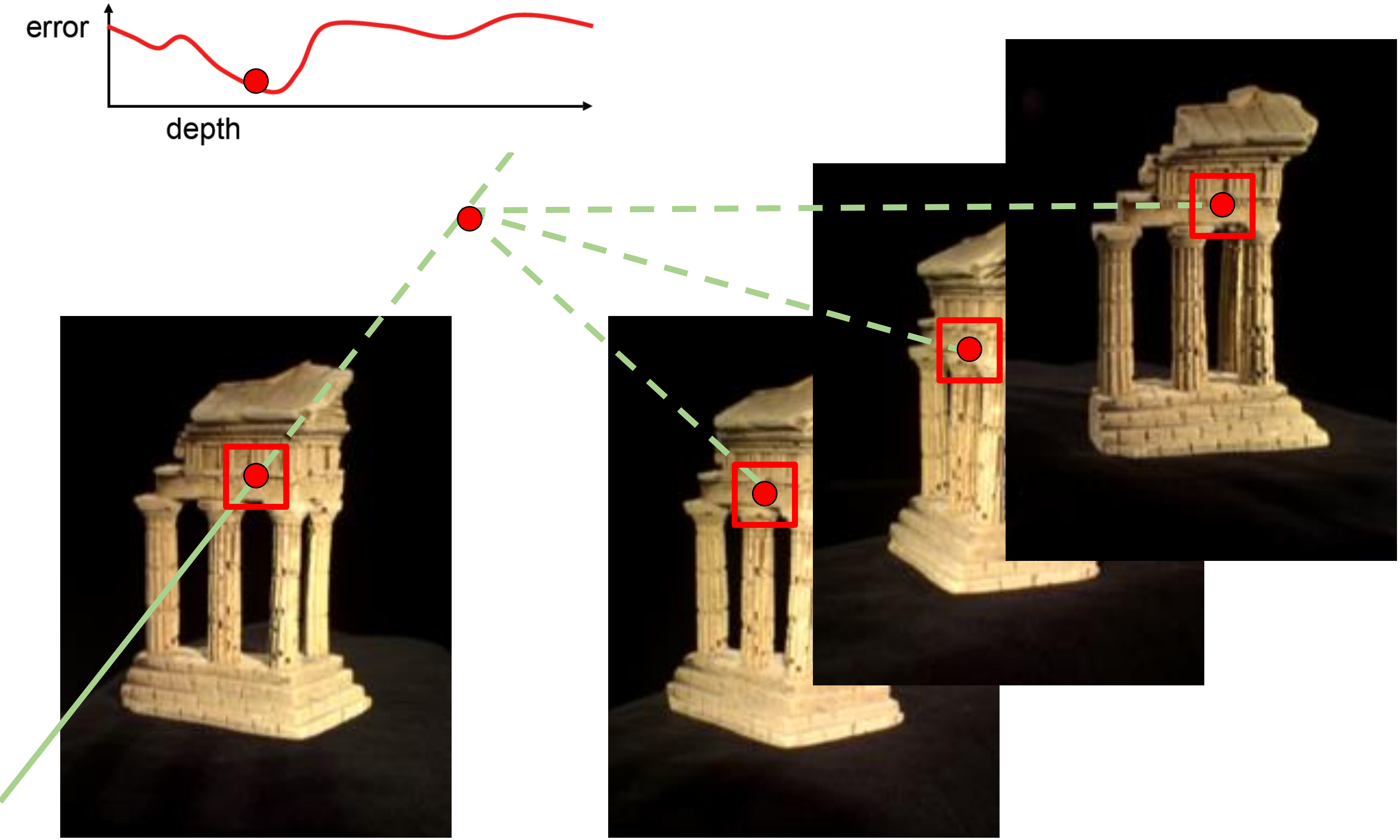
Source: Y. Furukawa

Multi-view stereo: Basic idea



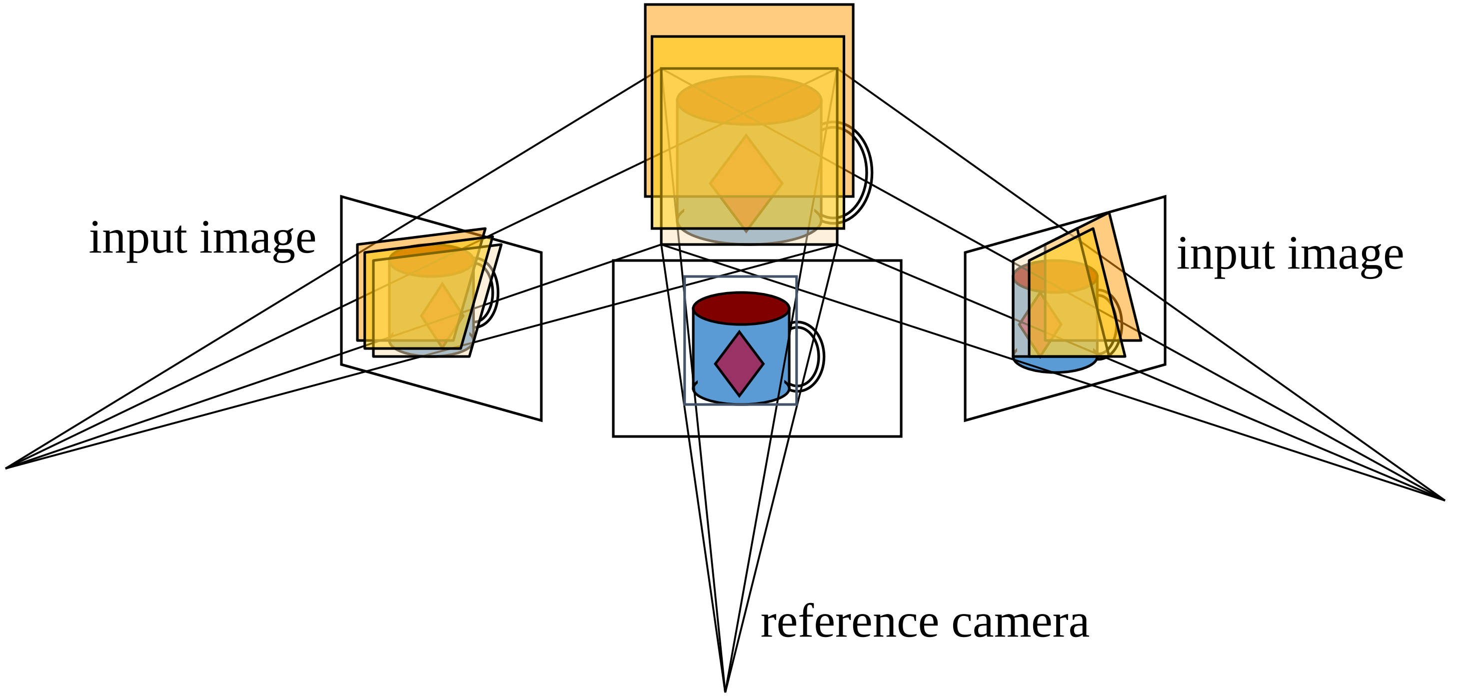
Source: Y. Furukawa

Multi-view stereo: Basic idea



Source: Y. Furukawa

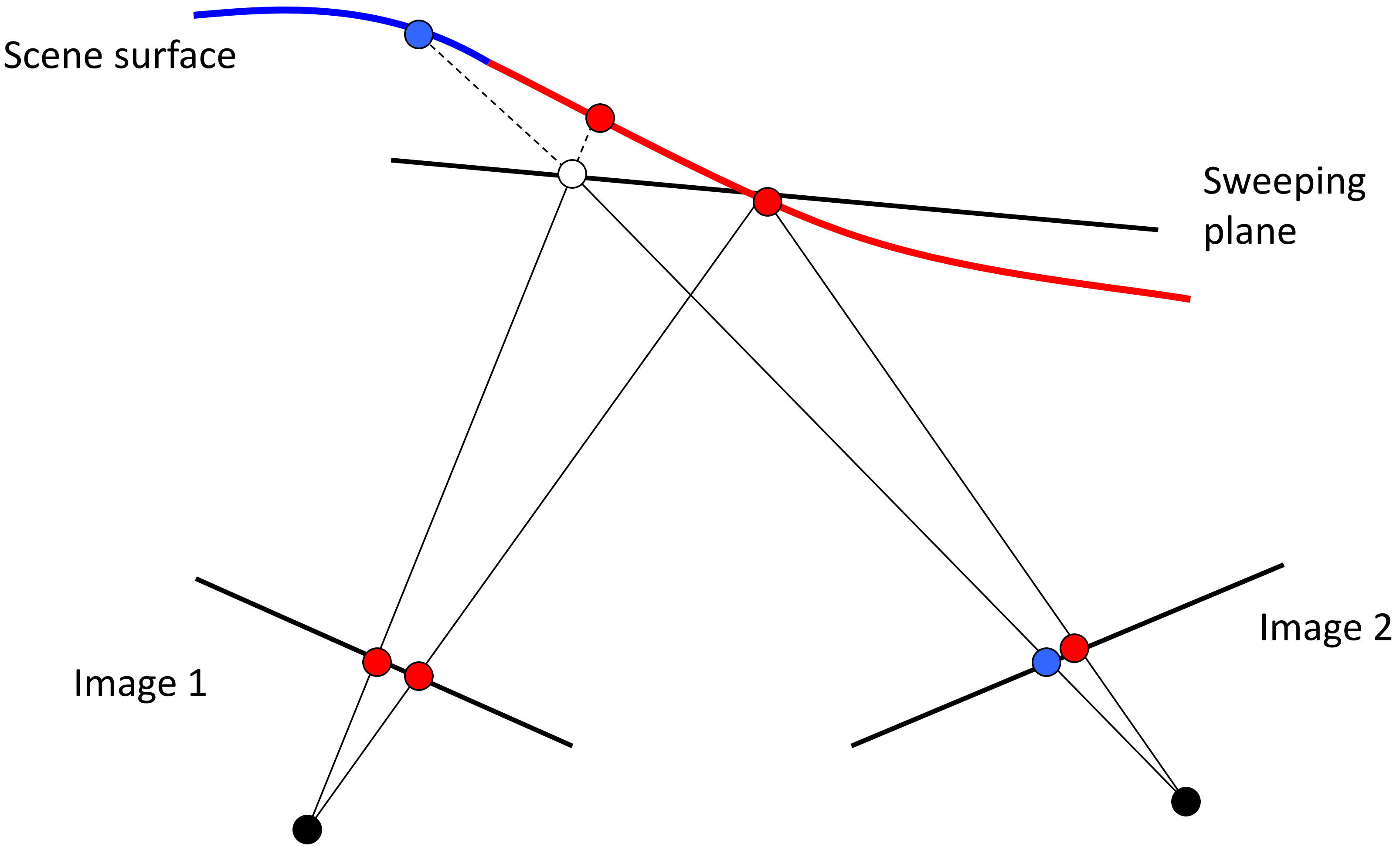
Plane Sweep Stereo



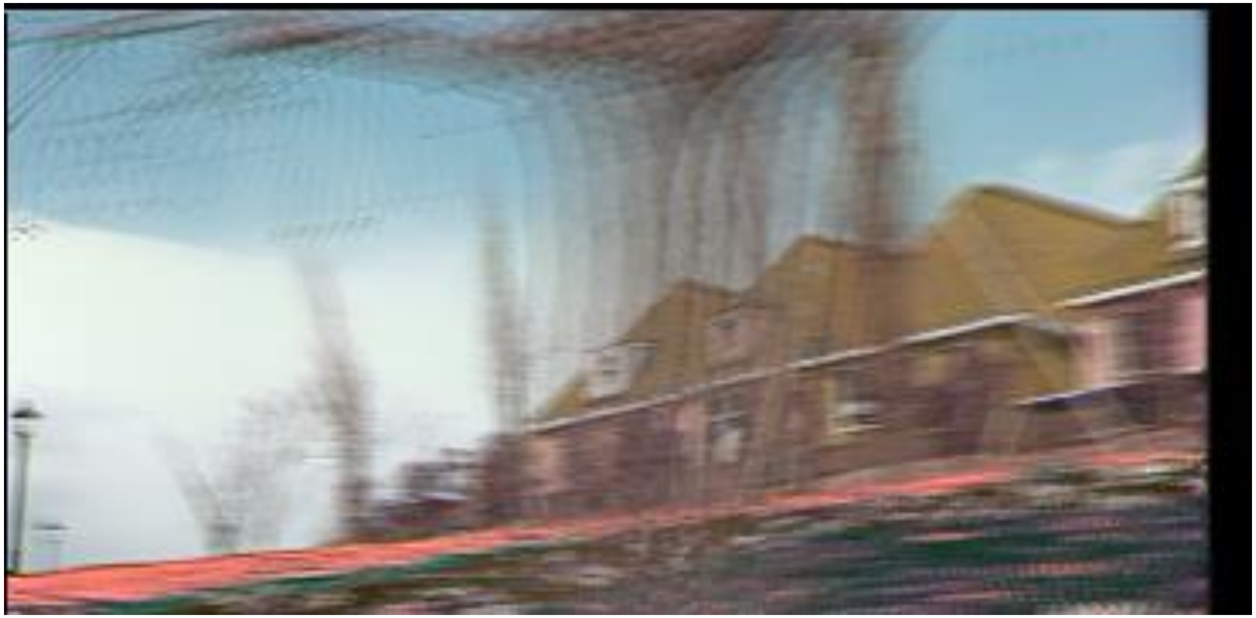
- Sweep family of planes at different depths w.r.t. a reference camera
  - For each depth, project each input image onto that plane
  - This is equivalent to a homography warping each input image into the reference view
- What can we say about the scene points that are at the right depth?

R. Collins. [A space-sweep approach to true multi-image matching.]() CVPR 1996.

Plane Sweep Stereo



Plane Sweep Stereo



- For each depth plane
  - For each pixel in the composite image stack, compute the variance
- For each pixel, select the depth that gives the lowest variance

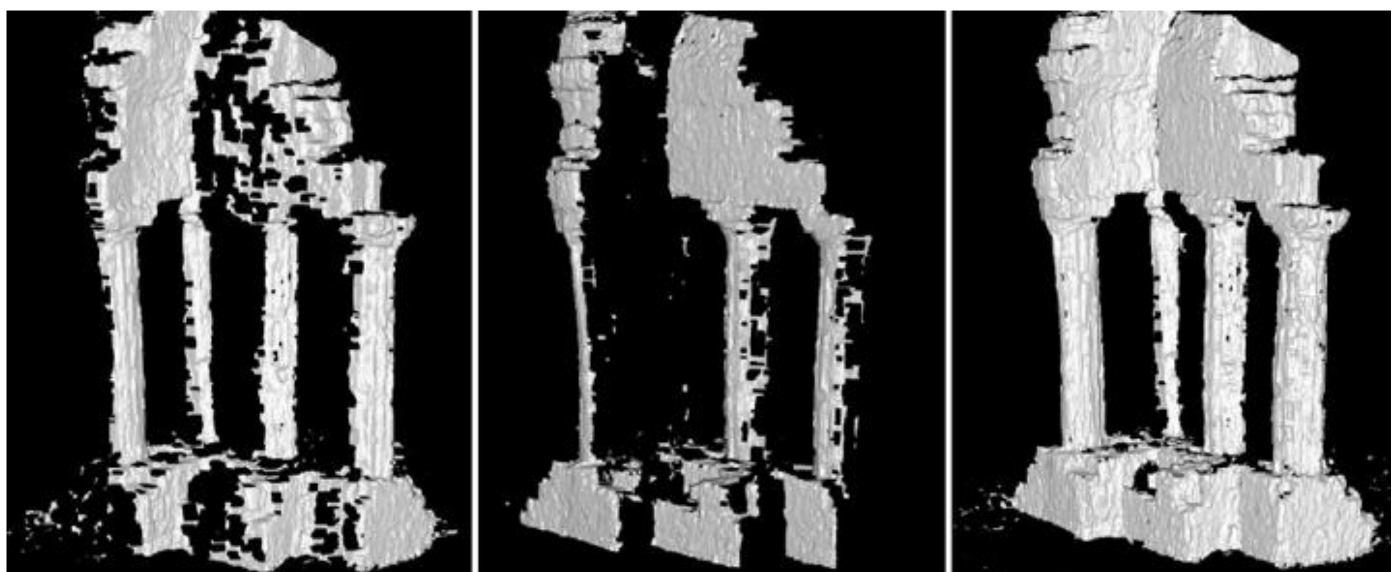
- Can be accelerated using graphics hardware

R. Yang and M. Pollefeys. *Multi-Resolution Real-Time Stereo on Commodity Graphics Hardware*, CVPR 2003

Merging depth maps

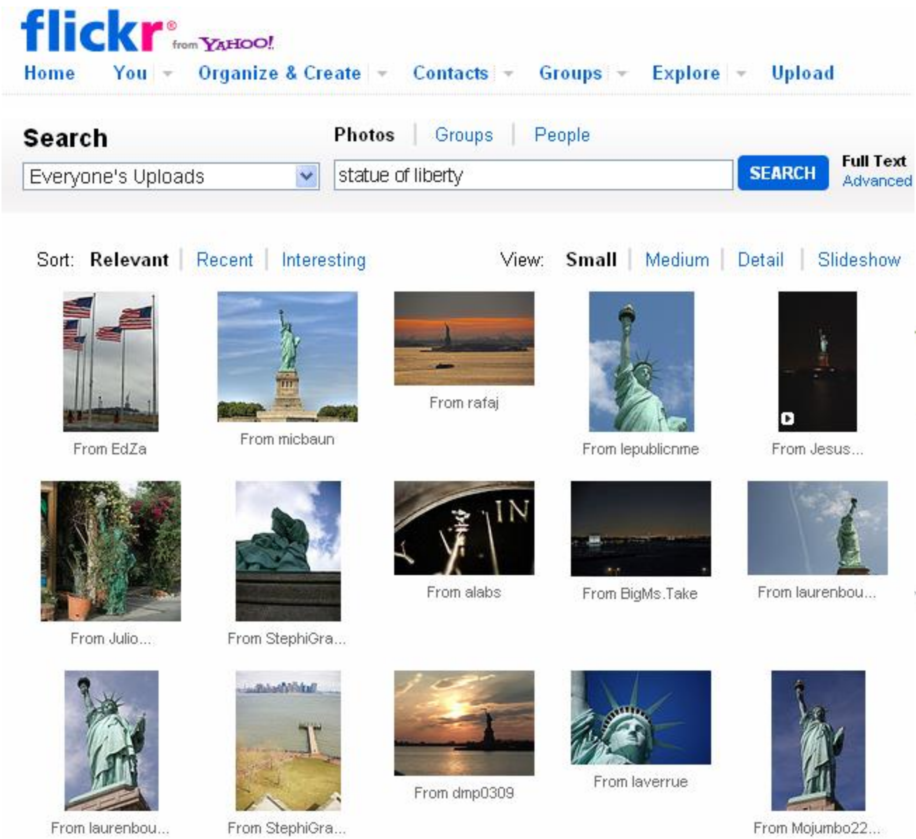
- Given a group of images, choose each one as reference and compute a depth map w.r.t. that view using a multi-baseline approach
- Merge multiple depth maps to a volume or a mesh (see, e.g., Curless and Levoy 96)

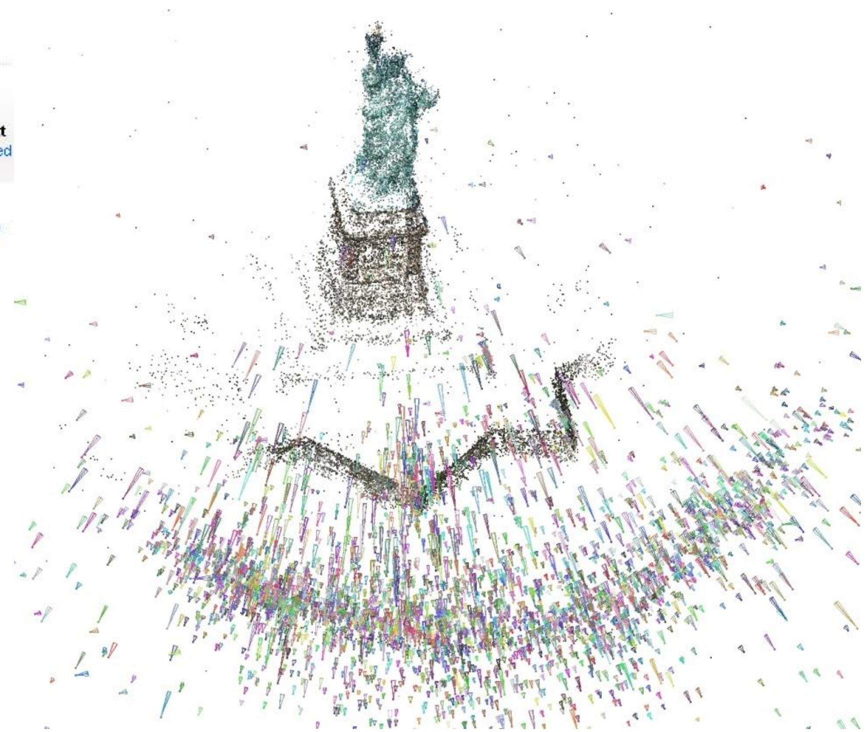
Map 1 Map 2 Merged

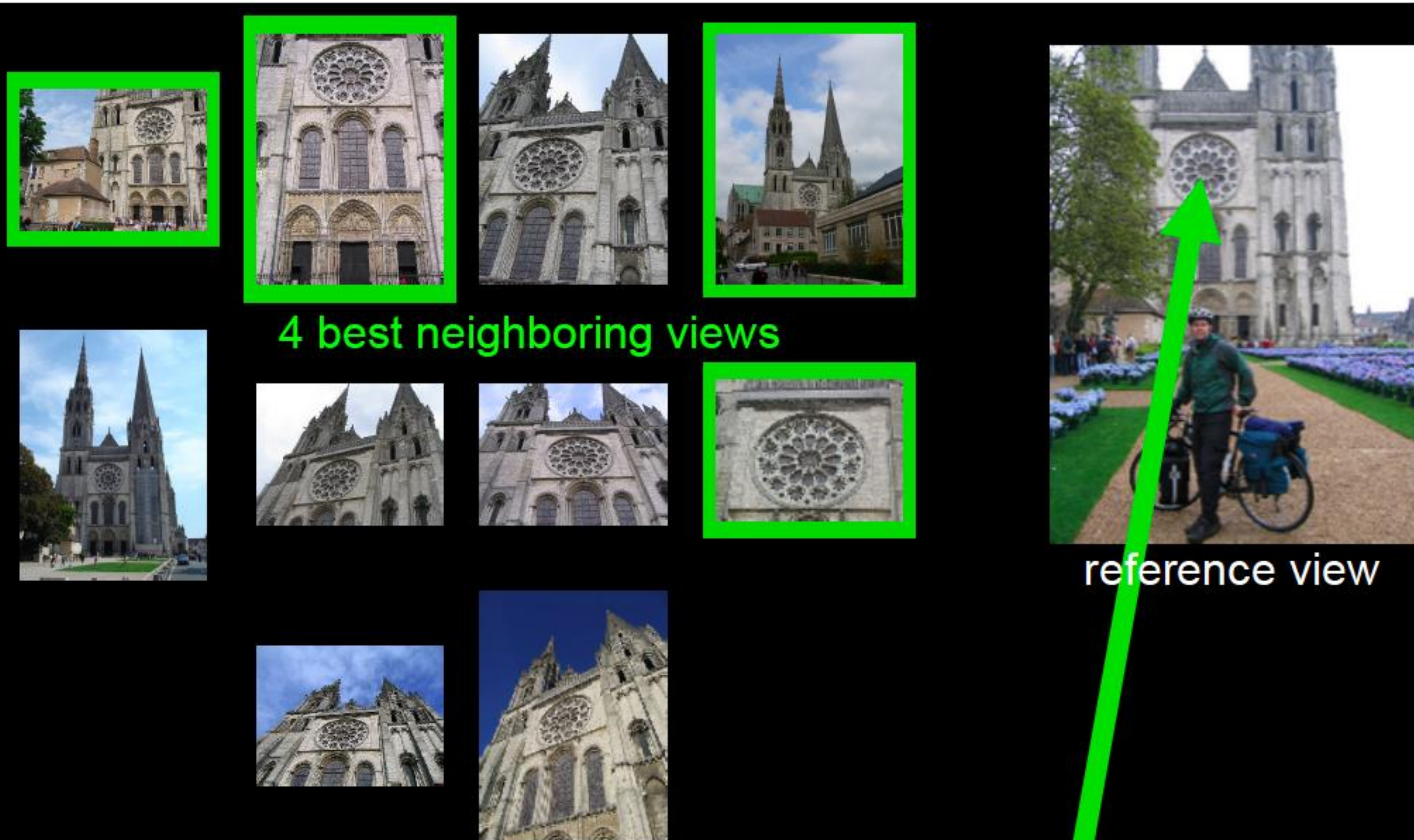


Stereo from community photo collections

- Need *structure from motion* to recover unknown camera parameters
- Need *view selection* to find good groups of images on which to run dense stereo

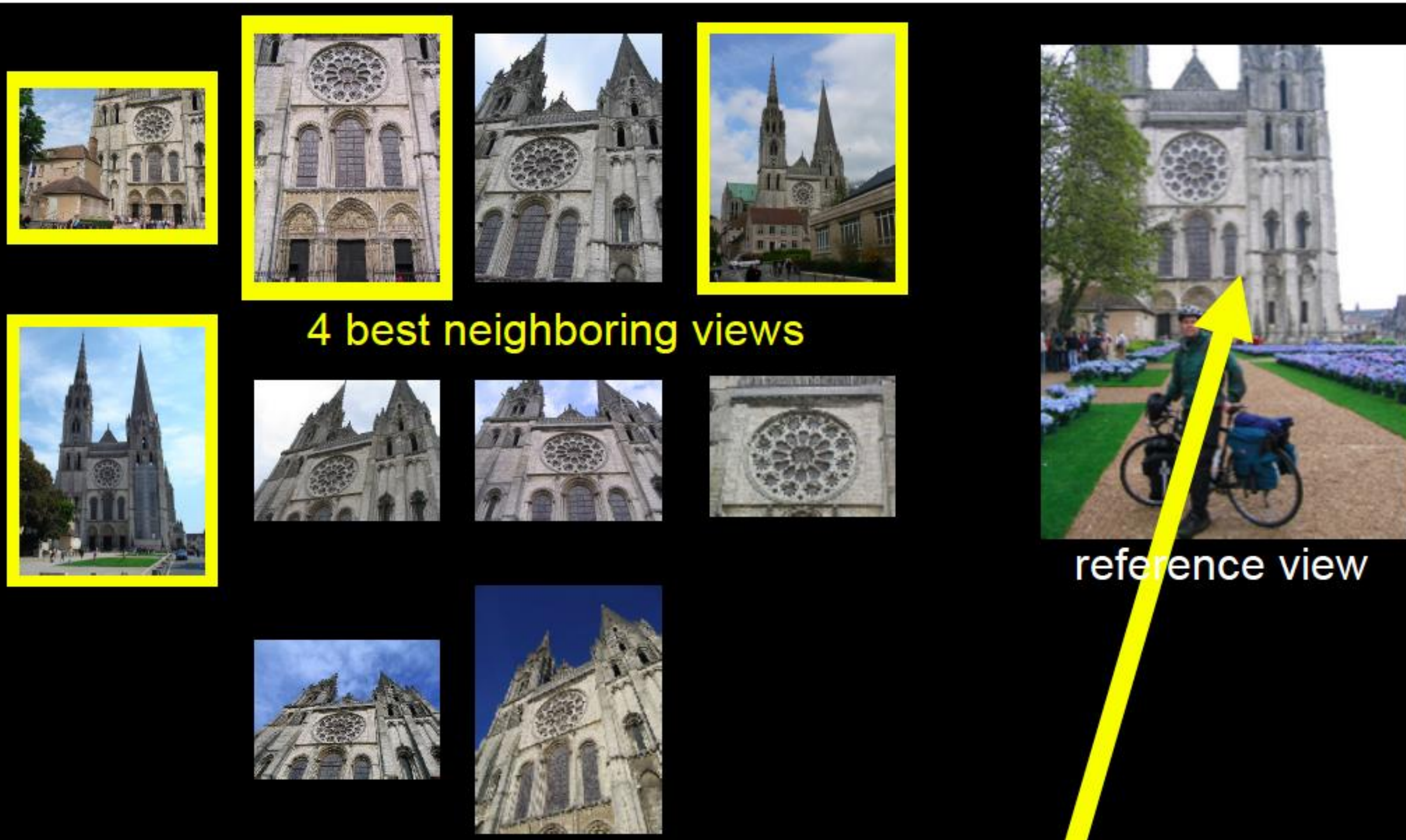






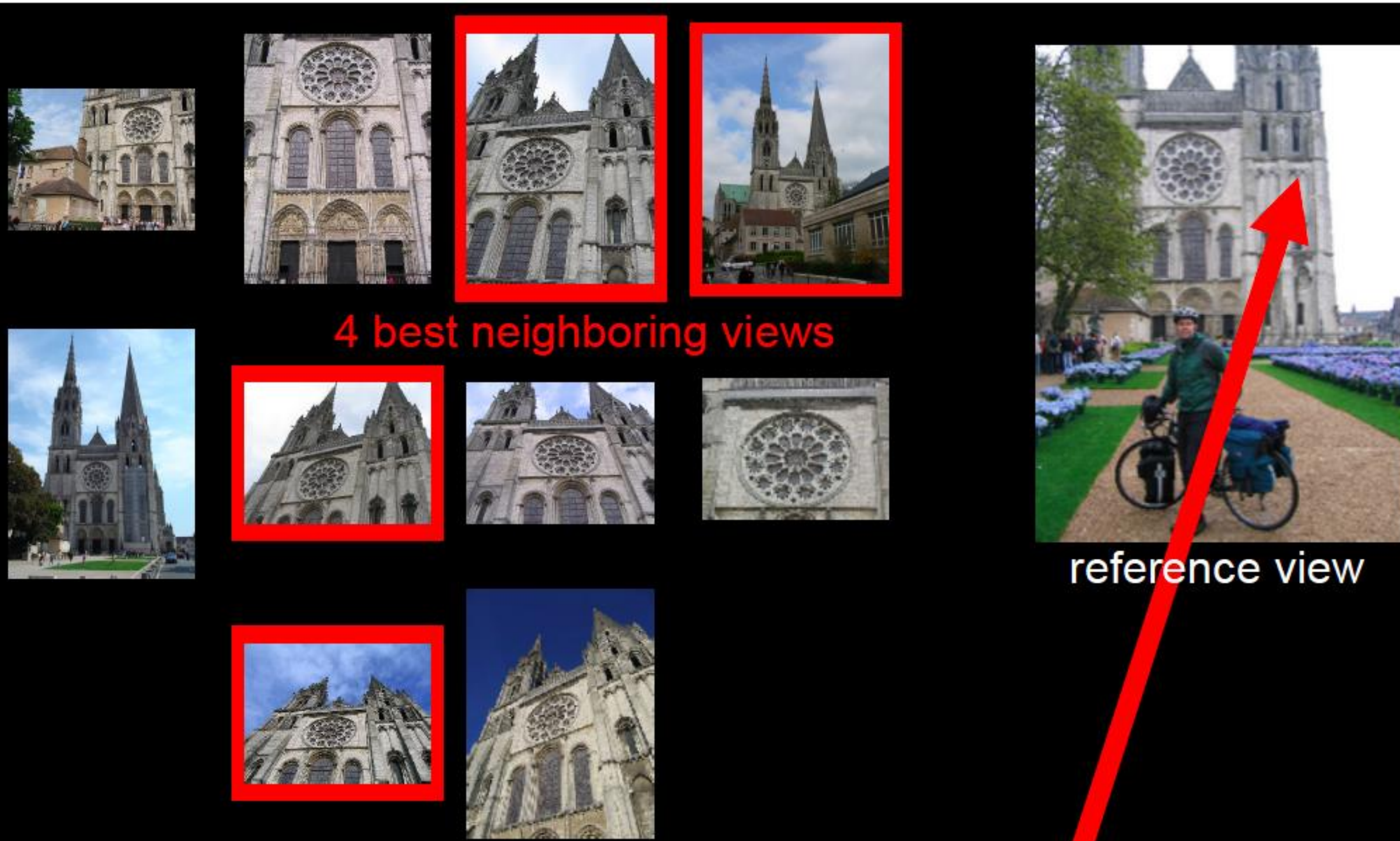
Local view selection

- Automatically select neighboring views for each point in the image
- Desiderata: good matches AND good baselines



Local view selection

- Automatically select neighboring views for each point in the image
- Desiderata: good matches AND good baselines



Local view selection

- Automatically select neighboring views for each point in the image
- Desiderata: good matches AND good baselines

Towards Internet-Scale Multi-View Stereo

- YouTube video, high-quality video

Yasutaka Furukawa, Brian Curless, Steven M. Seitz and Richard Szeliski, Towards Internet-scale Multi-view Stereo,CVPR 2010.

Internet-Scale Multi-View Stereo

The Visual Turing Test for Scene Reconstruction

Rendered Images (Right) vs. Ground Truth Images (Left)

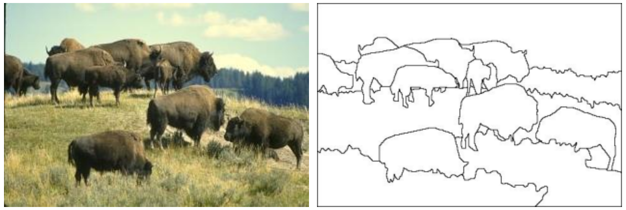
Q. Shan, R. Adams, B. Curless, Y. Furukawa, and S. Seitz, ["The Visual Turing Test for Scene Reconstruction,"]() 3DV 2013.

The Reading List

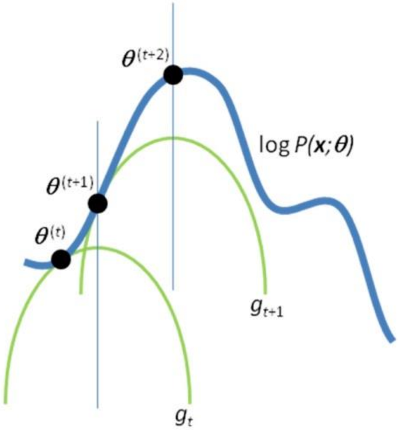
- "A computer algorithm for reconstructing a scene from two images", Longuet-Higgins, Nature 1981
- "Shape and motion from image streams under orthography: A factorization method." C. Tomasi and T. Kanade, *IJCV*, 9(2):137-154, November 1992
- "In defense of the eight-point algorithm", Hartley, PAMI 1997
- "An efficient solution to the five-point relative pose problem", Nister, PAMI 2004
- "Accurate, dense, and robust multiview stereopsis", Furukawa and Ponce, CVPR 2007
- "Photo tourism: exploring image collections in 3d", ACM SIGGRAPH 2006
- "Building Rome in a day", Agarwal et al., ICCV 2009
- https://www.youtube.com/watch?v=kyIzMr917Rc, 3D Computer Vision: Past, Present, and Future

Grouping and Segmentation

- **Image Segmentation**
  - Which pixels belong together?

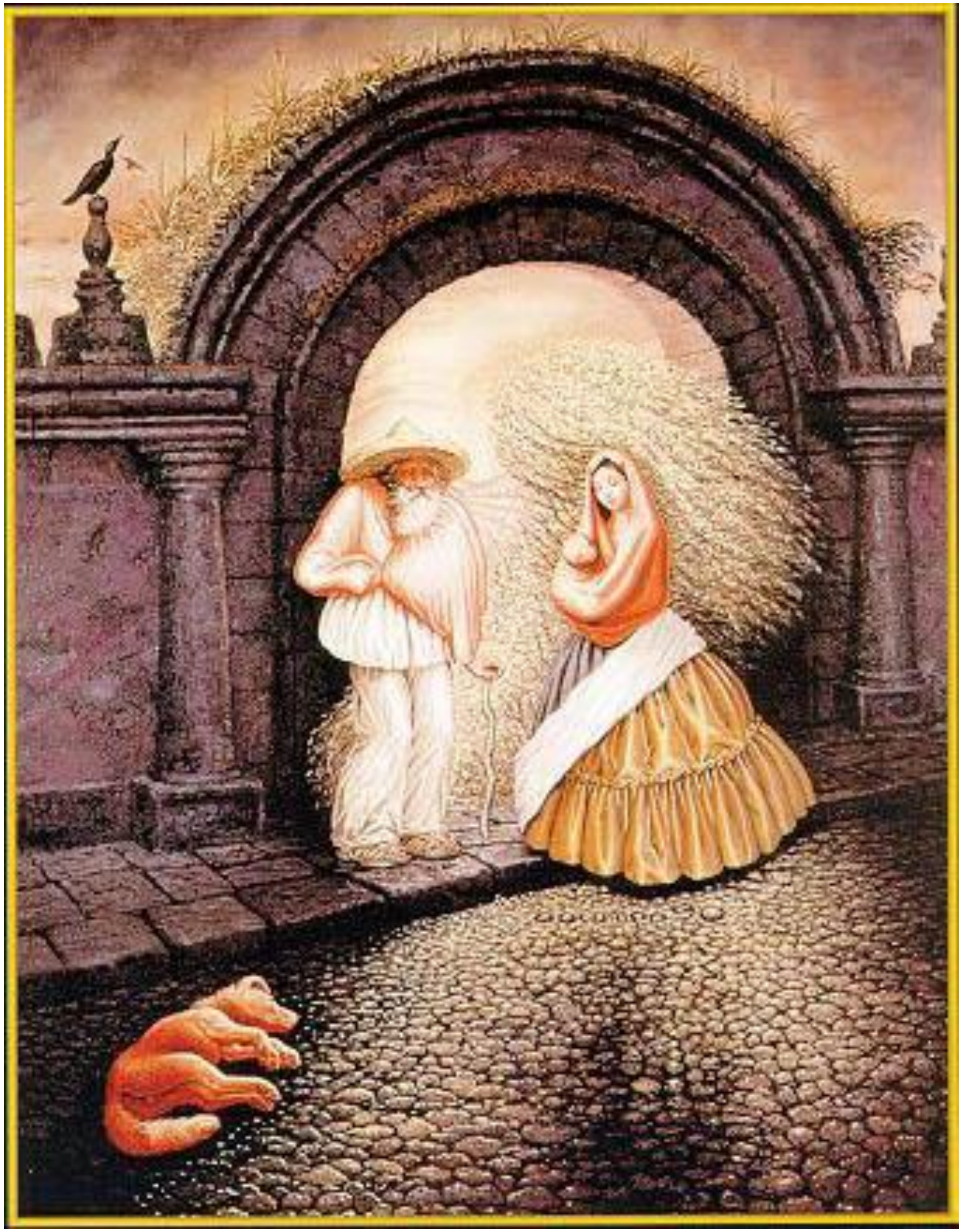


- **Hidden Variables, the EM Algorithm, and Mixtures of Gaussians**
  - How to handle missing data?



- **MRFs and Segmentation with Graph Cut**
  - How do we solve image labeling problems?

How many people?



Gestalt psychology or gestaltism

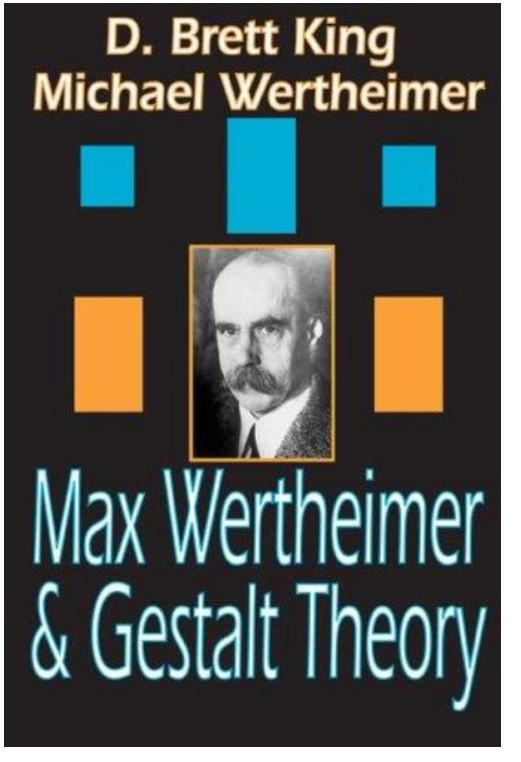
German: *Gestalt* - "form" or "whole”

Berlin School, early 20th century

Kurt Koffka, Max Wertheimer, and Wolfgang Köhler

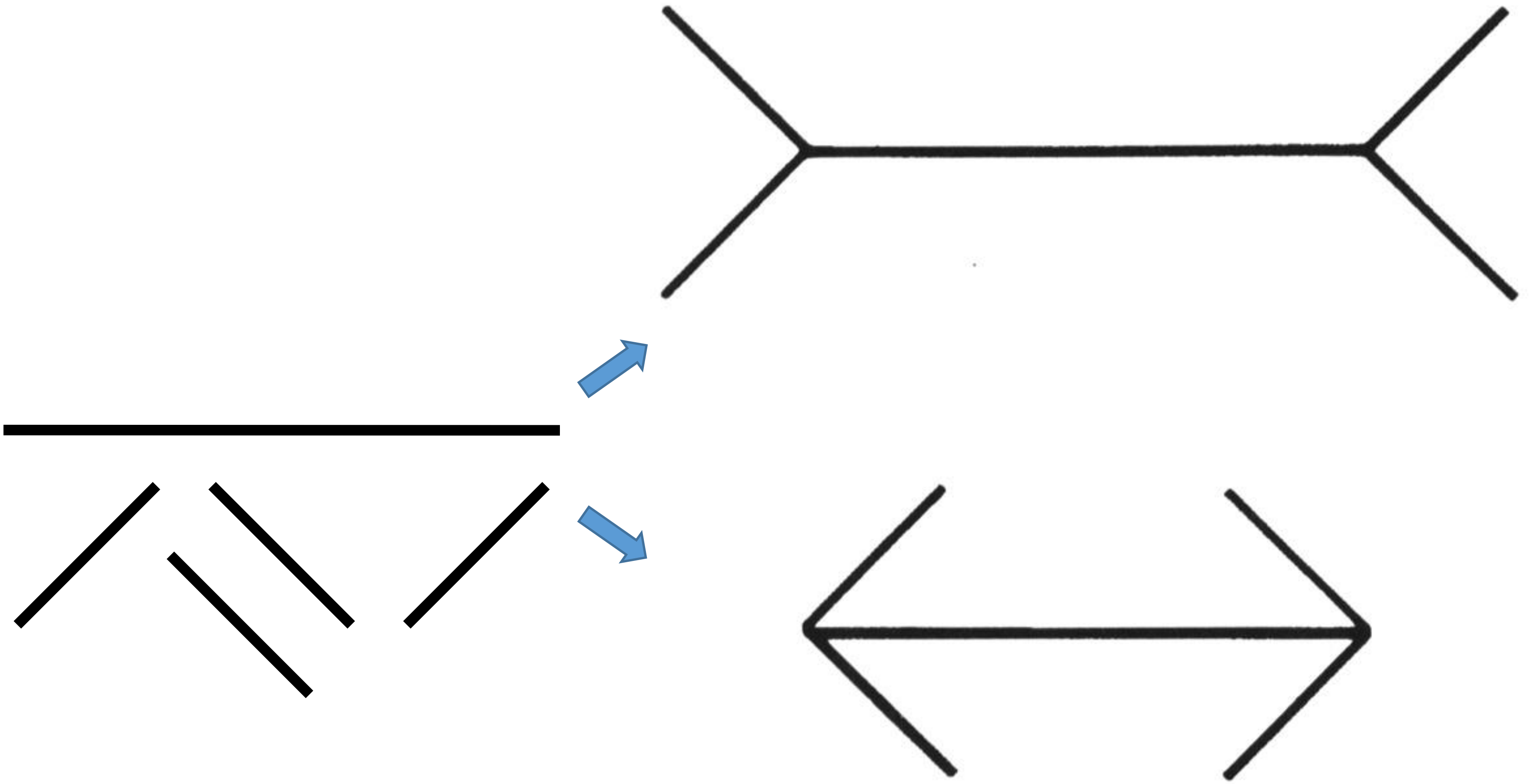
View of brain:

- whole is more than the sum of its parts
- holistic
- parallel
- analog
- self-organizing tendencies



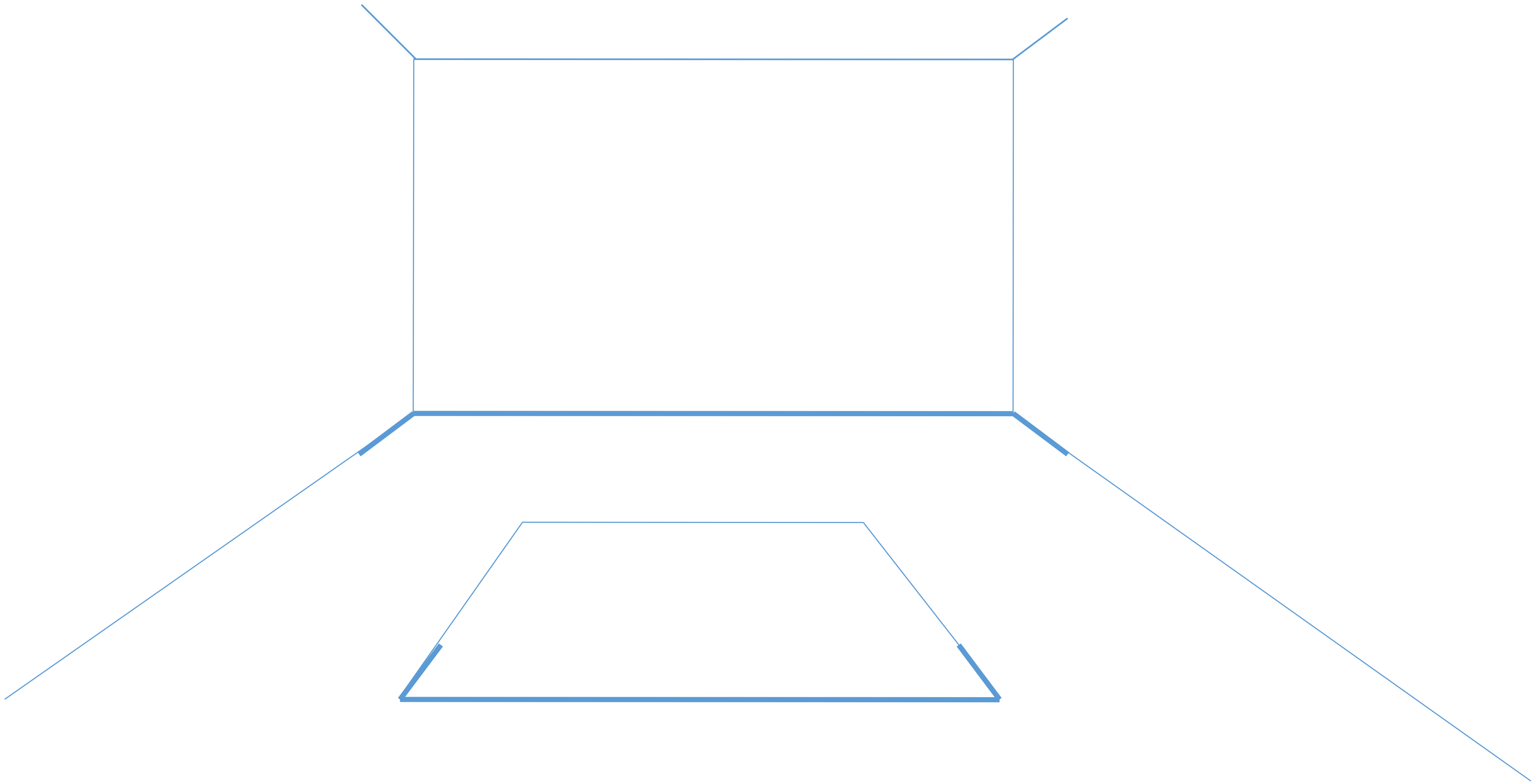
Slide from S. Saverese

Gestaltism

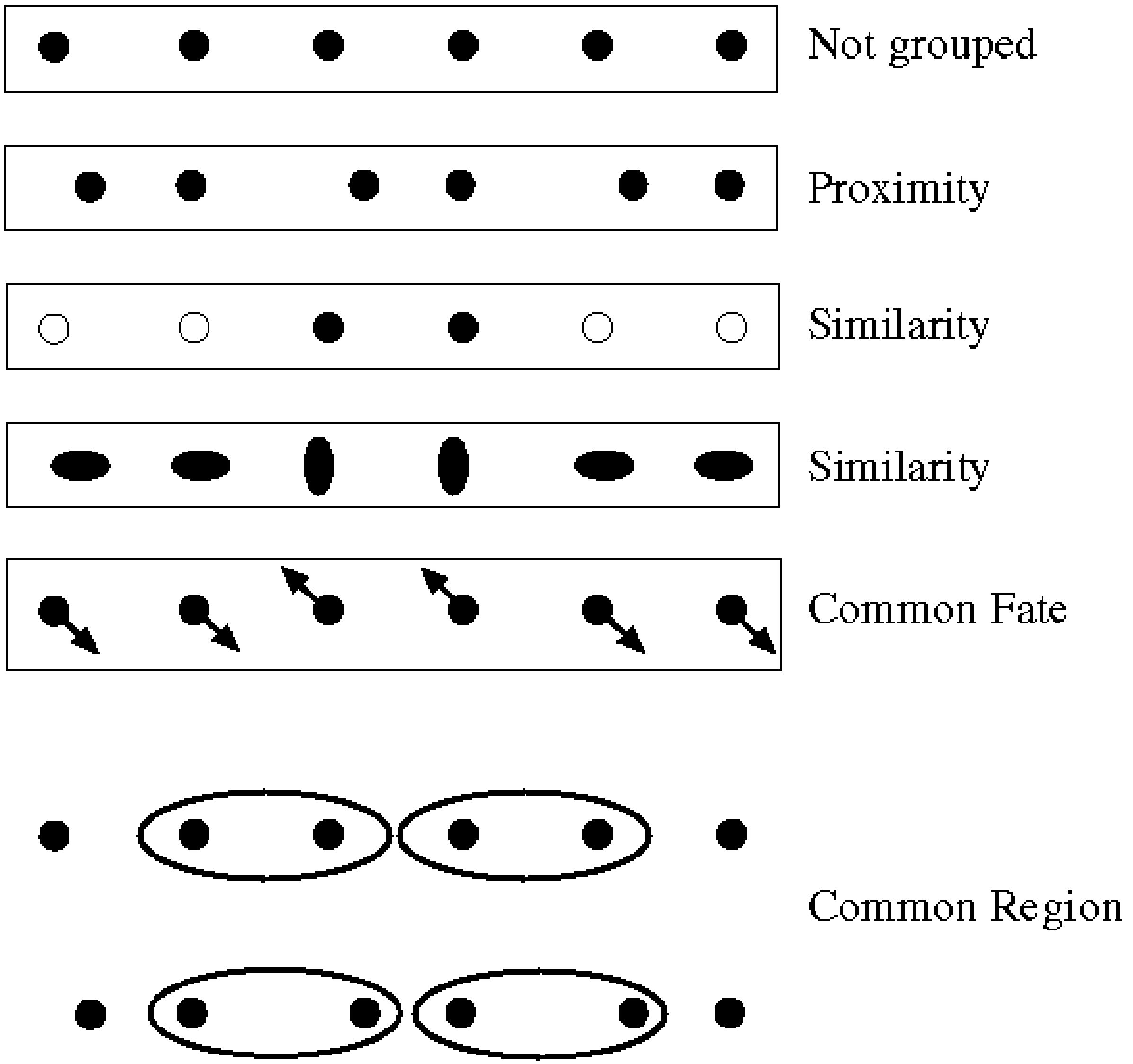


The Muller-Lyer illusion

We perceive the interpretation, not the senses

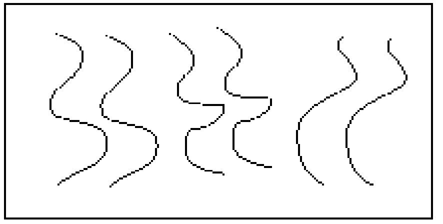


Principles of perceptual organization

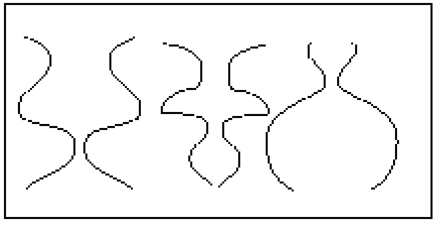


From Steve Lehar: The Constructive Aspect of Visual Perception

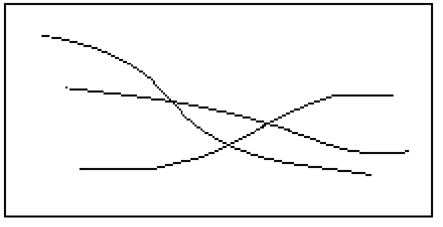
Principles of perceptual organization



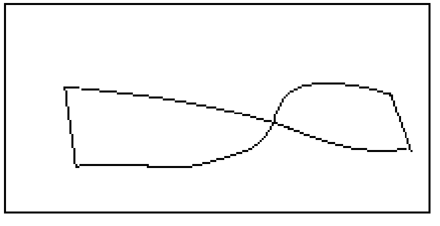
Parallelism



Symmetry

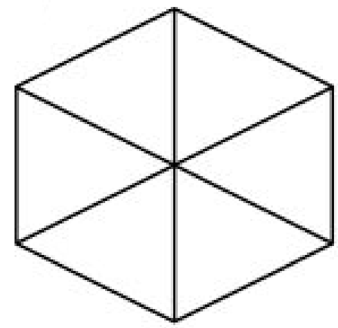


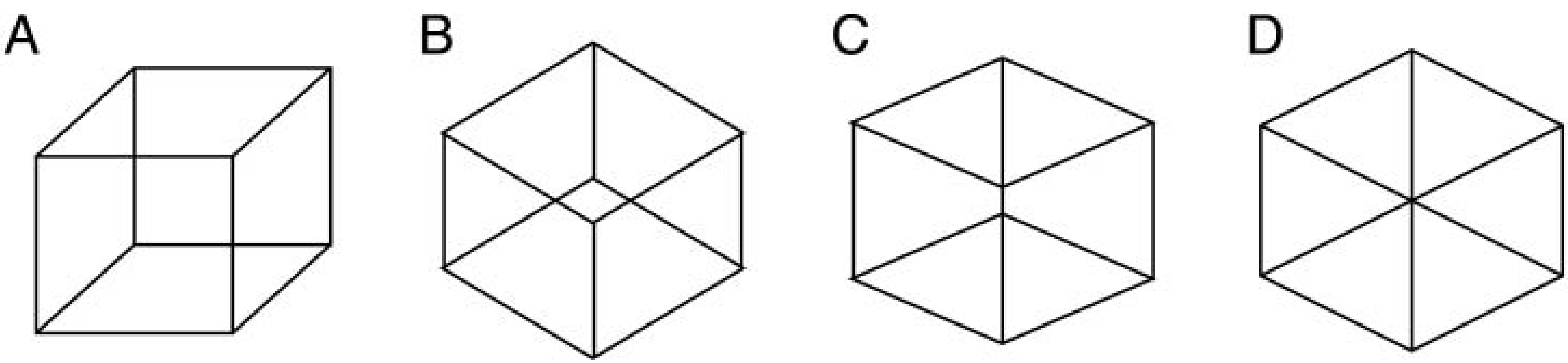
Continuity



Closure

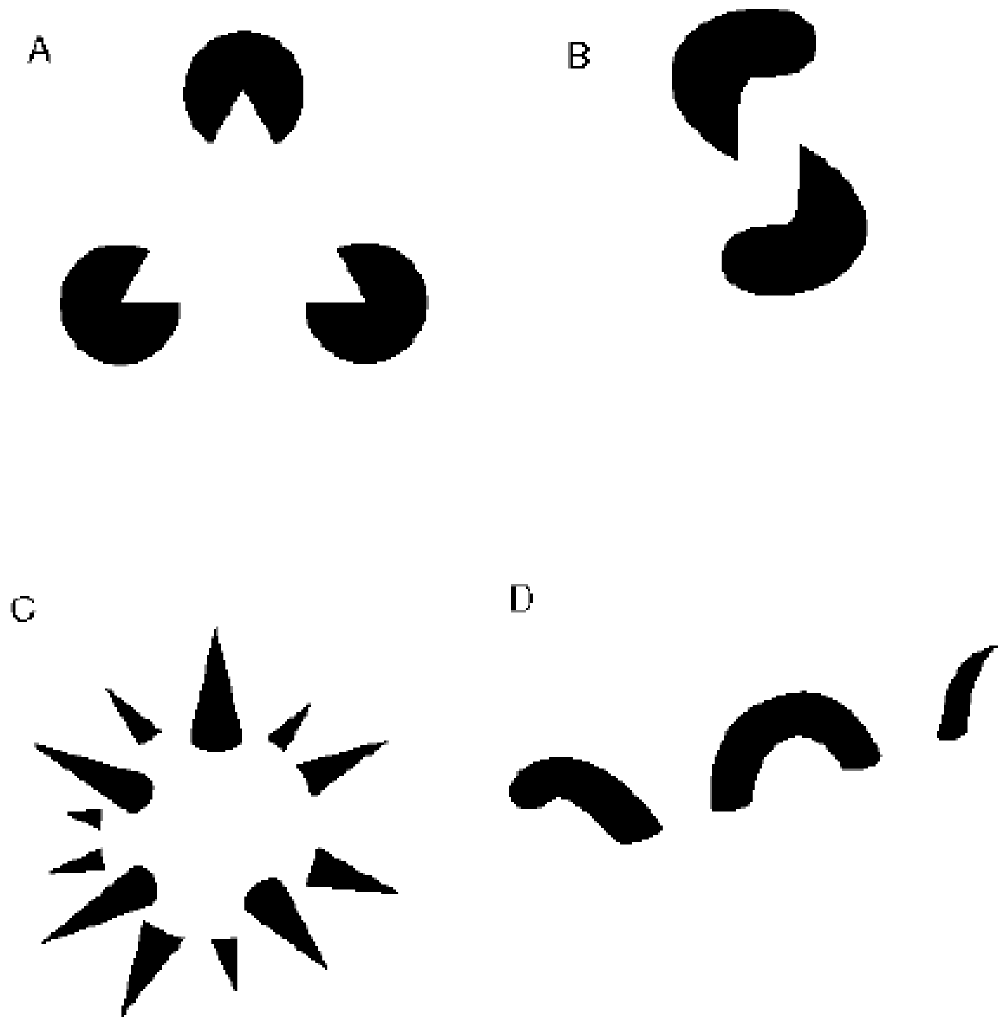
Gestaltists do not believe in coincidence





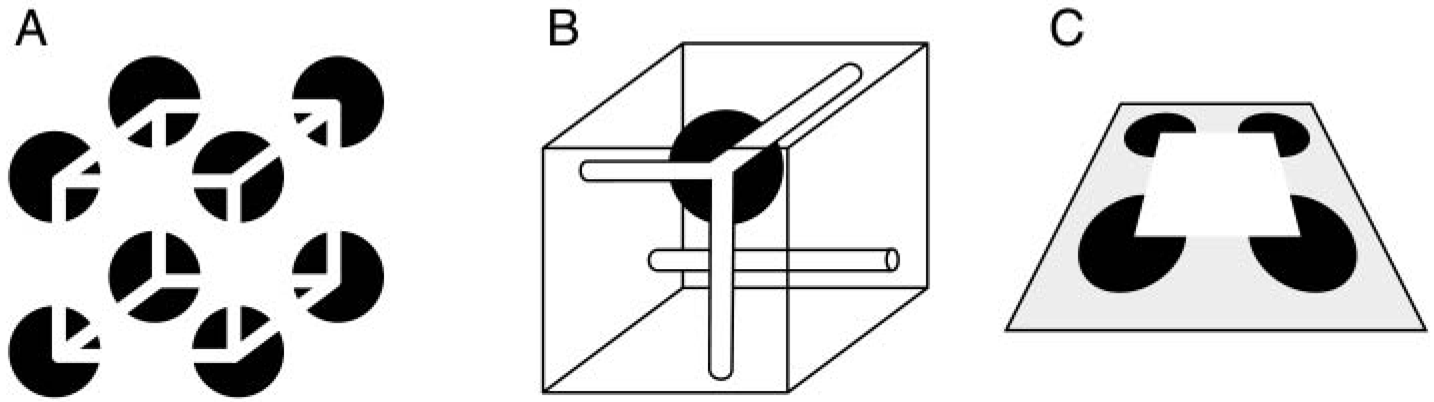
Emergence

Grouping by invisible completion



From Steve Lehar: The Constructive Aspect of Visual Perception

Grouping involves global interpretation



From Steve Lehar: The Constructive Aspect of Visual Perception

Grouping involves global interpretation

A B

From Steve Lehar: The Constructive Aspect of Visual Perception

Gestalt cues

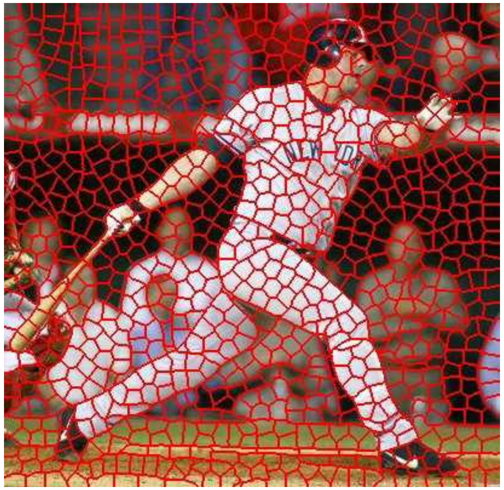
- Good intuition and basic principles for grouping
- Basis for many ideas in segmentation and occlusion reasoning
- Some (e.g., symmetry) are difficult to implement in practice

Image segmentation

Goal: Group pixels into meaningful or perceptually similar regions

Segmentation for efficiency: "superpixels"

[Felzenszwalb and Huttenlocher 2004]

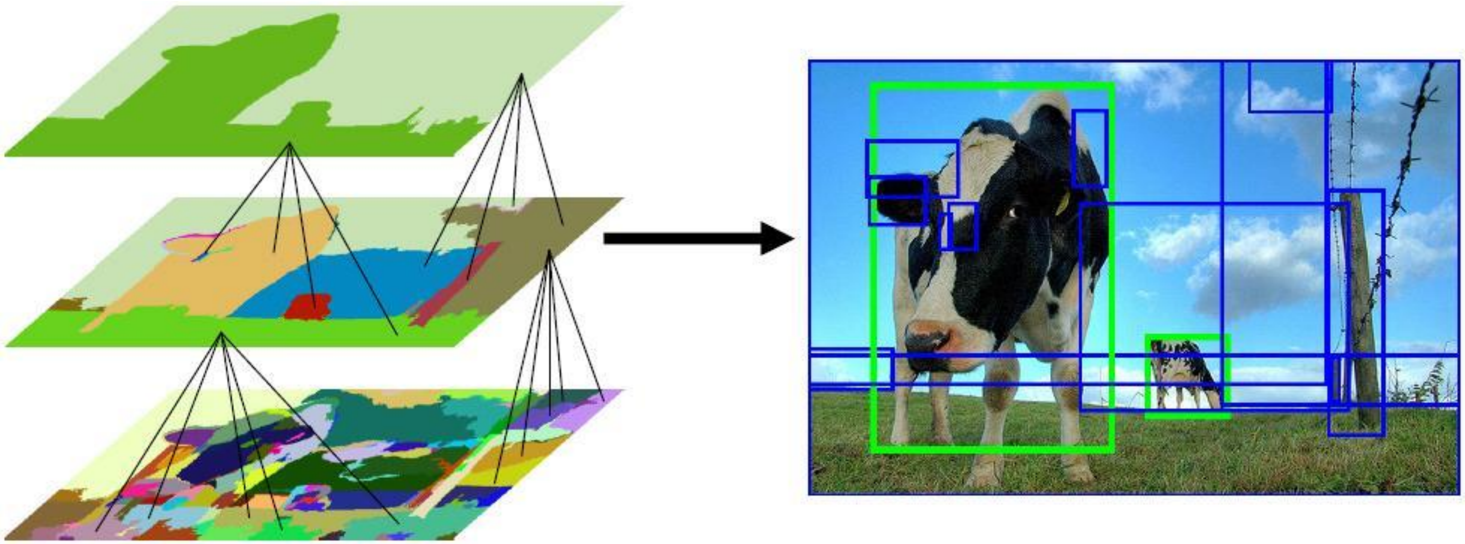


[Shi and Malik 2001]

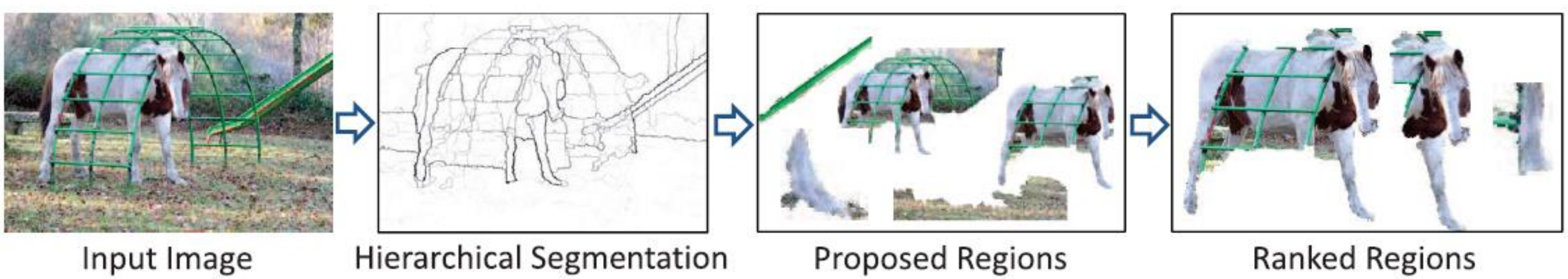
[Hoiem et al. 2005, Mori 2005]

Segmentation for feature support

Segmentation for object proposals



"Selective Search" [Sande, Uijlings et al. ICCV 2011, IJCV 2013]



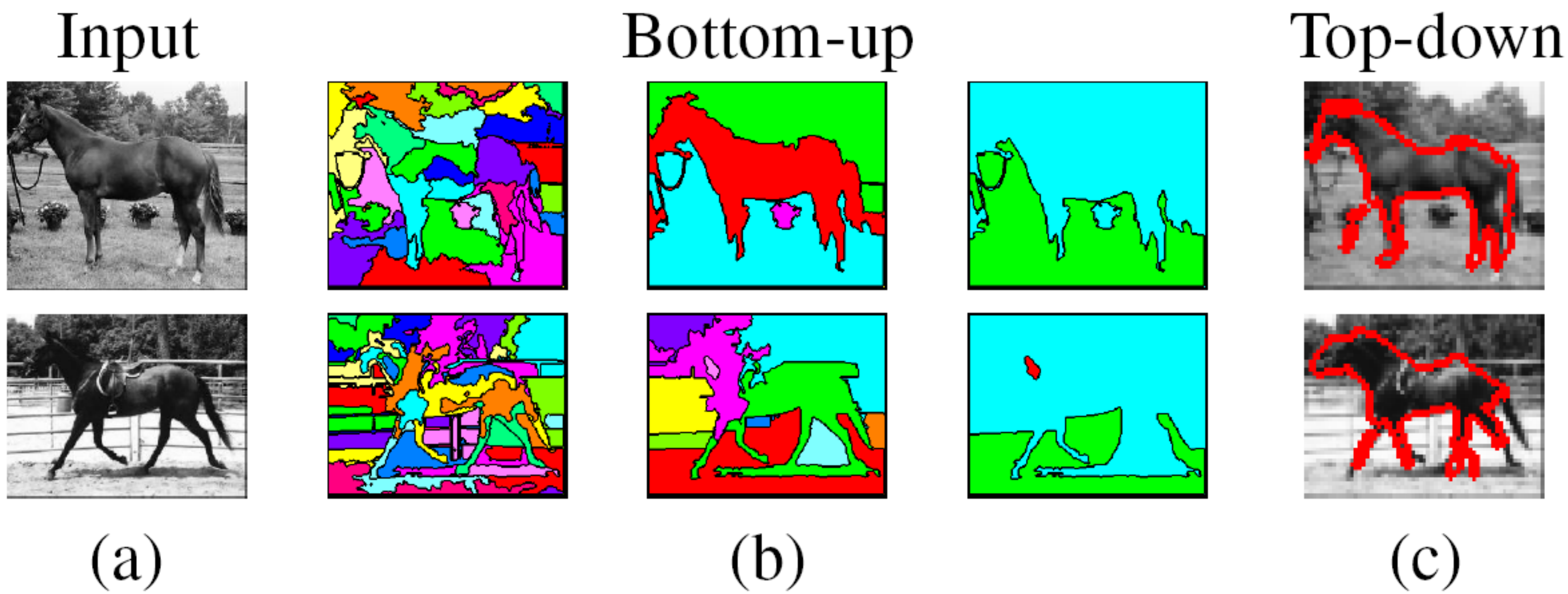
[Endres Hoiem ECCV 2010, IJCV 2014]

Segmentation as a result

Rother et al. 2004

Major processes for segmentation

- Bottom-up: group tokens with similar features
- Top-down: group tokens that likely belong to the same object

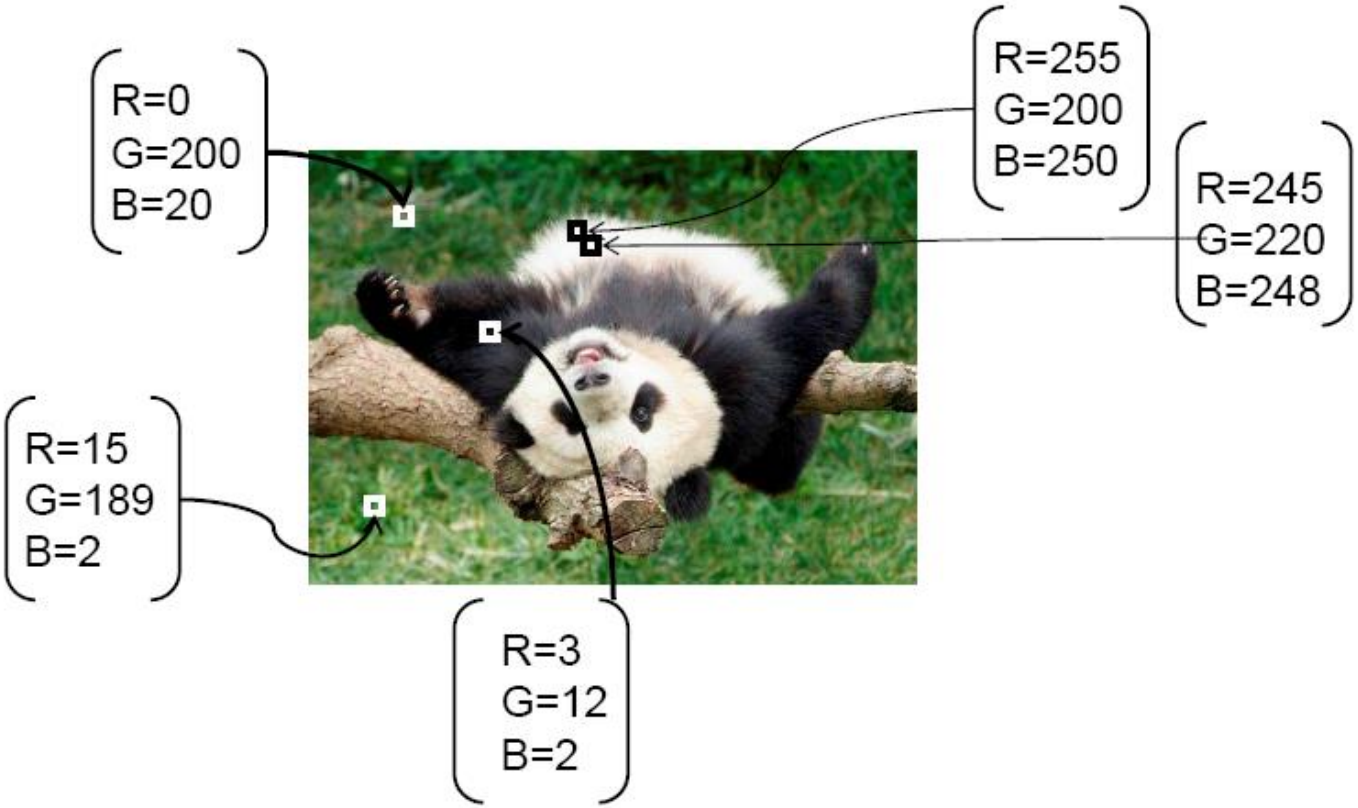


[Levin and Weiss 2006]

Segmentation using clustering

- Kmeans
- Mean-shift

Feature Space



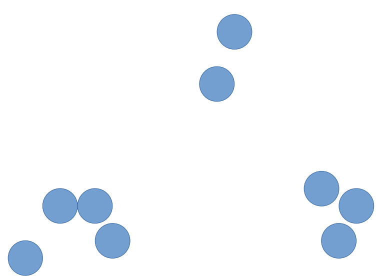
Source: K. Grauman

K-means algorithm

$$\underset{S,\mu_{i,i=1..K}}{\operatorname{argmin}} \sum_{i=1}^{K} \sum_{x \in S_i} ||x - \mu_i||^2$$

Partition the data into K sets $S = \{S_1, S_2, \ldots S_K\}$ with corresponding centers μ_i

Partition such that variance in each partition is as low as possible

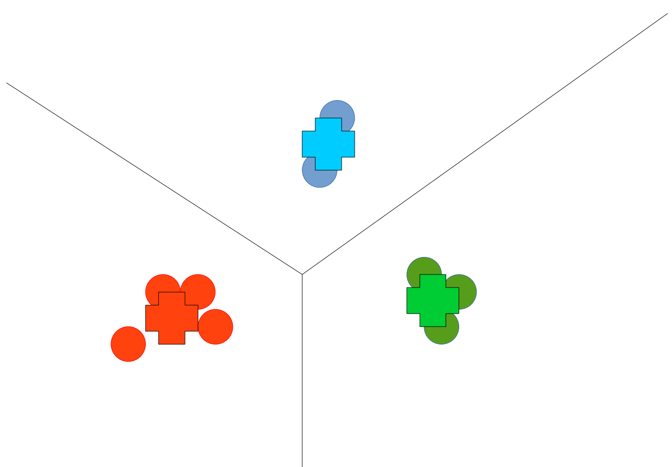


K-means algorithm

$$\underset{S,\mu_{i,i=1..K}}{\operatorname{argmin}} \sum_{i=1}^{K} \sum_{x \in S_i} ||x - \mu_i||^2$$

Partition the data into K sets $S = \{S_1, S_2, \ldots S_K\}$ with corresponding centers μ_i

Partition such that variance in each partition is as low as possible



K-means algorithm

1. Initialize K centers μ_i (usually randomly)
2. Assign each point x to its nearest center:

$$S^t = \underset{S}{\operatorname{argmin}} \sum_{i=1}^{K} \sum_{x \in S_i} ||x - \mu_i||^2$$

3. Update cluster centers as the mean of its members

$$\mu^t = \underset{\mu_{i,i=1..K}}{\operatorname{argmin}} \sum_{i=1}^{K} \sum_{x \in S_i} ||x - \mu_i||^2$$

4. Repeat 2-3 until convergence (t = t+1)

```matlab
function C = kmeans(X, K)

% Initialize cluster centers to be randomly sampled points
[N, d] = size(X);
rp = randperm(N);
C = X(rp(1:K), :);

lastAssignment = zeros(N, 1);
while true
  % Assign each point to nearest cluster center
  bestAssignment = zeros(N, 1);
  mindist = Inf*ones(N, 1);
  for k = 1:K
    for n = 1:N
      dist = sum((X(n, :)-C(k, :)).^2);
      if dist < mindist(n)
        mindist(n) = dist;
        bestAssignment(n) = k;
      end
    end
  end

  % break if assignment is unchanged
  if all(bestAssignment==lastAssignment), break; end;

  % Assign each cluster center to mean of points within it
  for k = 1:K
    C(k, :) = mean(X(bestAssignment==k, :));
  end
end
```

K-means clustering using intensity alone and color alone

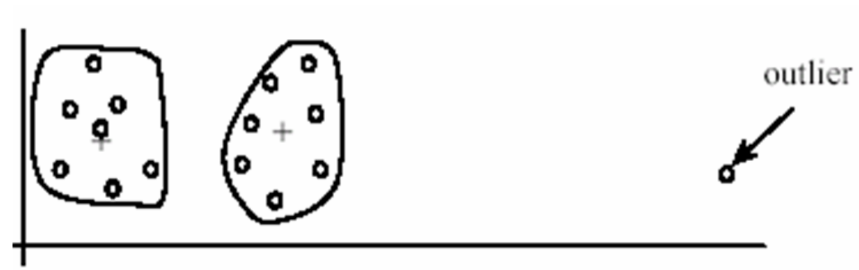
Image

Clusters on intensity

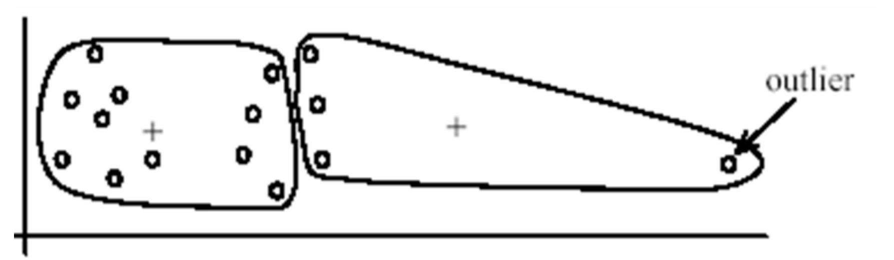
Clusters on color

K-Means pros and cons

- Pros
  - –Simple and fast
  - –Easy to implement
- Cons
  - –Need to choose K
  - –Sensitive to outliers
- Usage
  - –Rarely used for pixel segmentation



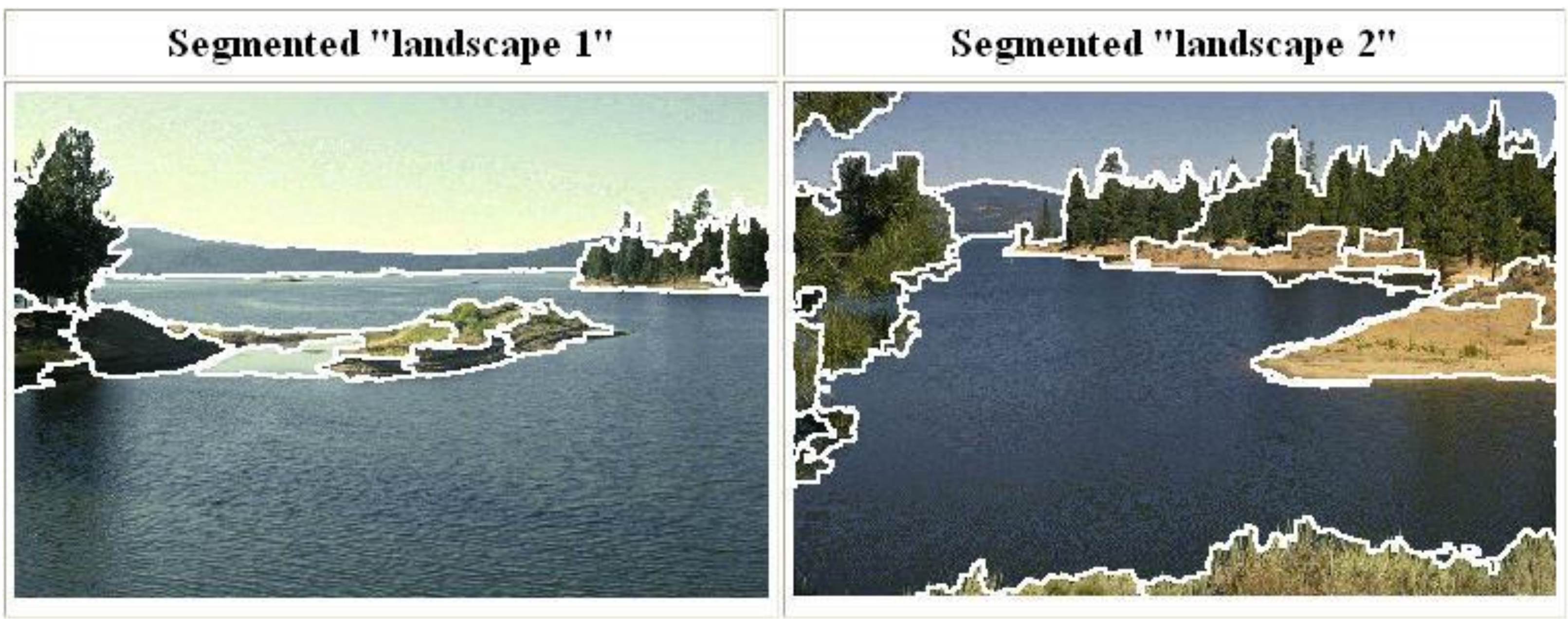
(B): Ideal clusters



Mean shift segmentation

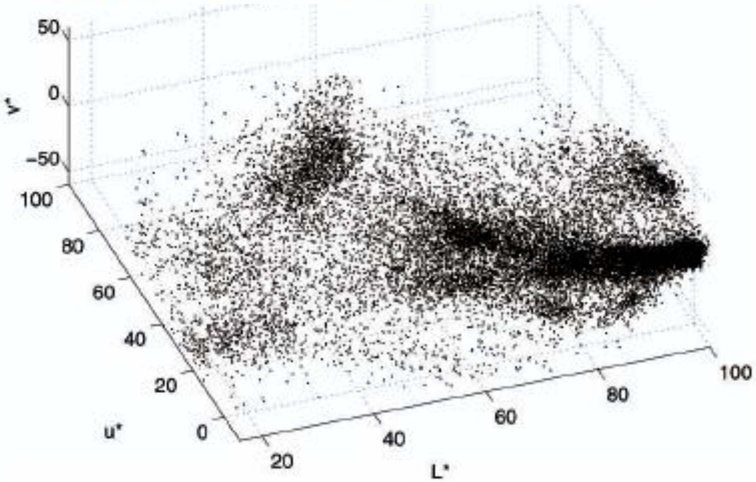
D. Comaniciu and P. Meer, Mean Shift: A Robust Approach toward Feature Space Analysis, PAMI 2002.

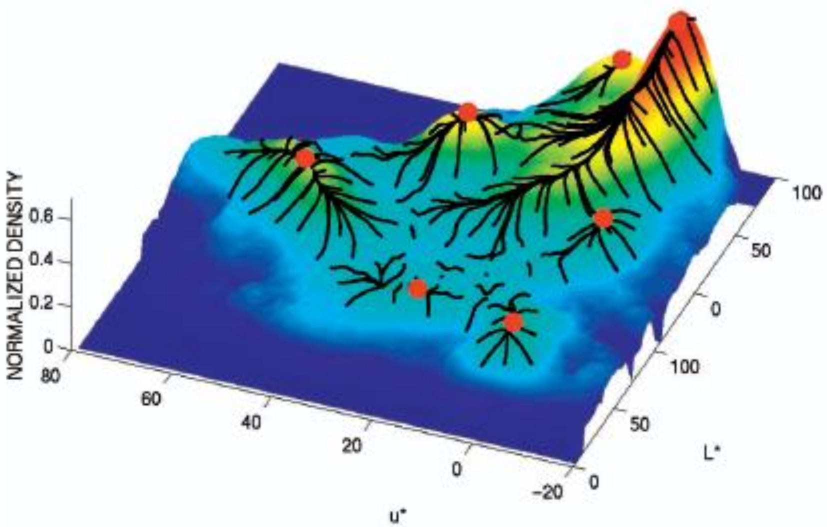
- Versatile technique for clustering-based segmentation

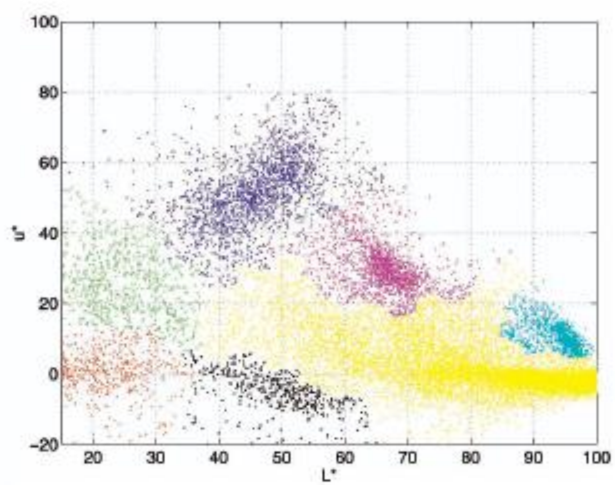


Mean shift algorithm

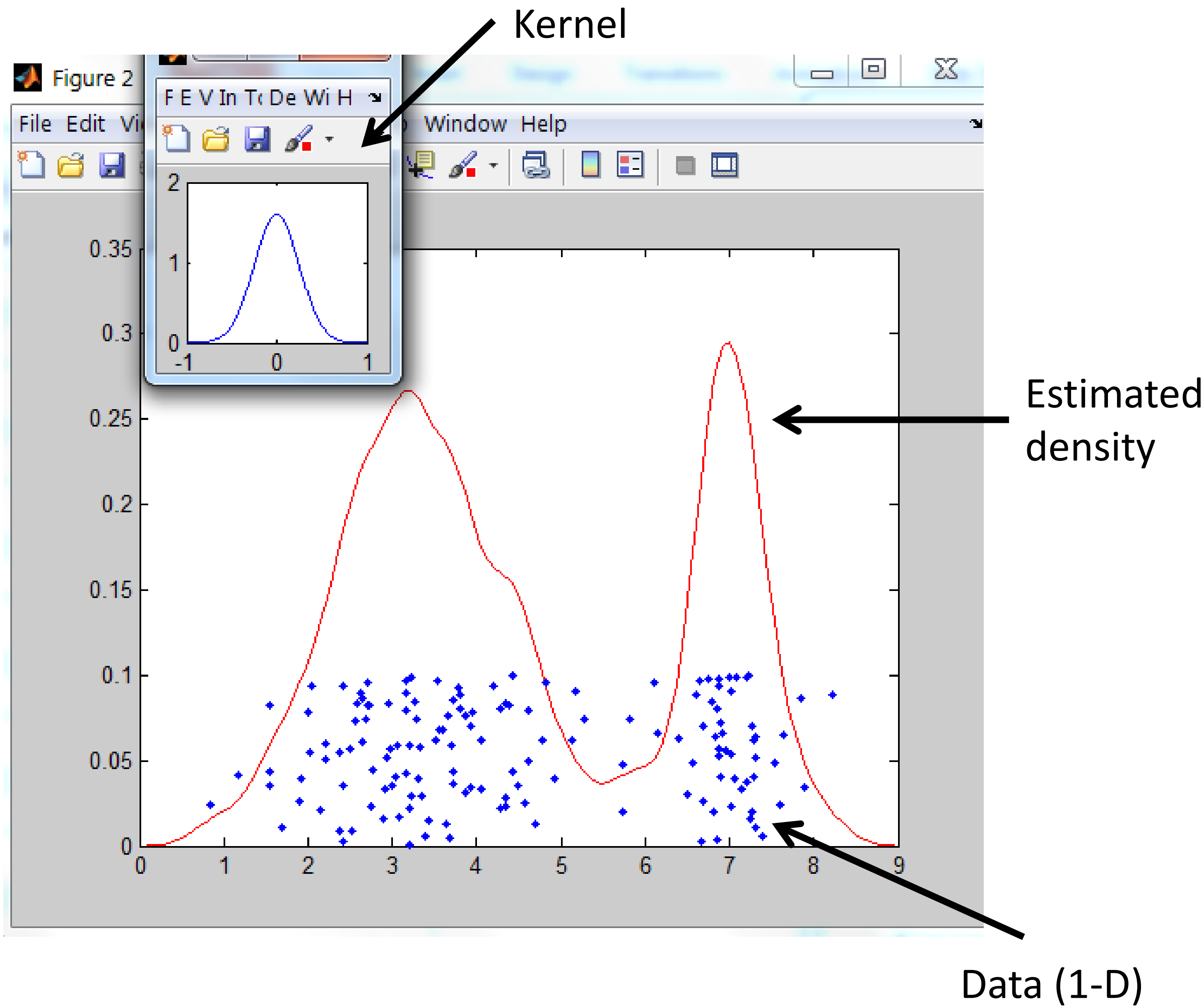
- Try to find *modes* of this non-parametric density







Kernel density estimation



Kernel density estimation

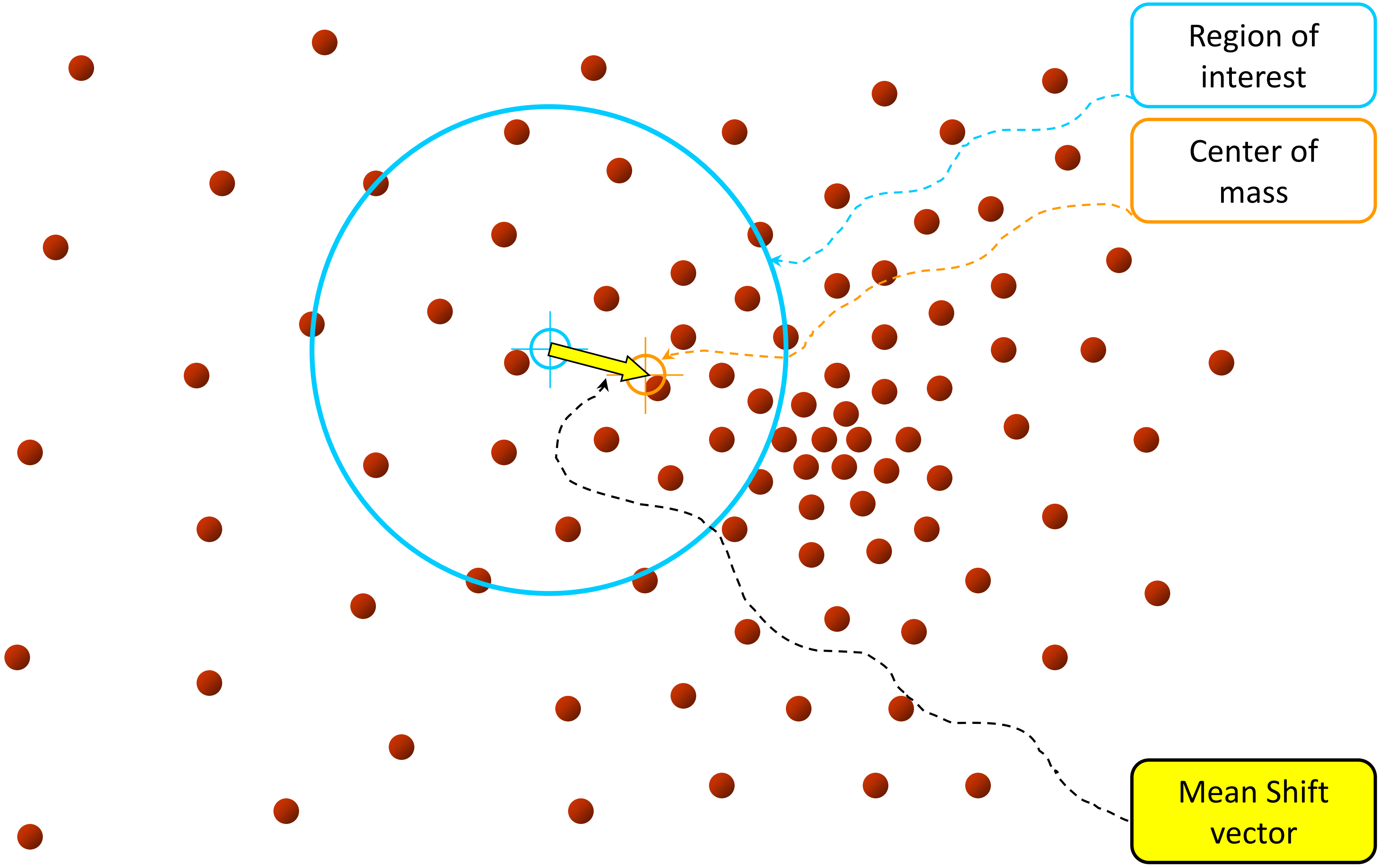
Kernel density estimation function

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

Gaussian kernel

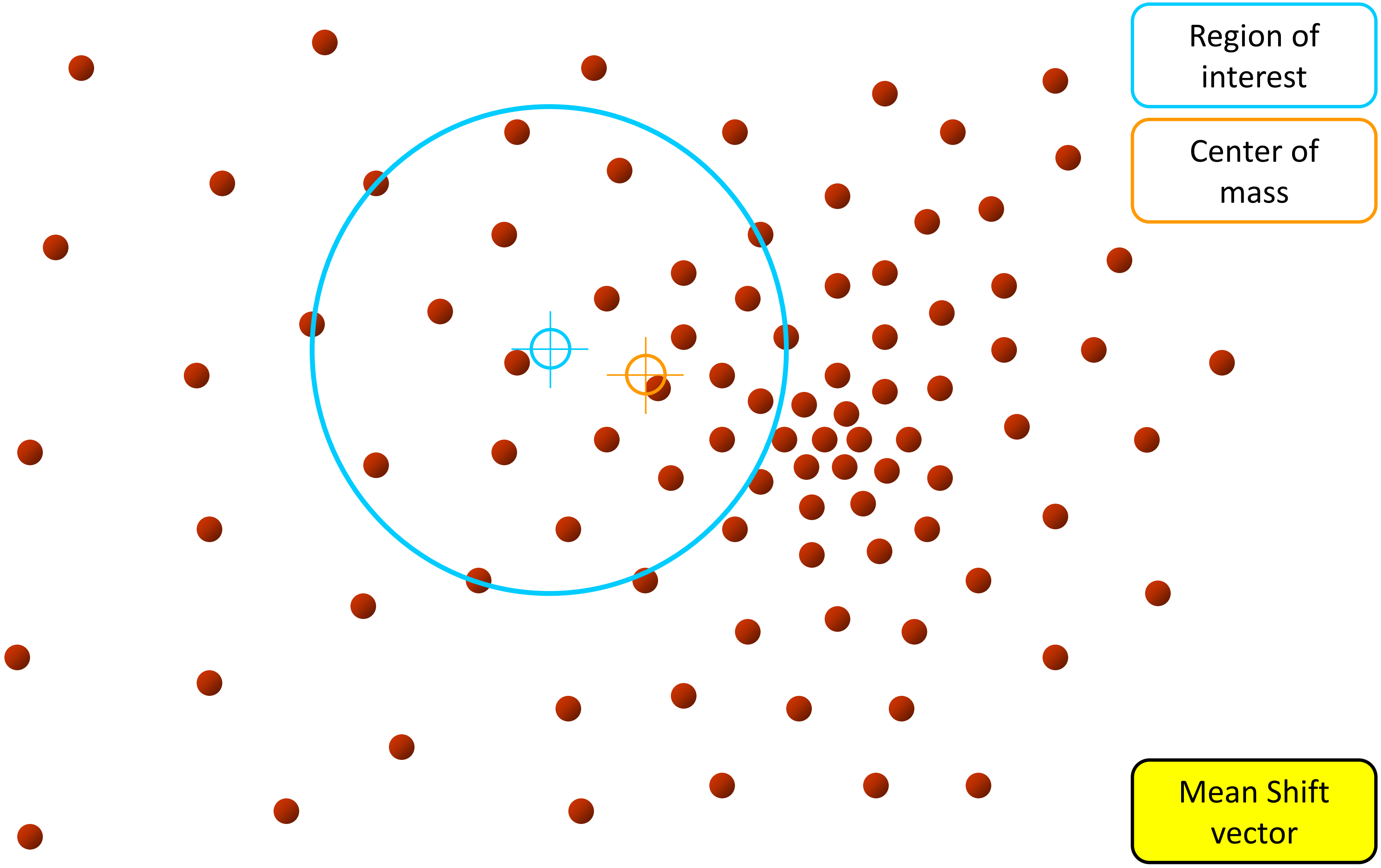
$$K\left(\frac{x - x_i}{h}\right) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-x_i)^2}{2h^2}}.$$

Mean shift



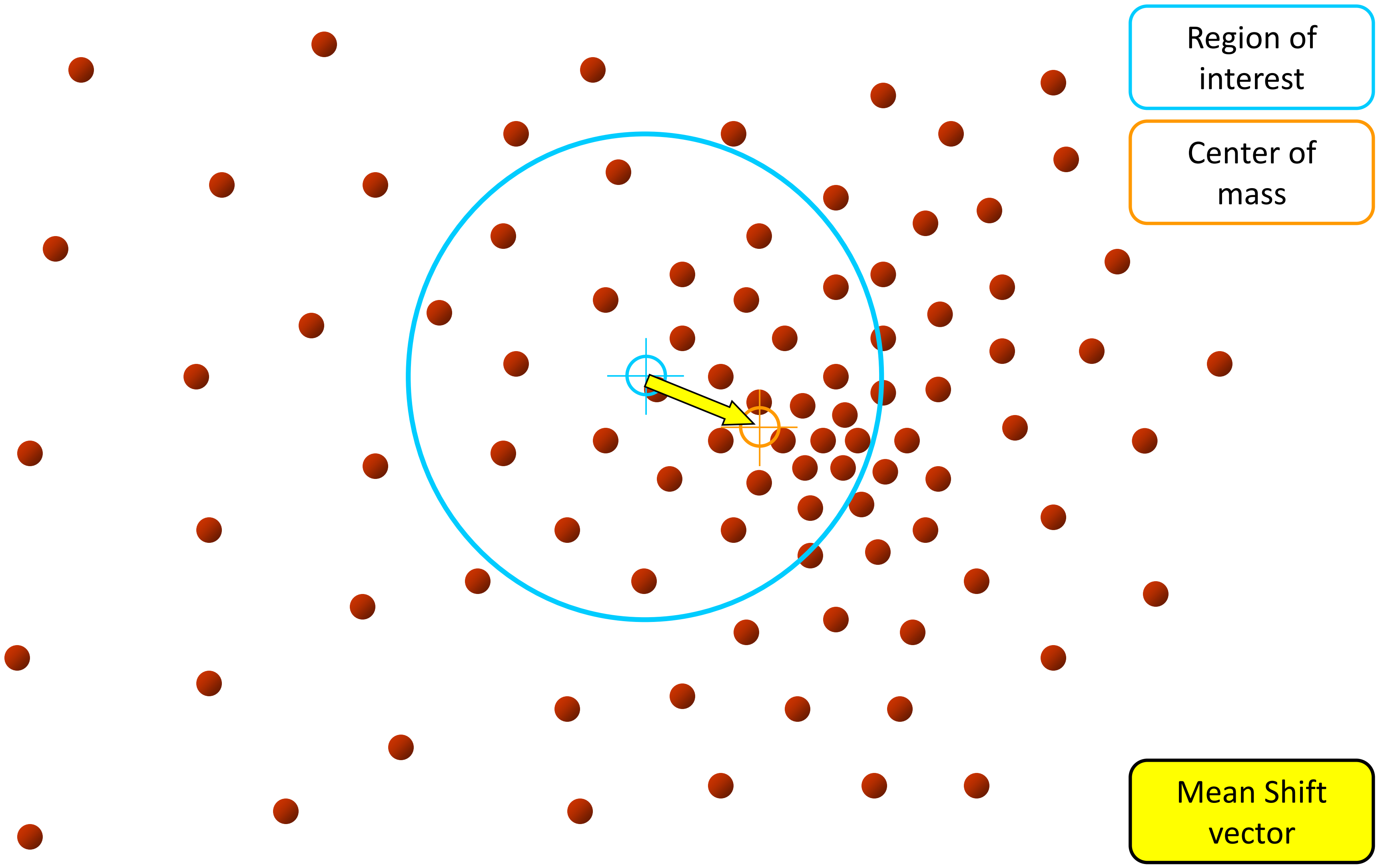
Slide by Y. Ukrainitz & B. Sarel

Mean shift



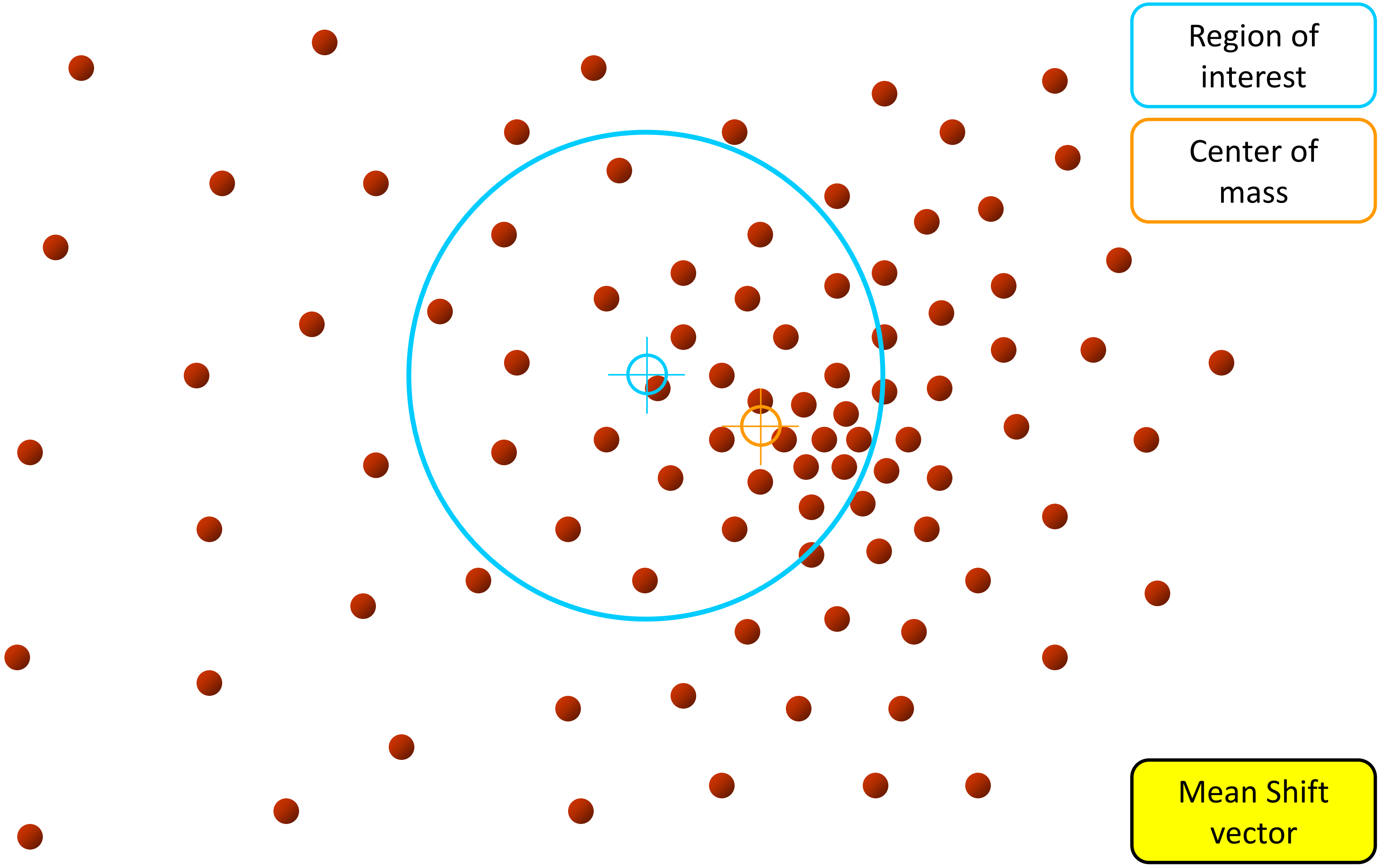
Slide by Y. Ukrainitz & B. Sarel

Mean shift



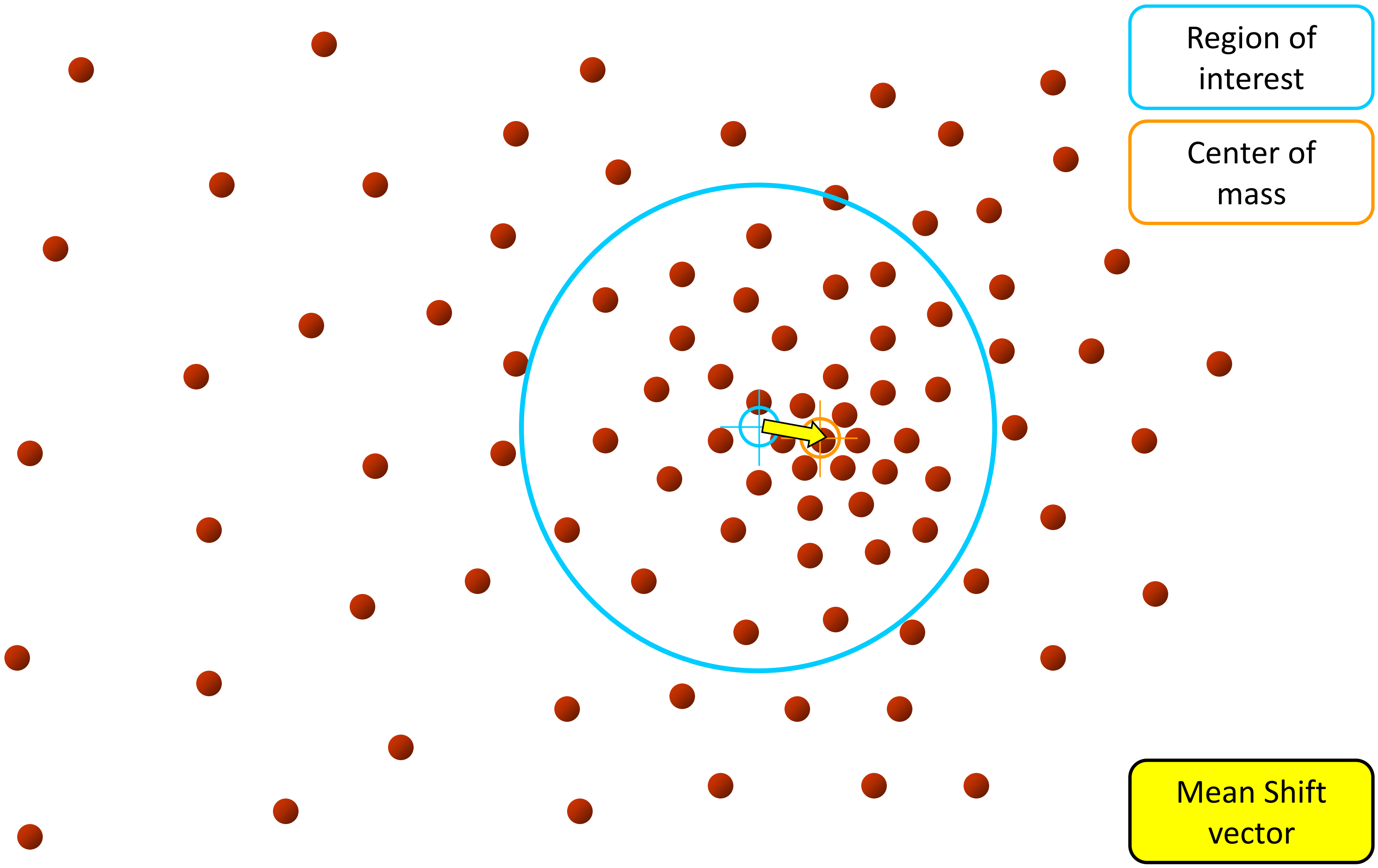
Slide by Y. Ukrainitz & B. Sarel

Mean shift



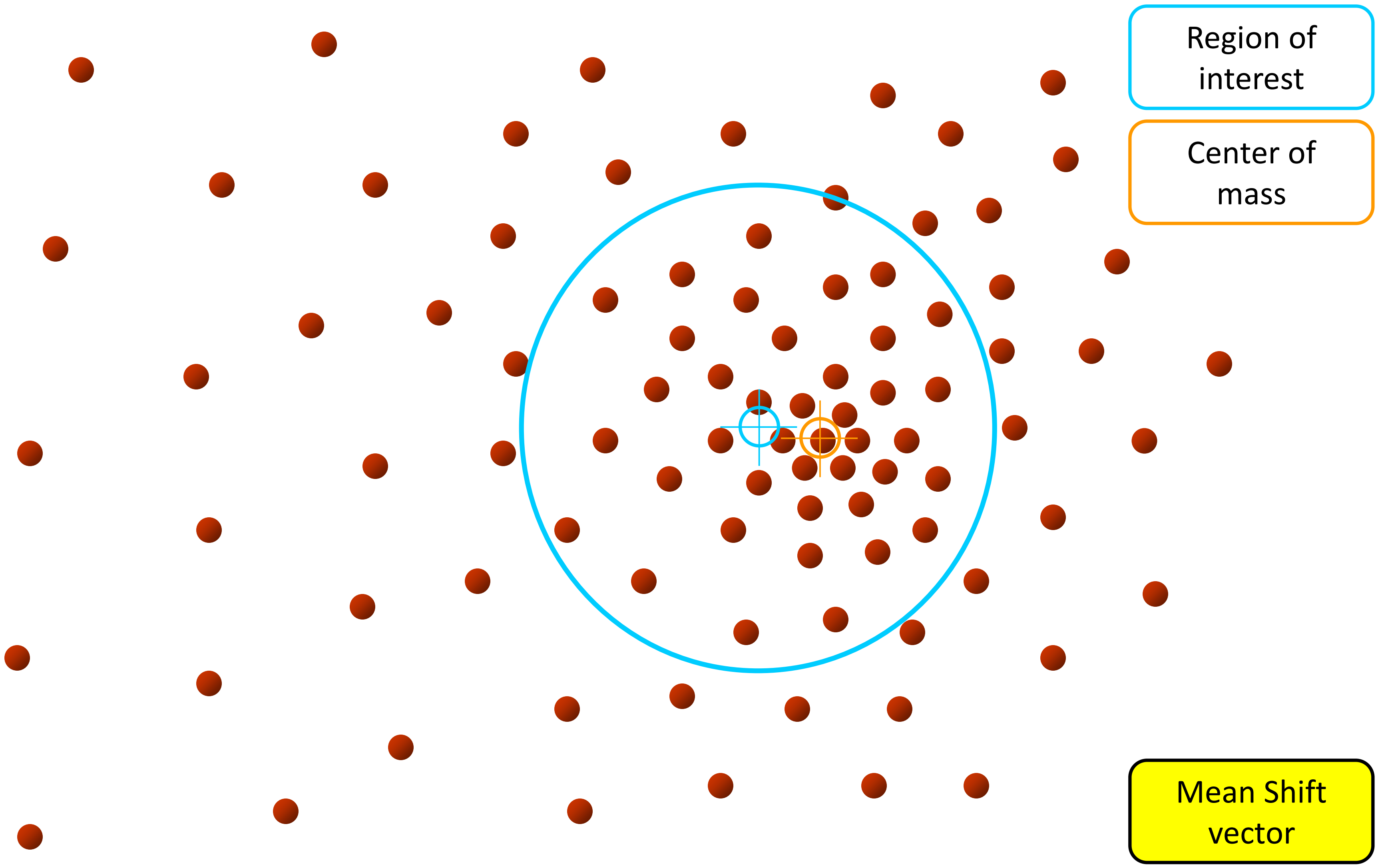
Slide by Y. Ukrainitz & B. Sarel

Mean shift



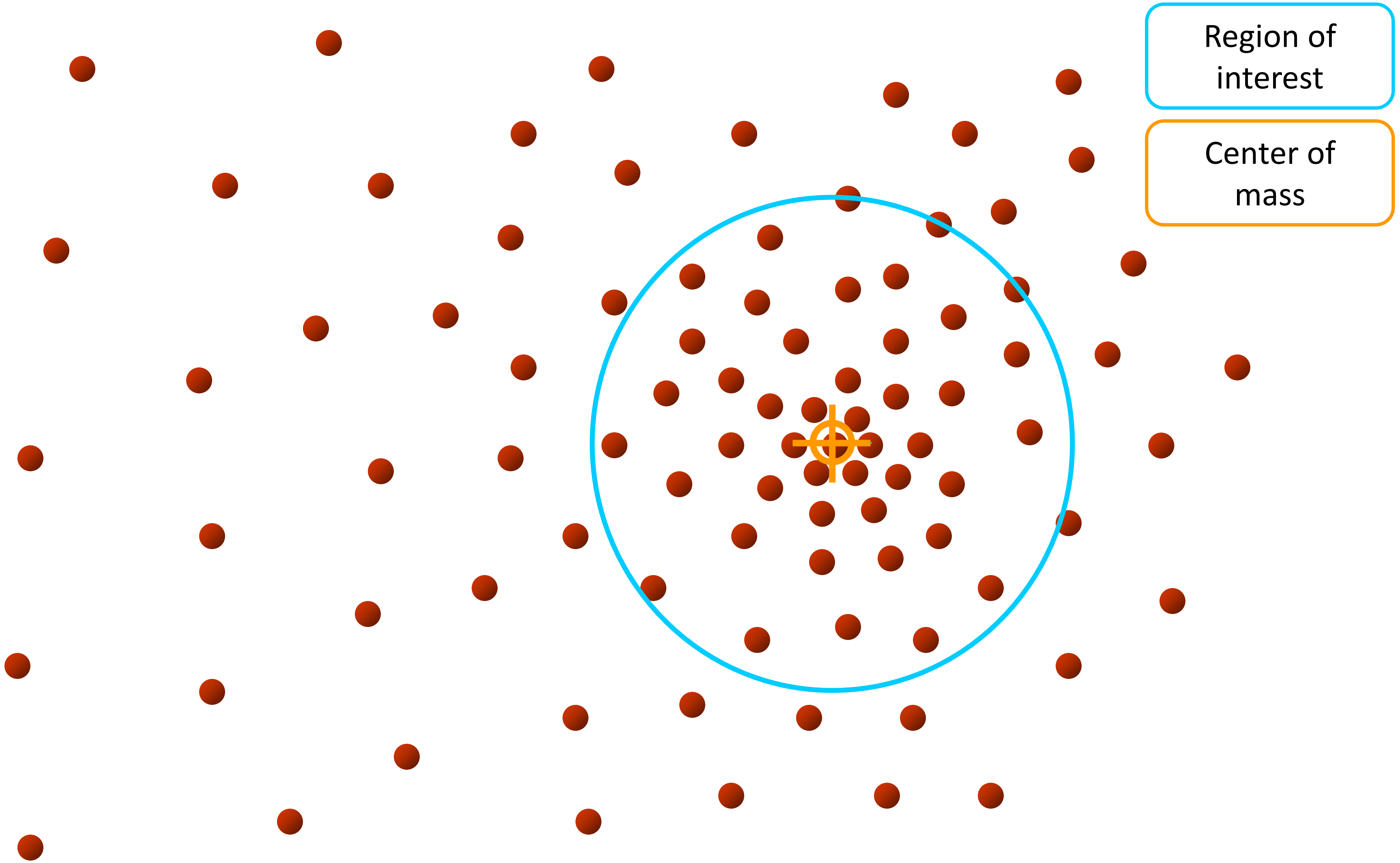
Slide by Y. Ukrainitz & B. Sarel

Mean shift



Slide by Y. Ukrainitz & B. Sarel

Mean shift



Slide by Y. Ukrainitz & B. Sarel

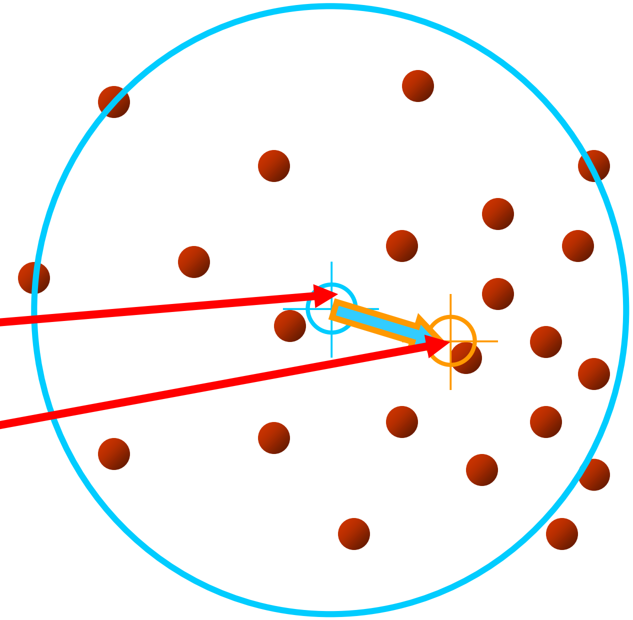
Computing the Mean Shift

<u>Simple Mean Shift procedure</u>:

- Compute mean shift vector

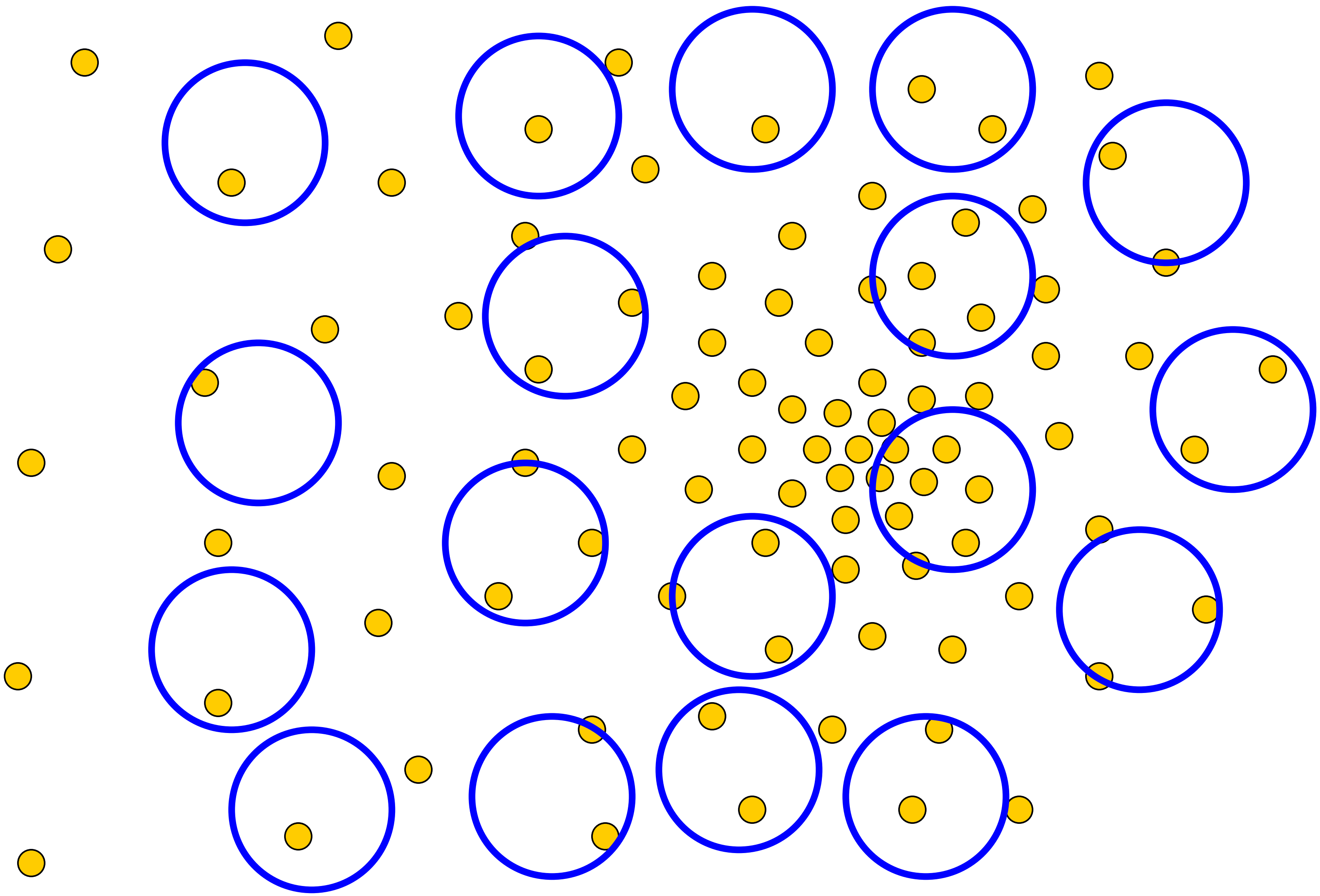
- Translate the Kernel window by **m(x)**

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



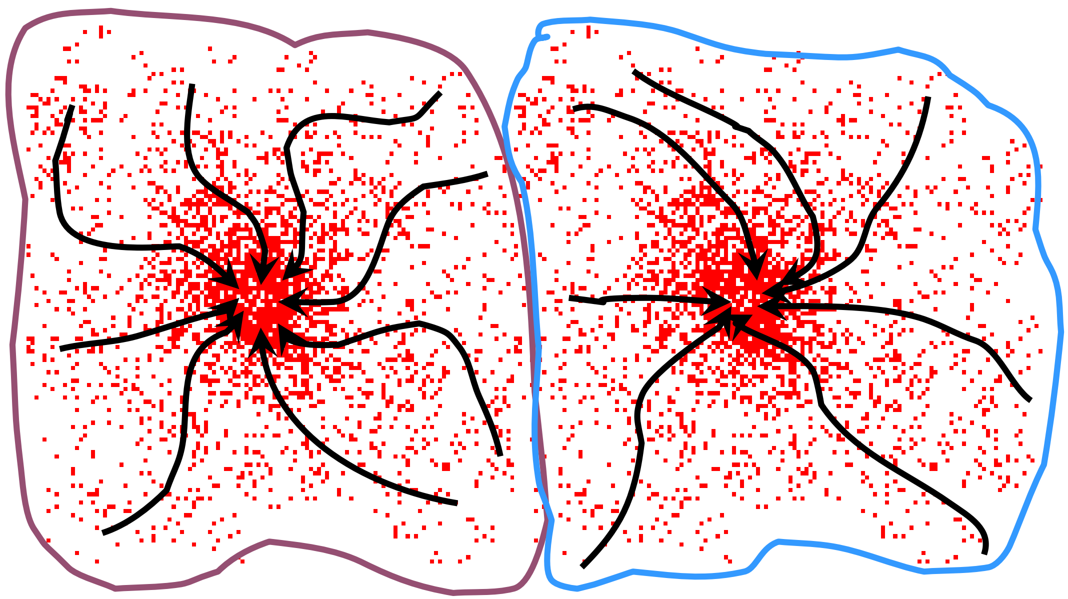
Slide by Y. Ukrainitz & B. Sarel

Real Modality Analysis



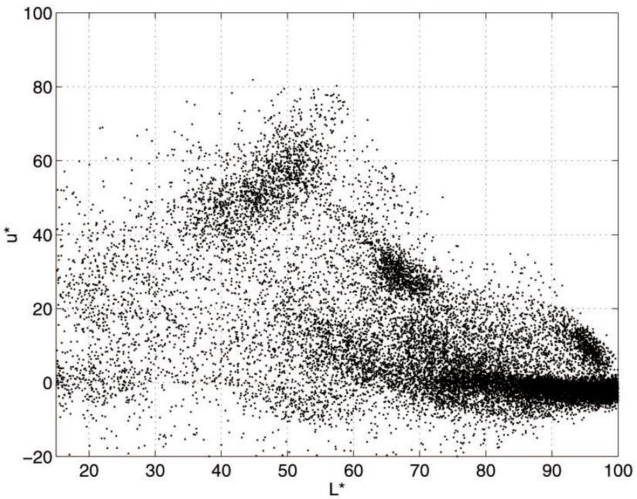
Attraction basin

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

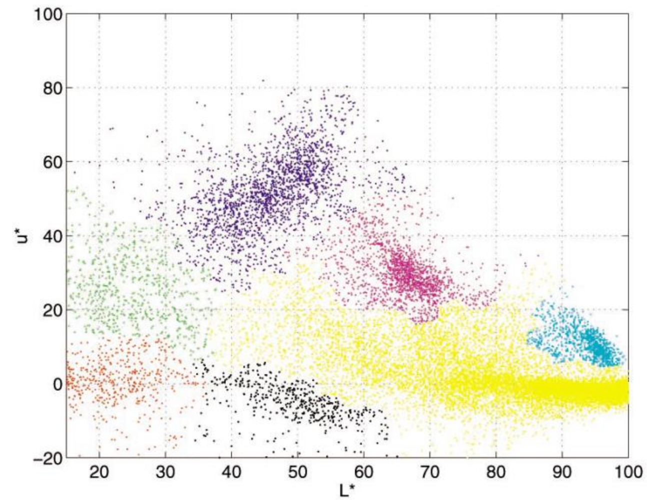


Slide by Y. Ukrainitz & B. Sarel

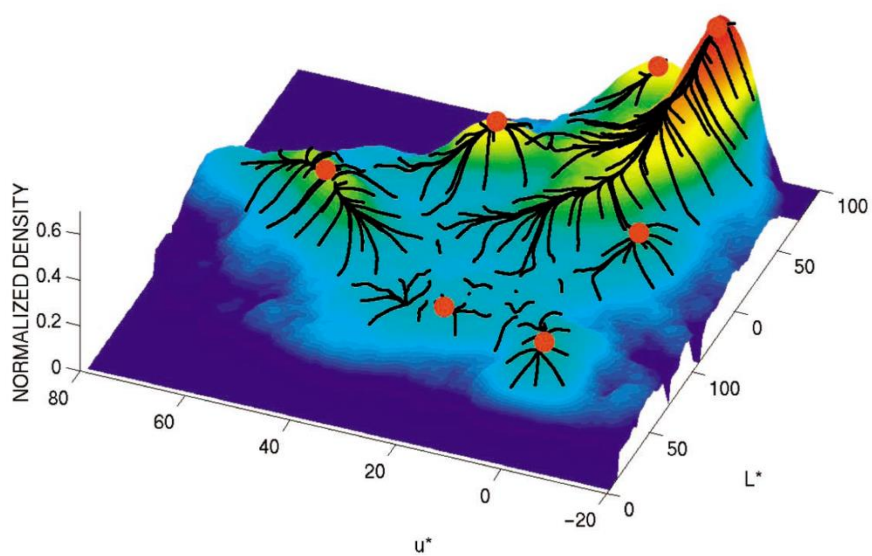
Attraction basin



(a)



(b)

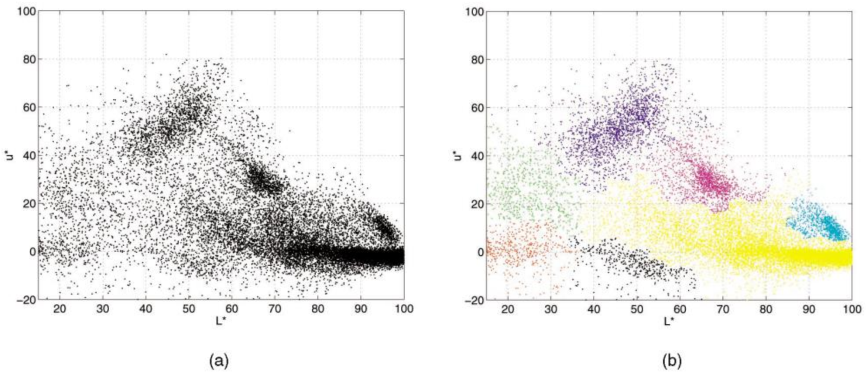


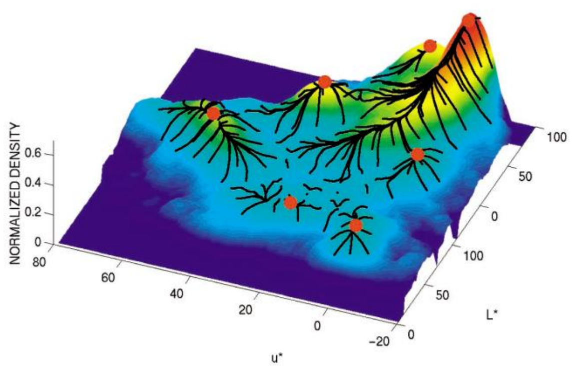
Mean shift clustering

- 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

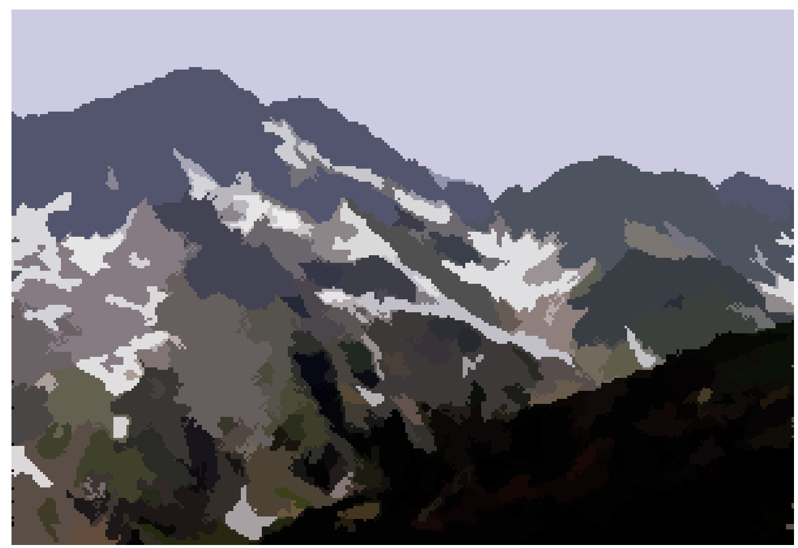
- Compute features for each pixel (color, gradients, texture, etc); also store each pixel's position
- Set kernel size for features K_f and position K_s
- Initialize windows at individual pixel locations
- Perform mean shift for each window until convergence
- Merge modes that are within width of K_f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



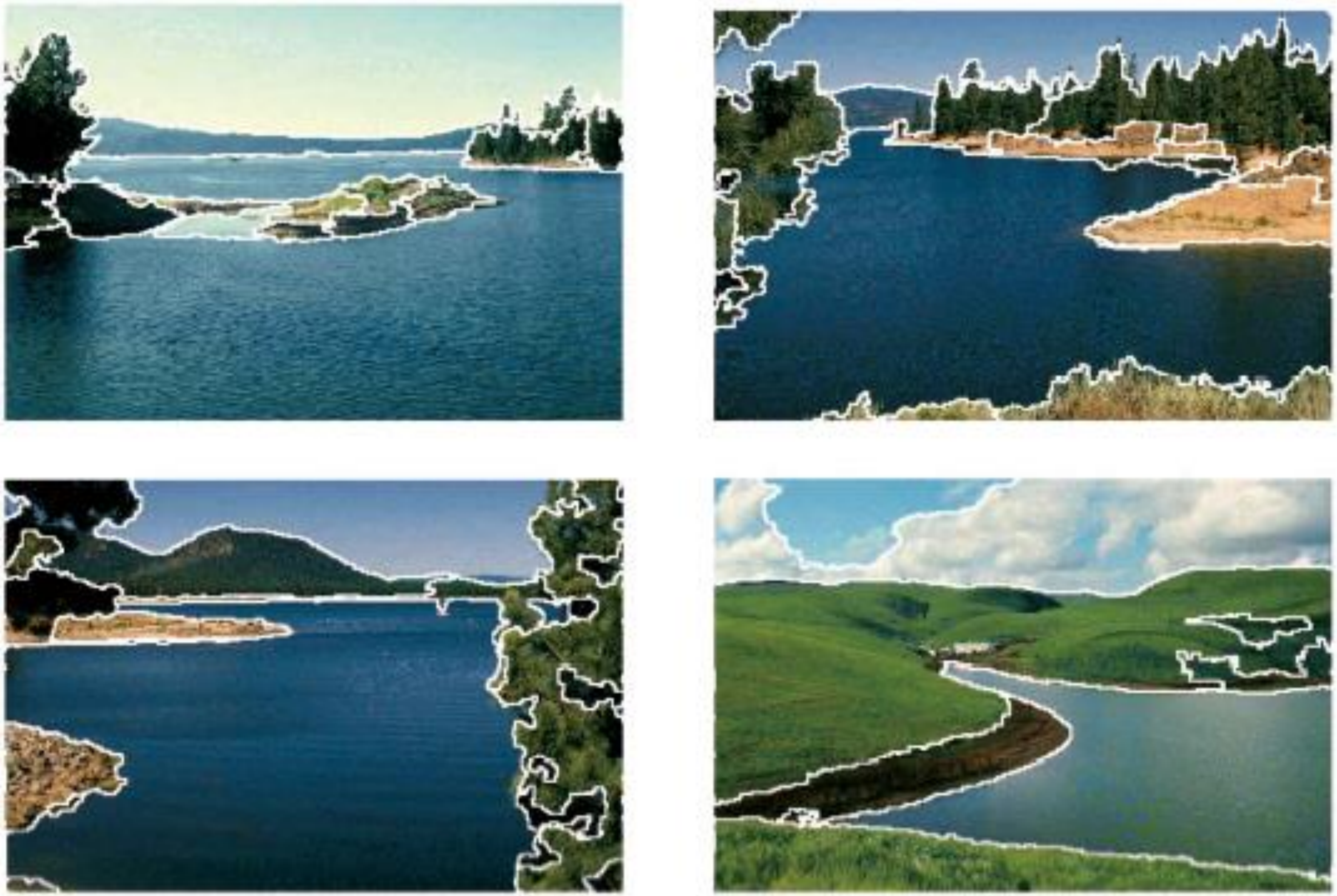


Mean shift segmentation results



http://www.caip.rutgers.edu/~comanici/MSPAMI/msPamiResults.html



http://www.caip.rutgers.edu/~comanici/MSPAMI/msPamiResults.html

Mean-shift: other issues

- Speedups
  - Binned estimation – replace points within some “bin” by point at center with mass
  - Fast search of neighbors – e.g., k-d tree or approximate NN
  - Update all windows in each iteration (faster convergence)
- Other tricks
  - Use kNN to determine window sizes adaptively
- Lots of theoretical support

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

Mean shift pros and cons

- Pros
  - Good general-purpose segmentation
  - Flexible in number and shape of regions
  - Robust to outliers
  - General mode-finding algorithm (useful for other problems such as finding most common surface normals)
- Cons
  - Have to choose kernel size in advance
  - Not suitable for high-dimensional features
- When to use it
  - Oversegmentation
  - Multiple segmentations
  - Tracking, clustering, filtering applications
    - D. Comaniciu, V. Ramesh, P. Meer: *Real-Time Tracking of Non-Rigid Objects using Mean Shift*, *Best Paper Award*, IEEE Conf. Computer Vision and Pattern Recognition (CVPR'00), Hilton Head Island, South Carolina, Vol. 2, 142-149, 2000

Mean-shift reading

- Nicely written mean-shift explanation (with math)

  http://saravananthirumuruganathan.wordpress.com/2010/04/01/introduction-to-mean-shift-algorithm/
  - Includes .m code for mean-shift clustering
- Mean-shift paper by Comaniciu and Meer

  http://www.caip.rutgers.edu/~comanici/Papers/MsRobustApproach.pdf
- Adaptive mean shift in higher dimensions

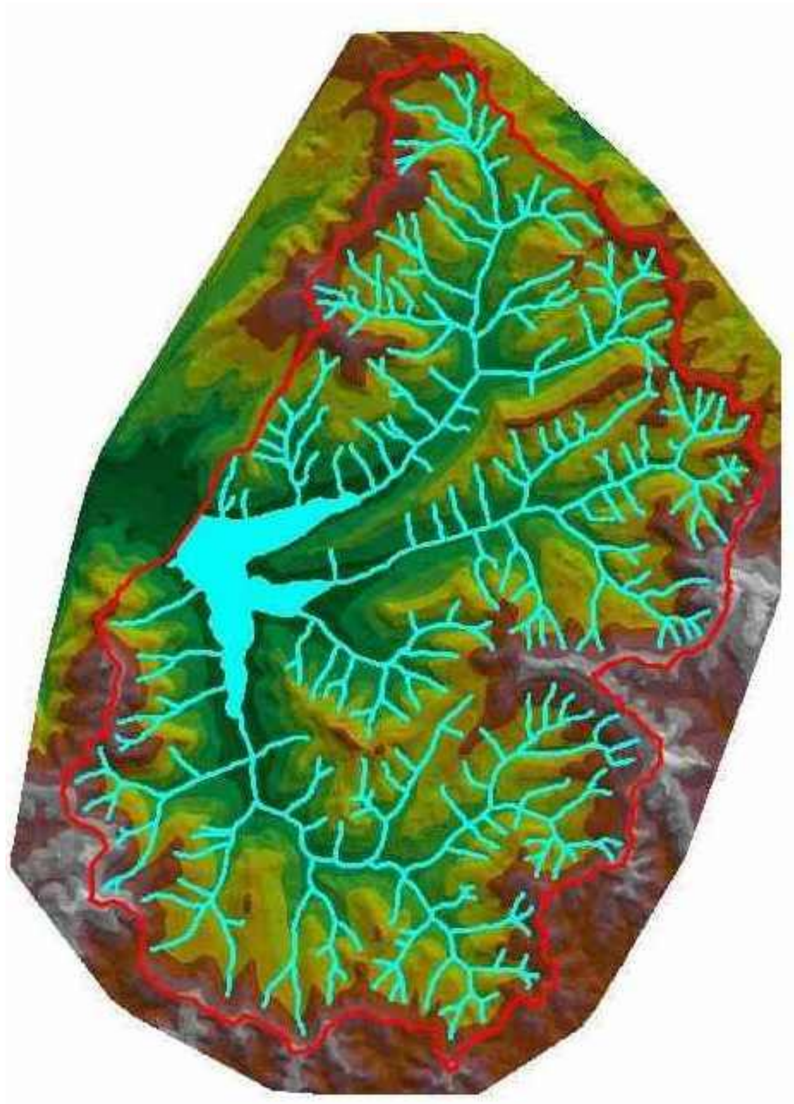
  http://mis.hevra.haifa.ac.il/~ishimshoni/papers/chap9.pdf

Superpixel algorithms

- Goal: divide the image into a large number of regions, such that each regions lie within object boundaries

- Examples
  - Watershed
  - Felzenszwalb and Huttenlocher graph-based
  - Turbopixels
  - SLIC

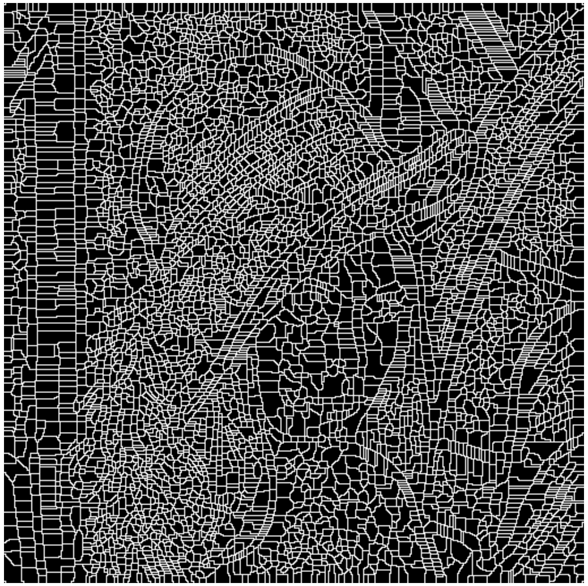
Watershed algorithm



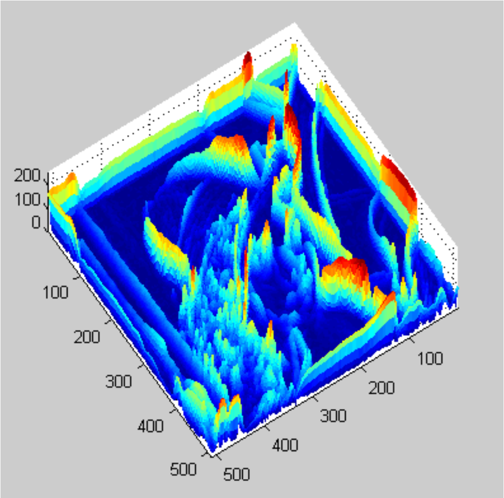
Watershed segmentation

Image

Gradient



Watershed boundaries



Meyer's watershed segmentation

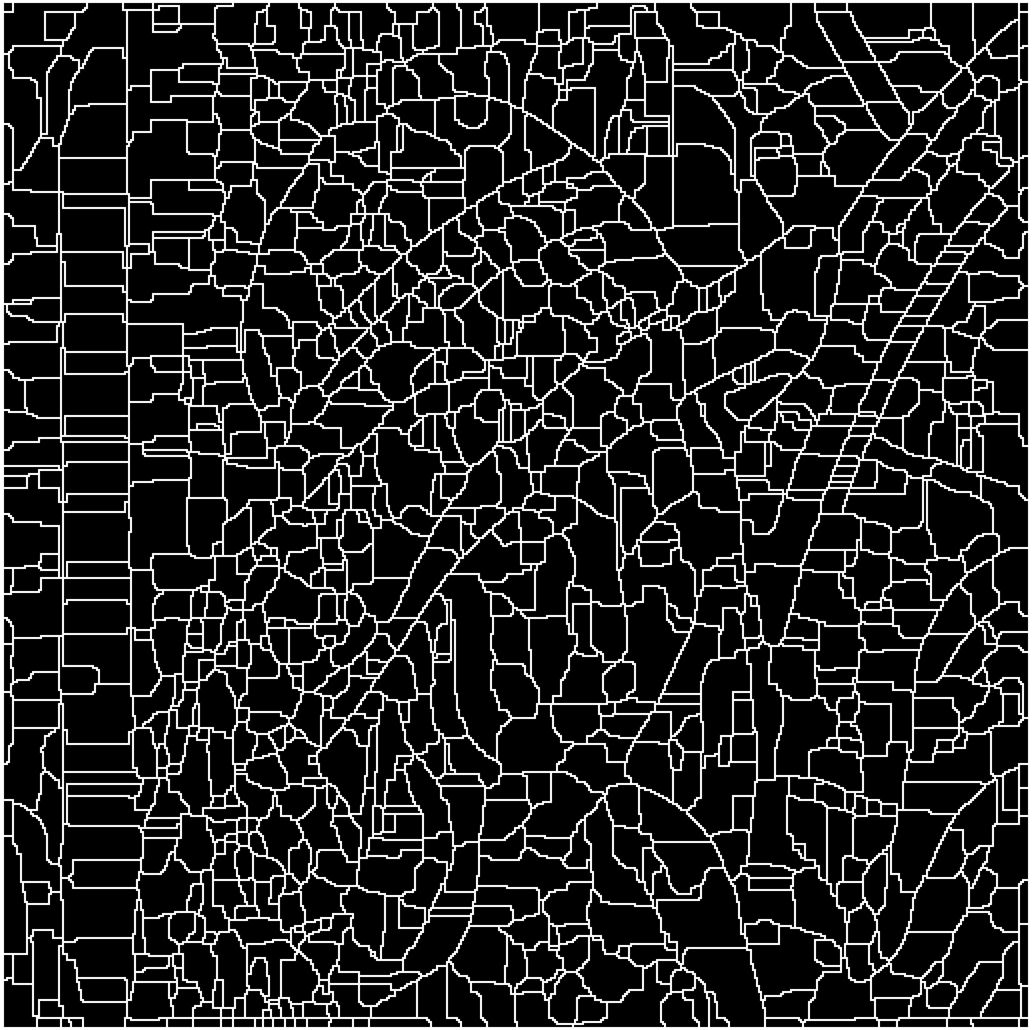
1. Choose local minima as region seeds
2. Add neighbors to priority queue, sorted by value
3. Take top priority pixel from queue
    1. If all labeled neighbors have same label, assign that label to pixel
    2. Add all non-marked neighbors to queue
4. Repeat step 3 until finished (all remaining pixels in queue are on the boundary)

Matlab: seg = watershed(bnd_im)

Meyer 1991

Simple trick

- Use Gaussian or median filter to reduce number of regions



Watershed usage

- Use as a starting point for hierarchical segmentation
  - Ultrametric contour map (Arbelaez 2006)

- Works with any soft boundaries
  - Pb (w/o non-max suppression)
  - Canny (w/o non-max suppression)
  - Etc.

Watershed pros and cons

- Pros
  - Fast (< 1 sec for 512x512 image)
  - Preserves boundaries
- Cons
  - Only as good as the soft boundaries (which may be slow to compute)
  - Not easy to get variety of regions for multiple segmentations
- Usage
  - Good algorithm for superpixels, hierarchical segmentation

Felzenszwalb and Huttenlocher: Graph-Based Segmentation

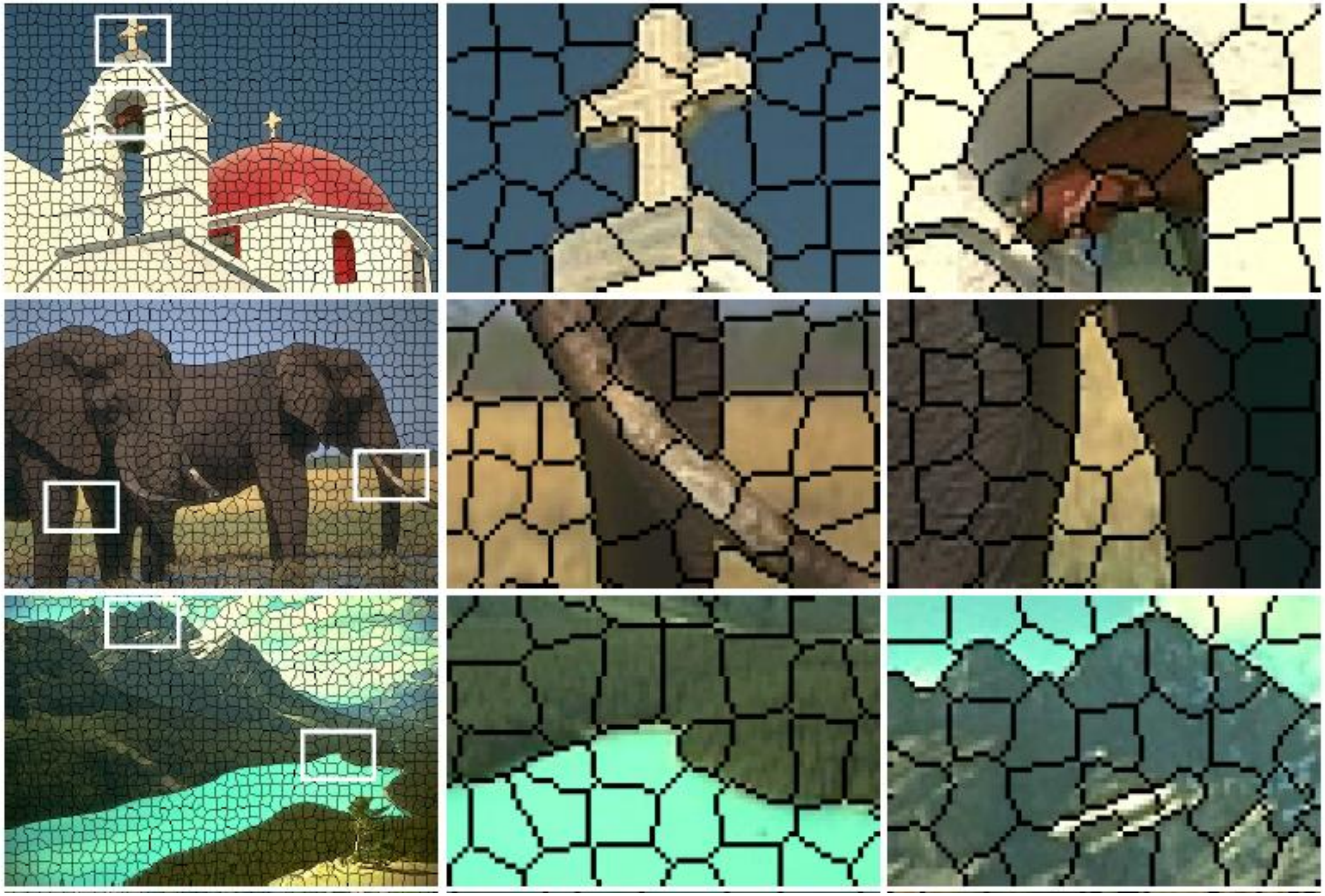
http://www.cs.brown.edu/~pff/segment/

+ Good for thin regions
+ Fast
+ Easy to control coarseness of segmentations
+ Can include both large and small regions
 - Often creates regions with strange shapes
 - Sometimes makes very large errors

Turbo Pixels: Levinstein et al. 2009

http://www.cs.toronto.edu/~kyros/pubs/09.pami.turbopixels.pdf

Tries to preserve boundaries like watershed but to produce more regular regions



SLIC (Achanta et al. PAMI 2012)

http://infoscience.epfl.ch/record/177415/files/Superpixel_PAMI2011-2.pdf

1. Initialize cluster centers on pixel grid in steps S
   - Features: Lab color, x-y position
2. Move centers to position in 3x3 window with smallest gradient
3. Compare each pixel to cluster center within 2S pixel distance and assign to nearest
4. Recompute cluster centers as mean color/position of pixels belonging to each cluster
5. Stop when residual error is small

+ Fast 0.36s for 320x240
+ Regular superpixels
+ Superpixels fit boundaries
- May miss thin objects
- Large number of superpixels

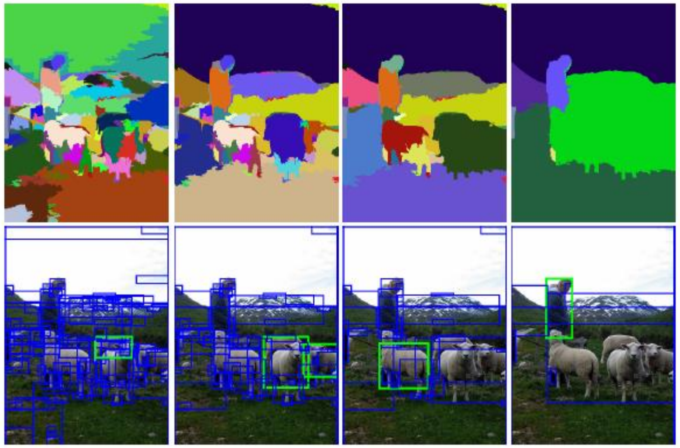
Choices in segmentation algorithms

- Oversegmentation
  - Watershed + Structure random forest
  - Felzenszwalb and Huttenlocher 2004
    http://www.cs.brown.edu/~pff/segment/
  - SLIC
  - Turbopixels
  - Mean-shift

- Larger regions (object-level)
  - Hierarchical segmentation (e.g., from Pb)
  - Normalized cuts
  - Mean-shift
  - Seed + graph cuts (discussed later)

Multiple segmentations

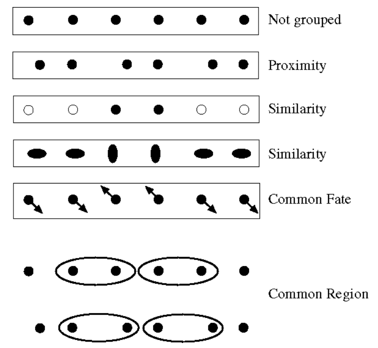
- When creating regions for pixel classification or object detection, don't commit to one partitioning



- Strategies:
  - Hierarchical segmentation
    - Occlusion boundaries hierarchy: Hoiem et al. IJCV 2011 (uses trained classifier to merge)
    - Pb+watershed hierarchy: Arbeleaz et al. CVPR 2009
    - Selective search: FH + agglomerative clustering
    - Superpixel hierarchy
  - Vary segmentation parameters
    - E.g., multiple graph-based segmentations or mean-shift segmentations
  - Region proposals
    - Propose seed superpixel, try to segment out object that contains it
(Endres Hoiem ECCV 2010, Carreira Sminchisescu CVPR 2010)

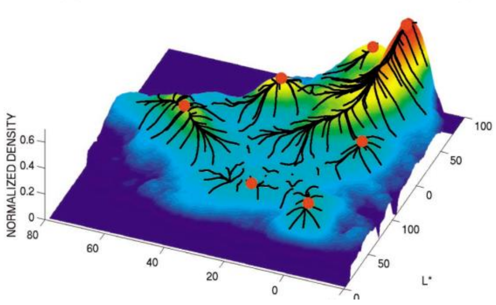
Things to remember

- Gestalt cues and principles of organization



- Uses of segmentation
  - Efficiency
  - Better features
  - Propose object regions
  - Want the segmented object

- Mean-shift segmentation
  - Good general-purpose segmentation method
  - Generally useful clustering, tracking technique



- Watershed segmentation
  - Good for hierarchical segmentation
  - Use in combination with boundary prediction