# 3. Data Structures for Image Analysis

LAKSHMAN@OU.EDU

# Different formulations

- Can be advantageous to treat a spatial grid as a:
  - Levelset
  - Matrix
  - Markov chain
  - Topographic map
  - Relational structure
  - Pyramid or quadtree
  - Parametric approximation

- Different formulations help in different contexts

# Level set

- Organizes the pixels in a grid by pixel value
  - Associative array
    - Between pixel value
    - And list of pixels that have that value
  - A “level” is a key-value pair

```
public class LevelSet {
  private Map<Integer,List<Pixel>> data = new TreeMap<Integer,
      List<Pixel>>();
  public void add(Pixel p){
    List<Pixel> level = data.get(p.getValue());
    if ( level == null ){
      level = new ArrayList<Pixel>();
      data.put(p.getValue(), level);
    }
    level.add(p);
  }
  public Map.Entry<Integer,List<Pixel>>[] getLevels(){
    return data.entrySet().toArray(new Map.Entry[0]);
  }
}
```

- Could also use a random-access array if range is known

# Creating a level set

- To create a level set, traverse the grid once

```
LatLonGrid input = ...;
LevelSet levelset = new LevelSet();
for (int i=0; i < input.getNumLat(); ++i){
  for (int j=0; j < input.getNumLon(); ++j){
    if ( input.getValue(i,j) != input.getMissing() ){
      levelset.add(new Pixel(i,j,input.getValue(i,j)));
    }
  }
}
```

- How would you use the level set to find top 10 pixels?

# Top 10 using a level set

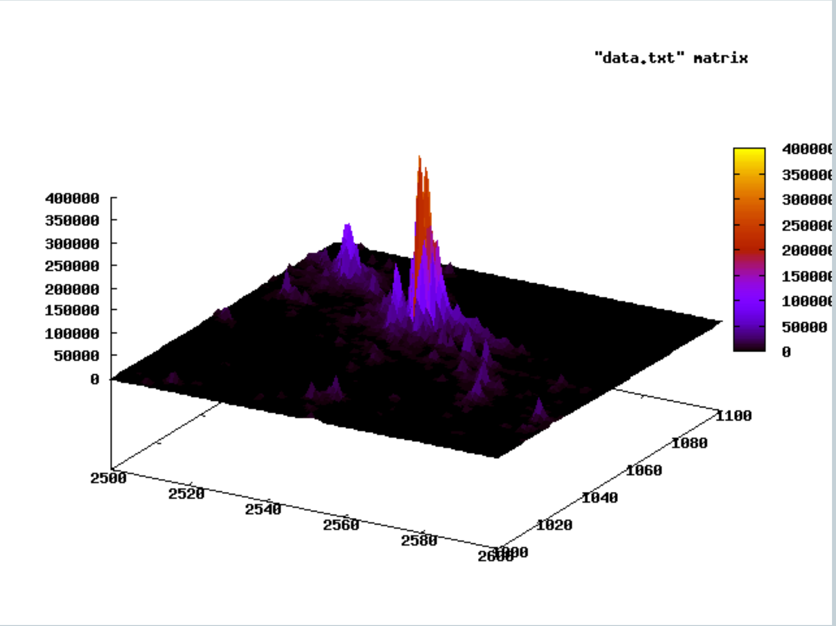
- Accumulate until you reach 10 pixels

```
// find the top n pixels
Map.Entry<Integer, List<Pixel>>[] levels = levelset.getLevels();
List<Pixel> result = new ArrayList<Pixel>();
int curr = levels.length;
while (result.size() < nth && curr > 0){
  curr = curr - 1; // next
  result.addAll(levels[curr].getValue()); // all pixels at this
      level
}
```

- Under what circumstances is this a better solution than a selection sort algorithm?

# Topographic Grids

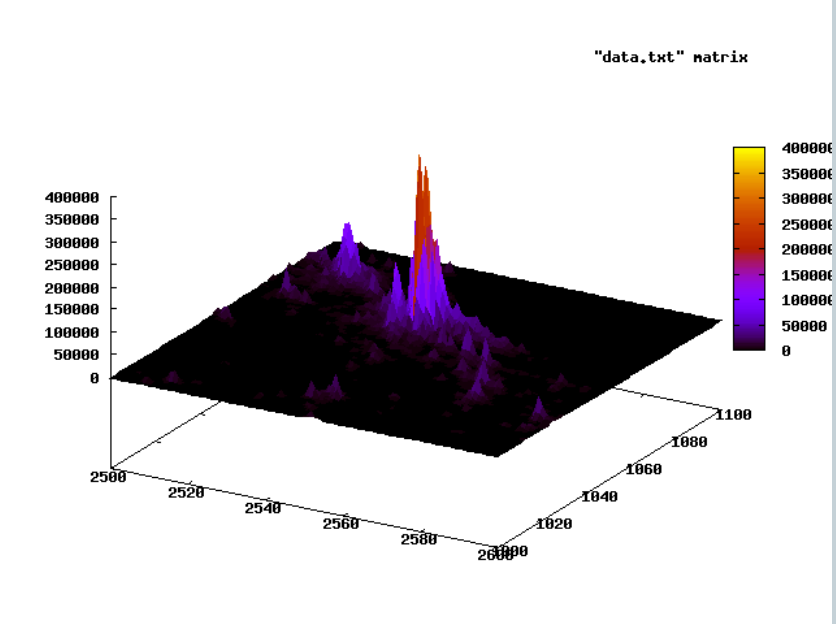
- Another way to think of an image is as a topographic map
  - For example, the population density dataset can be thought of as a surface where the height at a pixel is proportional to the number of people living in that pixel

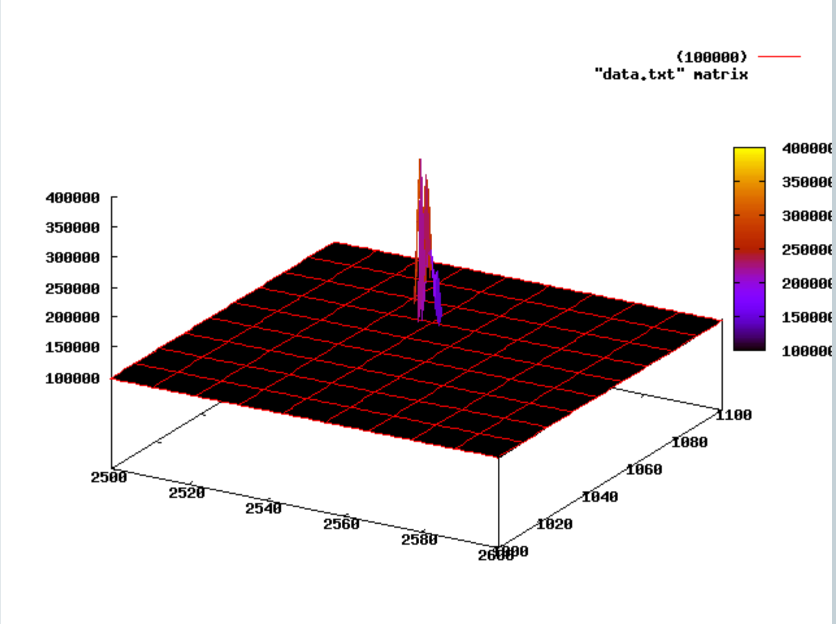



- Where is this representation useful?

# Thresholding

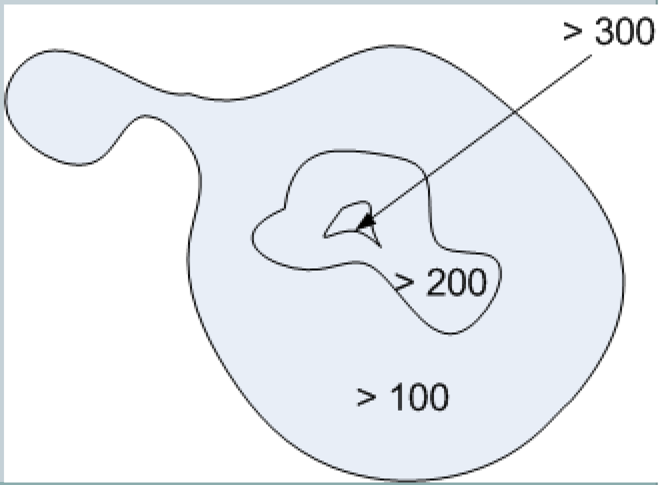
- Thresholding an image implicitly treats the image topographically



# Uses of topographic representation

- Topographic representations are useful when there are no strong edges
  - Physical processes often do not have strong edges
  - Yet, object identification depends on the presence of edges
- The topographic representation provides two important tools:
  - Contours can take the place of edges
  - Watersheds can replace objects
- Techniques to identify contours and watersheds exist
  - Will talk about this in this course



# Topographic distances

- You might want to consider distances taking into account topography

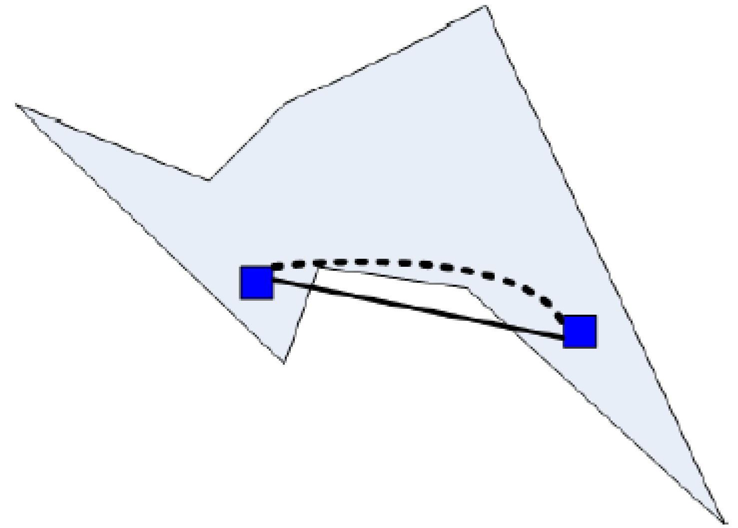
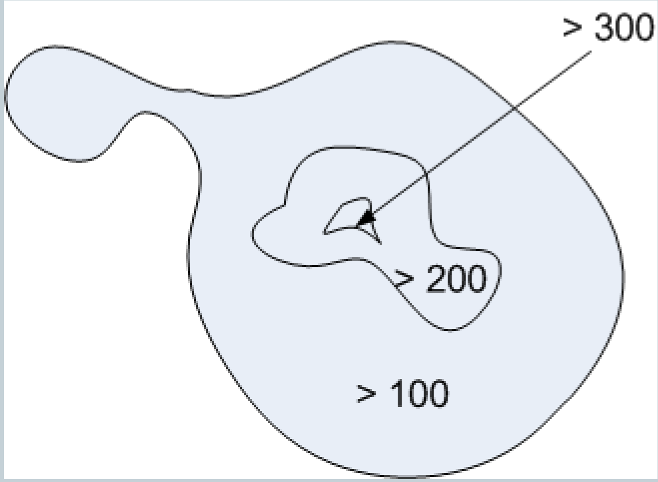


**Fig. 3.3** A topographical surface brings about concepts such as geodesic distances: distances between pixels along paths that lie entirely above a certain value. The dotted line is a geodesic distance whereas the straight line is an Euclidean distance (the shaded area denotes pixels that lie above threshold).

# Relational Structure

- When objects are dependent on topographic thresholds
  - Different thresholds give rise to hierarchical objects
  - A parent-child relationship exists between objects
- Tree representation lends itself to discussions of object-scale
  - Algorithms exist to identify hierarchical objects in images
    - We'll talk about these contiguity-enhanced clustering algorithms



# Markov Chain

- What's a 1st order Markov Random Field?
  - A field where the value at x_k depends on x_{k-1} but not on x_{k-2}, etc. other than indirectly
    - Spatial dependence is of the 1st order only
  - Greatly simplifies many image processing operations if we assume that the grid follows Markov properties
    - Enough to just consider the neighbors of a pixel

- In most of this course, we'll assume that your spatial grids were formed by a 1st order Markov process
  - Rationale behind many of the techniques that we'll discuss

# What's not Markov?

- Your data are not Markov if:
  - You have gaps within your grid
    - Have to fill in (interpolate) the gaps before applying any image processing technique
  - You have speckle noise in your data
    - Speckle is uncorrelated with neighboring points
    - Need to speckle-filter your data first
      - Most common speckle filter:  a median filter
  - Second order effects predominate in your data
    - Such as shadows caused by overshooting tops in satellite visible imagery
- Remember that most image processing operations assume Markov
  - Think carefully about where the Markov assumption fails and fix them before doing any automated analysis

# Direct use of Markov

- Run-length encoding

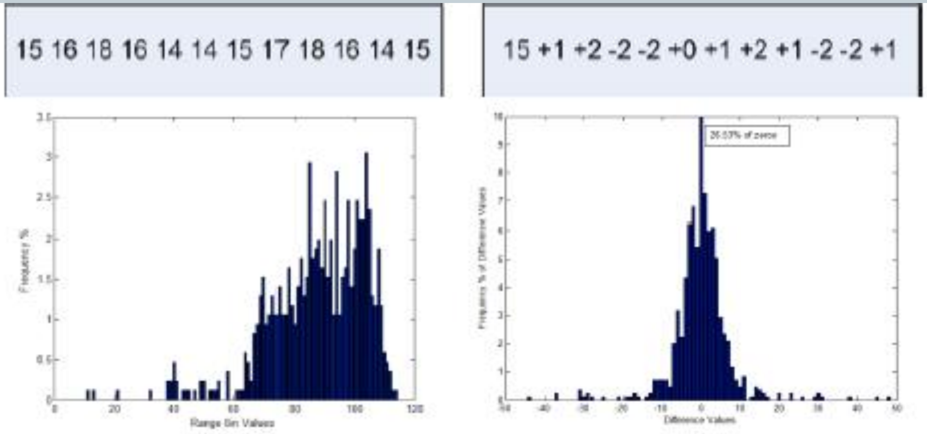



**Fig. 3.5** Run-length encoding replaces data values by deviations from an expected value where the expected value is the same as the previous value. The histogram of such differences is typically narrower than the original data (illustrated here on actual weather radar reflectivity data), allowing the data to be compressed better.

# Matrix formulation

- Can think of a spatial grid as a matrix X
  - Useful for simple scalar operations built into statistical packages
    - max(X), sumRows(X), etc.

- Any other use for matrix notation?

# Matrix formulation

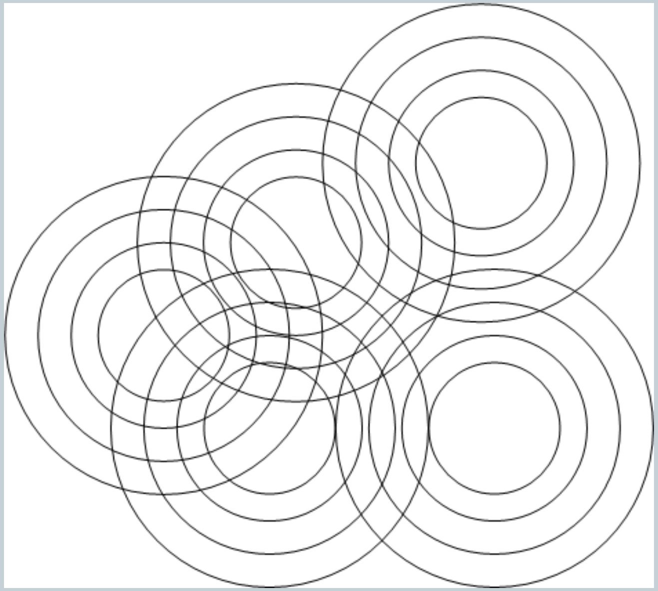
- Recall that in objective analysis the resulting value at a grid point is the weighted average of the inputs
  - Can think of the resulting grid as

$$X = a_1 X_1 + a_2 X_2 + \ldots + a_n X_n$$

  - The left-hand-side “X” is the final image as a matrix
  - What does $X_k$ look like?

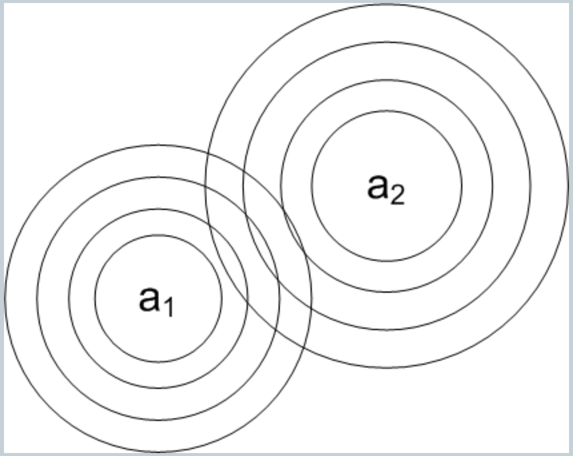
# Objective Analysis contd.

- In the objective analysis problem:
  - The $X_k$ are 2D Gaussians
  - The $a_k$ are the amplitudes of the Gaussians
  - The Gaussians' centers are at the station locations
  - Their sigmas are identical: equal to the radius of influence



# Inverse Problem

- What if you are given a spatial grid of data and need to do inverse?
  - For example, have concentration of pollutants over a city
    - e.g: You know the centers and sigmas
      - i.e. the locations of chemical plants that release that pollutant
      - and the range of influence of each plant
  - Want to find the contribution due to each plant



# Radial Basis Functions

- Can treat the X's as basis functions

$$X = a_1 X_1 + a_2 X_2 + \ldots + a_n X_n$$

  - The final grid is then a linear combination of Radial Basis Functions
- If the X's are known, the least-squares optimal amplitudes can be found using singular value decomposition

# RBF Formulation

- Treat the result image ('y') as a matrix with p rows and 1 column
  - p is the number of pixels in the image
  - The 2D Gaussian function (x is a 2x1 vector)
    - $h(x) = \exp(-(x-c)^T(x-c)/r^2)$
    - The center and r of each Gaussian is known
  - y at the row corresponding to x = $[\, a_1 h_1(x) + \ldots + a_m h_m(x) \,]$
    - m is the number of Gaussians
- The optimal amplitudes to minimize the least squares error is:
  - $a = (H^T H)^{-1} H^T y$
- Introduction to Radial Basis Function Networks by MJL Orr
  - Do a Google search and download from psu.edu
  - Read pages 6-12 in order to do assignment

# Parametric Approximation

- If you identify objects using the topographic formulation, you throw away data outside the peaks
  - The data corresponding to low values do not form part of the object
- Can approximate an image by a 2D function
  - The RBF was an attempt at doing this
  - But limited to Gaussians of known centers and sigmas
    - Not all that applicable
  - Gamma functions and Gaussian Mixture Models are more flexible
- Parametric approximations permit some interesting forms of analyses
  - Will look at ways to do the approximation and what can be achieved with those

# Assignment Part 1: Radial Basis Functions

- Simulate and solve the 2-source pollutant problem:
  - Simulate a spatial grid by objectively analyzing two point sources
  - Starting with the known locations and sigmas, determine the amplitude of the Gaussians using RBFs
    - How close do you get?
    - Extra credit: How does this change with noise?

# Extending a RBF

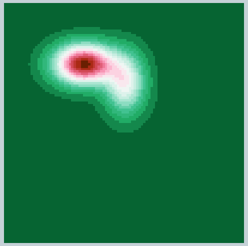
- Can you think of a spatial inverse problem where a RBF can be used?

- Recall that the RBF has an optimal solution only if you know the centers and sigmas apriori
  - Will the centers and sigmas be known apriori?
  - How could you address this issue?

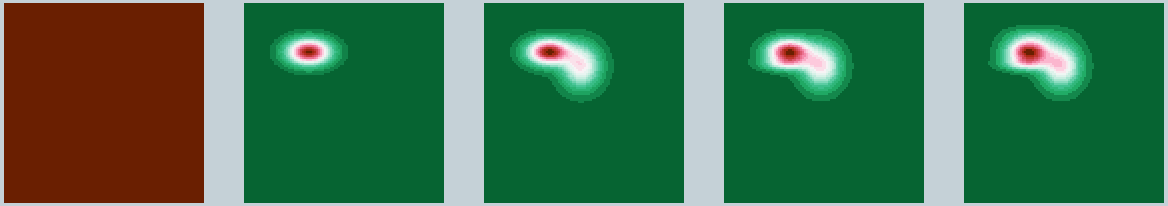
# Projection Pursuit

- An iterative procedure that is not optimal, but usually "good enough"
- Find centers and sigmas one-by-one
  1. Find first center and sigma
  2. Compute amplitude of the RBFs
  3. Compute image approximation from RBF and compute error
  4. If error > error-threshold or number-of-RBF < max-no-of-RBF
     - Find next center and sigma; Add to list of centers and sigmas
     - Go to step 2
- How to find a "good-enough" center and sigma?
  - Could find spatial mean (centroid) and spatial variance
  - Or use peak of error code to locate center and use distance from peak to half-its-value to come up with a  variance estimate
    - Better approach since RBFs are "local" estimators

# Projection Pursuit on Simulated Input



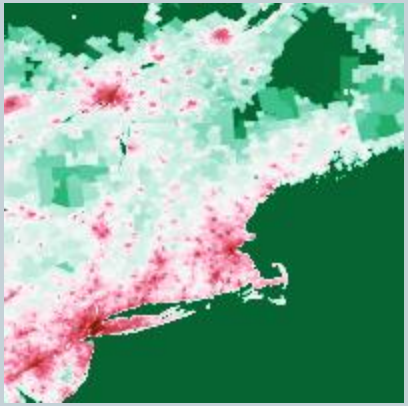
```
true RBF#0 center: [25,33 20] sigmax=8.0 sigmay=12.0
true RBF#1 center: [33,50 10] sigmax=12.0 sigmay=8.0
```

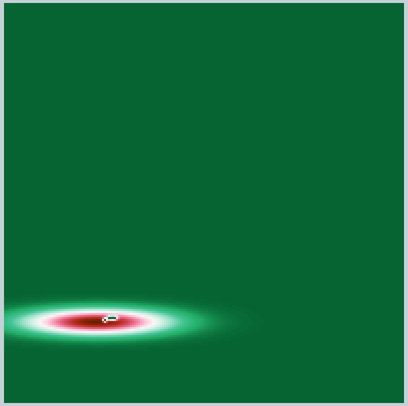


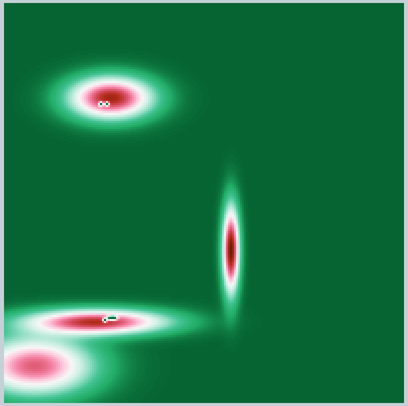
```
RBF#0 center: [24,32 20] sigmax=6.0 sigmay=10.0
RBF#1 center: [31,48 12] sigmax=10.0 sigmay=9.0
RBF#2 center: [30,30 6] sigmax=3.0 sigmay=11.0
RBF#3 center: [16,35 3] sigmax=4.0 sigmay=13.0
```

# Projection Pursuit on real data

- Use projection pursuit to fit RBFs to log-scaled population density







```
RBF#0 loc: [40.8541666669375,-73.8958333324846] ampl=589 sigmax=6.0 sigmay=6.0
RBF#1 loc: [40.8958333336045,-73.7291666658166] ampl=-615 sigmax=1.0 sigmay=1.0
RBF#2 loc: [40.9375000002715,-73.6458333324826] ampl=-265 sigmax=1.0 sigmay=1.0
RBF#3 loc: [40.9375000002715,-73.5624999991486] ampl=-628 sigmax=1.0 sigmay=1.0
RBF#4 loc: [42.3541666669495,-71.1041666657956] ampl=671 sigmax=21.0 sigmay=21.0
RBF#5 loc: [39.937500000026354,-75.18749999991616] ampl=514 sigmax=14.0 sigmay=14.0
RBF#6 loc: [45.5208333336415,-73.6041666658156] ampl=620 sigmax=10.0 sigmay=10.0
RBF#7 loc: [45.3958333336405,-73.6874999991496] ampl=-575 sigmax=1.0 sigmay=1.0
RBF#8 loc: [45.3958333336405,-73.8124999991506] ampl=-607 sigmax=1.0 sigmay=1.0
```

Why are there multiple centers near a point? How can we avoid this?

# Assignment Part 2: Top 5 population centers

- Find the top 5 population centers in North America
  - Mark the approximate metropolitan area of each

# Parametric approximation

- RBF:
  - Have to know centers and sigmas
- Projection pursuit:
  - Heuristic ways to choose centers and amplitudes
  - Tends to put centers displaced from true values
    - Because first center is a “’weighted average”
- Gaussian Mixture Model
  - Parametric approximation of a spatial grid

# Gaussian Mixture Model

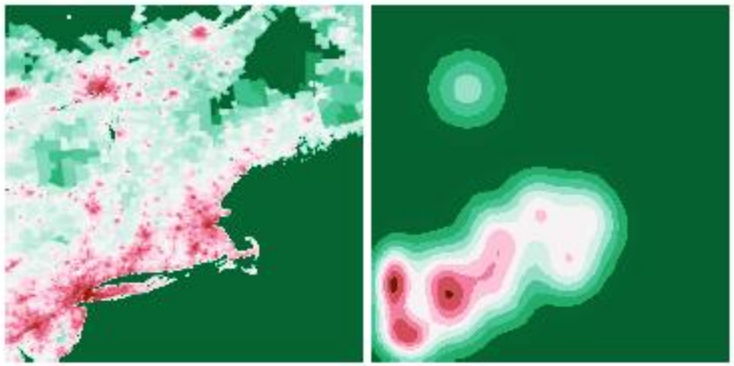


**Fig. 3.9** Approximating a spatial grid with a mixture of 10 Gaussians.

$$G(x,y) = \sum_{k=1}^{K} \pi_k f_k(x,y) \quad (3.5)$$

where the amplitudes $\pi_k$ are usually chosen so that they sum to 1. Each of the two-dimensional Gaussians, $f_k(x,y)$ is defined given the parameters $\mu_{x_k}$, $\mu_{y_k}$ and $\sum_{xy_k}$ as (dropping the subscript $k$ for convenience):

$$f(x,y) = \frac{1}{2\pi\sqrt{|\sum_{xy}|}} e^{-((x-\mu_x)(y-\mu_y))\sum_{xy}^{-1}((x-\mu_x)(y-\mu_y))^T/2} \quad (3.6)$$

ian and $\sum_{xy}$ the variance

$$\begin{pmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{pmatrix}$$

# GMM => Probability

- The GMM scaling has been chosen to add up to 1
  - Images will not sum up to one
  - But you can normalize the image so that it does
  - Then, approximate it with a GMM

- Procedure to approximate a GMM is expectation-minimization
  - What is EM?
  - Do you know any EM procedure?

# GMM E-step

- Assume an initial choice of parameters
- Then, compute the likelihood of the set of parameters (E step)

the likelihood of this given set of parameters. The probability that the pixel $x_i, y_i$ arose from the $k$th Gaussian component is given by:

$$P(k|x_i, y_i, \theta) = \frac{\pi_k f_k(x_i, y_i|\mu_{x_k}, \mu_{y_k}, \Sigma_{xy_k})}{P(x_i, y_i|\theta)} \tag{3.8}$$

where the probability (or *likelihood*) that the point $x_i, y_i$ is covered by the GMM given the set of parameters is given by:

$$P(x_i, y_i|\theta) = \sum_{k=1}^{K} \pi_k f_k(x_i, y_i|\mu_{x_k}, \mu_{y_k}, \sum_{xy_k}) \tag{3.9}$$

and $\theta$ is used as short-hand for all the parameters of all the $K$ components.

# GMM M-step

- Update the parameters of all K gaussians based on likelihood calculations

$$\mu_x = E(x) = \frac{\sum_{i=1}^{N}(P_k(x_i,y_i)x_i)}{\sum_{i=1}^{N}P_k(x_i,y_i)} \quad (3.10)$$

where $E(x)$ is the expected value of x (i.e., the mean value of $x$ in the grid). Similarly, $\mu_y$ is computed as $E(y)$ and $\sum_{xy}$ is computed as:

$$\begin{pmatrix} E((x-\mu_x)^2) & E((x-\mu_x)(y-\mu_y)) \\ E((x-\mu_x)(y-\mu_y)) & E((y-\mu_y)^2) \end{pmatrix} \quad (3.11)$$

Finally,the amplitude $\pi_k$ is computed as:

$$\pi_k = \frac{1}{N}\sum_{i=1}^{N}P_k(x_i,y_i) \quad (3.12)$$

- Now, go back to the E-step
  - This procedure converges after a few iterations

# GMM considerations: stopping

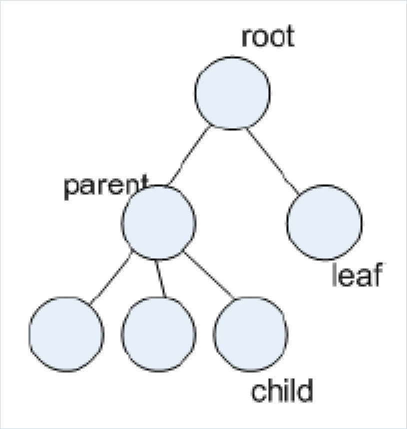
- How do we stop?
  - Look for improvement to be less than some threshold
    - Use GMM to approximate spatial grid
    - Computer error and stop when error falls below threshold
  - Problem: our threshold may be unrealistic for our K value
    - More common: look for percent improvement to be below threshold (from iteration to next iteration)
- Common to define improvement based on log(likelihood) as error measure

# GMM consideration: initializing

- How do we initialize GMM?
  - The number of Gaussians?
  - The initial means of the Gaussians?
  - Could start from level set and top N pixels as long as these are spaced reasonably far apart
    - Note that the GMM (unlike RBF) will move the centers
    - Not stuck with these centers

# Relational structure

- A tree is a hierarchical structure



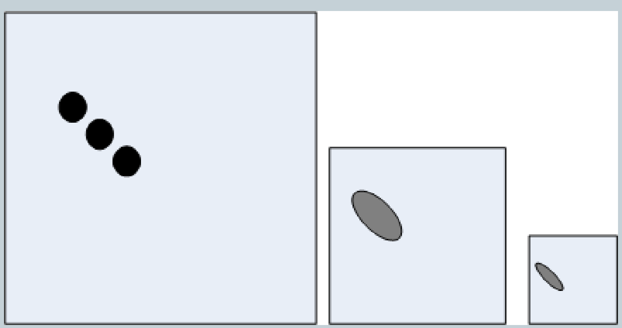

- What do the nodes represent (when processing spatial grids)?

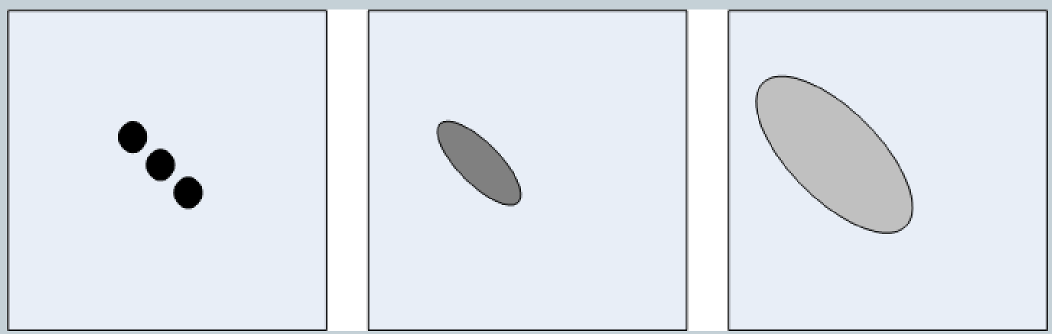
# Nodes in tree

- Nodes could represent:
  - Objects
    - Multi-scale approaches
    - Objects at different sizes
  - Coarser resolution spatial grids
    - Multi-resolution approaches

# Pyramidal Structure

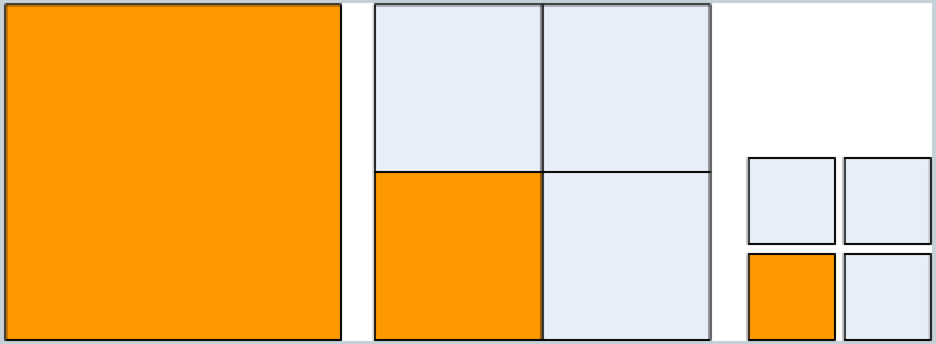
- Can decompose images into pyramidal structures
  - The structures remain the same size (in physical terms i.e. kms)
- We'll call this multi-resolution to distinguish it from multi-scale
  - In image processing literature, not all authors honor this distinction
  - In meteorological literature, not all authors understand the difference





# Wavelets and Multi-resolution

- Can be advantageous to be able to reconstruct an image from the pyramid decomposition of the image
  - Smoothing the image with Gaussians of different scales and resampling will not allow you to do this
  - Wavelets are special functions that give you this ability
    - Wavelets solve the multi-resolution reconstruction problem
      - Usually inappropriate to solve the multi-scale problem!



# Reading

- Parametric model to map obesity
  - Spatial variation using a Markov chain

M. Lahti-Koski, O. Taskinen, M. Simila, S. Mannisto, T. Laatikainen, P. Knekt, and L. Valsta. Mapping geographical variation in obesity in Finland. *Eur. J. of Public Health*, 18(6):637–643, 2008.

- Will discuss this paper in class next week