Region Segmentation

- ➢Binary Regions
- ➢k means clustering
- ➢Watershed Segmentation
- ➢Mean Shift Segmentation

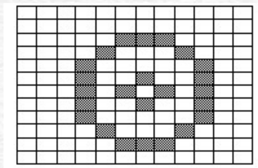
1

Segmentation

- Split images into smaller parts
  - Preferably corresponding to (parts of) objects.
- Two main approaches
  - Region based
  - Edge based
- Video segmentation
  - Breaking video into clips
  - Segmenting objects/regions consistently over time

2

Connectivity – Paradoxes



- Use pixel Adjacency to build contiguous regions
  - Objects
  - Background
  - Holes

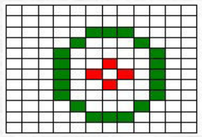
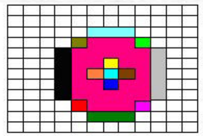
3

Connectivity – 4 adjacency & 8 adjacency

- Have to use either 4-adjacency or 8-adjacency
  - Label each non-zero pixel…

3	2	1
4	8	0
5	6	7

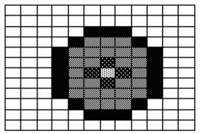
3	2	1
4	8	0
5	6	7

4

Connectivity –

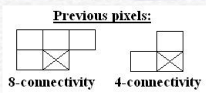
- What do we actually want?

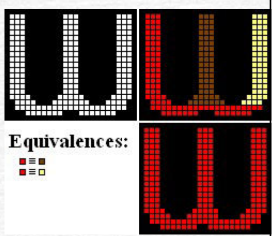


- One possibility
  - Treat background using 4-adjacency
  - Treat object using 8-adjacency
  - Treat holes using 4-adjacency
  - Treat objects in holes using 8-adjacency
  - …

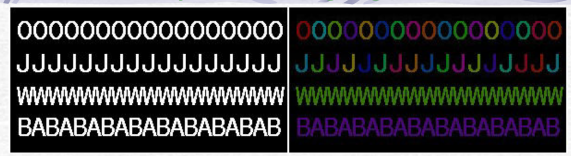
Connectivity – Connected Components Analysis

- Search image row by row
  - Label each non-zero pixel
  - If previous pixels are all background
    - Assign New Label
  - Otherwise
    - Pick any label from the previous pixels
    - If any of the other previous pixels have a different label
      - Note equivalence
- Relabel equivalent labels.



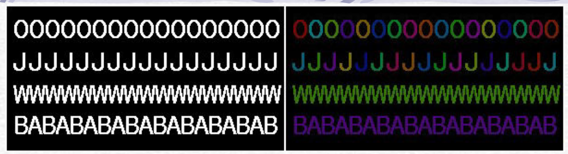


Connectivity – Extracting regions



```
vector<vector<Point>> contours;
vector<Vec4i> hierarchy;
findContours( binary_image, contours, hierarchy,
              CV_RETR_TREE, CV_CHAIN_APPROX_NONE );
```


Connectivity – Labelling regions



```
for (int contour=0; (contour < contours.size()); contour++)
 {
   Scalar colour( rand()&0xFF,rand()&0xFF,rand()&0xFF );
   drawContours( contours_image, contours, contour, colour,
                 CV_FILLED, 8, hierarchy );
 }
```

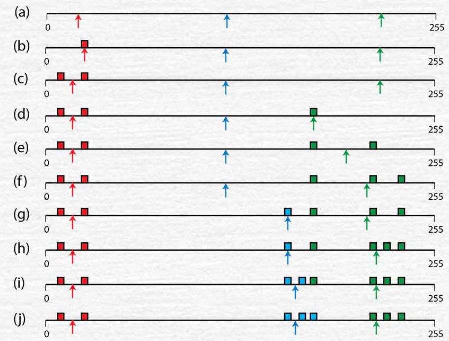
k means Clustering

- We'd like to
  - identify significant colours in images
    - Concise descriptions
    - Object tracking
  - reduce the number of colours in any image
    - Compression
- How do we find the best colours?
- k means clustering
  - Creates k clusters of pixels
  - Unsupervised learning

k means Clustering – Algorithm

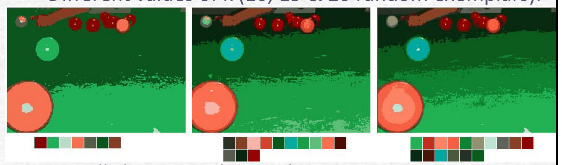
- Number of clusters (k) is known in advance (or determine the k with the maximum confidence)
- Initialise the k cluster exemplars either randomly or use the first k patterns or …
- 1st pass: Allocate patterns to the closest existing cluster exemplar and recompute the exemplar as the centre of gravity
- 2nd pass: Using the final exemplars from the first pass allocate all patterns cluster exemplars.

k means Clustering



k means Clustering

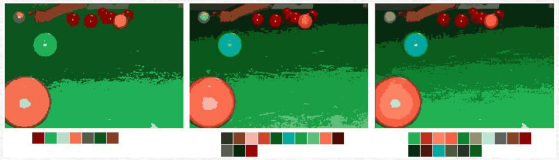
- Different values of k (10, 15 & 20 random exemplars):



- Not all clusters end up with patterns
- More exemplars generally gives a more faithful representation

k means Clustering

- Choosing the best number of exemplars
  - Evaluate the resulting clusters

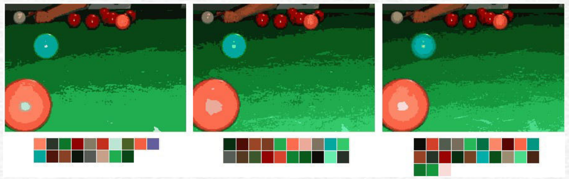


- Davies-Bouldin index measures cluster separation:
  - $DB = 1/k\ \Sigma_{1..k}\ \max_{i \neq j}\ ((\Delta_i + \Delta_j)/\ \delta_{i,j})$
- OR Check that distributions are normal

13

k means Clustering

- Using random exemplars gives non-deterministic results (30 random exemplars each time):



14

k means Clustering

```
// Store the image pixels as an array of samples
Mat samples(image.rows*image.cols, 3, CV_32F);
float* sample = samples.ptr<float>(0);
for(int row=0; row<image.rows; row++)
  for(int col=0; col<image.cols; col++)
    for (int channel=0; channel < 3; channel++)
         samples.at<float>(row*image.cols+col,channel) =
                    (uchar) image.at<Vec3b>(row,col)[channel];
// Apply k-means clustering, determining the cluster
// centres and a label for each pixel.
   ....
```


15

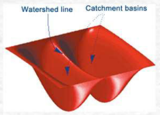
k means Clustering

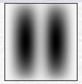
```
Mat labels, centres;
kmeans(samples, k, labels, TermCriteria( CV_TERMCRIT_ITER|
          CV_TERMCRIT_EPS, 0.0001, 10000), iterations,
          KMEANS_PP_CENTERS, centres );
// Use centres and label to populate result image
Mat& result_image = Mat( image.size(), image.type() );
for(int row=0; row<image.rows; row++)
  for(int col=0; col<image.cols; col++)
    for (int channel=0; channel < 3; channel++)
      result_image.at<Vec3b>(row,col)[channel] =
            (uchar) centres.at<float>( *(labels.ptr<int>(
                        row*image.cols+col )), channel);
```


16

Watershed segmentation

- In geology watersheds separate different catchment basins.
  - Where the rain gets caught
- In computer vision we
  - Identify all minima.
  - Label as different regions.
  - Flood from the minima extending the regions.
  - Where regions meet we have watershed lines.
- Minimum what?
  - Greyscale
  - Gradient
  - Inverse of the chamfer distance

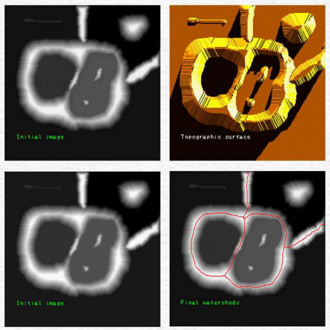




Images from https://uk.mathworks.com/company/newsletters/articles/the-watershed-transform-strategies-for-image-segmentation.html

17

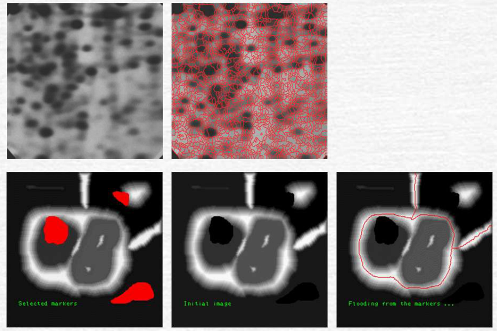
Watershed segmentation example



Images from http://cmm.ensmp.fr/~beucher/wtshed.html Reproduced with permission.

18

Watershed segmentation with markers



Images from http://cmm.ensmp.fr/~beucher/wtshed.html Reproduced with permission.

19

Watershed segmentation with markers

- Generally we get too many regions
  - Use a priori labels to identify ‘objects’ and expand from these rather than the minima.

20

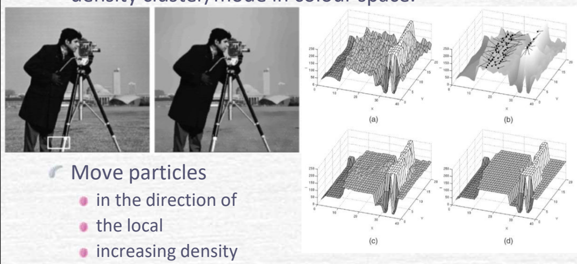
Mean Shift Segmentation

- k-means clustering
  - Requires the number of clusters to be known
  - Takes no account of spatial location
- Mean Shift Segmentation…
  - Do not need to know the number of clusters
  - Can provide spatial as well as colour segmentation
  - Developed by Comaniciu & Meer 2002

Image from "Mean Shift: A Robust Approach Toward Feature Space Analysis" Dorin Comaniciu,, and Peter Meer , IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 24, NO. 5, MAY 2002;

Mean Shift for segmentation– goal

- To associate each point with a particular high-density cluster/mode in colour space.



- Move particles
  - in the direction of
  - the local
  - increasing density

Images from "Mean Shift: A Robust Approach Toward Feature Space Analysis" Dorin Comaniciu,, and Peter Meer , IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 24, NO. 5, MAY 2002;

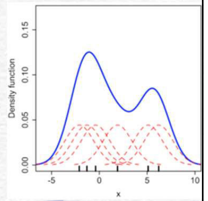
Kernel Density Estimation

- The Problem: Given a sparse dataset determine an estimate of density at each point.

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

  - Kernel Density K() function n samples x_i
  - h is the bandwidth
- Effectively
  - Smooth all data samples
  - Add them all together
- If the dataset is multidimensional:

$$\hat{f}(\mathbf{x}) = \frac{1}{nh^d}\sum_{i=1}^{n} K\Big(\frac{\mathbf{x}-\mathbf{x}_i}{h}\Big).$$



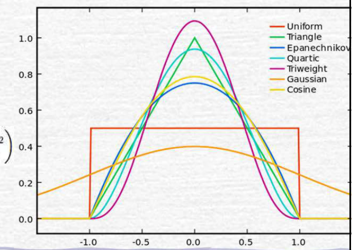
Kernel Density Estimation – Kernels

- Many different kernels.
- Kernel functions must integrate to 1.
- Typically use either
  - Uniform

$$k_E(x) = \begin{cases} 1-x & 0 \le x \le 1 \\ 0 & x > 1, \end{cases}$$

  - Gaussian

$$K_N(\mathbf{x}) = (2\pi)^{-d/2}\exp\Big(-\frac{1}{2}\|\mathbf{x}\|^2\Big)$$



Mean Shift – Basic Algorithm

- For each particle (pixel)
  - Estimate the local kernel density and more importantly the direction of local increasing density (the *mean shift vector)*
  - Shift the particle to the new mean.
  - Re-compute until the location stabilizes.
- Finally identify which pixels ended up in the same location
  - Mark these as members of the same cluster.
  - Determine the local mean of similar particles

Mean Shift – The local kernel density

- We limit the points included in the kernel density estimate based on distance and on similar to the current point.

$$K_{h_s,h_r}(\mathbf{x}) = \frac{C}{h_s^2 h_r^p} k\left(\left\|\frac{\mathbf{x}^s}{h_s}\right\|^2\right) k\left(\left\|\frac{\mathbf{x}^r}{h_r}\right\|^2\right)$$

  - We must use both a spatial kernel and a colour kernel.
  - Both can be Gaussian
  - Spatial kernel limits/weights the region to consider around the current point
  - Colour/range kernel limits/weights the colour/intensity of the points to be included in the mean.

Maths from "Mean Shift: A Robust Approach Toward Feature Space Analysis" Dorin Comaniciu,, and Peter Meer , IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 24, NO. 5, MAY 2002;

Mean Shift – The mean shift vector

- For radially symmetric kernels $K(\mathbf{x}) = c_{k,d} k(\|\mathbf{x}\|^2)$
- So the KDE becomes $\hat{f}_{h,K}(\mathbf{x}) = \frac{c_{k,d}}{nh^d} \sum_{i=1}^{n} k\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)$ as $\hat{f}(\mathbf{x}) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)$
- We are interested in the rate of change…

$$\hat{\nabla} f_{h,K}(\mathbf{x}) \equiv \nabla \hat{f}_{h,K}(\mathbf{x}) = \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^{n} (\mathbf{x}-\mathbf{x}_i) k'\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)$$

- Set $g(x) = -k'(x)$ so

$$= \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^{n} (\mathbf{x}_i-\mathbf{x}) g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)$$

$$= \frac{2c_{k,d}}{nh^{d+2}} \left[\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)\right] \left[\frac{\sum_{i=1}^{n} \mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)} - \mathbf{x}\right]$$

- Define the mean shift vector as $\mathbf{m}_{h,G}(\mathbf{x}) = \frac{\sum_{i=1}^{n} \mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)} - \mathbf{x}$

Maths from "Mean Shift: A Robust Approach Toward Feature Space Analysis" Dorin Comaniciu,, and Peter Meer , IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 24, NO. 5, MAY 2002;

Mean Shift – Spatial kernel functions

Varying the Spatial radius and varying the colour radius

Mean Shift – Pros and cons

- + Do not need to know the number of clusters *a priori*.
- + Provides spatial as well as colour segmentation.
- - Selection of kernel widths can be very hard.
- - It is quite slow particularly if there are a lot of clusters.

Images from "Mean Shift: A Robust Approach Toward Feature Space Analysis" Dorin Comaniciu,, and Peter Meer , IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 24, NO. 5, MAY 2002; From http://www.comaniciu.net/Papers/MsRobustApproach.pdf

29