IMAGE DATA ANALYSIS (6CFU)

MODULE OF

REMOTE SENSING

(9 CFU)

A.Y. 2022/23

MASTER OF SCIENCE IN COMMUNICATION TECHNOLOGIES AND MULTIMEDIA

MASTER OF SCIENCE IN COMPUTER SCIENCE, LM INGEGNERIA INFORMATICA

PROF. ALBERTO SIGNORONI

CLUSTERING AND UNSUPERVISED CLASSIFICATION

Back to the general problem of delineation of Spectral Classes

- The successful application of ***maximum likelihood classification*** (and other supervised classification methods) is dependent upon delineating correctly the ***spectral classes*** in the image data of interest.
  - This is necessary since each class is to be modeled by a *normal probability distribution*, or to be otherwise identified based upon training data, as discussed for supervised classification.
  - If a class happens to be *multimodal*, and this is not resolved, then clearly the modeling cannot be very effective.
  - ***Users*** of remotely sensed data can only specify the ***information classes***.
- Occasionally it might be possible to guess the number of *spectral classes* in a particular *information class* but, in general, ***the user would have little idea of the number of distinct unimodal groups*** that the data falls into in multispectral space.
  - *Gaussian mixture modeling* can be used for this purpose but the *complexity* of estimating simultaneously the number of Gaussian components, and their parameters, can make this approach difficult to use.

1. ***Clustering*** procedures are *practical alternatives* that can be used for that purpose; these are methods that have been applied in many data analysis fields to ***enable inherent data structures to be determined***.

2. ***Clustering can also be used for unsupervised classification (and data mining)***.
   - In this technique *an image is segmented into* ***unknown*** *classes*. It is the ***task of the user to label those classes afterwards***.

- There are a great number of clustering methods. In this lesson only those commonly employed in the fields of our interest are treated.

SIMILARITY METRICS AND CLUSTERING CRITERIA

How many clusters?

How many clusters?

Six Clusters

Two Clusters

Four Clusters

Similarity Metrics and Clustering Criteria

- Clustering implies a grouping of pixels in multidimensional (multispectral) space.
  - Pixels belonging to a particular cluster are therefore *spectrally similar*.
- In order to quantify this relationship, it is necessary to devise a **similarity measure**.
  - *Many similarity metrics* have been proposed but *those used commonly in clustering* procedures are usually ***simple distance measures in multispectral space***.
  - The *most frequently encountered* are Euclidean distance and L_1 (or interpoint) distance.
  - If $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are two pixels whose similarity is to be checked then the **Euclidean distance** between them is

$$\begin{aligned} \mathrm{d}(\boldsymbol{x}_1, \boldsymbol{x}_2) &= ||\boldsymbol{x}_1 - \boldsymbol{x}_2|| \\ &= \{(\boldsymbol{x}_1 - \boldsymbol{x}_2)^t (\boldsymbol{x}_1 - \boldsymbol{x}_2)\}^{\frac{1}{2}} \\ &= \left\{ \sum_{i=1}^{N} (x_{1_i} - x_{2_i})^2 \right\}^{\frac{1}{2}} \end{aligned} \tag{9.1}$$

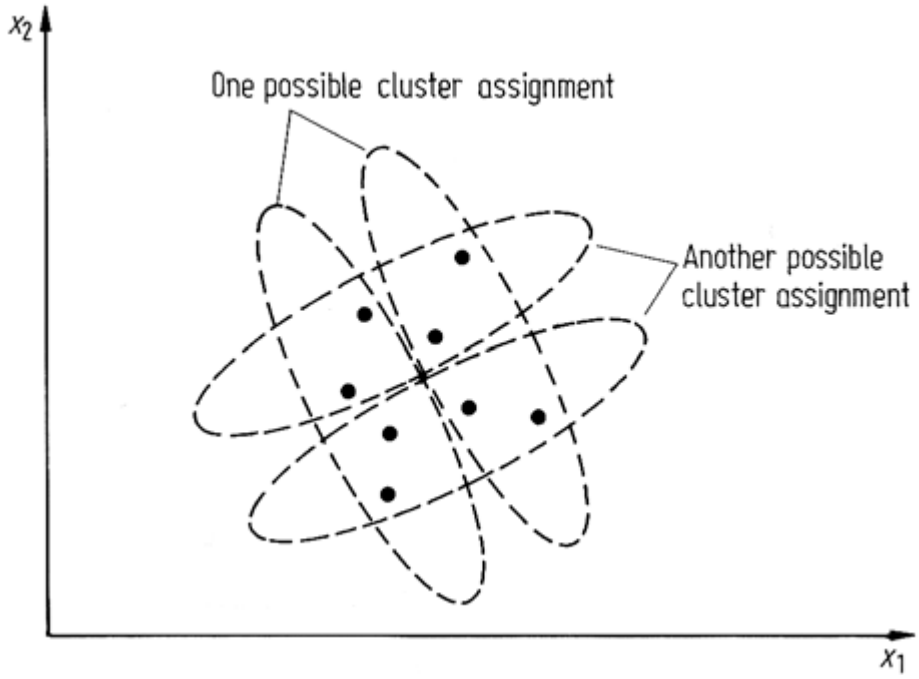
    where N is the number of spectral components.
  - The ***L_1* distance** between the pixels is

$$\mathrm{d}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sum_{i=1}^{N} |x_{1_i} - x_{2_i}|. \tag{9.2}$$

  - Clearly *the latter is computationally faster* to determine. However, it can be seen as less accurate than the Euclidean distance measure.

Similarity Metrics and Clustering Criteria

- By using a distance measure, it should be possible to determine clusters in data.
- Often however **there could be several acceptable clusters assignments** of the data, as depicted in Figure, so that
  - once a candidate clustering has been found it is desirable to have a means by which the "quality" of clustering can be measured.
  - The availability of such a measure should allow one cluster assignment of the data to be chosen over all others.



Similarity Metrics and Clustering Criteria

- A common clustering criterion or *quality indicator* is the **sum of squared error** (**SSE**) measure, defined as

$$\text{SSE} = \sum_{C_i} \sum_{x \in C_i} (x - m_i)^t (x - m_i)$$

$$= \sum_{C_i} \sum_{x \in C_i} ||x - m_i||^2 \qquad (9.3)$$

  where m_i is the mean of the ith cluster and $x \in C_i$ is a pattern assigned to that cluster.
  - The outer sum is over all the clusters.
- This measure computes the *cumulative distance* of each pattern from its cluster centre for each cluster individually, and then sums those measures over all the clusters.
  - *If SSE is small, the distances from patterns to cluster means are all small and the clustering would be regarded favourably.*
- *Other* quality of clustering measures exist.
  - One popular one is to derive a "within cluster scatter measure" by determining the average covariance matrix of the clusters, and a "between cluster scatter measure" by looking at the means of the clusters compared with the global mean of the data.
    - These two measures are combined into a single figure of merit as discussed in Duda, Hart and Stork (2001) and Coleman and Andrews (1979). It can be shown that figures of merit such as these *are similar* to the sum of squared error criterion.

Similarity Metrics and Clustering Criteria

- It is of interest to note that SSE has a theoretical minimum of zero, which corresponds to all clusters containing only a single data point.
  - As a result, if an iterative method is used to seek the natural clusters or spectral classes in a set of data then it has a guaranteed termination point, at least in principle.
  - In practice, it may be too *expensive and meaningless* to allow natural termination.
  - Instead, iterative procedures are often stopped when an acceptable degree of clustering has been achieved.
- It is possible now to consider the implementation of an actual clustering algorithm.
  - While it should depend upon a progressive minimisation (and thus calculation) of SSE this is impracticable since it requires an enormous number of values of SSE for the evaluation of all candidate clusterings.
  - For example, there are approximately $C^P/C!$ ways of placing P patterns into C clusters (Duda, Hart and Stork, 2001).
  - This number of SSE values would require computation at each stage of clustering to allow a minimum to be chosen.
- Rather than embark upon such a rigorous and computationally expensive approach the **heuristic procedures**, as described in the following, are usually adopted in practice.
- Similarity metrics for clustering can incorporate measures other than spectral likeness.
  - **Spatial proximity** might be important in some applications as might components that account for categorical information.
  - For example, clustering crop pixels might be guided by all of multispectral measurements, soil type and spatial contiguity.
  - These more general metrics are not covered here.

THE ITERATIVE OPTIMIZATION (MIGRATING MEANS, K-MEANS) CLUSTERING ALGORITHM

The Basic Algorithm

- The **iterative optimization algorithm** (also known as **k-means**) is implemented by the following set of basic steps:

1. The procedure is initialised by selecting C points in multispectral space to serve as candidate cluster centres. Let these be called

$$\hat{\boldsymbol{m}}_i, i = 1, \ldots C.$$

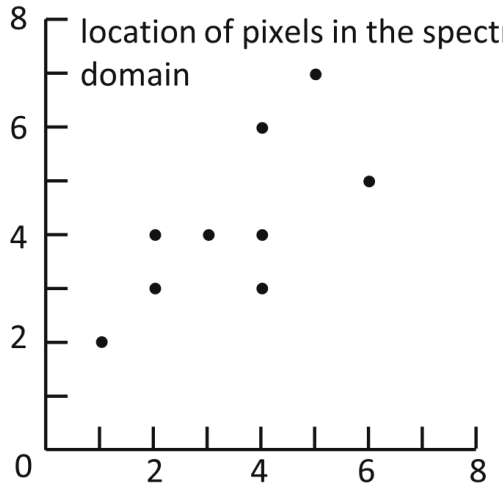
   The selection of the $\hat{\boldsymbol{m}}_i$ at this stage is arbitrary with the exception that no two may be the same. To avoid anomolous cluster generation with unusual data sets it is generally wise to space the initial cluster means uniformly over the data. This can also serve to enhance convergence.
   Besides choosing the $\hat{\boldsymbol{m}}_i$ the number of clusters C, must be specified beforehand by the user.
2. The location $\boldsymbol{x}$ of each pixel in the segment of the image to be clustered is examined and the pixel is assigned to the nearest candidate cluster. This assignment would be made on the basis of the Euclidean or even $L1$ distance measure.
3. The new set of means that result from the grouping produced in Step 2 are computed. Let these be denoted

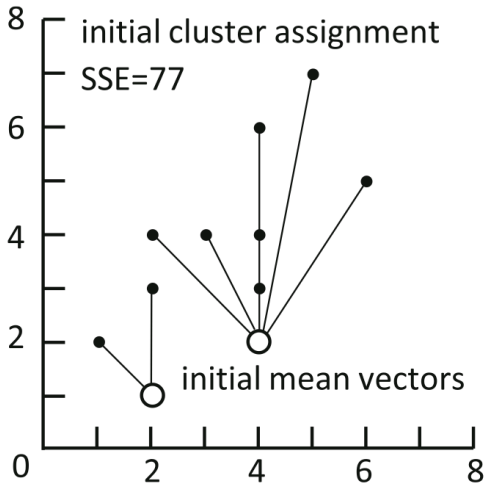
$$\boldsymbol{m}_i, i = 1, \ldots C.$$

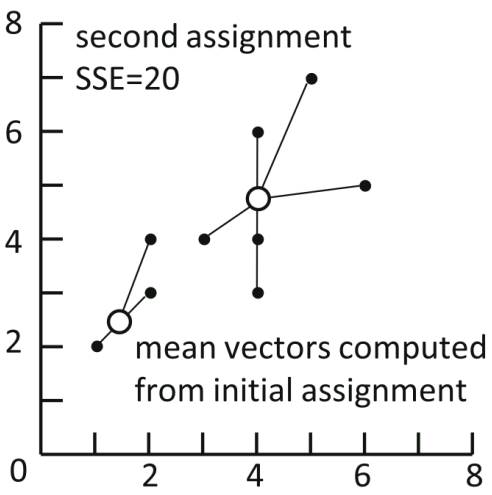
4. If $\boldsymbol{m}_i = \hat{\boldsymbol{m}}_i$ for all i, the procedure is terminated. Otherwise $\hat{\boldsymbol{m}}_i$ is redefined as the current value of $\boldsymbol{m}_i$ and the procedure returns to Step 2.

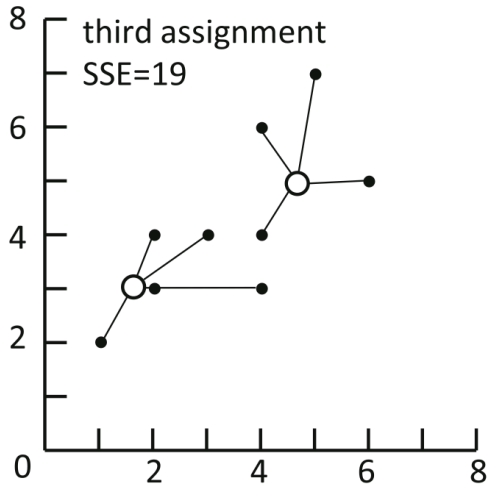
The Basic Algorithm

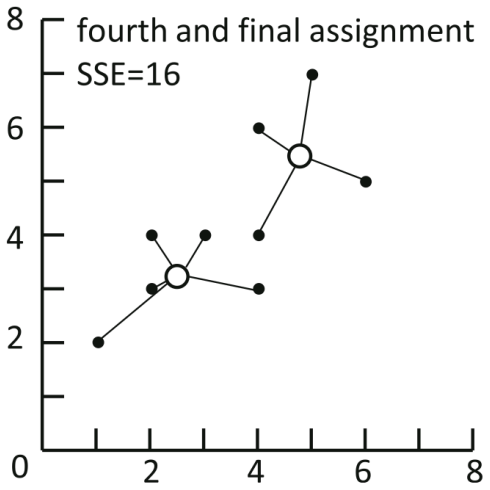
- The iterative optimization procedure is *illustrated for a simple set of two dimensional patterns* in Figure.

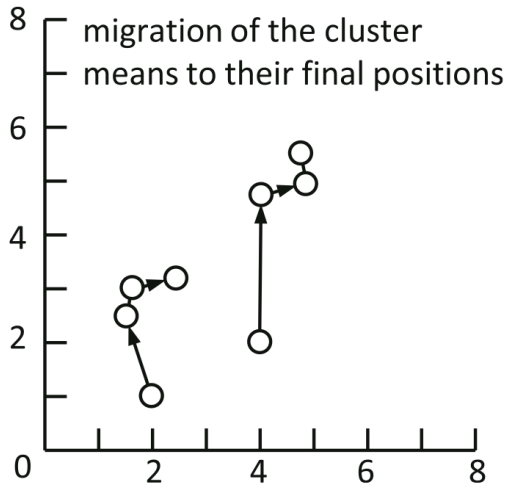












Isodata - Mergings, Deletions and Splitting Elongated Clusters

- Once clustering is completed, or at any suitable intervening stage, the *clusters can be examined* to see whether
  i. any clusters contain so few points as to be meaningless (e.g. that they would not give acceptable statistics estimates if used in training a maximum likelihood classifier), or
  ii. some clusters are so close together that they represent an unnecessary or indeed an injudicious division of the data, and thus they should be merged.
  - In view of what already seen for the number of training pixels required for each class, a guideline exists for (i.), namely that a cluster would be of little value for training a maximum likelihood classifier if it did not contain about $10N$ points where N is the number of spectral components.
  - More advanced means for deciding whether clusters should be merged can also be devised (feature reduction and separability techniques).

- Another stage that can be inserted into the clustering algorithm is to separate elongated clusters into two new clusters.
  - Usually this is done by prespecifying a standard deviation in each spectral band beyond which a cluster should be halved.
  - Again this can be done after a set number of iterations, also specified by the user.

- The combination of k-means with the above cluster operations lead to the so called ***ISODATA*** **(**Iterative Self-Organizing data analysis**) algorithm**.

Choice of Initial Cluster Centres

- Initialisation of the k-means iterative optimization procedure requires specification of the number of clusters expected, along with their starting positions.

- In practice the actual or optimum **number** of clusters to choose will not be known.
  - Therefore it is often chosen conservatively high, having in mind that resulting inseparable clusters can be consolidated
    - after the process is completed,
    - or at intervening iterations, if a merging operation is available.

- The choice of the **initial locations** of the cluster centres is *not critical* although evidently it will have an influence on the time it takes to reach a final, acceptable clustering.
  - Since no guidance is available in general, the following is a logical procedure (Phillips 1973).
    - The initial cluster centres are chosen uniformly spaced along the *multidimensional diagonal* of the multispectral pixel space.
      - This is a line from the origin to the point corresponding to the maximum brightness value in each spectral component (corresponding to 255 for 8 bit data, etc.).
    - This choice *can be refined if the user has some idea* of the actual range of brightness values in each spectral component, say by having previously computed histograms.
      - In that case the cluster centres would be initialised along a diagonal through the *actual multidimensional extremities* of the data.
  - Choice of the initial locations of clusters in the manner described is a reasonable and effective one since they are then *well spread over the multispectral space in a region in which many spectral classes occur, especially for correlated data* such as that corresponding to soils, rocks, concretes, etc.

Clustering Cost

- Obviously the major limitation of the k-means and isodata techniques is the need to prespecify the number of cluster centres.
  - If this specification is *too high* then *a posteriori merging* can be used; however this is an expensive strategy.
  - On the other hand, if *too few* are chosen initially then some *multimodal spectral classes* will result which, in turn, will prejudice ultimate classification accuracy.
- Irrespective of whether too many or too few clusters are used, **the k-means and ISODATA approaches are computationally expensive** since, at each iteration, every pixel must be checked against all cluster centres.
  - Thus for C clusters and P pixels, $P \cdot C$ distances have to be computed at each iteration and the smallest found.
  - For N band data, each Euclidean distance calculation will require N multiplications and N additions, ignoring the square root operation in (9.1) since that need not be carried out.
  - Thus for 20 classes and 10^6 pixels, 100 iterations isodata clustering requires 2×10^9 multiplications per band of data!
  - To decrease computational complexity random data sampling can be accomplished
    - similar role with respect to the training pixels for supervised classification
    - e.g. factor 100 subsampling leads to 20×10^6 the number of multiplication for band of data.

UNSUPERVISED CLASSIFICATION AND CLUSTER MAPS - EXAMPLES

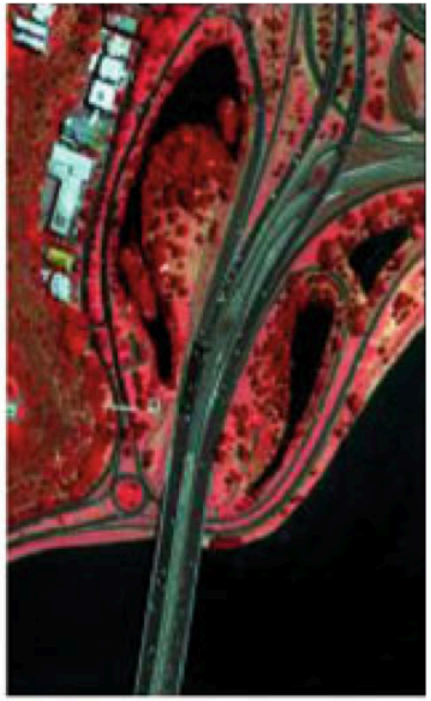
Unsupervised Classification and Cluster Maps

- At the completion of clustering, pixels within a given group *are usually given a color (or symbol) label* to indicate that they belong to the same cluster or spectral class.
- **Using these labels a cluster map can be produced**; this is a map corresponding to the image which has been clustered, *but in which the pixels are represented by their label rather than by the original (possibly multispectral) data*.
  - If, for computational reasons, only part of an image is used to form the cluster centres, all pixels can be allocated to one of the clusters through a minimum distance assignment.
  - The availability of a cluster map allows a classification to be made.
  - If some pixels with a given label can be identified with a particular ground cover type (by means of maps, site visits or other forms of reference data) then all pixels with the same label can be associated with that class.
- This method of image classification, depending as it does on *a posteriori recognition of the classes, is called* ***unsupervised classification*** since the analyst plays no part until the computational aspects are complete.
  - *Often unsupervised classification is used as a stand-alone technique*, particularly when reliable training data for supervised classification cannot be obtained or is too expensive to acquire.
- However, it is also of value, as noted earlier, to determine the spectral classes that should be considered in a subsequent supervised approach. This is pursued in some **hybrid unsupervised/supervised methodologies**.

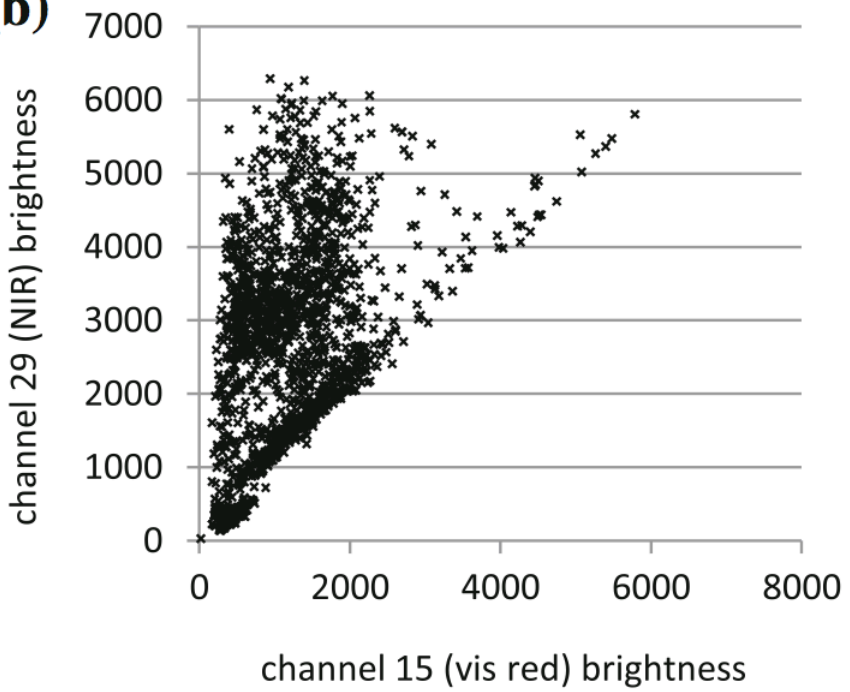
Clustering Example

- To illustrate the nature of the results produced by the k-means algorithm consider the segment of *HyMap* (hyperspectral) imagery
  - The data Fig.(**a**) is about a highway interchange near the city of Perth in Western Australia. It was recorded in January 2010 and consists of vegetation, roadway pavements, water and bare and semi-bare areas (see http://www.hyvista.com/technology/sensors/hymap/)
  - Only 5 channels (out of 128) was used for the clustering: Fig.(**b**) shows a scatter diagram for the image in which a near infrared channel (29) is plotted against a visible red channel (15).
  - See Table 1 for the full space of channels considered in this example.

(a)

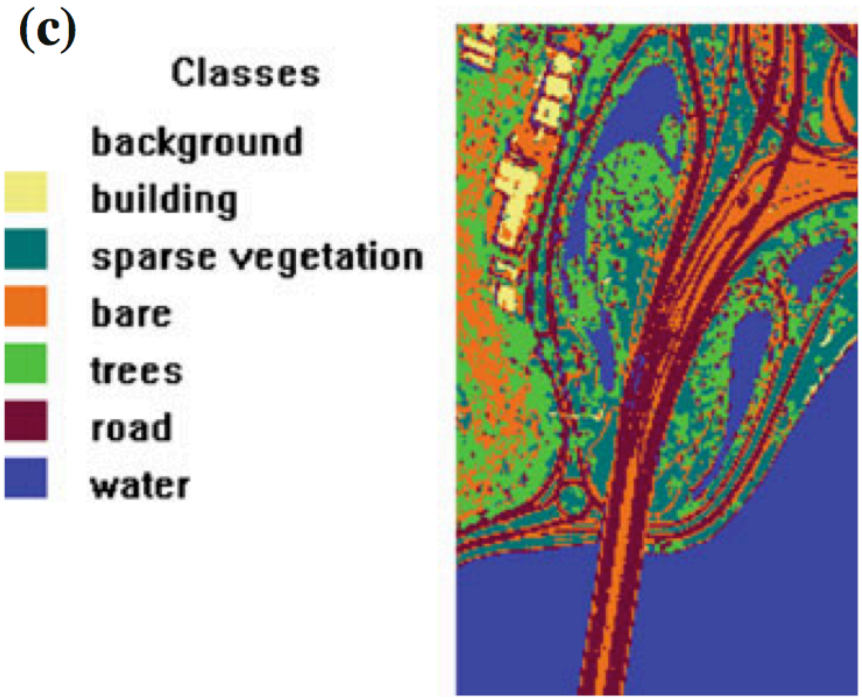


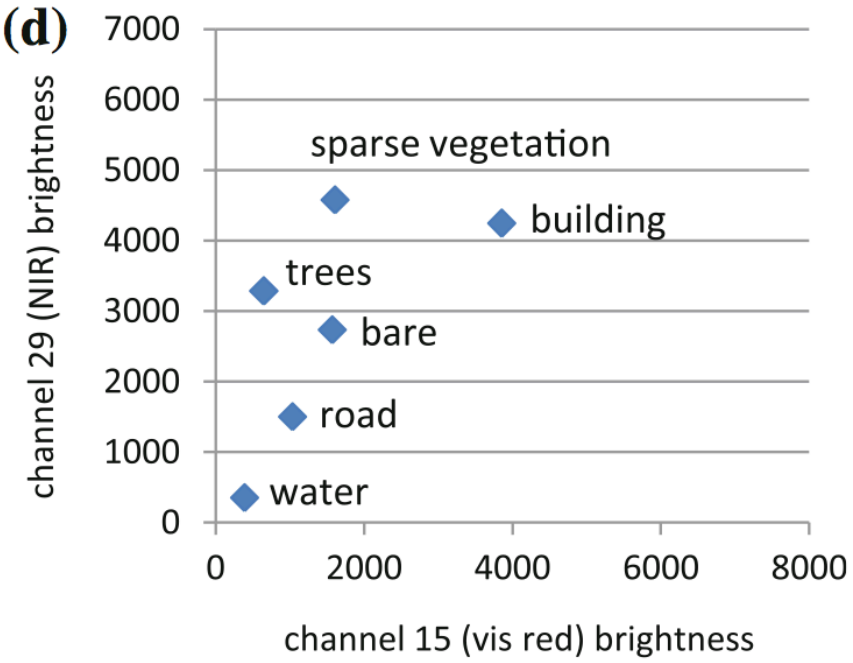
(b)



Clustering Example

- The data was clustered using the k-means (Isodata) procedure available in *MultiSpec*© (try it! https://engineering.purdue.edu/~biehl/MultiSpec/index.html)
  - The algorithm was asked to determine six clusters, since a visual inspection of the image showed that to be reasonable. No merging and splitting options were employed, but any clusters with fewer than 125 pixels at the end of the process were eliminated.
  - The results shown in Fig.(**c**) were generated after 8 iterations. The cluster means are plotted in just two dimensions in Fig.(**d**)
  - See Table 2 for a full description of the obtained clusters' means.





Clustering Example

- Table 1: HyMap channels used in the k-means clustering example

Channel	Band centre (nm)	Band width (nm)
7 (visible green)	511.3	17.6
15 (visible red)	634.0	16.4
29 (near infrared)	846.7	16.3
80 (middle infrared)	1616.9	14.8
108 (middle infrared)	2152.7	30.2

- Table 2: Cluster centres (means) for the k-means (Isodata) exercise in Figure

		Cluster mean vectors (on 16 bit scale)				
Cluster	Label	Channel 7	Channel 15	Channel 29	Channel 80	Channel 108
1	Building	3511.9	3855.7	4243.7	4944.2	4931.6
2	Sparse veg	1509.6	1609.3	4579.5	3641.7	2267.0
3	Bare	1333.9	1570.7	2734.3	2715.1	2058.7
4	Trees	725.6	650.6	3282.4	1676.2	866.6
5	Road	952.3	1037.1	1503.7	1438.5	1202.3
6	Water	479.2	391.1	354.8	231.0	171.6

A SINGLE PASS CLUSTERING TECHNIQUE

Single Pass Algorithm

- **In order to reduce the cost of clustering image data**, alternatives to iterative optimization have been proposed and are widely implemented in software packages for remote sensing image analysis.
  - Often what they gain in speed they may lose in accuracy; however, if the user is aware of their characteristics they can usually be employed effectively.
  - One fast clustering procedure which requires only a single pass through the data is described in the following.

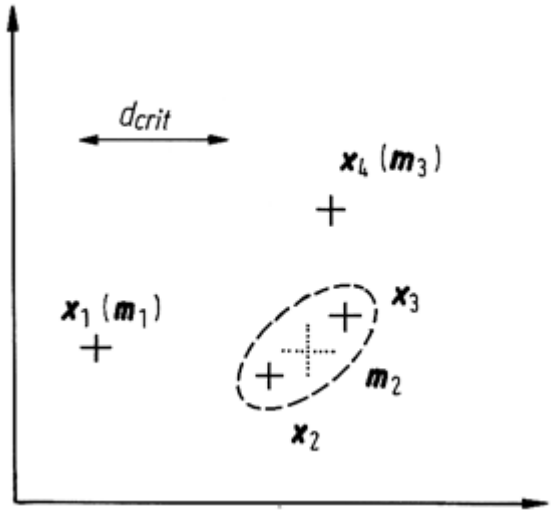
1. Not all the region to be clustered must be used in developing cluster centres but rather, for cost reduction, **a randomly selected sample** may be chosen, and the samples **arranged into a two-dimensional array**.
   - The first row of samples is then used to obtain a starting set of cluster centres.
   - This is initiated by adopting the first sample as the centre of the first cluster.
   - If the second sample in the first row is further away from the first than a user specified ***critical distance,*** then it is used to form another cluster centre.
   - Otherwise, the two samples are said to belong to the same cluster and their mean is computed as the new cluster centre.

Single Pass Algorithm

- This procedure, which is illustrated in Figure, is applied to all samples in the first row.
- Once this row has been exhausted the multidimensional standard deviations of the clusters are computed.

2. Each sample in the second and subsequent rows is checked to see which cluster it is closest to.

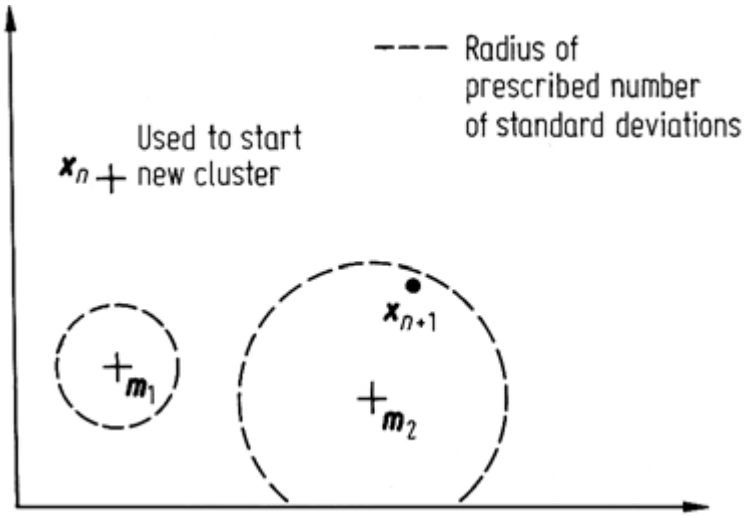
- It is assigned to that cluster, and the cluster statistics recomputed, if it lies within a user-prescribed ***number of standard deviations***.



Single Pass Algorithm

- Otherwise, it is used to form a new cluster centre (which is assigned a nominal standard deviation). This is depicted in Figure.

- In this manner all the samples are clustered and clusters with less than a prescribed number of pixels are deleted.
  - Should a **cluster map** be required then the original segment of image data is scanned pixel by pixel and each pixel labelled according to the class it is closest to (on the basis usually of Euclidean distance).
  - Should it be an **outlying** pixel in terms of the available cluster centres it is **not** labelled.



Advantages and Limitations

- Apart from speed, a major advantage of this approach over the k-means procedure is its **ability to create cluster centres as it proceeds**.
  - It is therefore not necessary for the user to specify beforehand the required number of clusters.
- However, the method has **two limitations**.
  - First, the user has to have a feel for the parameters required by the algorithm.
    - In particular the user has to specify the ***critical distance parameter*** sensibly to enable the initial cluster centres to be established in a reasonable manner.
    - Also, the user has to know ***how many standard deviations*** should be used in assigning pixels in the second and subsequent lines of samples to existing clusters.
    - Clearly, with experience, these parameters can be estimated reasonably.
  - The second limitation is that the method is dependent upon the first line of samples to initiate the clustering.
    - Since it is only a one pass algorithm and has no feedback checking mechanism by way of iteration, its ultimate set of cluster centres can depend significantly on the character of the first line of samples.

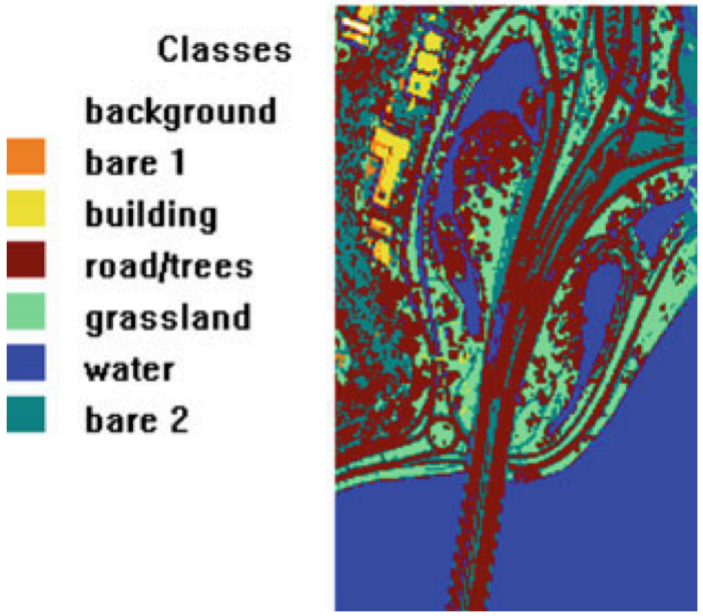
Strip Generation Parameter

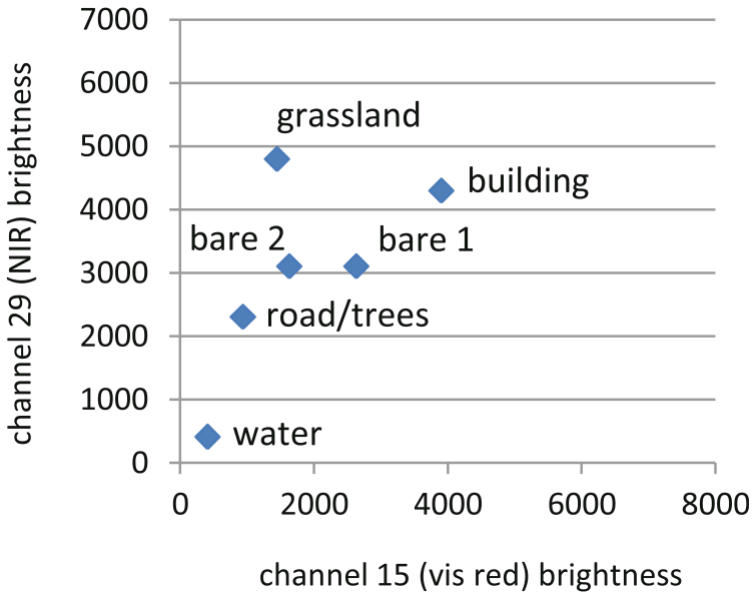
- Adjacent pixels along an image line frequently belong to the same cluster, as is to be expected, particularly for images of cultivated regions.
  - A method therefore for enhancing the speed of clustering is to compare a pixel with its predecessor and assign it to the same cluster *immediately* if it is similar.
  - The *similarity check* often used is quite straightforward, consisting of a check of the brightness difference in each spectral band.
  - The *difference allowable* for two pixels to be considered part of the same cluster is called the ***strip generation parameter***.

Variations on the Single Pass Algorithm

- The technique outlined in the preceding section has a number of ***variations***.
  - For example, the *initial cluster centres can be specified by the user* or alternatively can be created *from the data* using a critical distance parameter as illustrated in a previous Figure.
  - Moreover, rather than use a multiplier of standard deviation for assigning pixels from the second and subsequent rows of samples, *some algorithms proceed exactly as for the first row*, with standard deviation information not used at all.
  - Some algorithms use the L_1 metric of (9.2), rather than Euclidean distance, and some check *inter-cluster distances* and merge if this is indicated; periodically small clusters can also be eliminated.
  - The package known as ***MultiSpec***, also uses just critical distance parameters over the full range, although the user can specify a different critical distance for the second and later rows of samples (Landgrebe and Biehl, 2004).

Example





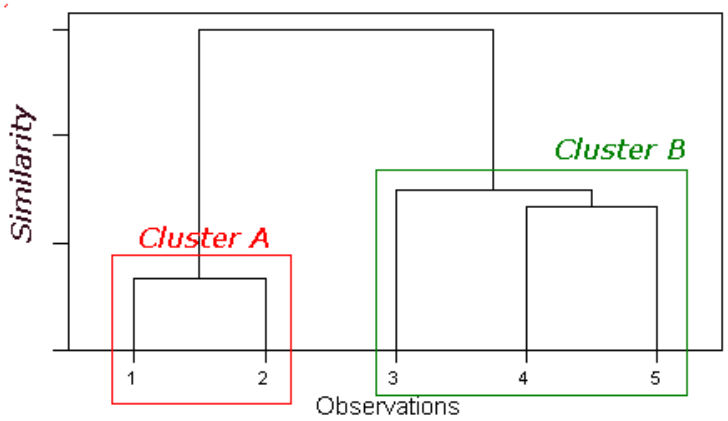
- Note that the clusters are (unavoidably) different from those computed with k-means, therefore a slightly different class naming has been adopted.

		Cluster mean vectors (on 16 bit scale)				
Cluster	Label	Channel 7	Channel 15	Channel 29	Channel 80	Channel 108
1	Bare 1	2309.7	2632.9	3106.1	3713.2	3663.4
2	Building	3585.8	3901.5	4300.5	4880.7	4870.2
3	Road/trees	900.4	940.0	2307.6	1640.2	1143.4
4	Grassland	1441.3	1447.2	4798.6	3455.6	2028.6
5	Water	490.4	408.2	409.0	274.9	207.5
6	Bare 2	1372.7	1630.5	3105.7	3033.3	2214.8

AGGLOMERATIVE HIERARCHICAL CLUSTERING

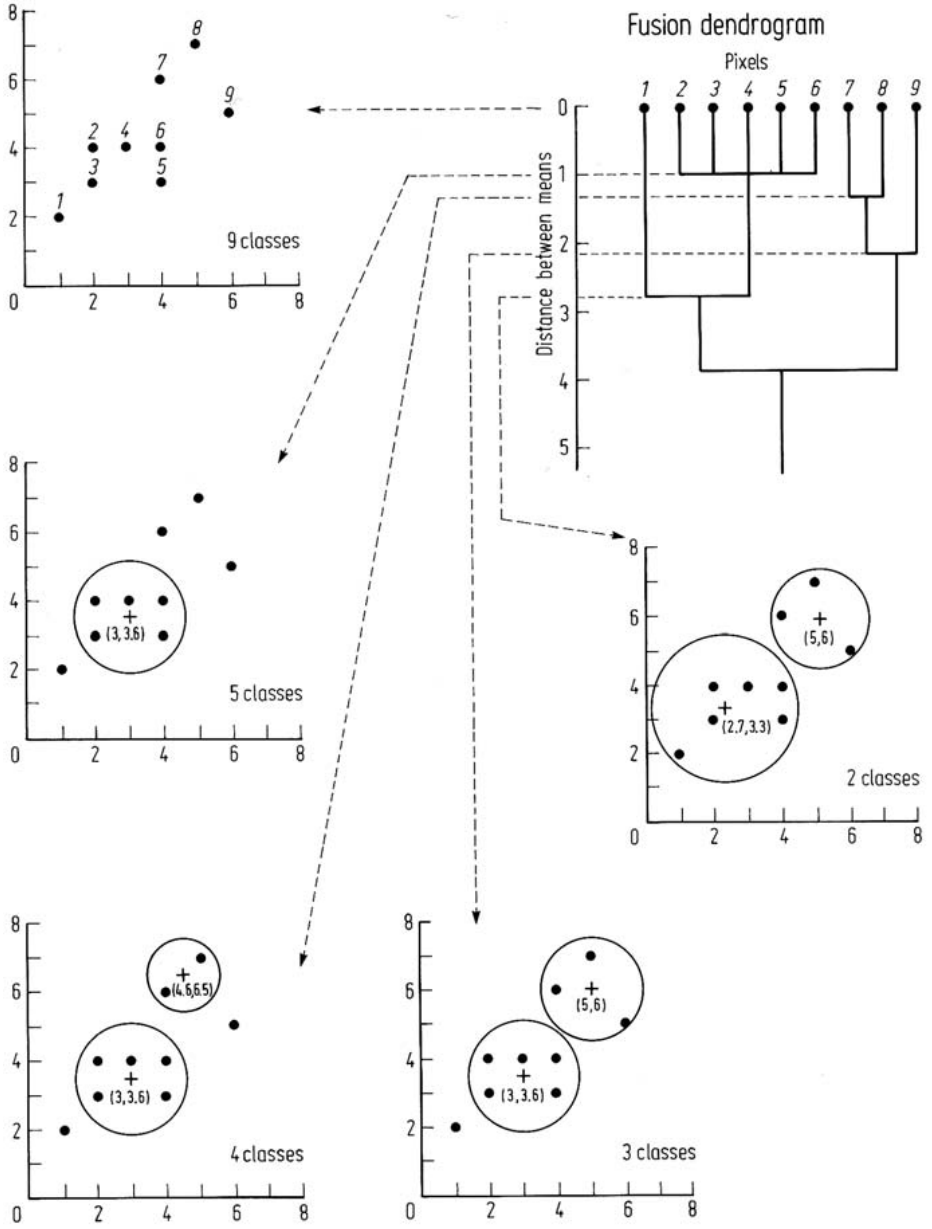
Agglomerative Hierarchical Clustering

- Another clustering technique that does not require the user to specify the number of classes beforehand is **hierarchical clustering**.
- In fact this method produces an *output that allows the user to decide* the set of natural groupings into which the data falls.
- **The procedure**
  - commences by assuming all pixels are individual clusters,
  - it then systematically merges neighbouring clusters by checking distances between means.
  - This is continued until all pixels appear in a single, larger cluster.
- An important aspect of the approach is that the *history of mergings*, or fusions as they are usually called in this method, is displayed on a ***dendrogram***.
  - As can be seen in Figure, this is a diagram that shows at what distances between centres particular clusters are merged.



Agglomerative Hierarchical Clustering

- An **example** of hierarchical clustering, along with its fusion dendrogram is shown in Figure.
  - This uses the same two dimensional data set seen in a previous example,
  - but note that the ultimate cluster compositions are slightly different.
- This demonstrates again that different algorithms can and do produce different clusterings.



Agglomerative Hierarchical Clustering

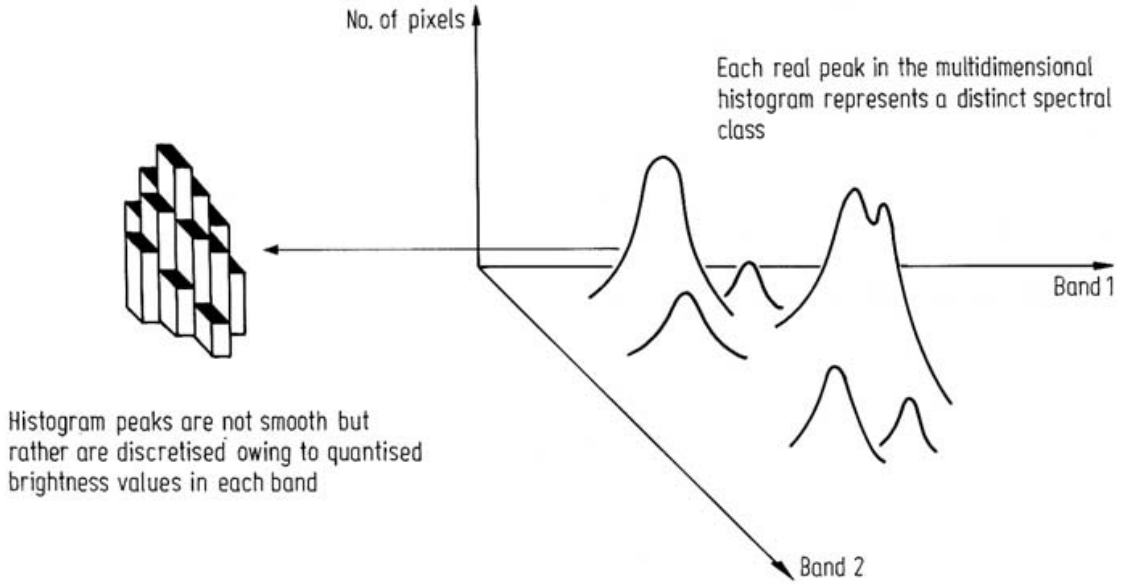
- The fusion dendrogram of a particular hierarchical clustering exercise can be ***inspected*** in an ***endeavour to determine the intrinsic number of clusters*** or spectral classes in the data.
  - **Long vertical sections** in the dendrogram between fusions indicate regions of **"stability"** which **reflect natural data groupings**.
  - In the previous Figure the longest region on the distance scale between fusions corresponds to **two** clusters in the data (with pixel 1 and 9 added somehow later).
  - *One could conclude therefore that this data falls most naturally into two groups*.
- In the example presented, similarity between clusters was judged on the basis of Euclidean distance.
  - Other similarity measures exist and are sometimes used, including divergence metrics (covered in Chap.10 of the textbook).

- The method given above is called ***agglomerative*** in view of its starting with a large number of clusters which it fuses progressively into a single cluster.
  - ***Divisive*** hierarchical clustering procedures also exist in which the data is initialised as a single cluster which is progressively subdivided; these are more expensive computationally and are rarely used.
  - Indeed hierarchical clustering generally does not find a lot of application in remote sensing image analysis since usually a large number of pixels is involved.
  - Nevertheless it is a useful technique for small image data segments particularly since it can reveal data structure.

CLUSTERING BY HISTOGRAM PEAK SELECTION

Clustering by Histogram Peak Selection

- A **multidimensional histogram** of a segment of image data may exhibit ***peaks*** **at the locations of spectral classes or clusters**.
  - Consequently, a further clustering technique that can be adopted (with remote sensing data) is to construct such a histogram and then search it to find the location of its peaks.
  - Pixels are then associated with the nearest peak to produce the clusters. This method has been described by Letts (1978).
- In using histogram peak selection as a clustering technique it is important to keep in mind that the data and the histogram are ***discrete*** in nature and no continuous, as shown in Figure.



Clustering by Histogram Peak Selection

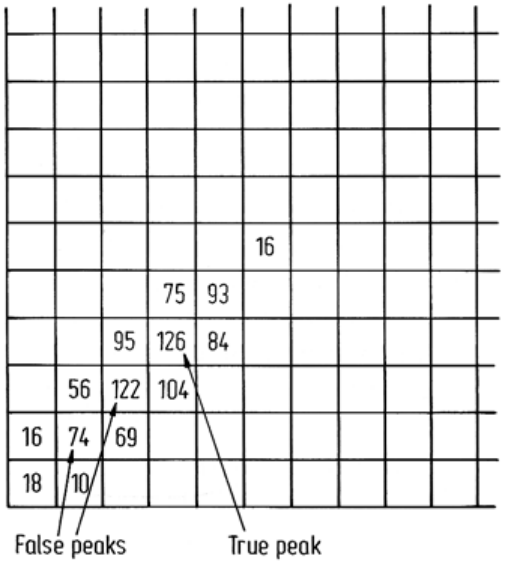
- To see the implications of this, consider the following calculation.
  - A 100 pixel by 100 pixel image segment consists of 10,000 pixels.
  - Suppose this corresponds to data with four spectral components each quantised into 256 levels of brightness.
  - Then the corresponding four dimensional histogram will have $(256)^4$ = 4295 million bins or locations into which counts (pixels) will be accumulated.
- If the bins were filled uniformly then a very sparse histogram would result.
  - Indeed, on the average, there would be only one pixel per half a million bins.
  - Each pixel therefore would appear as a local peak, which clearly would not be a true cluster.
  - The bins of course would not be filled uniformly but nevertheless with bins only one brightness value wide in each spectral component, many artificial peaks will result from some isolated bins occupied by a single pixel and surrounded by empty bins.
- To circumvent this problem the histogram is accumulated with bins which are several brightness values wide in each dimension.

Clustering by Histogram Peak Selection

- In addition the *dynamic range* of the data in each dimension is ascertained beforehand from an inspection of the individual histograms in those dimensions.
- As an ***illustration***,
  - if the individual spectral component histograms for the four bands covered the ranges (35,95), (25,105), (20,80) and (5,65)
  - and bin sizes of 10 brightness values were chosen for each dimension
  - then the total number of four dimensional bins is now 6×8×6×6 = 1728.
  - With a 100×100 pixel image segment therefore, there are, on the average, 6 pixels per bin which is probably acceptable (although low) to guarantee that peaks determined represent the location of real clusters in the data and not artifacts.
  - Clearly resolution is sacrificed but this is necessary to yield an acceptable clustering by this approach.
- The maximum detection algorithm used in this clustering procedure cannot be too sophisticated otherwise the method becomes too expensive to implement.
  - Usually it consists of locating bins in which the count is higher than in the *neighbouring bins along the same row and down the same column*.

Clustering by Histogram Peak Selection

- For correlated data *this can sometimes lead to false indications of peaks*, as depicted in Figure, in the vicinity of true peaks.
- This will be so particularly for smaller bin sizes. A better maximum detection procedure is to check *diagonal neighbours as well* but of course this doubles the search time.



- Clearly this technique is only useful when the dimensionality of the data is low (just a few spectral bands).
  - Because of the enormous number of bins that would be generated, and the extreme sparseness of the resulting histogram, the method is not applicable to hyperspectral data sets.