目標偵測及分類

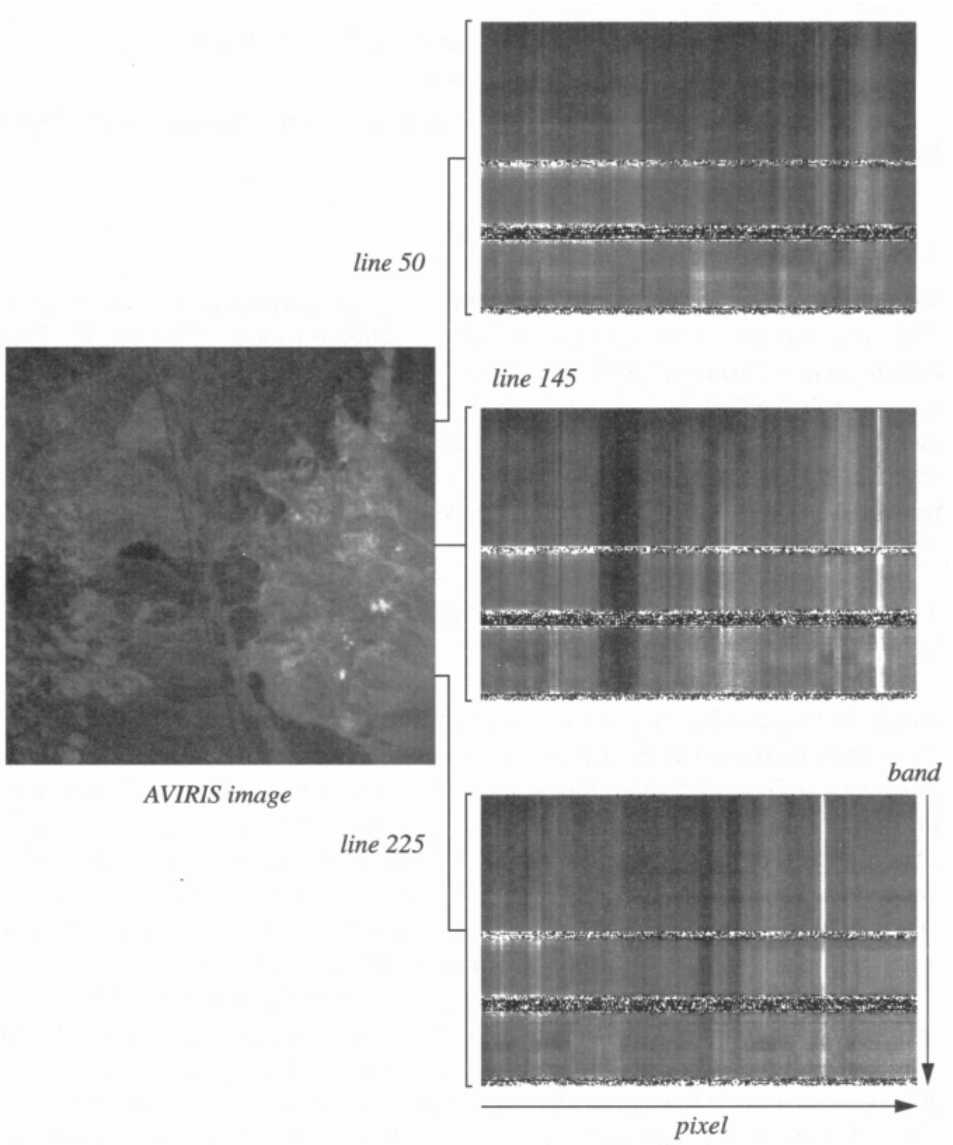


FIGURE 9-45. Visualization of hyperspectral image data by spatial spectrograms. A vertical profile of the display is the spectrum measured for the pixel in that column of the given line. These displays are equivalent to horizontal slices through the hyperspectral image cube (Fig. 1-7). Similar displays could be made along image columns or any arbitrary image profile. The horizontal noise lines are at atmospheric absorption bands. The display is much like the visualization of a single line of BIL-formatted data, shown in Fig. 1-12 of Chapter 1. The same format is also produced directly by a 2-D array imaging spectrometer.

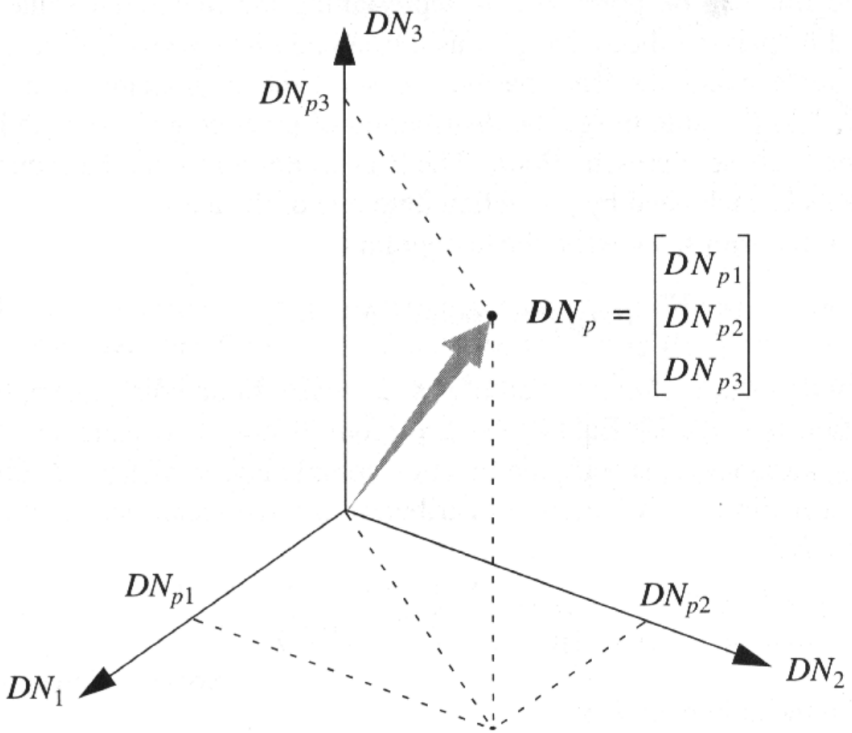


FIGURE 4-5. Visualization of a three-band multispectral image pixel $\mathbf{DN}_p$ as a vector in three-dimensional space.

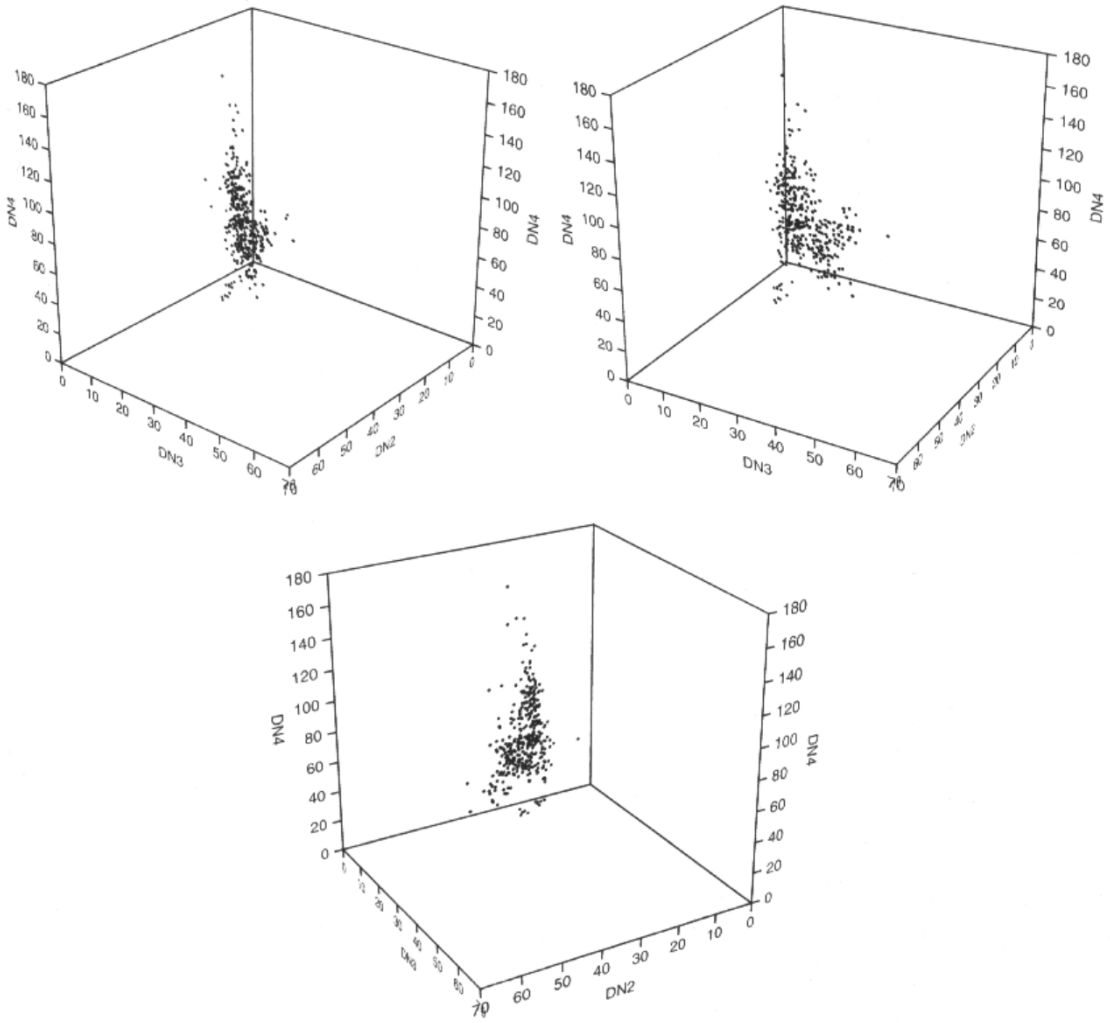


FIGURE 4-6. Three-band scatterplots of bands 2, 3 and 4 of a TM image, viewed from three different directions. Only every 20th sample and line of the image are used to calculate these scatterplots, so that they are not too dense. Every dot in the scatterplot represents one or more pixels with a particular spectral vector. Note that the image data occupies a small fraction of the total DN volume.

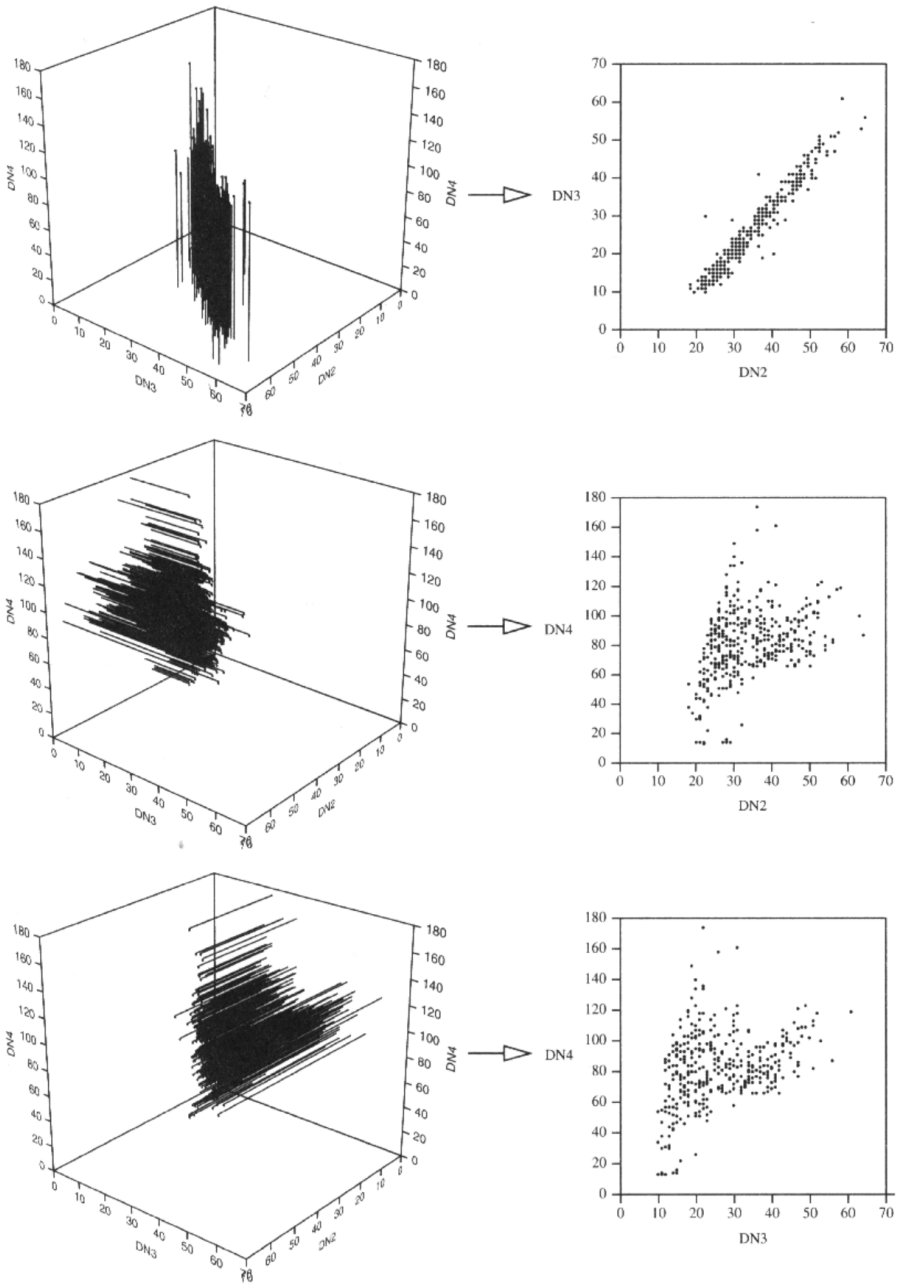


FIGURE 4-7. Reduction of 3-D scatterplots to 2-D scatterplots by projections onto the three bounding planes. The 2-D scatterplots provide multiple views of the data, but do not contain all the information that exists in 3-D.

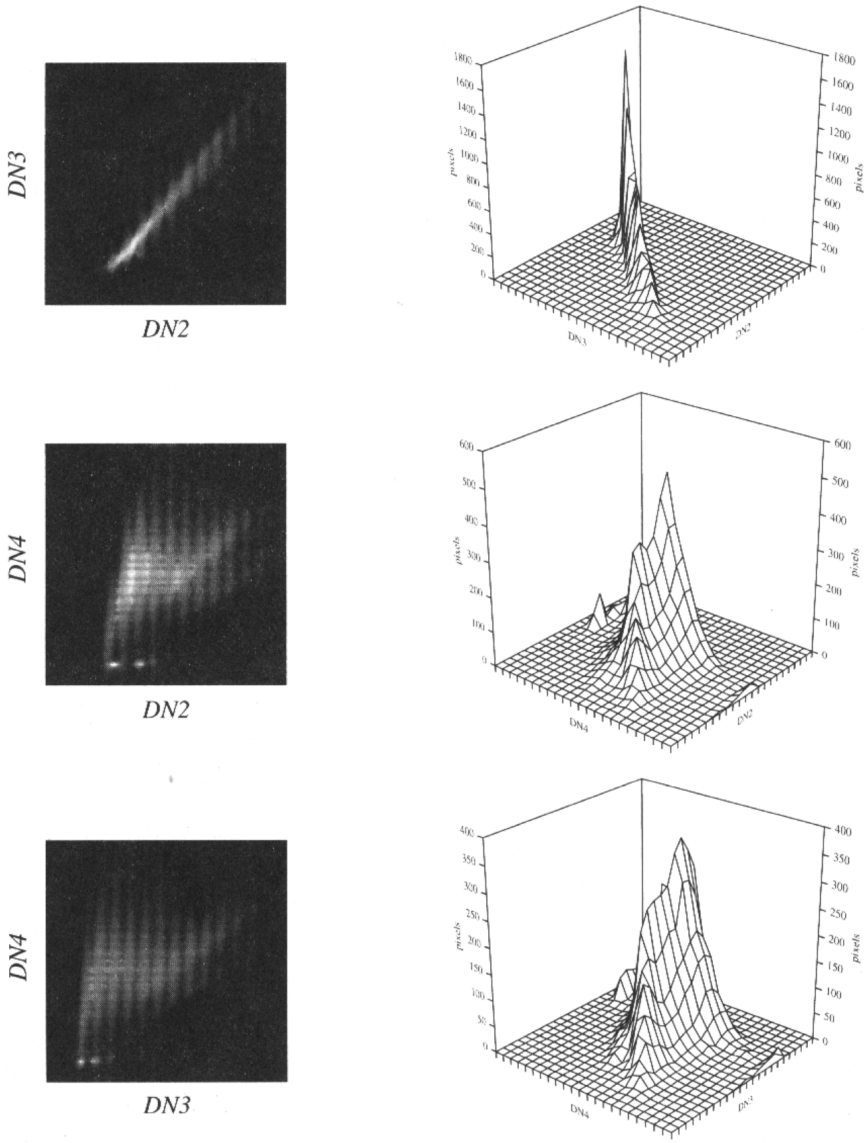


FIGURE 4-9. Two-dimensional scattergrams with density coded as grey levels and displayed as surfaces. The data are from the TM image of Fig. 2-13. Note the two bright "clusters" of pixels at the bottom of the DN4 versus DN2 and versus DN3 greyscale representations. These are from the Briones (left) and San Pablo (right) Reservoirs. The units of the surface plots are number of pixels per ***DN****.*

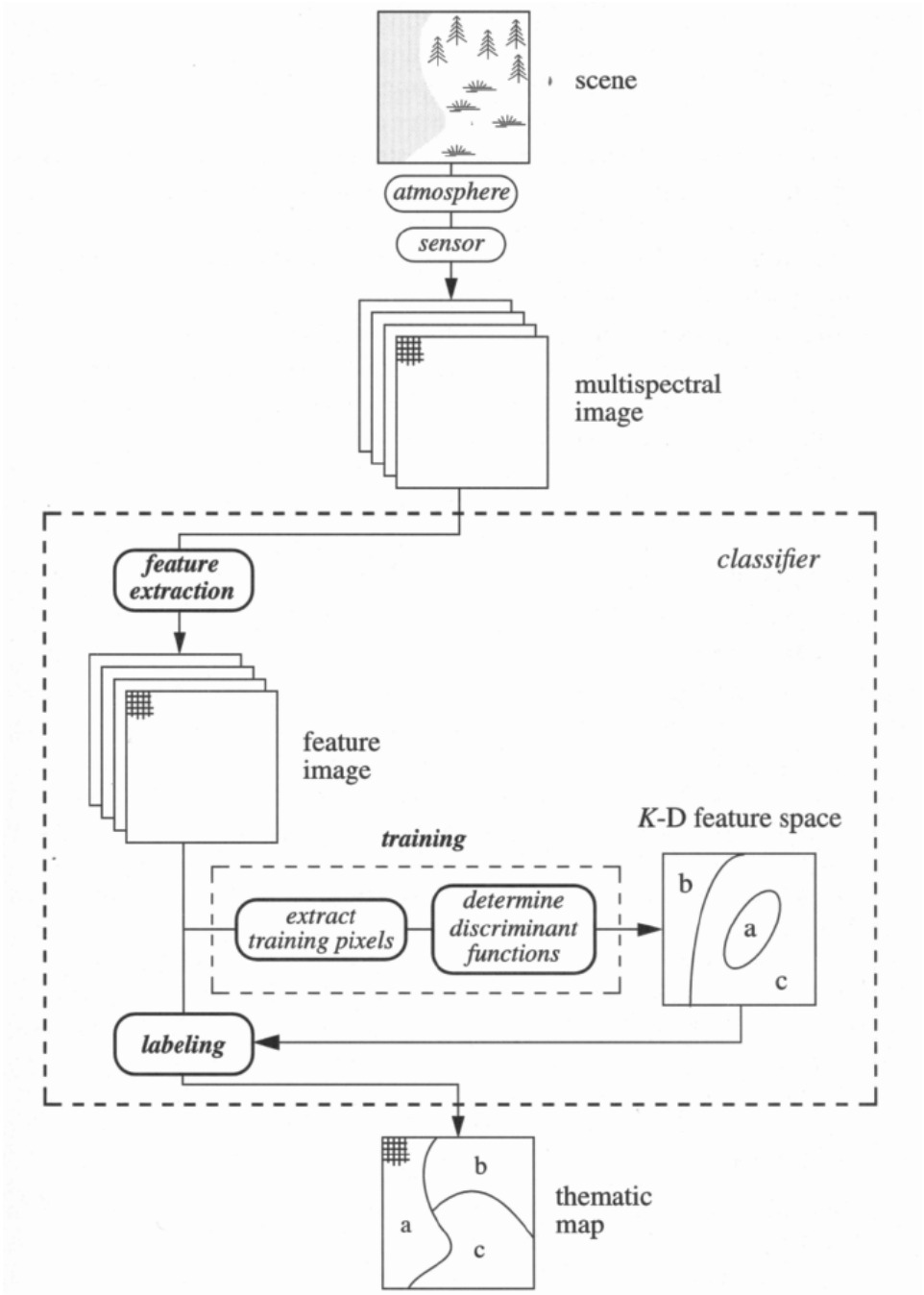


FIGURE 9-1. The data flow in a classification process.

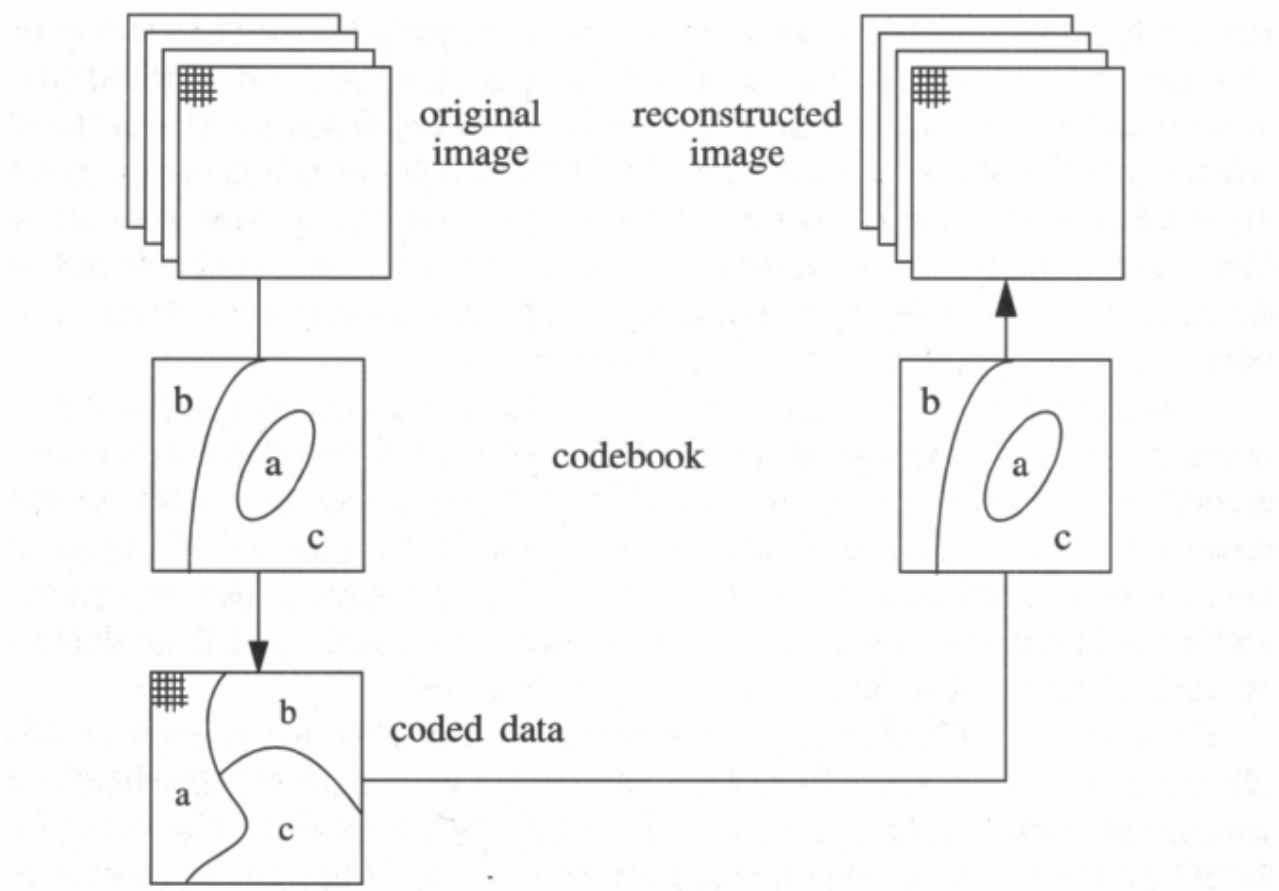


FIGURE 9-2. Classification as a data compression technique. The training step of Fig. 9-1 has been previously performed to create the codebook. The left column is the encoding stage that takes place at the source transmitter, and the right column is the decoding stage that takes place at the receiver. The decoded image may or may not be a perfect reconstruction, as discussed in the text.

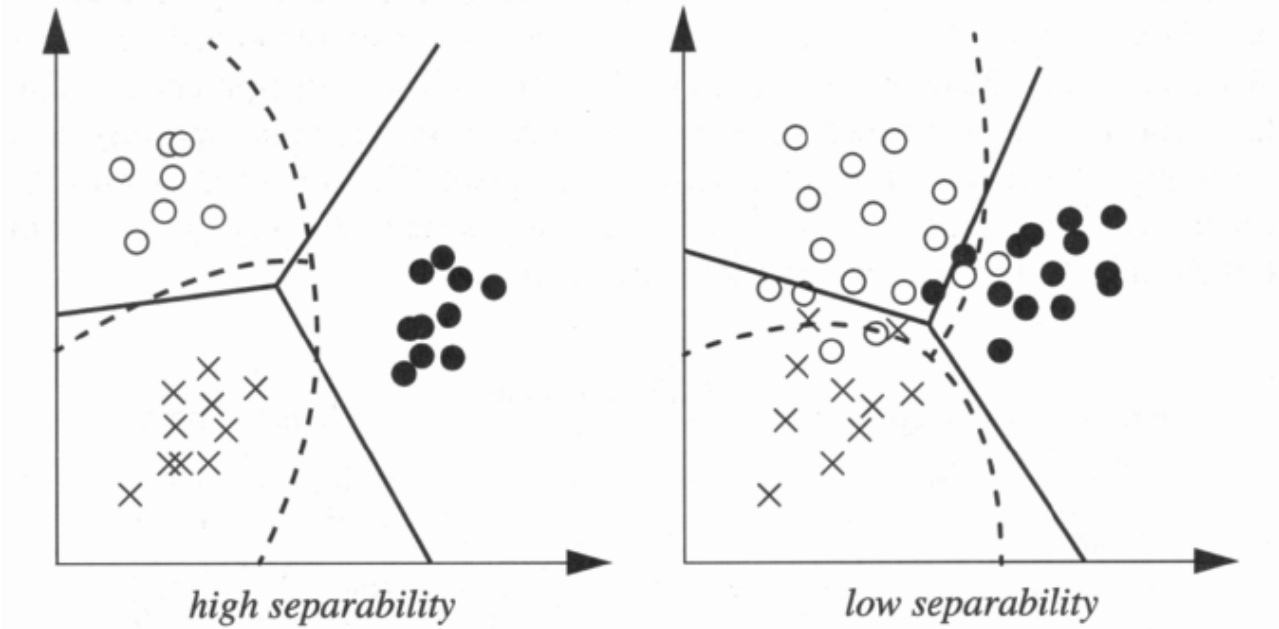


FIGURE 9-3. Two possible situations for training data in feature space and candidate decision boundaries. If the training classes are highly separable, there are many potential decision region partitions that can separate the classes without error; either the solid or dashed lines would do equally well. If the training data from different classes overlap, then a smooth decision boundary is impossible without misclassifications. The exact form of the decision boundary is then critical to the classification error.

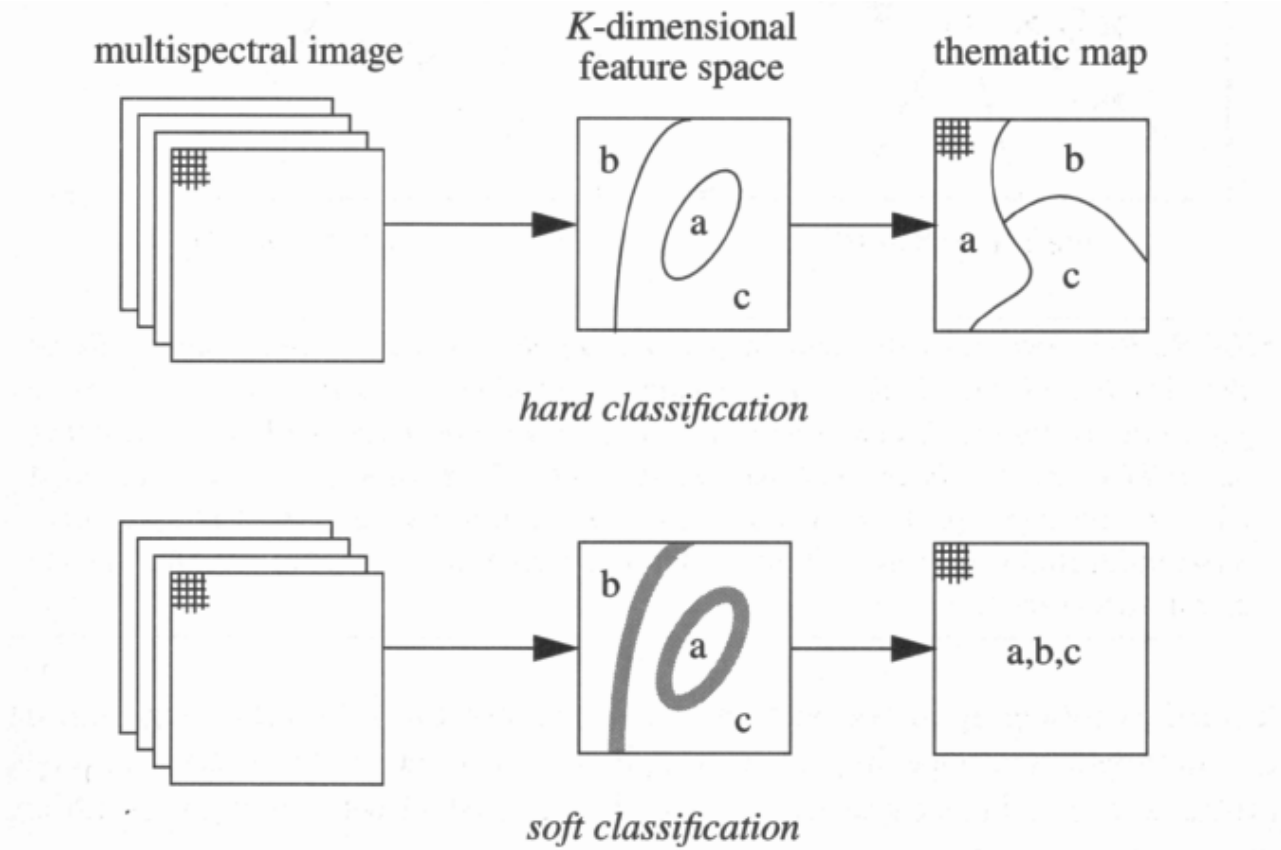


FIGURE 9-4. One way to view the difference between hard and soft classification. In both cases pixel spectral vectors from the multispectral image are passed through the decision space for labeling. In a hard classification the decision is "winner-take-all," with only one label being permitted at each pixel. In a soft classification, the decision is multivalued, with the possibility of more than one label per pixel. Each label has an associated likelihood of being correct. These likelihoods can be interpreted in a number of ways, one of which is that they indicate the proportion of each category within the pixel, à la the mixing model discussed in Sect. 9.8.1.

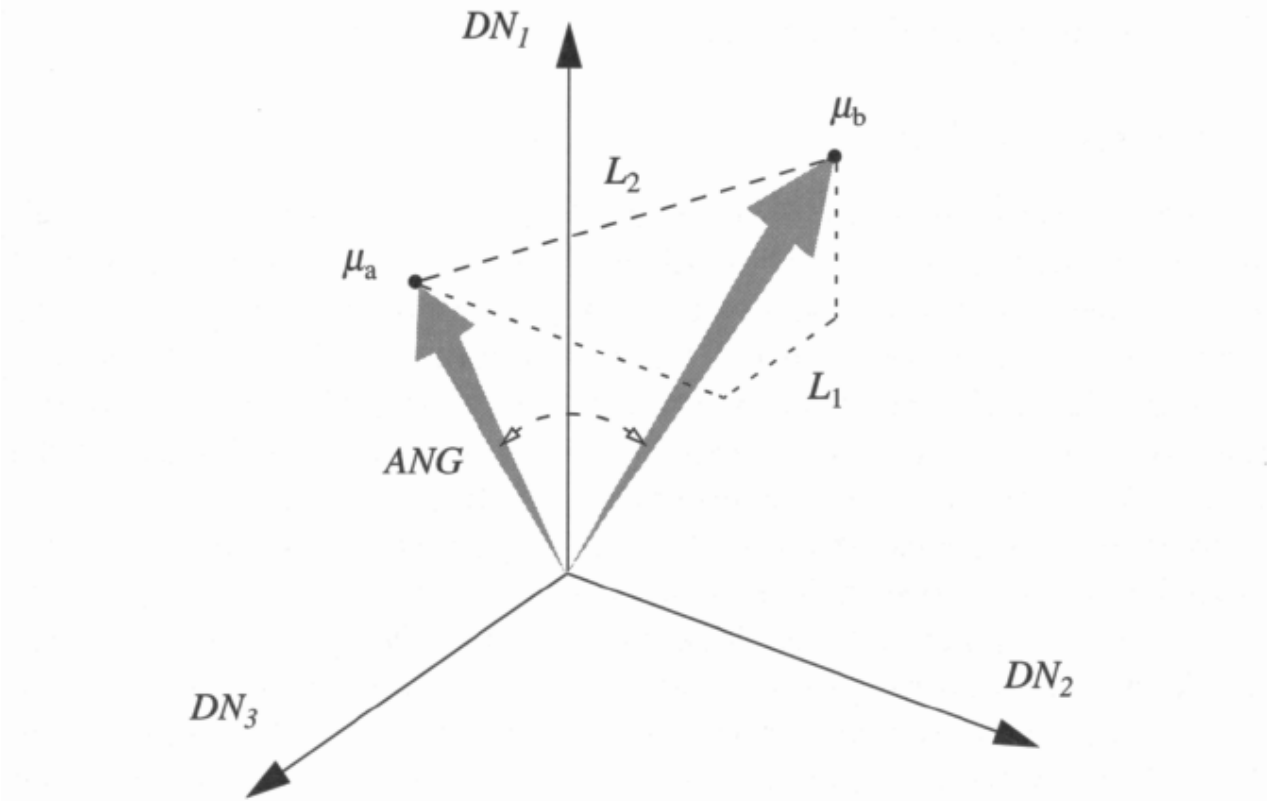


FIGURE 9-5. The L_1, L_2, and ANG distance measures depicted for two vectors in 3-D. Note that ANG, the arc-cosine of the normalized inner (dot) product of the two vectors, is independent of the length of either vector. This property makes it useful for classification of data that is not corrected for topographic shading (refer to Fig. 4-38).

監督性全像素分類法

Distance measure

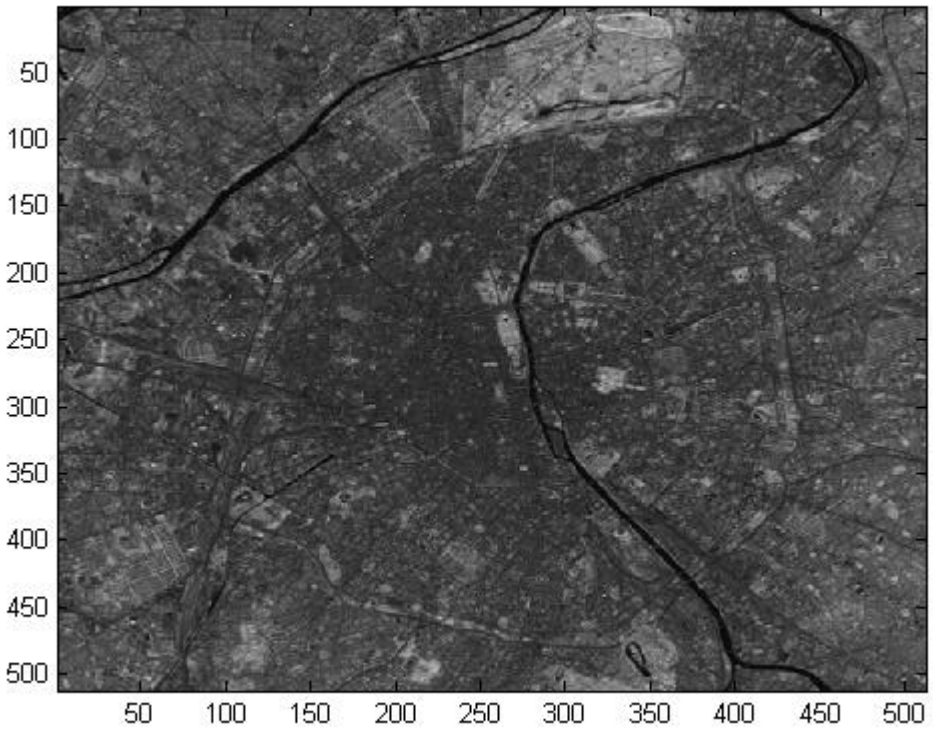
TABLE 9-3. Distance measures between two distributions in feature space. The city block, Euclidean, and angular measures ignore the covariances of the distributions. The normalized city block and MH measures are extensions that include covariance information for each class. The last five measures assume normal class distributions for a and b. All of these distance measures are scalars. Derivations of the normal distribution-based distance measures can be found in many books on statistical pattern recognition, including (Duda and Hart, 1973; Swain and Davis, 1978; Richards, 1993)

name	formula
city block	$L_1 = \lvert\mu_a - \mu_b\rvert = \sum_{k=1}^{K} \lvert m_{ak} - m_{bk}\rvert$
Euclidean	$L_2 = \lVert\mu_a - \mu_b\rVert = [(\mu_a - \mu_b)^T(\mu_a - \mu_b)]^{1/2} = \left[\sum_{k=1}^{K} (m_{ak} - m_{bk})^2\right]^{1/2}$
angular	$ANG = \operatorname{acos}\left(\frac{\mu_a^T \mu_b}{\lVert\mu_a\rVert \lVert\mu_b\rVert}\right)$
normalized city block	$NL_1 = \sum_{k=1}^{K} \frac{\lvert m_{ak} - m_{bk}\rvert}{(\sqrt{c_{ak}} + \sqrt{c_{bk}})/2}$
Mahalanobis	$MH = \left[(\mu_a - \mu_b)^T \left(\frac{\boldsymbol{C}_a + \boldsymbol{C}_b}{2}\right)^{-1} (\mu_a - \mu_b)\right]^{1/2}$
divergence	$D = \frac{1}{2} tr\left[(\boldsymbol{C}_a - \boldsymbol{C}_b)(\boldsymbol{C}_b^{-1} - \boldsymbol{C}_a^{-1})\right] + \frac{1}{2} tr\left[(\boldsymbol{C}_a^{-1} + \boldsymbol{C}_b^{-1})(\mu_a - \mu_b)(\mu_a - \mu_b)^T\right]$
transformed divergence	$D^t = 2[1 - e^{-D/8}]$
Bhattacharyya	$B = \frac{1}{8} MH + \frac{1}{2} \ln\left[\frac{\lvert(\boldsymbol{C}_a + \boldsymbol{C}_b)/2\rvert}{(\lvert\boldsymbol{C}_a\rvert\lvert\boldsymbol{C}_b\rvert)^{1/2}}\right]$
Jeffries-Matusita	$JM = \left[2(1 - e^{-B})\right]^{1/2}$

監督性全像素分類法

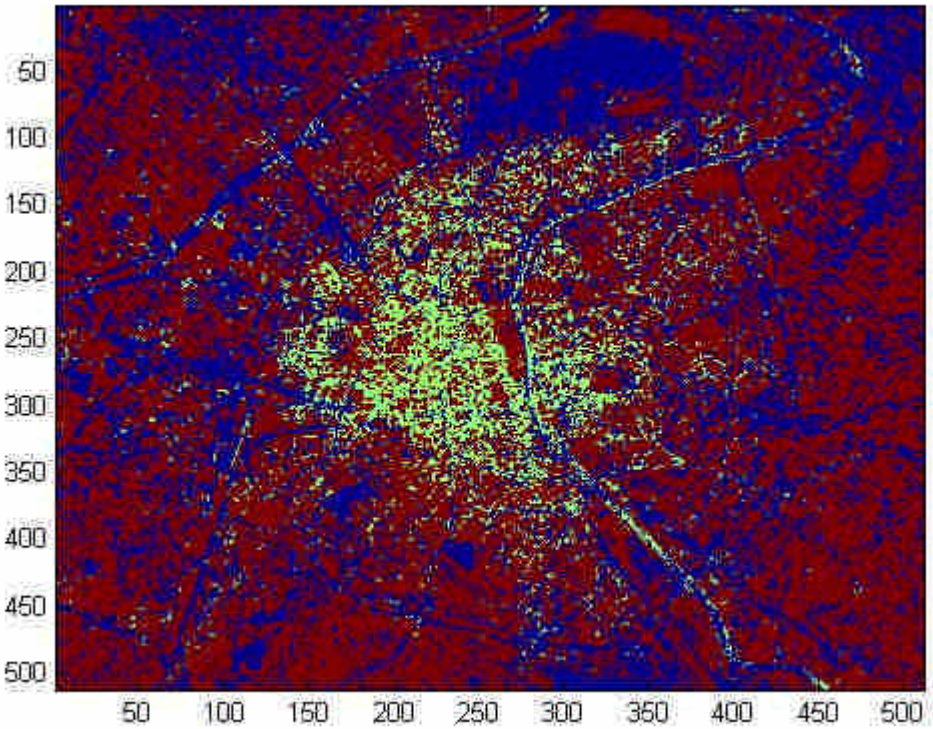
最小距離分類法

```
fid=fopen('paris.lan','r');
fseek(fid,128,-1);
A=fread(fid,[512*7 512],'uint8');
for i=1:7
   Cub(:,:,i)=A(512*(i-1)+1:512*i,:);
end
```



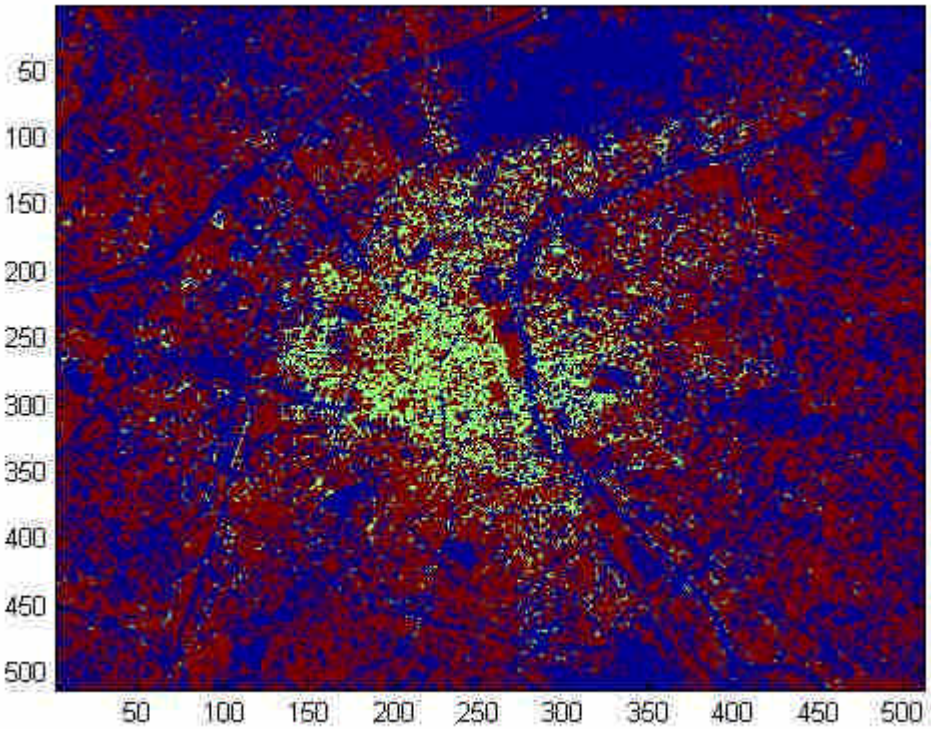
Original image

Euclidean Distance




```
for i=1:512
    for j=1:512
        dis(1)=norm(shiftdim(Cub(i,j,:))-m1);
        dis(2)=norm(shiftdim(Cub(i,j,:))-m2);
        dis(3)=norm(shiftdim(Cub(i,j,:))-m3);
        [Y,k]=min(dis);
        class(i,j)=k;
    end
end
figure
imagesc(class)
```

City Block Distance




```
for i=1:512
   for j=1:512
      dis(1)=sum(abs(shiftdim(Cub(i,j,:))-m1));
      dis(2)=sum(abs(shiftdim(Cub(i,j,:))-m2));
      dis(3)=sum(abs(shiftdim(Cub(i,j,:))-m3));
      [Y,k]=min(dis);
      class1(i,j)=k;
   end
end
figure,imagesc(class1)
```

非監督性全像素分類法

K-means clustering

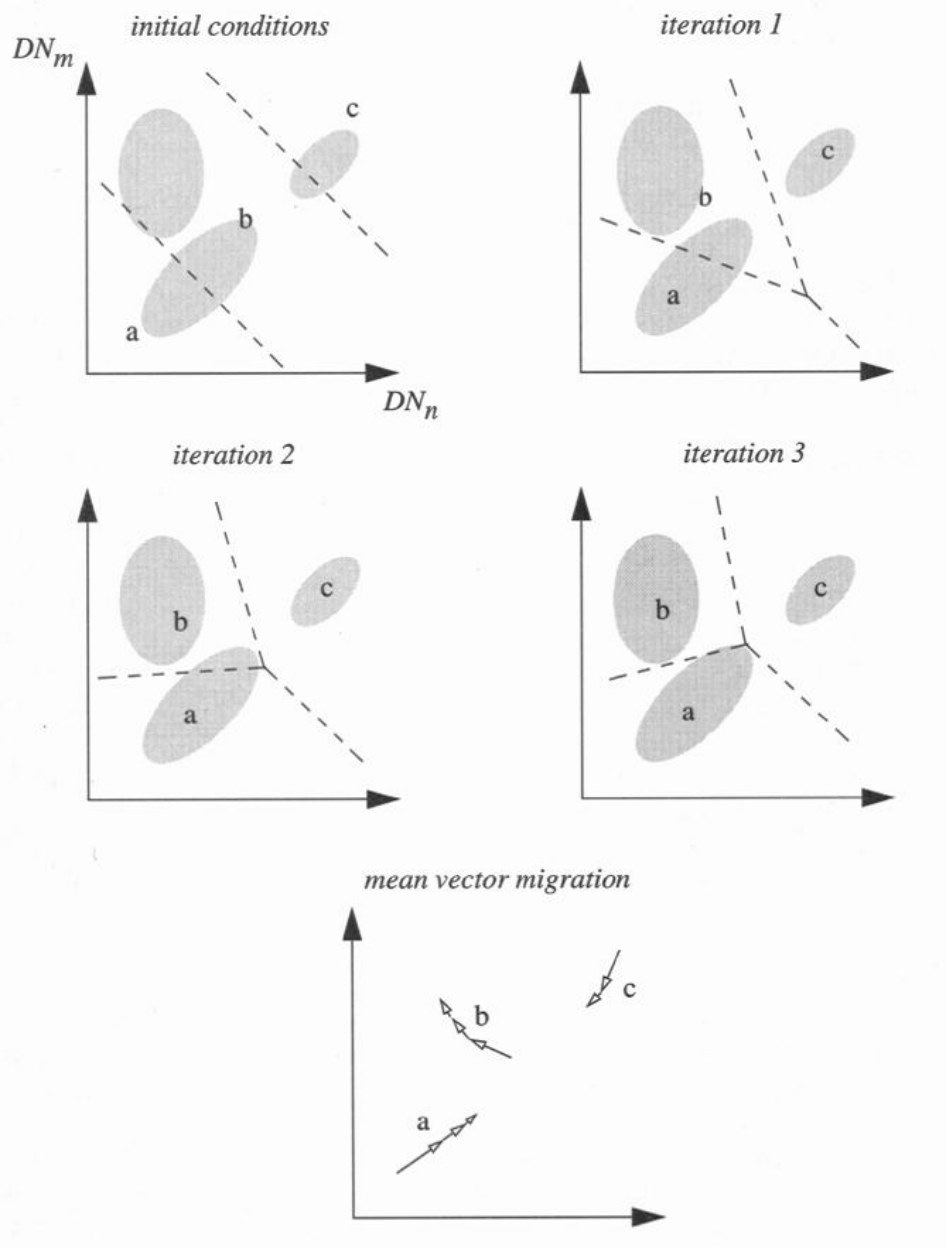


FIGURE 9-6. An idealized data distribution during three iterations of the K-means clustering algorithm with the nearest-mean decision criterion. The data is depicted as three distinct clusters, with the current, estimated mean vector for each class located at a, b and c. The initial "seed" locations are equidistant along the feature space diagonal. In the bottom figure, the movement of the estimated cluster means is shown.

非監督性全像素分類法

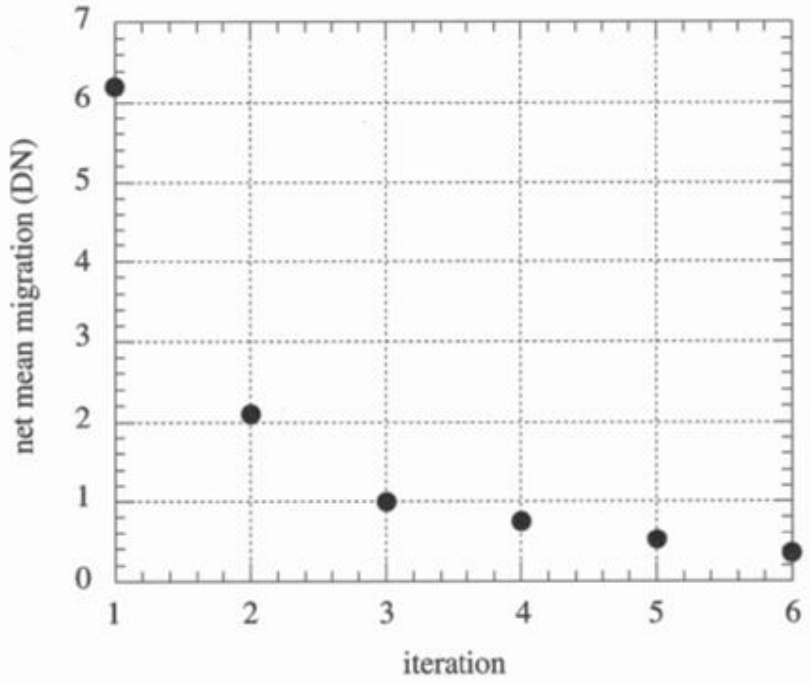


FIGURE 9-7. Typical behavior of the net mean migration from one iteration to the next in the K-means algorithm. The plot is for real image data.

Example

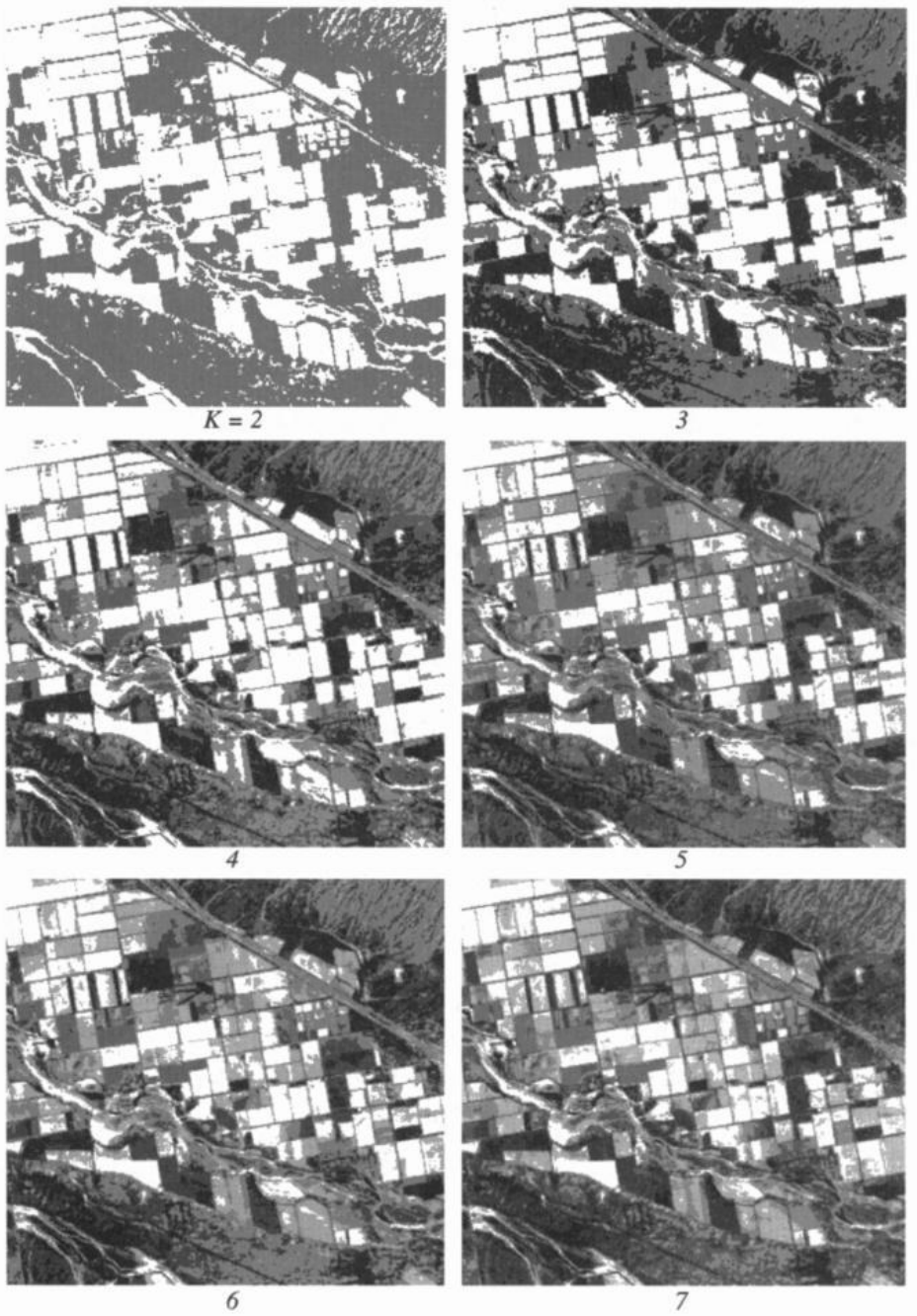


FIGURE 9-8. Final cluster maps for different numbers of clusters.

Final cluster means

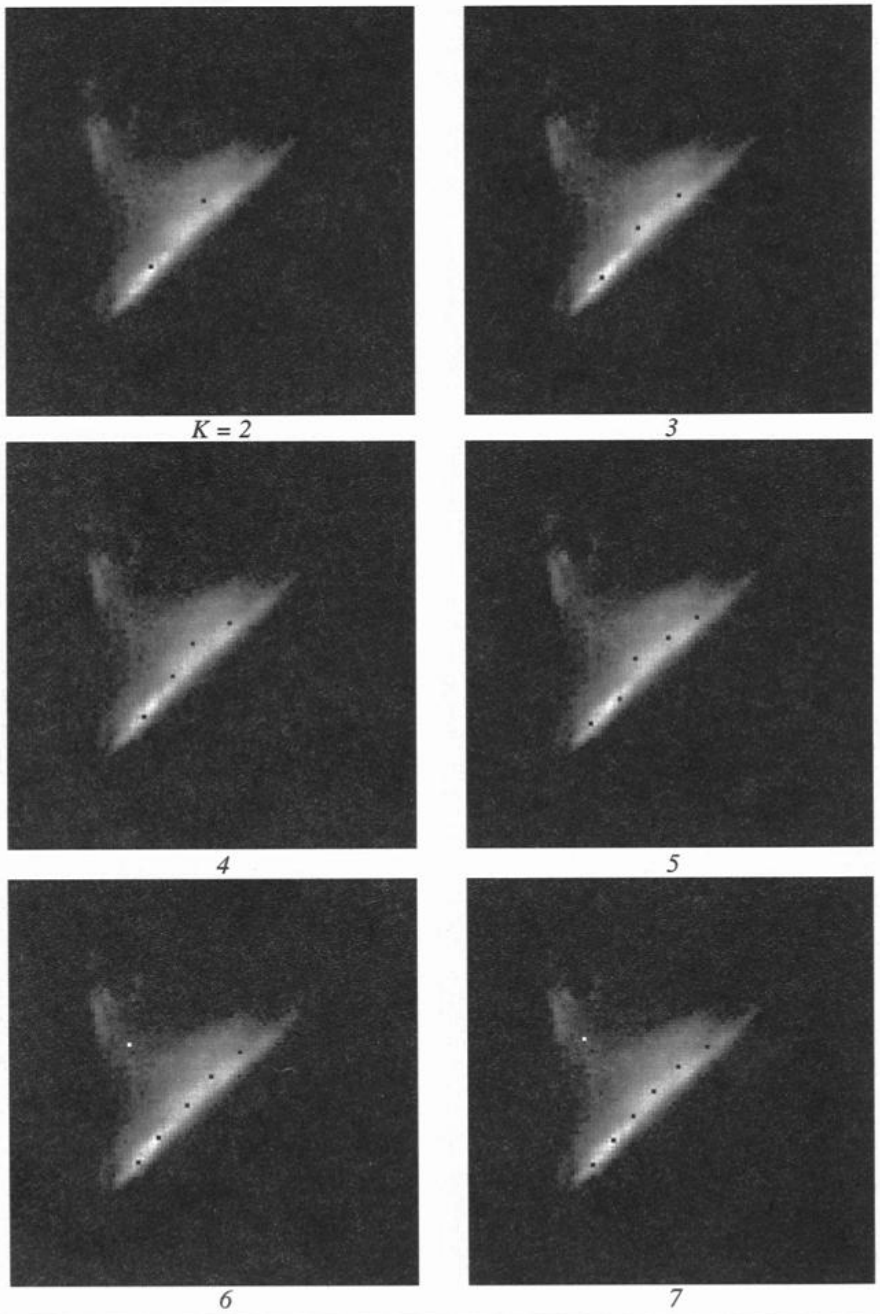


FIGURE 9-9. Band 4 versus band 3 scattergrams for Fig. 9-8 with the final cluster means. The new cluster center that first appears for K equal to 6 represents vegetation.

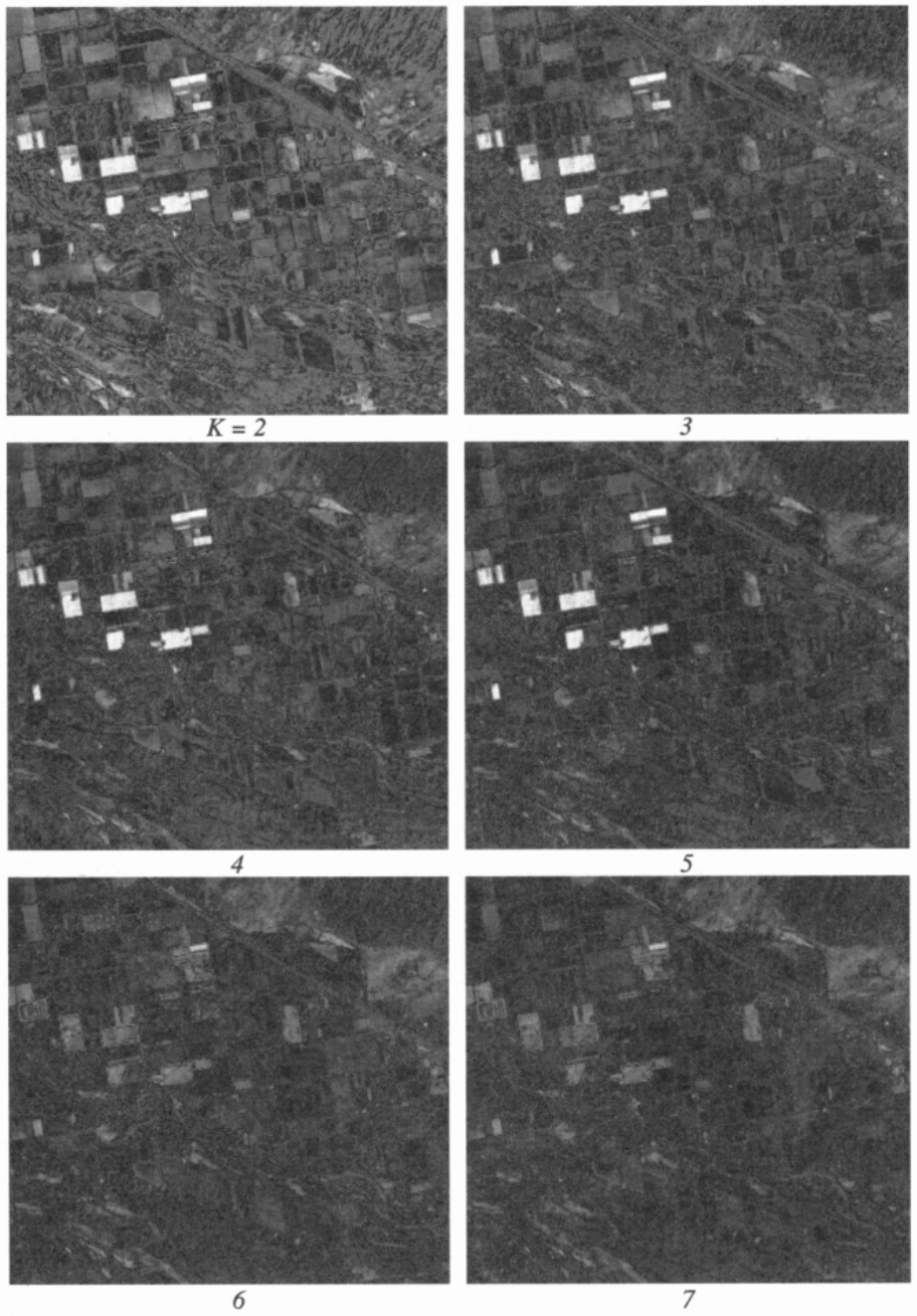
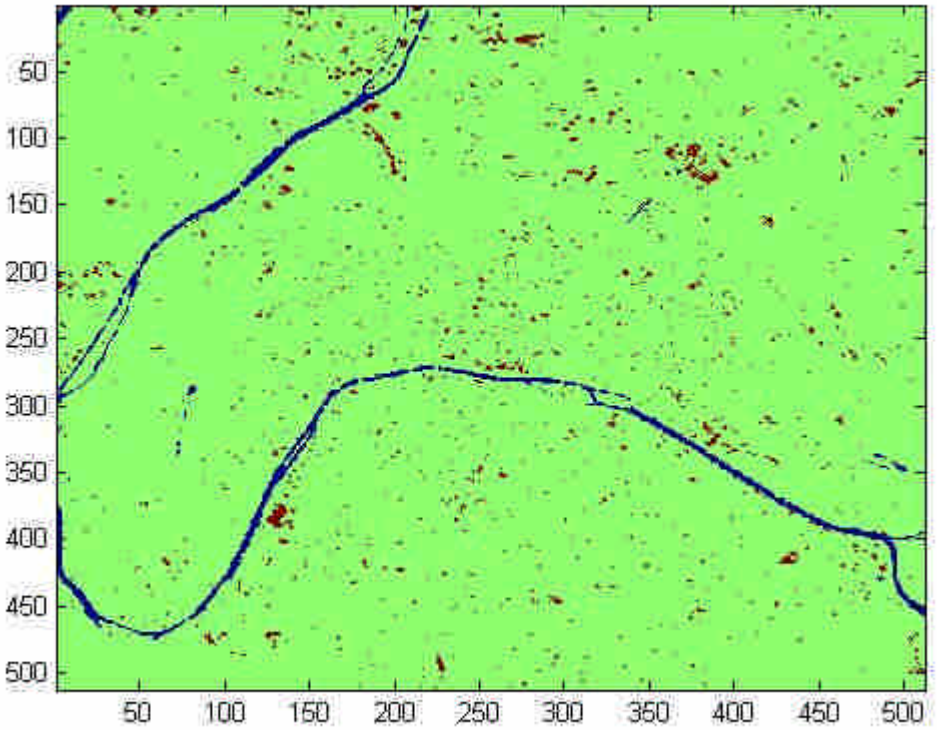


FIGURE 9-10. Residual magnitude error maps between the original image and the approximation given by the cluster mean DNs. Note the greatly reduced error in the crop fields when six or more clusters are used.

非監督性全像素分類法

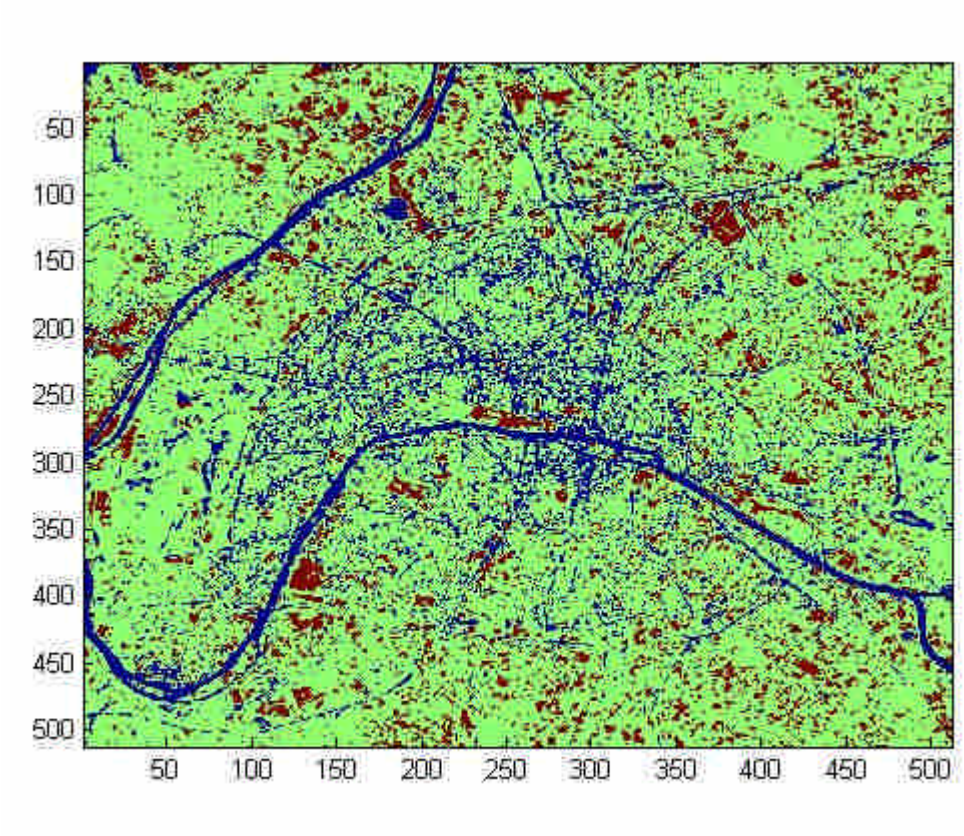
K-mean




```
M=rand(7,3)*255;     %先隨機取中心點
for i=1:512
    for j=1:512
        for k=1:3
            dis(k)=norm(shiftdim(Cub(i,j,:))-M(:,k));
        end
        [Y,m]=min(dis);
        class(i,j)=m;
    end
end
figure
imagesc(class)
%%% 新中心點
for i=1:7
    for k=1:3

      M(i,k)=sum(sum(Cub(:,:,i).*(class==k)))/nn
      z(class==k);
    end
end
```

非監督性全像素分類法




```
%%重複運算
M1=zeros(7,3)
while sum(sum(abs(M-M1)))>1
    M1=M;
    for i=1:512
        for j=1:512
            for k=1:3
            dis(k)=norm(shiftdim(Cub(i,j,:))-M(:,k));
            end
        [Y,m]=min(dis);
        class(i,j)=m;
        end
    end
    figure,imagesc(class)
    drawnow
    for i=1:7
        for k=1:3

      M(i,k)=sum(sum(Cub(:,:,i).*(class==k)))/nnz(
      class==k);
        end
    end
end
```

監督性次像素分類法

An example of spatial mixing

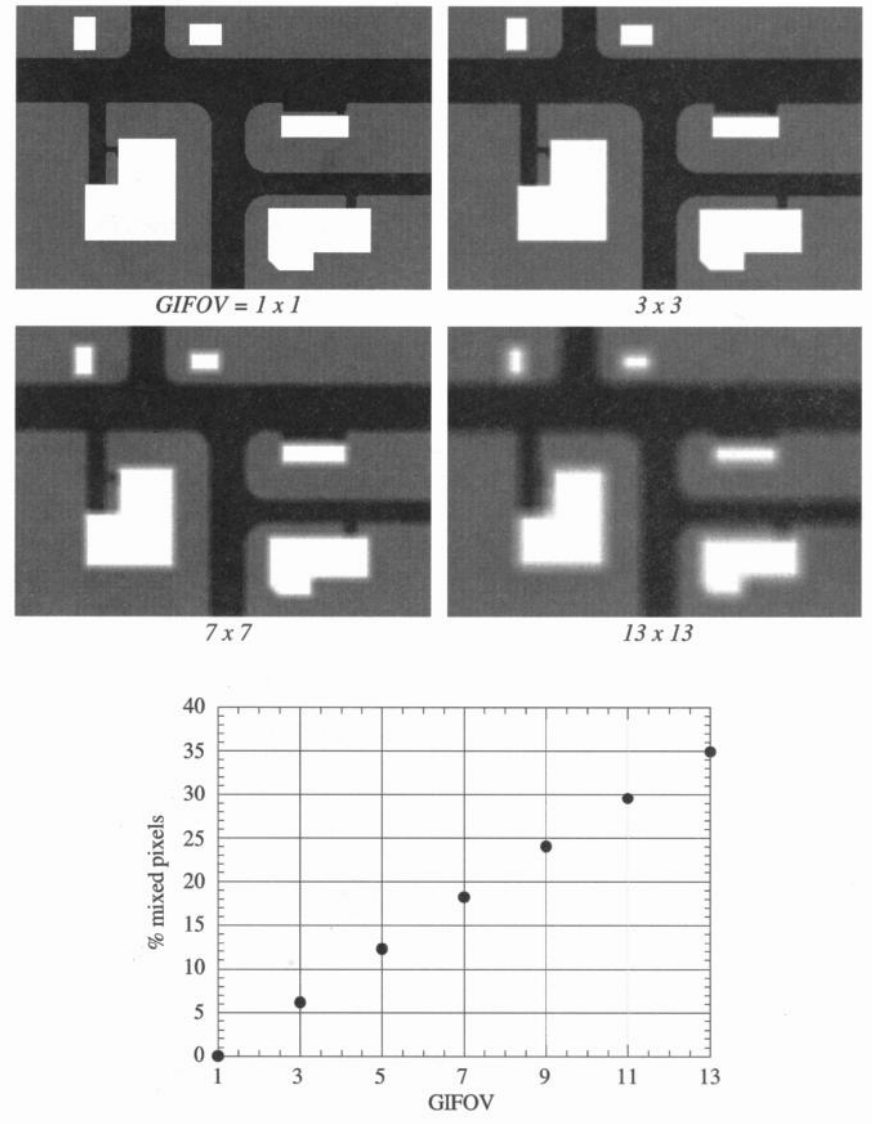


FIGURE 9-39. Simple example to illustrate spatial mixing. A synthetic scene consisting of three types of objects was created at a pixel size of one (upper left). Simulated images were then generated by spatial averaging over a range of GIFOVs. The percentage of mixed pixels for various GIFOVs is shown in the bottom graph.

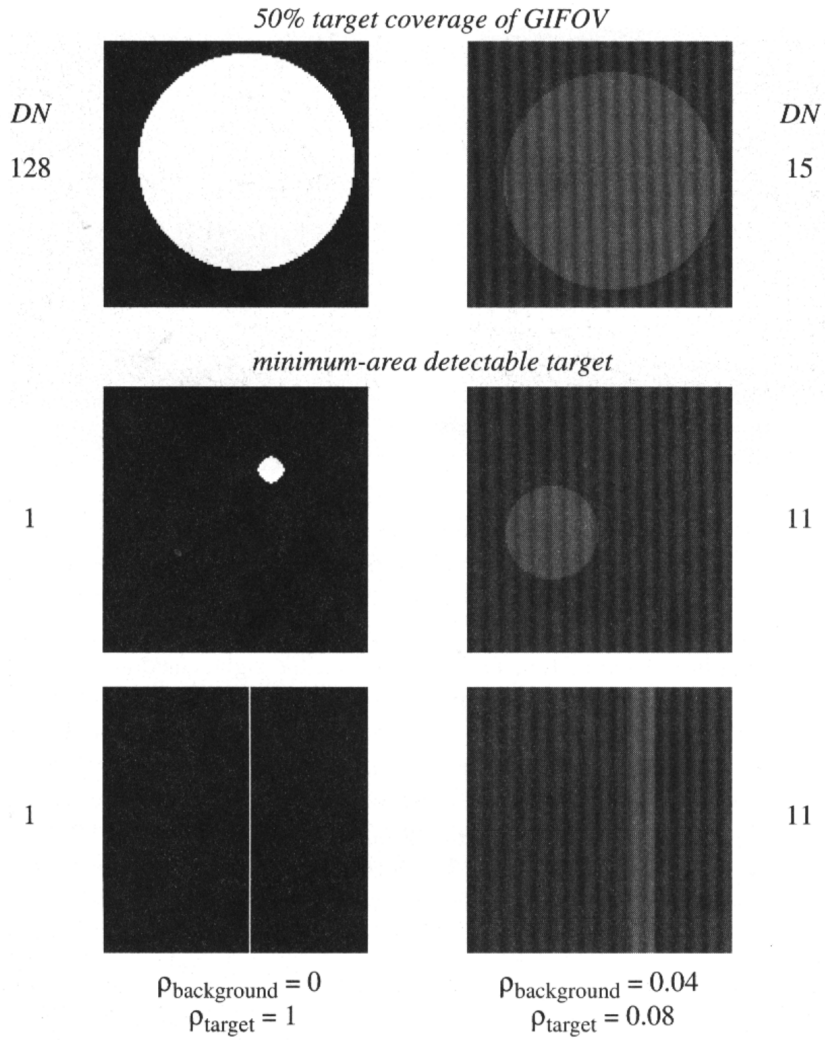
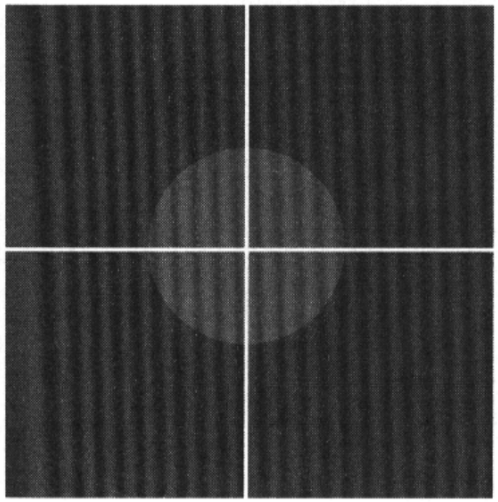
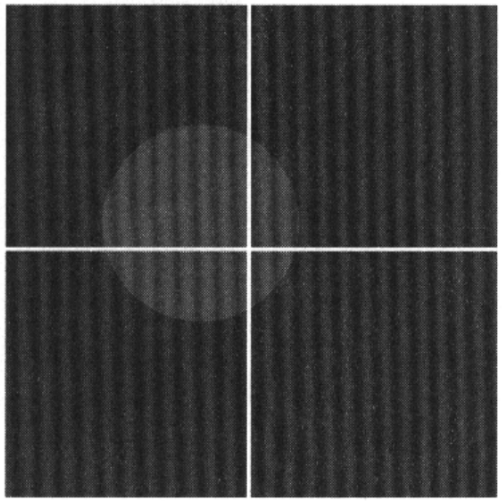


FIGURE 3-3. Detectability analysis for a single target at two different contrasts to the surrounding background and an idealized sensor. The squares represent the area sampled by a single GIFOV. The minimum-area detectable target results in a one DN difference from the background. Note that the target does not have to be centered in the GIFOV (if the sensor response is uniform across the GIFOV); the spatially-integrated signal will be the same for any internal location. For the same reason, the shape of the target cannot be discerned from a single pixel. If the target is linear, such as a road or bridge, then the shape may be inferred by the context of several pixels. A case in point is Fig. 3-2.



DN = 11	*DN* = 11
DN = 11	*DN* = 11



DN = 13	*DN* = 11
DN = 12	*DN* = 10

FIGURE 3-4. The effect of spatial phasing between the pixel grid and the ground target. Four adjacent GIFOVs are shown and the target area is 50% of the GIFOV area. On the left, the target fills 12.5% of each GIFOV. On the right, the target occupies 30%, 5%, 2%, and 13% of the four GIFOVs. The location of the grid is unpredictable prior to imaging any given scene; it could be anywhere within ± 1/2 pixel interval with equal probability. The imaging of long linear features can sometimes allow precise measurement of the phase to a small fraction of a pixel.

監督性次像素分類法

Linear Mixing Model

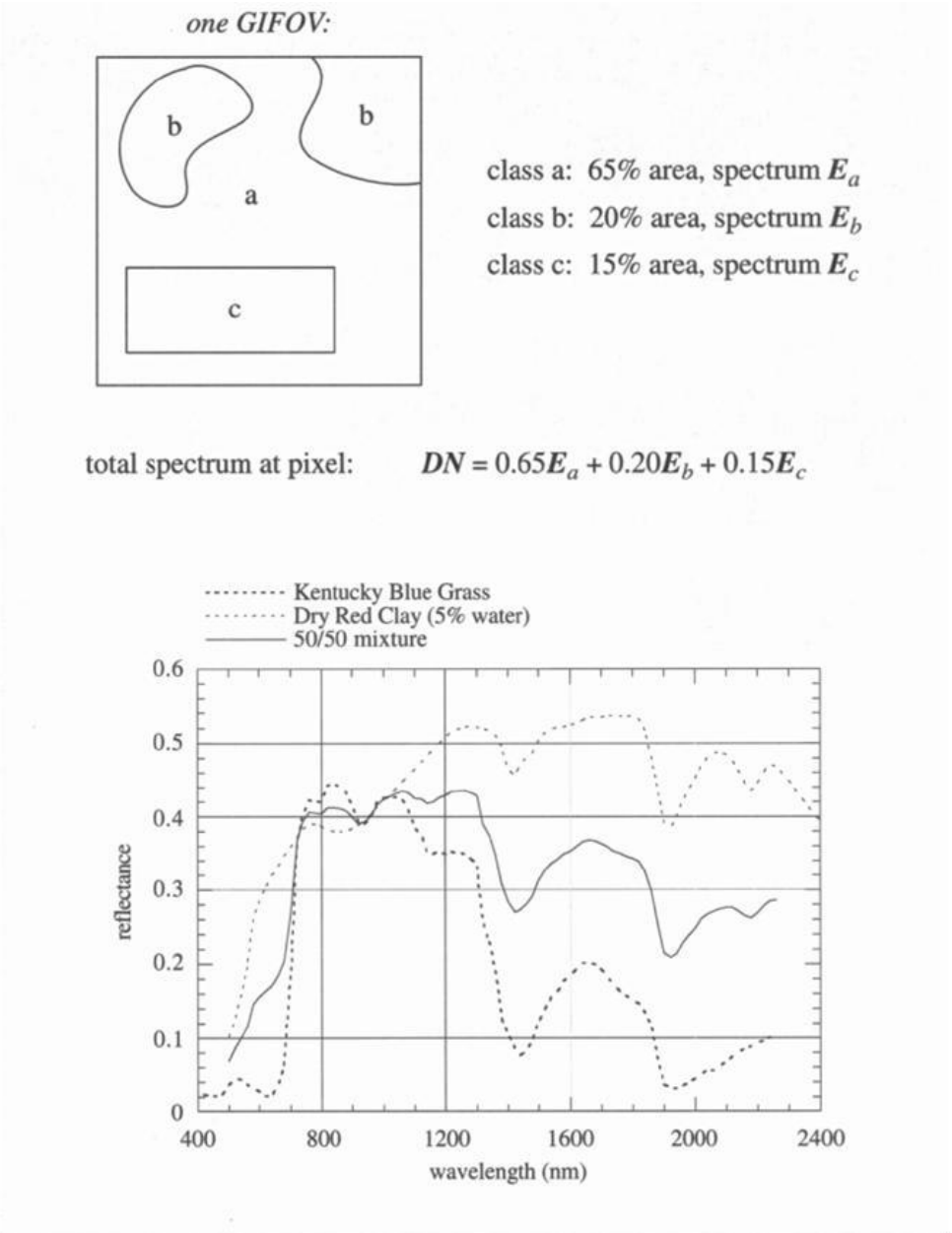
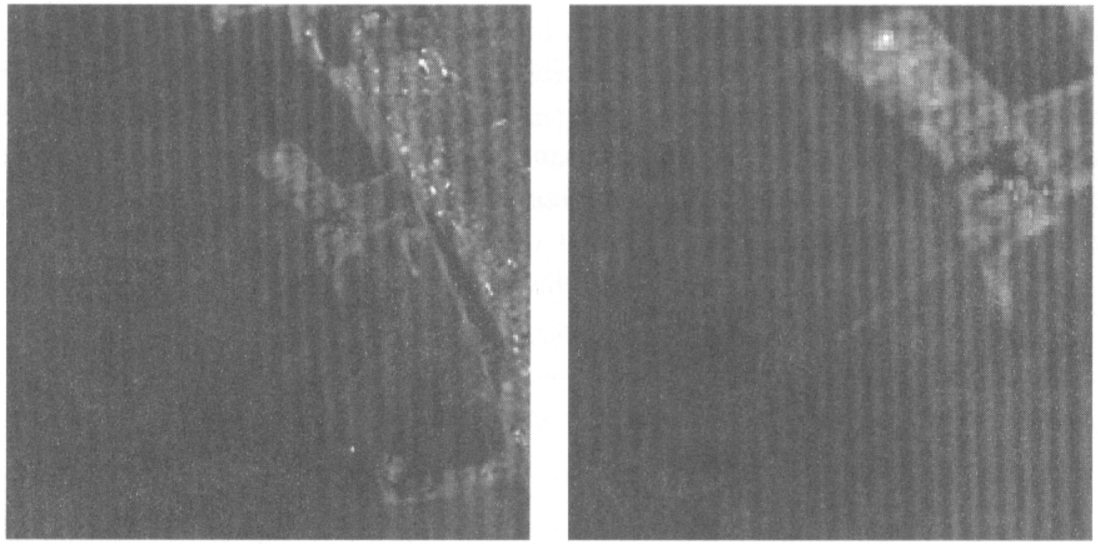


FIGURE 9-40. The linear mixing model for a single GIFOV. The boundaries between objects can be any shape and complexity; only the fractional coverages and individual spectral reflectances are important. An example using the spectral reflectance data of Chapter 1 is shown below. The mixed signature has lost most of the characteristic vegetation "edge" at 700nm.



1X *2X*

FIGURE 3-2. Example of sub-pixel object detection. This is part of a TM band 3 image of San Francisco showing the Berkeley Pier. The pier is 7m wide and made of concrete. An older extension of the same width, but made of wood with a lower reflectance than concrete, is barely visible. (Acknowledgments to Joseph Paola for providing details on the size and construction of the pier.)

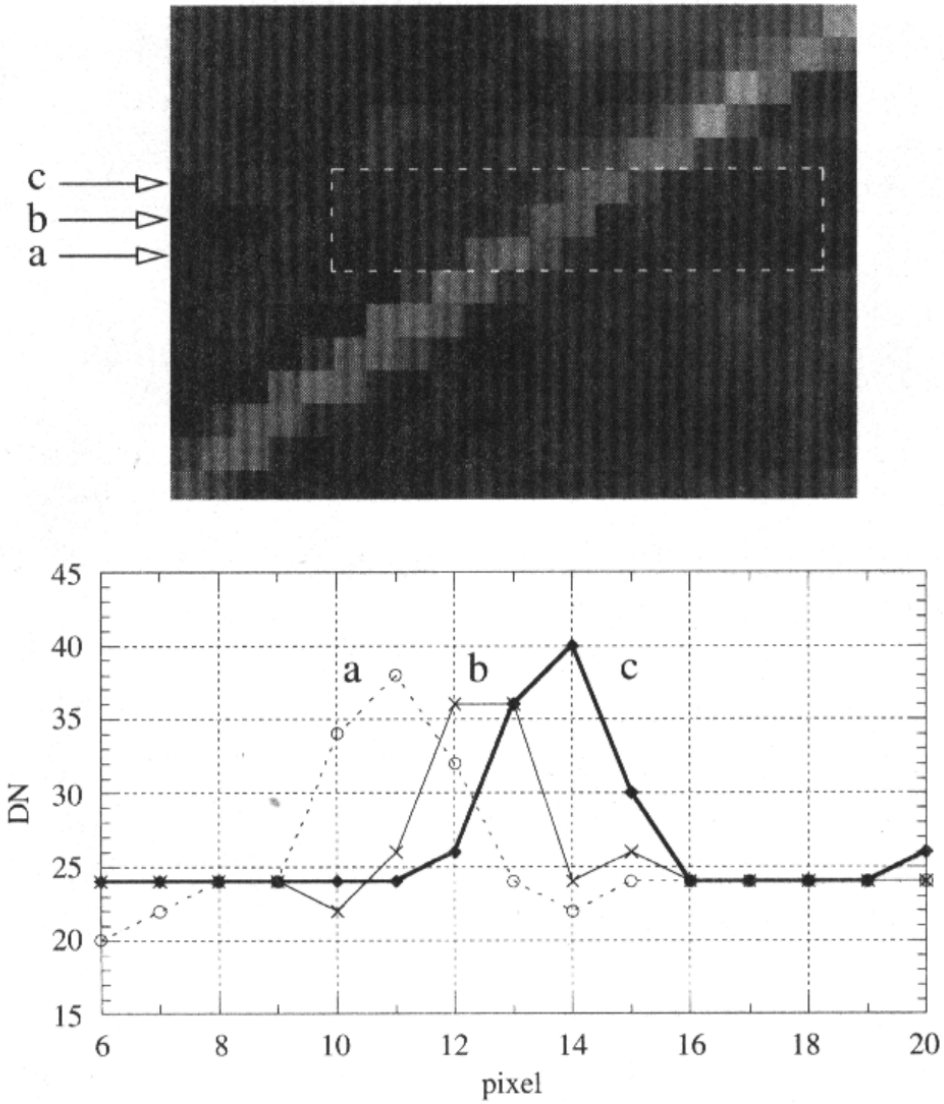


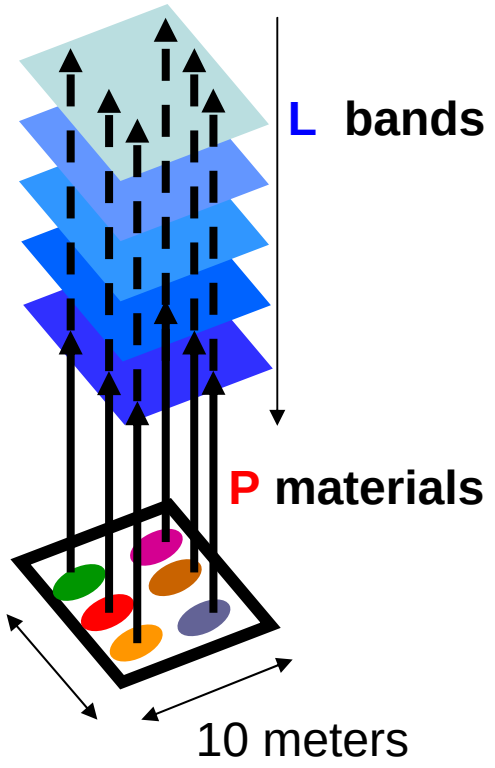
FIGURE 3-5. A contrast-enhanced enlargement of Fig. 3-2 and DN profile plots along three adjacent scanlines near the center of the pier, illustrating the sample-scene phase effect. The pier's profile is different in each line. (The linear interpolation between individual pixels is only to aid visualization of the graph)

監督性次像素分類法

- *Linear Spectral Mixture Analysis*

$$r = M \cdot \alpha + n$$

$$\begin{aligned} r_1 &= m_{11}\alpha_1 + m_{21}\alpha_2 + \cdots + m_{P1}\alpha_P + n_1 \\ r_2 &= m_{12}\alpha_1 + m_{22}\alpha_2 + \cdots + m_{P2}\alpha_P + n_2 \\ &\vdots \quad\quad \vdots \quad\quad \vdots \quad\quad \vdots \quad \vdots \\ r_L &= m_{1L}\alpha_1 + m_{2L}\alpha_2 + \cdots + m_{PL}\alpha_P + n_L \end{aligned}$$



$L < P \Rightarrow$ infinitely number of solutions

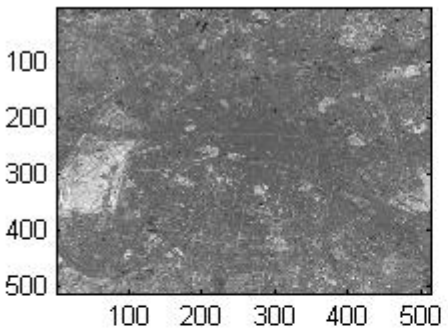
$L = P \Rightarrow$ only one solution

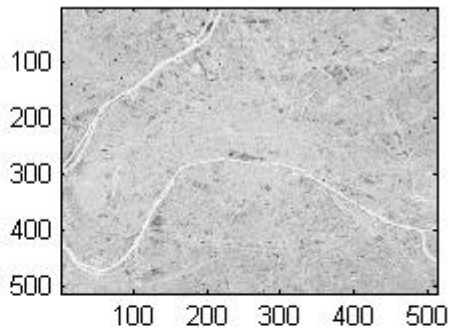
$L > P \Rightarrow$ least square error solution

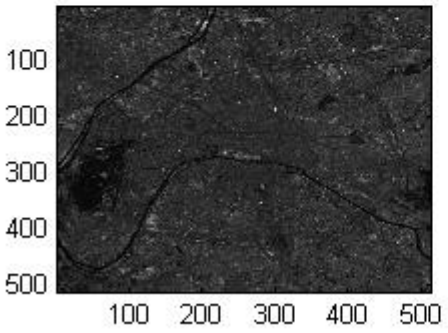
Least square error $\hat{\alpha}_{LS} = \left(M^T M\right)^{-1} M^T r$

監督性次像素分類法

最小方差法








```
for i=1:512
    for j=1:512
        abd(i,j,:)=inv(M'*M)*M'*shiftdim(Cub(i,j,:));
    end
end
figure
for i=1:3
    subplot(2,2,i)
    imagesc(abd(:,:,i))
end
colormap(gray)
```

監督性次像素分類法

最大似然法

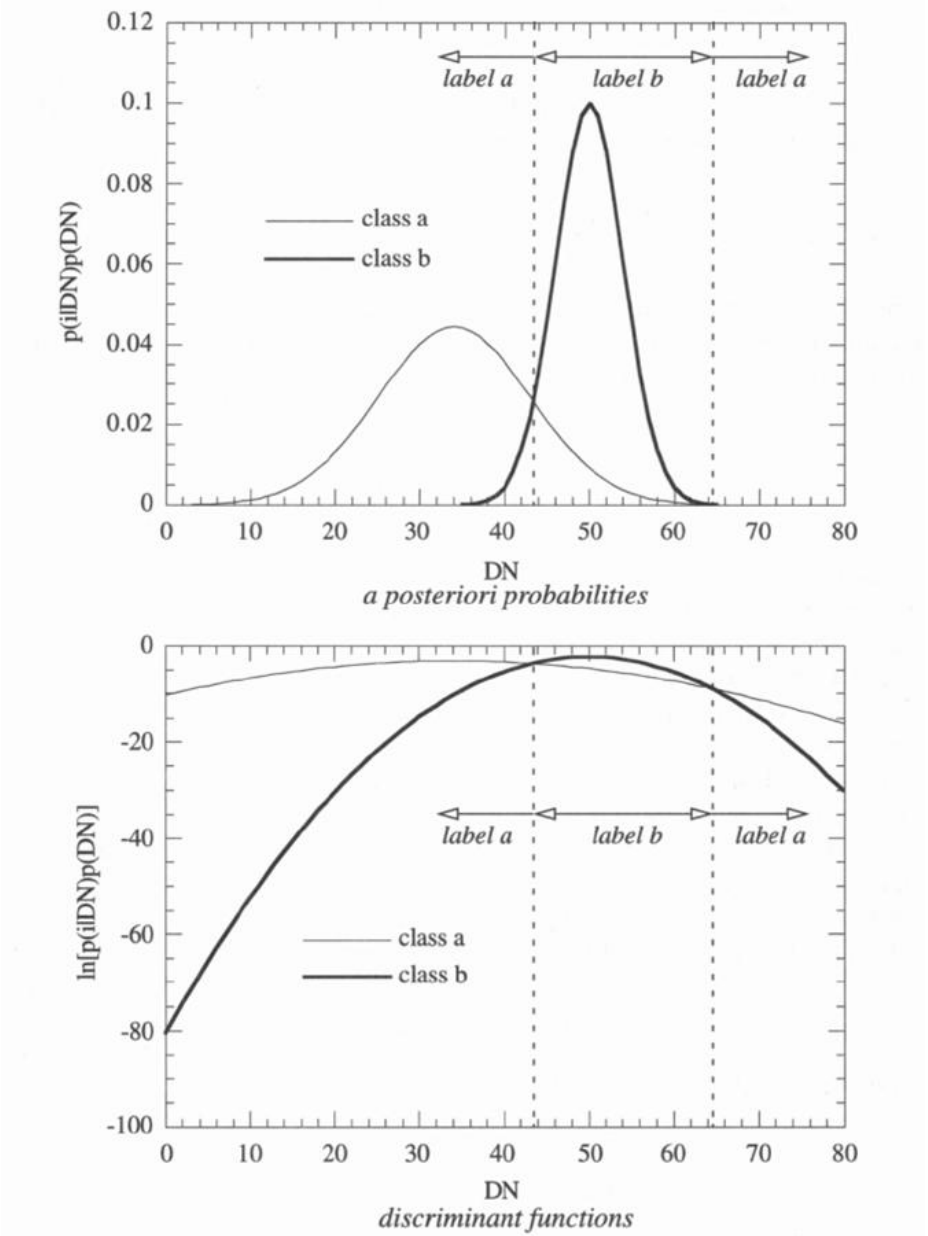


FIGURE 9-22. Maximum-likelihood decision boundaries for two Gaussian DN distributions in one dimension. The class statistics are: $\mu_a = 34$, $\sigma_a = 9$, $\mu_b = 50$, $\sigma_b = 4$. Note the decision boundary on the right is not visible in the upper graph because of the ordinate scale, but becomes clear in the discriminant function graph below.

監督性次像素分類法

最大似然法

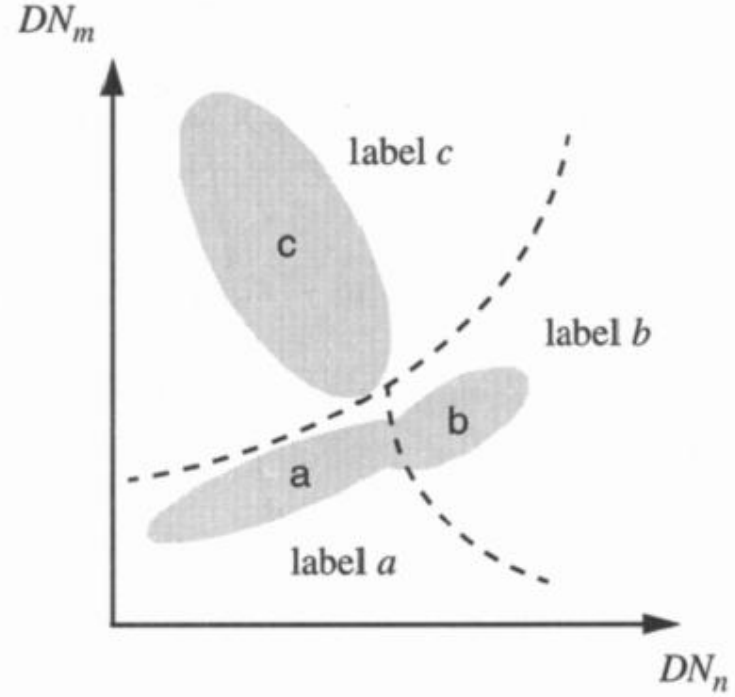


FIGURE 9-23. Maximum-likelihood decision boundaries for three classes in two dimensions, with Gaussian distributions for each class. The boundaries are quadratics in 2-D. Their dependence on the covariance matrices of the individual classes is shown in (Duda and Hart, 1973).

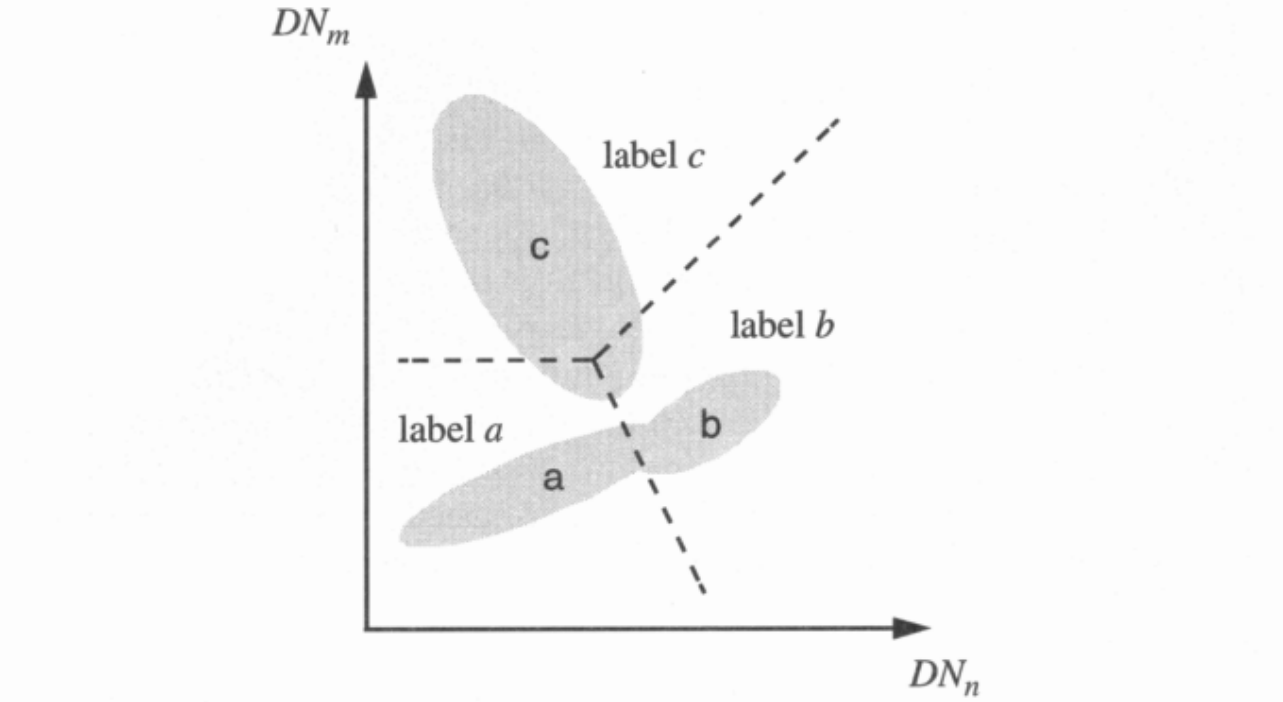


FIGURE 9-24. Nearest-mean decision boundaries for three classes in two dimensions, using the L_2-distance measure. Distance thresholds could be implemented either as in the level-slice algorithm, or as circular boundaries, centered at each class mean, at a specified radius.

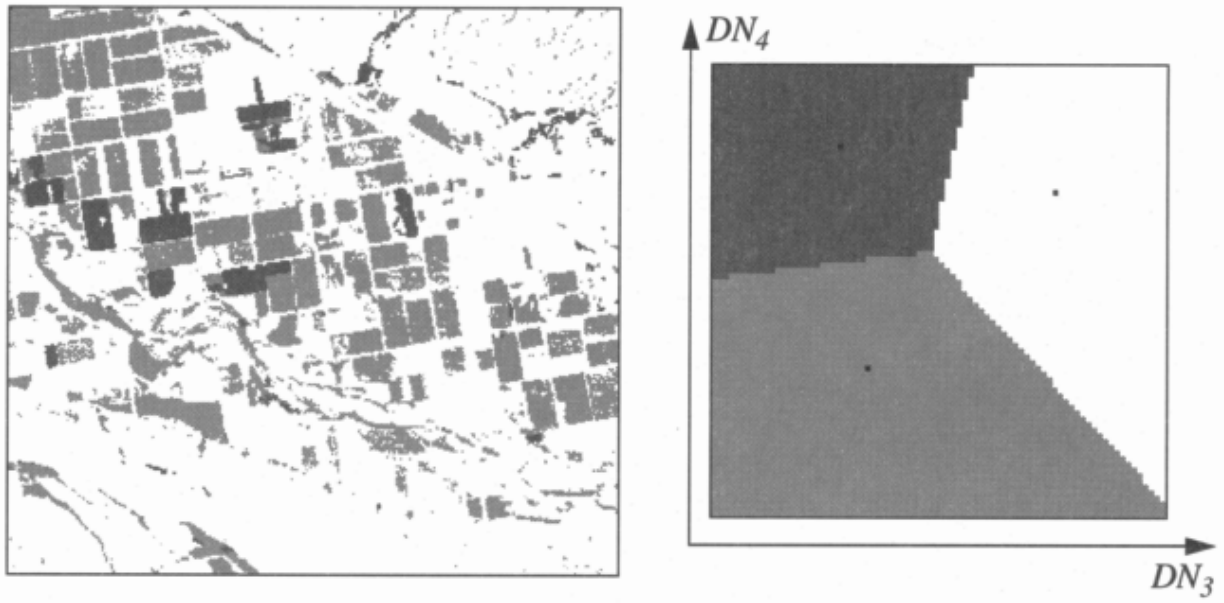


FIGURE 9-25. Nearest-mean classification results in image and feature space.

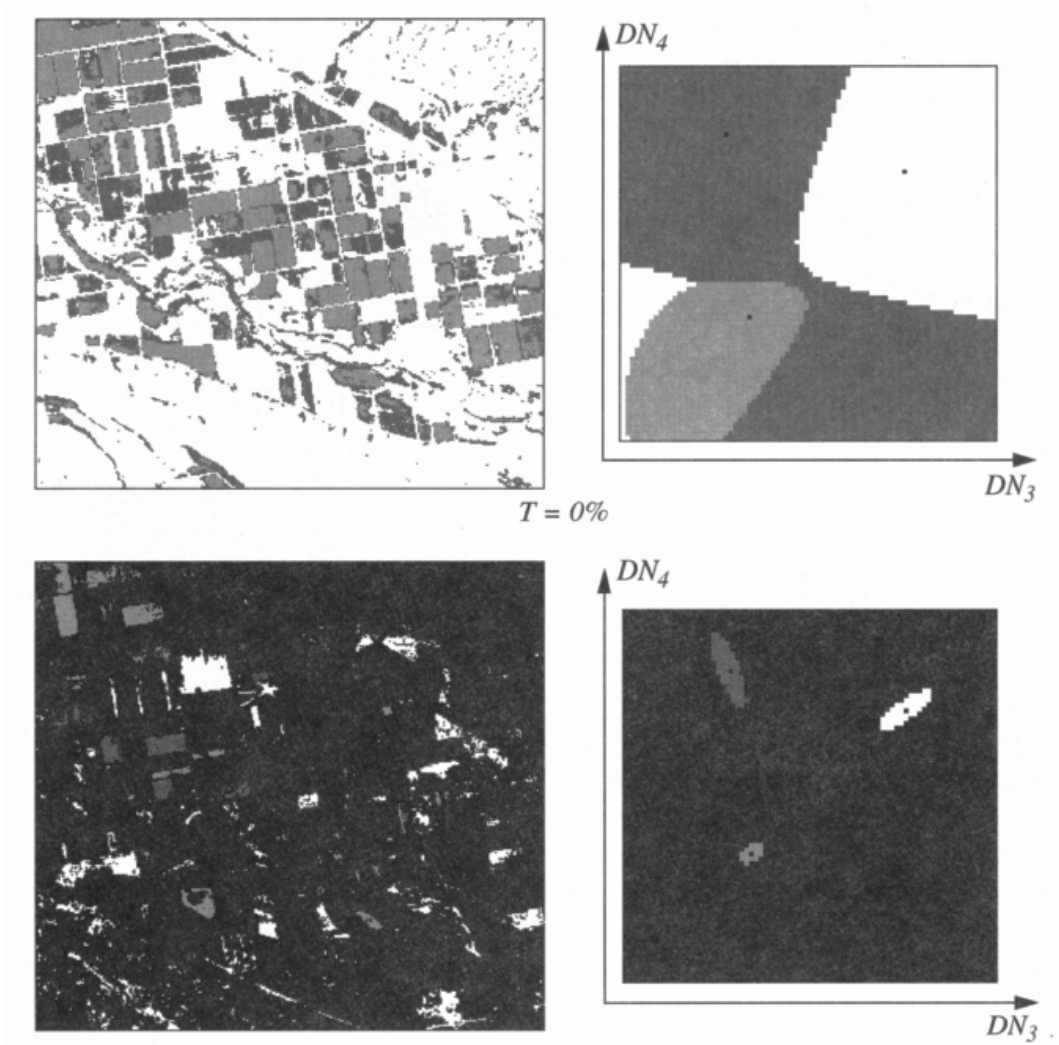
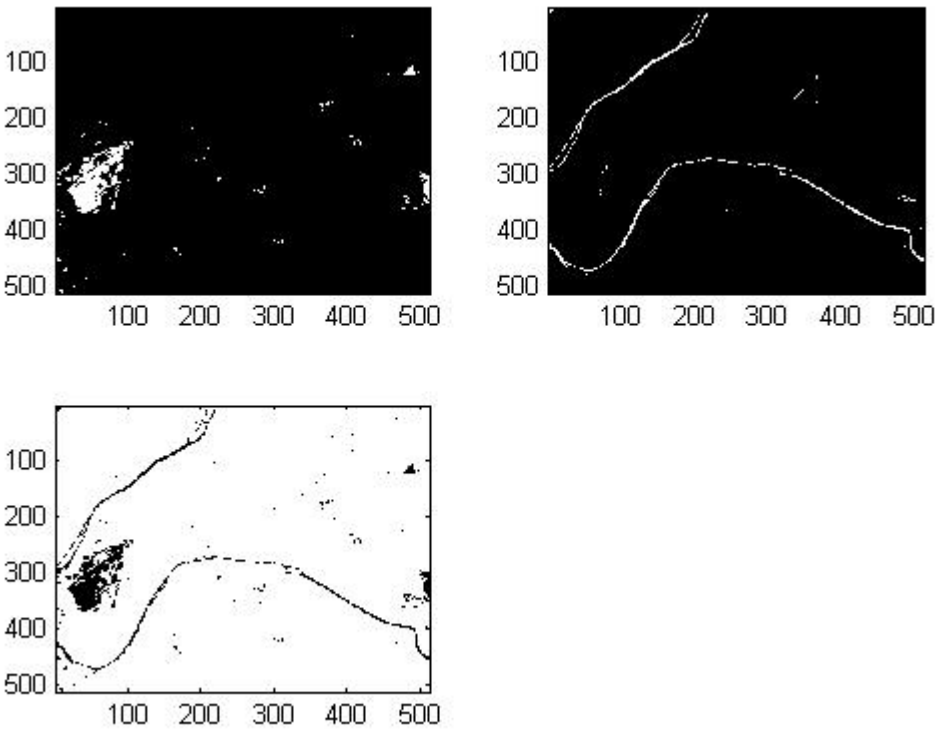


FIGURE 9-26. Maximum-likelihood classification results in image and feature space. Note the disjoint decision regions for the class "light soil" if no probability threshold is used. A threshold limits the distance from the mean. The threshold is specified as the percentage excluded volume under the Gaussian model distributions of each class. The fact that a low threshold of 1% results in such a large population of unlabeled pixels implies that the distributions for each class are highly localized near their means.

監督性次像素分類法

最大似然法




```
C1=(reshape(Cub(318:322,51:55,:),25,7))'*(
      reshape(Cub(318:322,51:55,:),25,7))/25
      -m1*m1';
C2=(reshape(Cub(279:281,265:270,:),18,7))
      '*(reshape(Cub(279:281,265:270,:),18,7
      ))/18-m2*m2';
C3=(reshape(Cub(225:230,291:300,:),60,7))
      '*(reshape(Cub(225:230,291:300,:),60,7
      ))/60-m3*m3';

for i=1:512
for j=1:512
y(3)=1/sqrt((2*pi)^7*det(C3))*exp(-
      1/2*(shiftdim(Cub(i,j,:))-
      m3)'*inv(C3)*(shiftdim(Cub(i,j,:))-m3));
y(2)=1/sqrt((2*pi)^7*det(C2))*exp(-
      1/2*(shiftdim(Cub(i,j,:))-
      m2)'*inv(C2)*(shiftdim(Cub(i,j,:))-m2));
y(1)=1/sqrt((2*pi)^7*det(C1))*exp(-
      1/2*(shiftdim(Cub(i,j,:))-
      m1)'*inv(C1)*(shiftdim(Cub(i,j,:))-m1));
abd(i,j,:)=y/sum(y);
end
end
for i=1:3
   subplot(2,2,i)
   imagesc(abd(:,:,i))
end
```

非監督性次像素分類法

Endmember choice

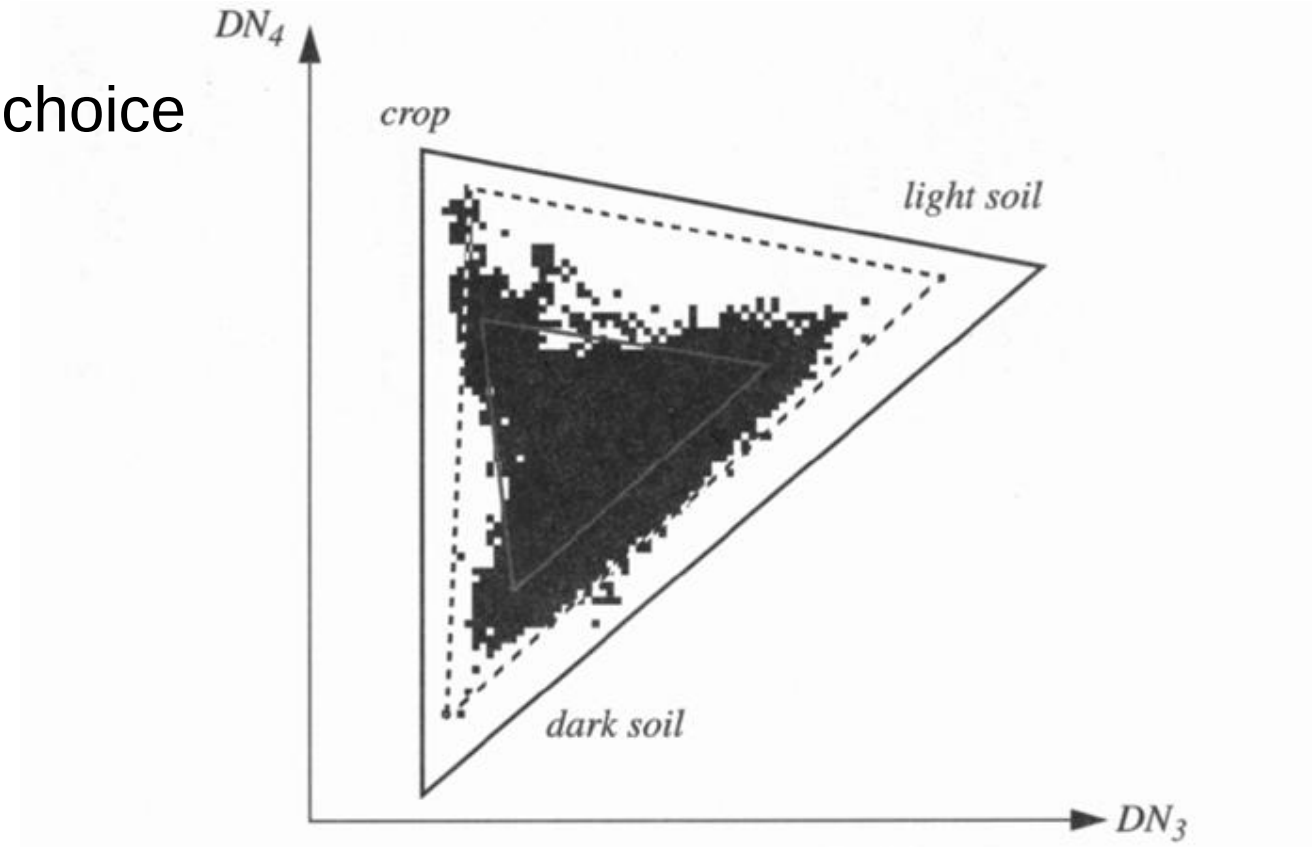


FIGURE 9-42. Three possible choices for endmembers for the classes "dark soil," "light soil," and "crop." The inner triangle is defined by the supervised class means used earlier. The middle triangle is defined by extreme pixels along each DN axis. The outer triangle contains all the pixels, but the endmembers are not actually present in the image. Only the outer triangle is consistent with Eq. (9 – 26).

非監督性次像素分類法

Unsupervised Remote Sensing Method

Step 1 Search Endmember

Step 2 Unmixing (Least square error)

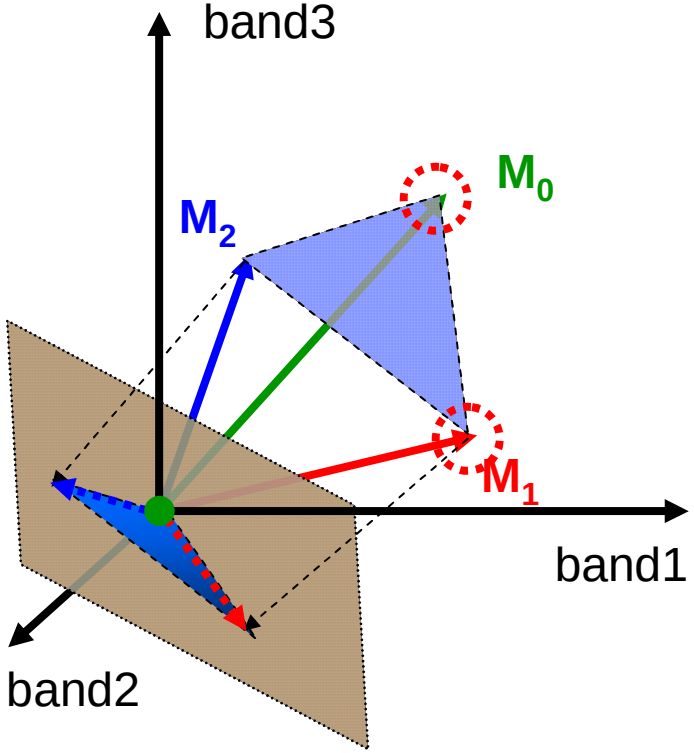
Choose largest as M_0

$$P_{M_0}^{\perp} = I - M_0 \cdot (M_0^T \cdot M_0)^{-1} \cdot M_0^T$$

Choose largest as M_1

$$P_{M_0M_1}^{\perp} = I - [M_0 M_1] \cdot ([M_0 M_1]^T \cdot [M_0 M_1])^{-1} \cdot [M_0 M_1]^T$$

$P_{M_0}^{\perp}$



Least square error $\hat{\alpha}_{LS} = \left(M^T M\right)^{-1} M^T r$

變異目標偵測

- RX-Algorithm
- High-order statistical method

RX Algorithm

$$\delta_{RX}(r) = (r-\mu)^T K_{L*L}^{-1} (r-\mu)$$

L : the number of spectral bands

r : pixel vector in an image

μ : the sample mean $[\mu = \frac{1}{N}\sum_{i=1}^{N} r_i]$

K_{L*L} : the sample covariance matrix $[\frac{1}{N}\sum_{i=1}^{N}(r_i-\mu)(r_i-\mu)^T]$

Whitening

$$\delta_{RXD}(r) = (r-\mu)^T K_{L*L}^{-1}(r-\mu)$$

$$= (r-\mu)^T A\,\Lambda^{-1}A^T(r-\mu)$$

$$= (r-\mu)^T A\,\Lambda^{-1/2}\Lambda^{-1/2}A^T(r-\mu)$$

:

$$= [s*(r-\mu)]^T[(s*(r-\mu)]$$

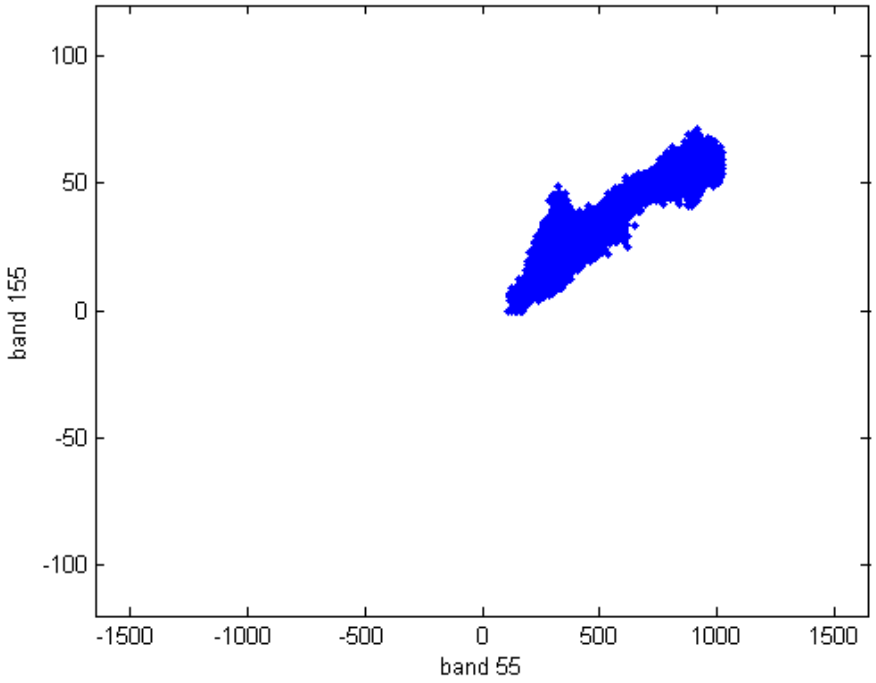
$$(s = \Lambda^{-1/2}A^T)$$

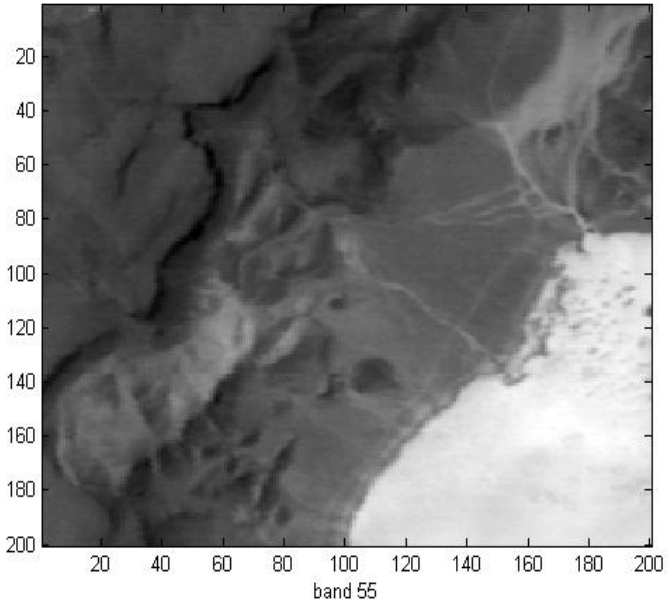
s : whitening matrix

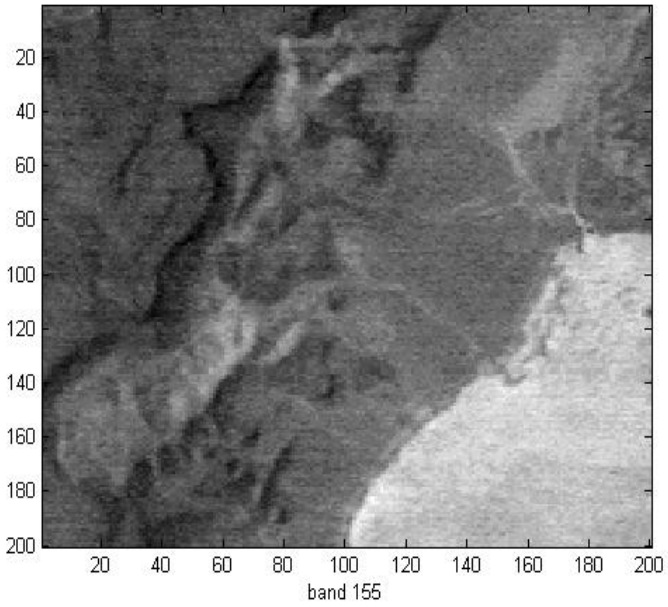
A : eigenvector matrix

Λ : eigenvalue matrix

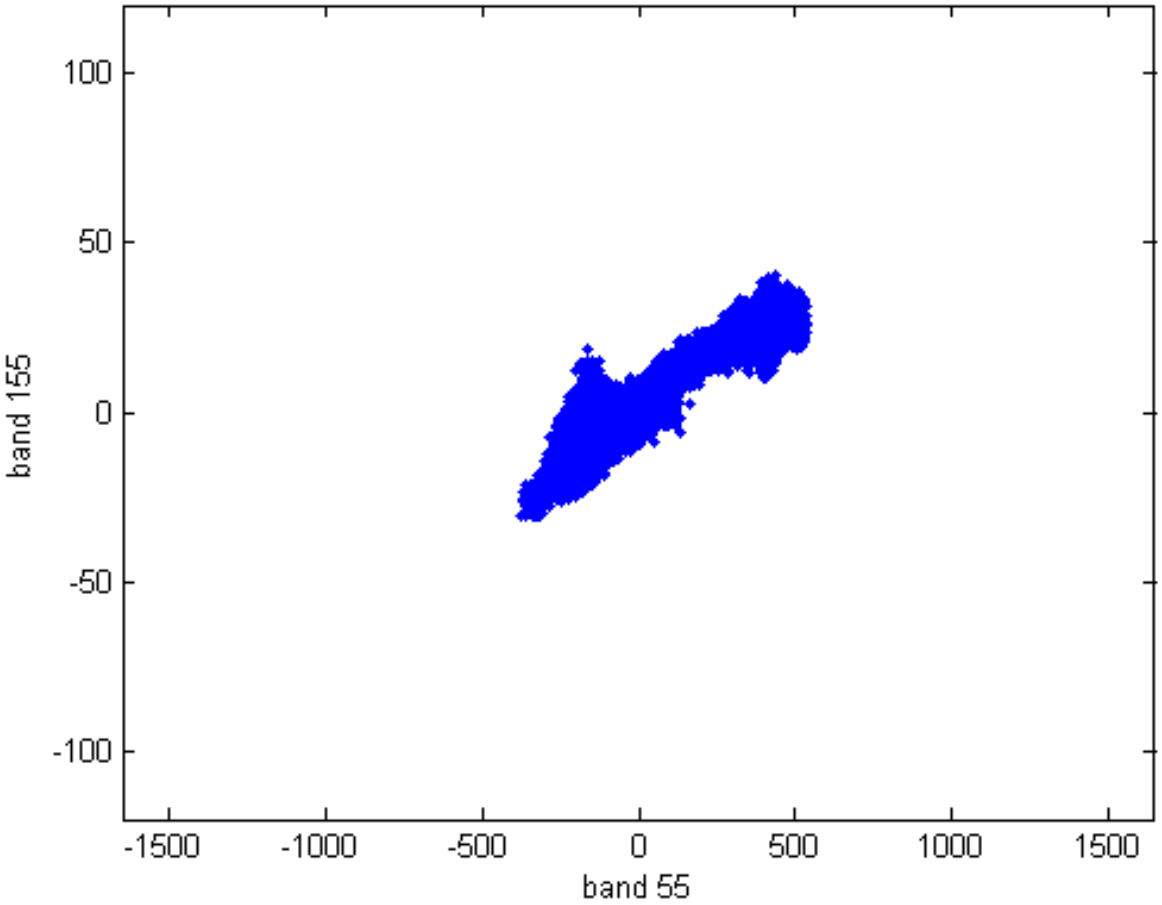
Original distribution





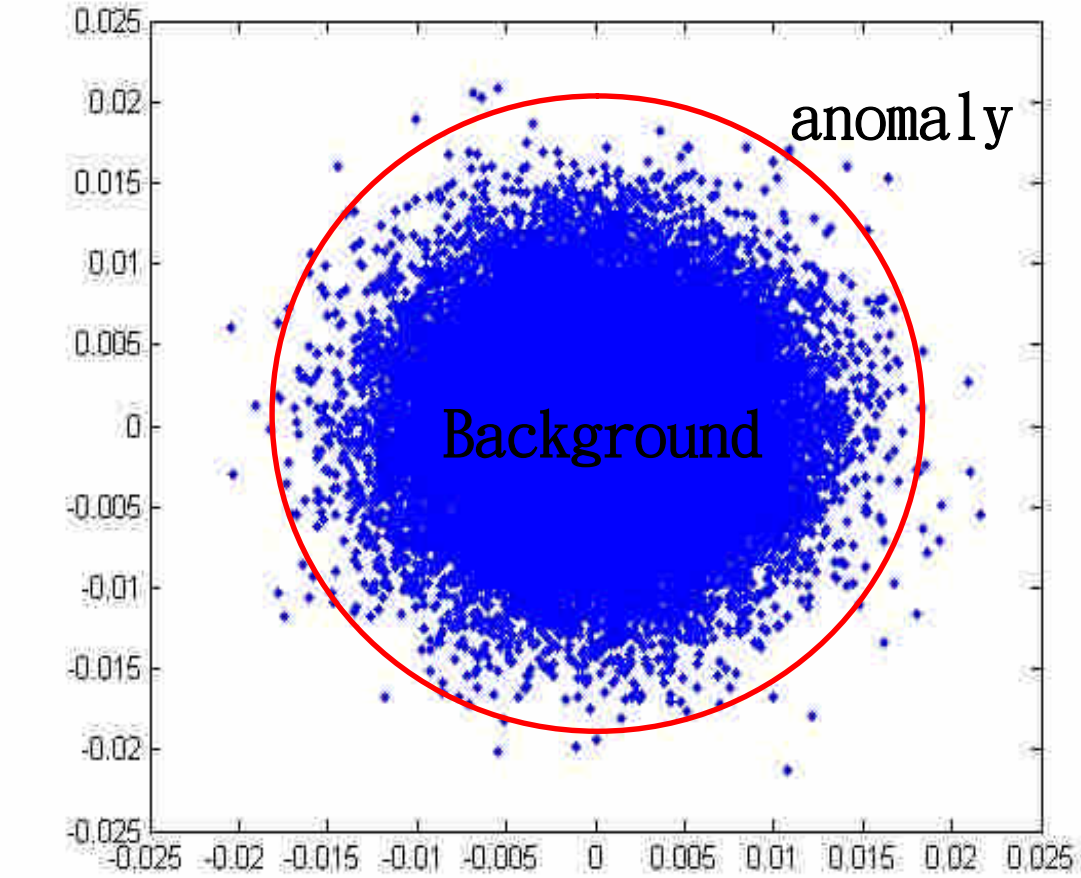


Centralize



Whitening

$$\delta_{RXD}(r) = [s*(r-\mu)]^T [s*(r-\mu)]$$

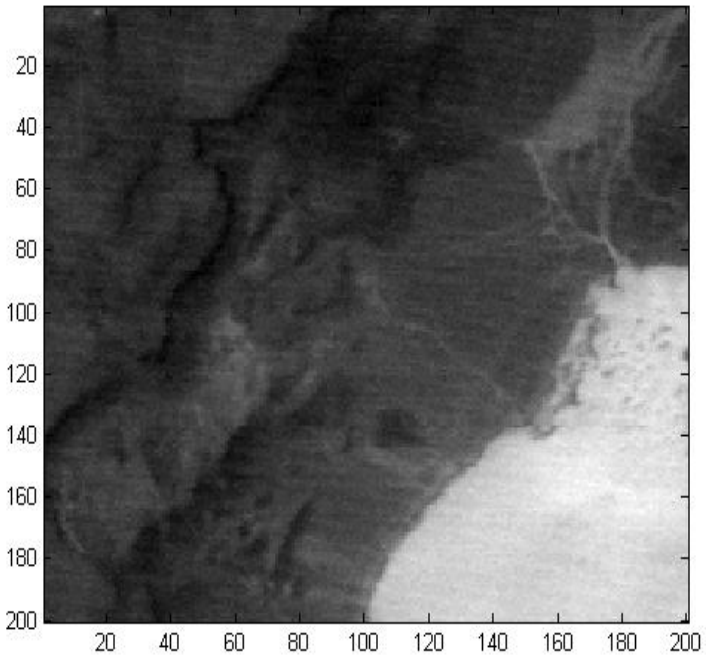


Matlab code (RX Algorithm)

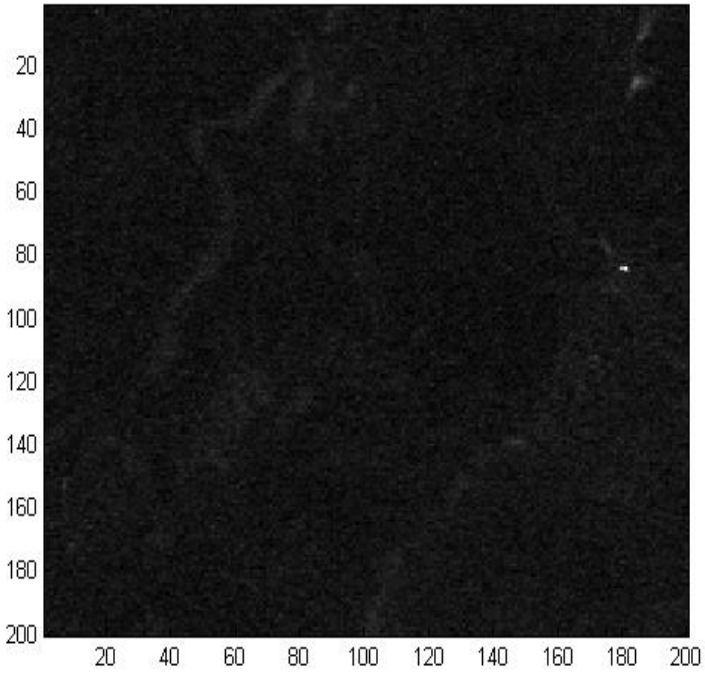
```
clear all; %清除變數
load ImageCub; % 載入高光譜影像資料，以AVIRIS LCVF爲例
data=reshape(ImageCub,40000,158); %將資料重新排列
data_mean=mean(data); %求平均值
data=data-repmat(data_mean,40000,1); %將資料減去平均值
cov_m=data‘*data; %計算covariance matrix

cm_inv=inv(cov_m); % 計算inverse of covariance matrix
%  依序將每一個像素代入RX算式中
for i=1:40000;
    maha(1,i)=data(i,:)*cm_inv*data(i,:)';
end
imagesc(reshape(maha,200,200)),colormap(gray);
% 將處理好的資料顯示出來
```

Result of RX algorithm



Original image



result

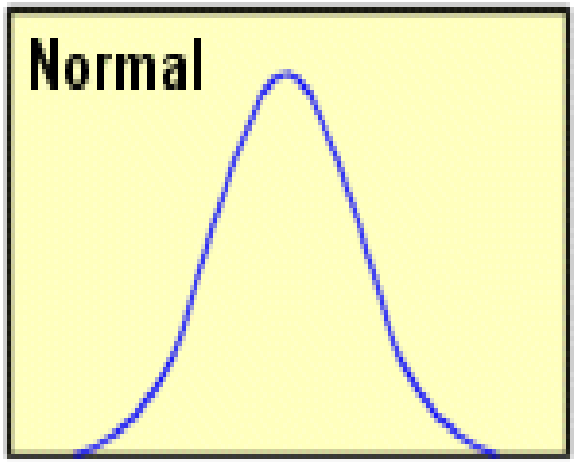
Detection by the high-order statistical method

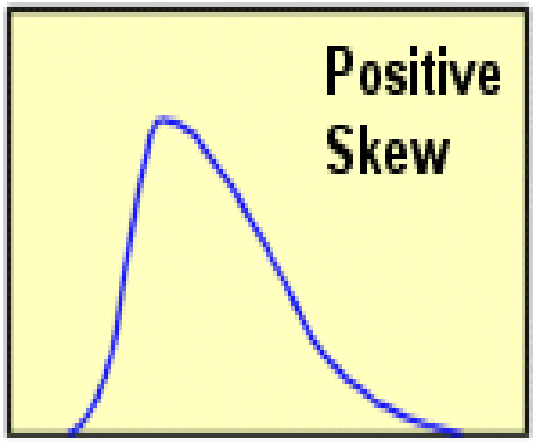
skewness： $\kappa_3(y) = \frac{E[(y-\mu)^3]}{\sigma^3}$

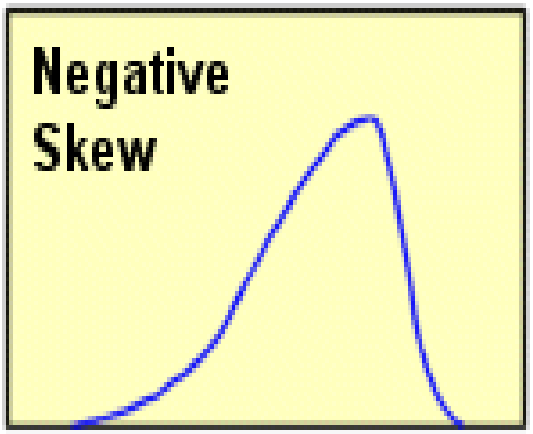
kurtosis： $\kappa_4(y) = \frac{E[(y-\mu)^4]}{\sigma^4}$

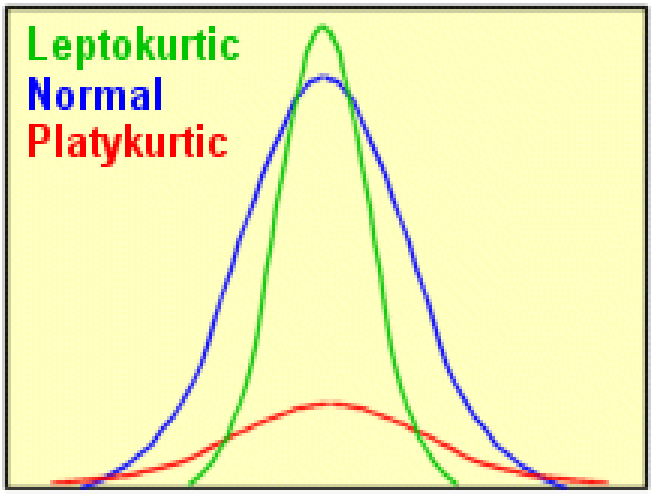
measure the asymmetry and the flatness of the data distribution

Distribution









Projection Searching Process

To find the projector which yields the maximum skewness, we impose a constrained problem as：

$$\max_{\mathbf{w}} \left\{ \frac{1}{N} \sum_{i=1}^{N} z_i^3 \right\} = \max_{\mathbf{w}} \left\{ \frac{1}{N} \sum_{i=1}^{N} \mathbf{w}^T \mathbf{y}_i \mathbf{y}_i^T \mathbf{w} \mathbf{w}^T \mathbf{y}_i \right\} \text{ subject to } \mathbf{w}^T \mathbf{w} = 1$$

Using Lagrange multiplier method, we construct the objective function as：

$$J(\mathbf{w}) = E[\mathbf{w}^T \mathbf{y}_i \mathbf{y}_i^T \mathbf{w} \mathbf{w}^T \mathbf{y}_i] - \lambda(\mathbf{w}^T \mathbf{w} - 1)$$

Partially differentiate the above equation with regard to **w** results in

$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = 3E[\mathbf{y}_i \mathbf{y}_i^T \mathbf{w} \mathbf{y}_i^T]\mathbf{w} - 2\lambda \mathbf{w} = 0$$

Denote $\lambda' = \frac{2}{3}\lambda$ ，

$$(E[\mathbf{y}_i \mathbf{y}_i^T \mathbf{w} \mathbf{y}_i^T] - \lambda' \mathbf{I})\mathbf{w} = 0$$

Since above equation is an eigen-problem，the solution $\mathbf{w}^*$ is the eigenvector of the matrix

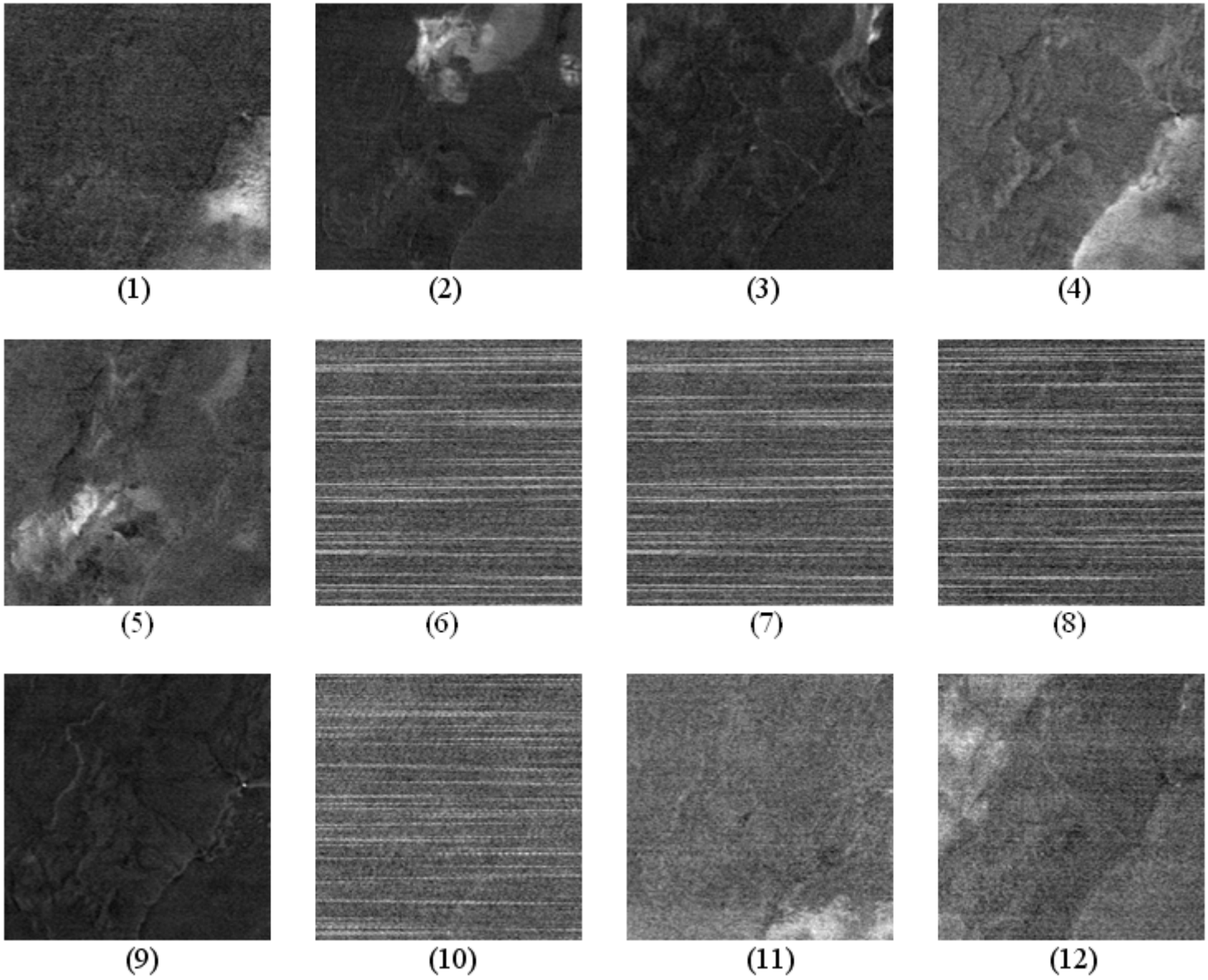
$$E[\mathbf{y}_i \mathbf{y}_i^T \mathbf{w} \mathbf{y}_i^T]$$

Algorithm

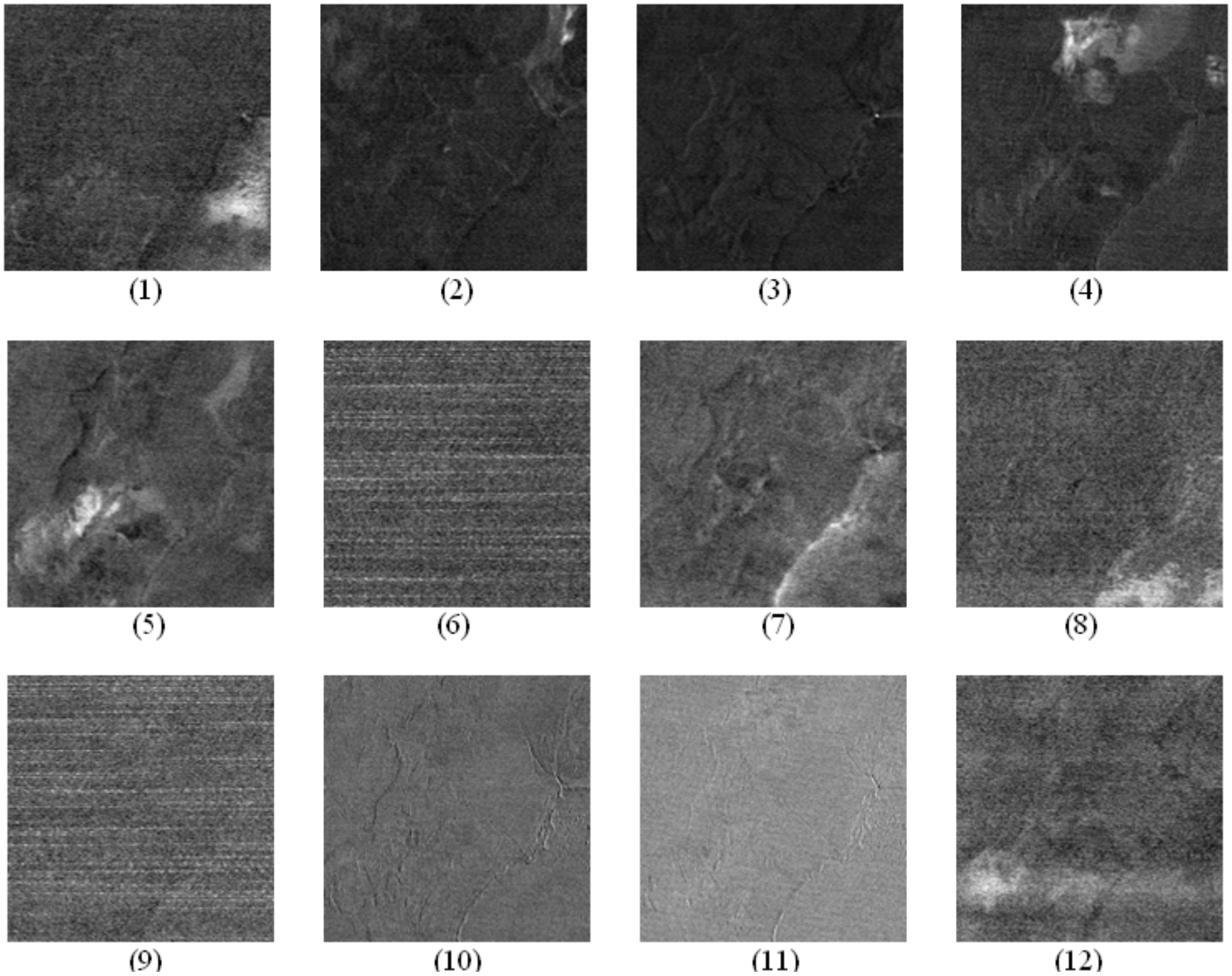
1. Centralize and decorrelate the original data set **X** by whitening. The resulting data set is **Y**.
2. Initialize the first projector $\mathbf{w} = \mathbf{w}_1$.
3. Calculate the matrix $E[\mathbf{y}_i \mathbf{y}_i^T \mathbf{w}_1 \mathbf{y}_i^T]$ if the searching criterion is maximizing skewness, or $E[\mathbf{y}_i \mathbf{y}_i^T \mathbf{w}_1 \mathbf{w}_1^T \mathbf{y}_i \mathbf{y}_i^T]$ if the searching criterion is maximizing kurtosis.
4. Find the eigenvector **v** of $E[\mathbf{y}_i \mathbf{y}_i^T \mathbf{w}_1 \mathbf{y}_i^T]$, or $E[\mathbf{y}_i \mathbf{y}_i^T \mathbf{w}_1 \mathbf{w}_1^T \mathbf{y}_i \mathbf{y}_i^T]$ for kurtosis which corresponds to the maximum eigenvalue.
5. If $\mathbf{w}_1 = \mathbf{v}$, then $\mathbf{w}_1$ is the desired projector $\mathbf{w}_1^*$ and go to step 6. Otherwise, replace $\mathbf{w}_1$ with **v**, and go to step 3.
6. Applying the $\mathbf{w}_1^*$ to the decorrelated data matrix **Y**, i.e., $\mathbf{z} = \left(\mathbf{w}_1^*\right)^T \mathbf{Y}$ can detect the first anomaly.

7. Construct the orthogonal subspace projector with $\mathbf{P}_{\mathbf{W}}^{\perp} = \mathbf{I} - \mathbf{W}(\mathbf{W}^T\mathbf{W})^{-1}\mathbf{W}^T$ where $\mathbf{W} = [\mathbf{w}_1]$ and project the $\mathbf{Y}$ onto the orthogonal subspace of $\mathbf{W}$ with $\mathbf{Y}' = \mathbf{P}_{\mathbf{w}}^{\perp}\mathbf{Y}$. Go to the step 2 to find a second projector with regard to $\mathbf{Y}'$ and detect a second anomaly.
8. After the k-th projector is determined, construct the $\mathbf{P}_{\mathbf{w}}^{\perp}$ with $\mathbf{W} = [\mathbf{w}_1\mathbf{w}_2\cdots\mathbf{w}_k]$ and calculate $\mathbf{Y}' = \mathbf{P}_{\mathbf{w}}^{\perp}\mathbf{Y}$ for the (k+1)-th projector and the (k+1)-th anomaly. The algorithm is stopped if the skewness or kurtosis of $\mathbf{z}$ is less than a prescribed threshold.

Detecting anomalies using skewness



Detecting anomalies using kurtosis



References

[1] Irving S. Reed and Xiaoli Yu, "Adaptive Multiple-Band CFAR Detection of an Optical Pattern with Unknown Spectral Distribution," IEEE transactions on acoustics. speech. and signal processing, vol.38, no.10, pp. 1760-1770, 1990.

[2] Chein-I Chang and Shao-Shan Chiang, "Anomaly Detection and Classification for Hyperspectral Imagery," IEEE transactions on geoscience and remote sensing, vol.40, no.6, pp. 1314-1325, 2002.

[3] Hsuan Ren, Qian Du, and James Jensen, "Efficient anomaly detection and discrimination for hyperspectral imagery," Proc. SPIE, vol.4725,pp. 234-241, 2002.

[4] http://cas.bellarmine.edu/tietjen/Stats/ResMethodsIntro2Stats.htm

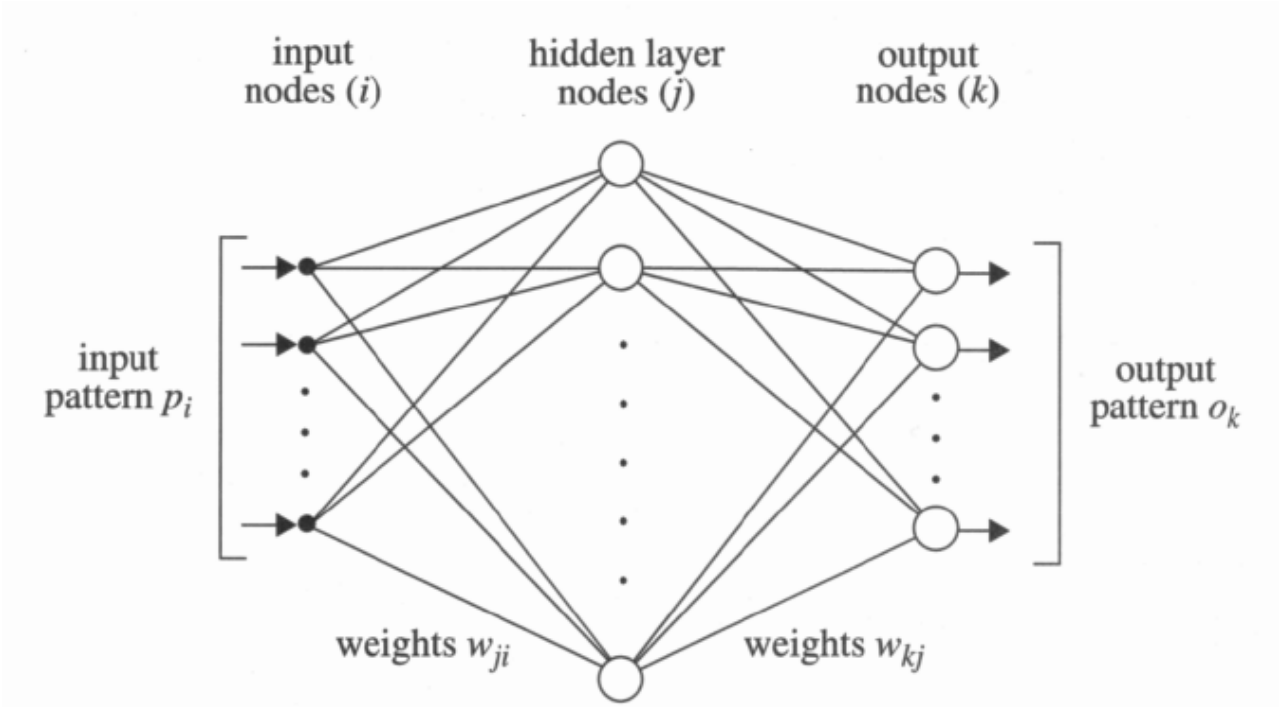


FIGURE 9-13. The traditional structure of a three-layer ANN.

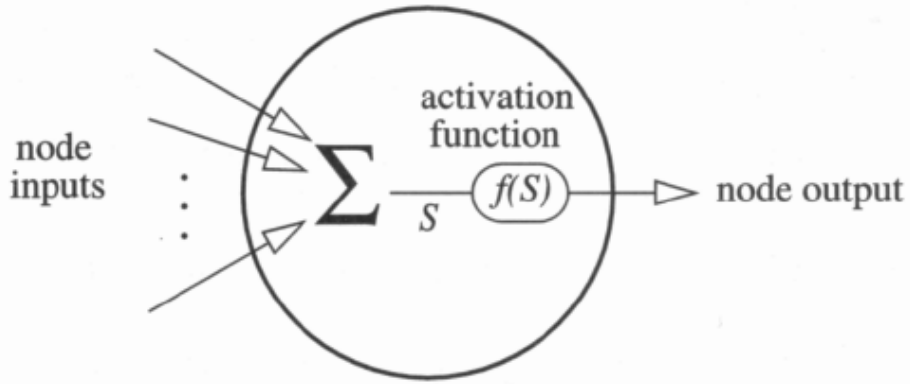


FIGURE 9-14. The components of a processing element.

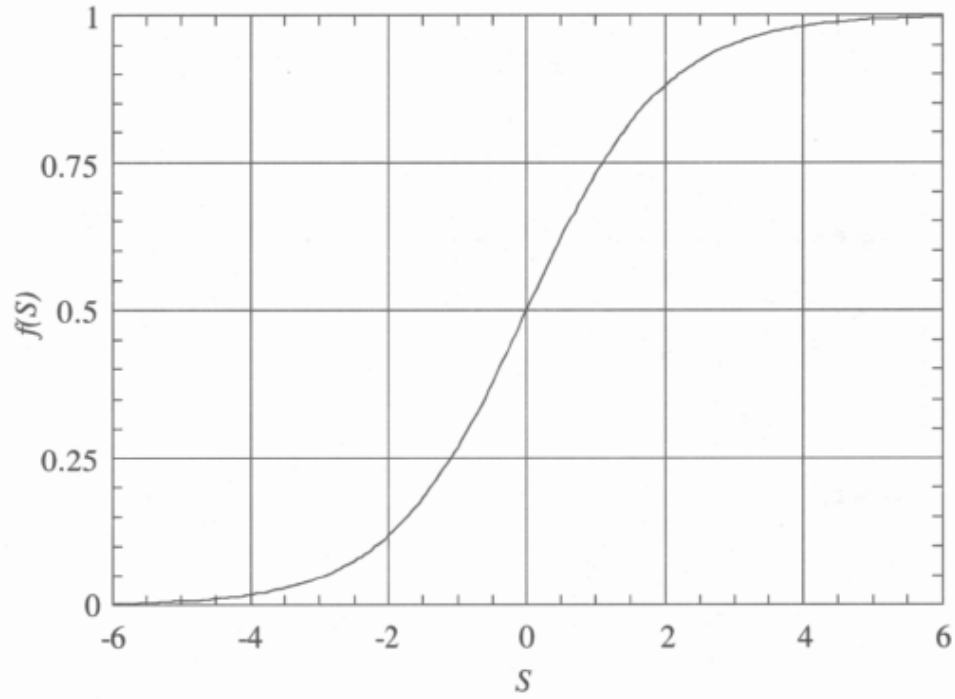


FIGURE 9-15. The sigmoid activation function.

1. Select training pixels for each class and specify the desired output vector $\boldsymbol{d}_k$ for class k, typically $d_m = 0.9$ $(m = k)$ and $d_m = 0.1$ $(m \neq k)$. These are the target values that the ANN attempts to produce.

2. Initialize weights as random numbers between 0 and 1 (typically small values near zero are used).

3. Set the frequency for weight updating to one of the following:

   - after every training pixel (sequential)
   - after all training pixels in each class
   - after all training pixels in all classes (batch)

   Batch training is commonly used as it minimizes the frequency of weight updates; the remainder of the algorithm below assumes batch training.

4. Propagate the training data forward through the net, one pixel at a time.

5. After each training pixel is propagated forward through the network, calculate the output $\boldsymbol{o}$ and accumulate the total error relative to the desired output $\boldsymbol{d}$ (the 1/2 factor is a mathematical convenience),

$$\frac{\|\varepsilon\|^2}{2} = \frac{1}{2}\sum_{p=1}^{P}\sum_{k}(d_k - o_k)^2 \qquad (9-5)$$

   Repeat over all P training patterns (pixels) for batch training.

6. After all training pixels have been used, adjust the weight w_{kj} by,

$$\begin{aligned}\Delta w_{kj} &= LR\frac{\partial \varepsilon}{\partial w_{kj}} \\ &= LR\sum_{p=1}^{P}(d_k - o_k)\frac{d}{dS}f(S)\Big|_{S_k} h_j\end{aligned} \qquad (9-6)$$

   where LR is a *Learning Rate* parameter used to control the speed of convergence.

7. Adjust the weight w_{ji} by,

$$\Delta w_{ji} = LR\sum_{p=1}^{P}\left\{\frac{d}{dS}f(S)\Big|_{S_j}\sum_{k}\left[(d_k - o_k)\frac{d}{dS}f(S)\Big|_{S_k} w_{kj}\right]p_i\right\} \qquad (9-7)$$

8. Repeat steps 4 through 7 until $\varepsilon <$ threshold.

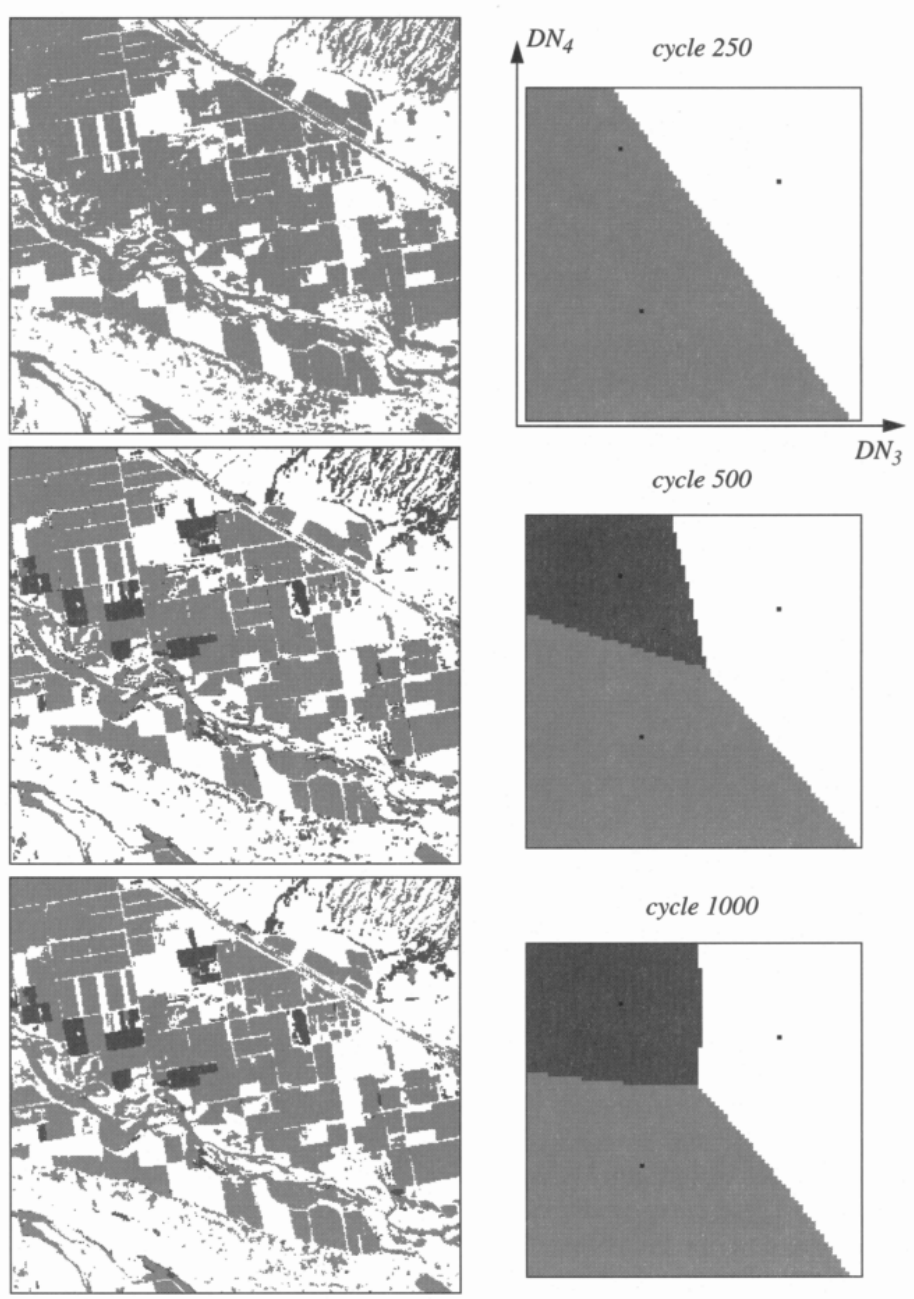


FIGURE 9-19. Hard maps and decision boundaries at three stages of the back-propagation algorithm. (The processing and results in this section were provided by Justin Paola, Oasis Research Center.)

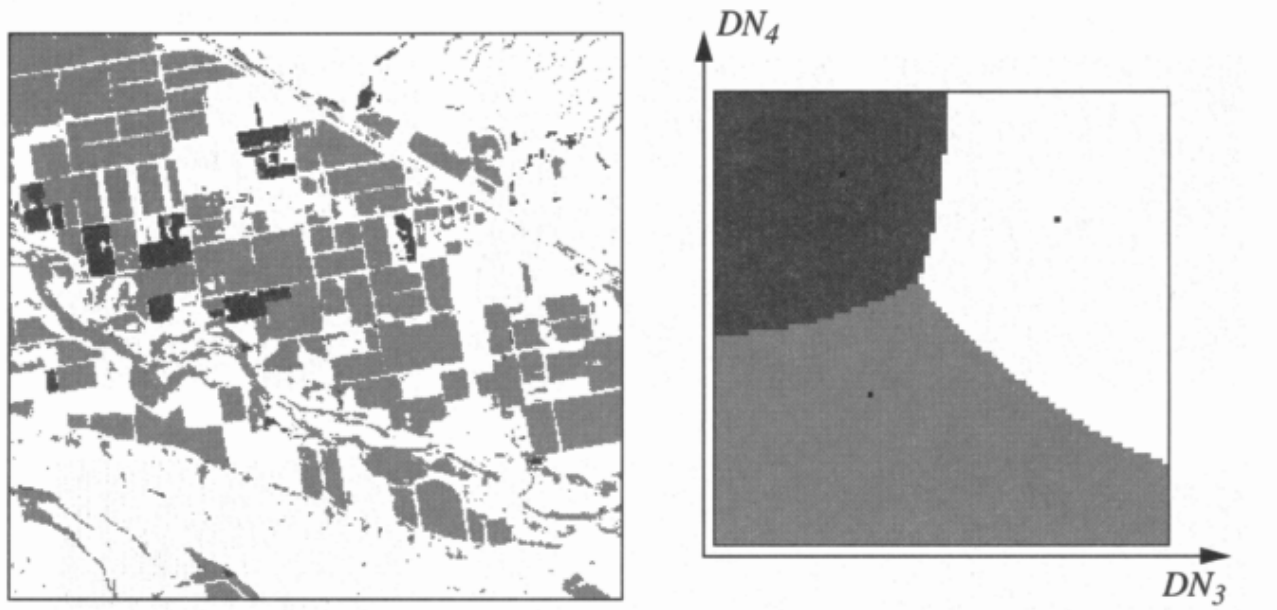


FIGURE 9-20. Final ANN hard classification map and decision boundaries at 5000 cycles.

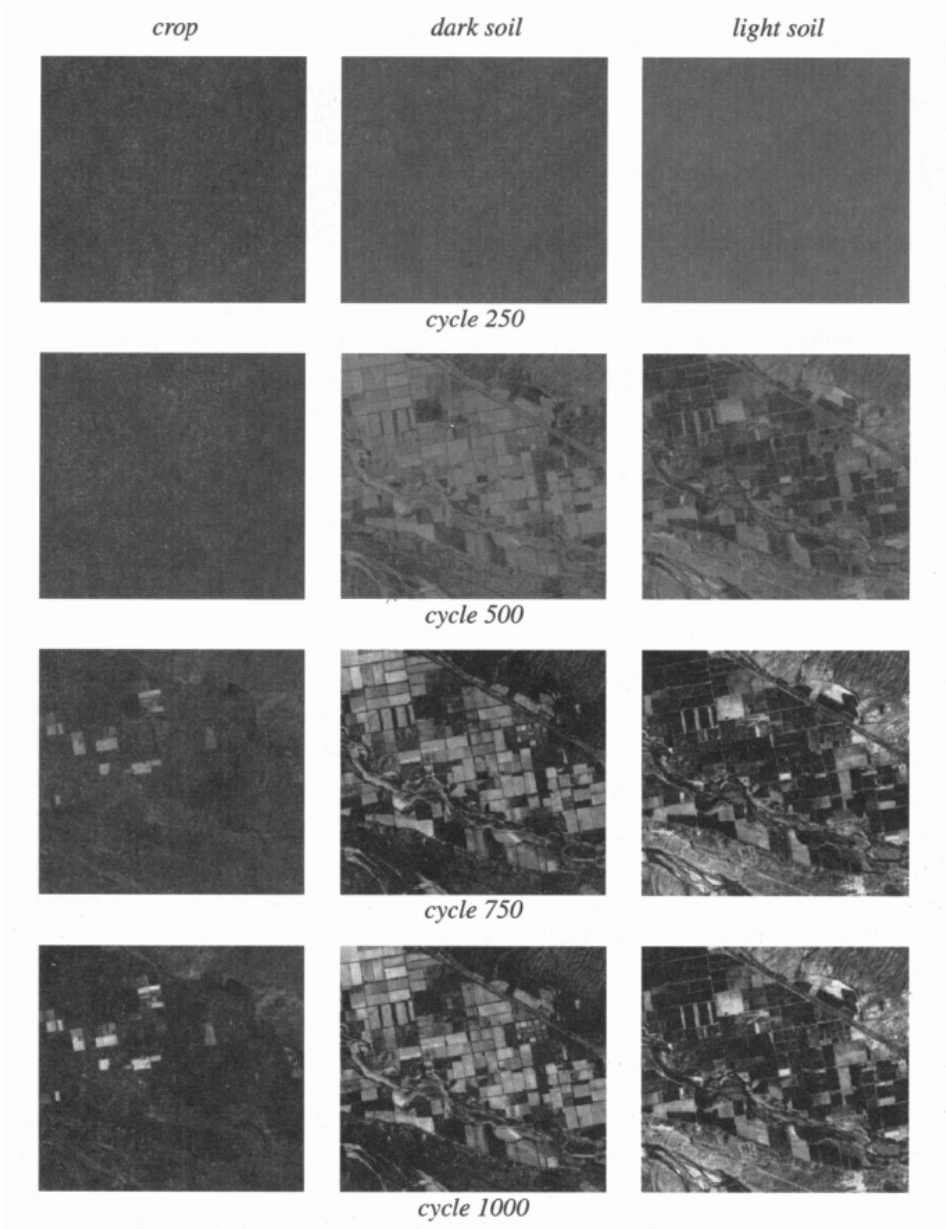


FIGURE 9-21. Soft classification maps at four stages of the back-propagation training process. The greylevel in each map is proportional to the class output-node value and all maps are scaled by the same LUT, so direct brightness comparisions are possible. Notice how the output-node values for all three classes increase as more training cycles are used.

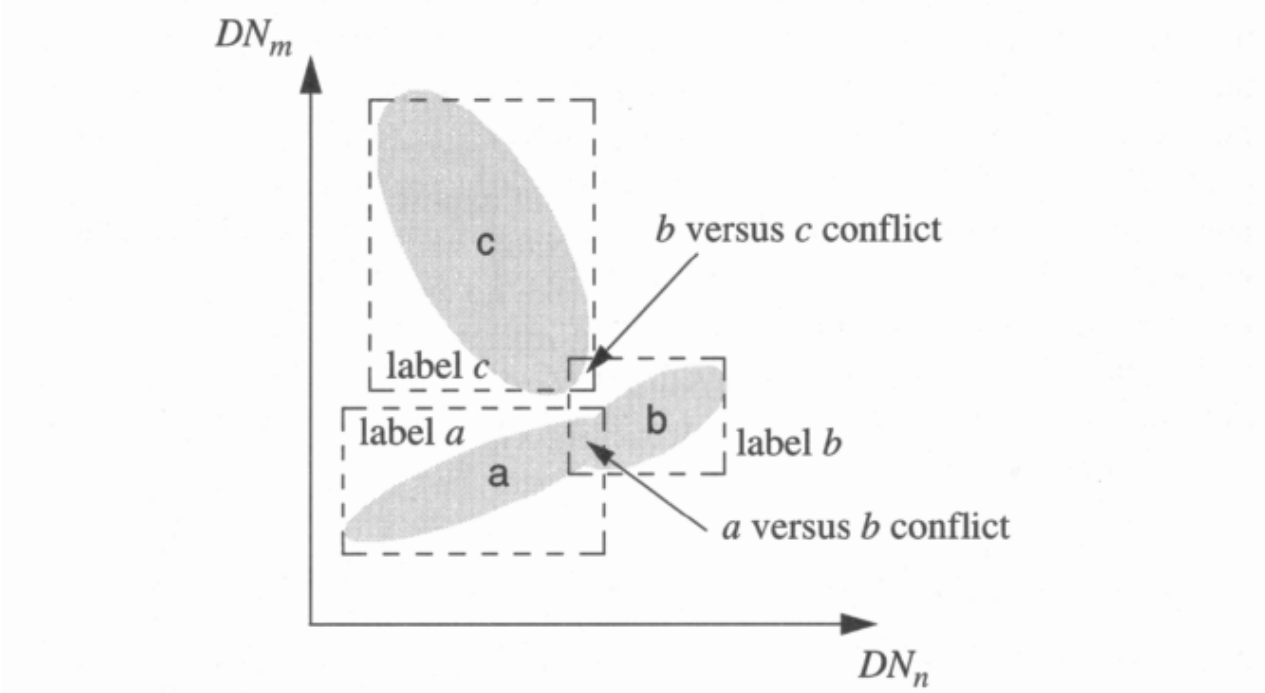


FIGURE 9-12. Level-slice decision boundaries for three classes in two dimensions. Pixel labeling within the conflict regions must be handled with an alternate decision, for example by the nearest-mean classifier.