Chapter 3

IMPLEMENTATION OF PARALLELEPIPED CLASSIFICATION ALGORITHM

3.1. Introduction

This chapter examines implementing the parallelepiped classification technique for location-wise classification. Location-wise classification technique is used to find the location of the image where large amount of resources are available. In this location-wise classification algorithm the parallelepiped classification technique is used for classifying the images after the location is identified.

Parallelepiped classification technique is one of the techniques used for classifying images. Among the various techniques used for classifying images, parallelepiped classification is used for classifying images based on the range of pixel value. In parallelepiped classification technique initially the mean value for all classes of image is calculated in each band of data. Then the cluster boundary value is calculated to all bands of data for the mean class values. The cluster boundary value is the minimum difference between all mean classes of data in each band. The range of pixels for classification is calculated based on the cluster boundary value.
Selecting pixels one by one from the image for which the classification process is needed, forms the first step of classification. First, it has to be found whether the pixel value comes under the cluster boundary range of the specified class. If the pixel comes under the particular class then the specified pixel value has to be highlighted and increase the count of the number of pixels that fall under this class has to be increased.

3.2. Background

The parallelepiped classifier (often termed multi-level slicing) divides each axis of multi-spectral feature space. The decision region for each class is defined on the basis of the lowest and the highest values on each axis. The accuracy of classification depends on the selection of the lowest and highest values considering the population statistics of each class. In this respect, it is most important that the distribution of the population of each class is well understood.

The method of parallelepiped classification, sometimes referred to as box decision rule or level slice procedures, is based on the ranges of values within the training data to define decision boundaries within a multidimensional data space [STSN2003]. The spectral values of unclassified pixels are projected into data space and those falling within the regions defined by the training data are assigned to the appropriate categories. The dimensions of the parallelepiped are
usually defined based upon a standard deviation threshold from the mean of each selected class.

Parallelepiped classification uses a simple decision rule to classify multispectral data. The decision boundaries form an n-dimensional parallelepiped in the image data space. The dimensions of the parallelepiped are defined based upon a standard deviation threshold from the mean of each selected class.

The parallelepiped classifier uses the threshold of each class signature to determine if a given pixel falls within the class or not [PCIG1999]. The threshold specifies the dimensions (in standard deviation units) of each side of a parallelepiped surrounding the mean of the class in feature space. If the pixel falls inside the parallelepiped, it is assigned to the class. However, if the pixel falls within more than one class, it is put in the overlap class (code 255). If the pixel does not fall inside any class, it is assigned to the null class (code 0). The parallelepiped classifier is typically used when speed is required.

The parallelepiped classifier or a scheme based upon a known absorption feature for a specific material. Classify the data samples on an absolute basis without regard to the spectral responses of other materials or classes that may be in the scene. Therefore design of such a classifier requires class definition through training samples only for the particular class under consideration [BJDL1996].
Parallelepiped classification calculates an $n$-dimensional parallelepiped for an image with $n$ bands (e.g. for 3 bands, this would be a 3-dimensional cube) with the boundaries being one standard deviation away from the mean. When the classifier is applied to the image, those pixels falling within a particular class's parallelepiped are assigned to that class. Those pixels, which do not lie within any parallelepipeds, are assigned to an unclassified category. However, in the case where two or more parallelepipeds overlap and the pixel falls within more than one, it is assigned to the first class to which it is compared [JS2000].

3.3. Description

The number of pixels used for classification is initialized. It is the product of resolution of x and y co-ordinates. The number of bands used in this classification technique is initialized as three. In this, the processing bands refer to red, green or blue. The number of class of images used for classification is initialized as three. The mean values for the three classes of images are assigned. For each class of images three bands of mean values are calculated. The mean values are obtained by summing up all the pixel information for a band and dividing by the number of pixels. This mean value is calculated for each band of data. For example the variable mclass[1][2] refers to the mean value for the band 2 (green) of class 1 image. Similarly the mean value for different bands for different classes is assigned.
The cluster boundary for each band of data is calculated. The cluster boundary is the smallest difference between different classes. This cluster boundary plays a vital role in assigning a pixel to a particular class. Initially the difference between various bands of pixels of class 1 and class 2 is found. Then the difference between various bands of pixels of class 1 and class 3 is found. Then the difference between various bands of pixels of class 2 and class 3 is found. The smallest difference is selected and assigned for each band as the cluster difference. The range of pixel value selected for classification is calculated based on the cluster difference. Two limits are selected for the range, one is the low limit and the other is the high limit. This range is calculated for all bands of data.

The row value and the column value which indicate the location where classification is necessary are used to find the beginning and end of row value and the beginning and end of column value for the purpose of classification. If the beginning of row is less than zeros (top boundary of image) the row value is initialized as zero. Similarly if the beginning of column value is less than zeros (left boundary of the image) the column value is initialized as zero. If the end of row is greater than or equal to y resolution it is initialized as one less than the y resolution. Similarly if the end of column is greater than or equal to x resolution it is initialized as one less than the x resolution. Only for the co-ordinates of row and column beginning and row and column ending, the
classification is performed. The classification is based on the range values. From the original image, certain numbers of pixels that fall on the particular class are printed.

3.4. Flow Diagram

START

Assign the number of pixels, the number of bands, the number of classes and the mean value for various bands of each class.

Initialize the cluster boundary to a high value.

Find the cluster boundary value for different bands of data.

A
Calculate the pixel range based on the cluster boundary for assigning a pixel to a particular class.

Assign the beginning of row and column as well as the end of row and column for indicating the location for classification using the row and column value specified.

Classify the selected area of image using the pixel range.

Print the number of pixels classified under various classes

STOP
3.1. High Level Algorithm

Step 1: Assign the number of pixels, the number of bands, the number of classes and the mean value for various bands of each class.

Step 2: Initialize the cluster boundary to a high value.

Step 3: Find the cluster boundary value for different bands of data.

Step 4: Calculate the pixel range based on the cluster boundary for assigning a pixel to a particular class.

Step 5: Assign the beginning of row and column as well as end of row and column for indicating the location for classification using the row and column value specified (set the boundary location for classification).

Step 6: Classify the selected area of image using the pixel range.

Step 7: Print the number of pixels classified under various classes.
3.6. Algorithm for location-wise classification using the parallelepiped technique

Algorithm locclppel(pixels[], rowval, colval)

// pixels – the one dimensional pixel array that contains the pixel value of an image.
// rowval – the row location for classification.
// colval – the column location for classification.
// npix – number of pixels.
// nband – number of bands.
// nclass - number of classes.
// mclass[i][j] – the mean value of each class.
// i – the class number.
// j – the band number.
// rcb – cluster boundary for red.
// gcb – cluster boundary for green.
// bcb – cluster boundary for blue.
// nofpx(c )- number of pixels classified under class c.
// c – the class number.

1. npix = (xres*(yres+1));
2. nband = 3;
3. nclass = 3;
4. for (c=1; c<=nclass; c++)
5. nofpix[c] = 0;
6. mclass[1][1] = -16796591;
7. mclass[1][2] = -16770740;
8. mclass[1][3] = -16796591;
9. mclass[2][1] = -12945563;
10. mclass[2][2] = -12927899;
11. mclass[2][3] = -12945563;
12. mclass[3][1] = -4541305;
13. mclass[3][2] = -4555326;
14. mclass[3][3] = -4541305;
15. rcb = 99999999;
16. gcb = 99999999;
17. bcb = 99999999;
18. for (c=1; c<=nclass-1; c++)
19. for (tc=c+1; tc<=nclass; tc++)

{ 
20. difr = Math.abs(mclass[c][1]) - Math.abs(mclass[tc][1]);
21. difg = Math.abs(mclass[c][2]) - Math.abs(mclass[tc][2]);
22. dlib = Math.abs(mclass[c][3]) - Math.abs(mclass[tc][3]);
23. if (difr < rcb) rcb = difr;
24. if (difg < gcb) gcb = difg;
25. if (dlib < bcb) bcb = dlib;
 } // close for loop tc

26. rcb /=2;
27. gcb /=2;
28. bcb /=2;
29. System.out.println("class selected is "+clas);
30. rl = mclass[clas][1] -rcb;
31. rh = mclass[clas][1] +rcb;
32. gl = mclass[clas][2] -gcb;
33. gh = mclass[clas][2] +gcb;
34. bl = mclass[clas][3] -bcb;
35. bh = mclass[clas][3] +bcb;
36. rowbegin = rowval-50;
37. rowend = rowval ;
38. colbegin = colval -50;
39. colend = colval ;
40. if (rowbegin <= 0) rowbegin = 0;
41. if (rowend >= yres) rowend = yres-1;
42. if (colbegin <= 0) colbegin = 0;
43. if (colend >= xres) colend = xres-1;
44. k = 0;
45. for(i = rowbegin; i<=rowend;i++)
46. for(j = colbegin; j<=colend;j++)
        {
47. pidx = (j + xres*i);
48. band[k][1] = pixels[pidx];
49. band[k][2] = pixels[pidx + 1];
50. band[k][3] = pixels[pidx + 2];
51. k = k + 1;
}
52. for( p = 0 ; p<npix-1 ; p++)
  {
53.   modipix[p] = pixels[p];
  }
54. for(i = rowbegin; i<=rowend; i++)
55. for(j = colbegin; j<=colend; j++)
  {
56.   pidx = (j + xres*i);
57.   if (((modipix[pidx] > rl) && (modipix[pidx] < rh)) &&
       ((modipix[pidx+1] > gl) && (modipix[pidx+1] < gh)) &&
       ((modipix[pidx+2] > bl) && (modipix[pidx+2] < bh)))
     {
       a.   nofpix[clas]++;
     }
   } // close for loop j
58. System.out.print("pixels under class "+clas);
59. System.out.println(" : "+nofpix[clas]);
3.7. Complexity Analysis

The assignments of data in step 1 and step 2 of the high level algorithm take place in constant time.

The calculation of cluster boundary in step 3 depends upon the number of classes \( c \) selected and the number of bands \( b \) available. So the time is \( O(cb) \).

Calculation of pixel range in step 4 for classification is a constant. The value assigned as the beginning of row and column as well as the end of row and column for indicating the location for classification in step 5 is a constant.

Classifying the selected area of image using the pixel range in step 6 depends on row \( r \) and column \( c \) value selected for classification. So the computing time in this step is \( O(rc) \).

3.8. Merits and Demerits

The decision region for each class is defined on the basis of the lowest and highest value on each axis. The parallelepiped method is a rectangle, where the lowest and the highest pixel values for the class in each band make up the boundary. The parallelepiped classifier uses the threshold of each class signature to determine if a given pixel falls within the class or not.
Parallelepiped is a more accurate method than the minimum distance to means. Restricting the assignment of classes, to only those pixels that fall within one standard deviation of the class mean ensures greater certainty in the final class assignments [JS2000].

The accuracy of classification depends on the selection of the lowest and the highest values in consideration of the population statistics of each class. The drawback is (in many cases) poor accuracy. Unknown classifications result when pixels fall outside all of the parallelepipeds. In many cases this results in poor accuracy and a large number of pixels are classified as ties.

The drawback is that the final output contains pixels which are not assigned to any class. One remedial measure for this situation is to extract the unassigned pixels and classify them using another classifier, such as Maximum Likelihood method [JS2000].
3.7. Sample Result

3.9.1. Description of Results

Figure 3.9.1.1 shows the result of implementing the parallelepiped classification technique for location-wise classification.

![Original Image](image1) ![Classified under class 1 at 100,100](image2)

Figure 3.9.1.1. Implementation of Parallelepiped classification

Note: 1) Class 1 denotes the black area.
2) 100,100 denotes x-axis 100 pixels and y-axis 100 pixels.
3) White space represents the classification of black area.

The results of location-wise classification are used to find the resources by specifying the coordinates. The location calculating formula is used to find the related information of pixels from the array of pixel values. The parallelepiped classification algorithm is used to find the classification of the resources in a precise way. This classification is helpful for the scientist to find the place where a large number of valuable resources are available.
3.9.2. Analysis of Results

The classification of the whole image named as mg-org using parallelepiped classification is shown in table 3.9.2.1. The total number of pixels considered for classification is 43000.

The result indicates that a maximum number of pixels are classified under class 1 resource. The next maximum numbers of pixels are classified under class 3. Only a minimum numbers of pixels are classified under class 2.

Table 3.9.2.1. Classification of mg-org using Parallelepiped the classification technique

<table>
<thead>
<tr>
<th>Serial No</th>
<th>Class selected for classification</th>
<th>Number of pixels classified</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Class 1</td>
<td>4808</td>
</tr>
<tr>
<td>2</td>
<td>Class 2</td>
<td>1013</td>
</tr>
<tr>
<td>3</td>
<td>Class 3</td>
<td>1862</td>
</tr>
</tbody>
</table>

The classification of the whole image named as fische-ocean using parallelepiped classification is shown in table 3.9.2.2.
The result indicates that a maximum number of the pixels are classified as black resource. The next maximum numbers of pixels are classified as fish. Only minimum numbers of pixels are classified as algae.

Table 3.9.2.2. Classification of fische-ocean using the Parallelepiped classification technique

<table>
<thead>
<tr>
<th>Serial No</th>
<th>Class selected for classification</th>
<th>Number of pixels classified</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Black</td>
<td>3335</td>
</tr>
<tr>
<td>2</td>
<td>Fish</td>
<td>1755</td>
</tr>
<tr>
<td>3</td>
<td>Algae</td>
<td>997</td>
</tr>
</tbody>
</table>

The classification of the image named as img-org in location-wise using parallelepiped classification is shown in table 3.9.2.3. The location selected for classification is 100,100. The total number of pixels considered for classification in the selected location is 2601.

The result indicates that a maximum number of the pixels are classified under class 1 resource. The next maximum numbers of pixels are classified under class 2. Only a minimum number of pixels are classified under class 3.
Table 3.9.2.3. Location-wise classification of img-org using the Parallelepiped classification technique

<table>
<thead>
<tr>
<th>Serial No</th>
<th>Class selected for classification</th>
<th>Number of pixels classified</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Class 1</td>
<td>398</td>
</tr>
<tr>
<td>2</td>
<td>Class 2</td>
<td>67</td>
</tr>
<tr>
<td>3</td>
<td>Class 3</td>
<td>20</td>
</tr>
</tbody>
</table>

The classification of the image named as fische-ocean in location-wise using parallelepiped classification is shown in the table 3.9.2.4. The location selected for classification is 50,125. The total number of pixels considered for classification in the selected location is 2601.

The result indicates that a maximum number of the pixels are classified as fish. The next maximum numbers of pixels are classified as black area. Only a minimum number of pixels are classified as algae.
Table 3.9.2.4. Location wise classification of fische-ocean using Parallelepiped classification technique

<table>
<thead>
<tr>
<th>Serial No</th>
<th>Class selected for classification</th>
<th>Number of pixels classified</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Black</td>
<td>121</td>
</tr>
<tr>
<td>2</td>
<td>Fish</td>
<td>494</td>
</tr>
<tr>
<td>3</td>
<td>Algae</td>
<td>65</td>
</tr>
</tbody>
</table>