Digital image analysis
Excercise 3: Texture based segmentation using classification
Spring 2006

The aim of this exercise is to segment a real image into regions by using classification. Different statistical methods of segmenting/classifying images into regions will be tried. Your report should at least include answers to the numbered questions together with discussions. For the main part of the exercise (Section 3): Remember to tell both how you did, and more important why!

1 Getting started

Before you begin, study the text book (GW 12.2.2) and the lecture notes on Maximum Likelihood (ML) classifiers. Note! You will find the main assignment of the exercise in Section 3. If you do not remember the basics of IMP, have Exercise 1 at hand. Don’t forget that IMP has a Help function. Start IMP from a directory where you have write permission:

/it/kurs/bild1/imp &

All images needed in this exercise can be found in the directory:

/it/kurs/bild1/images/

2 Thresholding vs. ML classification

Open the image handBW.ipad. The image shows a hand and a plastic object on a textile background. Your task is to segment the image into the three classes hand, object, and background as accurate as possible. By studying the histogram, try to achieve the segmentation by using multiple thresholding (Segment → Threshold → multiple).

1. Why is multiple thresholding unsuccessful on the image?

Let’s try a different approach, i.e., classification. Mark a few small training areas using Edit → EditRegion. Use different colors for different classes. Choose Segment → Classify → MLTrainAndClassify. This function first trains the classifier according to the pixels in the training areas. Thereafter, it classifies the rest of the image. (When the classification is completed a file called .IMPtempClassifier is saved on the directory where IMP was started. This file contains the definition of the classifier.)

2. Compare the classified result with the thresholded result. Are there any differences? Explain what the classifier does.

ML-classification performs better for multi-layer images, e.g., color (RGB) images with three layers (or bands). Open the color image hand.ipad. To see the separate color bands choose Convert → Extract layer → SplitBands (or Convert → Extract layer → SplitRGB). All this information will be used in the ML-classification.
For comparative purposes, make sure that the same training areas are used for this color image as were applied to the grey-level version. Input images and regions are chosen by clicking in the image using the mouse. IMP will reset the parameters after each run, and you must always make sure you have the correct input image and region. Now, choose Segment → Classify → MLTrainAndClassify with Classifier type set to General.

3. Explain the result compared to the result for the grey-level image.

To see how difficult it can be to solve the segmentation task in a real application, try the same method on the images classcellsBW.ipad and classcells.ipad. This cell image is more complex, and you must be very careful when you choose your training areas. Try different training areas and different classifier types.

4. IMP offers you 4 choices of classifier. Each one of them makes assumptions about the data.

In Figure 1 you will find four scatter plots. Combine the scatterplots with the four classifiers that IMP offers. In each scatterplot you should also sketch the ML decision line.

Figure 1: Scatterplots.
### IMP Classifier

| Simplifying assumptions (Equal class probabilities, 0-1 loss functions, and Gaussian probability density functions are assumed in all cases) |
|------------------|----------------------------------|
| MINDIST | Independent variables. Same variance for all variables. Same covariance matrix for all classes. |
| GENERAL | None. |
| EQUAL COVAR | Same covariance matrix for all classes. |
| UNCORRELATED | Independent variables. |

### 3 ML classification based on texture

Open the image forest2.ipad shown in Figure 2. This is a grey-level image, i.e., it consists of one layer only. The task is to classify the image into forest, open areas, and road. Several actions have to be done prior to a satisfactory classification. The idea is to use the fact that there is some kind of structure of the forest, which the open areas and road lack.

![forest2.ipad](image)

Figure 2: forest2.ipad

The forest areas look somewhat bumpy, but since the “bumpiness” is not a pixel-wise property, it is necessary to classify each pixel by using a small local neighborhood. Only mean or median filtering the image, followed by single-layer classification may help to give a general classification of the different regions, but fine detail will be lost, and this method is therefore not recommended.

Instead, create a multi-layer (at least 4-layer) image by using the function **Convert → Insert layer → add spectral BANDS**. Create new image layers from the original image. If you want to see the different layers of a multi-layer image, and not only the first layer, the
IMAGEHANDLER contains a BROWSE-function. You reach it by pressing the right mouse button somewhere in the image that you want to browse.

Use your imagination and some of the following hints to create layers with useful information that together will result in a satisfactory multidimensional classification:

- Original
- Filtering: linear and non-linear
- Image arithmetic
- Neighbor information (can be obtained by a shift)

To shift an image, you can follow the steps below:

Choose Edit → ExtractImage. Extract an image which is two pixels smaller than the original image, both in $x$ and $y$-direction, using the starting points $(0,0)$, $(0,1)$, $(0,2)$, $(1,0)$, ... and size 254. Images from $(0,0)$ - $(2,2)$ will represent a $3 \times 3$ surrounding of every pixel. A simpler version is to take only the corner pixels in consideration, i.e., the starting points $(0,0)$, $(0,2)$, $(2,0)$, $(2,2)$. Remember that these layers will be smaller than the original image. Layers that have different sizes can not be put together in a multi-layered image.

When the ML-classification is completed, consider some post-processing. Segment → Relaxation could be a useful operation on the resulting region if it looks speckled.

Once more, open the image forest2.ipad, but now use File → Open → Read image & regions. The colored overlay groundTruth shown on the image is a manually made classification. To examine the area of each class you can use Extract → CalcRegFeatureData with Positional features selected.

5. What layers did your final multi-layered image contain? Motivate your choice of these particular layers.

6. Which version of the ML-classifier did you use? Why?

7. Relaxation is sometimes also called a majority filter. Explain what it does to the resulting regions.

8. Calculate the percentage of forest in the image from your classification result.

9. Compare the manually made ground truth with your classification result.

10. Your comments on the exercise.

Good luck!