TMS016 Spatial Statistics and Image Analysis

Chalmers University of Technology

Project report

Project group number 10

Emma Stavås
stavase@student.chalmers.se
Martina H. Billstein
marbil@student.chalmers.se
Devosmita Chatterjee
devcha@student.chalmers.se

May 17, 2019

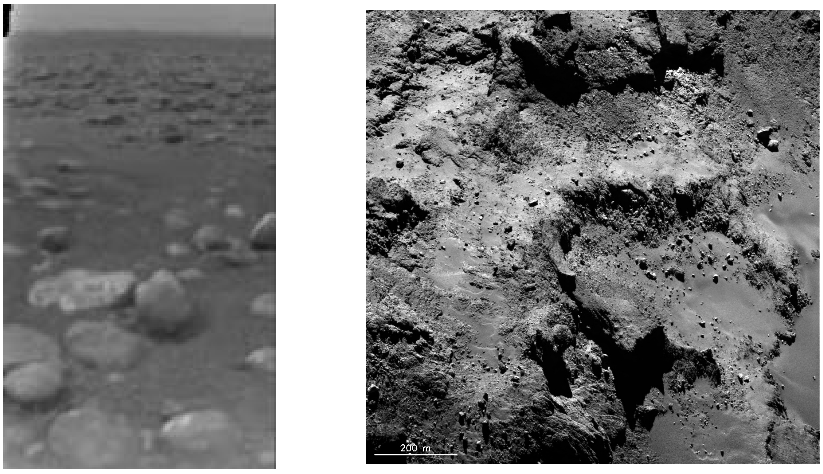
Contents

Part I

Image Reconstruction

1 Introduction

Certain amount of pixels of Titan and Rosetta images are lost during data transfer from the spacecraft. In this project, we use Gaussian Markov Random Fields to reconstruct the missing pixel values of the images of Titan and Rosetta. The original Titan and Rosetta images can be seen in figure 1.



(a) *Titan image* **(b)** *Rosetta image*

Figure 1: *Original images of Titan and Rosetta*

2 Procedure

Matlab code from the first four computer exercises is mainly used in the project by changing some of the parameters.
We start with $p_c = 0.5$ which means half of the pixels are observed and the other half is missing. We fit the model to the spatial data of the image given by

$$Y = B\beta + X + \epsilon$$

where X is the Gaussian random field. First, we use ordinary least-squares (OLS) method to estimate the parameter of the model.

$$\hat{\beta} = (B^T B)^{-1} B^T Y.$$

Then, we use generalized least-squares (GLS) method to estimate the parameter of the model with Matérn covariance function with $\nu = 1$ as given. The estimation is based on only first 10000 observed pixels as per the given instructions.
Now, the Gaussian field X is approximated using a Gaussian Markov Random Field with precision matrix τQ. We compute the precision matrix using the given stencil where the value of the parameters κ and σ are same as that obtained from least-squares estimation of the Matérn variogram to the binned estimate. Then, we perform the kriging predictor which is defined by

$$E[X_p|X_o] = \mu_p - Q_p^{-1} Q_{po}(X_o - \mu_o).$$

We calculate the kriging predictor to reconstruct the missing pixel values of the image.
We use the same above procedure to reconstruct both the images for different percentages of missing pixels just by changing the value of p_c.

3 Results

3.1 Titan image

The plot of binned variogram estimate as well as the estimated Matérn variogram for Titan image is shown in figure 2.

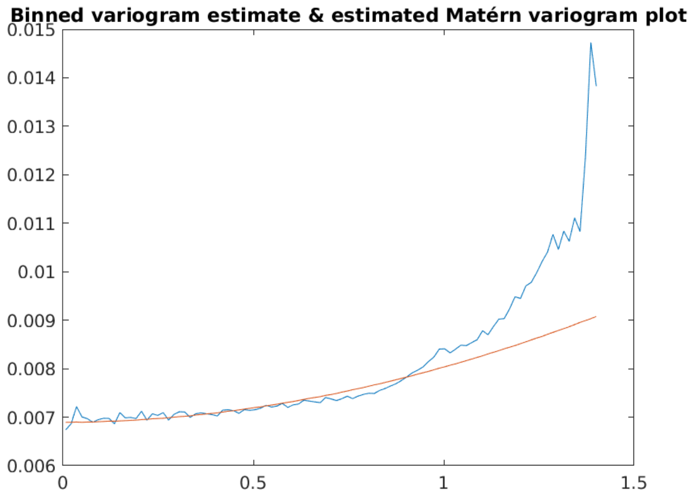


Figure 2: *This figure shows the plot of binned variogram estimate as well as the estimated Matérn variogram for Titan image.*

The parameter estimate for Titan image is presented in table 1.

Table 1: *This table represents the parameter estimates ($\hat{\beta}_{OLS}$ & $\hat{\beta}_{GLS}$) for Titan image using OLS and GLS.*

Image	$\hat{\beta}_{OLS}$	$\hat{\beta}_{GLS}$
Titan.jpg	0.3920	0.3692

The value of kappa (κ) affects the generalised least squares estimate for Titan image which is presented in table 2.

Table 2: *This table represents the different values of the parameter estimate for Titan image for different κ. Here β indicates the true parameter value while $\hat{\beta}_{OLS}$ & $\hat{\beta}_{GLS}$ indicate the parameter estimates for Titan image using OLS and GLS.*

Different κ	β	$\hat{\beta}_{OLS}$	$\hat{\beta}_{GLS}$
κ	0.3922	0.3920	0.3692
$\frac{\kappa}{4}$	0.3922	0.3920	-0.6980
$\frac{\kappa}{2}$	0.3922	0.3920	0.3017
2κ	0.3922	0.3920	0.2829
4κ	0.3922	0.3920	0.6714

Image reconstructions of Titan for different percentages of missing pixels are shown in figure 3, figure 4 and figure 5. Based on the reconstructed images, the percentage of missing pixel values of Titan image should not be more than 60% if we want to get a reasonable estimate of the true image.

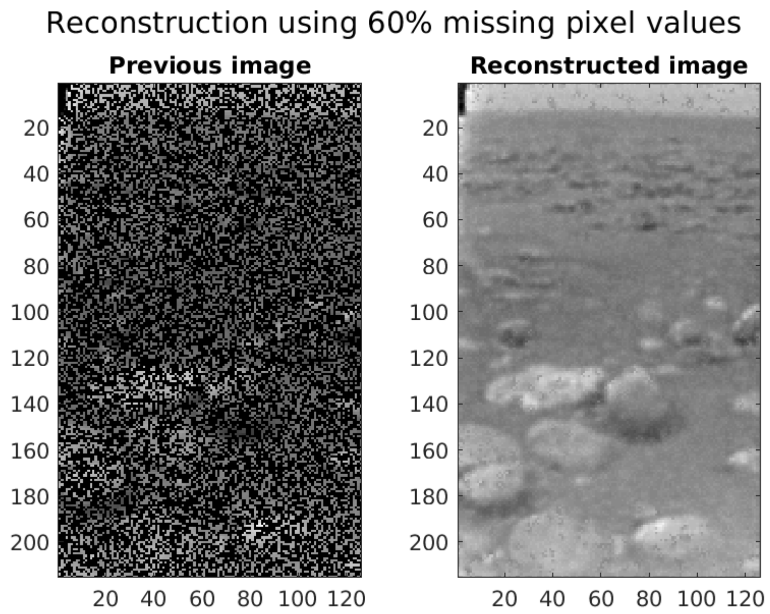


Figure 3: *This figure shows the reconstruction of the Titan image for 60% missing pixels. The "Previous image" resembles what the image would look like if it was constructed of only observed values, and no estimated values for missing pixel values.*

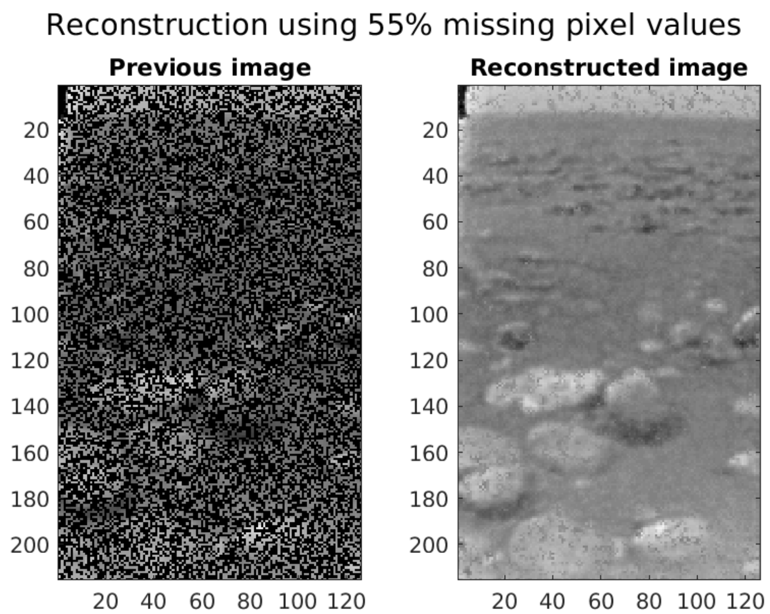


Figure 4: *This figure shows the reconstruction of the Titan image for 55% missing pixels.*

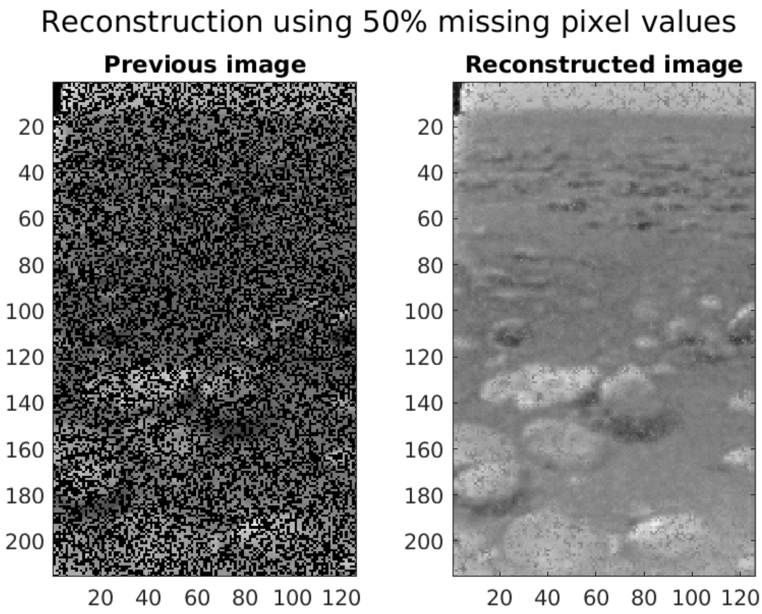


Figure 5: *This figure shows the reconstruction of the Titan image for 50% missing pixels.*

3.2 Rosetta image

The plot of binned variogram estimate as well as the estimated Matérn variogram for Rosetta image is shown in figure 6.

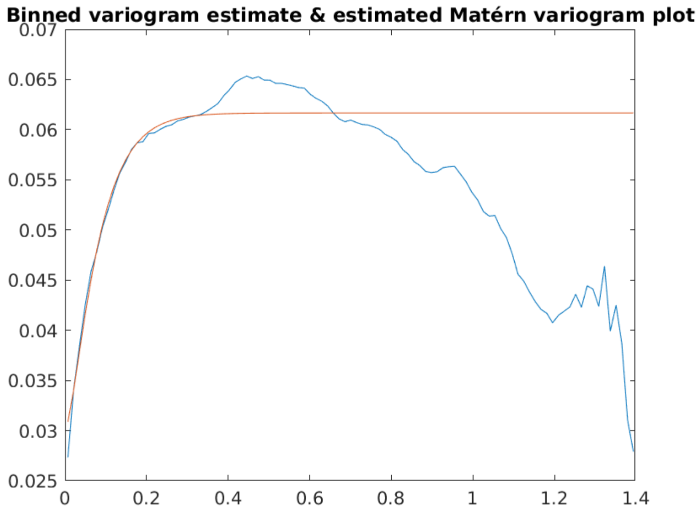


Figure 6: *This figure shows the plot of binned variogram estimate as well as the estimated Matérn variogram for Rosetta image.*

The parameter estimate for Rosetta image is presented in table 3.

Table 3: *This table represents the parameter estimates ($\hat{\beta}_{OLS}$ & $\hat{\beta}_{GLS}$) for Rosetta image using OLS and GLS.*

Image	$\hat{\beta}_{OLS}$	$\hat{\beta}_{GLS}$
Rosetta.jpg	0.3864	0.3734

The value of kappa (κ) affects the generalised least squares estimate for Rosetta image which is presented in table 4.

Table 4: *This table represents the different values of the parameter estimate for Rosetta image for different κ. Here β indicates the true parameter value while $\hat{\beta}_{OLS}$ & $\hat{\beta}_{GLS}$ indicate the parameter estimates for Rosetta image using OLS and GLS.*

Different κ	β	$\hat{\beta}_{OLS}$	$\hat{\beta}_{GLS}$
κ	0.3843	0.3864	0.3734
$\frac{\kappa}{4}$	0.3843	0.3864	0.3234
$\frac{\kappa}{2}$	0.3843	0.3864	0.3593
2κ	0.3843	0.3864	0.3795
4κ	0.3843	0.3864	0.3824

Image reconstructions of Rosetta for different percentages of missing pixels are shown in figure 7, figure 8, figure 9 and figure 10. The percentage of missing pixel values of Rosetta image should not be more than 65% if we want to get a reasonable estimate of the true image.

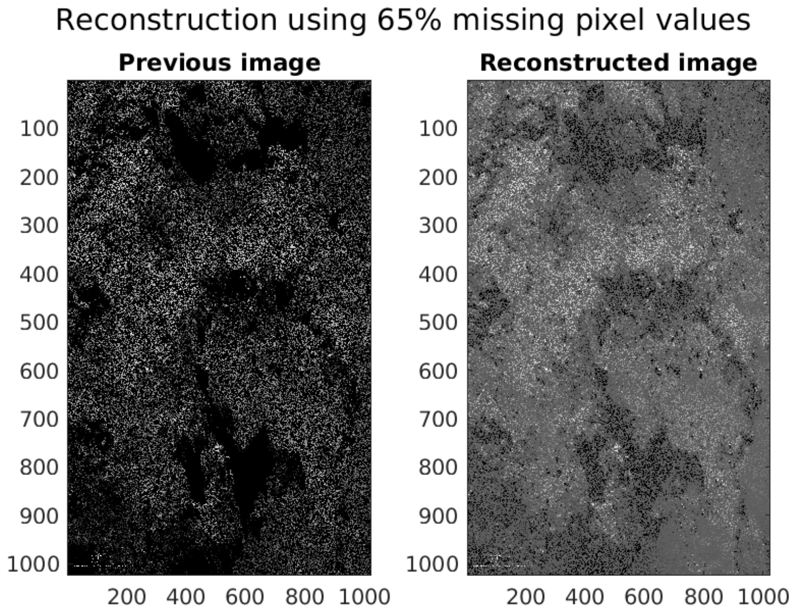


Figure 7: *This figure shows the reconstruction of the Rosetta image for 65% missing pixels.*

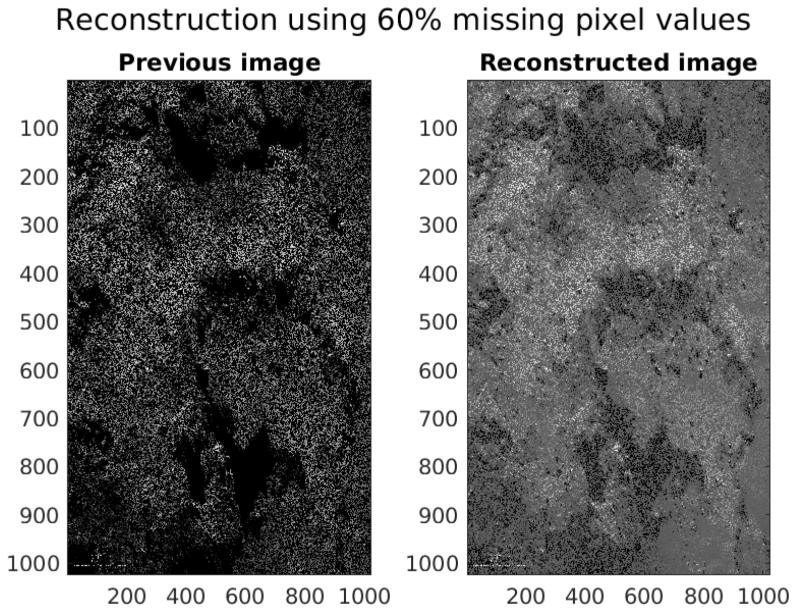


Figure 8: *This figure shows the reconstruction of the Rosetta image for 60% missing pixels.*

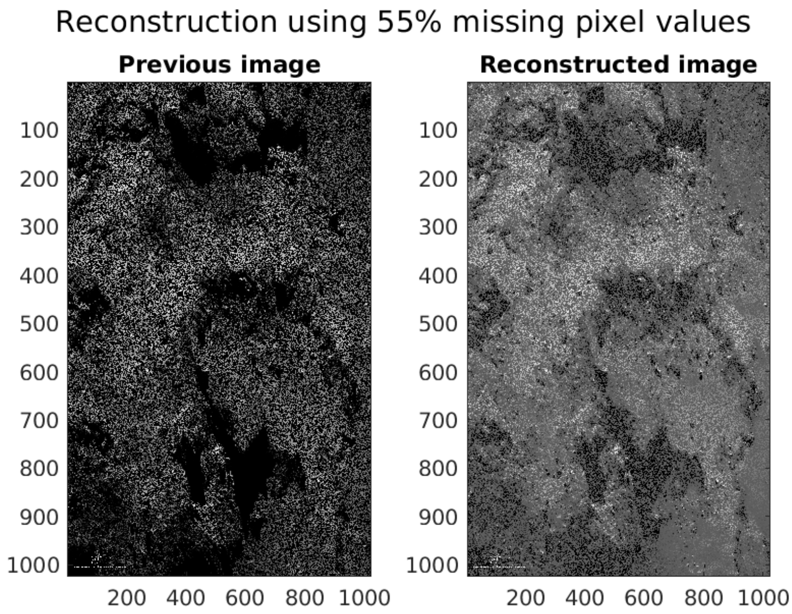


Figure 9: *This figure shows the reconstruction of the Rosetta image for 55% missing pixels.*

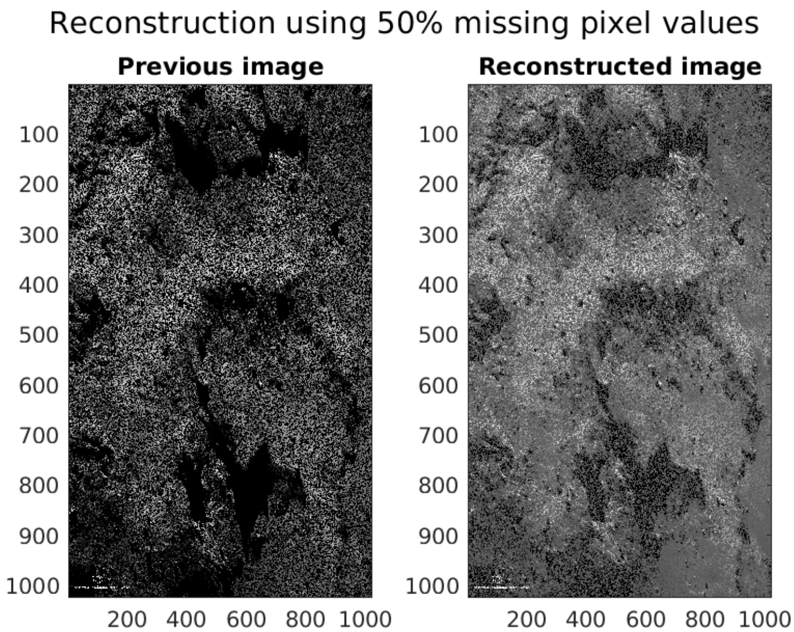


Figure 10: *This figure shows the reconstruction of the Rosetta image for 50% missing pixels.*

Part II

Image Segmentation

4 Introduction

This project aims to investigate the impact of noise on three different algorithms when segmenting an image. The algorithms are the k-means, Gaussian mixture models and Markov fandom field methods. To do so an image representing a material with varying permeability is analyzed.

5 Procedure

MATLAB code from the computer exercises was used for the different algorithms with some tuning of input parameters. According to the instruction the classes were set to $K = 2$. The neighbourhood structure used for the Markov random field mixture models was

$$N = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

The code in its entirety can be found attached to the report.

The visualization of the images was done using imagesc. A colorbar was added to the images to simplify the interpretation. For the segmented images the pixel values are binary, so the pixels will either belong to class 1 or 2 in the colorbar.

According to the instructions of the second task, Gaussian noise was then added to each pixel in the image. The noise was simulated using

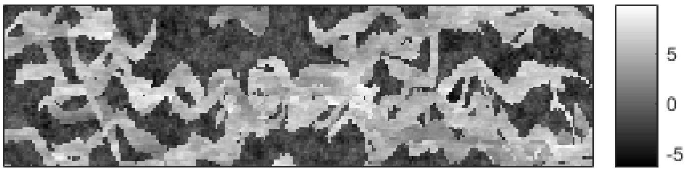
```
noise = noise_sigma * randn(size(image));.
```

The variance of the noise was adjusted by tuning the noise_sigma variable. Two values of σ^2 was tested, $\sigma^2 = 1$ and $\sigma^2 = 3$.

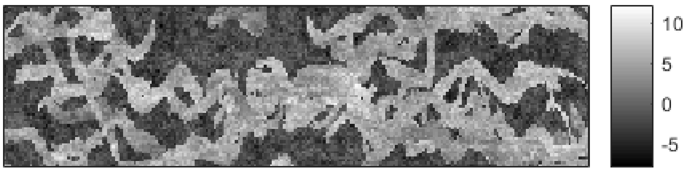
6 Results

The results are presented method-wise to be able to clearly compare the impact of noise on the segmentation and classification. Since the true labels of the pixels in some sense are unknown, other than that it is visually possible to distinguish two classes, when a result is referred to as better or worse it is with a quite ad hoc visual quality measure in mind.

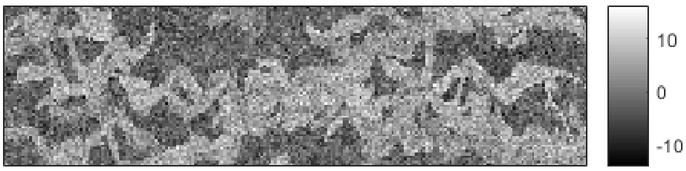
The original image and the noisy images without segmentation are presented in figure 11a, 11b and 11c. From the visualization it is apparent that after adding noise the image is not as clear as before. This is also indicated by the colorbar at the right of the figures, which has a larger scale of values after adding the noise. When noise with variance = 1 is added a human eye is however still possible to separate two levels of permeability from each other. This is harder for noise with variance = 3, but not impossible.



(a) *The original image.*



(b) *Image with Gaussian noise added to each pixel.*
Mean = 0, var = 1.



(c) *Image with Gaussian noise added to each pixel.*
Mean = 0, var = 3.

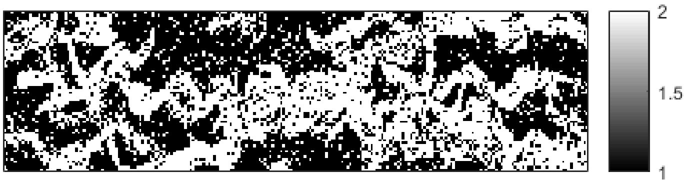
Figure 11: *A comparison between the original image and the image with Gaussian noise with variance 1 and 3. The color bar indicates the color corresponding to certain pixel values.*

6.1 K-means algorithm

In figure 12a, 12b and 12c the results for the k-means algorithm are presented. The performance deteriorates as expected when the noise increases. At variance = 1 the algorithm still performs good with the vast majority of pixel classified correctly. with variance = 3 there are several misclassified pixels. These pixels appears almost as a noise scattered over the image, and better results would perhaps be able to acquired with some postprocessing.

(a) *K-means segmented image without noise.*

(b) *K-means segmented image with Gaussian noise added to each pixel. Mean = 0, var = 1.*



(c) *K-means segmented image with Gaussian noise added to each pixel. Mean = 0, var = 3.*

Figure 12: *A comparison between the original image and the k-means segmented image with Gaussian noise with variance 1 and 3.*

6.2 Gaussian mixture model method

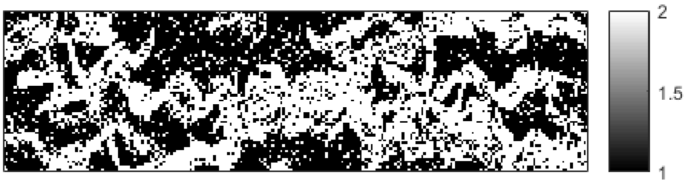
In figure 13a, 13b and 13c the results from the GMM method is presented. These results resembles the ones from the k-means algorithm in character. Not unexpectedly, the algorithm needs more iterations to converge when the noise is increased.

6.3 Markov random field mixture model method

In figure 14a, 14b and 14c the results from the MRF method is presented. Just like k-means and GMM the algorithm preforms well when there is no noise in the image. However, when the variance increases the segmentation is a bit different from the other two methods. MRF is more unlikely to scatter pixels of different classes, which makes the method robust when the noise is increased. Also here, the iterations and computational time needed to converge increases with increasing noise. For variance = 3 the iterations reaches the set maximum 100 every run.

(a) *GMM segmented image without noise.*

(b) *GMM segmented image with Gaussian noise added to each pixel. Mean = 0, var = 1.*

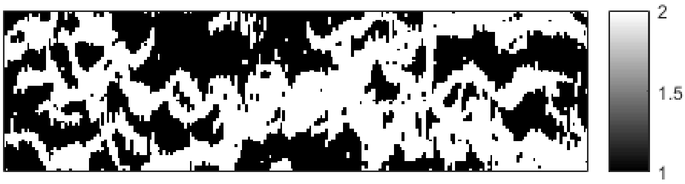


(c) *GMM segmented image with Gaussian noise added to each pixel. Mean = 0, var = 3.*

Figure 13: *A comparison between the original image and the GMM segmented image with Gaussian noise with variance 1 and 3.*

(a) *MRF segmented image without noise.*

(b) *MRF segmented image with Gaussian noise added to each pixel. Mean = 0, var = 1.*



(c) *MRF segmented image with Gaussian noise added to each pixel. Mean = 0, var = 3.*

Figure 14: *A comparison between the original image and the MRF segmented image with Gaussian noise with variance 1 and 3.*

7 Comparison

The results indicate that segmentation with Gaussian mixture models and k-means worsen significantly when there is large enough noise in an image. Markov random field mixture models seems more robust to noise, but is not unaffected.

A comparison of pixels assigned to the same class when no noise is added is presented in table 5. The table supports the observation that the three methods are similar in results for segmentation of the original image. In table 6 the same comparison is presented when noise is added. These tables shows that the difference between MRF and the other two methods is larger when the noise is increased. Note that this is only a comparison between the methods. A high percentage doesn't necessarily mean a correct classification.

	K-means		
K-means	-	**GMM**	
GMM	99.19%	-	**MRF**
MRF	99.14%	99.83%	-

Table 5: *Percentage of pixels assigned the same class when no noise is added to the image.*

Table 6: *Percentage of pixels assigned the same class when noise with variance 1 and 3 is added to the image.*

	K-means		
K-means	-	**GMM**	
GMM	99.13%	-	**MRF**
MRF	98.35%	98.87%	-

(a) *Percentage of pixels assigned the same class when noise with variance 1 is added to the image.*

	K-means		
K-means	-	**GMM**	
GMM	99.86%	-	**MRF**
MRF	88.88%	88.94%	-

(b) *Percentage of pixels assigned the same class when noise with variance 3 is added to the image.*