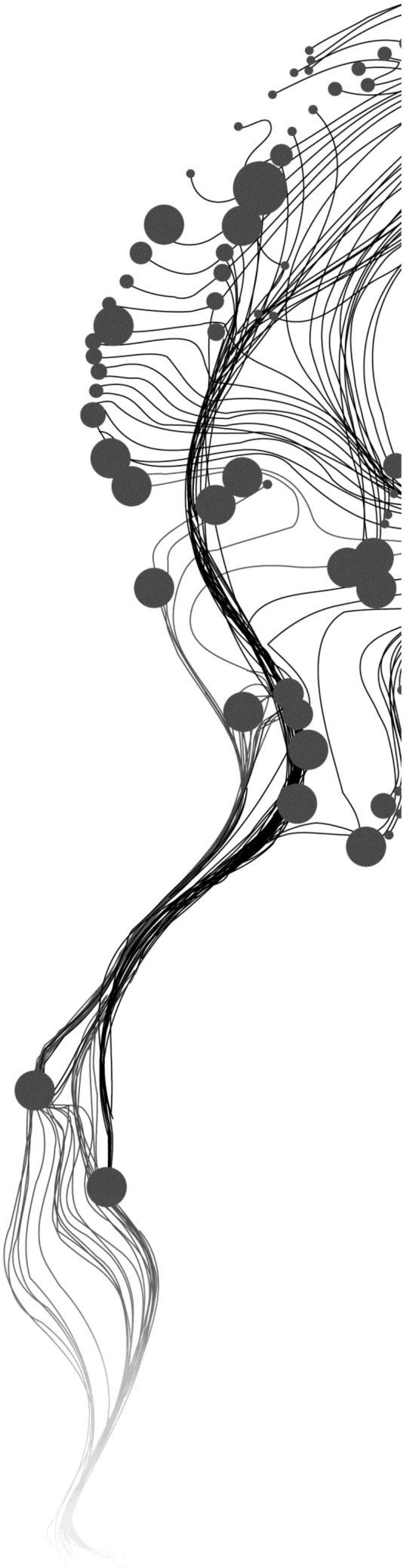
Super Resolution Mapping with Support Vector Machine

NAZANIN SEPEHRI
March, 2011

SUPERVISORS:
Dr.V.A.Tolpekin
Prof.Dr.Ir.A.Stein



Super Resolution mapping with Support Vector Machine

NAZANIN SEPEHRI

Enschede, The Netherlands, March, 2011

Thesis submitted to the Faculty of Geo-Information Science and Earth Observation of the University of Twente in partial fulfilment of the requirements for the degree of Master of Science in Geo-information Science and Earth Observation.
Specialization: Geoinformatics

SUPERVISORS:
Dr.V.A.Tolpekin
Prof.Dr.Ir.A.Stein

THESIS ASSESSMENT BOARD:
Prof.Dr.Ir.A.Stein (Chair)
A.K. Bucksch MSc (External Examiner)

ABSTRACT

Super resolution mapping (SRM) is a method that divides pixels of input image into finer resolution classified map. Among the existing techniques for SRM, Markov random field (MRF) based on SRM has been introduced as method that use contextual information for SRM to reduce the number of isolated and misclassified pixels. Current SRM either ignore within class variance or model it with normal distribution. Many classes are normally distributed but not all of them and to classify, the classes that are not normally distributed, it is better to use non-parametric classifications methods such as support vector machine (SVM). The SVM classification method has been presented better accuracy than other classifications methods such as maximum likelihood. SVM does not make assumption for distributed of classes and contains transformation with kernel functions for non-linear separable classes. It also does not need big training set. Therefore to apply SRM on images with non-normally distribution classes, in this study SVM was incorporated into MRF-SRM classification method.

The data that used in this study were two synthetic images and a remote sensing image. The synthetic images produced in different class distributions and remote sensing image is from different sources, optical image and radar image. To estimate mixture probability distance of each data from separating hyperplane and the histogram of those distances are used. The histogram of mixed pixels is obtained by mixing the histograms of distances for pure data set. The interpolation method is used to find the value of mixture probability for a mixed pixel with known proportion of each class. Additionally the proposed SVM mixture probability method is used in likelihood energy of MRF-SRM. The accuracy assessment of the method is done by RMSE for mixture probability and by kappa coefficient for application of MRF-SRM with SVM.

The SVM mixture model gives identical RMSE value and final SRM results are smooth maps of fine resolution. Also this method converts multiband dataset to a single band size (distance from hyperplane) that makes the implementation faster. The experimental results from application of the method on synthetic images and remote sensing data show that the MRF-SRM method incorporated with SVM is suitable for the images with any kind of distributions.

Keywords

Super resolution mapping (SRM), Markov random field (MRF), Support vector machine (SVM), Mixture probability.

ACKNOWLEDGEMENTS

I acknowledge my special gratitude to my supervisors Dr. Valentyn Tolpekin and Professor Dr. Alfred Stein for guiding me throughout my research. Thanks a lot for their support, encouragement and constructive comments during my work.

I would like to convey my acknowledgment to all the teachers and staff in ITC department who helped me with great ideas since the beginning of the course till the completion of the thesis.

I am so thankful to my colleagues in GFM2, 2009 for their kindly friendship during the 18 months of study. It was a wonderful experience to study with people from many countries all around the world.

My very special thanks go to Ivar M. Ledezma Casablanca for all his helps and moral supports during my study and my hard times.

Heartfelt thanks to my lovely parents, my brothers and sister for all their love, prayer and encouragement, their supports helped me to pursue my studies.

This work is dedicated to my dear parents Manochehr Sepehri and Nasrin Nikraftar.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. Background

Land cover classification has been identified as one of the requirements to manage and understand the environment. Remote sensing has the ability of data acquisition to produce land cover classification maps from satellite images. With this technology users do not need to do field work to get information of large area because satellite images can cover a large area just in one image. In satellite images the spatial resolution is very important to interpret land cover information and users have been interested in more detailed information of the ground through these images.

Fine spatial resolution images help to have more spatial detail in classification map. There are some problems in using these images such as, usually they have fewer spectral bands and more mixed pixels than coarse resolution images and it is often expensive to use fine resolution images for covering a large area. So it is useful to identify a technique that can obtain finer resolution classification map from coarse resolution images (Foody, 2006; Kasetkasem, et al., 2005; Tatem, et al., 2001; Tolpekin & Stein, 2009).

SRM is a method which divides a large pixel into a finer classified map (Verhoeye & De Wulf, 2002). It changes a soft classification result into a finer scale hard classification map. But after producing the initial SRM map, spatial distribution of classes is still unknown and there are many pixels that do not have similar classification as their neighbours. Therefore statistical correlation between neighbouring pixels should be computed (Kasetkasem & Varshney, 2002).

In many studies the concept of contextual model such as Markov Random Field (MRF) has been used in image classification. With MRF, pixels are not considered as an isolated pixel (B. Tso & Mather, 2001). This algorithm is a useful tool to solve spatial dependency between neighbourhood pixels in initial SRM map (Kasetkasem, et al., 2005). MRF-based SRM described in (Kasetkasem, et al., 2005) uses Bayesian classification where the class spectral values have been modelled with normal (Gaussian) distribution. Many classes can be described by normal distribution but not all of them, for instance radar images are exponentially, Gamma or Rayleigh distributed, or in high resolution images like, Quick Bird, DN values of the classes are often not normally distributed. To classify these classes non-parametric classification methods are appropriate. These methods do not make assumption about probability distribution.

Many non-parametric classifiers have been developed. One of the most popular is Support Vector Machine (SVM). The concept of this method is based on discriminating the classes optimally by a decision boundary, a hyper-plane. Hyper-plane should be in maximum distance from training samples of both classes (Brown, et al., 1999).

1.2. Problem Statement

MRF-based SRM requires a model for spectral mixture i.e. the mixed pixel's spectral value in relation to pixel composition for classes. In existing MRF-based SRM the classes are assumed to be normally distributed for modelling mixed pixels (Kasetkasem, et al., 2005; Tolpekin & Stein, 2009). To apply MRF-based SRM technique for non-normally distributed classes, a non-parametric method, such as SVM, might be used. Application of MRF-based SRM to model spectral mixture with SVM has not been described in literature. This is the aim of this research.

1.3. Research Objective

The main objective of the research is to improve MRF-based SRM technique by incorporating SVM classification method, for modelling conditional probability of spectral values of mixed pixels.

1.4. Research questions

1) How to estimate mixture probability for mixed pixel with SVM?
2) How to integrate SVM mixture probability with MRF-based SRM?
3) How to estimate a parameter of MRF based SRM technique incorporated with SVM?
4) How to validate the results of the MRF based SRM technique incorporated with SVM?

1.5. Research approach

The research starts with literature review about SRM, MRF and SVM, to know about the advantage and limitation of each method. As the objective of the study is to apply MRF based SRM for different distributions, two synthetic images with normal and non-normally distribution for the classes prepared.
Initial study focused on normal distribution class then the results generalised for other distributions. SVM classification and different properties of this classification method studied. The algorithm to find the mixture probability from the results of SVM is executed on synthetic images and its accuracy was obtained. Then developed method for SVM mixture incorporated with MRF based SRM algorithm. At the end the obtained results are evaluated.
General Framework of the research is shown in Fig1-1.

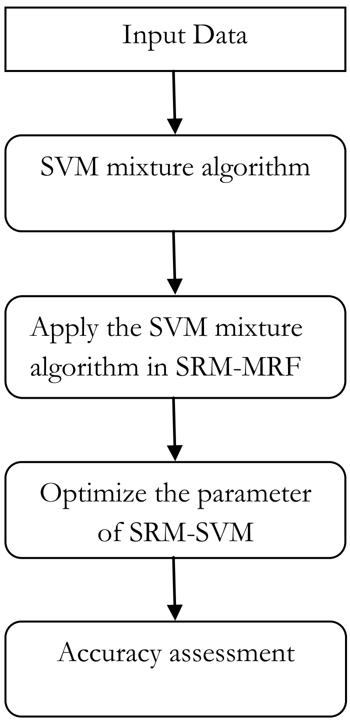


Figure 1-1General approach of the thesis

1.6. Structure of the thesis

The thesis organized in eight chapters. The first chapter is about background of the research, objective, questions, problem statement and research set up. Second chapter includes literature review of the researches that are related to super resolution mapping, Markov random field and support vector machine. Chapter three explains the remote sensing data that is used and the software that implementation was done with them. Chapter four describes the methods was used for implementation. It is about their mathematic background and how they can be applied on the methods. Chapter five describes the process of applying the methods. Chapter five of the thesis explains about the result of the implementation. Chapter seven discusses the results and analyse them. And the last chapter makes conclusion and recommendation for further researches.

2. LITRERATURE REVIEW

2.1. Land Cover Classification

Landcover identification from the earth's surface is important in many fields such as agricultural, hydrological, environmental and ecological. One of the sources of data acquisition for land cover information is remotely sensed images. The advantage of remote sensing is that producers do not need to go to the field to gather information and the information can be extracted from satellite sensors. General properties of remote sensing instruments are their spatial, spectral and temporal resolution and the number of captured spectral bands. Spatial resolution is the geometric characterise of satellite images, it is the ability to distinguish between target points and to measure the distance between them. Spectral resolution is the width of spectral bands of an image. Temporal resolution refers to the difference of time between the multiband images taken from different moments and from same area (Mather & Koch, 1987). Extracting information of land cover from satellite images is by means of classification which is the process to define each pixel labelled as one class. Usually, classification is done by the spectral information of the pixel. Most of the methods for classification are based on statistical algorithms that the pixels in the same class are in the same probability distribution.

Classification can be done by two main methods: supervised and unsupervised classification. In supervised classification the user define the property of the classes by training sets of pixels that have similarity in spectral properties. Then the computation of classification will be done with parametric classification that use mean and covariance of classes or non-parametric methods such as neural networks and support vector machines (Richards & Jia, 2006). In unsupervised classification the definition of classes and the process itself to do the classification is done automatically. Mostly unsupervised classification methods uses clustering algorithm. They can be used to define spectral composition of classes for primary information of supervised classification. Presenting the information of land cover classification is done in thematic maps, that is a map wherein each set of pixel with similar values is represented by thematic categories (Richards & Jia, 2006).

2.2. Mixed pixel

Pixel is the smallest part of the image. The objective of land cover classification is based on the assumption that each pixel corresponds to a single class but it is not always true. When the Instantaneous Field of View (IFOV) , the area on the ground which is view by the sensor, has more than one type of land cover or object, then the pixel may have more than one class and is defined as a mixed pixel (Fisher, 1997; Foody, 2006), this means that for each pixel the spectral signature reflects the different surface materials (Zhu, 2005). The nature of the classes also has influence on mixing the classes for example mixture of classes is more in mixed of vegetation classes rather than mix of vegetation and soil classes (Kasetkasem, et al., 2005). Four main types of mixed pixels are introduced in a paper from Fisher, (Fisher, 1997):

- Boundaries are between more than one mapping units
- The integrated between phenomena
- Linear sub-pixel objects

- Small sub pixel object

Classification of mixed pixels is done by soft or sub-pixel classification. Different class labels in each mixed pixel are identified with class memberships. The output of soft classification is shown by the thematic map for each class. these maps show the degree of membership in each pixel (Haglund, 2000).

2.3. Spectral unmixing

A variety of methods are used to classify mixed pixels usually these methods estimate the fraction of each class in one pixel (Foody, 2006). Spectral unmixing is the process that decomposes each pixel into the number of classes. The proportion of each class is represented by fraction or abundance of that class (Keshava, 2003). Linear spectral mixture is a common approach that is used to solve spectral mixture problems. The assumption of linear spectral mixture is based on linear mixing of received signal between different land cover within the pixel (Zhu, 2005). In description of linear spectral mixture the weights are derived from the proportion of each class in the pixel (Bastin, 1997).

2.4. Super Resolution Mapping

Super Resolution Mapping (SRM) is a method that divides a coarse resolution pixel to finer pixels and prepare classified map in finer resolution. SRM has been done with variety of algorithms, such as; knowledge-based procedure, Hopfield neural networks, linear optimization, genetic algorithm and neural network predicted wavelet coefficients. This section describes number of works that have been done with different SRM algorithms.

Super resolution mapping with Hopfield neural networks was developed in a paper from Tatem, et al. (2001). They used Hopfield neural networks as an energy minimization tool for fuzzy classification results and presented spatial distribution of classes between pixels only for simulated imagery (Tatem, et al., 2001). They extended their research in Tatem, et al. (2003) by applying their algorithm on Land sat TM agricultural imagery. The results represent that SRM with Hopfield neural networks can produce higher accuracy than traditional algorithms and class for each pixel are correctly located, but it does not have accurate result for complex features (Tatem, et al., 2003).

The research for sub-pixel mapping with the application of linear optimization techniques was done in Verhoeye and De Wulf (2002) paper. In that research coarse resolution images were used but if the main assumption about spatial dependency exists, it is possible to apply also on any resolution. The algorithm used limited number of classes with known spatial dependency and it was not able to locate the objects that are smaller than a pixel (Verhoeye & De Wulf, 2002). Mertens, et al. (2003) continued Verhoeye and De Wulf (2002) study and developed genetic algorithm in SRM to locate sub-pixels. Genetic algorithm is a fast method base on natural principles. Finding many parameters from the algorithm was the disadvantage of that approach. The results showed that the measured accuracy was higher than conventional hard classification (Mertens, et al., 2003).

Boucher and Kyriakidis (2006) introduced geostatistical algorithm of indicator Kriging and indicator stochastic simulation.in SRM. In their research the prior spatial information model was parameterized explicitly with variogram models that show spatial variety of classes in fine resolution pixels. They continued their work in Boucher, et al. (2008) by using training image additional to variogram models as prior information. They showed that their methodology can be used for spatial analyse (Boucher, et al., 2008).

In most of those algorithms accuracy of SRM depends on the accuracy of classification method and spatial dependency between pixels was used only after finding fraction of each classes (Kasetkasem, et al., 2005).

Kasetkasem, et al. (2005) introduced MRF base SRM method. The initial SRM classified map was generated from raw coarse resolution image. As MRF can model the statistical correlation between neighbouring pixels, it used to define spatial dependency between pixels. They showed that by incorporating MRF in SRM, the classified map has less number of misclassified pixels so the land cover map is smoother and more connected (Kasetkasem, et al., 2005).Influence of class separability was studied by Tolpekin and Stein (2009).They used MRF-SRM method of Kasetkasem, et al. (2005) in their research. Smoothness parameter was introduced, it controls the balance between two parameter of MRF; prior and conditional energy. They reported that SRM quality is related to smoothness parameter, scale factor and class separability. Class separability would be less important by increasing scale factor (Tolpekin & Stein, 2009).

2.5. Markov Random Field

Context defined as spatial, spectral and temporal dimension. Spectral dimension is different bands of electronic spectrum. Spatial dimension is the correlation between pixels in spatial neighbourhood. Temporal dimensional is the context between images of same area in different time (Solberg, et al., 1996).

Researches have been done about usability of MRF in remote sensing such as segmentation, classification, texture analysis and recently in super resolution mapping.

Hu and Fahmy (1992) developed an algorithm with MRF for supervised and unsupervised segmentation. Their algorithm combined binomial model for texture and the multi-level logistic model for region distribution. In supervised segmentation maxima of a posterior (MAP) was used and for unsupervised segmentation a new parameter estimation was presented that can extract parameter directly from a given image (Hu & Fahmy, 1992). Unsupervised segmentation for classification of multispectral image with MRF was proposed in Sarkar, et al. (2002). Region adjacency graph Madevska-Bogdanova, et al. (2004) applied on the original image by using MRF. Minimization of energy function for MRF was done by multivariate statistical test. Results for classification were compared with maximum likelihood procedure and the accuracy of their method was higher in different samples (Sarkar, et al., 2002).

Melgani and Serpico (2003) used MRF algorithm to increase the accuracy and reliability of the classification to extract better temporal information. They improved an algorithm base on the perception of 'minimum perturbation' that was implemented with pseudo inverse technique for minimisation of sum of squared errors. Acceptable accuracy was obtained from their algorithm (Melgani & Serpico, 2003). Unsupervised classification for radar images with hidden Markov chain models and mixture estimation considered in Fjortoft, et al (2003). They determined the distribution families and parameters of classes by generalization of mixture estimation. The algorithm had good results but it has difficulty in estimation of the regularity parameter (Fjortoft, et al., 2003). Tso and Olsen (B Tso & Olsen, 2005) improved contextual information based on MRF and multi-scale fuzzy line process for image classification. They used panchromatic and multi-spectral IKONOS images as data. The parameter estimated with probability histogram for boundary pixels and maximum a posterior margin (MPM) applied to find the solution. Their results presented success in generating the patch-wise classification patterns, and increasing the accuracy and visual interpretation (B Tso & Olsen, 2005).

2.6. Support Vector Machine

Support Vector Machine (SVM) is a non-parametric method that classifies data by drawing separating hyperplane between classes in feature space. This section discusses some studies that were done with SVM in remote sensing.

Vapnik and Cortes (1995) introduced support vector machine as a binary classification in their study. Their study SVM contains three main ideas; optimal hyper-planes, dot product (to extend results from

linear to non-linear) and soft margin (for error in training set). SVM classification method was compared with other algorithms and results shows that SVM has higher accuracy (Vapnik & Cortes, 1995).
Huang, et al. (2002) used TM and MODIS images to study image classification by SVM. Selecting kernel function and kernel parameter was considered in their research. They compared different parameter for polynomial kernel and RBF kernels and results revealed that kernel type and its parameter affect the shape of hyperplane and influence the results for SVM classification. They also compared three different classification methods, Maximum Likelihood, Neural Network and Decision Tree Classifiers with SVM. Their study showed that SVM has higher accuracy than the other three classification methods, especially for high dimensional space. Also SVM has more stability in overall accuracy (Huang, et al., 2002).
Foody and Mathur (2004b) studied training samples for SVM classification they used three bands of SPOT HRV image and chose training samples from agricultural crops. They analysed data in two classes and investigated that by using SVM classification only training samples that placed in vicinity of hyper-plane are needed and other samples does not affect the SVM results (Foody & Mathur, 2004b).
The parameters that affect the SVM classification were discussed in Watanachaturaporn (2004). They applied SVM on hyper-pectral image from AVIRIS sensor. Different penalty value for three multiclass classification methods (one against the rest, pairwise and directed acrylic graph were applied) also different kernels compared. They investigated that for each set of data there is an optimum penalty value but it takes more time for classification with higher penalty value (Watanachaturaporn, et al., 2004).
Bruzzone and Persello (2009) presented a context-sensitive SVM classifier. They applied the method on two set of image data, IKONOS image for low resolution set and Land-Sat image for medium resolution. The aim of their method is to reduce the effect of mislabelled data in training set on defining the hyperplane in SVM classifier therefore learning algorithm is less related to unpredictable training data. They compared method with other algorithms and showed that their results are more accurate and stable for noisy training set (Bruzzone & Persello, 2009).
The results of classification with SVM is hard classification that label each class only with one label, but as mentioned in 2.1.1 , naturally there are many mixed pixels so it is required. SVM probabilistic method can be improved by fitting the output of SVM to a sigmoid which defined in the paper from Platt (1999). Maximum likelihood estimation was used to estimate the parameter of the sigmoid in Lin, et al. (2001) improved the Platt's Method and their algorithm could be used in calculating posterior probability of SVM output.
Lin (2002) applied a fuzzy membership to each input point in SVM, therefore with different input, different decision would be optimized (Lin, 2002). By improving Lin (2002) results Bovolo, et al. (2010) found the membership of an unknown pixel with SVM and developed their method for multiclass classification. The method has all the properties of crisp SVM such as; could be apply for high dimensional data and good generalization capability .Their results have better accuracy for sub-pixel classification rather than fuzzy classification with neural network (Bovolo, et al., 2010).
Support vector machine classification was defined as binary classification. Numbers of methods have been used to improve SVM for multiclass classification and the most popular ones are one-against-one, one-against-all and directed acyclic graph. In one-against-one many binary classifiers compare together while in one-against-all each class compares to the rest of classes, directed acyclic graph is also works by many binary classifiers (Hsu & Lin, 2002).
Hsu and Lin (2002) studied decomposition implementation of two method. Results showed that for big problems the methods that use all data at once needs less training data and one-against-one and directed acyclic graph are more appropriate than the other methods of multi classification of SVM (Hsu & Lin, 2002). Foody and Mathur (2004a) also developed classification of airborne thematic map (ATM) data with SVM to multiclass classification. They classified the same data with a discriminate analysis, decision tree

and multilayer perception neural network. They used the method one-against-all in their research because classification parameter such as penalty value or kernel function need to estimate only one time and it needs fewer support vectors. The accuracy of each classification method was related to the number of training set and with more training set more accurate classification was obtained. But the most accurate classification was derived from SVM multiclass classification (Foody & Mathur, 2004a).

3. MATERIALS

3.1. Remote sensing Data

In order to apply the method on remote sensing image, an optical image with normal distribution classes and a radar image with exponential distribution classes are selected. Brief introduction about these images and the study area is explained in his chapter.

3.1.1. Location of Study area

The study area is located in Hengelo a town in the center of the Twente area in the east of the Netherlands. Geographical information of the area is approximately $52°14'47''$ N, $6°48'39''$ E. The area includes many types of classes such as water bodies, trees, agricultural fields and buildings. Figure 3-1 shows the location of study area. From this area ERS image as radar image and Spot-5 as optical image are selected for implementation. The reference data is the topographic map of the Netherlands 1:10000, this reference is used to visual interpretation of the final product.

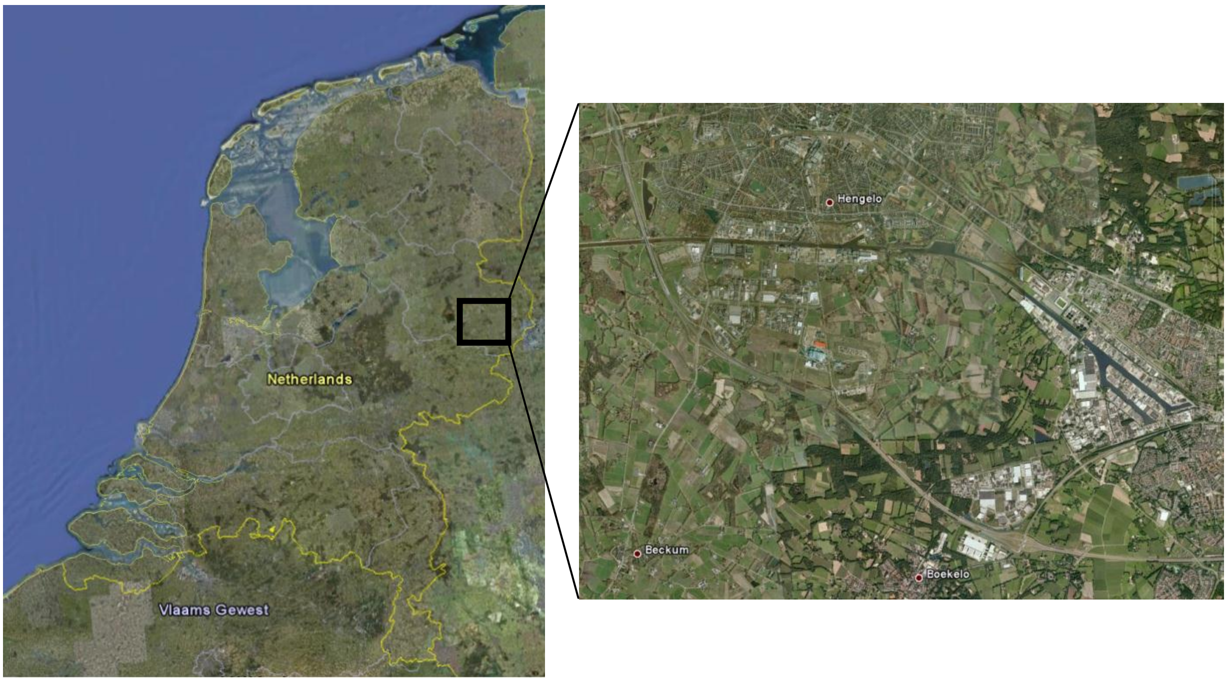
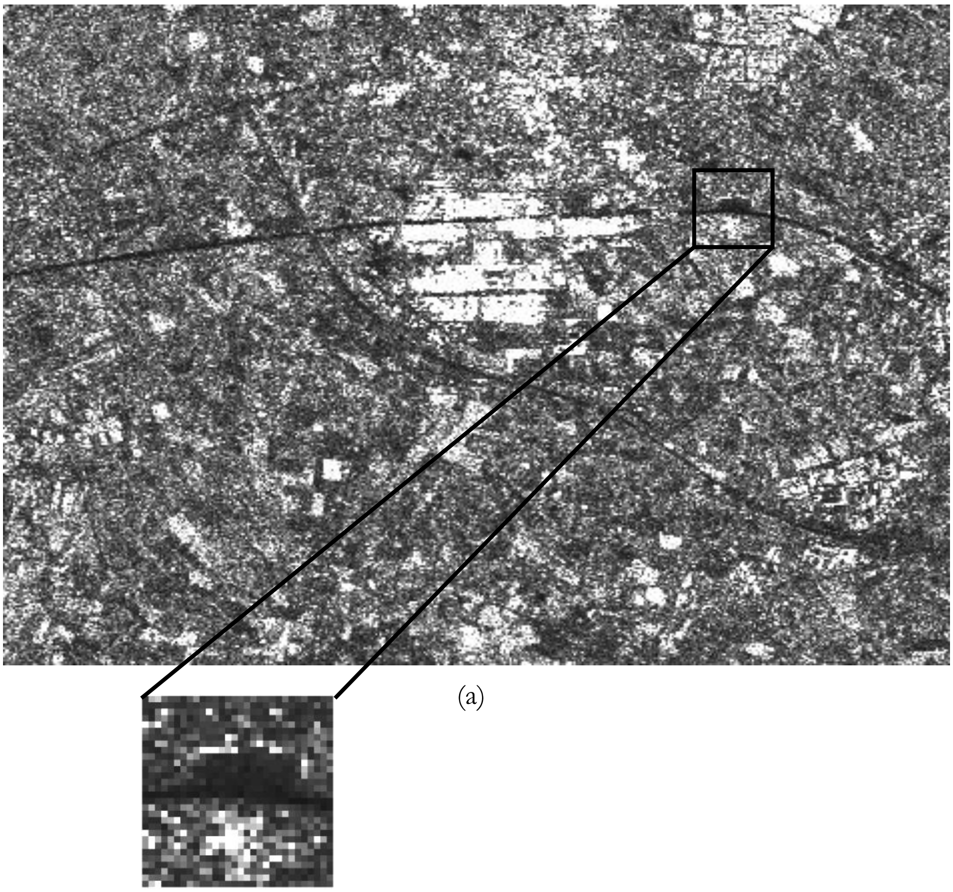


Figure 3-1 Study area, Source: Google Earth

3.1.2. ERS image

European Remote Sensing satellite (ERS1) was launched in 1991. It has an image Synthetic Aperture Radar (SAR), a radar altimeter and powerful instruments to measure surface temperature. Another satellite from ERS, ERS2, was launched in1995 with additional sensor to study about atmospheric ozone. This satellite was built with two specialised radar and an infrared imaging sensor. ERS is useful to monitor natural disaster such as floods and earthquake in elusive parts of the earth (ERS)

The values for radar images appear with several signals that are called Speckle. It shows the reflectance of earth surface as "salt and pepper " in the image which causes problems to interpret the image. To overcome this problem some methods such as multi-looking is used. Multi- looking method can reduce the variance of speckle (Ferretti, et al., 2007; Tough, et al., 1995). The ERS satellite image that used in this research is acquired in year 2002 from the Hengelo. The multi-looking is used in the ERS image to help for visualisation and interpretation the image. The multi-looking process was done with range of 6 and azimuth of 1. One subset of 30×30 pixels was prepared from ERS image in of C-band, the pixel size is approximately 20×20 (Figure3-2).



(a)

(b)

Figure 3-2 The C-band of ERS image (a) Hengelo area, (b) 30×30 pixels subset

3.1.3. Spot-5 image

Spot-5 earth observation satellite was launched in May 2002 from the Guiana Space Centre in Kourou. It is an optical satellite that has two high resolution geometrical (HRG) instruments. Its spatial resolution is 5m and 2.5m in panchromatic band and 10m in multispectral bands. The width imaging swath of this satellite can cover 60×60 km or 60×120 km, that can be asset for application of medium-scale mapping (Spot-5).

The spot image that is used in this study was acquired in 2002 and covers study area. To use this image with ERS for SRM-SVM method, the red band from multispectral bands is selected. The spot image was co-registered with ENVI software, the reference image for co-registering is ERS image. Then from the same area as ERS subset a subset from spot is selected. Figure 3-3 shows the red band of spot image and its subset before and after co-registration.

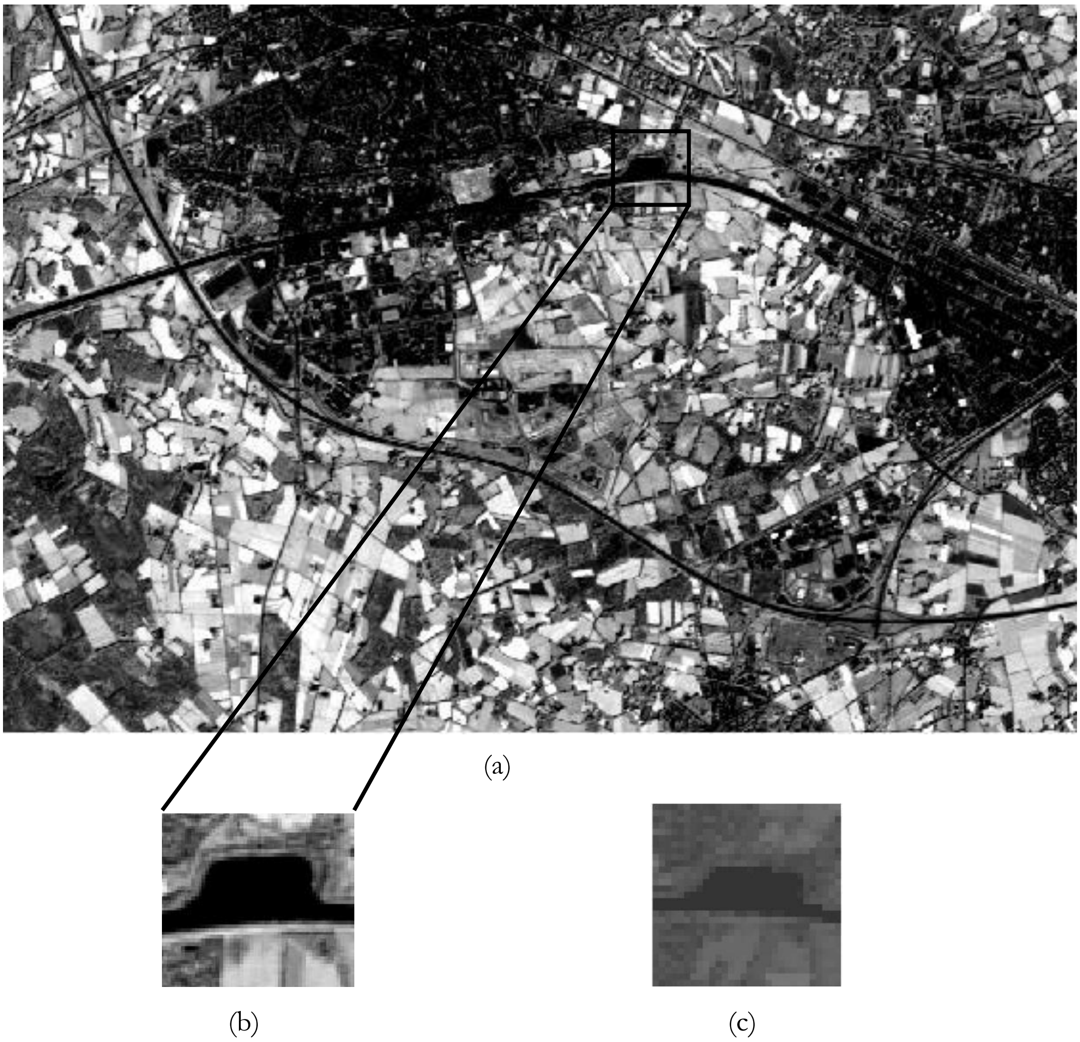


Figure 3-3 The spot image in red band (a) area of Hengelo, (b) subset before co-registeration (c)subset after co-registeration

3.2. Software

3.2.1. The R software

The R software is a programming language for statistical computation and graphics. It is useful for storage facility and it is appropriate for calculating arrays in matrices. The R software has simple programming language for loops, conditional and makes simple input and output ("An introduction to R,"). In this study preparing synthetic data, statistical calculation and preparing some plots is done with R.

Kernlab package in R

The R software has the ability to solve SVM classification. Four package for SVM classification in R were introduced, e1071, kernlab, klaR and svmpath (Karatzoglou, et al., 2006).

Package Kernlab aims to prepare a flexible SVM implementation. It has most of the SVM formulations and kernels. Its kernels are Gaussian RBF, polynomial, linear, sigmoid, Laplace, Bessel RBF, spline, and ANOVA RBF, interested kernel can be select by the user. Kernlab also has the ability to apply multiclass classification and do SVM classification with C-svc or nu-svc. This package uses one against one and one against all multiclass classification. In Kernlab SVM classification is implemented with function ksvm. (Karatzoglou, et al., 2006).

3.2.2. ENVI

ENVI is software to process and analyses the geospatial images. It includes spectral tools and radar analysis. ENVI is written in IDL (Interactive Data Language) that is a programming language to integrate image processing (Banks, 2000). In this research ENVI was used for analysing images, co-registering, selecting subsets and extract training set from images.

4. METHODS

4.1. Super Resolution Mapping with Markov Random Field

Super Resolution Mapping (SRM) is a technique that produces fine spatial resolution classified map from coarser satellite image. The finer resolution pixels are inside the coarse pixel and summation of their value is the same as coarse pixel (Tatem, et al., 2001). After dividing coarse pixel to finer pixels, class label should be assigned to each fine pixel with maximum special dependency.

Let y be the coarse resolution image and x the fine resolution classified map from y. The scale factor between Y and x is S, so if the number of pixels in coarse image is M×N for fine image this number is SM×SN, and each coarse pixel has S^2 number of fine pixels. After applying the scale just the number of pixels will change and the area in both images and the number of bands remains the same. Dimension of image could be set as a matrix with M×N pixels and each coarse pixel identified by b_i that $i = \{1, \dots, \mathrm{M} \times \mathrm{N}\}$. Fine resolution pixels identify as $a_{i|j}$, *i* is the number of coarse pixels in matrix and $j = \{1, \dots, \mathrm{S}^2\}$ is the number of fine pixels, therefore $a_{i|j}$ is *j*th fine pixel belongs to pixel b_i. The relation between x and Y is established as degradation model for pixel b_i is:

$$Y(b_i) = \frac{1}{\mathrm{S}^2} \sum_{i=1}^{\mathrm{S}^2} x(a_{i|j}) \quad 4.1$$

The first step to produce initial SRM map is, divide each pixel with the scale factor S into S^2 fine pixels (sub-pixels). These sub-pixels labelled randomly and do not have correct class label so a method should be applied to rebuild initial SRM and labelling sub-pixels correctly (Kasetkasem, et al., 2005). The process of finding spatial dependency for SRM in this research was done by Markov Random Field (MRF) algorithm. MRF and its combination with SRM is described in detail in the follow sections.

4.1.1. Neighbourhood system

If y set as an image that pixel (i, j) in this image can be indexed as k, where $1 \leq k \leq m$ and $m =$ *M×N* is the number of pixels in the image. So *B* can be defined as a set of sites (Li, 2009):

$$B = \{1, 2, \dots, m\} \quad 4.2$$

Sites on a lattice are spatially regular. For image with size of M×N a rectangular lattice can be defined as:

$$B = \{(i, j) | 1 \leq i, j \leq m\} \quad 4.3$$

The sites in *B* are related to each other with a neighbourhood system. The neighbourhood system for *B* is:

$$N = \{N_r | \forall_r \in B\} \quad 4.4$$

Where N_r is the set of neighbours of pixel, r. Relationship between neighbours has the following properties:

1) A site is not neighbouring to itself: $r \notin N_r$
2) The relationship between neighbouring is mutual: $r \in N_{r'} \leftrightarrow r' \in N_r$

In the neighbourhood system, the first order neighbouring system is defined as four pixels that share the same border with pixel *r*, it shown in Fig4.1.a. The second order neighbouring system, Fig4.1.b contains four pixels that share their corners with pixel *r*. Higher order of neighbouring can be defined in similar way as shown in Fig4.1.c that is up to five order neighbouring. For image *Y* a pixel has four nearest neighbours as (Li, 2009):

$$N_{i,j} = \{(i-1,j),(i+1,j),(i,j-1),(i,j+1)\} \tag{4.5}$$

Pixels at the boundary of image have three neighbours, and at corners has two neighbours.

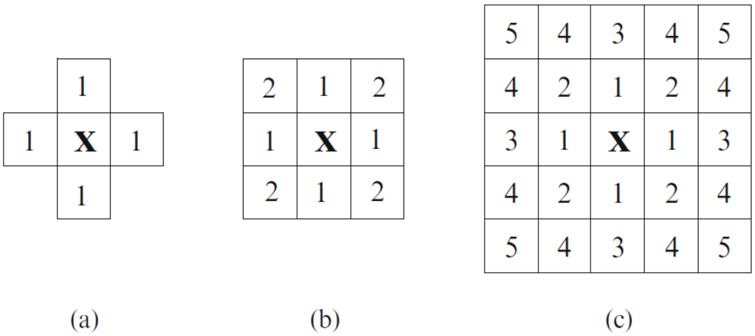


Figure 4-1 Neighborhood on a set of irregular sites source: (Li, 2009)

Neighbourhood order would be changed in SRM related to scale factor. If windows size shown as W_{size} then the relation between window size and scale factor is (Kassaye, 2006):

$$W_{size} = 2(S-1)+1 \tag{4.6}$$

4.1.2. MRF and Gibbs Random Field

Let $x = \{x_1, x_2, \dots, x_m\}$ be a group of random variable on the set of B and each value in x takes a label from label set of *L*. Group X is called random field .If B is an image with m number of pixels then x can be set as DN value of pixels and *L* is the set of class labels. By applying MRF algorithm in classification class labels is assigned to the pixels with their spatial dependency. Markov random field is a random field that used a neighbourhood system and has three following properties (Li, 2009):

1) Positivity: $P(x) > 0$, it can be observed in practice and the joint probability $\mathrm{P(x)}$ for all random fields is uniquely determined by local conditional property.

2) Markovianity: $P(x_r|x_{B-r}) = P(x_r|x_{N_r})$, x_{B-r} denotes all the pixels in the set of B except r and N_r is the neighbouring of pixel r. This property define that labelling of pixel r just depends on its neighbouring pixels.

3) Homogeneity: $P(x_r|x_{N_r})$ is equal for all pixels r, this property defines that conditional property for pixel r given the neighbouring pixels is not related to the location of r in *B*.

MRF is related to Gibbs random field (GRF). Probability density function in GRF is defined as (B. Tso & Mather, 2001):

$$P(x) = \frac{1}{Z} exp\left[-\frac{U(x)}{T}\right] \tag{4.7}$$

Where $U(x)$ is energy function, T is a constant termed temperature and Z is partition function.

$$Z = \sum_i exp\left[-\frac{U(x)}{T}\right] \quad (4.8)$$

Where i in this equation is all possible configuration for x.

Energy function in GRF defines as number of cliques. These cliques are subsets and all pairs of sites are mutual neighbours. Energy function with its clique is:

$$U(x) = \sum_{c\epsilon C} V_c(x) \quad (4.9)$$

With different types of cliques it can be written as:

$$U(x) = \sum_{\{r\}\epsilon C_1} v_1(x_r) + \sum_{\{r,r\prime\}\epsilon C_2} V_2(x_r, x_{r'}) + \sum_{\{r,r',r"\}\epsilon C_3} V_3(x_r, x_{r'}, x_{r"}) + \cdots \quad (4.10)$$

$V_c(x)$ is potential function with respect to clique type C. First order clique:

$$C_1 = \{\{r\}|r \in B\}$$

Second order clique is:

$$C_2 = \{\{r, r'\}|r' \in N_r, r \in B\}$$

And:

$C_3 = \{\{r, r', r"\}|r, r', r" \in B$ are neighbours to one another$\}$

For every MRF there is a unique GRF, however the GRF is defined as cliques on the neighbourhood system. An MRF describes for local properties but GRF is defined for global property of whole image (B. Tso & Mather, 2001).

Posterior energy for image classification

For labelling a pixel, considered contextual information, posterior energy is used. This posterior energy is an objective function and constructed from Bayesian formulation. Context in Bayesian formula is a priori information addition to pixel label that based on pixel DN value. Conditional probability for Bayesian formula for label x_r given the observation c_r in pixel *r* is (B. Tso & Mather, 2001):

$$P(c_r|x_r) \propto P(x_r|c_r)P(x_r) \quad (4.11)$$

By using the definition of Gibbes field in the equation 4-7 the posterior energy can be defined as:

$$U(c_r|x_r) = U(x_r|c_r) + U(x_r) \quad (4.12)$$

Equation 4-7 shows that minimising the energy function $U(x_r)$ is equal to maximising the $P(x_r)$. $U(x_r)$ is called priori energy and mostly is based on pairwise clique potential function that can be written as(Li, 2009):

$$U(x_r) = \sum_{r\epsilon B} V_1(x_r) + \sum_{r\in B} \sum_{r\in N_r} V_2(x_r, x_{r'}) \quad (4.13)$$

If label set just has two labels $L = \{-1,1\}$ then the energy function define as:

$$U(x_r) = \sum_{\{r\}\epsilon C_1} \alpha x_r + \sum_{\{r,r'\}\epsilon C_2} \beta\, x_r\, x_{r'} \quad (4.14)$$

For a single clique C_1, $V_1(x_r)$ is not dependent to label and can be written as:

$V_1(x_r) = \alpha_k$ if label for x_r is k

β is constant that reflect interaction coefficients between r and r' (Li, 2009).

$V_2(x_r, x_{r'}) = -\beta$ If sites on clique r, r' have the same label

$V_2(x_r, x_{r'}) = \beta \; or \; 0$ Otherwise 4.15

So the prior energy is (Li, 2009):

$$U(x_r) = \sum_{r \in B} \sum_{r' \in N_r} V_2(x_r, x_{r'}) \quad 4.16$$

And the posterior energy would be rewritten as:

$$U(c_r|x_r) = U(x_r|c_r) + \sum_{r \in B} \sum_{r' \in N_r} V_2(x_r, x_{r'}) \quad 4.17$$

Class label define by estimating maximum a posterior (MAP) of $P(c_r|x_r)$. This means minimising the posterior energy:

$$\hat{c} = \arg\min_c U(c|x) \quad 4.18$$

4.1.3. SRM

In initial SRM finer image x could classify as an MRF with neighbourhood system $N(a_{i|j})$, and each pixel in the image x assigned as only one class $c(a_{i|j}) = \alpha$, $\alpha \in \{1,2, \dots, L\}$. The prior probability is $P(c)$, the conditional probability that image y is observed with the true SR map is $P(y|c)$. The posterior probability is $P(c|y)$. According to equation 4-7:

$$P(c) = \frac{1}{Z} \exp\left[-\frac{U(c)}{T}\right] \quad 4.19$$

$$P(y|c) = \frac{1}{Z} \exp\left[-\frac{U(y|c)}{T}\right] \quad 4.20$$

$$P(c|y) = \frac{1}{Z} \exp\left[-\frac{U(c|y)}{T}\right] \quad 4.21$$

- **Prior energy**

By using equation 4-10, $U(c)$ is the prior energy and can be written as sum of pair-site interaction:

$$U(c) = \sum_{i,j} U\left(c(a_{i|j})\right) = \sum_{i,j} \sum_{l \in N(a_{i|j})} \omega(a_l) \delta(c(a_{i|j}), c(a_l)) \quad 4.22$$

Where $U\left(c(a_{i|j})\right)$ is the local contribution to the prior energy from pixel $c(a_{i|j})$ and $\omega(a_l)$ is the weight of the contribution from pixel $a_l \in N(a_{i|j})$ to prior energy. $\omega(a_l) = q\varphi(a_l)$, $\sum_{l \in N(a_{i|j})} \varphi(a_l)$=1, $0 \leq q < \infty$ controls the overall magnitude of the weights. Larger value for q cause more smooth results. In $\varphi(a_l)$ an isotropic equation is used which is related to distance between pixels $a_{i|j}$ and a_l , $d(a_{i|j}, a_l)$:

$$\varphi(a_l) = \frac{1}{\Omega} \left(\frac{d(a_{i|j}, a_l)}{r_x}\right)^{-n} \quad 4.23$$

Where Ω is a normalise constants and n is a power-law index and r_x is pixel size in fine resolution map. When $c(a_{i|j}) = c(a_l)$ prior energy is zero and it is equal or larger than 1 otherwise (Tolpekin & Stein, 2009).

So the equation 4-23 can be rewritten as:

$$U(c) = \sum_{i,j} U\left(c(a_{i|j})\right) = q \sum_{i,j} \sum_{l \in N(a_{i|j})} \varphi(a_l) \; \delta(c(a_{i|j}), c(a_l)) \quad 4.24$$

- **Likelihood energy**

By using the assumption of spatially uncorrelated spectral values of Y, likelihood probability is:

$$P(\mathrm{y}|c) = \prod_{i,j} P\left(\mathrm{y}(b_i)\middle|c(a_{i|j})\right) \tag{4.25}$$

If all the classes are normally distributed, it would be:

$$P(\mathrm{y}|c) = \prod_{i,j} \frac{1}{(2\pi)b^{\frac{k}{2}}|C_i|^{\frac{1}{2}}} \times exp(-\tfrac{1}{2}(\mathrm{y}(b_i) - \mu_i)' C_i^{-1}(\mathrm{y}(b_i) - \mu_i)) \tag{4.26}$$

Where C_i is covariance and μ_i is mean for mixing distribution. Likelihood energy is equal to:

$$U(\mathrm{y}|c) = \sum_{i,j} U\left(\mathrm{y}(b_i)\middle|c(a_{i|j})\right) \tag{4.27}$$

- **Posterior energy**

Refer to equation 4-12 posterior energy is:

$$U(c|\mathrm{y}) = q \sum_{i,j} \sum_{l \in N(a_{i|j})} \varphi(a_l)\, \delta(c(a_{i|j}), c(a_l)) + \sum_{i,j} U\left(\mathrm{y}(b_i)\middle|c(a_{i|j})\right) \tag{4.28}$$

To control the contribution of prior a likelihood energy smoothness parameter, λ, is introduced in posterior equation λ is:

$$\lambda = \frac{q}{1+q}$$

So if the equation 4-28 divided to $1+q$ the posterior energy become (Tolpekin & Stein, 2009):

$$U(c|\mathrm{Y}) \propto \lambda \sum_{i,j} \sum_{l \in N(a_{i|j})} \varphi(a_l) \delta\left(c(a_{i|j}), c(a_l)\right) + (1-\lambda) \sum_{i,j} U\left(\mathrm{Y}(b_i)\middle|c(a_{i|j})\right) \tag{4.29}$$

MAP, equation 4-18, is used to find the appropriate class label for pixels. To estimate MAP three algorithms that usually use are, simulated annealing, iterated conditional model and maximier of posterior (B. Tso & Mather, 2001). The number of all possible class labels for all pixels is large. Simulated annealing (SA) is useful method to minimise function, so it is suitable for SRM. Simulated annealing algorithm is explained briefly in next section.

Simulated Annealing Algorithm

Simulated annealing (SA) is a stochastic algorithm for combinational optimization (Li, 2009). It simulated a physical annealing producer that physical material is melted and slowly cooled down to find a low energy configuration. If any x that are random variable on the set of B has the probability:

$$P_T(x) = [P(x)]^{1/T}$$

$T > 0$ is the temperature parameter. When $T \to \infty$, probability is uniform distribution and when $T \to 0$ $P_T(x)$ is on the pick of $P(x)$ (Li, 2009). This algorithm starts with high value of T as initial value then in each iteration this value would be decreased. The iterations will be continued until $T \to 0$. For each pixel r, $U(c|x_r)$ and $U(c|x_{r'})$ and $\Delta = U(c|x_r) - U(c|x_{r'})$ is obtained and if $\Delta < 0$ then $x_{r'}$ is replaced by x_r otherwise another random value for $x_{r'}$ would be selected. The steps repeats again until the system become frozen (B. Tso & Mather, 2001).

4.2. Support Vector Machine

Support Vector Machine (SVM) is a supervised classification method. It uses optimal algorithms to locate best boundary between classes in feature space (Huang, et al., 2002). The boundary is called separating hyperplane and has maximum margin from both classes (Vapnik & Cortes, 1995). SVM just works with pixels that are in the vicinity of classes therefore with small training set it is possible to have accurate classification (Foody & Mathur, 2004b). It also has the ability to work with high dimensional feature space by applying kernel function (Karatzoglou, et al., 2006). Figure 4-2 shows an example of hyperplane between two classes in two dimensional feature space.

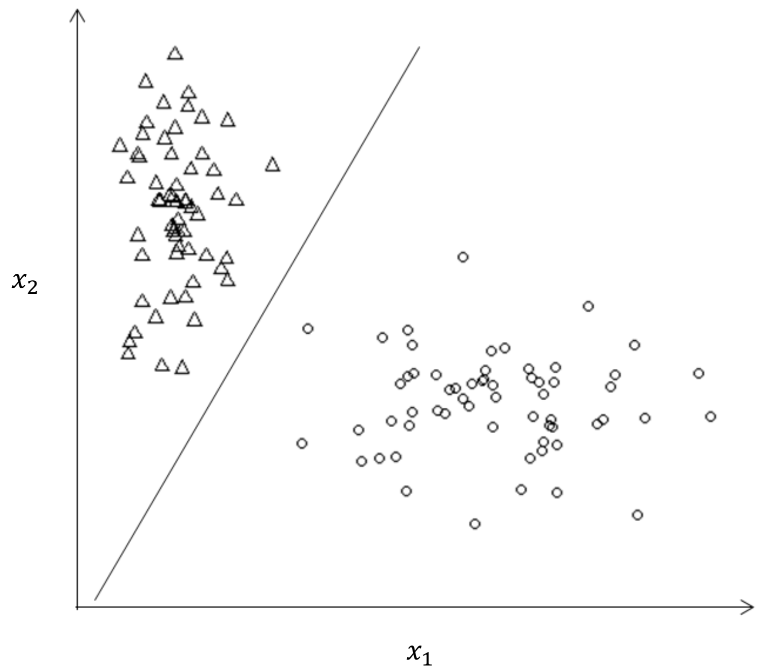


Figure 4-2 2D feature space with two linear separable classes separated

4.2.1. Linear Separable SVM

Suppose that $X = (x_1, x_2, \ldots, x_n)$ $x \in R^n$ are the training samples from two classes in n dimensional feature space with $y \in \{-1, +1\}$ as labels of classes and they can be represented as $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, hence equation for optimal hyperplane between two classes can be written as :

$$f(x_i) = wx_i + b = 0 \qquad 4.30$$

Where w is weight vector and x_i is vector for ith data.

In Figure 4-2 separating hyperplane between two classes is in 2D space and its equation is:

$$w_1x_1 + w_2x_2 + b = 0$$

Two hyperplanes parallel to the optimal hyperplane and on vicinity of boundary pixels is called marginal hyperplanes and for separable data they called hard margins. Figure 4-3 shows marginal hyperplane in 2D feature space. The marginal hyperplanes are in the same distances to optimal hyperplane and the equations for them are:

$$wx_i + b = 1 \quad x_i \in class1 \tag{4.31}$$

$$wx_i + b = -1 \quad x_i \in class2 \tag{4.32}$$

For the data that are not on the marginal hyperline the equations are:

$$\text{If } \ wx_i + b \geq 1 \quad x_i \in class1 \tag{4.33}$$

$$\text{If } \ wx_i + b \leq -1 \quad x_i \in class2 \tag{4.34}$$

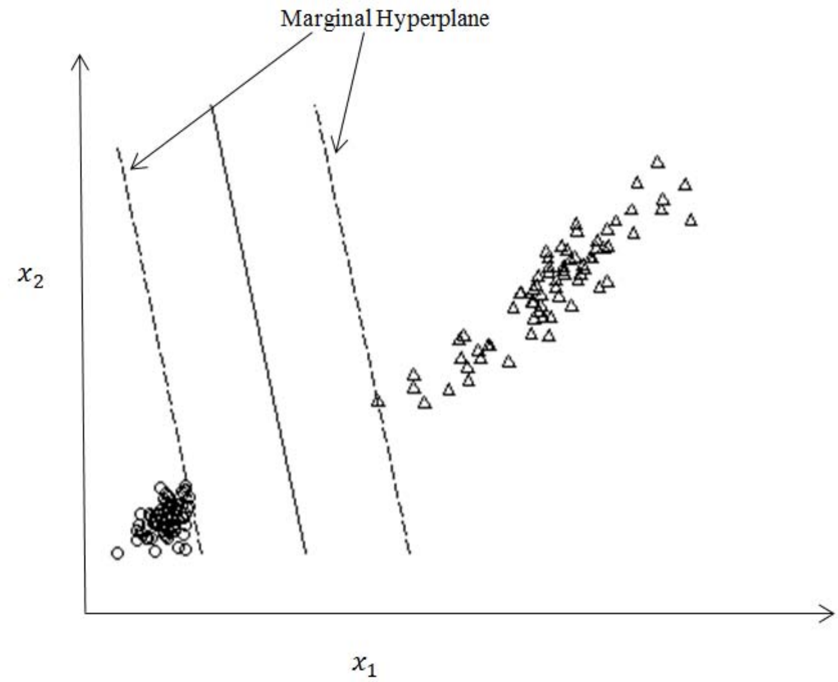


Figure 4-3 Marginal hyperplans between classes

Training data that confirm the equations 4-31 and 4-32 are called support vectors. SVM classification is depends on these support vectors and by eliminating one of them the results for hyperplane will change (Burges, 1998). The perpendicular distances between the marginal hyperplanes is $2/\|w\|$, where $\|w\|$ is the length of weight vector. When this distance is maximize the separating hyperplane has the best position. By maximising the distance the $\|w\|$ should be minimized, minimizing $\|w\|$ is a quadratic programming (QP) problem and could be done by Lagrange multipliers (Richards & Jia, 2006). It should be minimize subject to:

$$f(x_i) = wx_i + b \tag{4.35}$$

To give:

$$\mathrm{L} = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{k} \alpha_i f(x_i) \quad 4.36$$

$$\mathrm{L} = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{k} \alpha_i \left(y_i(wx_i + b) - 1\right) \quad 4.37$$

Where L is Lagrangian, $\alpha_i \geq 0$ are Lagrange multipliers and k is the number of support vectors. Each data in training set has one α and it is zero for all data except support vectors (Burges, 1998). To find the values of w and b that minimise L the following step would be done:

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{k} \alpha_i y_i x_i = 0$$

$$w = \sum_i \alpha_i y_i x_i \quad 4.38$$

And:

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{k} \alpha_i y_i = 0$$

$$\sum_{i=1}^{k} \alpha_i y_i = 0 \quad 4.39$$

Equation (4.37) can be rewritten as (Richards & Jia, 2006):

$$\mathrm{L} = \frac{1}{2}\left(\sum_{i=1}^{k} \alpha_i y_i x_i\right)\left(\sum_{j} \alpha_j y_j x_j\right) - \sum_{i=1}^{k} \alpha_i \left[y_i\left(\left(\sum_{j} \alpha_j y_j x_j\right) x_i + b\right) - 1\right]$$

The simpler way to write this equation is:

$$\mathrm{L} = \sum_{i=1}^{k} \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j x_i x_j$$

The $\alpha_i$that are not equal to zero lie on the marginal hyperplans so $\alpha_i \neq 0$ are corresponds to support vectors and shown as α_i^0 (Vapnik & Cortes, 1995). If α_i^0 put in equation 4-38 optimal training vector would be obtained as (Richards & Jia, 2006):

$$w = \sum_{i=1}^{k} \alpha_i^0 y_i x_i \quad 4.40$$

Support vectors are on marginal hyperplanes, these values are in the equations 3.31 and 3.32 that can write as:

$$y_i(wx_i + b) - 1 = 0 \quad 4.41$$

So the value for b is obtained from k number of support vectors:

$$b = \frac{1}{k} \sum_{i=1}^{k} w . x_i \qquad (4.42)$$

The decision rule for classification of pixel x_i in SVM method is:

$$g(x_i) = \operatorname{sgn}(w^0 x_i + b^0) \qquad (4.43)$$

4.2.2. Non-Separable SVM

If data are separable equation 4-45 can be used for classification with support vector machine, but to deal with non-separable training sets, SVM classification will have error with that equation, so a penalty value for misclassify errors and non-negative variables ξ_i are introduced (Figure 4-5)(Huang, et al., 2002). This variables define the distance of the data from marginal hyperplane that passed through support vectors of the same class, marginal hypeplanes in this case are called soft marginal (Foody & Mathur, 2004b).

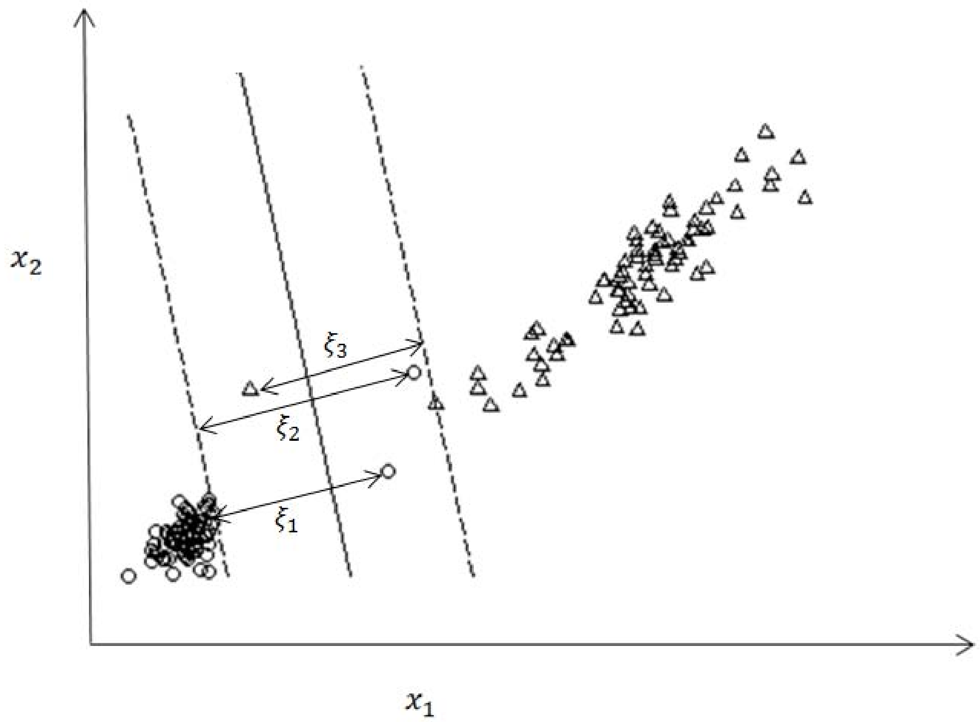


Figure 4-4 Error for non-separable data in SVM (ξ_i is the error)

$$\xi_i \geq 0 \, , i = 1, \dots, n \qquad (4.44)$$

With error values the equations 4-33 and 4-34 would be changed to:

$$wx_i + b \geq 1 - \xi_i \qquad \text{If } x_i \in \text{class1} \qquad 4.45$$

$$wx_i + b \leq -1 + \xi_i \quad \text{If } x_i \in \text{class2} \qquad 4.46$$

By minimizing ξ_i subset of minimal training errors would be found. To separate training set without errors these subset is excluded from dataset. A new optimal separating hyperplane combined with these error s required. To find that hyperplane following functional minimize (Vapnik & Cortes, 1995):

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \qquad 4.47$$

The first part of the equation is for maximising the margin and the second part is for penalizing data that are in the wrong side of separable hyperplanes. The basic concepts of SVM is to find a balance between maximising of the margin and minimising the training errors (Hsu & Lin, 2002). The value C is a constant for penalty value of misclassification error this parameter controls the magnitude of the errors with data that are in the wrong side of hyperplane. The value C is selected by user, if it is chosen very small then predictor function is simple and if it is selected very big the analysis will over fit training data (Foody & Mathur, 2004b). To find hyperplane, Lagrange multipliers is used (Hastie, et al., 2003):

$$\min_{w,\xi}\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$$

Subject to:

$$\xi_i \geq 0 \quad , \quad y_i(wx_i + b) \geq 1 - \xi_i$$

And the Lagrange function is:

$$\mathrm{L} = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{k}\alpha_i\left(y_i(wx_i + b) - (1 - \xi_i)\right) - \sum_{i=1}^{n}\mu_i\xi_i \qquad 4.48$$

Where μ_i are positive constrains to enforce variable ξ_i be positive (Burges, 1998). The dual problem will be:

$$\mathrm{L} = \sum_{i}^{k}\alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j x_i x_j$$

It should be maximized subject to:

$$0 \leq \alpha_i \leq C \quad \text{and} \quad \sum_i \alpha_i y_i = 0$$

4.2.3. Non-linear SVM

If data are not linearly separable, SVM algorithm that explained in the previous sections cannot be applied for SVM classification. To solve the problem of nonlinear separability, input values transform to higher dimensional feature space H with the function ϕ (Vapnik & Cortes, 1995).

$$R^k \rightarrow H\,, x_i \rightarrow \phi(x_i)$$

As the dataset x_i is transformed to higher dimensional space and moreover working with ϕ in H is complicate so training algorithm can only be done by dot product from $\phi(x_i)\cdot\phi(x_j)$. Now if there is a kernel function that (Huang, et al., 2002):

$$k(x_i, x_j) = \phi(x_i) \cdot \phi(x_j) \qquad 4.49$$

Then instead of using dot product kernel function $k(x_i, x_j)$ can be used.

Non-linear SVM classification has the same properties and equations as linear SVM. The equation 4-35 for the hyperplane in new feature space is:

$$w \cdot \phi(x_i) + b = 0 \qquad 4.50$$

So the other equations change to:

$$\mathrm{L} = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{k} \alpha_i \left(y_i(w \cdot \phi(x_i) + b) - 1\right) \qquad 4.51$$

And:

$$\mathrm{L} = \sum_{i=1}^{k} \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j k(x_i, x_j)$$

$$w = \sum_i \alpha_i^0 y_i \phi(x_i) \qquad 4.52$$

$$f(x_i) = \sum_{i=1}^{k} \alpha_i y_i k(x_i, x_j) + b^0 \qquad 4.53$$

And the decision function is:

$$g(x) = \mathrm{sgn}\left(\sum_{i=1}^{k} \alpha_i y_i k(x_i, x_j) + b^0\right) \qquad 4.54$$

4.2.3.1. Kernel functions

Two of popular kernel functions in remote sensing and also SVM classification method are:

- The linear kernel implementing the simplest of all kernel functions:

$$k(x_i, x_j) = (x_i, x_j) \qquad 4.55$$

- The Gaussian Radial Basis Function (RBF) kernel:

  usually RBF kernel is used when there is no prior information about data (Karatzoglou, et al., 2006).

$$k(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right) \qquad 4.56$$

4.2.4. Distance to hyperplane in SVM classification

In order to incorporate SVM with SRM, mixture probability should be formulated with SVM outputs Probabilistic output from SVM was done in previous researches, but they didn't consider mixture probability. To find mixture probability with SVM in this research distance from hyperplane, histograms and interpolation were used.

As mentioned before SVM classification based on separating hyperplane with equations 4-35 for linear support vector and with equation 4-52 for non-linear support vector. Distance between the separating hyperplane and data x_i can be calculated and from decision function of SVM class label for x_i would be obtained, thus with distance and class label probability that x_i belongs to which class can be estimated. If the equation of hyperplane is $f(x)$ then distance of data x_i from the hyperplane is:

$$\text{Distance}(x_i) = \frac{f(x_i)}{\sqrt{\|w\|^2}} \tag{4.57}$$

Where w is the weight of hyperplane.

If linear SVM classification is used weight and bias for hyperplane could be find simply and use them for compute the distances, but in non-linear SVM classification method w is related to $\phi(x)$. Since computing $\phi(x)$ is difficult, to find the weight objective function and Lagrange multipliers in equation 4-37 used in the following way:

$$L = \frac{1}{2}\|w\|^2 - \sum_{i=1} \alpha_i \; y_i f(x_i)$$

As i is for support vectors, and for support vectors $y_i f(x_i) = \pm 1$ so:

$$\|w\|^2 = 2(L + \textstyle\sum_i \alpha_i) \tag{4.58}$$

Distance between x_i and hyperplane $f(x_i)$ is:

$$\text{Distance}(x_i) = \frac{f(x_i)}{\sqrt{2(L + \sum_i \alpha_i)}} \tag{4.59}$$

Formulating distance with equation 4-61 is helpful to calculate distance from separating hyperplane with any kind of kernel function that is used in SVM classification.

4.3. Linear Interpolation

Interpolation is a method to find interested values of a new data set that is subset of known values. If (a_k, b_k), $k = \{1, \ldots, n\}$, there is only one polyline that goes through that set of points. This polyline is called interpolating polyline and write as:

$$I(a_k) = b_k, k = \{1, \ldots, n\}$$

Now if there is another point a_i, and it is between two points a_m, a_{m+1} of dataset, then the value b_i for that point can be calculated as (Moler, 2004):

$$I(a_i) = b_i = \frac{(a_i - a_m)(b_{m+1} - b_m)}{(a_{m+1} - a_m)} + b_m \tag{4.60}$$

Where b_m and b_{m+1} are values related to a_m and a_{m+1}. The equation 4-62 is called linear interpolation equation.

Figure 4-5shows the linear interpolation between a_m and a_{m+1}, these points can be connected to each other with a straight line and as the slope is equal for the line through the points, any other value can be find with equation 4-62.

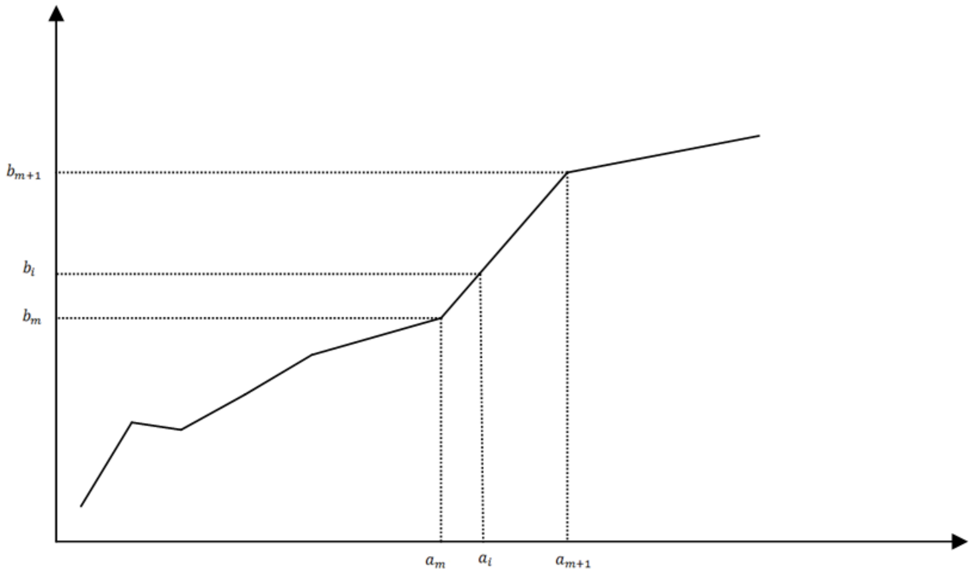


Figure 4-5 Interpolation a_i that is between a_m and a_{m+1}

5. IMPLEMENTATION

This chapter discuss about how methods that explained in Chapter four applied to obtain the objective of the research. Preparation Synthetic data is described in section 5.1. Section 5.2 describes the adopted method for SVM mixture probability with histograms and in section 5.3 combining SVM mixture probability with MRF-SRM algorithm is explained. (In this research MRF-SRM is the MRF based SRM method that uses normal assumption and SRM-SVM is the MRF based SRM that uses SVM in its assumption.)

5.1. Synthetic Images

Synthetic images allow the user to introduce parameters and the number of classes in the most appropriate way for the research. These images are extracted from a real image. The purpose of the research is to use different class distributions while having control on its parameters, thus synthetic images are generated to test the proposed method before applying the method on the remote sensing image.

DN values for pixels are generated from random values of classes, based on the reference image. The reference image is a Google image from an agricultural area in Flevoland, the Netherlands. This image has 60×60 number of pixels Figure 5-1 shows the reference image and reference landcover map that prepared from reference image.

Figure 5-1 (a) Google map image of Flevoland, the Netherlands source: Google map (b) Reference landcover map for two classes

Two synthetic images were generated with different class distribution, and different number of bands. The *synthetic image1* contains two bands and two classes with normal distributions, and the *synthetic image2* has two classes with exponential distributions and one band. *Synthetic image2* generated in one band to simulate radar image that is going to be use later on. The distribution of classes for *synthetic image2* is presented in Figure 5-2.

To prepare these synthetic images the R programming language was used in order to control certain statistical parameters.

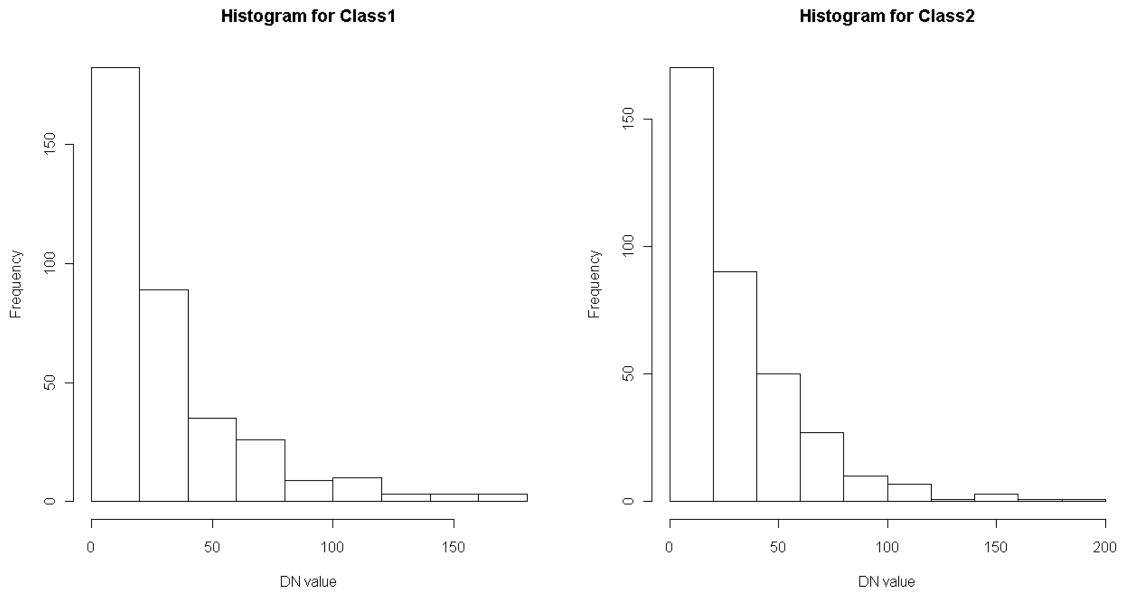
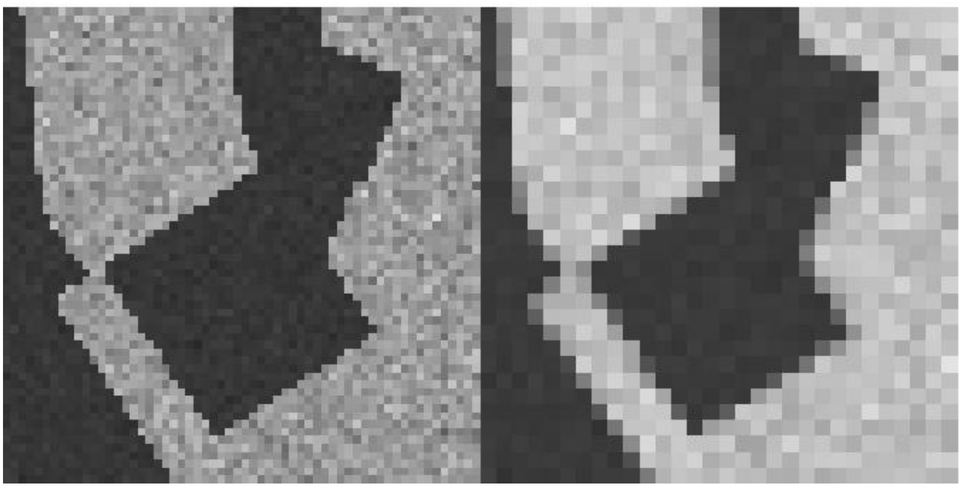


Figure 5-2 Exponential Distribution of classes in Synthetic image2 with one band

5.1.1. Synthetic image for SRM

As in SRM method import data is coarse resolution and final results is in fine resolution, to apply the SRM method, synthetic images are generated as fine resolution images, then coarse resolution images are prepared by applying spatial degradation with different scale factors in both synthetic images. The purpose to prepare fine image as the reference is to compare it with the results of the SR map. If S is the scale factor for SRM, each S^2 number of fine pixels is degraded to produce one coarse pixel. An example of degradation with $S = 2$ is shown in the Figure 5-3.



(a) (b)

Figure 5-3 Degradation of reference synthetic image, S=2 (a) Fine resolution image (b) Degraded image

5.2. Implementation of SRM-SVM

5.2.1. Mixture probability in SVM classification

As mentioned in Chapter four, SVM is a hard classification method which by decision function, equation 4-43, the class label of the data will be known. In order to estimate mixture probability, it is necessary to obtain the probability from SVM output results. To achieve this purpose a set of training data from pure classes was generated with parameter of classes in the *synthetic image1*. Each coarse resolution pixel has S^2 number of fine resolution pixels so if there is just two classes, a coarse mixed pixel in SRM can have $1 + S^2$ different proportion of that classes, therefore with scale factor 2, five set of mixed pixels can be prepared manually from training data.

Figure 5-4 shows the feature space for five proportions of two classes with scale factor two. As it is shown in the figure, the distance between mixed pixel and the SVM separating hyperplane changes related to the proportion of each class in the mixed pixel, moreover the mixed pixels are more near to the class with larger proportion. The probability that one pixel belongs to each class depends on the distance of the pixel from the separating hyperplane, so the distance to separating hyperplane can help to find the conditional probability.

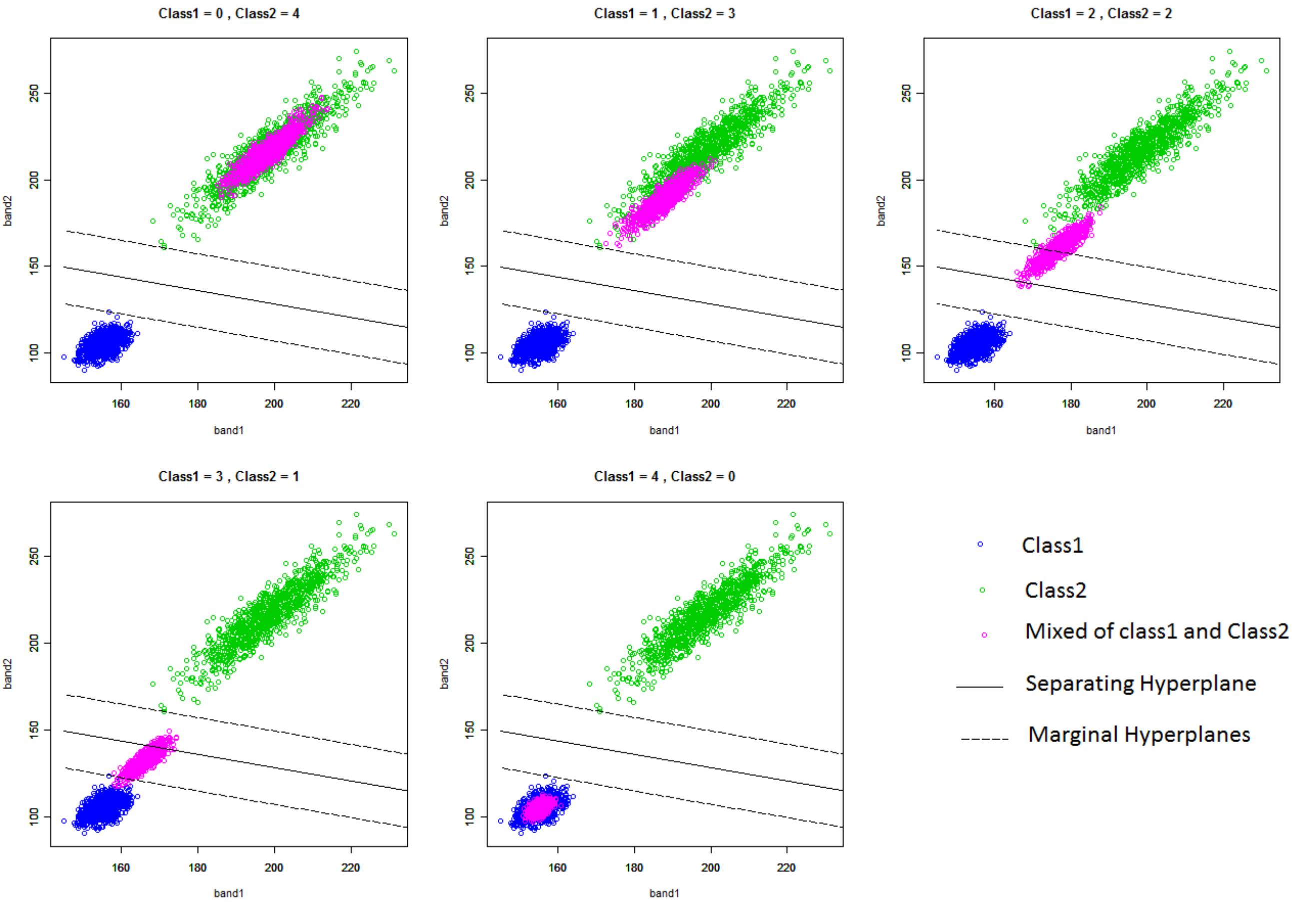


Figure 5-4 Feature space of SVM lines for two classes and mixed classes in five different proportion, S=2

In section 4.2.4 it was shown that the distance to the separating hyperplane can be calculated by using equation 4-59. The distances of the training data from SVM separating hyperplane are calculated with that equation and the set of distances for class1 is named as D_{c1} and for class2 is named as D_{c2}.
Distance is a mathematical concept that is always positive. In this research to define that from which side data is going to the separating hyperplane, the distance of pixel in one side of hyperplane is set as positive and in the other side is set as negative. Furthermore to find the mixture probability with distances, histograms and its concepts are used.

Histograms for distance

To obtain the histogram of mixture classes histograms for D_{c1} and D_{c2} are prepared. To mix these histograms, they prepared with the same number of bins. Selecting the same number of bins is possible in R software. As the histograms have different variance, the width of the bins is different between two histograms. Therefore to calculate the probability of classes, the width of the bins is multiplied to the density of each bin that extracted from histograms. If the width of histogram D_{c1} is shown as W_{d1} and the width of D_{c2} is shown as W_{d2}, then the width of bins in each histogram is:

$$W_{b1} = W_{d1} / \text{(number of bins)}$$

And : 5.1

$$W_{b2} = W_{d2} / \text{(number of bins)}$$

Where W_{b1} and W_{b2} are the width of bins in each histogram. The probability can be shown as:

$$P_{i1} = W_{b1} dc_{1i}$$

$$P_{i2} = W_{b2} dc_{2i}$$

Here dc_{1i} and dc_{2i} is the frequency in the i th bin of histograms.
In order to find the mixture probability, the probability of ith bin from D_{c1} (P_{i1}) was mixed with ith bin of D_{c2} (P_{i2}).

$$P_{im} = \theta_1 P_{i1} + \theta_2 P_{i2} \tag{5.2}$$

Furthermore the mid of each bin in histogram was extracted and mixed as:

$$M_{im} = \theta_1 M_{i1} + \theta_2 M_{i2} \tag{5.3}$$

Where θ_1 and θ_2 are the proportions of two classes in the pixel and they related to each other as:

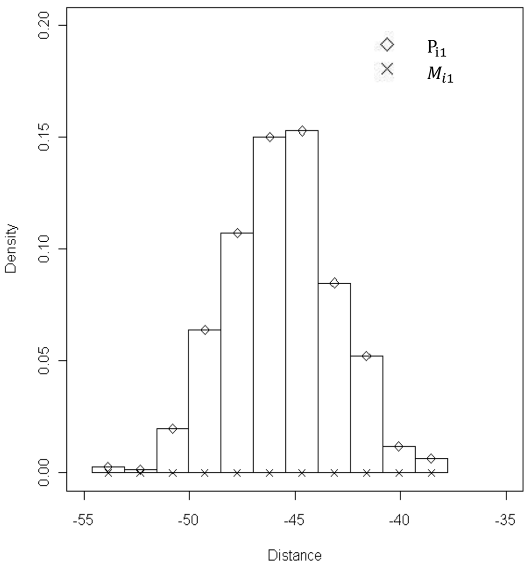
$$\theta_1 + \theta_2 = 1$$

The P_{im} are the values for the mixture probability it shows $P(\text{Distance}|\theta_1, \theta_2)$, M_{im} show the position of mixture bins. There is $1 + S^2$ number of estimated (M_{im} , P_{im}) that is obtained from the histograms of D_{c1} and D_{c2}. To calculate mixture probability for a mixed pixel, interpolation can be done with the values (M_{im} , P_{im}) and by assigning the distance of mixed pixel as the known value and the mixture probability as the unknown value.
Briefly to use the histograms for calculating mixture probability of pixel y_i with SVM classification the following should be done:

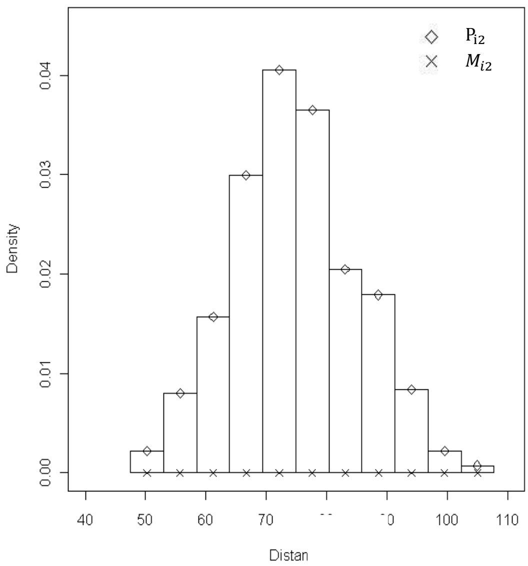
- Apply SVM classification to find hyperplane, D_{c1} and D_{c2}
- Calculate the distance of y_i from the hyperplane
- Calculate (M_{im} , P_{im}), with the proportion of classes in that pixel
- Set (M_{im} , P_{im}) as values for interpolation
- Set the distance of y_i as a known value and its probability is an unknown value
- Put distance of y_i in equation 4-60 and calculate the mixture probability related to y_i

The method was applied on two sets of distributions, one with class distribution of *synthetic image1* and the other with class distribution of *synthetic image2*.

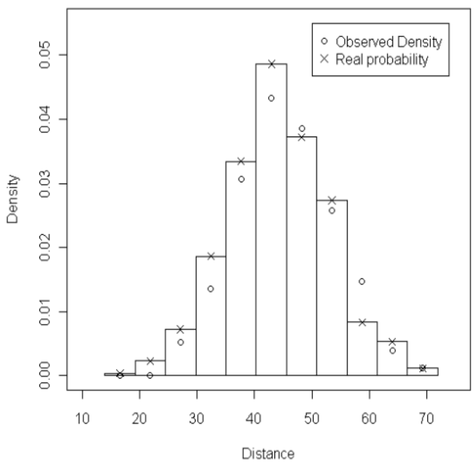
An illustration of this method is shown in Figure 5-5 that the histograms for D_{c1} and D_{c2} and the distances for mixed classes with the S=2 and the proportion of 25% for class1 and 75% for class2. The parameter of classes is the same as classes in *synthetic image1* with normal distribution. In plot (c) the estimated density for all mids of histogram with the above method and the density of histogram performed.



(a)



(b)



(c)

Figure 5-5 Histogram for distances S=2 (a) Distances for class1 D_{c1} (b) Distances for class2 D_{c2} (c)Distances for mixed pixels 25% of class1 and 75% of class2

5.3. SRM-SVM

After defining the above method and find the mixture probability with SVM, the method should be incorporated with SRM-MRF. The algorithms to apply MRF-SRM on the image were discussed in section 4-1 of the Chapter four. How to use SVM mixture algorithm in MRF-SRM is described in detailed in the following section.

5.3.1. Prior Energy

Prior energy in MRF-SRM is related to cliques and neighbourhood systems so does not affect by distribution of classes. The size of windows for neighbourhood system is assigned with the equation 4-6 and the equation 4-22 is used as prior energy for SRM-SVM.

5.3.2. Likelihood energy

Likelihood energy in MRF-SRM is computed with equation 4-27. The equation needs mixture probability for different proportion of classes in mixed coarse pixels. Algorithm for mixture model with SVM is based on the distances of data from separating hyperplane (section 4.2.1). Therefore when SVM classification is used in MRF-SRM instead of value of y its distance from separating hyperplane should be used. So the equations 4.1 for SRM would be changed to:

$$D_{y(b_i)} = \frac{1}{S^2}\sum_{i}^{S^2} d_{x(a_{i|j})} \tag{5.4}$$

Where D_Y is distance of pixel y from the separating hyperplane and $d_{x(a_{i|j})}$ is distance of jth fine pixel in ith coarse pixel from the separating hyperplane. Conditional probability also is changed from $P(y|c_i)$ to $P(D_y|c_i)$. So likelihood energy for MRF-SRM incorporated with SVM is:

$$U(\mathrm{D}_y|c) = \sum_{i,j} U\left(D_{y(b_i)}\middle|c(a_{i|j})\right) \tag{5.5}$$

5.3.3. Posterior energy

Posterior energy is related to prior and likelihood energy, according to change likelihood energy to equation 5-5, the equation 4-29 for posterior energy would be rewritten as:

$$U(c(a_{i|j})|D_{y(b_i)}) = \lambda U(c(a_{i|j})) + (1-\lambda)\sum_{i,j} U\left(D_{y(b_i)}\middle|c(a_{i|j})\right) \tag{5.6}$$

Smoothness parameter λ is between 0 and 1, if it is assigned as 0 the prior model is completely ignored. To optimize λ , the method is applied on synthetic images with different combination of scale factor and smoothness parameter. Appropriate label for each pixel is assigned with maximum a posterior algorithm.

5.3.4. Simulated annealing

To find the appropriate class for pixels in SR map, simulated annealing is used to find maximum a posterior (section 4.1.3). In this algorithm after each iteration the proportion of classes for coarse pixel would be changed. However the DN value for coarse pixel remains the same, so the distance of that pixel from separating hyperplane is the same during iterations. It means that the conditional probability $P(\text{Distance}|\theta_1,\theta_2)$ is different during the iteration.

In order to update the conditional probability, from the histograms of pure training set, P_{im} and M_{im} is computed for new proportions. With the method that explained in 5.2.1 the mixture probability is updated with new proportion for after each iteration.

5.4. Accuracy assement

The last step for implementation is to validate the results of the SVM mixture probability and SRM-SVM application. Accuracy is defined as the level of agreement between observed data and reference data.
For mixture probability accuracy assessment is done by finding root mean square error (RMSE). RMSE calculates the distance between estimated and real data.
In term of classification, accuracy assessment is to compare the class labels assigned with current method and label of pixels in reference data, usually reference data is from a ground truth labelling. By comparing the labels percentage of pixels that labelled correctly can be estimated. The result of accuracy is shown with error matrix or confusion matrix. Dimension of this matrix is $nc \times nc$, where nc is the number of classes. The row of matrix is for observed labels and the column is reference data labels (Richards & Jia, 2006). From error matrix overall accuracy defined as the number of pixels with correct label divided by total number of test pixels. Kappa coefficient is another accuracy value which is defined from error matrix. It is observed with sum of overall columns for each raw and sum of overall rows for each columns (Richards & Jia, 2006).
In this research accuracy assessment for SRM was done by overall accuracy, RMSE and kappa coefficient.

6. RESULTS

This chapter presents the obtained results from the process of SRM-SVM. Section 6.1 explains the results from application of SVM classification, methods that are going to use in SVM classification and SVM mixture probability. Section 6.2 is about the results obtained from applying SRM-SVM method on synthetic images and remote sensing mage.

6.1. Exprimental results from SVM mixture probability

To understand and illustrate the capability of SVM classification before apply the SRM-SVM method on the images, some parameters of SVM classification are studied in section 6.1.1 and results of the application on mixture probability studied in section 6.1.2

6.1.1. SVM classification

Training set is a sample of pure pixels from the user defined classes. Training set is used in supervised classification method before digital processing to identify how is the structure of each class (Schott, 2007). Since SVM is a supervised classification, it is necessary to select appropriate training set before applying classification method on the data.

According to the researches that have been done before in SVM classification method, just the support vectors in the training samples are needed and the other training samples do not affect classification results. Therefore SVM method has the ability to give accurate results even with small number of training samples, but there is a positive relation between the number of training samples and accuracy of classification (Foody & Mathur, 2004b). If the boundary of training samples is small, the obtained accuracy would be low and selecting huge training set is time consuming for SVM classification (Goumehei, 2010). According to (Goumehei, 2010) the number of training set in this research selected in the range of 30 to 1000 training samples.

After defining number of training data the pixel size of training in SRM should be known, as discussed before SRM is a method that produce fine resolution pixels from coarser pixel images, preparing fine resolution pixels does not change the number and kind of classes in the image. Training set is usually selected from pure pixels of input image, but in SRM after generating initial SRM there is possibility to choose the training set either from initial SRM with fine resolution pixels or from the original image with coarse resolution pixels. After applying SRM mean value for coarse and fine resolution will remain the same but related to scale the covariance is different in fine and coarse data. SVM is a non-parametric classification method and does not use class parameter so either of course or fine training sets that obtain SVM separating hyperplane with higher accuracy can be used as training set.

In order to find more accurate SVM hyperplanes two set of training data from coarse and fine resolution are generated with class parameter of *synthetic image1*. The first training data is contain 1000 samples of fine resolution data for each class, DN value of these pixels is prepared with the random generation then SVM classification is done on this fine data set and the separating hyperplane from the result of this classification is obtained which is called SHF (separating hyperplane of fine resolution).

To define coarse training dataset the scale factor set as two, thus each coarse pixel has four fine pixels coarse resolution pixels are generated by degradation of four fine pixels. SVM classification also is done on coarse training data and the obtained separating hyperplane for this set of data is called SHC (separating hyperplane of coarse resolution). In all classifications the linear kernel chose for application of SVM. Figure 6-1 shows the position of SHC and SHF in feature space related to each other. SHC is shown as the blue line and SHF is shown with the green line. The distance between norms of the lines is 0.584 and as it observed from the results of weight and bias for SHC and SHF in table 6-1, both lines are in the same direction. Difference between marginal distances in two SVM applications is because the spectral variance in coarse and fine resolution data is different.

Table 6-1 Wight and bias for the separating hyperplanes of fine resolution and coarse resolution data sets

	Weight	Bias	Marginal distance
Coarse resolution	(7.272e-3, 2.725e-2)	5.22	85.89
Fine resolution	(1.752e-2, 7.742e-2)	13.37	24.57

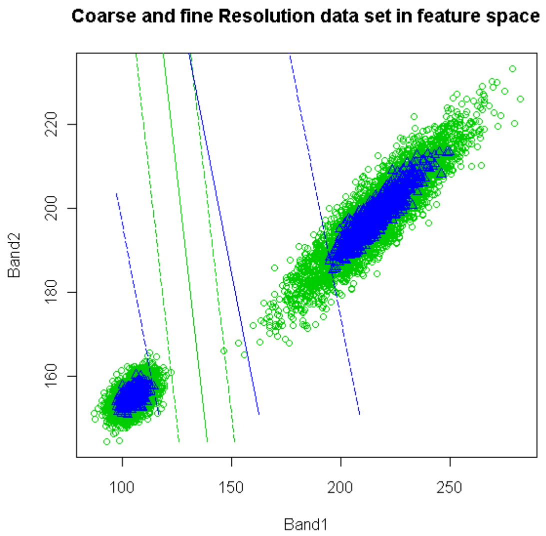


Figure 6-1 Feature space that compare SHC and SHF, in this plot coarse data and SHC present with blue color and fine data and SHF are green

In SRM-SVM algorithm separating line is important in mixture modelling so in order to find more accurate separating hyperplane, different numbers of training data with S=6 were prepared, by degrading fine resolution data, furthermore classification of coarse pixels was done by SHF and classification of fine pixels was done by SHC for each set of those data. Accuracy of this practice was obtained by overall accuracy and the results shown in Table 6-2.

Table 6-2 Experimental results of comparing coarse pixels training set and fine resolution training set with normal distribution classes, S=6

Number of samples for each		Overall accuracy	
fine resolution	Coarse resolution	fine pixels with SHC	coarse pixels with SHF
720	20	100%	100%
1800	50	99.97%	100%
2160	60	99.97%	100%

To apply the method on non-normal distribution classes, fine resolution data prepared with the same parameter as *synthetic omage2*. Coarse resolution pixels are generated with scale factor 6 by degradation of fine pixels. SVM classification applied on both set of data in the same way as described for normal distribution classes in last paragraph. The SVM hyperplanes obtained for both training data and the results of SVM classification for coarse resolution training data obtained from SHF of exponential and the results of SVM classification for fine resolution training data obtained from SHC of exponential, the result of these applications is shown in Table 6-3.

Table 6-3 Experimental results of comparing coarse pixels training set and fine resolution training set of exponential distribution S= 6

Number of samples for each		Overall accuracy	
fine resolution	Coarse resolution	fine pixels with SHC	coarse pixels with SHF
720	20	93.61%	100%
1800	50	96.86%	100%
2160	60	95.92%	100%

Table 6-2 and table 6-3 show that classifying coarse pixels with SHF has small error, because the difference in variance of coarse and fine data. Since the hyperplanes are in the same direction coarse data are always in the correct side of SHF. Overall accuracy for classifying fine pixel as data with SHC is high therefore the training set can be either of coarse resolution pixels or fine resolution pixels. In this research training set selected from coarse resolution data.

6.1.2. Experimental result of application SVM mixture probability

Mixture probability is used in likelihood energy (discussed in section 4.1.3). The appropriate method for SVM mixture probability was determined by histograms and interpolation methods (see section 5.2.1). To improve the method and validate its results, training set of pure coarse pixels with scale factors,$S = 2,3,6,10$ were generated by degradation from same classes in fine resolution.

Additionally a set of mixed pixels for each scale were prepared from fine resolution data. If scale factor is S, each coarse pixel can have $(1 + S^2)$ different proportions of two classes, so for each scale there is $(1 + S^2)$ sets of mixed pixels. For each of these sets with the equation 4-59 distances from separating hyperplane obtained from their relevant training set were computed, so $(1 + S^2)$ sets of distances are obtained from mixed pixels in each scale. These set of distances are considered as reference data and from their histograms reference probability and their mids extracted.

Furthermore the mixture SVM model that was explained in section 5.2.1 applied on the distances of pure training data in each scale and the set of mids and density are obtained. The mixture probabilities for the mids of the reference histogram is calculated and assigned as observed probabilities.

The above method applied on classes of both synthetic images and in different scales for each image, and the RMSE for observed data and referenced data calculated. Figure 6-2 (a),(c),(e) shows the results of RMSE for normal distribution classes in scales 2, 3 and 6, and Figure 6-2 (b),(d),(f) is the results for classes with exponential distribution in scales 2, 3 and 6. As it shown in the results the RMSE in both cases is acceptable and the variety of RMSE is low, in the plots θ_1 is the proportion of class1 in the mixed pixels (the proportion of class2 is 1- θ_1). The results show that the method is accurate to use in likelihood energy of MRF-SRM.

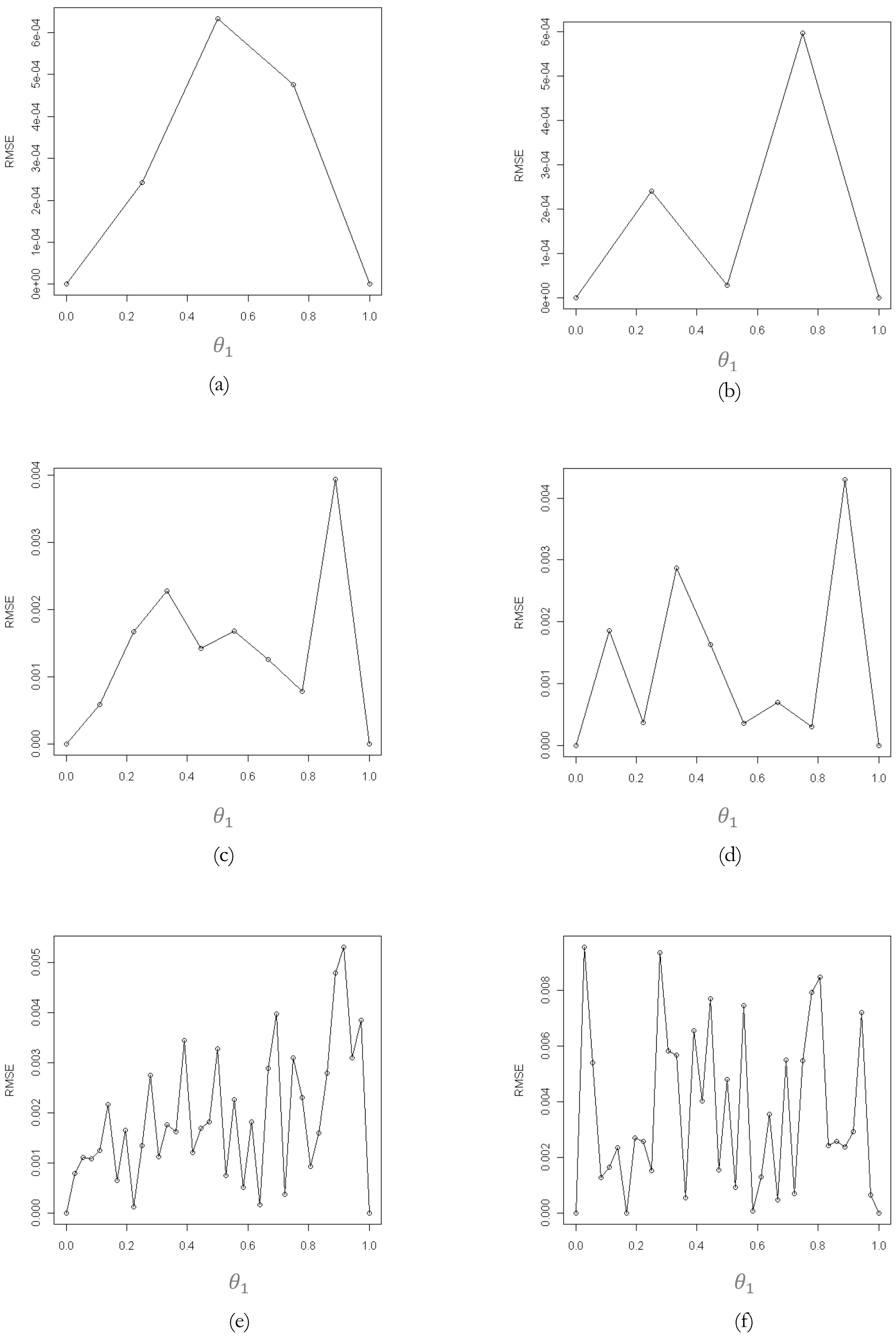


Figure 6-2 RMSE of mixture probability (a)S=2,classes are normally distributed (b)S=2,classes are exponentially distributed (c)S=3,classes are normally distributed (d)S=3,classes are exponentially distributed (e)S=6,classes are normally distributed (f)S=6,classes are exponentially distributed

6.2. Application of SRM-SVM

This section presents the results of incorporation SVM mixture model in MRF-SRM and analyse the results. First the result of applying the method on synthetic images and optimising the smoothness parameter is discussed and analysed in section 6.2.1. Furthermore the results from applying the method on real remote sensing data are presented in section 6.2.2.

6.2.1. Results from synthetic image

To implement SRM-SVM method several synthetic images in different scales and different class distributions were generated. For all of this images reference image prepared in fine resolution and with degradation of S^2 fine pixels (S is scale factor) as discussed in 5.1.1 the coarse resolution synthetic images are generated.

Label of pixels for the initial SRM is assigned randomly so the image is noisy, but by applying smoothness parameter, λ that discussed in section 5.3.3, the final SRM will appear smoother. This parameter controls relation between the values of prior and likelihood energy. As the accuracy for likelihood energy is different in each scale related to the scale of SRM (see the results of section 6.2.1) appropriate smoothness parameter is not the same in all scales. To optimise the appropriate value for smoothness parameter in the implementation value of λ changed in nine values 0.1, 0.2,…,0.9. When the value of λ is bigger the neighbourhood pixels has more effect on the class label so the final image will be smoother. The accuracy of the results is obtained by the kappa coefficient (see section 5.4).

To find the label of pixels simulated annealing (section 4.1.3) is used so the iterations continue until the system gets the freezing point. Number of iterations depends on the images, scale factor and smoothness parameter thus it is different in each application. The parameter for simulating annealing is initial temperature (T_0) that is optimized with values 0, 3 and the parameter for updating temperature (T_{upd}), If the initial temperature set as $T_0 = 3$ then it reduces after each iteration and select the SRM randomly, and By selecting $T_0 = 0$ the initial SRM is not selected randomly. In the following the results of implementation the method on *synthetic image1* and *synthetic image2* is discussed.

6.2.1.1. Synthetic image1

As mentioned in section 5.1.1 *Synthetic image1* has classes with normal distribution. The application of SRM-SVM on *synthetic image1* is in four scales 2, 4, 6 and 10. First the smoothness parameter in each scale was optimised. In this experiment the initial value of temperature is set as 3 and the effect of smoothness parameter on SRM-SVM is studied. The results are shown in Figure 6-3.

It can be obtained from the results that the value of kappa coefficient has negative relation with scale factor, and the reason is in higher scale the number of pixels are less, so the image does not have enough contextual information and the accuracy is lower.

As the application was done for identification of initial temperature for two values 0 and 3, the implementation of SRM-SVM is done again with combination of different smoothness parameter and different scale factors and $T_0 = 0$. The result of this experiment is shown in Figure 6-4.

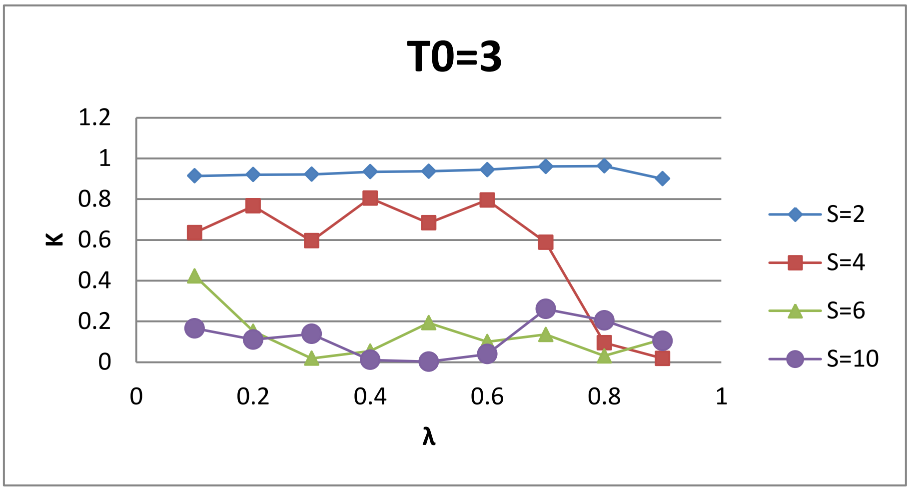


Figure 6-3 kappa coefficient of SRM results for normally distributed classes when T0=3

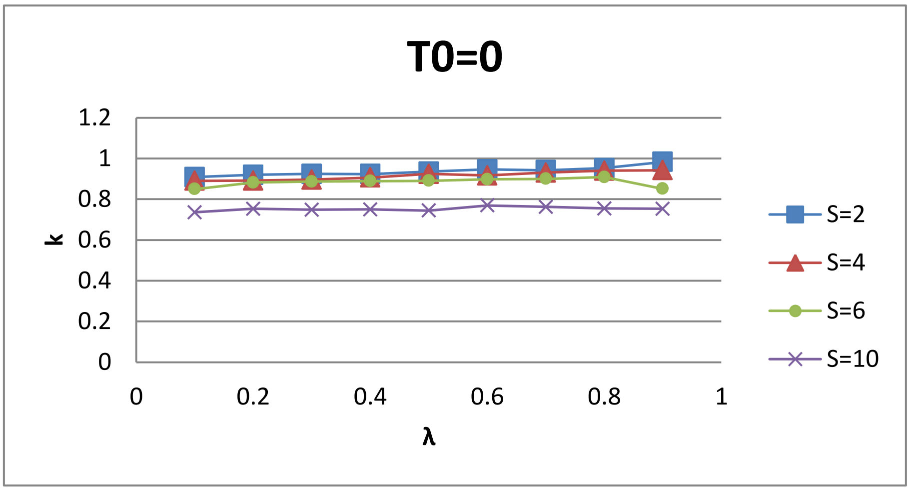


Figure 6-4 kappa coefficient of SRM results for normally distributed classes when T0=0

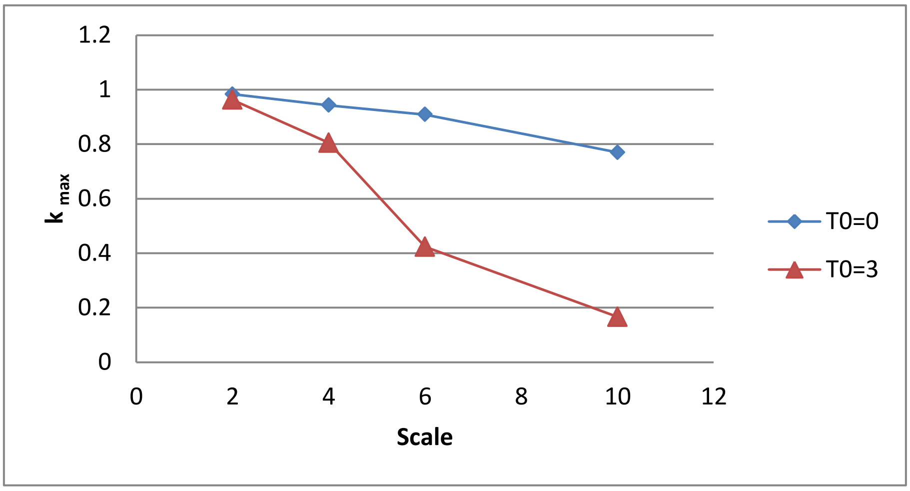


Figure 6-5 Compare the results of k max with different initial temperature

As it is obtained from the results, the range of k for different smoothing parameter is very close to the maximum obtained kappa coefficient so change the value of λ does not affect the accuracy of SRM-SVM. In Figure 6-4 the maximum kappa coefficient in each scale for different initial temperature compared to each other. From the results can be obtained that the better accuracy is optimised for $T_0 = 0$.

6.2.1.2. Synthetic image2

The main objective of this research is to apply SRM-SVM on images with non-normally distribution classes, therefore next step is to apply the method on *synthetic image2* with exponential distribution classes. From this image four coarse images were generated with scale 2, 4, 6 and 10 with. To apply SRM-SVM on *synthetic image2* the experimental results from application of method on *synthetic image1* were used, so the initial temperature for this synthetic set as zero. The results of implementation of SRM-SVM on *synthetic image2* with different smoothing parameters (0.1, 0.2,..., 0.9) is shown in Figure 6-6. The results show that the accuracy of application SRM-SVM on the exponential distribution data is acceptable accuracy.

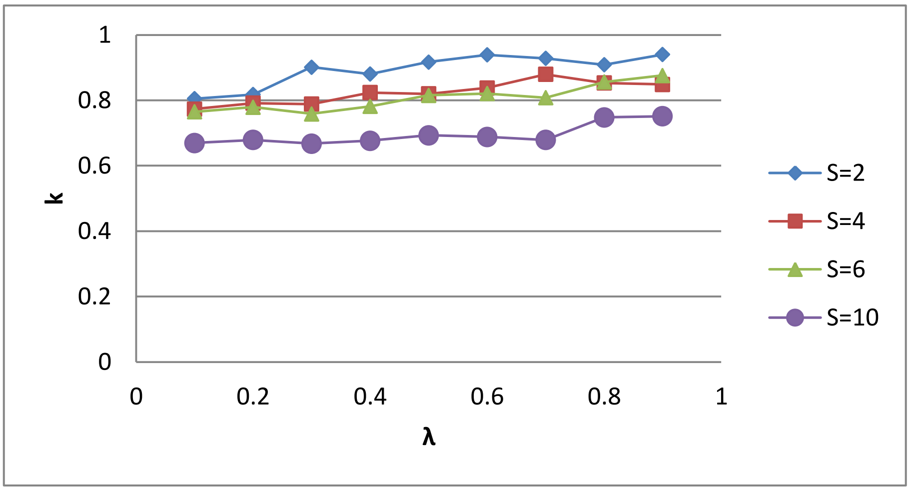


Figure 6-6 kappa coefficient value for SRM-SVM for exponential distribution classes

To compare the accuracy of SRM-SVM method with the MRF-SRM, MRF-SRM applied on *synthetic image2*. For this application coarse resolution images that is prepared in scale factor 2 is used. Figure 6-7 compares the results of MRF-SRM and SRM-SVM. As it observed from the results for the same value of λ the accuracy of SRM-SVM is higher than MRF-SRM.

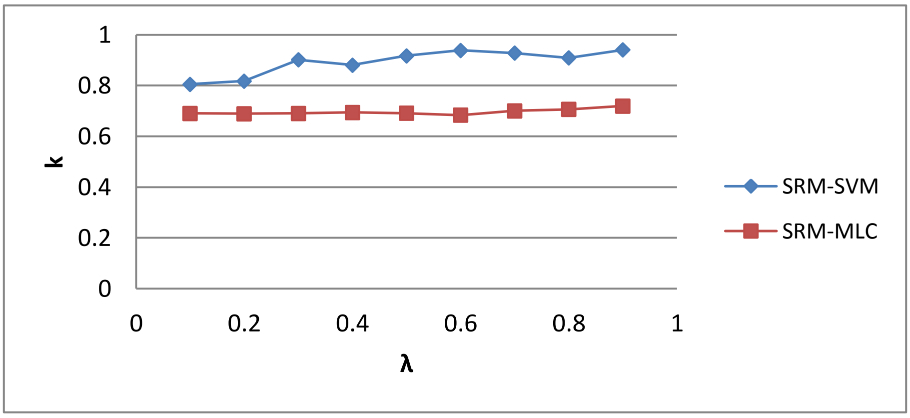
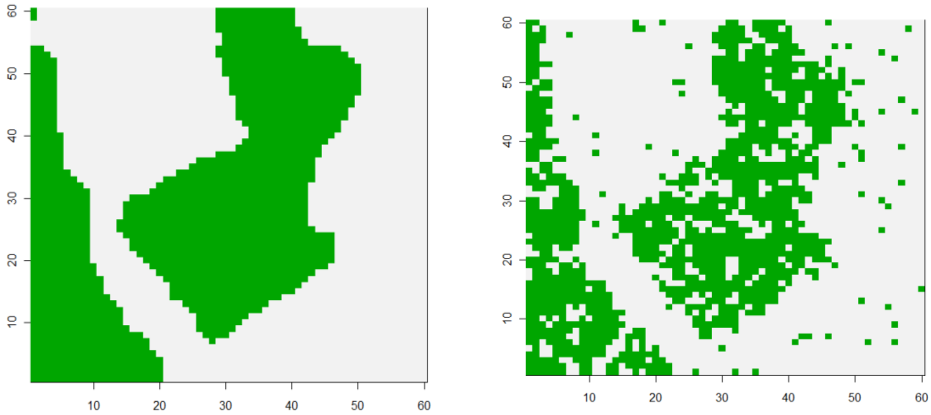


Figure 6-7 Results of SRM-SVM compare to the results of SRM-MLC for exponentially distributed classes with S= 2



(a) (b)

Figure 6-8 The results of SRM for S=2, (a) final SRM-SVM (b)final MRF-SRM

The maximum kappa coefficient for MRF-SRM observed at $\lambda = 0.9$ and for SRM-SVM maximum kappa is for $\lambda = 0.6$ the result of application of SRM for these smoothness parameter is shown in Figure 6-8. In both image the classes are exponentially distributed but the classification method is different. The result of classifying image with SVM is smoother than results from application of MLC classification.

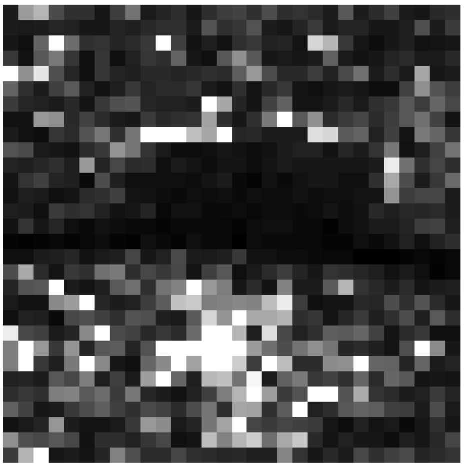
The results from *synthetic image2* show that the proposed method of SRM classification cooperated with SVM mixture probability is more appropriate for classes that are non-normally distributed. In the next section the results of application of the method on real image is discussed.

6.2.2. Results from real image

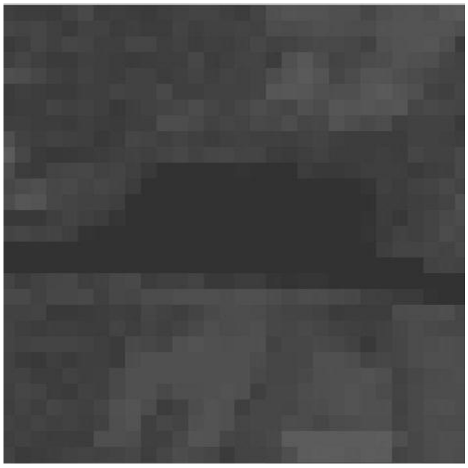
To test the ability of the proposed method on real data, the remote sensing data are prepared from radar image and optical image as discussed in section 3.1. The SRM method that is going to be used on this image is corporate with boundary sub-pixel which is used to show the smoother boundary of an object in finer spatial resolution.

The subsets are chosen from an area with a water body that is continued with canals in both sides (Figure 6-9). The aim is to find the boundary of the area that covers with water, therefore the water selected as one class and the other objects such as, urban area, vegetation, trees... selected as another class. Training set for both classes are selected from outside of these subsets.

For the ERS image from multi-looking is used to help for visualise and interpret the image. But still the values for the pixels are very big, so all the values from radar images contain training sets and values in the subset are divided by 1000. The ERS image set as band1 and spot image is band2. The two classes in band1 have exponential distributions but in band2 the class for water has normal distribution and the class for the other objects has multimodal distribution.



(a)



(b)

Figure 6-9 subsets (a) ERS subset (band1) (b) spot subset (band2)

To apply SRM-SVM the scale factor four is selected, with this scale the pixel size in final SRM is 5×5. The experimental results from last sections are used to set the parameter of SVM classification, smoothness parameter and initial temperature. SVM classification is done by linear kernel, smoothness parameter set as 0.7 and the initial temperature is zero. The numbers of iterations for SRM-SVM is 10. The result of application of SRM-SVM is shown in Figure 6-10. As it is observed in the image the boundaries are very smooth and the method classified subset accurate.

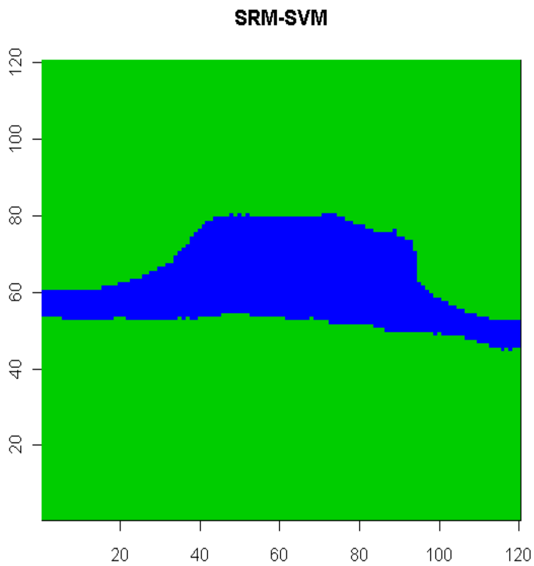


Figure 6-10 The results for SRM-SVM, S=4

Figure 6-11 is shown the SRM-SVM results compare to the shape file of the area, it shows that the method estimate the boundaries smooth and accurate. Figure 6-12 shows the result of implementation of SRM-MRF with assumption of normal distribution. The image compare to SRM-SVM has low quality and did not estimate the boundaries accurate.

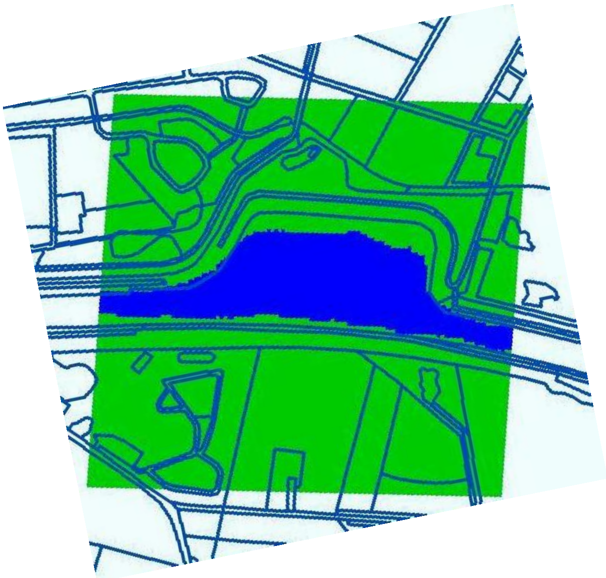


Figure 6-11Compare the results with the shape file

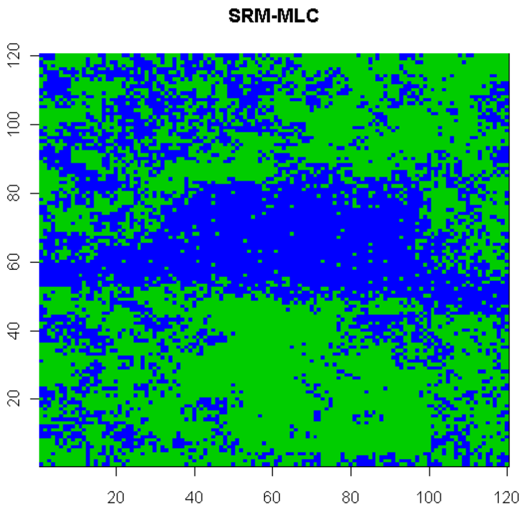


Figure 6-12 the results of SRM-MLC on the remote sensing image

7. DISCUSSION

This chapter discuss about the obtained results from chapter 5. In the first part the applicability of the SVM mixture probability method and its results is discussed in detail. Furthermore the results of implementing SVM mixture probability in MRF-SRM on different data are discussed.

The application is started by study on capability of selecting training set for super resolution mapping with SVM. From the results of applying SVM on two training sets of the same class but different spatial resolution it is observed that classifying coarser pixels with separating hyperplane of fine pixels (SHF) has small error. Additionally the results has very high accuracy for classifying fine pixels with separating hyperplane of coarse pixels (SHC) as shown in the Tables 6-2 and 6-3 of Chapter 6 for normally distributed classes and exponentially distributed classes. Both training sets have high classification accuracy but for classification coarse pixels with SHF this accuracy is identical because variance of coarse pixels is always smaller than finer resolution pixel and the mean for both of them is the same. Since the SHC and SHF are in the same direction (illustrated in Figure 5-1 of chapter 5) coarser data are always in the correct side of SHF. The results show that in the SRM-SVM the training set can be selected either from fine spatial resolution or from coarse spatial resolution data.

Figure 5-2 and Figure 5-3 shows the results of comparing the estimated SVM mixture probability with the same value from reference data, the accuracy is presented with RMSE. As it is obtained from the plots the accuracy for the set of data with normal distribution classes is higher than the data with exponentially distributed classes. The reason of this difference is that in the datasets that is used for this application, the histograms of distance for exponentially distributed classes have higher variance, when the variance of data is higher the width for the bins are larger and the distance between mids of bins is bigger, so the error in linear interpolation is higher. The results in the application for this research are accurate, but if for some classes the variance is big then with non-linear interpolation method better results will be obtained for estimation of mixture probability.

The applicability of proposed method is tested on two synthetic images with different class distributions and remote sensing image from different sources. The results of application SRM-SVM on both synthetic images show acceptable accuracy for the SRM-SVM method. The accuracy of classification is decreased for higher scale factor. This result is expected for MRF-SRM technique according to previous studies (Tolpekin & Stein, 2009). In SRM-SVM method the range of changing the kappa coefficient with different λ is not very different from maximum accuracy. It is observed from the results that the accuracy of final SRM is higher when initial temperature set as zero, with this value initial SRM does not select randomly and iteration starts with local minimum energy, it also has less number of iteration than higher temperatures.

In the similar way the application is repeated for synthetic image2. The effect of lower accuracy for mixture probability and also the small class separability for exponential distribution (Tolpekin & Stein, 2009) is the reason that the accuracy of SRM-SVM for each λ with exponential distribution is smaller than the accuracy the same λ for normal distribution. The sum of data with exponential distribution is not exponentially distributed and it will become Gamma distribution, by increasing the number of exponential data the distribution of data become normal distribution so by increasing the scale factor in *synthetic imag2* that is generated by degradation the accuracy is become the same as accuracy of normal distribution. Comparing the application of SRM-SVM method with another MRF-SRM method that use maximum

likelihood in its assumption shows that the value of accuracy is higher for all λ in SRM-SVM classification method, this is a good reason for suitably of SRM-SVM for the non-normal classes.
Figure 6-10 shows the performance of SRM-SVM on remote sensing image from different sources, radar and optical data. Comparing the result of this application with the results from MRF-SRM in figure 6-12 the improvement of accuracy and smoothness in the result from SRM-SVM is observed. This performance shows the capability of method for real data.

8. CONCLUSION AND RECOMMENDATIONS

8.1. Conclusion

The main objective of this study is to improve MRF-based SRM technique by incorporating SVM classification method to archive an SRM method that is suitable for all kinds of distributions. In order to address the research objective four research questions are posted and answered during the research. The mixture model estimate with SVM classification and then incorporate in MRF-SRM algorithm. In addition the parameter for this new algorithm estimate and the results accuracy assessment is done on the obtained results. Furthermore the application is done on two synthetic images with different distributions and one remote sensing image.

Before apply the method on the data capability of SVM to choose training set of different spatial resolution images from one class consider. This study is done on two sets of training samples with the same parameters as synthetic image1 and synthetic image2. The accuracy is identical for classification of data from one class with the training set of the same class in different resolution.

Additionally mixture probability with SVM classification estimated and tested. To propose the method distance of pixels from hyperplane in feature space is used. The results show that however the accuracy is lower for bigger scale factor, the method can estimate mixture probability of a mixed pixel with very high accuracy for any kind of class distribution. It is observed that the training data set for SRM incorporated with SVM can be selected either from coarse resolution or fine resolution pixels.

The method is tested on synthetic image1 with normal distribution and synthetic image2 with exponential distribution. Optimal parameter of SRM-SVM is observed for each synthetic image in different scales. Maximum accuracy is not the same for two synthetic images, but in both images it is more than 0.7 so the result is acceptable for both images. The results compared with the result of application MRF-SRM with MLC classification on synthetic image2. The observed results for SRM-SVM in the same scale factor has better accuracy than MRF-SRM with MLC classification. The maximum value for kappa coefficient for image with SRM-SVM application observed in $\lambda = 0.6$ and for image with MRF-SRM with MLC classification observed in $\lambda = 0.9$. Comparing the two final SRM maps from these applications in maximum kappa shows that the final SRM with SVM classification is smoother.

The SRM-SVM method is also applied on a real image from different sources, one radar image and one optical image so the data has two bands with different distributions. This application was done to estimate the boundary of an object in larger scale. The result of SRM-SVM shows very smooth boundary for the final map.

In conclusion the SRM-SVM method does not make the assumption of class distribution and can be applied in any kinds of image with normal or non-normal distributed classes. It converts the multiband data to single band size by using distance from hyperplane that makes the process faster, and as it uses contextual method in classification the SRM result is smooth with high classification accuracy.

8.2. Recommendation

The SRM-SVM method is a new and accurate technique, but there is more that can be done on this study, the following are recommended for further research:

1. Since SVM classification originally is binary classification method, there are some methods that improve SVM classification to a multiclass classification method. In this research the original

SVM was used and data is selected with two classes. Therefore it recommended that improve the research for multiclass classification.

2. During the study the influence of some parameters of MRF-SRM on the accuracy of the results was studied, it is necessary to consider the user-defined parameter of SVM such as parameter C and ν after incorporation with MRF-SRM.

3. The SRM-SVM method that applied for real image considered sub-pixel boundary, so it is recommended to apply the method for other type of sub-pixels.

LIST OF REFERENCES

Agency, E. S. ERS. Retrieved 02/05/2011, from http://www.esa.int/esaEO/SEMGWH2VQUD_index_0_m.html

Banks, J. (2000). *Getting started with AutoMod.*

Bastin, L. (1997). Comparison of fuzzy c-means classification, linear mixture modelling and MLC probabilities as tools for unmixing coarse pixels. *International Journal of Remote Sensing, 18*(17), 3629-3648.

Boucher, A., & Kyriakidis, P. (2006). Super-resolution land cover mapping with indicator geostatistics. *Remote Sensing of Environment, 104*(3), 264-282.

Boucher, A., Kyriakidis, P., & Cronkite-Ratcliff, C. (2008). Geostatistical solutions for super-resolution land cover mapping. *IEEE Transactions on Geoscience and Remote Sensing, 46*(1), 272-283.

Bovolo, F., Bruzzone, L., & Carlin, L. (2010). A Novel Technique for Subpixel Image Classification Based on Support Vector Machine. *IEEE Transactions on Image Processing, 19*(11), 2983-2999.

Brown, M., Gunn, S. R., & Lewis, H. G. (1999). Support Vector Machines for Optimal Classification and Spectral Unmixing. *Ecological Modelling, 120*, 167-179.

Bruzzone, L., & Persello, C. (2009). A Novel Context-Sensitive Semisupervised SVM Classifier Robust to Mislabeled Training Samples. *IEEE Transactions on Geoscience and Remote Sensing, 47*(7), 2142-2155.

Burges, C. J. C. (1998). A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery, 2*(2), 121-167.

corporation, S. I. Spot-5. Retrieved 2001/02/05, from http://www.satimagingcorp.com/satellite-sensors/spot-5.html

Ferretti, A., Monti-Guarnieri, A., Prati, C., Rocca, F., & Massonet, D. (2007). *InSAR Principles-Guidelines for SAR Interferometry Processing and Interpretation* (Vol. 19).

Fisher, P. (1997). The pixel: a snare and a delusion. *International Journal of Remote Sensing, 18*(3), 679-685.

Fjortoft, R., Delignon, Y., Pieczynski, W., Sigelle, M., & Tupin, F. (2003). Unsupervised classification of radar images using hidden Markov chains and hidden Markov random fields. *IEEE Transactions on Geoscience and Remote Sensing, 41*(3), 675-686.

Foody, G. (2006). Sub-Pixel Methods in Remote Sensing. In S. M. d. Jong & F. D. v. d. Meer (Eds.), *Remote Sensing Image Analysis* (pp. 37-49): Springer.

Foody, G., & Mathur, a. (2004a). A relative evaluation of multiclass image classification by support vector machines. *IEEE Transactions on Geoscience and Remote Sensing, 42*(6), 1335-1343.

Foody, G., & Mathur, A. (2004b). Toward intelligent training of supervised image classifications: directing training data acquisition for SVM classification. *Remote Sensing of Environment, 93*, 107-117.

Goumehei, E. (2010). *Contextual image classification with support vector machine.* University of Twente Faculty of Geo-Information and Earth Observation ITC, Enschede.

Haglund, A. (2000). *Towards soft classification of satellite data.*

Hastie, T., Tibshirani, R., & Friedman, J. H. (2003). *The Elements of Statistical Learning.* California: Springer.

Hsu, C., & Lin, C. (2002). A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks, 13*(2), 415-425.

Hu, R., & Fahmy, M. (1992). Texture segmentation based on a hierarchical Markov random field model. *Signal processing, 26*(3), 285-305.

Huang, C., Davis, L. S., & Townshend, J. R. G. (2002). An assessment of support vector machines for land cover classification. *International Journal of Remote Sensing, 23*(4), 725-749.

An introduction to R. Retrieved 15/02/2011, from http://cran.r-project.org/doc/manuals/R-intro.html

Karatzoglou, A., Meyer, D., & Hornik, K. (2006). Spport Vector Machines in R. *Journal of Statistical Software, 15*(9), 1-28.

Kasetkasem, T., Arora, M. K., & Varshney, P. K. (2005). Super-Resolution Land Cover Mapping Using a Markov Random Field Based Approach Remote Sensing of Environment. *Remote Sensing of Environment, 96*(3-4), 302-314.

Kasetkasem, T., & Varshney, P. K. (2002). An Image Change Detection Algorithm Based on Markov Random Field Models. *IEEE Transactions on Geoscience and Remot Sensing, 40*(8), 1815-1823.

Kassaye, R. H. (2006). *Suitability of Markov random field based method for super resolution land cover mapping.* International Institute for geo information Sience and earth Observation(ITC), Enschede.

Keshava, N. (2003). A survey of spectral unmixing algorithms. *Lincoln Laboratory Journal, 14*(1), 55-78.

Li, S. (2009). *Markov random field modeling in image analysis*. New York: Springer.
Lin, C. (2002). Fuzzy support vector machines. *IEEE Transactions on Neural Networks, 13*(2), 464-471.
Madevska-Bogdanova, A., Nikolik, D., & Curfs, L. (2004). Probabilistic SVM outputs for pattern recognition using analytical geometry. *Neurocomputing, 62*, 293-303.
Mather, P. M., & Koch, M. (1987). *Computer processing of remotely sensed images : an introduction*: Wiley & Sons.
Melgani, F., & Serpico, S. (2003). A Markov random field approach to spatio-temporal contextual image classification. *IEEE Transactions on Geoscience and Remote Sensing, 41*(11), 2478-2487.
Mertens, K., Verbeke, L., Ducheyne, E., & De Wulf, R. (2003). Using genetic algorithms in sub-pixel mapping. *International Journal of Remote Sensing, 24*(21), 4241-4247.
Moler, C. (2004). *Numerical computing with MATLAB*: Society for Industrial Mathematics.
MRF.
Platt, J. C. (1999). Probabilistic outputs for support vector machines and Comparisons to Regularized Likelihood method. 61–74.
Richards, J. A., & Jia, X. (2006). *Remote sensing digital image analysis : an introduction* (Fourth edition ed.). Berlin etc.: Springer-Verlag.
Sarkar, A., Biswas, M., Kartikeyan, B., Kumar, V., Majumder, K., & Pal, D. (2002). A MRF model-based segmentation approach to classification for multispectral imagery. *IEEE Transactions on Geoscience and Remote Sensing, 40*(5), 1102-1113.
Schott, J. R. (2007). *Remot Sensing:The Image Chain Approach*. New York: Oxford University Press.
Solberg, A. H. S., Taxt, T., & Jain, A. K. (1996). A Markov Random Field Model for Classification of Multisource Satellite Imagery. *IEEE Transactions on Geoscience and Remote Sensing 34*(1), 13.
Tatem, A., Lewis, H., Atkinson, P., & Nixon, M. (2003). Research Article Increasing the spatial resolution of agricultural land cover maps using a Hopfield neural network. *International Journal of Geographical Information Science, 17*(7), 647-672.
Tatem, A., Lewis, H. G., Atkinson, P. M., & Nixon, M. S. (2001). Super-Resolution Target Identification from Remotely Sensed Images Using a Hopfield Neural Network. *IEEE Transactions on Geoscience and Remot Sensing, 39*(4), 781-797.
Tolpekin, V. A., & Stein, A. (2009). Quantification of the Effects of Land-Cover-Class Spectral Separability on the Accuracy of Markov-Random-Field-Based Superresolution Mapping. *IEEE Transactions on Geoscience and Remot Sensing, 47*(9), 3283-3297.
Tough, R. J. A., Blacknell, D., & Quegan, S. (1995). A Statistical Description of Polarimetric and Interferometric Synthetic Aperture Radar Data. *Proceedings: Mathematical and Physical Sciences, 449*(1937), 567-589.
Tso, B., & Mather, P. M. (2001). *Classification methods for remotely sensed data*. London etc.: Taylor & Francis.
Tso, B., & Olsen, R. (2005). A contextual classification scheme based on MRF model with improved parameter estimation and multiscale fuzzy line process. *Remote Sensing of Environment, 97*(1), 127-136.
Vapnik, V., & Cortes, C. (1995). Support-vector networks. *Machine learning, 20*(3), 273-297.
Verhoeye, J., & De Wulf, R. (2002). Land cover mapping at sub-pixel scales using linear optimization techniques. *Remote Sensing of Environment, 79*(1), 96-104.
Watanachaturaporn, P., Varshney, P., & Arora, M. (2004). Evaluation of factors affecting support vector machines for hyperspectral classification. *The American Society for Photogrammetry & Remote Sensing ASPRS*.
Weng, R. C., Lin, H. T., & Lin, C. J. (2001). *A Note on Platt's Probabilistic Outputs for Support Vector Machines*.
Zhu, H. (2005). Linear spectral unmixing assisted by probability guided and minimum residual exhaustive search for subpixel classification. *International Journal of Remote Sensing, 26*(24), 5585-5602.

APPENDIX1

Preparing SVM mixture model in R

```
#Path
rm(list = ls())
require(MASS)
require(mvtnorm)
require(pixmap)
require(scatterplot3d)
require(kernlab)
require(GLDEX)
Path <- 'D:\\Data_Prepration\\'
#
# Read the training set data: determine class means and covariances
#
# Path to training set files
Path_ts <- paste(Path,'Statistical_Report_2classes_2bands_test\\',sep='')
Filename <- 'Mean.txt'
Inputfile <- paste(Path_ts,Filename,sep='')
temp <- read.table(Inputfile, skip = 1)
d <- dim(temp)
#Scale Factor
S<-2
# File dimensions
# Number of bands
Nb <- d[1]
# Number of classes
Ncl <- d[2]
mu   <- array(rep(0,Ncl*Nb),c(Ncl,Nb))
Cov  <- array(rep(0,Ncl*Nb*Nb),c(Ncl,Nb,Nb))
Cinv <- array(rep(0,Ncl*Nb*Nb),c(Ncl,Nb,Nb))
mut <- data.matrix(temp)
mu[,] <- t(mut)
# Read covariance matrices
for(k in 1:Ncl)
{
  Inputfile <- paste(Path_ts, 'Cov_',k,'.txt', sep='')
  temp <- read.table(Inputfile, skip = 0)
  Cov[k,,] <- as.matrix(temp)
  for(i in 1:(Nb-1))
  {
    for(j in (i+1):Nb)
    Cov[k,i,j]<-Cov[k,j,i]
  }
}
```

```
#*******************************************************************************************
# generate random numbers from exponential distribution
#*******************************************************************************************
rand_exp <- function(n,sigma)
{
  xt <- rnorm(n,0,sqrt(0.5*sigma))
  yt <- rnorm(n,0,sqrt(0.5*sigma))
  I <- xt^2 + yt^2
  return(I)
}
#*************************************************************************************
#Preparing DN value for Fine resolution, Dx and Coarse resolution, Dy Images
#*************************************************************************************
#Normal classes
#number of Samples
Nsample<-1000
Dx<-array(0,c(0,Nb))
Dy<-array(0,c(0,Nb))

for(k in 1:Ncl)
{
	DN<-array(0,c((S^2),Nb))
	Dxc<-array(0,c(0,Nb))
	Dyc<-array(0,c(0,Nb))
	for(i in 1:Nsample)
	{
		for(j in 1:(S^2))DN[j,] <- rmvnorm(1, mu[k,], Cov[k,,])
		Dxc<-rbind(Dxc,DN)
		Dyc<-rbind(Dyc,c(sum(DN[,1])/(S^2),sum(DN[,2])/(S^2)))
	}
	Dx<-rbind(Dx,Dxc)
	Dy<-rbind(Dy,Dyc)
}
#Non-normal classes
Nsamplex<-Nsample/4
NDx<-array(0,c(0,Nb))
NDy<-array(0,c(0,Nb))
for(k in 1:Ncl)
{
	NDN<-array(0,c((S^2),Nb))
	NDxc<-array(0,c(0,Nb))
	NDyc<-array(0,c(0,Nb))
	for(i in 1:Nsample)
	{
		NDN[,1] <-rand_exp((S^2),mu[k,1])
		NDN[,2] <-rand_exp((S^2),mu[k,2])
```

```
                    NDxc<-rbind(NDxc,NDN)
                    NDyc<-rbind(NDyc,c(sum(NDN[,1])/(S^2),sum(NDN[,2])/(S^2)))
          }
          NDx<-rbind(NDx,NDxc)
          NDy<-rbind(NDy,NDyc)
}

#*********************************************************************************
# SVM classification of the image Dy (Coarse resolution image)
#*********************************************************************************
# SVM training
# SVM parameters
sigma_SVM <-1
C_SVM <- 10
nu_SVM <- 0.9

Trainingsety<- data.frame(B1=Dy[,1],B3=Dy[,2],class=c(rep(-1,Nsample),rep(1,Nsample)))
#NTrainingsety<- data.frame(B1=NDy[,1],B3=NDy[,2],class=c(rep(-1,Nsample),rep(1,Nsample)))
# Linear kernel
svm_modely<- ksvm(class~.,data=Trainingsety,scaled=FALSE,type="C-
svc",kernel="vanilladot",C=C_SVM,prob.model=TRUE)
#Radial Basis kernel "Gaissian"
#svm_modely <- ksvm(class~.,data=NTrainingsety,scaled=TRUE,type="C-svc",cache =
2000,kernel="rbfdot",kpar=list(sigma=sigma_SVM),C=C_SVM,prob.model=TRUE)

#***********************
#Calculate W & B for Dy
#***********************
#Number of SVMs
LenIy<-nSV(svm_modely)

#Value of alpha
alphaiy<-alpha(svm_modely)[[1]]

#Index of SVMs
SViy<-alphaindex(svm_modely)[[1]]

#The value of the SVMs
DSVMy<-as.vector(Trainingsety[SViy,,])

#Class label of each SVMs
yy<-DSVMy[3][[1]]

#Value fo SVMs in trainingset
xy<-array(0,c(LenIy,Nb))
for(k in 1:LenIy)
```

```
xy[k,]<-c(DSVMy[k,1],DSVMy[k,2])

#Biased for hyperplane
By<-b(svm_modely)

#Objective value of SVM classification
OB<-obj(svm_modely)

#weight for hyperplane
W<-sqrt(2*(OB+sum(alphaiy)))

#Function for RBF kernel
RBFK<-rbfdot(sigma=sigma_SVM)

#Function for linear kernel
LIK<-vanilladot()

#Kernel value
FK<-array(0,c(2*Nsample,LenIy))

#Desicion function for each data
Fst<-array(0,c(2*Nsample,LenIy))

#Distance from hyperplane(for pure training set)
Dist<-array(0,2*Nsample)
Dist1<-array(0,2*Nsample)

#the LIK should change for RBF kernel if the kernel change
for(i in 1:LenIy)
{
        for(j in 1:(2*Nsample))
        {
                FK[j,i]<-LIK(xy[i,],Dy[j,])
                Fst[j,i]<-yy[i]*alphaiy[i]*FK[j,i]
        }
        Dist1<-Dist1+Fst[,i]
}

Dist<-(Dist1-By)/W


Distance<-function(D){
                                #Kernel value
                                F<-array(0,LenIy)

                                #Desicion function for each data
```

```
                Fs<-array(0,LenIy)

                for(k in 1:LenIy)
                {
                        F[k]<-LIK(xy[k,],D)
                        Fs[k]<-yy[k]*alphaiy[k]*F[k]
                }

                        Dist<-(sum(Fs)-By)/W
                        return(Dist)
                }

#************************************************************
#Mixed pixels
#Array for Distances
Disty<-array(0,c(Nsample,(1+S^2)))
#Mean of Deistances
MDisty<-array(0,(1+S^2))
#Fractions
NPc <- array(0,c(Ncl,(1+S^2)))
for(ep in 1:(1+S^2))
{
#Number of pixels of each class in one coarse pixel
NPc[1,ep]<- ep-1
NPc[2,ep]<- (S^2)-NPc[1,ep]
Arr1 <- array(0,c(NPc[1,ep],Nb))
Arr2 <- array(0,c(NPc[2,ep],Nb))
Dmix  <- array(0,c(0,Nb))
Sum <- array(0,c(1,Nb))
for(k in 1:Nsample)
{
        if(NPc[1,ep]!=0){
                Arr1 <- rmvnorm(NPc[1,ep],mu[1,],Cov[1,,])}
        if(NPc[2,ep]!=0){
                Arr2 <- rmvnorm(NPc[2,ep],mu[2,],Cov[2,,])}
        for(j in 1:Nb)Sum[,j]<-(sum(Arr1[,j])+sum(Arr2[,j]))/(S^2)
        Dmix<-rbind(Dmix,Sum)
}
#*********************************************************************
#Distance from Coarse SVM line
#*********************************************************************
#Distance of mixed pixels from SVM line
xFm<-as.vector(Dmix[,1])
yFm<-as.vector(Dmix[,2])
for(i in 1:Nsample)Disty[i,ep]<-Distance(Dmix[i,])
MDisty[ep]<-mean(Disty[i,ep])
```

```
}
#*****************************************************************
#Codes for mixing probabilities
#*****************************************************************
#The number of breaks(for ploting histograms)
bmin<-floor(min(min(Dist[1:Nsample]),min(Dist[(Nsample+1):(2*Nsample)])))
bmax<-ceiling(max(max(Dist[1:Nsample]),max(Dist[(Nsample+1):(2*Nsample)])))
Bin<-(bmin:bmax)
xlimy<-c(bmin,bmax)
#Error
RMSE<-array(0,(1+S^2))
#Number of bins
NBin=10
#porbability for each proportions
P1<-array(0,c(NBin,(1+S^2)))
P2<-array(0,c(NBin,(1+S^2)))
P<-array(0,c(NBin,(1+S^2)))
Pm<-array(0,c(NBin,(1+S^2)))
#set mids of histograms as x axis for approximation
M1<-array(0,c(NBin,(1+S^2)))
M2<-array(0,c(NBin,(1+S^2)))
M<-array(0,c(NBin,(1+S^2)))
Mm<-array(0,c(NBin,(1+S^2)))
#approximate probability
PA<-array(0,c(NBin,(1+S^2)))
for(i in 1:(1+S^2))
{
        H1<-hist.su(Disty[,(1+S^2)],col=2,prob=TRUE,nclass=NBin,plot=FALSE)
        H2<-hist.su(Disty[,1],col=3, prob=TRUE,nclass=NBin,plot=FALSE)
        H3<-hist.su(Disty[,i],col=4, prob=TRUE,nclass=NBin,plot=FALSE)
        Bin1<-(H1$breaks[2]-H1$breaks[1])
        Bin2<-(H2$breaks[2]-H2$breaks[1])
        Bin3<-(H3$breaks[2]-H3$breaks[1])
        P1[,i]<-(H1$density)*Bin1
        P2[,i]<-(H2$density)*Bin2
        P[,i]<-(H3$density)*Bin3
        M1[,i]<-H1$mids
        M2[,i]<-H2$mids
        M[,i]<-H3$mids
        Pm[,i]<-frac[1,i]*P1[,i]+frac[2,i]*P2[,i]
        Mm[,i]<-frac[1,i]*M1[,i]+frac[2,i]*M2[,i]
        PA[,i]<-approx(Mm[,i],Pm[,i], xout=M[,i])$y
        PA[is.na(PA)] <- 0
        RMSE[i]<-sqrt(((sum(P[,i]-PA[,i]))^2)/NBin)
}
```

APPENDIX2

The codes for SRM-SVM in R
(These codes only include the likelihood energy with SVM)

```
#
# Mixing distributions
#
Nmix <- 1+S^2
#Histograms for pure classes
H1<-hist.su(Distc1,col=2,prob=TRUE,nclass=100,plot=FALSE)
H2<-hist.su(Distc2,col=3, prob=TRUE,nclass=100,plot=FALSE)
#Bins of the histograms
Bin1<-(H1$breaks[2]-H1$breaks[1])
Bin2<-(H2$breaks[2]-H2$breaks[1])
P1<-(H1$density)#*Bin1
P2<-(H2$density)#*Bin2
#mids of each bin
M1 <- H1$mids
M2 <- H2$mids
Nbins <- length(P1)
P_mix <- array(0,c(Nmix,Nbins))
M_mix <- array(0,c(Nmix,Nbins))
for(k_mix in 1:Nmix)
{
  frac_k <- (k_mix-1)/(S^2)

  P_mix[k_mix,] <- P1*frac_k+P2*(1-frac_k)
  M_mix[k_mix,] <- M1*frac_k+M2*(1-frac_k)
}
######################################################
# Energy functions
#Prior Energy
I <- function(x,y){
        val <- 1
        if(x==y) val <- 0
        return(val)
}
xS <- function(x){
        val <- ceiling(x/S)
        return(val)
}
Frac_update <- function(i,j)
{
        val <- array(rep(0,Ncl),Ncl)
        for(ki in 1:S)
        for(kj in 1:S)
```

```
            {
              cln <- F[(i-1)*S+ki,(j-1)*S+kj]
              val[cln] <- val[cln] + 1
            }
            val <- val / (S^2)
            return(val)
}
Uprior <- function(i,j){
            F1 <- F[(Neigh_Coord[i,j,1]):(Neigh_Coord[i,j,2]),(Neigh_Coord[i,j,3]):(Neigh_Coord[i,j,4])]
            W1 <- Weight[(Neigh_Coord[i,j,1]-i+1+WSize):(Neigh_Coord[i,j,2]-
            i+1+WSize),(Neigh_Coord[i,j,3]-j+1+WSize):(Neigh_Coord[i,j,4]-j+1+WSize)]
            F0 <- F1 - F[i,j]
            F0[F0!=0] <- 1
            val <-  sum(W1 * F0)
            return(val)
}
Ulikelihood <- function(i,j){
            CProbB[xS(i),xS(j),] <- Frac_update(xS(i),xS(j))
            y0 <- Ddeg[xS(i),xS(j),]
            Distan<-Distance(y0)
            frac_k <- 1 + CProbB[xS(i),xS(j),1]*(S^2)

            if( (Distan >= min(M_mix[frac_k,]))&&(Distan <= max(M_mix[frac_k,])) )
            {


              val<-approx(M_mix[frac_k,],P_mix[frac_k,], xout=Distan)$y
            }else val <- 0

            if(val<1e-6) val <- 1e-6
            val <- -log(val)
# New: normalize the conditional energy
            val <- val / Norm_like
            return(val)
}
```